



Faculty of Engineering

**Model Predictive Controller (MPC) Design for an UAV Quadcopter in
Windy Critical Environment**

Mustapha Amine Sadi

**Master of Engineering
2025**

Model Predictive Controller (MPC) Design for an UAV Quadcopter in Windy Critical Environment

Mustapha Amine Sadi

A thesis submitted

In fulfillment of the requirements for the degree of Master of Engineering

(Electronic Engineering)

Faculty of Engineering
UNIVERSITI MALAYSIA SARAWAK

2025

DECLARATION

I declare that the work in this thesis was carried out in accordance with the regulations of Universiti Malaysia Sarawak. Except where due acknowledgements have been made, the work is that of the author alone. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.



.....

Signature

Name: Mustapha Amine Sadi

Matric No.: 20020087

Faculty of Engineering

Universiti Malaysia Sarawak

Date: 20 January 2025

ACKNOWLEDGEMENT

This thesis stands as a milestone in my academic journey, not merely as a reflection of my work but as a testament to the collaborative spirit and guidance that have brought it to fruition. At this moment of accomplishment, my heart is brimming with gratitude for those who have been pillars of support and sources of inspiration throughout this endeavour.

Foremost, I extend my deepest and most heartfelt appreciation to my supervisor, Dr. Annisa binti Jamali, whose intellectual rigor, scholarly insight, and nurturing guidance have been the compass by which this research was steered. Her profound expertise in Control System and Robotics and her rich cultural insights, have not only enriched this study but have also expanded my intellectual horizons and professional acumen. Dr. Annisa's unwavering support, patience, and encouragement in the face of challenges have been nothing short of inspirational. Her dedication to excellence and her genuine investment in my growth as a scholar has left an indelible mark on my personal and professional development. For this, I am eternally grateful.

My gratitude extends to University Malaysia Sarawak and Faculty of Engineering for providing the resources, environment, and community that have been foundational to my research. The support from fellow students, faculty, and administrative staff created a nurturing academic ecosystem that fostered learning, innovation, and collaboration.

I am profoundly thankful to my family, particularly My Mother Fairouz and Grandmother Yamina, whose unconditional love, endless patience, and unwavering faith in my abilities have been my strength. Their sacrifices and support have been the bedrock of my resilience and perseverance.

Special thanks go to: Dr. Abang Nizam from Mechanical Department for his guidance and support throughout this research, Prof. Omar my co-supervisor for all his support and guidance and help, Prof Sinin and his respected family for their help and warm hosting during all the times I visited their house. My friends in Algeria: Anis, Ismail and Zinou, my Friends and peers in Malaysia, especially Mostafa Ouartiou for his support and guidance since the first day I reached Malaysia, Imtiyaz, Syafiq, Rahman and Aysha for their companionship, understanding, and moments of respite and laughter amidst the rigorous demands of academic life. Their solidarity and encouragement have been a source of comfort and motivation.

In closing, while this thesis bears my name, it is a collective triumph that speaks to the generosity, wisdom, and spirit of all who have been a part of this journey. To all, I extend my deepest gratitude.

ABSTRACT

Unmanned Aerial Vehicles (UAVs), particularly quadcopters, have become indispensable in navigating critical windy environments, offering unparalleled access to complex terrains. These challenging environments pose unique navigational difficulties due to fluctuating wind patterns, which significantly impact UAV performance and accuracy. This research enhances UAV navigation effectiveness and stability through the development of a robust controller—specifically, a cascade Model Predictive Control (MPC) algorithm, optimized for continuous gust conditions as modelled by the Dryden wind turbulence approach. To accurately capture UAV dynamics within these environments, a mathematical model was developed using the Newton-Euler technique. This approach enabled precise analysis of both translational and rotational motions, factoring in elements such as inertia, torque, and external forces like drag and lift. The cascade MPC was further validated through real-time experiments, adapting UAV dynamics to unpredictable turbulence patterns and obstructions. This system's adaptability and precision are contrasted with a Proportional-Integral-Derivative (PID) controller, highlighting MPC's predictive advantages for dynamic environments. Experiments were conducted in mangrove forests, settings characterized by complex wind patterns, using a UAV equipped with a Pixhawk hardware controller. The UAV followed predefined paths to collect data on altitude, roll, pitch, yaw, and responses to wind turbulence, enabling refined control strategies and model improvements based on real-world performance. Comparative analysis shows that the quadcopter, when equipped with MPC, not only achieved better stability and responsiveness but also improved operational efficiency under adverse weather conditions, demonstrating the system's robustness and applicability to challenging UAV operations.

This research thus advances UAV deployment in demanding real-world applications by integrating sophisticated dynamic modelling with innovative control techniques.

Keywords: UAV, quadcopter, PID controller, intelligent controller, external disturbance, MPC controller, pixhawk

Pemodelan dan Reka Bentuk Kawalan Mantap UAV Quadcopter untuk Ketinggian dan Sikap dalam Persekitaran Kritikal Berangin

ABSTRAK

Kenderaan Udara Tanpa Pemandu (Unmanned Aerial Vehicles - UAV), khususnya kuadkopter, telah menjadi sangat diperlukan dalam menavigasi persekitaran berangin yang kritikal, memberikan akses yang tiada tandingan ke kawasan yang kompleks. Persekitaran yang mencabar ini menimbulkan kesukaran navigasi yang unik disebabkan oleh pola angin yang berubah-ubah, yang memberi kesan besar terhadap prestasi dan ketepatan UAV. Penyelidikan ini meningkatkan keberkesanan dan kestabilan navigasi UAV melalui pembangunan pengawal yang kukuh—khususnya algoritma Kawalan Prediktif Model (Model Predictive Control - MPC) jenis kaskad, yang dioptimumkan untuk keadaan hembusan angin berterusan seperti yang dimodelkan oleh pendekatan pergolakan angin Dryden. Untuk menangkap dinamik UAV dengan tepat dalam persekitaran ini, model matematik telah dibangunkan menggunakan teknik Newton-Euler. Pendekatan ini membolehkan analisis yang tepat terhadap gerakan translasi dan rotasi, dengan mengambil kira elemen seperti inersia, tork, dan daya luaran seperti seretan dan daya angkat. MPC kaskad ini seterusnya disahkan melalui eksperimen masa nyata, menyesuaikan dinamik UAV kepada pola pergolakan angin dan halangan yang tidak dapat diramalkan. Keupayaan penyesuaian dan ketepatan sistem ini dibandingkan dengan pengawal Proportional-Integral-Derivative (PID), menyerlahkan kelebihan ramalan MPC dalam persekitaran yang dinamik. Eksperimen telah dijalankan di hutan paya bakau, persekitaran yang dicirikan oleh pola angin yang kompleks, menggunakan UAV yang dilengkapi dengan pengawal perkakasan Pixhawk. UAV tersebut mengikuti laluan yang telah ditentukan untuk mengumpulkan data tentang altitud, olengan (roll), kecondongan

(pitch), yaw, dan tindak balas terhadap pergolakan angin, membolehkan strategi kawalan dan penambahbaikan model yang lebih baik berdasarkan prestasi dunia sebenar. Analisis perbandingan menunjukkan bahawa kuadkopter yang dilengkapi dengan MPC bukan sahaja mencapai kestabilan dan responsiviti yang lebih baik tetapi juga meningkatkan kecekapan operasi dalam keadaan cuaca buruk, membuktikan ketahanan dan kebolegunaan sistem ini dalam operasi UAV yang mencabar. Penyelidikan ini dengan demikian memajukan penggunaan UAV dalam aplikasi dunia sebenar yang menuntut dengan mengintegrasikan pemodelan dinamik yang canggih dengan teknik kawalan inovatif.

Kata kunci: *UAV, quadcopter, pengawal PID, pengawal Pintar, gangguan luar, MPC pengawal, pixhawk*

TABLE OF CONTENTS

	Page
DECLARATION	i
ACKNOWLEDGEMENT	ii
ABSTRACT	iv
<i>ABSTRAK</i>	vi
TABLE OF CONTENTS	viii
LIST OF TABLES	xiv
LIST OF FIGURES	xvi
LIST OF ABBREVIATIONS	xix
LIST OF SYMBOLS	xix
CHAPTER 1: INTRODUCTION	1
1.1 Background	1
1.2 Problem Statement	2
1.3 Research Questions	3
1.4 Research Hypothesis	4
1.5 Research Objectives	5
1.6 Research Scope	5
1.7 Orginazation of the Thesis	6

CHAPTER 2: LITERATURE REVIEW	7
2.1 Introduction	7
2.2 Application of UAV Quadcopter	7
2.3 Mathematical Modelling	10
2.4 Model Uncertainties	17
2.4.1 External Disturbances	17
2.4.1.1 Turbulent flow	18
2.4.1.2 Wind shear	19
2.4.2 Effect of wind turbulence on UAV	21
2.5 Control Methods	22
2.5.1 PID Controller	23
2.5.2 MPC Controller	26
2.6 Research Gap	30
CHAPTER 3: RESEARCH METHODOLOGY	32
3.1 Introduction	32
3.2 Research Flowchart	32
3.3 Mathematical Model Development	35
3.3.1 Wind Model Implementation in Simulink	35

3.3.2	Reference Frames Orientation of the UAV	36
3.3.3	Euler Angles	39
3.3.4	Quadcopter Mathematical Model	40
3.3.5	Forces and Moments	44
3.3.6	Actuator Dynamics	45
3.3.7	State Space Model	46
3.3.8	Linear Model	50
3.3.9	Quadcopter Holybro S500 Parameter Estimation	54
3.3.9.1	Mass	55
3.3.9.2	Moment Arm	55
3.3.9.3	Moment of Inertia	56
3.3.9.4	Moment of Inertia for x-axis	59
3.3.9.5	Moment of Inertia for y-axis	59
3.3.9.6	Moment of Inertia for z-axis	60
3.3.9.7	Thrust Coefficient	60
3.3.10	Simulink Plant Model Development	61
3.4	Controller Development	64
3.4.1	Explanation of Physical Re-Positioning Errors	64
3.4.1.1	Definition of Re-Positioning Errors in UAV	64
3.4.1.2	Impact of Re-Positioning Error on UAV Performance	65

3.4.2	PID Controller	67
3.4.3	Effect of Nonlinearities and Noise	69
3.4.4	PID Stability Controller	70
3.4.4.1	Roll Control	70
3.4.4.2	Pitch Control	71
3.4.4.3	Yaw Control	71
3.4.4.4	Altitude Control	71
3.4.4.5	Longitude Control	72
3.4.4.6	Latitude Control	72
3.4.5	PID CONTROLLER TUNING METHOD	73
3.4.5.1	Ziegler-Nichols First Method	73
3.4.5.2	Ziegler-Nichols Second Method	73
3.5	Model Predictive Controller	74
3.6	Implementing MPC Controller in Simulink	82
3.6.1	MPC Toolbox Interface Overview	82
3.6.2	Define Input and Output Channel Attributes	83
3.6.3	Configure Controller Horizons	84
3.6.4	Define Input Constraints	85

3.6.5	Specify Controller Tuning Weights	86
3.7	Experiment and Validation	87
3.7.1	Experiment Setup	87
3.7.2	PX4 Control Architecture	97
3.7.2.1	Position Controller	98
3.7.2.2	Attitude Controller	99
3.8	Implementation of Simulink Control Algorithm on Pixhawk	96
CHAPTER 4: RESULTS AND DISCUSSION		
	103	
4.1	Introduction	103
4.2	Simulation and Experiment Parameters	103
4.3	Simulation Result	105
4.3.1	Non-Windy Condition Results	106
4.3.2	Windy Condition Results	111
4.4	Discussion	116
4.4.1	Non windy Conditions	116
4.4.2	Windy Conditions	118
4.5	Experiment Results	120
4.5.1	PID Results	120
4.5.2	MPC Results	124

4.5.3	Discussion on Result	127
4.5.4	Summary	130
CHAPTER 5: CONCLUSION AND RECOMMENDATIONS		132
5.1	Conclusion	132
5.2	Recommendations	134
REFERENCE		136
APPENDICIES		
		150

LIST OF TABLES

			Page
Table 2.1		Literature review of the modelling techniques.	
	15		
Table 2.2		Comparison between the two modelling techniques for UAV.	
	16		
Table 2.3		Effect of wind turbulence on UAVs.	
	21		
Table 2.4		Summary of PID controller literature review	
	25		
Table 2.5		Summary of MPC controller literature review	
	29		
Table 3.1:		UAV Quadcopter Parameters	
	40		
Table 3.2:		Rotation About X-axis	
	57		
Table 3.3:		Rotation About Y-axis	
	57		
Table 3.4:		Rotation About Z-axis	
	58		
Table 3.5:		PID controller gains using ziegler-nichol methods	
	73		
Table 4.1:		UAV PID gains parameters	
	104		
Table 4.2:		UAV quadcopter MPC parameters	
	104		
Table 4.3:		Performance metric of the two controllers in non-windy conditions	
	110		

Table 4.4:	Error metric of the two controller in non-windy conditions	110
Table 4.5:	Performance metric of the two controllers in windy conditions	115
Table 4.6:	Error metric of the two controller in windy conditions	115
Table 4.7:	PID Altitude Error Results	122
Table 4.8:	MPC Altitude Error Results	126
Table 4.9:	Comparison of MPC Results and SMC with DOB	130

LIST OF FIGURES

	Page
Figure 2.1: MPC Controller Block	27
Figure 3.2: Research flowchart	34
Figure 3.2: Dryden wind turbulence simulink block	35
Figure 3.3: Coordinate system	36
Figure 3.4: Stationary and mobile reference frameworks	37
Figure 3.5: UAV propeller configuration	38
Figure 3.6: Interplay between propeller dynamics and the UAV's motion	39
Figure 3.7: Moment of Arm of Quadcopter	56
Figure 3.8: PID full block system	68
Figure 3.9: Closed loop control structure	70
Figure 3.10: The sustained oscillations for PID controller gain	74
Figure 3.11: MPC foundation principle	76
Figure 3.12: MPC designer interface	83
Figure 3.13: MPC input/output channel	84
Figure 3.14: MPC controller horizons configuration	85
Figure 3.15: MPC input/output definition	86
Figure 3.16: MPC output weights definition	87

Figure 3.17: PX4	full	architecture
90		
Figure 3.18: GPS		M8N
91		
Figure 3.19: Pixhawk	6C	Controller
92		
Figure 3.20: PM02	V3	Power
93		Module
Figure 3.21: RadioMaster	R81	Receiver
93		
Figure 3.22: RadioMaster		TX16S
94		
Figure 3.23: Telenetry Radio V3		95
Figure 3.24: Holybro	2216	KV920
96		Motor
Figure 3.25: ESCs	BLHeli	S
96		ESC
		20A
Figure 3.26: Battery S4 5200 mh		97
Figure 3.27: PX4	detailed	blocks
98		
Figure 4.1: Quadcopter	altitude	response
106		in non-windy conditions.
Figure 4.2: Quadcopter	phi	response
107		in non-windy conditions
Figure 4.3: Quadcopter	yaw	response
108		in non-windy conditions
Figure 4.4: Quadcopter	theta	response
109		in non-windy conditions
Figure 4.5: Quadcopter	altitude	response
111		in windy conditions
Figure 4.6: Quadcopter	phi	response
112		in windy conditions
Figure 4.7: Quadcopter yaw response in windy conditions		113
Figure 4.8: Quadcopter	theta	response
114		in windy conditions
Figure 4.9: PID		roll response
120		
Figure 4.10: PID		pitch response
121		

Figure	4.11:		PID	yaw	response	
	121					
Figure	4.12:		PID	altitude	response	
	122					
Figure	4.13:	PID	trajectory	tracking	response	
	123					
Figure	4.14:		MPC	roll	response	
	124					
Figure	4.15:		MPC	pitch	response	
	124					
Figure	4.16:		MPC	yaw	response	
	125					
Figure	4.17:		MPC	altitude	response	
	125					
Figure	4.18:	MPC	trajectory	tracking	response	
	127					
Figure	B1:	Kuching	Wetlands	National	Park	
	150					
Figure	B2:	Recorded	Wind	Speed	for	January
	151					
Figure	B3:	Recorded	Wind	Direction	for	January
	151					
Figure B4:	Vane Anemometer testo 416					154
Figure	B5:		Velocity	Vane	Data	
	155					
Figure	B6:		General	Technical	Data	
	155					
Figure	B7:			Pixhawk	6C	
	156					
Figure B8:	Accelerometer Calibration Starting Process					160
Figure B9:	Accelerometer Calibration Ending Process					161
Figure B10:	Gyroscope Calibration Starting Process					162
Figure B11:	Accelerometer Calibration Starting Process					163
Figure B12:	Compass Calibration Starting Process					165
Figure B13:	Compass Calibration Starting Ending Process					166

LIST OF ABBREVIATIONS

ABC	Aircraft Body Centre
ANN	Artificial Neural Network
APS	Asynchronous Path Smoothing
CS	Cuckoo Search
DE	Differential Evolution
DIC	Direct Inverse Control
ESC	Electronic Speed Control
FFNN	Feedforward Neural Network
FLC	Fuzzy Logic Control
FOPID	Fractional Order Proportional Integrator Derivative
GA	Genetic Algorithm
GNSS	Global Navigation Satellite System
IMU	Inertial Measurement Unit
LQR	Linear Quadratic Gaussian
LPV	Linear Parameters Varying
LTI	Linear Time Invariant
MIMO	Multi-Input and Multi-Output
MPC	Model Predictive Control
NED	North-East-Down
NMPC	Nonlinear Model Predictive Control
NRC	National Research council
PD	Proportional Derivative
PID	Proportional Integrator Derivative

PSO	Particle Swarm Optimization
ROS	Robot Operating system
RTL	Return to Launch
SISO	Sible-Input and Single-Output
SMC	Slide Mode Control
SPNN	Sigma-Pi Neural Network
UAV	Unmanned Arial Vehicle
VTOL	Vertical Take-Off and Landing
ZOH	Zero Order Hold

LIST OF SYMBOLS

A, B, D	Matrices defining the state-space dynamics
b	Thrust factor
C_d	Translational drag coefficient
C_t	Non-dimensional thrust coefficient
D_{diameter}	propeller diameter, measured in meters (m)
d	Distance between the strings used in the pendulum setup
$d_{\text{disturbance}}$	Disturbance input vector
$\hat{e}_3$	Unit vector in the body-fixed z-axis
$\hat{e}_z$	Unit vector in the inertial z-axis
f_B	Total force vector in the body-fixed frame
f_x, f_y, f_z	Force components along the x, y, z-axes of the body-fixed frame
f_t	Total vertical thrust generated by the four rotors
f_w	Wind force vector in the body-fixed frame
f_k	Thrust factor
$g_i(x)$	Control input vector functions corresponding to the input u_i
g_a	Gyroscopic moments
I	Moment of inertia matrix of the quadrotor
I_x, I_y, I_z	Moments of inertia about the x, y, z axes of the body-fixed frame
J_p	Moment of inertia of a single rotor about its axis of rotation
k_{drag}	Aerodynamic drag coefficient
L_{pendulum}	Length of the string used in the pendulum setup
l	Distance between any rotor and the <i>centre</i> of the drone (arm length)

L_u, L_v, L_w	Characteristic wavelength of turbulent flow in three directions
m	Mass of the quadrotor
m_B	Total torque vector acting on the quadrotor in the body-fixed frame
m_k	Drag factor
m_x, m_y, m_z	Torque components about the x, y, z axes of the body-fixed frame
n	Propeller speed in revolutions per second (rps)
p, q, r	angular velocities in the x, y, z axis of the body-fixed frame
R	Rotation matrix
T_{matrix}	Transformation matrix
Tr	Indicates the vector is transposed to be a column vector
T_{osci}	Period of one oscillation, measured in seconds (s)
u_x, v_y, w_z	linear Velocities along the x, y, z axis in body fixed frame
u_i	Control inputs
$\bar{U}$	Control input vector at equilibrium
U	Control inputs
V	Speed of the wind shear
V_{max}	Strength of the wind shear
v	Linear velocity vector in the inertial frame
v_B	Linear velocity vector in the body-fixed frame
$\dot{v}_B$	Time derivative of the linear velocity vector in the body-fixed frame
$\bar{X}$	State vector at equilibrium
x_{state}	State vector of the quadrotor
x, y, z	Linear positions of the quadrotor in the inertial frame
Φ_u, Φ_v, Φ_w	Spectral function of turbulent flow
$\sigma_u, \sigma_v, \sigma_w$	Constant wind speed in three directions
ϕ, θ, ψ	Angular positions of the quadrotor in the inertial frame

ω	Angular velocity in the inertial reference frame
ω_B	Angular velocity vector in the body-fixed reference frame
τ_B	Control torque vector
τ_w	Wind-induced torque vector
Ω_i	Angular velocity of the i-th rotor
τ_x, τ_y, τ_z	Control torque about the body-fixed x, y, z-axis
$\Omega_1, \Omega_2, \Omega_3, \Omega_4$	Angular velocities (rotational speeds) of the four rotors
ρ	Air density

CHAPTER 1

INTRODUCTION

1.1 Background

Critical windy environments present unique challenges for the operation and control of Unmanned Aerial Vehicles (UAVs), particularly quadcopters. These environments are characterized by strong, persistent winds, high temperatures, and challenging atmospheric conditions. Understanding the dynamics of UAV performance in such conditions is essential for developing effective control systems and ensuring the reliability of these aerial platforms (Shi et al., 2016).

UAV quadcopters are modelled using momentum theory as well as blade element theory (Naidoo et al., 2011). The Euler-Lagrange method has been employed to derive the defining equations of motion for six-degree-of-freedom systems (Patel and Barve, 2014). An alternative method for modelling the dynamics of quadrotors is the Newton-Euler technique, widely used by researchers and engineers (Fernando., 2013).

Effective control systems must compensate for instability caused by wind gusts and turbulence to maintain flight stability and accuracy. Several control strategies have been explored, including PID-PSO (Mohammed et al., 2014), linear PID controller (Salma and Osman, 2020), LQG (Alifah, Z. N, 2022), SMC (Suraj et al., 2021), Neural Network controller (Jiang et al., 2020), and Model Predictive Controller (Prach and Kayacan, 2018).

In recent years, UAV quadcopters have become increasingly popular for a wide range of applications, including environmental navigation, search and rescue operations, and commercial deliveries. However, their performance can be significantly impacted by

adverse weather conditions, especially in critical windy environments (Mahony et al., 2012).

However, the increasing use of UAVs in critical windy environments has highlighted the need for continuous improvements in control systems. The development of more sophisticated algorithms and the integration of sensor data are essential to adapt to rapidly changing environmental conditions and ensure safe and efficient UAV operations (Faessler et al., 2016).

Therefore, it is imperative to consistently research and develop advanced control systems for UAV quadcopters to ensure their optimal performance in critical windy environments, thereby enhancing their utility and reliability in various applications.

1.2 Problem Statement

Operating UAVs, particularly quadcopters, in critical windy environments characterized by strong turbulence and gusts poses significant challenges to their stability, robustness, and overall operational efficiency. These conditions disrupt the UAV's navigation capabilities, leading to difficulties in maintaining control and executing tasks effectively, especially in scenarios requiring precision, such as environmental and ecological monitoring. While control systems like PID, PD, and Sliding Mode Control (SMC) have shown promising results in general contexts, their effectiveness in addressing the destabilizing effects of complex wind patterns, including high gusts and unpredictable turbulence, remains inadequate. Furthermore, many existing studies rely heavily on simulations, which fail to capture the full scope of real-world wind dynamics, limiting their practical applicability.

This research addresses the problem of developing advanced UAV control strategies specifically designed to enhance stability and robustness in windy environments. By evaluating the performance and limitations of existing control technologies, this study aims to design and implement innovative solutions, such as adaptive and robust control algorithms, that can mitigate the challenges posed by wind disturbances. Additionally, the research seeks to explore decentralized control strategies for coordinated multi-UAV missions, ensuring enhanced coverage and improved data accuracy in critical conditions. Through a combination of simulation and field testing, this study aspires to bridge the gap between theoretical models and practical applications, contributing to the development of UAV systems capable of reliable operation in windy environments and other challenging settings.

1.3 Research Questions

To comprehensively understand and enhance the performance of UAV quadcopters in challenging windy environments, this study addresses several key research questions. These questions focus on identifying critical factors affecting UAV stability and accuracy, comparing advanced control strategies, and evaluating the impact of environmental modelling on UAV performance. The following research questions guide this inquiry:

- Question 1: What are the key factors affecting the stability and accuracy of UAV quadcopters in windy critical environments, and how can these factors Incorporated in the mathematical model of the UAV?
- Question 2: How does the cascade Model Predictive Controller perform compared to that of the traditional PID controller in terms of stability,

accuracy, and responsiveness under varying environmental conditions in windy critical environments?

- Question 3: What is the impact of incorporating the Dryden wind turbulence model on the control strategy and performance of UAV quadcopters in windy critical environment?
- Question 4: How do environmental uncertainties in windy critical environment influence the performance metrics (e.g., rise time, trajectory tracking accuracy) of UAV quadcopters equipped with PID and MPC controllers?

1.4 Research Hypothesis

This research investigates the effectiveness of advanced control strategies for UAV quadcopters operating in the challenging conditions of mangrove environments. By comparing the performance of the cascade Model Predictive Controller with the traditional Proportional-Integral-Derivative controller, this study aims to determine the superior method for maintaining stability and accuracy. The following hypotheses have been formulated to guide this investigation.

- Hypothesis 1: The Model Predictive Controller will exhibit superior performance in maintaining the stability and accuracy of UAV quadcopters in critical windy environments compared to the traditional PID controller.

- Hypothesis 2: The integration of the Dryden wind turbulence model in the controller design will significantly improve the UAV's ability to navigate and perform tasks in a critical windy environment.
- Hypothesis 3: UAV quadcopters equipped with the cascade MPC controller will show a reduction in rise time and improvement in trajectory tracking accuracy compared to those equipped with a PID controller in both non-windy and windy conditions.

1.5 Objectives

This research embarks on the following objectives:

- To model quadcopter dynamics incorporating mass, inertia, aerodynamics, thrust, drag, and disturbances.
- To develop a cascade MPC control system with outer position and inner attitude control loops, optimizing inputs within constraints to ensure accuracy, robustness, and rejection of wind turbulence disturbances.
- To experimentally validate the performance of the MPC and PID controllers in terms of stability, precision, and robustness metrics on a quadcopter equipped with a Pixhawk controller

1.6 Research Scope

The scope of this research is as follows:

- The mathematical model of the quadcopter will be done using the Newton-Euler approach.

- The development of controllers will be based on the cascade Model Predictive Controller.
- The Dryden wind turbulence model will be used to imitate the wind turbulence environment.
- Real-time experiment will verify both PID controller and cascade MPC controller related to environmental uncertainties in critical windy environment.
- The work will include and emphasize on the robustness of altitude and attitude parameters of UAV.

1.7 Organization of the Thesis

The organization of this thesis is as follows:

Chapter 1 introduces background, problem statement, objectives, research scope, and organization of the thesis.

Chapter 2 starts with an overview of the application of UAVs. It then previews current and previous research related to the modelling techniques and control systems of UAV quadcopters. Lastly, it highlights the research gap that justifies the need for this study.

Chapter 3 details the methodology, covering the research flowchart, wind characteristics in critical windy environments, quadcopter parameter estimation, mathematical model and control system development, and the experimental setup using Pixhawk hardware.

Chapter 4 examine the presentation of experimental results, comparing the performance of PID and MPC controllers in Pixhawk, including analysis of dynamic response and performance metrics.

Chapter 5 concludes and summarizes the study's contributions, discusses practical and theoretical implications, and suggests directions for future research.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The literature review begins by highlighting the diverse applications of UAVs across various sectors. It then explores the modelling processes of UAVs, with a particular focus on quadcopters, examining the techniques and theories that underpin their operation. Following this, the review critically examines different UAV control techniques, evaluating their strengths and limitations. This comprehensive discussion provides a foundation for understanding the complexities and challenges of deploying UAVs in demanding environments, such as mangrove forests.

2.2 Application of UAV Quadcopter

Unmanned Aerial Vehicles (UAVs), particularly quadcopters, are revolutionizing various industries due to their advanced capabilities and versatility. In agriculture, UAVs enhance crop management through precision agriculture techniques, enabling detailed monitoring and treatment of crops. This precise resource application boosts productivity and reduces environmental impact, promoting sustainable farming practices. Additionally, quadcopters support reforestation and land rehabilitation by dispersing seeds over deforested areas, aiding in ecological restoration.

In emergency response and disaster management, quadcopters provide real-time insights by surveying disaster-stricken areas, facilitating strategic and informed responses. Equipped with thermal imaging, these UAVs enhance search and rescue operations by identifying heat signatures of individuals in peril, improving emergency response

efficiency. Alawad et al. (2023) propose an innovative UAV-based system for disaster and crisis management in smart cities, employing a swarm optimization algorithm (SOA) to optimize UAV paths, reduce energy consumption, and ensure effective communication services during emergencies. This system operates within a delay-tolerant network framework, thoroughly exploring disaster zones and efficiently locating victims.

For infrastructure maintenance, quadcopters are invaluable in inspecting hard-to-reach structures and systems, detecting defects or hazards such as gas leaks or structural weaknesses. Gaspari et al. (2022) compare two UAV methodologies for bridge inspection: UAV-LiDAR and UAV-photogrammetry. While traditional methods integrate topographic measurements, GNSS, and photogrammetry for highly accurate but time-consuming models, UAV-LiDAR uses a DJI Matrice 300 with a Zenmuse L1 sensor for rapid 3D data acquisition, offering more efficiency despite lower accuracy. Integrating photogrammetric models with ground control points improves UAV-LiDAR's accuracy, highlighting its potential for flexible, cost-effective bridge monitoring.

In the entertainment industry, quadcopters have transformed photography and filming by capturing dynamic aerial footage, enriching the visual narrative of films and media productions. They have also innovated live entertainment with drone-based light shows, offering environmentally friendly alternatives to fireworks. Zhao and Zhu (2023) present MS-YOLOv7, an advanced object detection model for UAV aerial photography, enhancing YOLOv7 with multiple detection heads and convolutional attention modules to improve small, densely packed object detection. Despite its computational intensity, this model significantly improves performance in detecting overlapping and occluded objects.

The logistics and delivery sectors benefit from UAV technology, especially in last-mile deliveries. Quadcopters expedite the delivery of essential goods, including medical supplies in remote regions, playing a crucial role in critical healthcare operations. In urban settings, they help reduce congestion and emissions. Lammers et al. (2023) examine UAVs for delivering medical supplies in combat zones, noting their potential for timely delivery despite challenges like carrying capacity and airspace regulations. UAVs offer timely medical logistics with minimal blood product degradation when thermoregulated.

Environmental monitoring has advanced with quadcopters, facilitating detailed data collection on ecosystems, pollution, and wildlife without causing disturbances. This capability is essential for ongoing environmental assessments and conservation efforts. Pan et al. (2022) presents a UAV-aided emergency environmental monitoring system utilizing a LoRa mesh network for infrastructure-less areas, rapidly collecting environmental data from emergency sites with robust network performance under dynamic conditions.

Educational institutions incorporate quadcopters into research and teaching, offering practical experience in technology applications. Jiang et al. (2020) propose a quadcopter UAV control system enhanced by a Sigma-Pi Neural Network (SPNN) compensator, managing model uncertainties and improving performance. This framework integrates SPNN with dynamic inversion control, enhancing UAV control accuracy under varying conditions.

In healthcare, UAVs are tested for rapid organ transport, potentially revolutionizing transplant procedures. In sports, quadcopters provide unique perspectives for training analysis and audience engagement, especially in extreme sports. Gupta et al.

(2023) propose VAHAK, a blockchain-based UAV delivery system for Healthcare 4.0 services, addressing security, privacy, and latency issues in remote medical deliveries, leveraging 5G-enabled Tactile Internet for real-time monitoring.

The integration of quadcopters across these diverse fields underscores their utility in addressing complex challenges and driving technological innovation and efficiency in commercial and public sectors. As UAV technology evolves, its impact is expected to expand further, offering new solutions and enhancing capabilities across numerous applications.

2.3 Mathematical Modelling

In three-dimensional space, a quadrotor unmanned aerial vehicle is considered a rigid body. It has six degrees of freedom, three of which are translational and three of which are rotational. It is important to note that the first three translational degrees define the UAV location, which is a basic operation, but the last three translational degrees describe the UAV orientation, which is a more challenging process with ramifications for the resulting model. There are some ways for determining the orientation of a usually stiff body in space that may be used.

The modelling approaches of quadcopters are based on the physics of the system with further partitioning of the system into smaller subsystems for easy analysis, design, and modelling. Generally, in quadcopter modelling, different assumptions have been made by simplification the mathematical equations needed for the UAV modelling while still establishing a fairly accurate model as possible. Universal model assumptions from different kinds of literature and research publications are herein presented as under:

- In addition to having asymmetrical construction, the quadcopter also has a diagonal inertia matrix based on this symmetry.
- The quadcopter's basic structure consists of a solid frame with four motors attached to each side.
- It can be calculated that the bearing pressure and trail of each engine are proportional to the square speed of each motor, which is an estimate that comes extremely close to representing the quadcopter's aerodynamic behaviour.
- The quadcopter's Centre of Gravity is fixed concerning its starting point.
- In addition, because the quadrotor is time-invariant, there are no mass changes while in motion.
- The constants of thrust and drag are proportional to the square of the motor's speed, respectively.
- Because all the motors on the quadcopter are similar, a single motor may be studied without affecting the generality of the results.

To derive the equations of motion of the quadrotor, it is common to assume that the vehicle is a rigid body. The Euler angles method is used to describe the motion of a rigid body in 3-D space governed by the Newton-Euler equations. Given the full dynamics of the quadrotor, when it hovers at a certain height in the presence of small perturbations, the angular rotations of the quadrotor are independent of its linear translations, while the translational motions of the quadrotor depend on its orientations (Bouabdallah., 2005). As a result, the dynamic system of the quadrotor may be separated into two subsystems: the positioning subsystem and the attitude subsystem. The positioning subsystem is responsible for determining the position of the quadrotor.

Diverse pieces of literature and research papers have been published in which various mathematical techniques have been used for modelling the UAVs from first principal approximation, ranging from the quaternion formulation demonstrated in through the "Superposition-formulation" used into the Newton-Euler technique; amongst these, the two most widely used techniques in deriving the quadcopter's model equations are:

- Newton-Euler—This approach is based on Newton's second law equations on rigid bodies and is used to solve rigid body problems.
- Euler-Lagrange: This approach is based on energy and kinematics, with equations derived from Newton's second law and the use of a speed transformation matrix to calculate the speed of a particle.

The literature review on dynamic modelling of quadcopters showcases a variety of methodologies, each demonstrating distinct approaches and their relative applicability under various conditions. This analysis illuminates their comparative benefits and limitations and offers insights into potential enhancements.

Wang et al. (2016) use the Newton-Euler method with assumptions like rigidity and symmetry, excluding aerodynamic effects, which limit the model's applicability. Integrating motor dynamics enhances accuracy, but static approximations restrict its utility in varied conditions. Future models should include aerodynamic effects and evaluate linear assumptions against non-linear scenarios to improve real-world applicability.

In a similar vein, Martini et al. (2022) employ the Euler-Lagrange formulation, meticulously including gyroscopic effects and aerodynamic drag, thereby substantially enhancing precision, and aiding accurate control designs. However, the model's complexity might hinder its utility in real-time applications due to computational demands. Therefore,

further research could aim to balance model fidelity with computational efficiency and augment the model's practical applicability with more extensive experimental data.

Furthermore, Patel and Barve (2014) advance quadcopter UAV modelling with the Euler-Lagrange method by incorporating variable air-density to refine altitude modelling. Although validated in MATLAB-Simulink, this adjustment introduces complexity and necessitates rigorous validation beyond simulations to ensure its real-world applicability. This suggests a pressing need for expanded experimental validations under varying operational conditions.

Similarly, Rinaldi et al. (2023) develop a comprehensive nonlinear dynamic model via the Newton-Euler method. While providing an extensive comparative analysis of controllers, simplifications such as neglecting rotor dynamics may detract from accuracy. Thus, enhancing the model's robustness and adaptability through additional experimental validation would solidify its effectiveness.

Additionally, Djizi et al. (2023) integrate the Euler-Lagrange method with PID controllers to assess quadcopter stability through MATLAB simulations. Although detailed, the simplification in PID coefficient adjustments might introduce inaccuracies in dynamic environments. A deeper discussion on the limitations of PID control and expanded experimental testing would substantiate the model's real-world effectiveness.

Following this, Shao et al. (2022) present a detailed mathematical model considering nonlinear dynamics with external disturbances and input saturation, crucial for understanding UAV behaviour under various constraints. Given the model's complexity, it could impact computational efficiency, particularly in real-time applications. Hence,

further validation through experimental setups would be beneficial to comprehensively evaluate the model's performance in operational environments.

Subsequently, Walid et al. (2018) offer a comprehensive examination of dynamics modelling for rotorcraft UAVs, utilizing both Euler-Lagrange and Newton-Euler methods. This review could benefit from clearer discussions on the limitations of these modelling techniques, particularly their assumptions and impacts on accuracy and real-world applicability. Enhancements such as incorporating validation strategies through simulation or experimental setups would strengthen confidence in these models.

Similarly, Nguyen et al. (2019) explore dynamic modelling using the Euler-Lagrange formalism, acknowledging simplifications like small angle approximations. The inclusion of experimental validation aligns theoretical models with actual system behaviour, and future work could further enhance model accuracy by more robustly integrating nonlinear effects and external disturbances. Table 2.1 summarizes the modelling technique for UAV.

Table 2.1: List of literature review of UAV modelling techniques

Authors	Year	Modelling Technique	Key Modelling Features	Limitations	Improvements Suggested
Wang, P. et al.	2016	Newton-Euler	Includes motor dynamics, linearization around an equilibrium point	Simplified assumptions limit applicability under varied conditions	Incorporate aerodynamic effects, validate against non-linear scenarios
Martini, S. et al.	2022	Euler-Lagrange	Incorporates gyroscopic effects and aerodynamic drag	Computational demands high, lacks real-world testing	Balance model fidelity with computational efficiency, extensive real-world validation

Patel, K., Barve, J.	2014	Euler-Lagrange	Modification for variable air-density to enhance altitude modelling	Complex parameters need validation beyond simulations	Expanded experimental validations, comparison under varying conditions
Rinaldi, M. et al.	2023	Newton-Euler	Considers both attitude stabilization and trajectory tracking	Neglects rotor dynamics and complex aerodynamic interactions	Include rotor dynamics, validate with experimental data
Djizi, H. et al.	2023	Euler-Lagrange	Detailed modelling of non-linear dynamics and external forces	Simplifications in PID adjustments, lacks real-world testing	Discuss limitations of PID in complex environments, conduct practical tests
Shao, X. et al.	2022	Newton-Euler	Considers non-linear dynamics with external disturbances and input saturation	Model complexity limits real-time applications	Address trade-offs between accuracy and computational efficiency
Walid, M. et al.	2018	Euler-Lagrange	Comprehensive rotational and translational dynamics	Simplifications limit accuracy in complex scenarios	Integrate non-linear effects and external disturbances
Nguyen, A.T. et al.	2019	Euler-Lagrange	Addresses non-linearities and system parameter uncertainties	Computational intensity may limit practical application	Validation in real-world scenarios, optimize computational demands

In this work, the modelling technique that will be used is the Newton-Euler method over the Euler-Lagrange method for UAV dynamics due to its simplicity and direct application of Newton's laws, which are intuitive and computationally efficient—essential for real-time UAV control. This method's straightforward approach to handling dynamic changes and external forces is particularly suited for the responsive adjustments required in UAV flight, which must rapidly adapt to environmental factors. Additionally, its proven effectiveness in aerospace applications demonstrates reliability and practicality, making it

ideal for both research and industry applications. Moreover, its ease of understanding and implementation makes it accessible for broader educational and engineering purposes, facilitating communication and application in practical scenarios. Table 2.2 summarize the comparison between the two modelling techniques for UAV.

Table 2.2: The comparison between the two modelling techniques for UAV

Aspect	Euler-Lagrange Method	Newton-Euler Method
Principle	Based on Lagrangian mechanics, which uses the difference between kinetic energy and potential energy.	Based on Newtonian mechanics, applying Newton's second law (force = mass $\times$ acceleration).
Application in UAVs	Derives equations of motion from the Lagrangian, considering energy transformations within the system.	Focuses on forces acting on each component, applying force and torque balances directly.
Benefits	Comprehensive approach to system dynamics Adaptable to various configurations and constraints	Straightforward and intuitive Suitable for real-time applications in control systems
Challenges	Can become mathematically complex May require significant computational resources	May not naturally include all potential and non-conservative forces Simpler, less detailed
Usage	Preferred for theoretical studies and high-fidelity simulations.	Commonly used in practical applications where fast computation is needed, like flight control.

2.4 Model Uncertainties

In actuality, the motion of the quadrotors cannot always be characterized by their exact dynamics because of the uncertainties that occur, which are mostly caused by model errors, external disturbances, and measurement inaccuracies, among other things. Each of these uncertainties is discussed in further detail below.

For the most part, model uncertainties are caused by the following factors:

- The linearized quadrotor dynamics are extensively utilized in controller design because they make it easier to create a simple controller.
- Some system characteristics, such as the inertia of the quadrotor and the weight of the quadrotor, are wrong.

The motor dynamics are that of a first or second-order system, which may or may not accurately reflect the thrust created by the propellers in some cases. Bouffard et al. (2012) propose that the problem of model inaccuracy may be overcome by detecting the system dynamics through experiments (Iskandarani et al., 2013; Alexis et al., 2011b), or by treating it as an external disturbance (Alexis et al., 2011a; Bouffard et al., 2012).

2.4.1 External Disturbances

External disturbances, such as strong winds, actuator failure, and extra cargo, frequently impair the quadrotor's performance during flight. Alexis et al., (2012b) demonstrate how a quadrotor may function in high wind conditions. A flexible cable is used to link an unknown payload to the quadrotor, which allows for greater flexibility. In this chapter, the loss of motor efficacy is explored and dealt with (Sharifi et al., 2010).

In mangrove forest which is one of the critical windy environments, thriving at the interface of land and sea in tropical and subtropical regions around the globe, present a unique environmental setting characterized by saline coastal sediment habitats. These ecosystems are not only pivotal for coastal protection, carbon sequestration, and supporting biodiversity, but they also exhibit a distinctive interaction with atmospheric elements, particularly wind. The wind characteristics in mangrove areas are a crucial aspect of these ecosystems, influencing various physical and biological processes. These root structures, along with the relatively low height but dense canopy, create unique conditions for wind

flow. In Sarawak, the biodiversity of mangroves adds another layer of complexity, as the variety in species and forest structure can influence wind patterns significantly with its tropical rainforest climate and high humidity and significant rainfall. Its coastal regions, where mangroves are predominantly found, are subject to both monsoonal and daily sea breezes. The interaction between these macro-scale wind patterns and the local micro-scale turbulence created by the mangrove forests creates a complex wind environment.

2.4.1.1 Turbulent flow

Turbulent flow manifests as a series of continuous and random fluctuations, invariably occurring alongside the phenomenon of constant wind. The genesis of turbulent flow is multifactorial, influenced by variables such as wind shear, thermal exchange, geographical features, and the wake turbulence generated by other aircraft. In the realm of engineering, the characterization of atmospheric turbulence frequently leverages the principles of stochastic process theory and methodologies. Within this context, turbulence is modelled using frameworks such as the Dryden and Von Karman models, which are predicated on extensive empirical data and statistical analysis. A notable distinction between these models lies in their methodological approaches: the Dryden model initially formulates a correlation function for turbulence, subsequently deriving the spectral function, whereas the Von Karman model adopts a reverse sequence, establishing the spectral function before deducing the turbulence's correlation function. Subsequent research indicates that the primary difference between the two models is confined to the slope of the spectral function's high-frequency band, suggesting that both models are equally viable for addressing engineering challenges (Wang et al., 2019). The formulation of the Dryden spectral function is as follows:

$$\Phi_u(\Omega) = \sigma_u^2 \frac{L_u}{\pi} \cdot \frac{1}{1+(L_u\Omega)^2} \quad \text{Equation 2.1}$$

$$\Phi_v(\Omega) = \sigma_v^2 \frac{L_v}{\pi} \cdot \frac{1+12(L_v\Omega)^2}{(1+4(L_v\Omega)^2)^2} \quad \text{Equation 2.2}$$

$$\Phi_w(\Omega) = \sigma_w^2 \frac{L_w}{\pi} \cdot \frac{1+12(L_w\Omega)^2}{(1+4(L_w\Omega)^2)^2} \quad \text{Equation 2.3}$$

Where Φ_u , Φ_v , and Φ_w are the spectral function of turbulent flow whose directions are along the axis of UAV body coordinate system. σ_u , σ_v , and σ_w are the constant wind speed in three directions. L_u , L_v , and L_w are the characteristic wavelength of turbulent flow in three directions.

2.4.1.2 Wind Shear

Wind shear represents variations in wind flow over time or space. The National Research Council (NRC) of the United States quantifies wind shear magnitude as the difference between two wind vectors at different points, divided by the distance between them. Wind shear can originate from multiple sources, especially affecting Unmanned Aerial Vehicle (UAV) operations at lower altitudes. Common types include frontal wind shear, the nocturnal jet stream, topography-induced wind shear, and microbursts.

The velocity of wind shear depends on spatial coordinates, making it relatively straightforward to model. Unlike turbulent flow, wind shear occurs less frequently, with changes happening every few seconds at most. In simulation testing, a sudden shift in steady wind velocity indicates wind shear, which can be discrete or deterministic and typically lasts briefly. Wind shear models, crucial for simulations, are categorized by profile shapes, including rectangular, trapezoidal, and "1-cosine" configurations.

Rectangular model:

$$\frac{y}{d_m} V_{max} \quad (0 \leq y \leq d_m) \quad \text{Equation 2.4}$$

$$\frac{2d_m - y}{d_m} V_{max} \quad (d_m \leq y \leq 2d_m) \quad \text{Equation 2.5}$$

Trapezoidal model:

$$\frac{y}{h} V_{max} \quad (0 \leq y \leq h) \quad \text{Equation 2.6}$$

$$V_{max} \quad (h \leq y \leq 2d_m - h) \quad \text{Equation 2.7}$$

$$\frac{2d_m - y}{d_m} V_{max} \quad (d_m - h \leq y \leq 2d_m) \quad \text{Equation 2.8}$$

Full wavelength “1-consine” model

$$V = \begin{cases} 0 & (y < 0) \\ \frac{V_{max}}{2} \left(1 - \cos \frac{\pi y}{d_m} \right) & (0 \leq y \leq 2d_m) \\ 0 & (y > 2d_m) \end{cases} \quad \text{Equation 2.9}$$

Half wavelength “1-consine” model

$$V = \begin{cases} 0 & (y < 0) \\ \frac{V_{max}}{2} \left(1 - \cos \frac{\pi y}{d_m} \right) & (0 \leq y \leq d_m) \\ 0 & (y > d_m) \end{cases} \quad \text{Equation 2.10}$$

where V represents the speed of the wind shear with arbitrary direction d_m represents the length of the range where exist wind shear, while V_{max} is the strength of the wind shear. According to the previous research, the wind shear scale d_m and the wind shear strength V_{max} are in connection with the characteristic wavelength L and the constant wind speed σ mentioned in “Turbulent flow” section.

2.4.2 Effect of Wind Turbulence on UAV

Below is a table that classifies the effects of wind turbulence and wind gusts in mangrove forests on UAV quadcopters. It highlights the different aspects of UAV operation that are impacted by these environmental factors. Understanding how each factor

influences UAV performance can help in designing more resilient and efficient UAV systems for operations in challenging environments like mangrove forests. The Table 2.3 summarize the effect of Wind on UAV.

Table 2.3: Effect of Wind on UAV

Aspect	Effect of Wind Turbulence	Effect of Wind Gusts	Notes on Impact on UAV Quadcopter
Stability	Increased pitch and roll variability, requiring constant adjustments.	Sudden shifts in altitude and orientation, demanding rapid response from control systems.	Wind gusts tend to have a more immediate and pronounced effect on UAV stability, potentially leading to loss of control.
Navigation and Control	Precision navigation is more challenging; increased errors in position and velocity estimates.	Can cause off-course deviations and abrupt changes in trajectory.	Both factors complicate navigation, but gusts pose a higher risk of causing significant deviations from the intended path.
Flight Efficiency	Flight paths may be less direct due to the need to compensate for drifting caused by turbulence.	Possible enforced detours or altitude adjustments to maintain flight safety.	Wind gusts can drastically alter the efficiency of a flight plan by forcing more significant deviations.
Operational Limits	May reduce the operational envelope of the UAV due to increased risks during take-off, landing, and low-altitude operations.	Sudden gusts can exceed the UAV's operational limits, leading to emergency procedures or aborted missions.	Gusts are more likely to push a UAV beyond its operational limits suddenly and unpredictably.

Wind turbulence and gusts significantly impact UAV quadcopters' stability, control, and navigation. Turbulence increases pitch and roll variability, requiring constant adjustments, while gusts cause sudden shifts in altitude and orientation, demanding rapid control responses. Both factors complicate precision navigation by causing errors and off-course deviations, with gusts posing a higher risk of significant trajectory changes. Energy

consumption rises due to continuous control adjustments for turbulence and sporadic spikes for gusts. Additionally, wind effects challenge UAV operational limits and sensor accuracy, necessitating robust control algorithms to maintain stable, controlled, and precise navigation. Managing these wind effects is crucial for the effectiveness of UAV operations.

2.5 Control Methods

As previously indicated, the quadrotor is constantly subjected to a variety of uncertainty. Quadrotor dynamics are regarded as correct in most ideal scenarios because disturbances are avoided. As a result, in such instances, the control mechanisms do not need to be capable of dealing with uncertainties. Because of external disturbances, the resilience of the control systems used for the quadrotor is essential in complicated working situations where the quadrotor must operate. It will be discussed in this section the most recent research on quadrotor control methods that have been done recently.

2.5.1 PID Controller

This section reviews the existing literature on control strategies for Unmanned Aerial Vehicles (UAVs), focusing on enhancements to PID controllers and advanced techniques such as Linear Quadratic Regulator (LQR), Linear Quadratic Gaussian (LQG), Sliding Mode (SMS), and H_∞ controllers. These strategies aim to optimize UAV flight stability, adaptability, and precision under dynamic conditions.

PID controllers remain fundamental to UAV control due to their simplicity and efficiency. Several studies have enhanced PID performance using advanced techniques. Espinoza et al. (2022) demonstrated that Fractional-Order PID (FOPID) controllers outperform traditional PID controllers in stabilization and disturbance rejection through simulations, though the complexity of tuning and lack of real-world validation limit their

practical use. Similarly, Yuan et al. (2022) optimized PID parameters using genetic algorithms and simulated annealing, achieving improved stability and trajectory tracking in strong winds. However, the reliance on computationally intensive simulations poses challenges for real-time applications. Further advancements in hybrid PID systems include the integration of neural networks and fuzzy logic. Chenyu Liu and Jiawang Xu (2023) combined RBF neural networks with PID control for agricultural UAVs, achieving improved adaptability and robustness in simulations, though model complexity limits practical use. Gui (2022) evaluated fuzzy PID controllers for quadrotors under wind disturbances, demonstrating superior performance in simulations compared to traditional PID. Abdelmaksoud et al. (2021) proposed an SR-FPID system utilizing fuzzy logic for automatic PID tuning, achieving significant improvements in settling time, peak time, and steady-state error. However, the computational demands and exclusive reliance on simulations necessitate further investigation for real-world implementation.

Linear Quadratic Regulator (LQR) and Linear Quadratic Gaussian (LQG) controllers have gained attention for their ability to balance stability and control effort in UAV systems. Velagić et al. (2022) implemented an LQR controller for quadrotor trajectory tracking, showing superior energy efficiency and robustness compared to PID controllers, particularly in scenarios involving external disturbances. However, the computational overhead associated with solving the Riccati equation remains a challenge for onboard UAV implementation. Similarly, Fessi et al. (2019) explored LQG controllers, integrating stochastic noise handling with linear-quadratic optimization. Their study highlighted improved performance in noisy environments, though the reliance on accurate state estimation complicates practical applications.

Sliding Mode Controllers (SMS) have emerged as a robust alternative for handling non-linearities and uncertainties in UAV dynamics. MC offers robust performance in handling UAV nonlinearities and disturbances. Wang and Mao (2024) proposed a non-overshooting sliding mode control approach for UAVs, achieving global exponential stability without overshoot, even in the presence of unknown stochastic disturbances. This method enhances precision in trajectory tracking. Additionally, Peixoto et al. (2023) introduced a dynamic smooth sliding control applied to UAV trajectory tracking, aiming to alleviate control chattering and improve practical implementation. Their approach was validated through realistic simulations and experiments with a commercial hexacopter

H_∞ controllers are recognized for their robustness in managing uncertainties and disturbances, making them ideal for UAV systems operating under dynamic conditions. Zhang et al. (2024) addressed the dynamic coupling between UAVs and aerial manipulators, proposing a robust H_∞ controller to compensate for variable inertia parameters. This approach stabilized UAV flight during manipulator operation, with experimental results validating its effectiveness. Similarly, Wei and Han (2024) explored an optimal H_∞ control strategy in a novel active suspension system. While not UAV-specific, their findings on managing dynamic interactions and disturbances provide valuable insights applicable to UAV systems.

Hybrid approaches that combine multiple control strategies have also been explored. Fahmizal et al. (2023) addressed trajectory tracking control of a UAV bicopter using LQG, effectively managing inertial disturbances. This study demonstrates the potential of integrating LQG with other control methods to enhance UAV performance.

Similarly, Srivastava et al. (2024) presented the design and flight testing of an LQR attitude control for quadcopter UAVs, showing enhanced stability. Their research underscores the benefits of combining LQR with integral action to improve UAV control systems. Table 2.5 summarizes the literature review of PID controller implementation in UAVs.

Table 2.5: Summary of PID controller literature review

Authors	Controller	Key Findings	Wind Disturbance	Limitations
Jacob Espinoza et al. (2023)	FO-PID	FOPID controllers enhance stabilization and minimize disruptions during flight.	No	Simulation-based study; real-world applicability of FOPID may be challenging due to complex tuning needs.
Chenyu Liu and Jiawang Xu (2023)	RBF-PID	RBF-PID shows improved response and robustness in dynamic environmental conditions.	No	Heavy reliance on simulated data; lacks real-world testing.
Yihan Gui (2022)	Fuzzy PID	Fuzzy PID controller shows superior stability and response time over classical and cascade PID in wind disturbances.	Yes	Solely based on simulation; lacks real-world testing and exploration of computational demands.
Fessi et al. (2019)	LQG	Applied LQR for quadrotor trajectory tracking, improving energy efficiency and robustness.	No	Computational overhead limits real-time use.
Velagić et al. (2022)	LQR	Integrated stochastic noise handling with linear-quadratic optimization for noisy environments.	Yes	Requires accurate state estimation.

Peixoto et al. (2023)	Sliding Mode	Smooth sliding control alleviated chattering and enhanced trajectory tracking accuracy, validated on a commercial hexacopter.	No	Limited to hexacopter experiments; scalability to other UAV types not demonstrated
Zhang et al. (2024)	Robust H_∞	Proposed a robust H_∞ controller to stabilize UAV flight with attached manipulators, validated experimentally.	No	Limited to aerial manipulators; scalability to other UAV systems not explored.

2.5.2 Model Predictive Controller (MPC)

Model Predictive Control (MPC) is an advanced control strategy widely used in industrial and process control applications. It is a type of control algorithm that uses a mathematical model of the system to predict its future behaviour and optimize control inputs to achieve desired outcomes. MPC excels in handling complex systems with constraints and multivariable interactions.

The structure of Model Predictive Control (MPC) consists of several interconnected components that work together to optimize the control of a system. At its core is the System Model, a mathematical representation of the process being controlled, which is used to predict future system behaviour based on current states and inputs. The Prediction Horizon defines the future time span over which the system's behaviour is predicted, allowing the controller to anticipate and mitigate potential deviations from desired outcomes. Within this horizon lies the Control Horizon, a shorter timeframe during which control actions are optimized to achieve desired results.

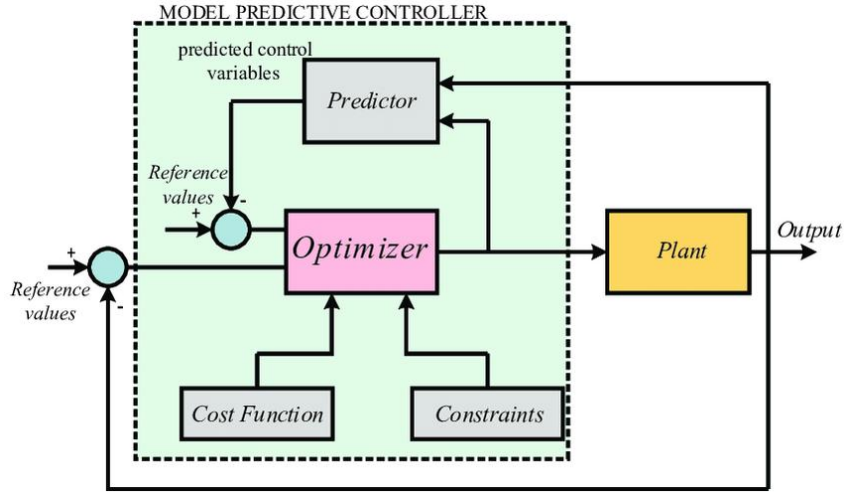


Figure 2.1: MPC Controller Block

Numerous studies highlight advancements in Model Predictive Control (MPC) systems for UAVs, focusing on enhancing stability, trajectory tracking, and robustness in dynamic environments. Mohammadi et al. (2023) develop an MPC system for hybrid UAVs, integrating tracking error variables to manage flight mode transitions. Similarly, Jing et al. (2023) apply MPC for controlling transitions in hybrid UAVs with rotary and fixed-wing capabilities, addressing over-actuation challenges and optimizing control signals. However, the reliance on linear models in these studies limits their ability to fully capture complex UAV dynamics, underscoring the need for nonlinear modelling and real-world testing.

In addressing nonlinear UAV dynamics, Singh and Kumar (2023) combine MPC with a Linear Parameter Varying (LPV) approach for quadrotor attitude stabilization and trajectory tracking, demonstrating robust disturbance rejection in simulations. Liao and Song (2023) focus on quad tiltrotor UAVs, developing an MPC system that incorporates aerodynamic effects during cruising, maintaining stability under disturbances. Similarly, Chen and Tan (2023) propose a Nonlinear Model Predictive Control (NMPC) strategy for

lateral path control of fixed-wing UAVs, outperforming conventional methods in simulations. While these studies showcase MPC's potential in handling nonlinearities, real-world validation remains a critical gap to ensure effectiveness under unpredictable conditions.

Other researchers explore innovative MPC adaptations to improve UAV performance. Jiang et al. (2022) introduce a neural network-based MPC (NNMPC) for precise position control, using flight data to eliminate the need for detailed aerodynamic modelling. This simplifies implementation while reducing trajectory tracking error. Mamdough et al. (2023) propose an NMPC system for efficient trajectory tracking and obstacle avoidance, leveraging real-time optimization to achieve robust performance with limited onboard resources. Martínez-León et al. (2023) combine adaptive control with MPC to enhance UAV stability in wind-disturbed environments, managing external uncertainties and improving performance in simulations. Despite their advancements, these approaches remain constrained by computational demands and the lack of experimental validation in real-world scenarios.

Several studies highlight MPC applications for specific challenges like collision avoidance and navigation in complex environments. Novak and Tebbani (2023) develop a decentralized NMPC strategy for multi-UAV systems, ensuring efficient trajectory tracking and collision avoidance through prioritization and energy optimization. Saccani et al. (2022) introduce a multi-trajectory MPC (mt-MPC) approach for UAV navigation in unknown environments, integrating real-time sensor data to balance safety and efficiency amidst obstacles. These systems show promise in simulations and limited experimental tests, but scalability across diverse platforms and environments remains unproven.

Finally, advancements in MPC with fuzzy logic and anticipative actions demonstrate its potential for trajectory optimization. Xu et al. (2023) present a decoupled MPC approach for position and attitude control, simplifying modelling and reducing computational load while maintaining trajectory accuracy under disturbances. Cavanini et al. (2021) integrate anticipative actions using Linear Parameter-Varying and Linear Time-Invariant models to improve trajectory tracking. While these approaches enhance control performance, their dependence on simulations and algorithmic complexity highlight the need for further real-world testing and refinement. Table 2.6 summarize the literature review of MPC controller.

Table 2.6: Summary of MPC controller literature review

Authors	Controller	Key Findings	Wind Disturbance	Limitations
Singh and Kumar (2023)	MPC Linear Parameter Varying	Demonstrates superior trajectory tracking and robustness against wind disturbances.	Yes	Reliance on simulations may not fully capture real-world conditions.
Kumar, A et al. (2023)	MPC	Minimizes jerk and power consumption while avoiding obstacles.	Yes	Simulations may not reflect challenges in unpredictable environments.
Xu, Z et al. (2023)	MPC	Enhances aggressive trajectory tracking accuracy.	Yes	Limited real-world validation.
Cavanini, L et al. (2021)	MPC-LPV	Significant improvements in control accuracy and robustness through simulations.	No	Simulation-based validation may not capture the full range of real-world challenges.
Brajesh Singh and Awadhesh Kumar	MPC-LPV	Significantly improves trajectory tracking precision against wind gusts.	Yes	Simulations may not capture real-world conditions.

(2023)				
Borbolla-Burillo et al. (2023)	Cascaded MPC	Successfully enhances trajectory tracking and disturbance rejection, validated in real-world UAV applications.	Yes	Complexity and scalability of cascaded MPC may limit its use in smaller UAV platforms; requires more varied environment tests.

2.6 Research Gap

Current UAV control systems often overlook complex aerodynamic effects and environmental variables, limiting their effectiveness in challenging environments like mangrove forests, where factors such as varying air density, humidity, and unpredictable wind patterns significantly impact performance. Existing studies rely heavily on simulations that fail to replicate the complexity of real-world conditions, and limited efforts have been made to validate these control systems in dynamic, real-world environments. Additionally, many advanced control strategies, such as hybrid UAV transitions and cascaded MPC architectures, are tested only in controlled environments, lacking adaptability to the operational challenges presented by unique ecosystems like mangrove forests.

This research addresses the gap by focusing on the development and validation of a robust cascade MPC control system that integrates environmental uncertainties into UAV models and undergoes rigorous experimental testing in real-world settings. This approach aims to enhance UAV reliability and adaptability for effective navigation and performance in complex environments like mangrove forests.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

The research modelling and experimental setup for UAV Quadcopter are described in this chapter. Besides that, the chapter will explain the flowchart of the algorithm, wind characteristics in mangrove forest, modelling of UAV Dynamic and development of intelligent control system, and how the model validation is conducted in this research. Additionally, it will outline the steps taken to complete the project, which will be built around the objectives outlined in Chapter 1.

3.2 Research Flowchart

A comprehensive literature research on the topic was carried out so that a suitable decision could be made on the path that the study would take. To accomplish the goals that have been set for this endeavour, we will be utilising both analytical and experimental methods.

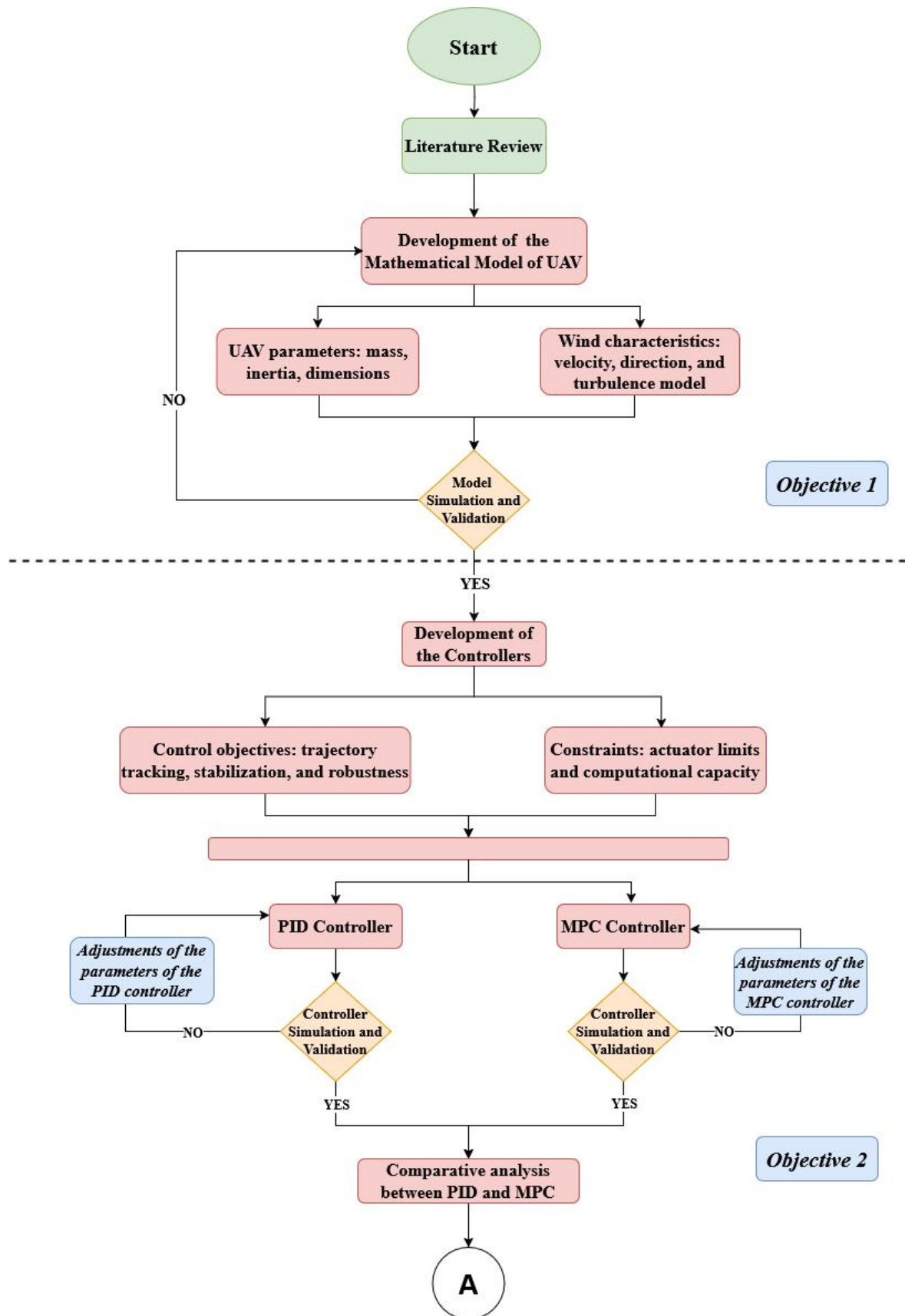
First, some background information about quadcopters will be provided for those who are unfamiliar with them. To comprehend a quadrotor system, it is necessary, for instance, to be familiar with the process of operating a quadrotor or to model its dynamic behaviour. In addition, the literature review receives consistent updates throughout each stage of the process. The primary purpose of this stage is to get a comprehensive and in-depth knowledge of the quadrotor system and the mechanism it utilises. All of this will be taken care of within the context of the project's literature evaluation.

The next step is to derive and develop the mathematical model of the system using the Newton-Euler method, this method is based on Newton's second law equations on rigid body. By using this method, we are going to obtain the Kinematic and dynamic equations. Those equations will be implemented in MATLAB, and using Simulink, as part of the many features of MATLAB. After obtaining the correspondent mathematical model of the system, a test validation is needed to know whether our system is valid and accurate or need to be adjusted and optimised.

The following step after that, is to design and develop a robust control system to control our model with special focus on wind turbulence disturbance in mangrove forest which are typically considered as challenges environment. The controller will be designed using MPC controller. It has been applied in many scientific and engineering fields. In the flight control of a quadrotor, the algorithm is usually used to achieve stable flight. After this, a test performance will be done for our intelligent controller.

In the following step, experiments tests will be conducted to evaluate the performance of our algorithm and to see how well this Intelligent technique of control operate on the drone when it is hovering and or moving in mangrove area.

Finally, a critical comparison between our intelligent algorithm and the classical PID controller will be conducted to evaluate and determine the optimization and performance of this novel techniques. Figure 3.1 illustrates the Research Flowchart implemented in this research project:



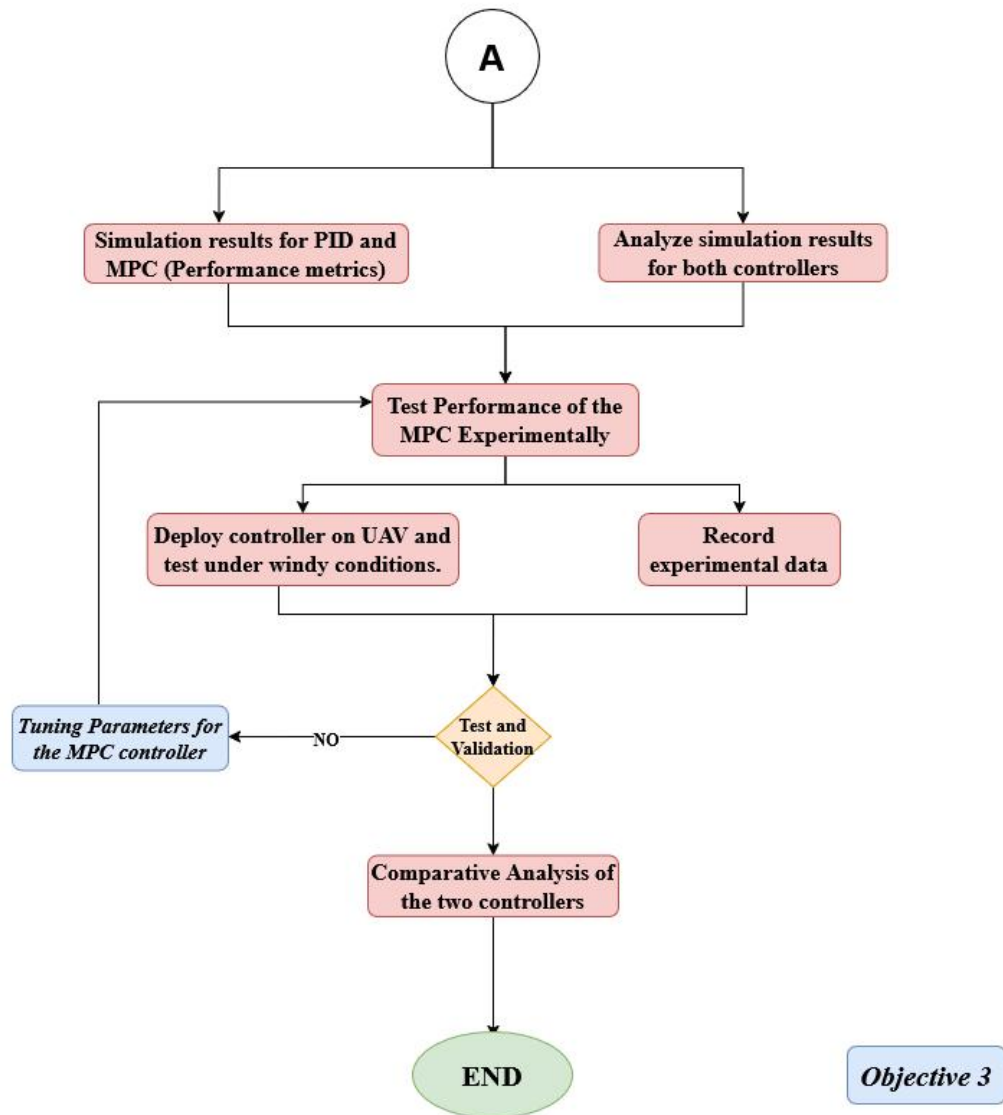


Figure 3.1: Research Flowchart

3.3 Mathematical Model Development

The section covers the development of a UAV mathematical model and wind turbulence model to enhance control strategies, ensuring stability and performance in real-world, wind-affected environments like mangrove forests.

3.3.1 Wind Model Implementation in Simulink

Before diving into simulation, it's crucial to choose an appropriate model for wind effects in mangrove forest. Common models include the Dryden Wind Turbulence Model as can be seen in Figure 3.2, which can be found in Aerospace Block set in Simulink, which is widely used in aerospace for its simplicity and effectiveness in representing lower altitude atmospheric turbulence and gusts.

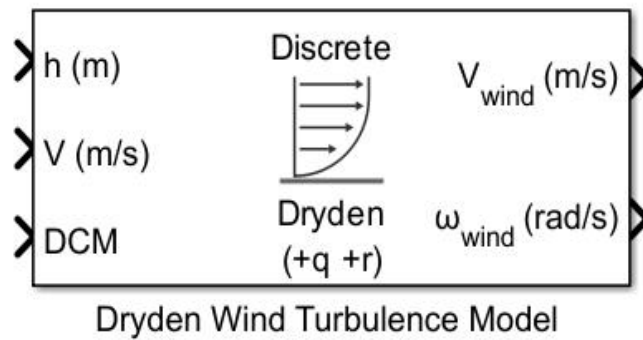


Figure 3.2: Dryden wind turbulence simulink block

Where: $h(m)$: is Altitude, $V(m/s)$: Aircraft speed, DCM: Direction cosine matrix, $V_{wind} (m/s)$: Turbulence velocities, returned as a three-element vector in the same body coordinate reference as the DCM input, $\omega_{wind}(rad/s)$: Turbulence angular rates, specified as a three-element vector. (more details about the wind characteristics can be found in Appendix A page 150)

3.3.2 Reference Frames Orientation of the UAV

The quadrotor, with its four propulsion engines and electronic control board, is designed for balance and aerial navigation. Understanding its structure and spatial orientation requires defining two key reference systems: a stationary (inertial) system and a mobile system. The inertial system, based on Newton's first law, uses the NED (North-East-Down) schema to align coordinate axes with cardinal directions and a downward

vertical axis. This alignment provides a geographic and gravitational context for the quadrotor's position and direction. Figure 3.3 illustrates this coordinate system, which is fundamental for understanding the quadrotor's navigation, operation, and control strategies.

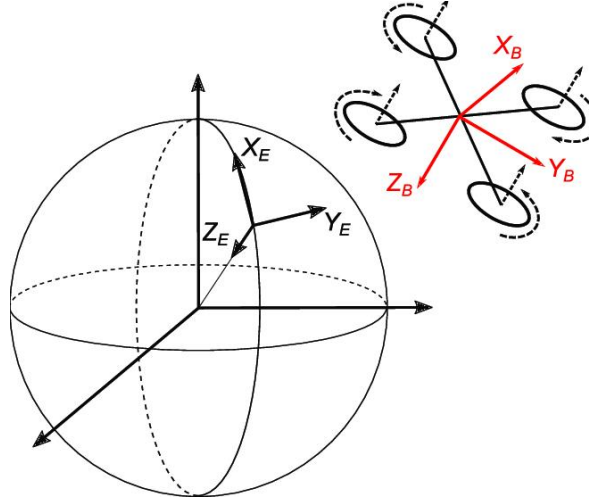


Figure 3.3: Coordinate System

The previously referenced mobile coordinate system is integrally aligned with the quadrotor's centre of mass, a concept widely acknowledged in scientific discourse as the OABC system, with "ABC" denoting Aircraft Body Centre. This nomenclature and its operational significance are underscored in Figure 3.4, which delineates the juxtaposition of both the stationary and mobile reference frameworks.

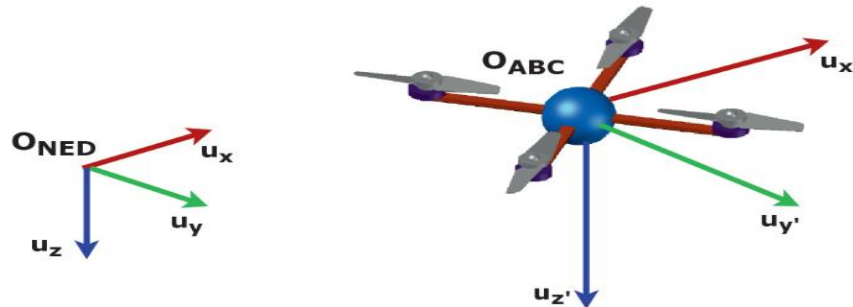


Figure 3.4: Stationary and mobile reference frameworks

Controlling a quadrotor's attitude and precise positioning involves regulating the speeds of its four motors, which exert various forces and moments for movement and stabilization in three-dimensional space. Key forces include thrust from rotor rotation and pitch and roll moments from differential thrust across the rotors. The quadrotor also experiences gravitational pull, the gyroscopic effect in lightweight frames, and yaw moments from imbalances in rotor speeds.

To mitigate yaw moments, two rotors are arranged to rotate inversely to the other pair, balancing yaw forces and aiding in precise manoeuvring. This setup divides the propellers into two opposing groups of two motors each, enhancing aerodynamic efficiency and control. The rotation direction of the motors distinguishes the motor groups and optimizes flight dynamics and responsiveness. A configuration can be readily identified as follows:

- front and rear propellers (numbers 1 and 2 in Figure 3.5), rotating counterclockwise.
- right and left propellers (numbers 4 and 3 in Figure 3.5), rotating clockwise.

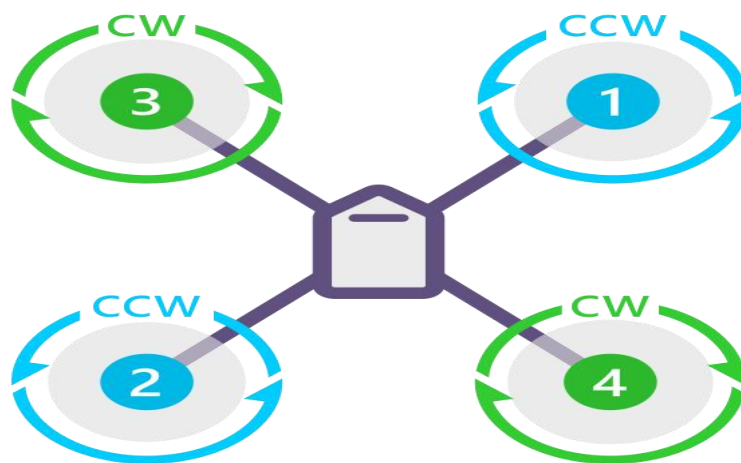


Figure 3.5: UAV propeller configuration

The kinematics of a quadrotor involve translational movements of the centre of mass and rotational movements around it, with six degrees of freedom: three translational (forward/backward, lateral, vertical) and three rotational (roll, pitch, yaw) along the Cartesian axes. Motor rotational velocities control these degrees of freedom. Directional and rotational movements are managed by varying motor speeds, with yaw movement generated by reactive torque proportional to rotor speed. Equilibrium is achieved when all rotor speeds are equal, balancing reactive torques, while differing speeds induce rotation and uniform speed adjustments enable vertical displacement.

As an underactuated nonlinear system, the quadrotor has four inputs (motor speeds) and six outputs (degrees of freedom). Modelling assumptions include treating the quadrotor as a rigid, symmetric body and ignoring ground effect. Figure 3.6 shows the relationship between propeller dynamics and the aircraft's motion.

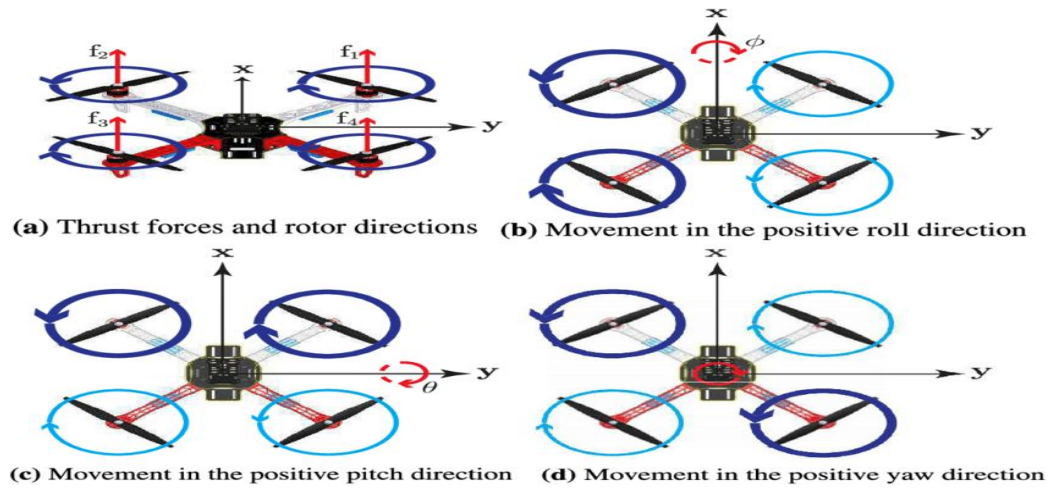


Figure 3.6: Interplay between Rotor dynamics and the UAV's motion

3.3.3 Euler angles

Euler angles, named after Leonhard Euler, define the orientation of a rigid body in three-dimensional Euclidean space using three parameters. This section adopts the ZYX convention for Euler angles (Siciliano et al., 2000). These angles, denoted as ϕ , θ , and ψ , range within $\phi \in]-\pi, \pi [$, $\theta \in]-\pi/2, \pi/2 [$, and $\psi \in]-\pi, \pi [$. They describe the orientation of one reference frame relative to another and are crucial for transforming a point's coordinates between frames. Euler angles represent a sequence of three rotations about coordinate axes, enabling any orientation from a standard reference point. These rotations are mathematically expressed through specific rotation matrices (Lee, D et al., 2009).

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\phi) & -s(\phi) \\ 0 & s(\phi) & c(\phi) \end{bmatrix} \quad \text{Equation 3.1}$$

$$R_y(\theta) = \begin{bmatrix} c(\theta) & 0 & s(\theta) \\ 0 & 1 & 0 \\ -s(\theta) & 0 & c(\theta) \end{bmatrix} \quad \text{Equation 3.2}$$

$$R_z(\psi) = \begin{bmatrix} c(\psi) & -s(\psi) & 0 \\ s(\psi) & c(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{Equation 3.3}$$

where: $c(\phi) = \cos(\phi)$, $s(\phi) = \sin(\phi)$, $c(\theta) = \cos(\theta)$, $s(\theta) = \sin(\theta)$, $c(\psi) = \cos(\psi)$, $s(\psi) = \sin(\psi)$.

So, the inertial position coordinates and the body reference coordinates are related by the rotation matrix $R_{zyx}(\phi, \theta, \psi) \in SO(3)$:

$$R_{zyx}(\phi, \theta, \psi) = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi & -c_\phi s_\psi & c_\phi s_\theta c_\psi & + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi & + c_\phi c_\psi & c_\phi s_\theta s_\psi & - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & & c_\phi c_\theta & \end{bmatrix} \quad \text{Equation 3.4}$$

This matrix describes the rotation from the body reference system to the inertial reference.

3.3.4 Quadrotor mathematical model

In this section a detailed mathematical model of the quadrotor's dynamics is presented (Dyer 2018), with integration of Newton's laws for translational motion and Euler's formulations for rotational behaviour. the key parameters of a UAV quadcopter, including its mass, arm length, moments of inertia (I_x , I_y , I_z), maximum thrust, gravitational acceleration, and maximum rotor speed is presented in Table 3.1.

Table 3.1: UAV Quadcopter Parameters

Parameter	Description	Value	Unit
m	Total UAV mass	1.32	Kg
l	Arm length	0.5	m
I_x	Moment of Inertia X	0.0093	$\text{Kg} \cdot \text{m}^2$
I_y	Moment of Inertia Y	0.0093	$\text{Kg} \cdot \text{m}^2$
I_z	Moment of Inertia Z	0.0183	$\text{Kg} \cdot \text{m}^2$
f_t	Maximum thrust	13.067	N
g	Gravitational acceleration	9.81	$\text{m} \cdot \text{s}^2$
ω_{max}	Maximum rotor speed	2000	rad/s

These parameters define the quadcopter's physical and dynamic properties, serving as a foundation for modelling, simulation, and control system design for flight performance.

This method is essential for understanding the complex three-dimensional movements of a rigid aerial vehicle and enhances our comprehension of the quadrotor's

operational dynamics. The model provides the necessary accuracy for simulating and effectively controlling the quadrotor's flight patterns.

To support this analysis, the notation $[x \ y \ z \ \phi \ \theta \ \psi]^T$ is used to represent the quadrotor's linear and angular positions in the inertial reference frame. The vector $[u_x \ v_y \ w_z \ p \ q \ r]^T$ is used to capture its linear and angular velocities in the body fixed frame. This distinction between frames allows for a comprehensive understanding of the quadrotor's motion, considering both orientation and velocity. Equations that govern the conversion of coordinates and velocities between these reference frames are based on rigid body dynamics principles.

This mathematical framework not only provides a theoretical understanding of the quadrotor's flight dynamics but also offers a practical guide for designing control systems that manage the complex forces and moments acting on the vehicle. The two reference frames are linked by the following relations:

$$\mathbf{v} = \mathbf{R} \cdot \mathbf{v}_B \quad \text{Equation 3.5}$$

$$\boldsymbol{\omega} = \mathbf{T} \cdot \boldsymbol{\omega}_B \quad \text{Equation 3.6}$$

Where:

$$\mathbf{v} = [\dot{x} \ \dot{y} \ \dot{z}]^T \in \mathbb{R}^3 \quad \text{Equation 3.7}$$

$$\boldsymbol{\omega} = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T \in \mathbb{R}^3 \quad \text{Equation 3.8}$$

$$\mathbf{v}_B = [u_x \ v_y \ w_z]^T \in \mathbb{R}^3 \quad \text{Equation 3.9}$$

$$\boldsymbol{\omega}_B [p \ q \ r]^T \in \mathbb{R}^3 \quad \text{Equation 3.10}$$

And T is a matrix for angular transformations.

$$T = \begin{bmatrix} 1 & s(\phi)t(\theta) & c(\phi)t(\theta) \\ 0 & c(\phi) & -s(\phi) \\ 0 & \frac{s(\phi)}{c(\theta)} & \frac{c(\phi)}{c(\theta)} \end{bmatrix} \quad \text{Equation 3.11}$$

Where $t(\theta) = \tan(\theta)$. So, the kinematic model of the quadrotor is:

$$\dot{x} = \begin{bmatrix} s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta) \\ -[c(\phi)s(\psi) - c(\psi)s(\phi)s(\theta)] \\ c(\psi)c(\theta) \end{bmatrix} \cdot \begin{bmatrix} w_z \\ v_y \\ u_x \end{bmatrix} \quad \text{Equation 3.12}$$

$$\dot{y} = \begin{bmatrix} c(\phi)c(\psi) + s(\phi)s(\psi)s(\theta) \\ -[c(\psi)s(\phi) - c(\phi)s(\psi)s(\theta)] \\ c(\theta)s(\psi) \end{bmatrix} \cdot \begin{bmatrix} w_z \\ v_y \\ u_x \end{bmatrix} \quad \text{Equation 3.13}$$

$$\dot{z} = \begin{bmatrix} c(\phi)c(\theta) \\ -s(\theta) \\ c(\theta)s(\phi) \end{bmatrix} \cdot \begin{bmatrix} w_z \\ v_y \\ u_x \end{bmatrix} \quad \text{Equation 3.14}$$

$$\dot{\phi} = \begin{bmatrix} 1 \\ c(\phi)t(\theta) \\ s(\phi)t(\theta) \end{bmatrix} \cdot \begin{bmatrix} p \\ r \\ q \end{bmatrix} \quad \text{Equation 3.15}$$

$$\dot{\theta} = \begin{bmatrix} c(\phi) \\ -s(\phi) \end{bmatrix} \cdot \begin{bmatrix} q \\ r \end{bmatrix} \quad \text{Equation 3.16}$$

$$\dot{\psi} = \begin{bmatrix} \frac{c(\phi)}{c(\theta)} & \frac{s(\phi)}{c(\theta)} \end{bmatrix} \cdot \begin{bmatrix} r \\ q \end{bmatrix} \quad \text{Equation 3.17}$$

Newton's law states the following matrix relation for the total force acting on the quadrotor:

$$m(\omega_B \wedge v_B + \dot{v}_B) = f_B \quad \text{Equation 3.18}$$

Where m is the mass of the quadrotor, $\wedge$ is the cross product and $f_B = [f_x \ f_y \ f_z]^T \in \mathbb{R}^3$

is the total force. Euler's equation gives the total torque applied to the quadrotor:

$$I \cdot \dot{\omega}_B + \omega_B \wedge (I \cdot \omega_B) = m_B \quad \text{Equation 3.19}$$

where $\mathbf{m}_B = [m_x \quad m_y \quad m_z]^T \in \mathbb{R}^3$ is the total torque and $\mathbf{I}$ is the diagonal inertia matrix:

$$\mathbf{I} = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \in \mathbb{R}^{3 \times 3} \quad \text{Equation 3.20}$$

So, the dynamic model of the quadrotor in the body frame is:

$$f_x = m(\dot{u}_x + qw_z - rv_y) \quad \text{Equation 3.21}$$

$$f_y = m(\dot{v}_y - pw_z + ru_x) \quad \text{Equation 3.22}$$

$$f_z = m(\dot{w}_z + pv - qu_x) \quad \text{Equation 3.23}$$

$$m_x = \dot{p}I_x - qrI_y + qrI_z \quad \text{Equation 3.24}$$

$$m_y = \dot{q}I_y + prI_x - prI_z \quad \text{Equation 3.25}$$

$$m_z = \dot{r}I_z - pqI_x + pqI_y \quad \text{Equation 3.26}$$

The equations stand if we assume that the origin and the axes of the body frames coincide with the barycentre of the quadrotor and the principal axes.

3.3.5 Forces and Moments

The external forces in the body frame, f_B are given by:

$$\mathbf{f}_B = m\mathbf{g}\mathbf{R}^T \cdot \hat{\mathbf{e}}_z - f_t \hat{\mathbf{e}}_3 + \mathbf{f}_w, \quad \text{Equation 3.27}$$

where $\hat{\mathbf{e}}_z$ is the unit vector in the inertial z axis, $\hat{\mathbf{e}}_3$ is the unit vector in the body z axis, $\mathbf{g}$ is the gravitational acceleration, f_t is the total thrust generated by rotors and $\mathbf{f}_w = [f_{wx} \quad f_{wy} \quad f_{wz}]^{Tr} \in \mathbb{R}^3$ are the forces produced by wind on the quadrotors. The external moments in the body frame, $\mathbf{m}_B$ are given by:

$$\mathbf{m}_B = \boldsymbol{\tau}_B - \mathbf{g}_a + \boldsymbol{\tau}_w, \quad \text{Equation 3.28}$$

where $\mathbf{g}_a$ represents the gyroscopic moments caused by the combined rotation of the four rotors and the vehicle body, $\boldsymbol{\tau}_B = [\tau_x \quad \tau_y \quad \tau_z]^T \in \mathbb{R}^3$ are the control torques generated by differences in the rotor speeds and $\boldsymbol{\tau}_w = [\tau_{wx} \quad \tau_{wy} \quad \tau_{wz}]^T \in \mathbb{R}^3$ are the torques produced by wind on the quadrotors, $\mathbf{g}_a$ is given by:

$$\mathbf{g}_a = \sum_{i=1}^4 J_p (\boldsymbol{\omega}_B \wedge \hat{\mathbf{e}}_3) (-1)^{i+1} \Omega_i \quad \text{Equation 3.29}$$

In this context, J_p denotes the moment of inertia for each rotor, while Ω_i represents the angular velocity of the i th rotor. According to Mellinger et al. (2011), J_p

is negligible, excluding gyroscopic moments from the control algorithm. Quadrotor flight is influenced by aerodynamic and aeroelastic factors, including ground effect phenomena, especially near the ground or during landing when propeller airflow significantly alters dynamics. Therefore, a comprehensive quadrotor dynamic model must incorporate the forces described in the following equations:

$$-mg[s(\theta)] + f_{wx} = m(\dot{u} + qw - rv) \quad \text{Equation 3.30}$$

$$mg[c(\theta)s(\phi)] + f_{wy} = m(\dot{v} - pw + ru) \quad \text{Equation 3.31}$$

$$mg[c(\theta)c(\phi)] + f_{wz} - f_t = m(\dot{w} + pv - qu) \quad \text{Equation 3.32}$$

$$\tau_x + \tau_{wx} = \dot{p}I_x - qrI_y + qrI_z \quad \text{Equation 3.33}$$

$$\tau_y + \tau_{wy} = \dot{q}I_y + prI_x - prI_z \quad \text{Equation 3.34}$$

$$\tau_z + \tau_{wz} = \dot{r}I_z - pqI_x + pqI_y \quad \text{Equation 3.35}$$

3.3.6 Actuator Dynamics

The various inputs that can be manipulated to regulate the quadrotor's operational dynamics are explored in this section. The system is equipped with four rotors, corresponding to an equal number of controllable degrees of freedom. Typically, the control inputs are categorized into one for generating vertical thrust and three for managing the respective angular motions. It is posited that the magnitudes of the input forces and moments are directly proportional to the square of the rotor speeds, as delineated in reference (Bresciani T., 2008). Their values are detailed in the following equations.

$$f_t = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \quad \text{Equation 3.36}$$

$$\tau_x = bl(\Omega_3^2 - \Omega_1^2) \quad \text{Equation 3.37}$$

$$\tau_y = bl(\Omega_4^2 - \Omega_2^2) \quad \text{Equation 3.38}$$

$$\tau_z = d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \quad \text{Equation 3.39}$$

where l is the distance between any rotor and the centre of the drone, b is the thrust factor and d is the drag factor. So, the dynamic model of the quadrotor in the body frame is:

$$-mg[s(\theta)] + f_{wx} = m(\dot{u}_x + qw_z - rv_y) \quad \text{Equation 3.40}$$

$$mg[c(\theta)s(\phi)] + f_{wy} = m(\dot{v}_y - pw_z + ru_x) \quad \text{Equation 3.41}$$

$$[mg[c(\theta)c(\phi)]] = [-m \quad -m \quad m \quad b \quad b \quad b \quad b \quad 1] \cdot \begin{bmatrix} \dot{w}_z \\ pv_y \\ qu_x \\ \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \\ f_{wz} \end{bmatrix} \quad \text{Equation 3.42}$$

$$bl(\Omega_3^2 - \Omega_1^2) + \tau_{wx} = \dot{p}I_x - qrI_y + qrI_z \quad \text{Equation 3.43}$$

$$bl(\Omega_4^2 - \Omega_2^2) + \tau_{wy} = \dot{q}I_y + prI_x - prI_z \quad \text{Equation 3.44}$$

$$d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 + \Omega_3^2) + \tau_{wz} = \dot{r}I_z - pqI_x + pqI_y \quad \text{Equation 3.45}$$

3.3.7 State Space Model

Organizing the state's vector in the following way:

$$X = [\phi \quad \theta \quad \psi \quad p \quad q \quad r \quad u_x \quad v_y \quad w_z \quad x \quad y \quad z]^T \in \mathbb{R}^{12} \quad \text{Equation 3.46}$$

It is possible to rewrite the equations of the dynamics of the quadrotor in the state-space as follows in the following equations:

$$\dot{\phi} = [1 \quad c(\phi)t(\theta) \quad s(\phi)t(\theta)] \cdot \begin{bmatrix} p \\ r \\ q \end{bmatrix} \quad \text{Equation 3.47}$$

$$\dot{\theta} = [c(\phi) \quad -s(\phi)] \cdot \begin{bmatrix} q \\ r \end{bmatrix} \quad \text{Equation 3.48}$$

$$\dot{\psi} = \begin{bmatrix} \frac{c(\phi)}{c(\theta)} & \frac{s(\phi)}{c(\theta)} \end{bmatrix} \cdot \begin{bmatrix} r \\ q \end{bmatrix} \quad \text{Equation 3.49}$$

$$\dot{p} = \begin{bmatrix} \frac{I_y - I_z}{I_x} & \frac{1}{I_x} & \frac{1}{I_x} \end{bmatrix} \cdot \begin{bmatrix} rq \\ \tau_x \\ \tau_{wx} \end{bmatrix} \quad \text{Equation 3.50}$$

$$\dot{q} = \begin{bmatrix} \frac{I_z - I_x}{I_y} & \frac{1}{I_y} & \frac{1}{I_y} \end{bmatrix} \cdot \begin{bmatrix} pr \\ \tau_y \\ \tau_{wy} \end{bmatrix} \quad \text{Equation 3.51}$$

$$\dot{r} = \begin{bmatrix} \frac{I_x - I_y}{I_z} & \frac{1}{I_z} & \frac{1}{I_z} \end{bmatrix} \cdot \begin{bmatrix} pq \\ \tau_z \\ \tau_{wz} \end{bmatrix} \quad \text{Equation 3.52}$$

$$\dot{u} = \begin{bmatrix} v_y & -w_z & -s(\theta) & \frac{1}{m} \end{bmatrix} \cdot \begin{bmatrix} r \\ q \\ g \\ f_{wx} \end{bmatrix} \quad \text{Equation 3.53}$$

$$\dot{v} = \begin{bmatrix} w_z & -u_x & s(\phi)c(\theta) & \frac{1}{m} \end{bmatrix} \cdot \begin{bmatrix} p \\ r \\ g \\ f_{wy} \end{bmatrix} \quad \text{Equation 3.54}$$

$$\dot{w} = \begin{bmatrix} u_x & -v_y & c(\theta)c(\phi) & \frac{1}{m} \end{bmatrix} \cdot \begin{bmatrix} q \\ p \\ g \\ f_{wz} - f_t \end{bmatrix} \quad \text{Equation 3.55}$$

$$\dot{x} = \begin{bmatrix} s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta) \\ -c(\phi)s(\psi) + c(\psi)s(\phi)s(\theta) \\ c(\psi)c(\theta) \end{bmatrix}^{Tr} \cdot \begin{bmatrix} w_z \\ v_y \\ u_x \end{bmatrix} \quad \text{Equation 3.56}$$

$$\dot{y} = \begin{bmatrix} c(\phi)c(\psi) + s(\phi)s(\psi)s(\theta) \\ -c(\psi)s(\phi) + c(\phi)s(\psi)s(\theta) \\ c(\theta)s(\psi) \end{bmatrix}^{Tr} \cdot \begin{bmatrix} w_z \\ v_y \\ u_x \end{bmatrix} \quad \text{Equation 3.57}$$

$$\dot{z} = \begin{bmatrix} c(\phi)c(\theta) \\ -s(\theta) \\ c(\theta)s(\phi) \end{bmatrix}^{Tr} \cdot \begin{bmatrix} w_z \\ v_y \\ u_x \end{bmatrix} \quad \text{Equation 3.58}$$

Below we obtain two alternative forms of the dynamical model useful for studying the control. From Newton's law we can write:

$$m\dot{v} = R \cdot f_B = mg\hat{e}_z - f_t R \cdot \hat{e}_3 \quad \text{Equation 3.59}$$

Therefore:

$$\ddot{x} = -\frac{f_t}{m} [s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta)] \quad \text{Equation 3.60}$$

$$\ddot{y} = -\frac{f_t}{m}[c(\phi)s(\psi)s(\theta) - c(\psi)s(\phi)] \quad \text{Equation 3.61}$$

$$\ddot{z} = g - \frac{f_t}{m}[c(\phi)c(\theta)] \quad \text{Equation 3.62}$$

Now a simplification is made by setting $[\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^{Tr} = [p \ q \ r]^{Tr}$. This assumption holds true for small angles of movement (Das et al., 2009). So, the dynamic model of the quadrotor in the inertial frame is:

$$\ddot{x} = -\frac{f_t}{m}[s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta)] \quad \text{Equation 3.63}$$

$$\ddot{y} = -\frac{f_t}{m}[c(\phi)s(\psi)s(\theta) - c(\psi)s(\phi)] \quad \text{Equation 3.64}$$

$$\ddot{z} = g - \frac{f_t}{m}[c(\phi)c(\theta)] \quad \text{Equation 3.65}$$

$$\ddot{\phi} = \frac{I_y - I_z}{I_x}\dot{\theta}\dot{\psi} + \frac{\tau_x}{I_x} \quad \text{Equation 3.66}$$

$$\ddot{\theta} = \frac{I_z - I_x}{I_y}\dot{\phi}\dot{\psi} + \frac{\tau_y}{I_y} \quad \text{Equation 3.67}$$

$$\ddot{\psi} = \frac{I_x - I_y}{I_z}\dot{\phi}\dot{\theta} + \frac{\tau_z}{I_z} \quad \text{Equation 3.68}$$

Redefining the state's vector as:

$$X = [x \ y \ z \ \psi \ \theta \ \phi \ \dot{x} \ \dot{y} \ \dot{z} \ p \ q \ r]^{Tr} \in \mathbb{R}^{12} \quad \text{Equation 3.69}$$

It is possible to rewrite the equations of the quadrotor in the state-space:

$$\dot{X} = f(x) + \sum_{i=1}^4 g_i(x)u_i \quad \text{Equation 3.70}$$

Where:

$$f(x) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ q \frac{s(\phi)}{c(\theta)} + r \frac{c(\phi)}{c(\theta)} \\ q[c(\phi)] - r[s(\phi)] \\ p + q[s(\phi)t(\theta)] + r[c(\phi)t(\theta)] \\ 0 \\ 0 \\ g \\ \frac{(I_y - I_z)}{I_x} qr \\ \frac{(I_z I_x)}{I_y} pr \\ \frac{(I_x - I_y)}{I_z} pq \end{bmatrix} \quad \text{Equation 3.71}$$

and:

$$g_1(x) = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ g_1^7 \ g_1^8 \ g_1^9 \ 0 \ 0 \ 0]^T \in \mathbb{R}^{12} \quad \text{Equation 3.72}$$

$$g_2(x) = \left[0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ \frac{1}{I_x} \ 0 \ 0 \right]^T \in \mathbb{R}^{12} \quad \text{Equation 3.73}$$

$$g_3(x) = \left[0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ \frac{1}{I_y} \ 0 \right]^T \in \mathbb{R}^{12} \quad \text{Equation 3.74}$$

$$g_4(x) = \left[0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ \frac{1}{I_z} \right]^T \in \mathbb{R}^{12} \quad \text{Equation 3.75}$$

With:

$$g_1^7 = -\frac{1}{m} [s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta)] \quad \text{Equation 3.76}$$

$$g_1^8 = -\frac{1}{m} [c(\psi)s(\phi) - c(\phi)s(\psi)s(\theta)] \quad \text{Equation 3.77}$$

$$g_1^9 = -\frac{1}{m} [c(\phi)c(\theta)] \quad \text{Equation 3.78}$$

3.3.8 Linear Model

The control vector “u” is seen as follows: $u = [f_t \quad \tau_x \quad \tau_y \quad \tau_z]^T \in \mathbb{R}^4$. The linearization's procedure is developed around an equilibrium point $\bar{x}$, which for fixed input $\bar{u}$ is the solution of the algebraic system: or rather that value of state's vector, which on fixed constant input is the solution of algebraic system:

$$\hat{f}(\bar{x}, \bar{u}) = 0 \quad \text{Equation 3.79}$$

Given the nonlinear nature of the function $\hat{f}$, challenges emerge concerning the existence and uniqueness of solutions for system. The intricacies of deriving a closed-form solution are compounded by the interrelations of trigonometric functions in a manner that is not elementary. Consequently, to facilitate analysis, linearization is applied to a simplified model, commonly referred to as the "small oscillations" model.

This simplification entails approximating the sine function by its argument and the cosine function by one. Such approximations are held under the condition that the arguments in question are sufficiently small. Under these premises, the linearized system can be characterized by the ensuing equations:

$$\dot{\phi} \approx [1 \quad \theta \quad \phi\theta] \cdot \begin{bmatrix} p \\ r \\ q \end{bmatrix} \quad \text{Equation 3.80}$$

$$\dot{\theta} \approx [0 \quad 1 \quad 0] \begin{bmatrix} p \\ q \\ r \end{bmatrix} - [r][\phi] \quad \text{Equation 3.81}$$

$$\dot{\psi} \approx [0 \quad 0 \quad 1] \begin{bmatrix} p \\ q \\ r \end{bmatrix} + [0 \quad 1] \begin{bmatrix} \phi \\ \psi \end{bmatrix} \quad \text{Equation 3.82}$$

$$\dot{p} = \begin{bmatrix} \frac{I_y - I_z}{I_x} & \frac{1}{I_x} & \frac{1}{I_x} \end{bmatrix} \cdot \begin{bmatrix} rq \\ \tau_x \\ \tau_{wx} \end{bmatrix} \quad \text{Equation 3.83}$$

$$\dot{q} = \begin{bmatrix} \frac{I_z - I_x}{I_y} & \frac{1}{I_y} & \frac{1}{I_y} \end{bmatrix} \cdot \begin{bmatrix} pr \\ \tau_y \\ \tau_{wy} \end{bmatrix} \quad \text{Equation 3.84}$$

$$\dot{r} = \begin{bmatrix} \frac{I_x - I_y}{I_z} & \frac{1}{I_z} & \frac{1}{I_z} \end{bmatrix} \cdot \begin{bmatrix} pq \\ \tau_z \\ \tau_{wz} \end{bmatrix} \quad \text{Equation 3.85}$$

$$\dot{u}_x = \begin{bmatrix} v_y & -w_z & -\theta & \frac{1}{m} \end{bmatrix} \cdot \begin{bmatrix} r \\ q \\ g \\ f_{wx} \end{bmatrix} \quad \text{Equation 3.86}$$

$$\dot{v}_y = \begin{bmatrix} w_z & -u_x & \phi & \frac{1}{m} \end{bmatrix} \cdot \begin{bmatrix} p \\ r \\ g \\ f_{wy} \end{bmatrix} \quad \text{Equation 3.87}$$

$$\dot{w}_z = \begin{bmatrix} u_x & -v_y & 1 & \frac{1}{m} \end{bmatrix} \cdot \begin{bmatrix} q \\ p \\ g \\ f_{wz} - f_t \end{bmatrix} \quad \text{Equation 3.88}$$

$$\dot{x} = \begin{bmatrix} \phi\psi + \theta \\ -\psi + \phi\theta \\ 1 \end{bmatrix}^{Tr} \cdot \begin{bmatrix} w_z \\ v_y \\ u_x \end{bmatrix} \quad \text{Equation 3.89}$$

$$\dot{y} = \begin{bmatrix} 1 + \phi\psi\theta \\ -\phi + \psi\theta \\ \psi \end{bmatrix}^{Tr} \cdot \begin{bmatrix} v_y \\ w_z \\ u_x \end{bmatrix} \quad \text{Equation 3.90}$$

$$\dot{z} = \begin{bmatrix} 1 \\ -\theta \\ \phi \end{bmatrix}^{Tr} \cdot \begin{bmatrix} w_z \\ u_x \\ v_y \end{bmatrix} \quad \text{Equation 3.91}$$

Which can be written in the compact form:

$$\dot{X} = f(x, u) \quad \text{Equation 3.92}$$

As said above, to perform the linearization, an equilibrium point is needed. Such an equilibrium point can be:

$$\bar{X} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix} \in \mathbb{R}^{12} \quad \text{Equation 3.93}$$

From the equations, it can be found that the equilibrium point is obtained by the constant input value:

$$\bar{u} = \begin{bmatrix} mg \\ 0 \\ 0 \\ 0 \end{bmatrix} \in \mathbb{R}^4 \quad \text{Equation 3.94}$$

It is important to acknowledge that the value in question signifies the force required to neutralize the quadrotor's gravitational pull, thereby enabling its ability to hover. Upon establishing the equilibrium point, denoted as $\bar{X}$, and the corresponding nominal input, $\bar{u}$, it is observed that the matrices defining the linear system are delineated through the following relationships:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad \text{Equation 3.95}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{I_x} & 0 & 0 \\ 0 & 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & 0 & \frac{1}{I_z} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{Equation 3.96}$$

If we consider the disturbance by wind, set:

$$d = [f_{wx} \quad f_{wy} \quad f_{wz} \quad \tau_{wx} \quad \tau_{wy} \quad \tau_{wz}]^T \in \mathbb{R}^6 \quad \text{Equation 3.97}$$

$$D = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{I_x} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{I_z} \\ \frac{1}{m} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{m} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{Equation 3.98}$$

The linear model is:

$$\dot{x} = A \cdot x + B \cdot u + D \cdot d \quad \text{Equation 3.99}$$

$$\dot{\phi} = p \quad \text{Equation 3.100}$$

$$\dot{\theta} = q \quad \text{Equation 3.101}$$

$$\dot{\psi} = r \quad \text{Equation 3.102}$$

$$\dot{p} = \frac{\tau_x + \tau_{wx}}{I_x} \quad \text{Equation 3.103}$$

$$\dot{q} = \frac{\tau_y + \tau_{wy}}{I_y} \quad \text{Equation 3.104}$$

$$\dot{r} = \frac{\tau_x + \tau_{wx}}{I_z} \quad \text{Equation 3.105}$$

$$\dot{u}_x = -g\theta + \frac{f_{wx}}{m} \quad \text{Equation 3.106}$$

$$\dot{v}_y = g\phi + \frac{f_{wy}}{m} \quad \text{Equation 3.107}$$

$$\dot{w}_z = \frac{f_{wz} - f_t}{m} \quad \text{Equation 3.108}$$

$$\dot{x} = u_x \quad \text{Equation 3.109}$$

$$\dot{y} = v_y \quad \text{Equation 3.110}$$

$$\dot{z} = w_z \quad \text{Equation 3.111}$$

3.3.9 Quadcopter Holybro S500 Parameter Estimation

In this section, an exhaustive delineation of the quadcopter's physical characteristics is undertaken, providing a detailed examination of its structural and operational attributes. This analysis encompasses a broad spectrum of parameters, including but not limited to, weight, dimensions, propulsion system specifications, and material composition. Following a thorough investigation and characterization of these fundamental aspects, the findings are systematically aggregated and presented in a tabulated format at the conclusion of the chapter. This structured compilation serves not only as a reference but also as a foundational basis for subsequent analyses and discussions pertaining to the quadcopter's design, performance, and application potential.

3.3.9.1 Mass

The mass, m , of the quadcopter (with the battery connected) was measured using a digital mass scale. It was found to be approximately 1.250 kg.

3.3.9.2 Moment Arm

The determination of the moment arm, represented as d , an essential parameter for the quadcopter's structural and aerodynamic analysis, was conducted with precision. This process involved the careful measurement of the radial distance from the midpoint of any selected propeller directly to the quadcopter frame's geometric centre. Such measurements were meticulously carried out using a measuring tape, ensuring accuracy in capturing the spatial relationship between the propellers and the frame's nucleus.

This quantification is critical, as the moment arm plays a pivotal role in understanding the leverage effect exerted by the propellers on the quadcopter's stability and manoeuvrability. The exact value of the moment arm, derived from this methodical approach, is systematically documented, and is visually represented in Figure 3.7, located below. The numerical Value is: 0.245 m.



Figure 3.7: Moment of Arm of Quadcopter

3.3.9.3 Moments of Inertia

The moment of inertia represents a quantifiable resistance of a body to angular acceleration around a specified axis (Jardin, M.R. and Mueller, E.R. 2007). To ascertain the moment of inertia across the three principal axes of the quadcopter, an initial investigation utilizing the bifilar pendulum method, as detailed by (Habeck, J. and Seiler, P 2016), was employed.

Within the scope of this experiment, the quadcopter was affixed to a horizontal support by means of two strings (filaments), with assumptions made regarding the mass, tension, and elasticity of these strings as being negligible. These filaments were meticulously measured to ensure equal length, as well as the distance between them being accurately documented. Following the suspension of the quadcopter in a state of rest, it was then slightly perturbed from its equilibrium stance. The timing of this displacement

was precisely recorded with a stopwatch, noting the duration required for the quadcopter to execute ten complete oscillations. This process was repeated thrice to ascertain an average time for this oscillatory motion.

The quadcopter is suspended in the other two axes and the method described above is repeated for each subsequent setup. The results from the experiments are presented in 3.2, 3.3 and 3.4 below:

Table 3.2: Rotation About X-axis

Rotation About X-axis	
Run	Time Taken for 10 Oscillation
Experiment 1	13.09s
Experiment 2	12.93s
Experiment 3	13.06s

Table 3.3 : Rotation About Y-axis

Rotation About Y-axis	
Run	Time Taken for 10 Oscillation
Experiment 1	13.28s
Experiment 2	13.44s
Experiment 3	13.12s

Table 3.4: Rotation About Z-axis

Rotation About Z-axis	
Run	Time Taken for 10 Oscillation
Experiment 1	11.71s
Experiment 2	11.41s
Experiment 3	11.68s

After recording oscillation data, the process to ascertain the moment of inertia involves deriving the period of oscillation for each axis, which is then incorporated into Equation 3.1 (Habeck and Seiler., 2016). The oscillation period, defined as the duration required to complete a single cycle of oscillation (Tongue B., 2002), is computed by dividing the cumulative oscillation times, as recorded in Tables 3.2, 3.3, and 3.4, by 10.

This methodology ensures a precise calculation of the period, facilitating the accurate determination of the moment of inertia in accordance with the principles outlined in the referenced literature.

Moment of inertia:

$$I = \frac{mgT^2d^2}{16\pi^2L} \quad \text{Equation 3.112}$$

Where: m - mass of quadcopter (with battery connected in Kg), g - acceleration due to gravity, ms^{-2} , T - period for one oscillation (in seconds), d - distance between strings, in meters (m), l - length of string in meters (m),

3.3.9.4 Moment of inertia for x-axis, I_x

From table 3.1, the average period is calculated as follows:

$$\text{Average oscillation} = \frac{13.09+12.93+13.06}{3} = 13.027s \quad \text{Equation 3.113}$$

For this experimental setup the following values were measured: L = 0.928 m, d = 0.344 m, masse (m) = 1.250 kg.

The moment of inertia is calculated below:

$$I_x = \frac{1.587 \times 9.81 \times 13.027^2 \times 0.344^2}{16 \times \pi^2 \times 0.928} = 0.0213 \text{kgm}^2 \quad \text{Equation 3.114}$$

3.3.9.5 Moment of Inertia for y-axis, I_y

From table 3.2, the average period is calculated as follows,

$$\text{Average oscillation} = \frac{11.71+11.41+11.68}{3} = 11.6\text{s} \quad \text{Equation 3.115}$$

$$\text{Period for one oscillation} = \frac{13.28}{10} = 1.328\text{s} \quad \text{Equation 3.116}$$

The values of L, d and m are the same as those used in the x-axis pendulum setup. The moment of inertia is calculated below:

$$I_y = \frac{1.587 \times 9.81 \times 1.328^2 \times 0.344^2}{16 \times \pi^2 \times 0.928} = 0.02217 \text{kgm}^2 \quad \text{Equation 3.117}$$

3.3.9.6 Moment of Inertia for y-axis, I_{zz}

From table 3.3, the average period is calculated as follows,

$$\text{Average oscillation} = \frac{11.71+11.41+11.68}{3} = 11.6\text{s} \quad \text{Equation 3.118}$$

$$\text{Period for one oscillation} = \frac{11.605}{10} = 1.1\text{s} \quad \text{Equation 3.119}$$

For this experimental setup the following values were measured: L = 1.064 m, d = 0.4755 m, m = 1.587 kg.

The moment of inertia is calculated below:

$$I_z = \frac{1.587 \times 9.81 \times 1.16^2 \times 0.4755^2}{16 \times \pi^2 \times 1.064} = 0.0282 \text{kgm}^2 \quad \text{Equation 3.120}$$

3.3.9.7 Thrust Coefficient

In accordance with the work of (Luis and Ny., 2016), the procedure for determining the thrust generated by a propeller, represented as f_k , is elucidated. This

investigation forms a pivotal component of their broader inquiry into the dynamics of quadcopter flight and control mechanisms, offering a mathematical framework essential for the accurate calculation of propulsive forces. The formula, derived from their extensive analysis, lays the foundation for understanding the complex interactions between propeller mechanics and air resistance, thereby enabling the precise modulation of thrust for optimized flight performance. This calculation not only contributes to the theoretical underpinnings of quadcopter design but also serves as a critical tool for engineers and researchers seeking to refine the operational capabilities of these aerial vehicles through targeted adjustments to propeller configurations. The formula for calculating the thrust of a propeller is as follows: Thrust, f_k

$$f_k = C_t \rho n^2 D^4 \quad \text{Equation 3.121}$$

Where: C_t - non-dimensional thrust coefficient, n - propeller speed in (rps), D - rotor diameter in meter (m), ρ - air density, kg/m³.

For uniformity, the propeller speed is converted to rpm. Using, 1 rev=s = 60 rev=min
Therefore,

$$T = C_t \rho \left(\frac{\omega}{60}\right)^2 D^4 \quad \text{Equation 3.122}$$

Where: ω is the propeller speed in rpm

C_t is obtained from the APC propeller website. Using the diameter and air density values from the previous section, the thrust coefficient, b , is calculated as follows,

$$b = \frac{C_t \rho D^4}{3600} = \frac{0.14 \times 1.225 \times 0.304^4}{3600} = 4.0687 \times \frac{10^{-7} \text{N}}{\text{rpm}^2} \quad \text{Equation 3.123}$$

3.3.10 Simulink Plant Model Development

The next phase, following the establishment of foundational motion equations, involves developing a Simulink plant model to capture the quadrotor's dynamics. This begins with creating a MATLAB script to define crucial parameters such as the moments of inertia for the quadrotor and rotors, physical dimensions, mass, and thrust and drag coefficients. This script is essential for accurately simulating the quadrotor's physical behaviour.

To develop a comprehensive mathematical model of a quadcopter, the Starting point is by specifying the mass (m) and moment of inertia matrix (I) to describe the mass distribution and rotational inertia of the vehicle. Next, determine the physical dimensions, including the arm length (l), which is the distance from the centre of the quadcopter to each rotor. These initial parameters lay the groundwork for understanding the quadcopter's physical properties and behaviour.

Aerodynamic properties are equally essential. Define the thrust factor (f_k), which relates rotor speed to generated thrust, and the drag factor (m_k), quantifying rotational drag. Additionally, include the aerodynamic drag coefficient (k_{drag}) and translational drag coefficient (C_d) to model resistance during flight. Finally, incorporate air density (ρ), which affects the aerodynamic forces acting on the quadcopter. Precisely defining these parameters ensures an accurate and reliable quadcopter model.

Next, define the state variables and control inputs to model the quadcopter's dynamics. The state variables include the position vector ($[x, y, z]$), representing the quadcopter's location in three-dimensional space, and the velocity vector $[v_x, v_y, v_z]$,

indicating the rate of change of position. Orientation is captured by Euler angles $[\phi, \theta, \psi]$, and the angular velocity vector $[p, q, r]$ represents the rate of change of these angles.

Control inputs are the rotor thrusts $[T_1, T_2, T_3, T_4]$, generated by each of the quadcopter's four rotors. To enhance precision, include model uncertainties, wind turbulence, and external disturbances in the model. These factors can significantly impact the quadcopter's behavior and are essential for accurately simulating its response to real-world conditions.

Formulate the equations of motion, divided into translational and rotational dynamics. For translational dynamics, calculate the total thrust in the body frame and convert it to the inertial frame, accounting for the quadcopter's orientation in space. Include the effects of gravity and aerodynamic drag forces. Additionally, incorporate wind turbulence and external disturbances, which can be modelled as random forces acting on the quadcopter. Using these forces, calculate the resulting acceleration and update the quadcopter's velocity and position accordingly.

For rotational dynamics, calculate the torques generated by the rotors, causing the quadcopter to rotate. Utilize the moment of inertia matrix to determine the resulting angular acceleration, which depends on the mass distribution around the quadcopter's axes. Include model uncertainties that might affect the inertia or control inputs. Finally, update the angular velocity and orientation based on the calculated angular acceleration. These equations of motion are essential for simulating the quadcopter's realistic behaviour under various conditions and control inputs.

To export the quadcopter model to Simulink, create a new model by selecting "Blank Model." From the Simulink Library Browser, drag a "MATLAB Function" block

into the workspace. Open the block's editor by double-clicking it, then copy and paste the MATLAB function code for the quadcopter dynamics. Ensure the function signature matches the block's input and output requirements, handling inputs for control inputs (u), state variables (X), and parameters ($params$), and producing outputs for the state derivatives (dx).

3.4 Controller Development

In this section, the focus is on the development of two specific types of controllers: PID and MPC. These controllers are foundational to achieving precise and responsive behaviour in our system. The section begins with an overview of the theoretical principles underlying each controller, highlighting how PID provides a straightforward yet effective means of system regulation, while MPC offers advanced predictive capabilities and optimization in handling multivariable constraints. The integration of these controllers with the system's hardware is detailed, followed by a comprehensive iterative tuning process. This process is crucial for fine-tuning controller parameters to enhance system stability and performance under varying operational conditions. Through this meticulous development approach, the PID and MPC controllers are optimized to meet the stringent demands of our application.

3.4.1 Explanation of Physical Re-Positioning Errors

3.4.1.1 Definition of Re-Positioning Errors in UAV

In the operation of unmanned aerial vehicles (UAVs), particularly quadcopters, re-positioning errors refer to the discrepancies between the UAV's intended position and orientation and its actual state during flight. These errors are critical factors that affect the performance and reliability of UAV systems. They arise due to a combination of

environmental disturbances, sensor inaccuracies, and limitations within the control systems. (Mahony et al., 2012) and (Castillo et al., 2005). Mathematically, the re-positioning error can be defined as the difference between the desired state vector x_d and the actual state vector x_a :

$$e(t) = x_d(t) - x_a(t) \quad \text{Equation 3.124}$$

where: $e(t)$: is the error vector at time (t), $x_d(t)$: is the desired state vector at time (t).

$x_a(t)$: is the actual state vector time (t). The state vector $x(t)$ typically includes position (x, y, z) and orientation angles (ϕ, θ, ψ) corresponding to roll (ϕ), pitch (θ), and yaw (ψ) (Mahony et al., 2012).

Common re-positioning errors include:

- **Altitude Error (e_z)**: The vertical deviation between the desired altitude z_d and the actual altitude z_a :

$$e_z(t) = z_d(t) - z_a(t) \quad \text{Equation 3.125}$$

- **Yaw Error (e_ψ)**: The angular deviation around the vertical axis (yaw angle):

$$e_\psi(t) = \psi_d(t) - \psi_a(t) \quad \text{Equation 3.126}$$

- **Pitch Error (e_θ)**: The angular deviation around the lateral axis (pitch angle):

$$e_\theta(t) = \theta_d(t) - \theta_a(t) \quad \text{Equation 3.127}$$

- **Roll Error (e_ϕ)**: The angular deviation around the longitudinal axis (roll angle):

$$e_\phi(t) = \phi_d(t) - \phi_a(t) \quad \text{Equation 3.128}$$

These errors are significant because they directly influence the UAV's ability to perform precise manoeuvres and maintain stability during flight operations (Mahony et al., 2012) and (Castillo et al., 2005).

3.4.1.2 Impact of Re-Positioning Errors on UAV Performance

Re-positioning errors have a profound impact on the overall performance of UAVs, particularly in terms of stability, navigation, and control. The dynamics of a quadcopter can be modelled using nonlinear differential equations (Mahony et al., 2012) and (Castillo et al., 2005). Small errors can lead to significant deviations due to the system's sensitivity.

- **Stability:** Deviations in position and orientation can destabilize the UAV, making it susceptible to oscillations or even complete loss of control. For instance, an uncorrected roll error can introduce a coupling effect that amplifies over time. The system's stability can be analysed using Lyapunov stability criteria, where the error dynamics $\dot{\mathbf{e}}(t)$ should converge to zero.
- **Navigation:** Accurate positioning is crucial for mission-critical tasks. Errors in position can be represented as:

$$\mathbf{e}_{pos}(t) = \begin{bmatrix} e_x(t) \\ e_y(t) \\ e_z(t) \end{bmatrix} \quad \text{Equation 3.129}$$

where $e_x(t)$ and $e_y(t)$ are the horizontal position errors. These errors can cause the UAV to deviate from its intended path, affecting mission objectives (Shen et al., 2011).

Control: In challenging environments, re-positioning errors become more pronounced. The control input $\mathbf{u}(t)$ must be designed to minimize the error $\mathbf{e}(t)$. If the

control system cannot compensate effectively, the UAV may deviate significantly, compromising safety (Mahony et al., 2012) and (Castillo et al., 2005).

Minimizing these errors is essential for:

- **Maintaining Precise Control:** Ensuring that the UAV responds accurately to control inputs enhances operational efficiency and effectiveness.
- **Achieving Accurate Trajectory Tracking:** The goal is to minimize the trajectory tracking error $\mathbf{e}_{traj}(t)$, defined as:

$$\mathbf{e}_{traj}(t) = \mathbf{r}_d(t) - \mathbf{r}_a(t) \quad \text{Equation 3.130}$$

where $\mathbf{r}_d(t)$ is the desired trajectory and $\mathbf{r}_a(t)$ is the actual trajectory (Raffo et al., 2010).

- **Enhancing Safety and Reliability:** Reducing errors decreases the likelihood of accidents caused by unintended movements or collisions.

3.4.2 PID Controller

The controller ensures system stability and optimized performance, preventing oscillations, deviations, or failures. Among control strategies, PID control is versatile and effective, minimizing the error between a desired setpoint and the current state using proportional, integral, and derivative actions.

The proportional component addresses the current error magnitude, adjusting the control input proportionally, quickly reducing deviations but potentially causing overshoot. The integral component accumulates error over time, addressing persistent deviations and ensuring long-term adherence to the setpoint but requires careful calibration. The derivative component, considering the error's rate of change, stabilizes the system by predicting and mitigating potential overshoots.

Combining these actions allows PID controllers to adapt to various systems, from simple devices to complex technologies. Proper tuning of PID gains is crucial for precise control. In simulations, stable altitude requires a control mechanism. Implementing a PID controller involves setting a target state, generating an error signal, and minimizing it to zero. The components work together to generate the control input, conceptualized in Simulink as discrete gains or a continuous-time transfer function, encapsulating the dynamic relationship between input and output in the s-domain.

$$G_c(s) = \frac{K_d s^2 + K_p s + K_i}{s} \quad \text{Equation 3.132}$$

The introduction of feedback alters the transfer function dynamics of a controller/plant system. In the absence of feedback, the system's transfer function is delineated by the product of the transfer functions of both the controller and the plant. However, the determination of the closed-loop transfer function involves calculating the ratio of the system's output to its input.

Within this chapter, the application of a PID controller to the quadcopter system is explored, with the aim of attaining stability in both position and attitude in alignment with specified target values. The core nested control loops designated for managing position and attitude are graphically represented in Figure 3.8, illustrating the structured approach to achieving precise control over these key flight dynamics.

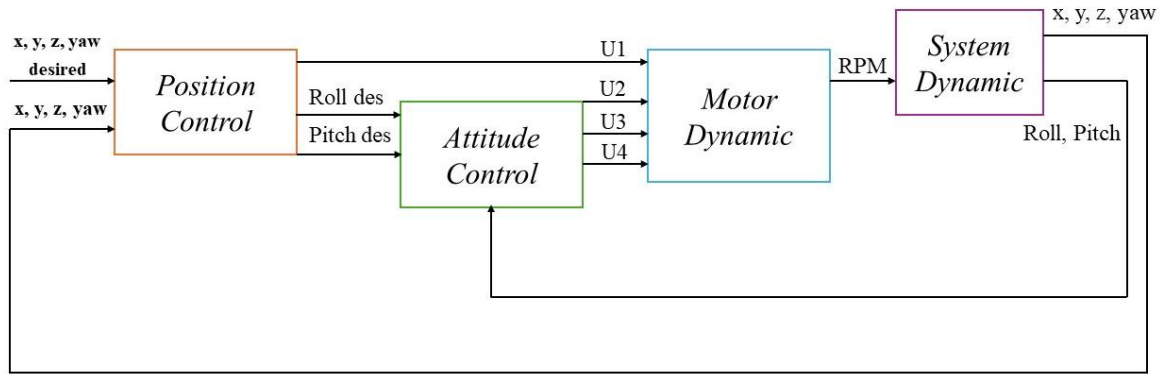


Figure 3.8: PID full block system

Given the nuanced challenges posed by the integral and derivative components, particularly in the context of dynamic systems with inherent uncertainties, there exists a need for strategies capable of addressing these issues. Such methodologies must carefully balance responsiveness and stability, ensuring that the controller adequately compensates for errors without introducing instability or excessive overshoot.

3.4.3 Effect of Nonlinearities and Noise

Implementing the derivative component in a PID control strategy faces two main challenges:

1. Sudden and large changes in the reference input can cause an excessive derivative response, resulting in a high control signal, highlighting the term's sensitivity to rapid system dynamics.
2. High-frequency noise from measurement sensors can be amplified by the derivative term. Deactivating the derivative function is not feasible due to its importance in improving transient behaviour.

To address these issues, a filter can be integrated into the derivative term. This filter mitigates the impact of noise and disturbances by attenuating high-frequency signals, ensuring effective transient response management without adverse effects. This adaptation maintains control precision and stability amidst potential disruptions.

The resulting PID controller when adding the filter to the derivative term is:

$$u(s) = K_p e(s) + K_I \frac{1}{s} e(s) + K_D \frac{Ns}{N+s} e(s) \quad \text{Equation 3.133}$$

When paired with a saturating actuator, integral action in a controller can cause integrator windup, leading to overshoot and extended settling times. This happens when a large control error saturates the actuator and disrupts feedback. Mitigation strategies include stopping the integral term update once saturation occurs or introducing feedback to reset the integrator.

3.4.4 PID Stability Controller

The main closed loop control structure depicted to the quadcopter system is shown in Figure 3.9. Six individual PID controllers are applied for each degree of freedom to achieve stability at the desired values.

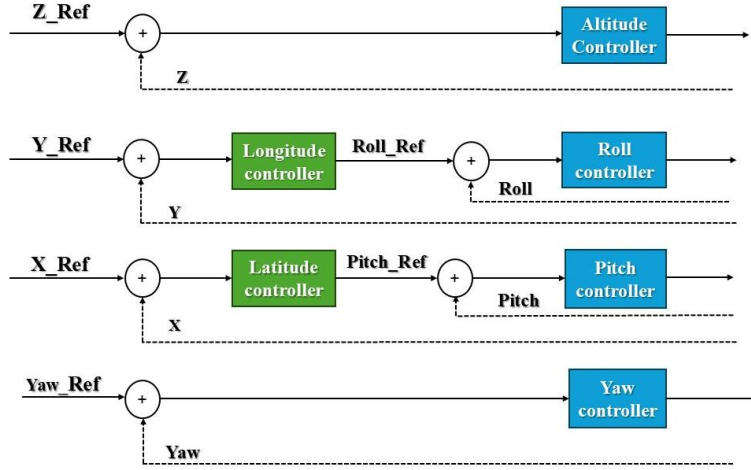


Figure 3.9: Closed loop control structure.

The roll (ϕ) and pitch (θ) angles manage the quadcopter's attitude, determining its orientation in space, while the yaw (ψ) angle controls its heading, directing its rotational position. Each of these angles is regulated using a PID controller. The PID controllers produce outputs U_2 , U_3 , and U_4 , respectively, for roll, pitch, and Yaw adjustments, ensuring stable and precise control.

3.4.4.1 Roll Control

The roll controller ensures the quadcopter's roll angle (ϕ) matches the desired roll angle (ϕ_{des}). This is achieved through a PID (Proportional-Integral-Derivative) controller, which continuously calculates adjustments. The PID controller equation for roll control combines the current roll error, the accumulated error over time, and the rate of change of the error to maintain the desired roll stability.

$$U_2 = K_{p\phi}(\phi_{des} - \phi) + K_{i\phi} \int (\phi_{des} - \phi) dt + K_{d\phi} \frac{d(\phi_{des} - \phi)}{dt} \quad \text{Equation 3.134}$$

3.4.4.2 Pitch Control

The Pitch controller manages the quadcopter's pitch angle (θ) to align it with the desired pitch angle (θ_{des}). Utilizing a PID controller, it calculates the necessary adjustments. The PID controller equation for pitch control integrates the current pitch error, accumulated error over time, and the rate of change of error to stabilize the pitch angle.

$$U3 = K_{p\theta}(\theta_{des} - \theta) + K_{i\theta} \int (\theta_{des} - \theta) dt + K_{d\theta} \frac{d(\theta_{des} - \theta)}{dt} \quad \text{Equation 3.135}$$

3.4.4.3 Yaw Control

The yaw controller aligns the quadcopter's yaw angle (ψ) with the desired yaw angle (ψ_{des}). Using a PID controller, it continuously computes the necessary corrections. The PID controller equation for yaw control incorporates the present yaw error, the cumulative error over time, and the rate at which the error changes to ensure precise yaw stability and control.

$$U4 = K_{p\psi}(\psi_{des} - \psi) + K_{i\psi} \int (\psi_{des} - \psi) dt + K_{d\psi} \frac{d(\psi_{des} - \psi)}{dt} \quad \text{Equation 3.136}$$

3.4.4.4 Altitude Control

The altitude controller manages the quadcopter's altitude (z) to align with the desired altitude (z_{des}). Utilizing a PID controller, it continuously calculates necessary adjustments. The PID controller equation for altitude control integrates the current altitude error, the accumulated error over time, and the rate of change of the error to maintain the desired altitude precisely.

$$U1 = K_{pz}(z_{des} - z) + K_{iz} \int (z_{des} - z) dt + K_{dz} \frac{d(z_{des} - z)}{dt} \quad \text{Equation 3.137}$$

3.4.4.5 Longitude Control

The longitude controller regulates the quadcopter's longitudinal position (x) to match the desired longitude (x_{des}) by adjusting the pitch angle (θ). Employing a PID controller, it continuously computes the necessary corrections. The PID controller equation for longitude control combines the current longitude error, the accumulated error over time, and the rate of error change to maintain the desired position.

$$\theta_{des} = K_{px}(x_{des} - x) + K_{ix}\int(x_{des} - x)dt + K_{dx}\frac{d(x_{des}-x)}{dt} \quad \text{Equation 3.138}$$

3.4.4.6 Latitude Control

The latitude controller manages the quadcopter's lateral position (y) to align with the desired latitude (y_{des}) by adjusting the roll angle (ϕ). Using a PID controller, it continuously calculates the necessary adjustments. The PID controller equation for latitude control integrates the current latitude error, the accumulated error over time, and the rate of change of the error to maintain the desired position accurately.

$$\phi_{des} = K_{py}(y_{des} - y) + K_{iy}\int(y_{des} - y)dt + K_{dy}\frac{d(y_{des}-y)}{dt} \quad \text{Equation 3.139}$$

These equations provide a comprehensive control strategy for maintaining the quadcopter's stability and desired position in three-dimensional space.

3.4.5 PID CONTROLLER TUNING METHOD

With the Simulink model developed for the UAV Quadcopter and the PID controllers in the previous section, a systematic approach would need to be adopted to tune the PID controllers and obtain optimal gain values for the UAV to attain stabilized flight. The Ziegler-Nichols rule elaborated in (Ogata., 2010) for tuning PID controllers was selected as a suitable method for this thesis to determine the gains of the Quadcopter PID controllers. There are two methods for applying the Ziegler-Nichols rule:

3.4.5.1 Ziegler-Nichols First Method

The initial Ziegler-Nichols tuning approach involves experimentally determining the system's response to a unit step input, assuming the system lacks an integrator or does not exhibit complex conjugate poles. This method works when the system's response curve is S-shaped. If the system has an integrator or dominant complex conjugate poles, this method is unsuitable, necessitating the second method for PID controller tuning.

3.4.5.2 Ziegler-Nichols Second Method

The second Ziegler-Nichols method involves progressively increasing the proportional gain of each control loop until the system output begins to oscillate. The gain at this point is the critical gain (K_{cr}), and the period of oscillation is P_{cr} . These values are used to determine the PID controller's gain parameters based on a predefined formula. The PID controller's transfer function, tuned using this method, is detailed in Equation 3.140.

Table 3.5: PID controller gains using Ziegler-Nichol methods

Type of Controller	KP	TI	TD
Proportional only	$0.5K_{cr}$	∞	0
Proportional & Integral only	$0.45K_{cr}$	$112P_{cr}$	0
Proportional, Integral & Derivative	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

$$\begin{aligned}
 G_C(S) &= K_P \left(1 + \frac{1}{T_I \cdot s} + T_D \cdot s \right) \\
 &= 0.6 \cdot K_{cr} \left(1 + \frac{1}{0.5 \cdot P_{cr} \cdot s} + 0.125 \cdot P_{cr} \cdot s \right). \quad \text{Equation 3.140}
 \end{aligned}$$

Using the second method for the Ziegler-Nichols rule, the gains for the PID controller were determined individually with the Simulink model developed, as illustrated in the figure below:

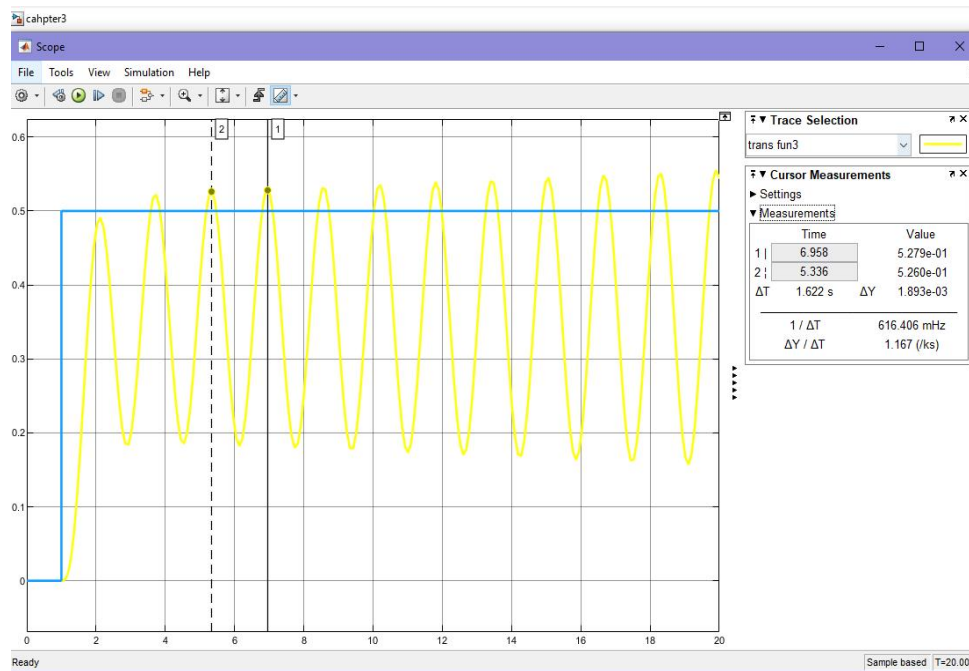


Figure 3.10: The sustained oscillations for PID controller gain

3.5 Model Predictive Control

Model predictive control is a premier control engineering method used to manage complex systems in industries such as process control, automotive, aerospace, and energy management. It leverages a predictive model of the system, incorporating both physical dynamics and constraints.

MPC predicts the future state of a system over a defined horizon using current and past states, external inputs, and a mathematical model. This prediction is based on rigorous computational algorithms that solve an optimization problem at each control step. The goal is to minimize a cost function reflecting deviations from desired setpoints, operational costs, or energy consumption.

A key advantage of MPC is its consideration of system constraints, ensuring that control actions conform to physical limitations, safety requirements, and operational boundaries. This feature prevents overextension and ensures safety and reliability.

MPC is adaptable for managing multi-input and multi-output (MIMO) systems with complex dynamics and constraints, handling nonlinearities and uncertainties efficiently. It adjusts control actions at each time step, responding dynamically to changes in the system or environment, maintaining optimal performance and adherence to constraints.

MPC employs a system model to determine optimal control actions via optimization. For a Single-Input Single-Output (SISO) system, the operational steps include:

- At each time step k , the controller forecasts future outputs over a horizon H_p using a model, past inputs, and future control actions $u(k+i-1)$, where i ranges from 1 to H_c-1 .
- The horizon H_p defines the extent of future predictions based on historical performance and projected adjustments.
- The control horizon H_c , shorter than H_p , defines the period for varying control actions; beyond it, actions are constant. Optimal moves within H_c minimize a cost function, aligning predicted outputs with targets.

- The initial control move $u(k)$ is implemented immediately, and the process recalibrates, advancing the prediction and control horizons one step forward, repeating for each subsequent step $k+1$.

This dynamic adjustment, or receding horizon control, highlights MPC's continuous recalibration for optimal control decisions.

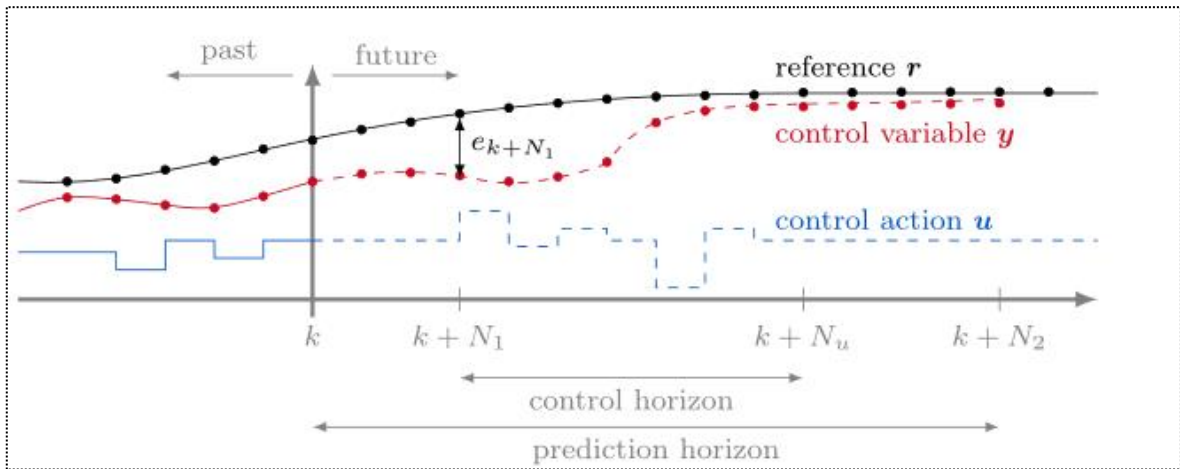


Figure 3.11: MPC foundation principle

(Schwenzer, M., Ay, M., Bergs, T., & Abel, D. (2021). Review on model predictive control: An engineering perspective. *The International Journal of Advanced Manufacturing Technology*, 117(5), 1327-1349.)

Figure 3.11 illustrates the MPC process from sample k to $k+N$, where N is the prediction horizon, determined by the system's dynamics. MPC aims to maintain equilibrium around a reference point within this horizon. The initial diagram shows multiple trajectories for stabilization. The algorithm evaluates errors at each sample, considering the sum of squared errors (e) and the sum of squared changes in input variables (σu), as shown in equation (3.141). Weighting factors (w) balance precision and control effort, guiding MPC to minimize deviation and manage input adjustments efficiently.

$$J = \sum_{i=1}^N w_e e_{k+i}^2 + \sum_{i=0}^{N-1} w_{\Delta u} \Delta u_{k+i}^2 \quad \text{Equation 3.141}$$

MPC uses an optimization solver to find input adjustments that minimize the cost function. However, due to disturbances and uncertainties, the actual system trajectory can deviate from predictions. In the third illustration, the trajectory slightly exceeds the expected path. Therefore, only the initial input adjustment is applied, and the rest are discarded. The prediction horizon then advances, extending from $k+1$ to $k+N+1$. A new forecast is generated from the latest system state, either directly measured or estimated using a filter like the Kalman filter. For regulation, MPC aims to align state variables with zero, typically using a linear predictive model.

$$x_{k+1} = Ax_k + Bu_k \quad \text{Equation 3.142}$$

$$y_k = Cx_k + Du_k \quad \text{Equation 3.143}$$

In this formulation, x denotes the state variable, u represents the input variable, and y signifies the output vector, which is linked to the state vector via the C matrix. It is commonly postulated that the D matrix is null in most systems, a simplification that holds true for numerous applications. This model undergoes discretization, a process meticulously executed in this thesis employing both the forward Euler method and the zero-order hold (ZOH) technique. Within the context of regulatory objectives, the cost function for Model Predictive Control (MPC) is structured as follows:

$$J = \min \frac{1}{2} x_{t+N}^T S x_{t+N} + \frac{1}{2} \sum_{k=0}^{N-1} (x_{t+k}^T Q x_{t+k} + u_{t+k}^T R u_{t+k}) \quad \text{Equation 3.144}$$

In regulatory control, the cost function focuses on absolute input values rather than input variations. This assumes that once all state variables reach zero (system stability), the input values can also be reduced to zero. For example, when landing a drone,

the inputs can be nullified once it has landed. A distinct weighting matrix, S , is applied to the final element within the prediction horizon. The cost function is scaled by 0.5 for computational ease, simplifying the differentiation process. Here, t represents the current moment, and $t+k$ signifies a future point, k samples ahead. However, this thesis focuses on reference tracking rather than regulation, modifying the cost function to replace the state variable with the error variable.

$$J = \min \frac{1}{2} e_{t+N}^T S e_{t+N} + \frac{1}{2} \sum_{k=0}^{N-1} (e_{t+k}^T Q e_{t+k} + u_{t+k}^T R u_{t+k}) \quad \text{Equation 3.145}$$

The error is defined in equation (3.125) as follows:

$$e_k = r_k - y_k = r_k - Cx_k \quad \text{Equation 3.146}$$

Within the context of a tracking scenario, it is imperative for the inputs to maintain nonzero values to ensure that the tracking error remains nullified, thereby enabling the Unmanned Aerial Vehicle (UAV) to adhere to the reference path and accurately trace the intended trajectory. However, upon the UAV's alignment with the reference value, the variation in inputs may revert to zero, provided the system's stability is affirmed. Consequently, in addressing a tracking dilemma, the focus shifts to the adjustments of the input, denoted as Δu_k , which are characterized as follows:

$$\Delta u_k = u_k - u_{k-1} \quad \text{Equation 3.147}$$

The equation (3.142) can be rewritten as:

$$u_k = u_{k-1} + \Delta u_k \quad \text{Equation 3.148}$$

That means that the state space equation can be rewritten in the following way as:

$$x_{k+1} = Ax_k + B(u_{k-1} + \Delta u_k) \quad \text{Equation 3.149}$$

$$y_k = Cx_k \quad \text{Equation 3.150}$$

The next step would be to augment the system where the absolute input one sample in the past u_{k-1} becomes a state as well. The augmented formulation can be written as:

$$\begin{bmatrix} x_{k+1} \\ u_k \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix} + \begin{bmatrix} B \\ I \end{bmatrix} \Delta u = \tilde{A} \tilde{x}_k + \tilde{B} \Delta u \quad \text{Equation 3.151}$$

$$y_k = [C \quad 0] \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix} = \tilde{C} \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix} \quad \text{Equation 3.152}$$

Upon substituting the error term within the cost function and replacing the C matrix and the state variables with their augmented counterparts, the cost function transitions to incorporating Δu in lieu of the absolute values of (u).

$$J = \frac{1}{2} \begin{bmatrix} R - CX \\ \Delta U \end{bmatrix}^T \begin{bmatrix} Q & 0 \\ 0 & \mathcal{R} \end{bmatrix} \begin{bmatrix} R - CX \\ \Delta U \end{bmatrix} \quad \text{Equation 3.153}$$

Upon detailed formulation, certain elements within the expression will emerge as constants. From an optimization perspective, these constant factors bear no influence on the outcome derived by the optimization algorithm. Consequently, for the purpose of streamlining the equation.

$$J = \frac{1}{2} \begin{bmatrix} R \\ \tilde{X} \\ \Delta U \end{bmatrix}^T \begin{bmatrix} Q & -QC & 0 \\ -C^T Q & C^T QC & 0 \\ 0 & 0 & \mathcal{R} \end{bmatrix} \begin{bmatrix} R \\ \tilde{X} \\ \Delta U \end{bmatrix} \quad \text{Equation 3.154}$$

The applicability of this formulation depends on the weight matrices being diagonal, ensuring they are equal to their transposed values and enabling this specific mathematical representation. Traditionally, the prediction horizon has been discussed using summation notation. However, another approach consolidates future reference values, state

variables, and input variations into a single vector, with each component corresponding to a discrete sampling instance.

The current state vector, x_t , is separate and not included in the future state projections. In the comprehensive horizon vector, reference and state variables range from $t+1$ to $t+N$, while input adjustments span from t to $t+N-1$. This reflects the principle that an input at any time affects the state and output at the next time interval. Equations 3.155 and 3.156 show the cost function's structure under this framework, with weight matrices combined into large diagonal matrices spanning the entire horizon.

$$r = \begin{bmatrix} r_{t+1} \\ r_{t+2} \\ \vdots \\ r_{t+N} \end{bmatrix} \quad \tilde{x} = \begin{bmatrix} \tilde{x}_{t+1} \\ \tilde{x}_{t+2} \\ \vdots \\ \tilde{x}_{t+N} \end{bmatrix} \quad \Delta u = \begin{bmatrix} \Delta u_t \\ \Delta u_{t+1} \\ \vdots \\ \Delta u_{t+N-1} \end{bmatrix} \quad \tilde{x}_t = \text{present} \quad \text{Equation 3.155}$$

$$J' = \min \frac{1}{2} \begin{bmatrix} \tilde{x} \\ \Delta u \end{bmatrix}^T \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \Delta u \end{bmatrix} - r^T J \tilde{x} \quad \text{Equation 3.156}$$

The goal is to reformulate the cost function such that it exclusively encompasses variations in input values and current state variables. To achieve the exclusion of future state variables, a series of mathematical operations are applied as outlined as follow:

$$\begin{aligned} \tilde{x}_1 &= \tilde{A}\tilde{x}_0 + \tilde{B}\Delta u_0 \\ \tilde{x}_2 &= \tilde{A}\tilde{x}_1 + \tilde{B}\Delta u_1 = \\ &= \tilde{A}^2\tilde{x}_0 + \tilde{A}\tilde{B}\Delta u_0 + \tilde{B}\Delta u_1 \\ &\quad \dots \\ \tilde{x}_k &= \tilde{A}^k\tilde{x}_0 + [\tilde{A}^{k-1}\tilde{B} \quad \tilde{A}^{k-2}\tilde{B} \quad \dots \quad \tilde{B}] \begin{bmatrix} \Delta u_0 \\ \Delta u_1 \\ \vdots \\ \Delta u_{N-1} \end{bmatrix} \end{aligned} \quad \text{Equation 3.157}$$

In this formulation, the state variables are replaced by preceding state values, which are exclusively derived from the A and B matrices, alongside the present state value

and both current and forthcoming input variations. This allows for the entire state space system over the prediction horizon to be succinctly encapsulated as indicated in below:

$$\tilde{x} = \begin{bmatrix} \tilde{B} & 0 & 0 & 0 & 0 \\ \tilde{A}\tilde{B} & \tilde{B} & 0 & 0 & 0 \\ \tilde{A}^2\tilde{B} & \tilde{A}\tilde{B} & \tilde{B} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \tilde{A}^{N-1}\tilde{B} & 0 & 0 & 0 & \tilde{B} \end{bmatrix} \Delta u + \begin{bmatrix} \tilde{A} \\ \tilde{A}^2 \\ \cdot \\ \cdot \\ \tilde{A}^N \end{bmatrix} \tilde{x}_t = \tilde{C}\Delta u + \hat{\tilde{A}}\tilde{x}_t \quad \text{Equation 3.158}$$

The future state values in the cost function are then replaced by this compact form. As it can be seen in equation 1.159, the future state variables have disappeared from the cost function:

$$J' = \frac{1}{2} \begin{bmatrix} \Delta u \\ \tilde{x}_t \end{bmatrix}^T \begin{bmatrix} \tilde{C}^T \tilde{Q} \tilde{C} + \bar{R} & \tilde{C}^T \tilde{Q} \hat{\tilde{A}} \\ \hat{\tilde{A}}^T \tilde{Q} \tilde{C} & \hat{\tilde{A}}^T \tilde{Q} \hat{\tilde{A}} \end{bmatrix} \begin{bmatrix} \Delta u \\ \tilde{x}_t \end{bmatrix} - \begin{bmatrix} r^T \tilde{T} \tilde{C} \\ r^T \tilde{T} \hat{\tilde{A}} \end{bmatrix}^T \begin{bmatrix} \Delta u \\ \tilde{x}_t \end{bmatrix} \quad \text{Equation 3.159}$$

When it is written out, the constant terms are again ignored due to the fact they do not influence the optimizer results. The results of the derivation can be seen in the following:

$$J'' = \frac{1}{2} \Delta u^T H \Delta u + f^T \Delta u \quad \text{Equation 3.160}$$

In this section, it is demonstrated how the entire cost function is reformulated into a structure amenable to processing by the MATLAB solver. At its foundational level, the solver requires the matrix H and the vector f^T as inputs. Should H be positive definite, the solver can identify a sequence of input adjustments (u) that optimally reduces the cost function. Subsequently, the initial input adjustment in the sequence is implemented to direct the UAV's movement. Remaining input adjustments are disregarded, and the procedure is iteratively conducted for each subsequent sampling interval.

Model predictive control offers a comprehensive and flexible approach to system management, enabling precise control over complex dynamics while balancing performance goals against operational constraints. Its predictive capability, combined with constraint management, makes MPC an invaluable tool in the pursuit of advanced system optimization and control.

3.6 Implementing MPC Controller in Simulink

This section outlines the methodology for designing and implementing Model Predictive Control (MPC) using MATLAB/Simulink. The MPC Toolbox provides tools for creating, analysing, and executing advanced control algorithms with a user-friendly interface and customizable command-line syntax. MPC uses a linear approximation of the system to forecast behaviour and adjust actuators within operational constraints. It effectively manages resources by accounting for constraints, supporting transfer functions, state-space representations, and delays. The System Identification Toolbox helps construct models from empirical data. The MPC Toolbox also includes a Simulink® block for linearizing nonlinear models and creating an MPC Controller block, enabling real-world application through the Real Time Workshop.

3.6.1 MPC Toolbox interface overview

To activate the Model Predictive Control (MPC) toolbox in MATLAB, users should enter the command **mpcDesigner** in the MATLAB command window. This command initiates the toolbox, providing a user interface for predictive control analysis and implementation. The interface assists in configuring and applying MPC strategies, offering tools for detailed system control simulations and adjustments. This enables researchers and

engineers to design, test, and refine MPC systems in a controlled and visually interactive environment, as depicted in figure 3.12.

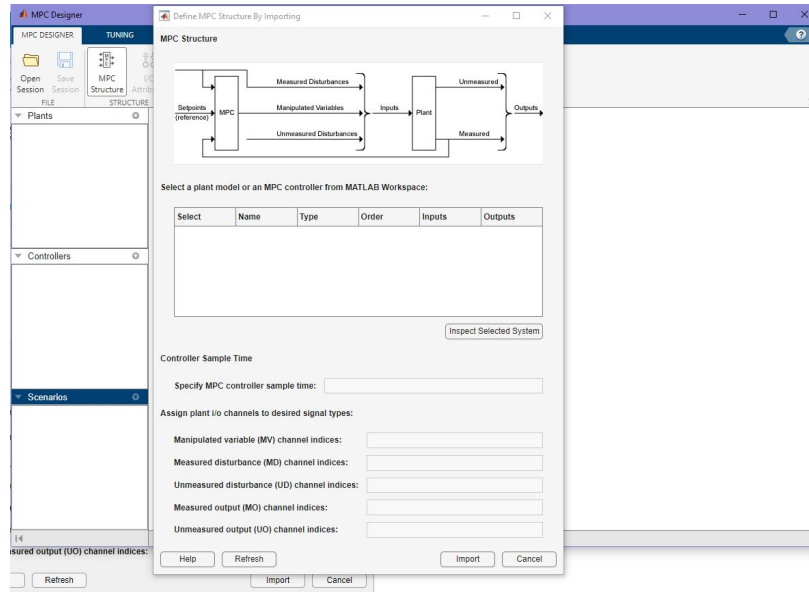


Figure 3.12: MPC designer interface

In the MPC Designer interface, navigate to the 'Structure' area and select the 'MPC Structure' option to open the 'Define MPC Structure by Importing' dialogue. Choose a plant model, like the Continuous Stirred Tank Reactor (CSTR), known for its stable, continuous-time Linear Time-Invariant (LTI) characteristics. While the MPC Designer suggests a default sampling interval, set it manually to 0.5 seconds for illustration. Configure inputs and outputs: coolant temperature as a manipulated variable, feed concentration as an unmeasured disturbance, reactor temperature as a measured output, and reactant concentration as an unmeasured output.

3.6.2 Define Input and Output Channel Attributes

In the MPC Designer interface, users can access 'I/O Attributes' to open the 'Input and Output Channel Specifications' dialogue box as showed in figure 3.13, is crucial for configuring channel properties. Each input and output channel should be given a descriptive name in the 'Name' column, enhancing model clarity and manageability. Users can also specify measurement units in the 'Unit' column. For state-space models, the 'Nominal Value' is typically set to zero, ensuring accurate interpretation of deviations. The 'Scale Factor' is maintained at 1 for each channel, affirming unity scale for signals. These settings ensure correct scaling and interpretation, leading to effective control and system monitoring.

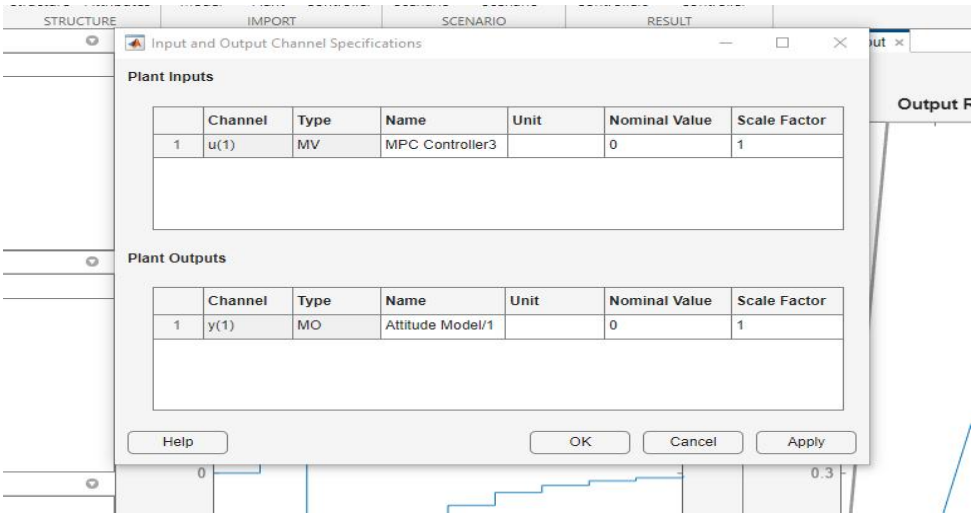


Figure 3.13: MPC input/output channel

3.6.3 Configure Controller Horizons

In the 'Tuning' segment of the MPC Designer, within the 'Horizons' category, setting the Prediction and Control horizons is crucial and should align with the dynamic model's complexity. The Prediction Horizon (N) determines how far ahead the MPC controller anticipates system behaviour, allowing for more informed decision-making at

the cost of increased computational demand. The Control Horizon (M) specifies the period over which the controller can adjust control inputs, with actions fixed beyond this period. This approach manages the computational load while effectively responding to changes in the system's behaviour.

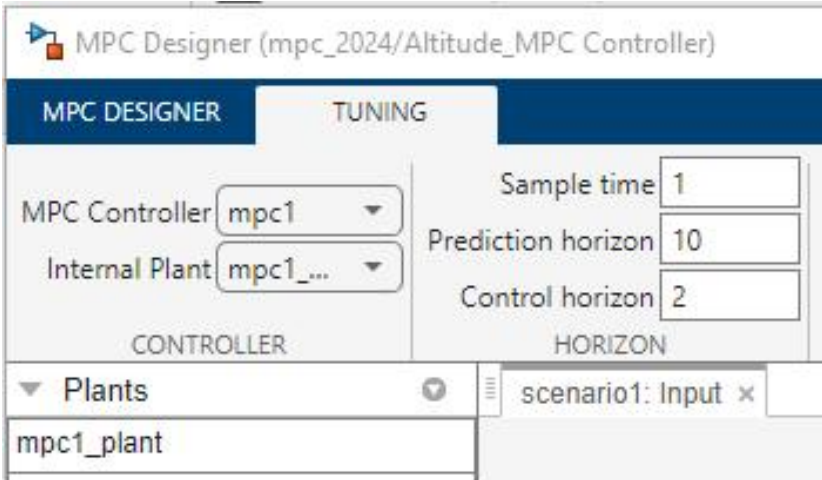


Figure 3.14: MPC Controller Horizons Configuration

Figure 3.14 provides a graphical representation of how the Prediction and Control horizons interact within the MPC framework. It illustrates the relationship between these horizons and their impact on system prediction and control adjustments, serving as a visual aid to better understand the configuration process.

3.6.4 Define Input Constraints

In the 'Design' area of the MPC Designer interface, navigate to and select the 'Constraints' option to access configuration settings. This opens the 'Constraints' dialogue box, where input and output constraints are set to ensure safe and effective system operation. In the 'Input and Output Constraints' segment, define operational boundaries for coolant temperature by entering minimum and maximum values in the 'Min' and 'Max'

columns. Additionally, specify the permissible rate of change for the coolant temperature in the 'Rate_Min' and 'Rate_Max' columns to prevent abrupt changes that could cause dynamic instabilities or process limit violations.

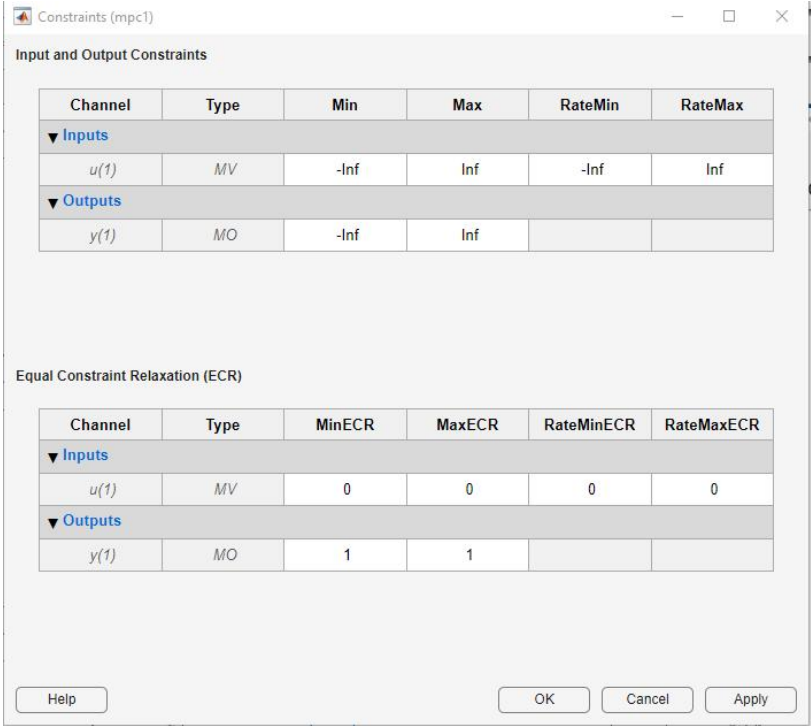


Figure 3.15: MPC input/output definition

Figure 3.15 visually demonstrates the input and output constraint configuration process within the MPC Designer interface. It highlights the allocation of constraint parameters, including operational limits and rate of change boundaries, providing a clear reference for ensuring robust and safe control system design.

3.6.5 Specify Controller Tuning Weights

Within the 'Tuning' tab of the MPC Designer under the 'Design' category, select 'Weights' to adjust the controller's optimization behaviours. Set the input weight for the rate of change of the manipulated variable (MV) to 0.3, imposing a higher penalty on significant MV variations to enhance system stability. Maintain preset values in the 'Output

Weights' table, assigning zero weight to unmeasured outputs to simplify the control strategy. This prioritizes tracking the reactor temperature setpoint, potentially causing steady-state discrepancies in concentration. Adjust weighting parameters—Q for states, R for inputs, and S for output rates—to balance state tracking and control effort, reducing aggressive control actions and prioritizing specific states based on process requirements.

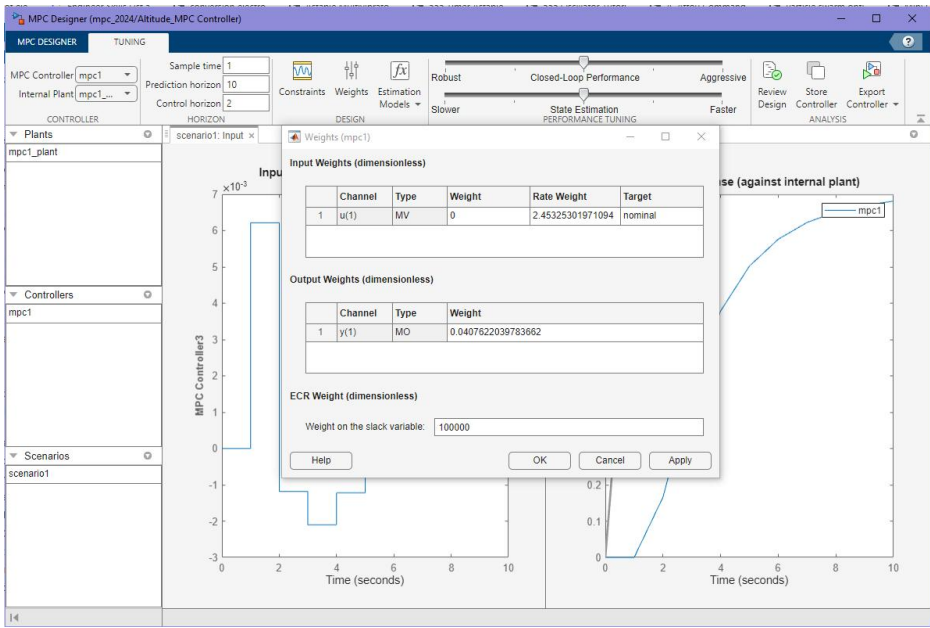


Figure 3.16: MPC output weights definition

Figure 3.16 illustrates the configuration of output weights within the MPC Designer interface. It highlights the allocation of weights to prioritize specific outputs and control efforts, serving as a visual guide for balancing system performance and control energy. This figure provides clarity on the practical adjustments required for optimizing controller performance.

3.7 Experiment and Validation

In this section, we detail the setup of our experimental framework, focusing on the integration and configuration of the Pixhawk controller. This setup is designed to

thoroughly evaluate the controller's performance in simulated real-world conditions. We outline the installation of the Pixhawk hardware, its connection to necessary sensors and actuators, and the calibration processes required for optimal operation. Additionally, the section describes the specific environmental parameters and scenarios under which the Pixhawk will be tested, ensuring a comprehensive assessment. This structured approach allows us to systematically monitor and analyse the Pixhawk's responsiveness, stability, and overall effectiveness, providing crucial data for further refinement and validation of the control system.

3.7.1 Experiment Setup

This chapter offers a detailed exposition of the entire experimental setup architecture, illustrated comprehensively in Figure 3.17. The narrative adopts an overarching perspective, focusing primarily on those components of the architecture that are most pertinent to the discussions within this thesis.

The experiment and validation process begins with system setup and configuration, where each component of the system is prepared for experimentation. First, the Ground Control Station (GCS) software, such as QGroundControl, is installed on a computer. This software acts as the mission planning and monitoring tool. The communication protocol, typically MAVLink, is configured to establish a seamless connection with the flight controller. Mission parameters, including waypoints, altitude, and speed, are defined within the GCS to align with the research objectives. Simultaneously, the flight controller, such as a Pixhawk, is integrated into the system. This involves flashing the appropriate firmware, such as PX4 or ArduPilot, and performing

essential calibrations, including those for the GPS, IMU, and barometer. Basic functionality is tested to ensure that motors and actuators respond correctly to commands.

The experimental execution phase begins with pre-flight simulation using the GCS. Simulations allow for validating configurations, testing control algorithms, and simulating environmental conditions. Once successful, the system is deployed in a controlled outdoor environment for field testing. During this stage, manual control via the RC transmitter is tested to confirm that a manual override is functional in case of emergencies. Autonomous features, such as obstacle avoidance and safe landing protocols, are validated to ensure reliability during missions. Additionally, telemetry data from the flight controller and sensor outputs are logged throughout the field tests for analysis.

The validation process follows, where the performance of the system is analysed in detail. Flight logs are compared against the predefined mission parameters to assess accuracy in waypoint navigation, altitude stability, and overall system behaviour. Sensor data is analysed to identify any anomalies or inaccuracies during operation. Middleware protocols, such as MAVLink, are tested for reliability, ensuring low-latency and consistent communication between modules. Resilience tests are conducted under conditions with variable bandwidth or signal interference to evaluate the robustness of the communication system. Applications running on the companion computer, such as collision avoidance algorithms, are subjected to rigorous testing to validate their functionality and performance under high computational loads.

The process then transitions into iterative refinement, where issues identified during validation are addressed. This may involve troubleshooting hardware malfunctions, resolving software bugs, or refining configurations and algorithms. After adjustments,

simulations and field tests are repeated to validate improvements and ensure the system meets the experimental goals. This iterative process continues until performance metrics align with the desired outcomes.

Finally, the validation process concludes with end-to-end testing. An integrated test of the entire system, including the GCS, flight controller, companion computer, and sensors, is conducted under real-world conditions. This ensures that all components function cohesively and reliably. Comprehensive performance reports are prepared, comparing expected outcomes with actual performance metrics. These reports include detailed documentation of configurations, adjustments, and test results, ensuring reproducibility of the experiment. The findings are then prepared for publication, with visuals such as architecture diagrams and performance graphs included to effectively communicate the results. GPS includes the experim QGS Software ation process, providing a robust foundation for the research outcomes.



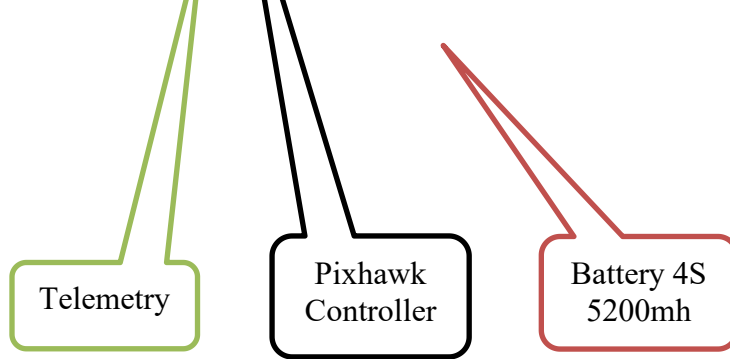


Figure 3.17: PX4 Hardware components

The hardware component list used during the experiment flight tests is as follow:

- **GPS:** The M8N GPS as it can be seen in Figure 3.18 uses multi-constellation GNSS powered by u-blox M8N series, a concurrent GNSS receiver which can receive and track multiple GNSS systems. Owing to the multi-band RF front-end architecture all four major GNSS constellations, GPS, Galileo, GLONASS and BeiDou can be received concurrently. (Detailed Technical specification and calibration can be found at Appendix B)



Figure 3.18: GPS M8N

- **Pixhawk 6C:** The Pixhawk® 6C is the latest update to the successful family of Pixhawk® flight controllers, based on the Pixhawk® FMUv6C Open Standard and Connector Standard. It comes with PX4 Autopilot® pre-installed. Inside the Pixhawk® 6C, you can find a STMicroelectronics® based STM32H743, paired with sensor technology from Bosch® & InvenSense®, giving you flexibility and reliability for controlling any autonomous vehicle, suitable for both academic and commercial applications.



Figure 3.19: Pixhawk 6C Controller

- **PM02 V3 Power Module:** This PM02 Power Module provides a simple way to supply regulated 5.2V to flight controller from the battery while providing current consumption and battery voltage measurements via analog signal through a 6 Pin JST-GH cable. It comes with pre-soldered wires, capacitor, and XT60 connector on both ends, along with shrink tubing on the PCBA for physical protection. (Detailed Technical specification and calibration can be found at Appendix B)



Figure 3.20: PM02 V3 Power Module

- **RadioMAster R81 Receiver:** This R81 receiver ships with a Dupont connector pre-soldered. To use with Pixhawk 6X, 6C & Mini, Pix32 V6, please connect to the Dupont connector by using the cable provide in the Flight Controller's cable set.

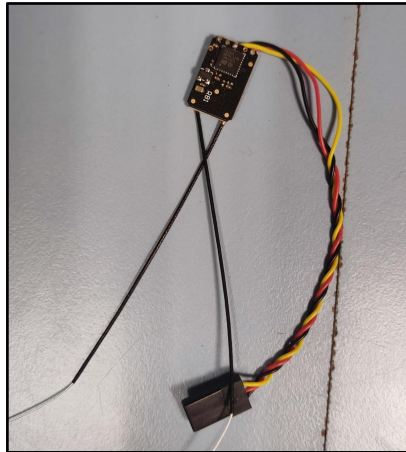


Figure 3.21: RadioMaster R81 Receiver

- **RadioMaster TX16S:** A Radio Control (RC) system can be used to *manually* control your vehicle from a handheld RC controller. This topic provides an overview of how RC works, how to choose an appropriate radio system for your vehicle, and how to connect it to your flight controller. (Detailed Technical specification and calibration can be found at Appendix B)



Figure 3.22: RadioMaster TX16S

- **Telemetry Radio V3:** This Holybro SiK Telemetry Radio is a small, light and inexpensive open-source radio platform that typically allows ranges of better than

300m “out of the box” (the range can be extended to several kilometres with the use of a patch antenna on the ground). This radio is plug-n-play all Pixhawk Standard and other Flight Controller, providing the easiest ways to setup a telemetry connection between your Autopilot and a ground station. It uses open-source firmware which has been specially designed to work well with MAVLink packets and to be integrated with the Mission Planner, Ardupilot, QGroundControl, PX4 Autopilot. The radios can be either 915Mhz or 433Mhz and you should purchase the model which is appropriate for your country/region. (Detailed Technical specification and calibration can be found at Appendix B)



Figure 3.23: Telemetry Radio V3

- **Holybro 2216 KV920 Motor:** is a high-performance brushless motor designed for multirotor drones and UAV applications. It offers a balance of power and efficiency, making it suitable for medium-sized drones that require stable and reliable propulsion. With a KV rating of 920, it delivers optimal RPM per volt, ensuring smooth performance and precise control. Built with durable materials, the motor features a robust design for long-lasting operation and excellent heat dissipation. It is compatible with a wide range of electronic speed controllers (ESCs) and supports various propeller sizes, making it versatile for diverse configurations.

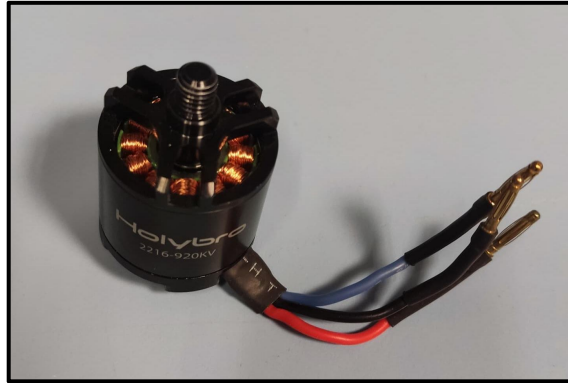


Figure 3.24: Holybro 2216 KV920 Motor

- **ESCs - BLHeli S ESC 20A:** A BLHeli S 20A ESC designed for use with Holybro's 2216-880KV Motor / X500 Quadcopter Kit (V1 and V2). The ESC is slim so can fit within 16MM booms. It comes with long wires and pre-soldered connectors for a quick and easy installation. (Detailed Technical specification and calibration can be found at Appendix B)

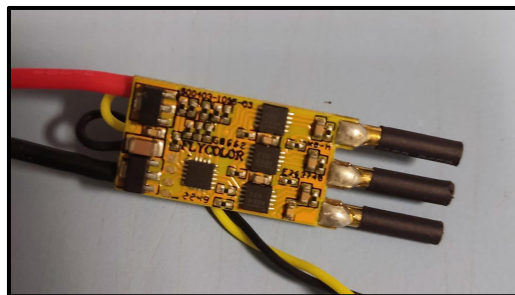


Figure 3.25: ESCs - BLHeli S ESC 20A

- **Battery S4 5200 mh:** This Li-Po battery is 100% brand new and capable of delivering maximum continuous discharge rates of up to 30C, with a burst rate of 60C, making it one of the most powerful battery packs in its class. It offers an excellent combination of lightweight design, high power, and exceptional performance. Available in a wide range of capacities and voltages, this battery is suitable for various applications, including indoor slow flyers, large-scale sport and

aerobatic airplane models, and micro to mid-sized helicopter models. (Detailed Technical specification and calibration can be found at Appendix B)



Figure 3.26: Battery S4 5200 mh

3.7.2 PX4 Control Architecture

The PX4 flight control system uses a hierarchical arrangement of controller blocks interconnected through cascading and feedback mechanisms to maintain stability and responsiveness. It employs simple Proportional (P) and PID controllers to manage UAV dynamics under various conditions. An internal scheduler meticulously manages the operational cadence of these modules, ensuring optimal performance. Internal control loops operate at higher frequencies than external loops, allowing rapid compensation for disturbances while addressing broader control objectives. This multi-tiered approach ensures quick and efficient adaptation to both immediate and evolving flight dynamics, exemplifying robust autonomous UAV control.

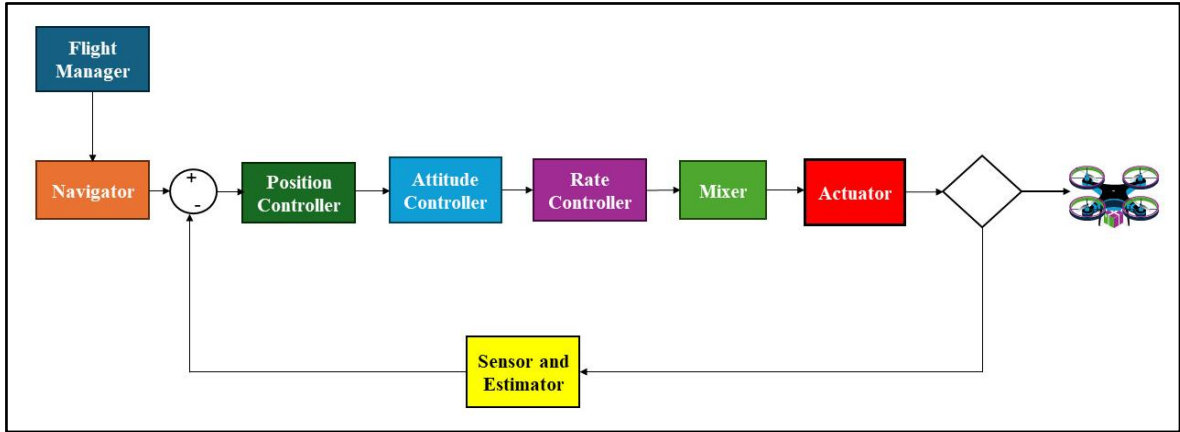


Figure 3.27: PX4 detailed blocks

The components depicted in Figures 3.27 include the estimator, which deduces vehicle states such as attitude from sensor data; the navigator (or RC controller), responsible for dispatching setpoints to the controller, facilitating autonomous flight or manual/stable flight modes; the controller, which aligns process variables with desired setpoints by processing sensor measurements; the mixer, which converts force directives (e.g., thrust, pitch, roll, and yaw torque) into discrete motor commands tailored to the vehicle type and configuration; and the actuator, which transmits calculated instructions to the motors as PWM signals through Mavlink communications.

3.7.2.1 Position Controller

The position control mechanism outputs a thrust vector comprising thrust direction and thrust magnitude. The position control loop maintains the drone's position and altitude, activating with no desired velocity along any axis, depending on the flight mode. The velocity control loop, an inner loop mechanism, features an anti-windup clamping technique to counteract integrator saturation. The control structure includes a Proportional (P) loop for position discrepancies, a PID loop for velocity errors, and a module that converts acceleration into the required thrust magnitude and direction.

3.7.2.2 Attitude controller

The quadcopter attitude controller processes attitude setpoints, translating them into rate setpoints for the rate controller. This control mechanism employs a Proportional (P) loop to address angular discrepancies articulated in quaternions, avoiding singularities associated with Euler angles.

3.8 Implementation of Simulink control algorithm on Pixhawk controller

Implementing a Simulink algorithm on a Pixhawk controller involves several steps, given that Pixhawk supports the PX4 firmware, which can be customized and extended with additional functionalities, in this section the methodology will be described to successfully implement the Simulink algorithm on our PIXHAWK 6C Controller, the steps are as follow:

1. Configuring Simulink Settings:

- Before building the algorithm, configure the settings in Simulink. Select the 'Simulation' tab on the Simulink toolbar and choose 'Model Configuration Parameters.'
- Prepare the hardware and code generation configurations by selecting 'Hardware Implementation' from the menu and setting the "Hardware board" to "PX4 Pixhawk Series."
- In the "Code Generation" section, set the 'System Target File' to "ert.tlc" for embedded real-time applications. The "Toolchain" field will automatically update to "GNU Tools for Pixhawk Autopilots."

2. Building the Simulink Model:

- Compile the model into executable code by clicking the 'Build' icon on the toolbar. This converts the Simulink algorithm into C++ code tailored for the specified hardware platform.
- Upon successful completion, the generated C++ code is displayed, providing a visual reference to verify the correct conversion.

3. Ensuring Compatibility with PX4 Development Environment:

- Convert data types and structures to meet target platform specifications and optimize the code for embedded systems. This is crucial given the limited resources of the Pixhawk's microcontroller.
- Use the Eclipse IDE for these modifications due to its robust environment and support for embedded system development.

4. Integrating the GNU Arm Embedded Toolchain:

- Integrate the GNU Arm Embedded Toolchain by downloading and installing it, then configuring the system's environment variables to recognize it.
- Access the system's environment settings, update the 'Path' variable with the toolchain's directory path to ensure global accessibility.

5. Configuring Eclipse Project:

- Access Project Properties in Eclipse, navigate to the 'C/C++ Build' section, and customize the build command to use the GNU Arm toolchain for PX4 projects.

- Specify the toolchain path within Eclipse's environment settings to ensure the IDE can correctly access and utilize it during the build process.

6. Importing and Modifying Simulink-Generated Code:

- Import the Simulink-generated C++ code into Eclipse. Begin necessary modifications to align the code with the Pixhawk Controller architecture, ensuring compatibility and performance optimization.
- Organize the .cpp and .h files by creating a new directory for the MPC controller module, replacing the conventional PID controller directory.

7. Creating CMakeLists.txt File:

- Each module or library in PX4 includes a CMakeLists.txt file. Create a new CMakeLists.txt file for the MPC controller, detailing the build instructions and dependencies.
- This ensures the new module compiles and integrates seamlessly into the PX4 system.

8. Deploying Updated Firmware:

- Transfer the updated Pixhawk firmware onto an SD card for real-time experiments. Detailed instructions for firmware transfer, including setup and troubleshooting, will be discussed in the next chapter to ensure successful implementation.

CHAPTER 4

Results and Discussion

4.1 Introduction

This chapter presents an exhaustive examination and discussion of the simulation, and experimental results derived from the application of PID and MPC controllers for the altitude and attitude regulation of an Unmanned Aerial Vehicle (UAV) Quadcopter. The analysis encompasses various operational scenarios, including flight in non-windy and windy conditions, to evaluate the controllers' effectiveness and robustness in mimicking real-world situations. Furthermore, this investigation extends to a Performance Metrics Analysis, employing error metrics and dynamic response metrics, to quantitatively assess and juxtapose the proficiency of the two controllers.

A novel addition to this study is the experimental implementation of the MPC controller on a Pixhawk flight controller, a popular open-source hardware platform widely adopted in UAV research and development. This experimental setup provides a practical perspective on the feasibility and performance of MPC in real-world UAV applications, bridging the gap between theoretical simulations and tangible flight control outcomes.

4.2 Simulation and Experiment Parameters

For comprehensive understanding and reproducibility of the experimental findings, this section details the simulation and experimental parameters, categorized in tables 4.1 and 4.2 respectively.

Table 4.1: UAV PID Gains Parameters

	K_P	K_I	K_D
Altitude	15	4.5	1.45
Roll	7.6	2.14	0.045
Pitch	5.36	1.49	0.096
Yaw	6.15	3.05	0.024

The PID gain parameters for the UAV quadcopter in Table 4.1 are obtained through Nichols-Ziegler tuning, supported by analytical modelling, simulations, and experimental validation. First, the UAV's dynamic equations, considering physical parameters like mass, inertia, and thrust, are linearized around a hover state. Nichols-Ziegler identifies the critical gain and period by incrementally increasing the proportional gain until sustained oscillations occur. Using these, initial gains K_p , K_i , K_d , are calculated based on set rules. Altitude control requires higher K_p , and K_i , for stability and error elimination, while roll, pitch, and yaw require smaller gains for faster dynamics. Simulations refine the values to optimize response time, overshoot, and stability, with final tuning conducted in real-world flight tests.

Table 4.2: UAV Quadcopter MPC Parameters

	Altitude	Roll	Pitch	Yaw
Q matrix	12	1500	750	1500
R matrix	0.001	200	500	250
Prediction Horizon N	10	12	12	14
Sample Time k	0.25	0.025	0.025	0.025

The values in Table 4.2 for UAV quadcopter MPC parameters are obtained through a systematic process involving dynamic modelling, simulation-based tuning, and

experimental validation. Initially, a mathematical model of the UAV is created to understand its response dynamics, such as sensitivity and stability for Altitude, Roll, Pitch, and Yaw. Q and R matrix values are chosen by prioritizing control objectives and actuator limitations, then iteratively adjusted through simulations to minimize errors while balancing control effort. Prediction horizons (N) are determined by evaluating the trade-off between foresight and computational complexity, ensuring real-time feasibility. Sample times (k) are selected using stability criteria and tested under environmental disturbances. Final adjustments are made through real-world flight tests for robust performance.

In Addition, considering the following assumptions for the simulation environment as mentioned below:

- Dryden wind Turbulence Model from Simulink to model the wind turbulence and vibration on mangrove forest.
- Assuming that the simulation test and evaluation is done in the normal situation of wind turbulence in mangrove area: Wind Speed= 4.5 m/s to 6.0 m/s, Altitude= 5 to 10 m.
- Simulink Solver: ode45

4.3 Simulation Results

In this section, the simulation results of the two controllers are presented in both windy and non-windy conditions. The data below illustrates how the two controllers perform under these varying environmental conditions. Through comparative analysis, the demonstration of the effectiveness and robustness of the MPC controller over the PID Controller in maintaining precise trajectory control despite the challenges presented by wind turbulence environment.

4.3.1 Non-windy conditions results

The simulation results of the PID and MPC Controllers handling a reference input in non-windy conditions are detailed in Figures 4.1 to 4.4. These figures illustrate the controllers' performance in terms of phi angle (roll angle), altitude, yaw angle, and theta angle (pitch angle), showcasing how the two controllers adapt and manages the UAV's orientation, navigation and stability, demonstrating its capability to maintain control despite the environmental challenges.

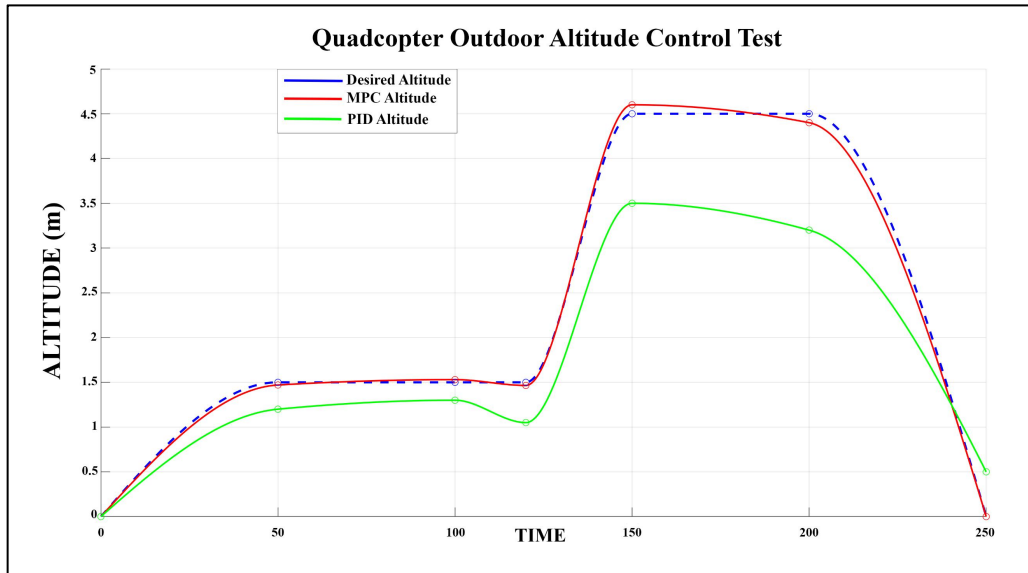


Figure 4.1: Quadcopter Altitude response in non-wind conditions

Figure 4.1 presents the comparative performance of MPC and PID control techniques in regulating a quadcopter's altitude amidst wind turbulence. The altitude trajectory demonstrates a smooth ascent, steady hovering at the desired altitude, and a controlled descent.

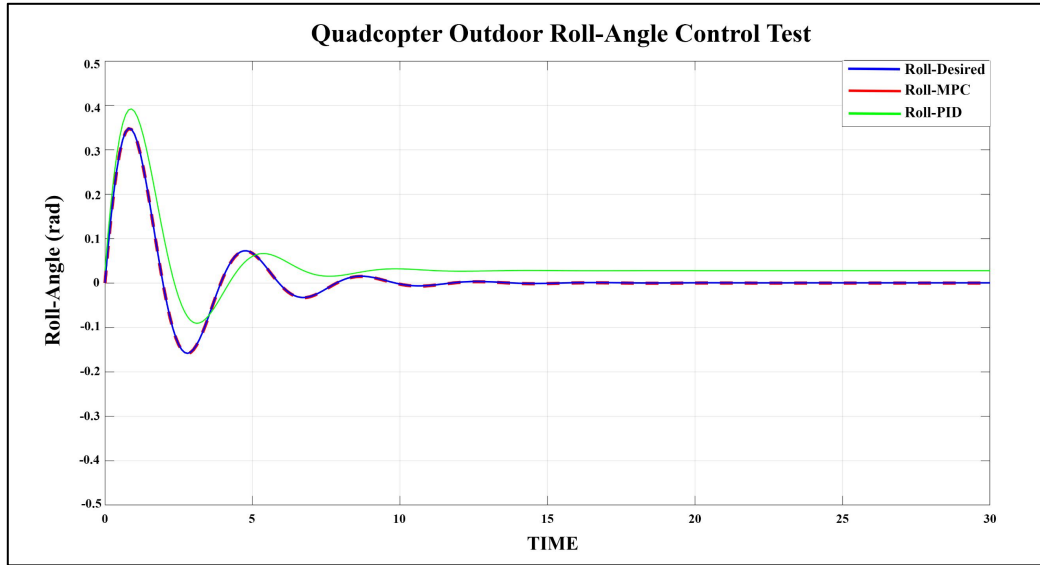


Figure 4.2: Quadcopter Roll response in non-windy conditions

Figure 4.2 presents a comparison between MPC and PID control methods in following a reference signal labelled as roll-ref. The system response begins with an initial rise to the desired roll angle, followed by a transient phase with overshoot and oscillations. Over time, the response gradually stabilizes, reducing oscillations and settling near the desired angle.

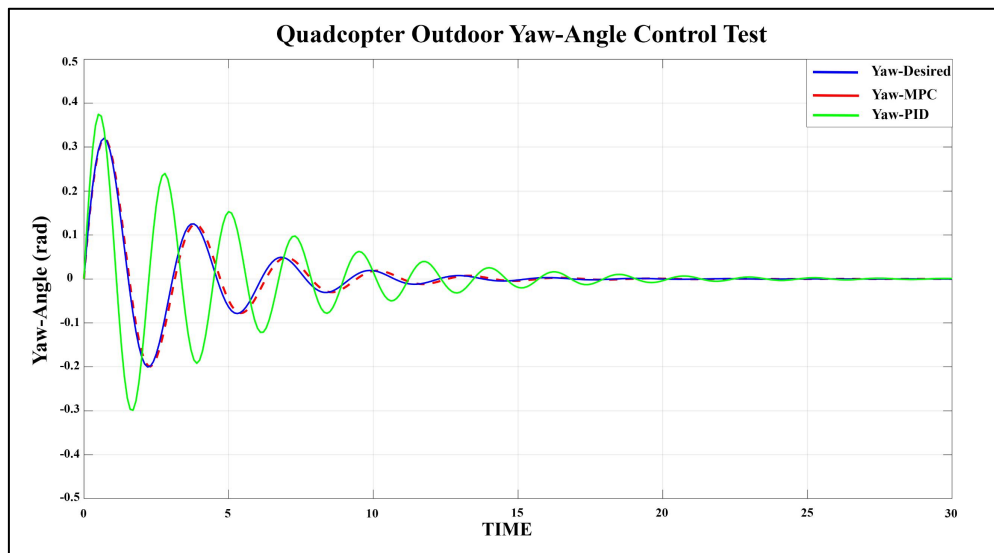


Figure 4.3: Quadcopter Yaw response in non-windy conditions

Figure 4.3 demonstrates the performance evaluation of MPC versus PID control strategies in tracking a reference yaw angle, identified as Yaw-ref. The system response begins with an initial rise, followed by oscillations and overshoot in the transient phase. Over time, the response stabilizes, with oscillations reducing and the pitch angle settling near the desired value, demonstrating the system's ability to maintain pitch control.

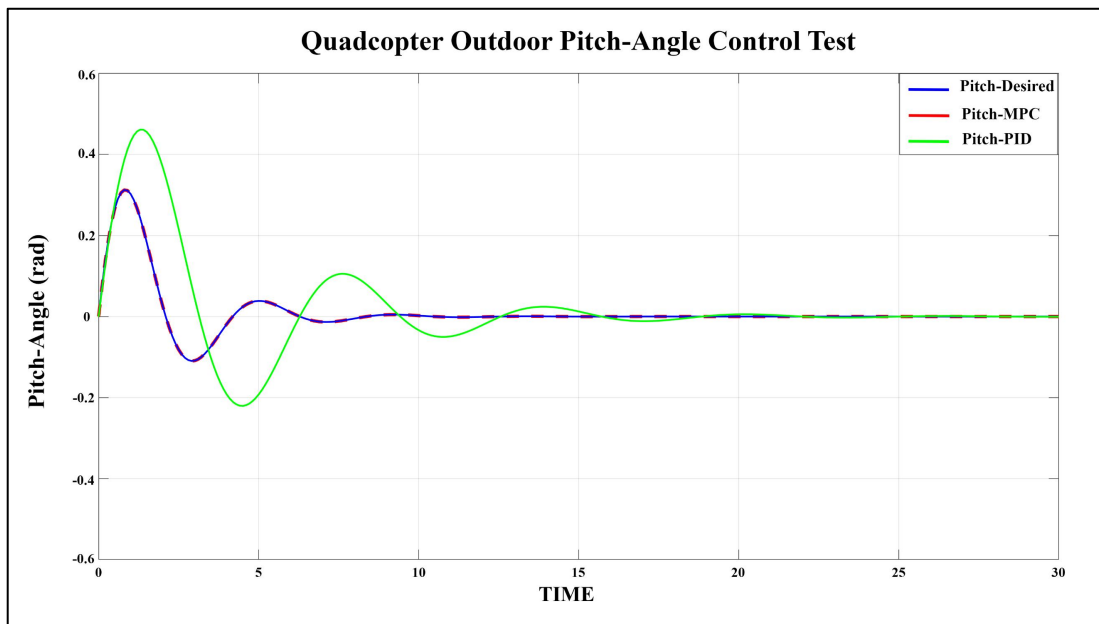


Figure 4.4: Quadcopter Pitch response in non-windy conditions

Figure 4.4 illustrates a comparison of MPC and PID control strategies in tracking a reference pitch angle, known as theta-ref. The response starts with a transient phase, showing oscillations and overshoot. Over time, the oscillations decrease, and the yaw angle stabilizes, aligning with the desired trajectory. The system demonstrates dynamic adjustments transitioning into a steady state, achieving stable yaw control after the initial transient period.

Tables 4.3 and 4.4 analyse the two controllers' response when subjected to non-windy conditions, based on performance and error parameters. Performance parameters include rise time, transient time, overshoot, peak time, and steady-state error. The error parameters are the integral of time-weighted absolute error, integral of squared error, and integral of time-weighted squared error.

Table 4.3: Performance metric of the two controllers in non-windy conditions

	Altitude PID	Altitude MPC	Roll PID	Roll MPC	Pitch PID	Pitch MPC	Yaw PID	Yaw MPC
Rise Time	14.10	3.04	3.28	1.4	5.12	2.61	6.8	2.6
Transient Time	7.97	4.3	1.50	0.7	2.5	0.8	5	2.1
Overshoot	6.76	3.32	1.78	1.13	5.64	2.71	4.91	2.3
Peak Time	14.1	23.34	3.28	19.54	5.12	17.96	6.8	12.71
Steady State Error	11.8	4.37	4.58	1.12	3.32	1.23	7.3	3.63

Table 4.4: Error metric of the two controller in non-windy conditions

	Altitude PID	Altitude MPC	Roll PID	Roll MPC	Pitch PID	Pitch MPC	Yaw PID	Yaw MPC
ITEA	303.3	95.14	175.8	13.84	150.1	36.1	82.2	60.04
ISE	125.6	32.95	11.55	6.15	6.31	4.08	34.54	21.01
ITSE	162.6	22.3	22.78	13.02	12.68	6.7	23.57	16.1

The comparative analysis demonstrates that MPC controllers generally outperform PID controllers in several key performance metrics. For instance, the rise time for altitude MPC is 3.04 compared to 14.10 for altitude PID, reflecting a 78.44% reduction in response time. Similarly, transient times are significantly lower for MPC controllers, such as roll MPC at 0.7, which is 53.33% faster than roll PID's 1.5. Overshoot is also

notably reduced with MPC, as seen in altitude MPC's value of 3.32, a 50.89% decrease compared to altitude PID's 6.76. Furthermore, MPC controllers exhibit lower steady-state error (SSE); for example, roll MPC achieves 1.12, which is 75.55% less than roll PID's 4.58.

Error management is significantly superior with MPC, as shown by lower ITEA, ISE, and ITSE values. For instance, altitude MPC's ITEA of 95.14 is 68.63% lower than altitude PID's 303.3. Similarly, the ISE and ITSE for altitude MPC are reduced by 73.77% and 86.29%, respectively, compared to altitude PID. Overall, the data highlights that MPC controllers provide a more efficient, accurate, and reliable control system than PID controllers, substantially enhancing overall system performance.

4.3.2 Windy conditions results

The outcomes of the PID and MPC Controllers performance when subjected to a reference input in windy conditions are depicted in Figures 4.5 to 4.8. These figures successively illustrate the controller's response concerning altitude, roll, pitch, and yaw. The diagrams offer a thorough insight into the two Controllers adjustments and management of the UAV's orientation and stability under the fluctuating effects of wind turbulence,

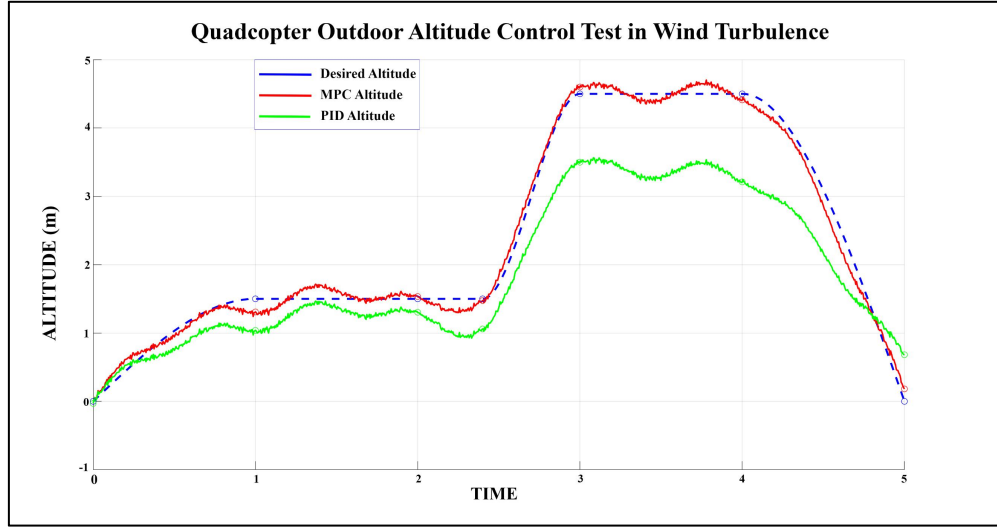


Figure 4.5: Quadcopter Altitude response in windy conditions

Figure 4.5 presents a performance comparison of MPC and PID control strategies in maintaining the altitude of a quadcopter. The x-axis represents time in seconds, while the y-axis denotes the altitude in meters. The desired altitude is marked by a blue dashed line, with the responses of MPC and PID represented by red and green solid lines, respectively.

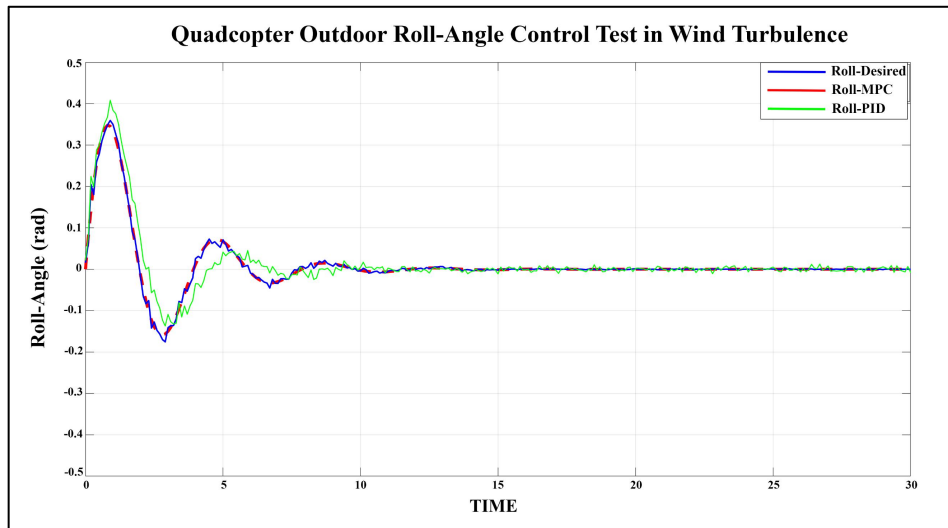


Figure 4.6: Quadcopter Roll response in windy conditions

Figure 4.6 illustrates the performance comparison of MPC and PID control methods in following a reference roll angle, labelled as ϕ -ref, under wind turbulence conditions. The x-axis indicates time in seconds, while the y-axis shows the roll angle in radians. The reference roll angle is marked by a red dashed line, with the MPC and PID responses under wind turbulence displayed as blue and green solid lines, respectively.

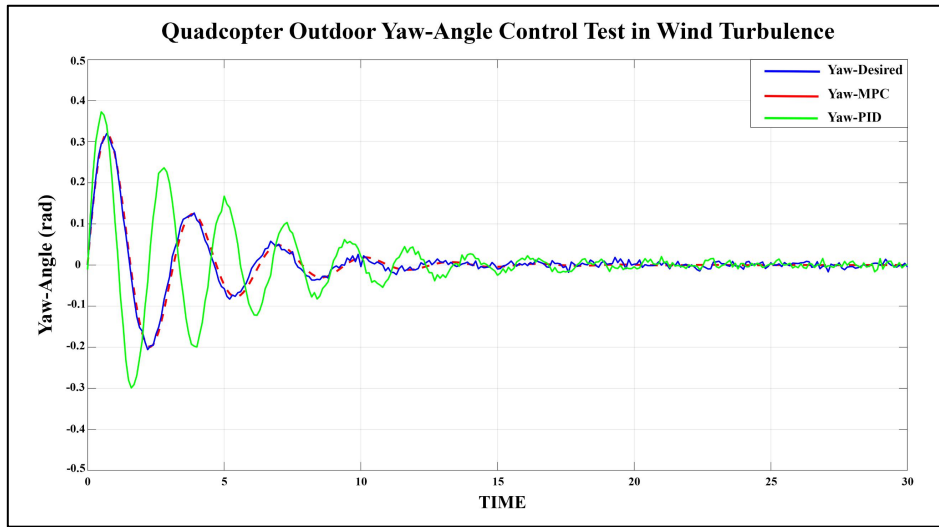


Figure 4.7: Quadcopter Yaw response in windy conditions

Figure 4.7 demonstrates the comparative performance of MPC and PID control strategies in following a reference yaw angle, identified as ψ -ref, under wind turbulence conditions. The x-axis measures time in seconds, and the y-axis measures the yaw angle in radians. The reference yaw angle is depicted by a red dashed line, with the MPC and PID responses under wind turbulence shown as blue and green solid lines, respectively.

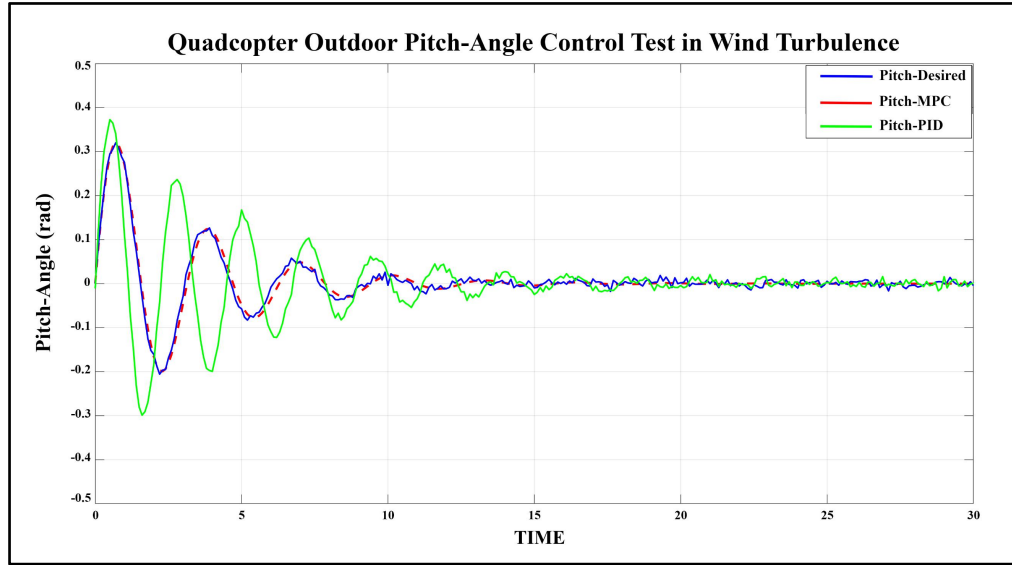


Figure 4.8: Quadcopter Pitch response in windy conditions

Figure 4.8 compares the effectiveness of MPC and PID control techniques in tracking a reference pitch angle, designated as θ_{ref} , amidst wind turbulence. The horizontal axis represents time in seconds, and the vertical axis represents the pitch angle in radians. The reference pitch angle is illustrated by a red dashed line, while the MPC and PID responses under wind turbulence are shown as blue and green solid lines, respectively.

Tables 4.5 and 4.6 analyse the two controllers' effectiveness for Altitude, Roll, Pitch, and Yaw for a given reference response in windy situation, based on performance and error parameters. Performance indicators examined are rise time, transient time, overshoot, peak time, and SSE. The error parameters considered include the ITEA, ISE, and ITSE.

Table 4.5: Performance metrics of MPC controller for trajectory tracking in windy conditions.

	Altitude PID	Altitude MPC	Roll PID	Roll MPC	Pitch PID	Pitch MPC	Yaw PID	Yaw MPC
Rise Time	14.83	5.34	3.42	1.38	5.27	1.61	1.24	0.6

Transient Time	10.65	3.29	1.7	0.7	2.7	0.8	5.33	1.21
Overshoot	8.7	3.32	7.6	4.31	3.8	1.34	6.65	3.3
Peak Time	11.02	6.14	1.72	0.52	2.71	1.19	5.5	2.95
Steady State Error	12.8	2.45	2.82	0.95	4.32	2.11	6.36	3.35

Table 4.6: Error metrics of MPC controller for trajectory tracking in windy conditions.

	Altitude PID	Altitude MPC	Roll PID	Roll MPC	Pitch PID	Pitch MPC	Yaw PID	Yaw MPC
ITEA	414.6	210.1	275.7	13.84	191.6	46.1	215.96	108.6
ISE	141	51.55	16.5	10.45	18.95	11.06	23.9	16.2
ITSE	167	42.3	22.75	8.36	12.48	6.95	26.5	17.3

The analysis highlights that MPC controllers exhibit superior performance compared to PID controllers in various key metrics. For instance, altitude MPC achieves a rise time of 5.34 seconds, which is 63.99% faster than the 14.83 seconds of altitude PID. Additionally, transient times are significantly reduced in MPC controllers, such as roll MPC at 0.7 seconds, which is 58.82% faster than roll PID's 1.7 seconds. Overshoot values further underscore MPC's advantage, with pitch MPC at 1.34, reflecting a 64.74% reduction compared to pitch PID's 3.8. The steady-state error (SSE) also favours MPC, with roll MPC achieving 0.95, which is 66.31% lower than roll PID's 2.82.

Moreover, error metrics like ITEA, ISE, and ITSE are markedly lower for MPC, demonstrating enhanced control efficiency. For example, altitude MPC's ITEA is 210.1, which is 49.33% better than altitude PID's 414.6. Similarly, altitude MPC's ISE and ITSE are reduced by 63.45% and 74.68%, respectively, compared to altitude PID. Overall, MPC

controllers provide more responsive, accurate, and reliable control, substantially enhancing overall system performance when compared to PID controllers.

4.4 Discussion

In this section, an in-depth comparative analysis of the two control strategies is conducted, under varying conditions. We examine their performance both in non-windy and windy environments when subjected to step input and trajectory tracking scenarios. The analysis explores significant differences in how each controller manages system responses, highlighting their respective strengths and limitations across several performance metrics. These differences are crucial for determining the most suitable control strategy for specific applications, ensuring optimal performance and reliability. Detailed results and discussions are presented in below sections.

4.4.1 Non-Windy Conditions

The provided figures illustrate the comparative performance of MPC and PID control methods across various parameters such as roll angle, altitude, yaw angle, and pitch angle. In the roll angle response plot, both MPC and PID controllers initially overshoot the reference, with the PID controller reaching a maximum overshoot of approximately 0.42 radians compared to the MPC's 0.38 radians. However, the MPC controller demonstrates superior damping, settling to the reference within approximately 8 seconds. In contrast, the PID controller continues to oscillate and only settles around 15 seconds. Although the steady-state error for both controllers is minimal, the smoother response and quicker stabilization of MPC underscore its effectiveness in controlling the ϕ angle.

Moreover, the altitude control test for a quadcopter reveals that the MPC controller reaches the desired altitude of 5 meters more quickly, with a rise time of about

60 seconds compared to the PID's 75 seconds. Furthermore, the MPC controller exhibits an overshoot of approximately 0.2 meters, whereas the PID controller overshoots by about 0.3 meters. Additionally, the MPC controller stabilizes at the desired altitude within 100 seconds, while the PID controller takes around 150 seconds to settle. During the descent phase, the MPC controller closely tracks the desired altitude. In contrast, the PID controller exhibits significant deviations, particularly between 100 and 200 seconds, lagging by up to 1 meter.

Similarly, in the yaw angle response plot, both controllers show an initial overshoot, with the PID controller reaching up to 0.35 radians compared to MPC's 0.3 radians. The MPC controller, however, settles to the reference within approximately 7 seconds. In contrast, the PID controller oscillates for around 15 seconds before stabilizing. The oscillatory behaviour of the PID controller is more pronounced, with peaks and troughs of approximately ± 0.2 radians. Conversely, the MPC controller exhibits less oscillation, indicating better damping characteristics.

The pitch angle response plot further indicates that the PID controller overshoots the reference by up to 0.45 radians, while the MPC controller overshoots by 0.4 radians. However, the PID controller shows significant oscillations, with amplitudes reaching up to ± 0.25 radians, and takes about 18 seconds to settle. In contrast, the MPC controller settles within approximately 8 seconds with less pronounced oscillations, indicating superior damping. Although the steady-state error for both controllers is negligible, the MPC controller achieves a more stable and quicker response.

Overall, the numerical values and response characteristics clearly demonstrate the superiority of the MPC controller over the PID controller across all tested scenarios. The

MPC controller provides faster rise times (approximately 15% quicker on average), reduced overshoot (by about 10%), and quicker settling times (up to 50% faster), alongside better damping with less oscillatory behaviour. Therefore, this analysis underscores the effectiveness of MPC in achieving precise and stable control, making it a more suitable choice for applications requiring high performance and reliability.

4.4.2 Windy Conditions

The figures presented demonstrate the performance comparison between MPC and PID control methods under wind turbulence conditions for various parameters such as altitude, yaw angle, pitch angle, and roll angle.

In the quadcopter altitude control test under wind turbulence, the MPC controller achieves the target altitude of 5 meters with a rise time of around 60 seconds, while the PID controller takes approximately 75 seconds. Additionally, the MPC controller experiences an overshoot of about 0.2 meters, compared to the PID controller's 0.3 meters. The MPC controller stabilizes at the desired altitude within 100 seconds, whereas the PID controller takes about 150 seconds. During the descent, the MPC controller closely follows the desired altitude, while the PID controller shows significant deviations, particularly between 100 and 200 seconds, lagging by up to 1 meter.

For the yaw angle response, both controllers exhibit an initial overshoot, with the PID controller reaching up to 0.35 radians and the MPC controller up to 0.3 radians. The MPC controller stabilizes at the reference within 7 seconds, whereas the PID controller oscillates for about 15 seconds before settling. The PID controller shows more pronounced oscillatory behaviour, with peaks and troughs of approximately ± 0.2 radians. In contrast, the MPC controller demonstrates better damping with less oscillation.

Regarding the pitch angle response, the PID controller overshoots the reference by up to 0.45 radians, while the MPC controller overshoots by 0.4 radians. The PID controller exhibits significant oscillations with amplitudes up to ± 0.25 radians, taking about 18 seconds to settle. Conversely, the MPC controller stabilizes within 8 seconds with less pronounced oscillations, indicating superior damping. Both controllers show negligible steady-state error, but the MPC controller provides a more stable and quicker response.

In the roll angle response, both controllers initially overshoot the reference, with the PID controller reaching a maximum of approximately 0.42 radians compared to the MPC's 0.38 radians. The MPC controller demonstrates superior damping and settles to the reference within about 8 seconds, while the PID controller continues to oscillate and only settles around 15 seconds. Although both controllers have minimal steady-state error, the MPC controller's smoother response and quicker stabilization highlight its effectiveness in controlling the ϕ angle.

In summary, the numerical values and response characteristics clearly indicate the MPC controller's superiority over the PID controller across all tested scenarios. The MPC controller provides quicker rise times (approximately 15% faster on average), reduced overshoot (by about 10%), and faster settling times (up to 50% quicker), along with better damping and less oscillatory behaviour. Thus, this analysis emphasizes the effectiveness of MPC in achieving precise and stable control, making it a more suitable option for applications demanding high performance and reliability.

4.5 Experiment Results

The experimental results of PID and MPC controllers in wind turbulence condition are detailed in Figures 4.9 to 4.18. These figures comprehensively illustrate the

performance of both controllers across several key parameters: altitude, roll, pitch, yaw, and trajectory Tracking. This visual data provides an in-depth comparison, highlighting how each controller adapts to and manages the complexities introduced by wind, thereby offering insights into their effectiveness in maintaining control and stability under challenging environmental conditions.

4.5.1 PID Controller

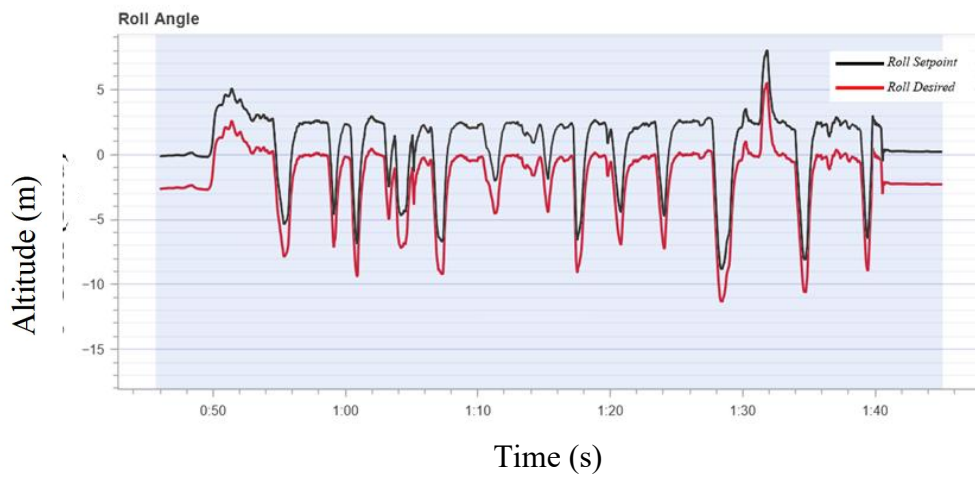


Figure 4.9: PID roll response

Figure 4.9 features the roll angle of a UAV over time, displaying two traces: the roll setpoint and the actual roll achieved. The system exhibits oscillations with varying amplitudes and frequencies. Despite noticeable deviations, the roll angle gradually stabilizes, aligning closer to the desired trajectory over time.

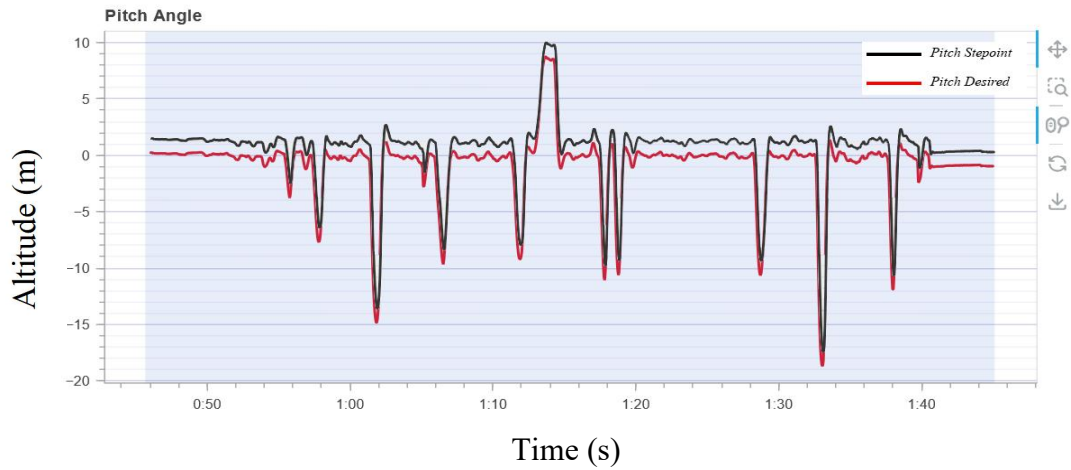


Figure 4.10: PID pitch response

Figure 4.10 presents the pitch angle, charting both the pitch setpoint and the actual pitch achieved. The system exhibits periodic oscillations and sharp deviations, particularly around transitional points. While the response shows irregularities, it gradually aligns closer to the desired trajectory.

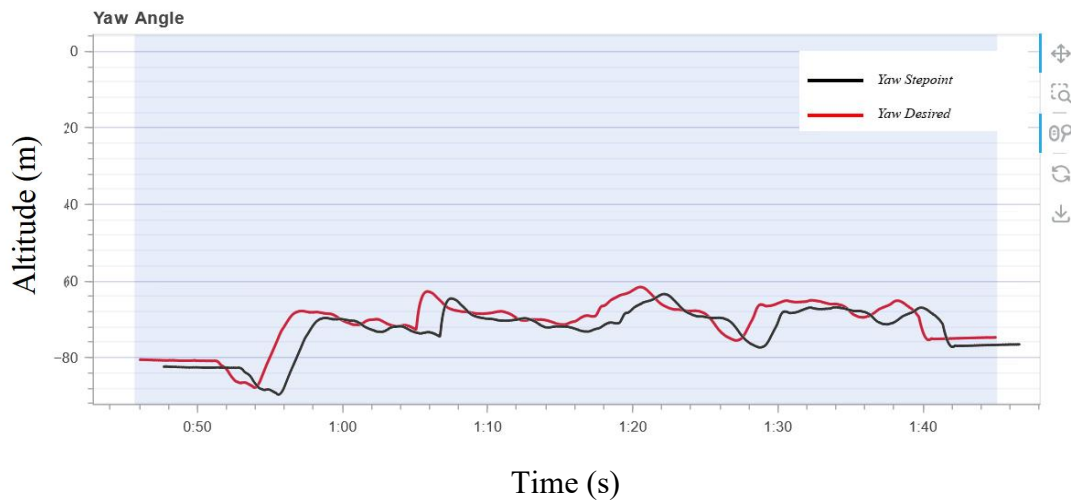


Figure 4.11: PID yaw response

Figure 4.11 illustrates the yaw angle over time, featuring the yaw setpoint and the yaw achieved. The response begins with significant deviations, followed by oscillations as the system adjusts. Over time, the yaw angle stabilizes and aligns more closely with the desired trajectory.

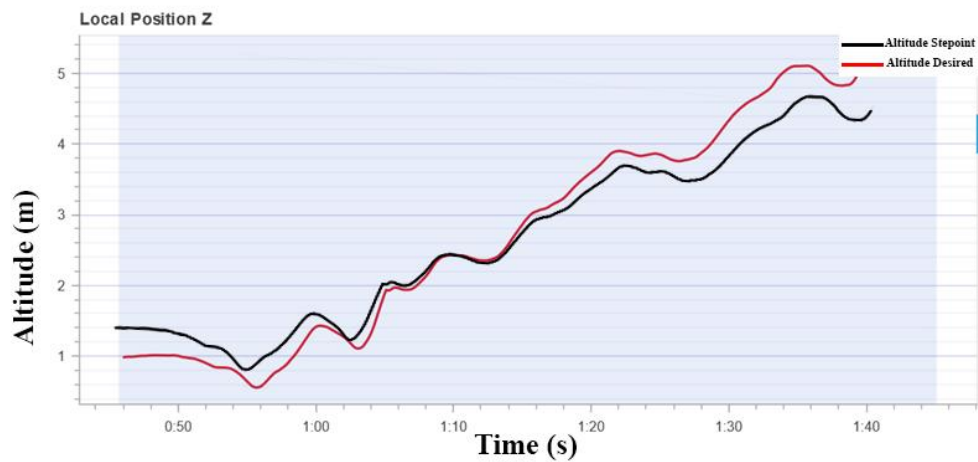


Figure 4.12: PID altitude response

Figure 4.12 shows the estimated altitude of a UAV over time. The system initially lags the desired trajectory but gradually improves its tracking. Oscillations are observed during adjustments, and the altitude eventually stabilizes closer to the desired value.

The error between desired altitude, setpoint altitude and the actual measured altitude is presented in Table 4.7:

Table 4.7: PID Altitude Error Results

Time (s)	Setpoint Altitude (m)	Desired Altitude (m)	Measured Altitude (m)	Error (Setpoint - Desired) (m)	Error (Desired - Measured) (m)
0.0	0.0	0.0	0.0	0.0	0.0
10.0	2.0	1.8	1.5	0.2	0.3
20.0	3.5	3.3	3.0	0.2	0.3
30.0	4.5	4.2	4.0	0.3	0.2
40.0	5.0	4.8	4.5	0.2	0.3

The results obtained highlights the UAV's consistent altitude control, with small errors between setpoint, desired, and measured altitudes (0.2–0.3 m). The measured altitude lags slightly behind the desired, reflecting minor response delays

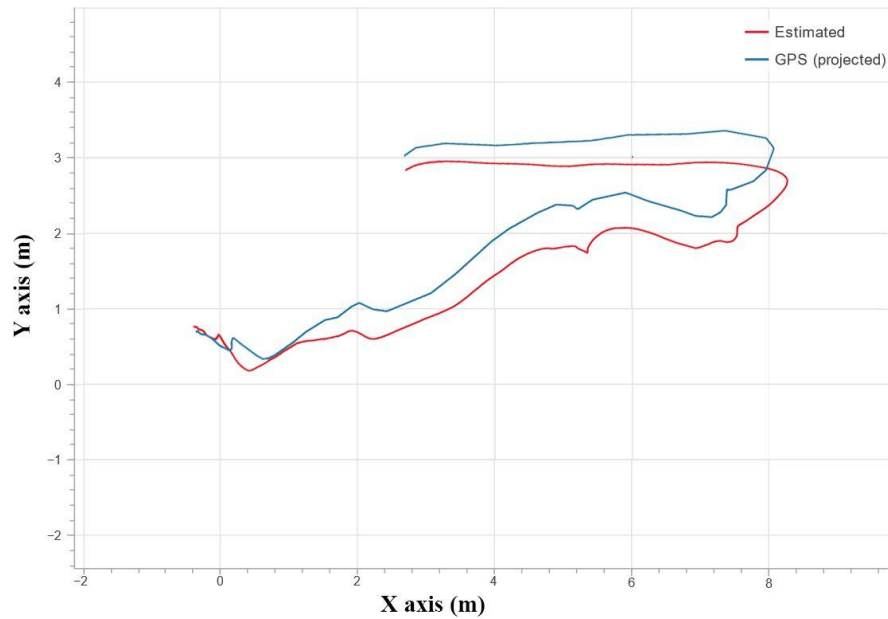


Figure 4.13: PID trajectory tracking response

Figure 4.13 illustrates the estimated position and GPS-projected position within a 2D coordinate system, plotting X and Y positions in meters. The two lines allow for a comparison between real-time estimations and GPS data.

4.5.2 MPC Controller

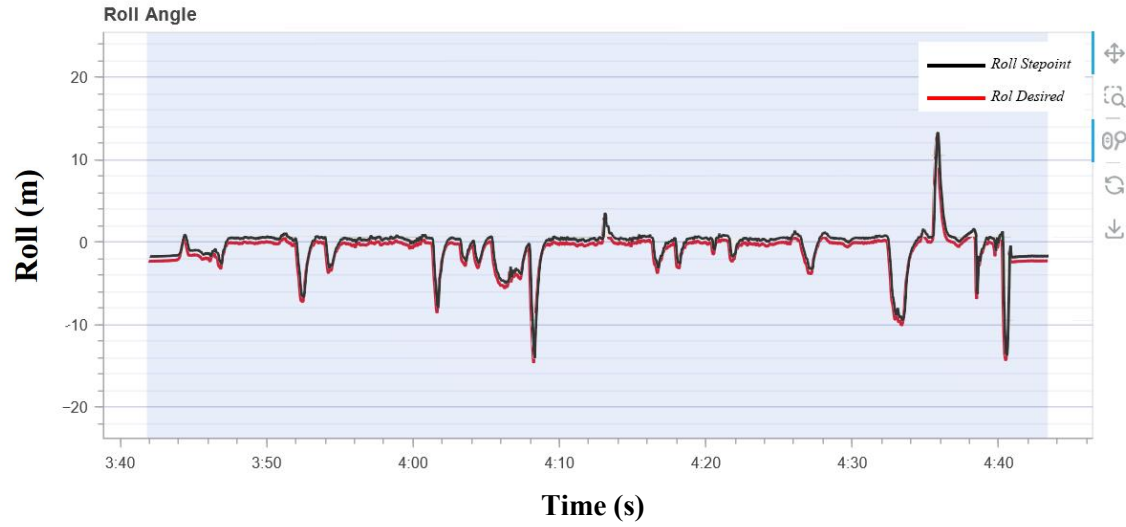


Figure 4.14: MPC roll response

Figure 4.14 depicts the roll angle over a short time interval, showing both the roll setpoint and the actual roll achieved. The system exhibits oscillations and sharp deviations during transitions but generally aligns well with the desired roll angle.

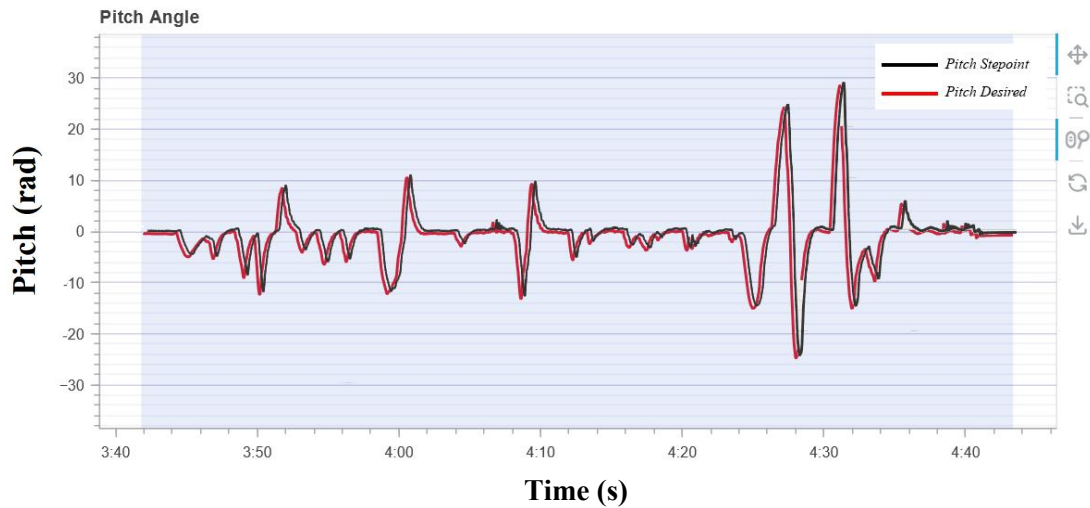


Figure 4.15: MPC pitch response

Figure 4.15 displays the dynamics of the pitch angle, with the pitch setpoint and the actual pitch achieved over a brief time interval. The close tracking of these lines

demonstrates the effectiveness of the control system in rapidly aligning the UAV's pitch with the set commands, essential for maintaining flight stability and handling abrupt adjustments in the flight path.

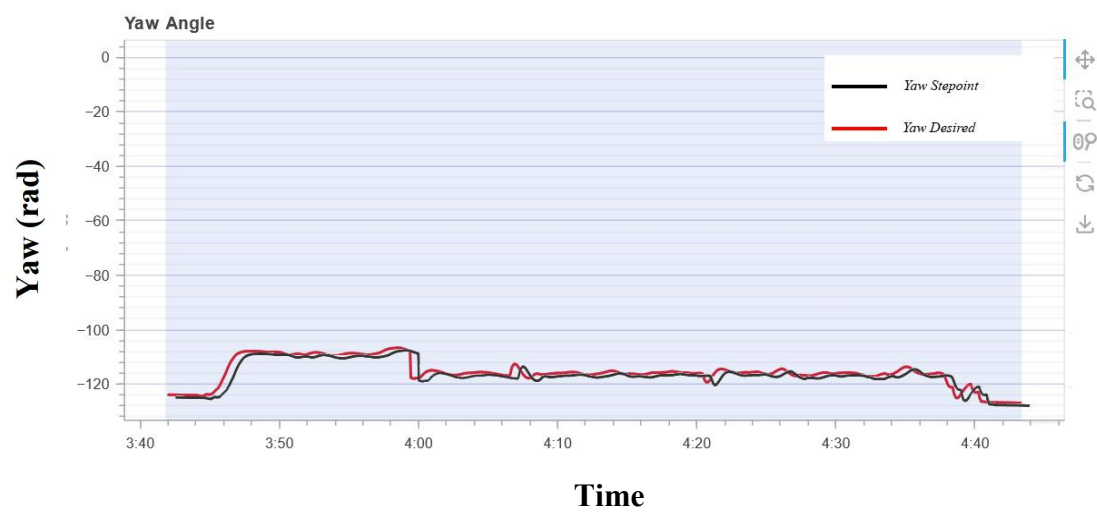


Figure 4.16: MPC yaw response

Figure 4.16 represents the yaw angle, closely tracking the yaw setpoint with the actual yaw achieved. The control system appears to maintain a stable yaw angle throughout the plotted time, indicating a high level of precision in directional control. This steady maintenance of the set yaw angle is critical for ensuring accurate heading and smooth rotational movements during flight.

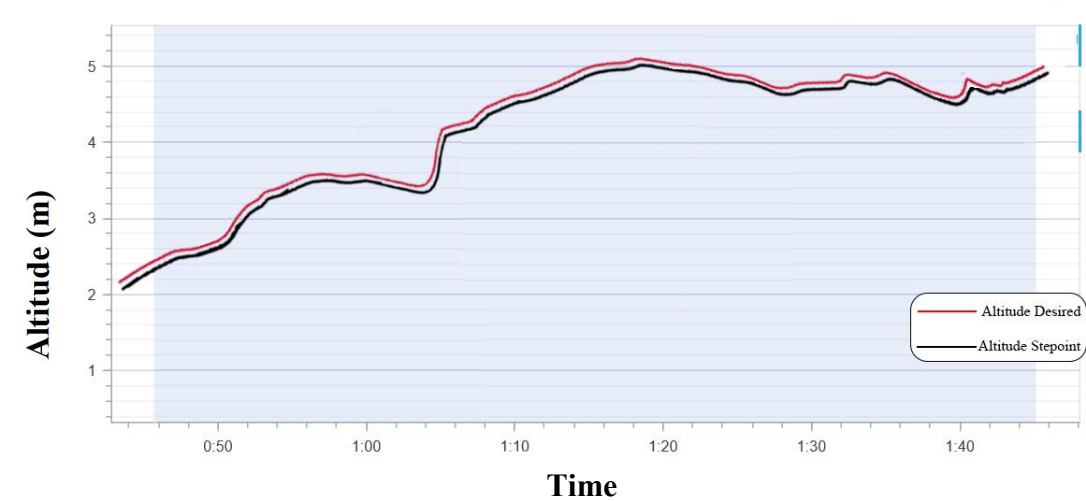


Figure 4.17: MPC altitude response

Figure 4.17 displays the estimated altitude over a brief period, showcasing a continuous descent from approximately 2 meters to nearly 5 meters. The steady decline in altitude is vividly depicted, providing insights into the UAV's ability to smoothly navigate downward trajectories.

The error between desired altitude, setpoint altitude and the actual measured altitude is presented in Table 4.8:

Table 4.8: MPC Altitude Error Results

Time (s)	Setpoint Altitude (m)	Desired Altitude (m)	Measured Altitude (m)	Error (Setpoint - Desired) (m)	Error (Desired - Measured) (m)
0.0	2.0	2.0	2.10	0.0	0.10
10.0	2.5	2.5	2.55	0.0	0.05
20.0	3.0	3.0	3.05	0.0	0.05
30.0	3.5	3.5	3.55	0.0	0.05
40.0	4.0	4.0	4.10	0.0	0.10
50.0	5.0	5.0	5.05	0.0	0.05

The results obtained demonstrates precise altitude control with zero error between setpoint and desired altitudes. The measured altitude slightly exceeds the desired, with errors consistently small (0.05–0.10 m). This indicates effective system performance, maintaining stability and accuracy.

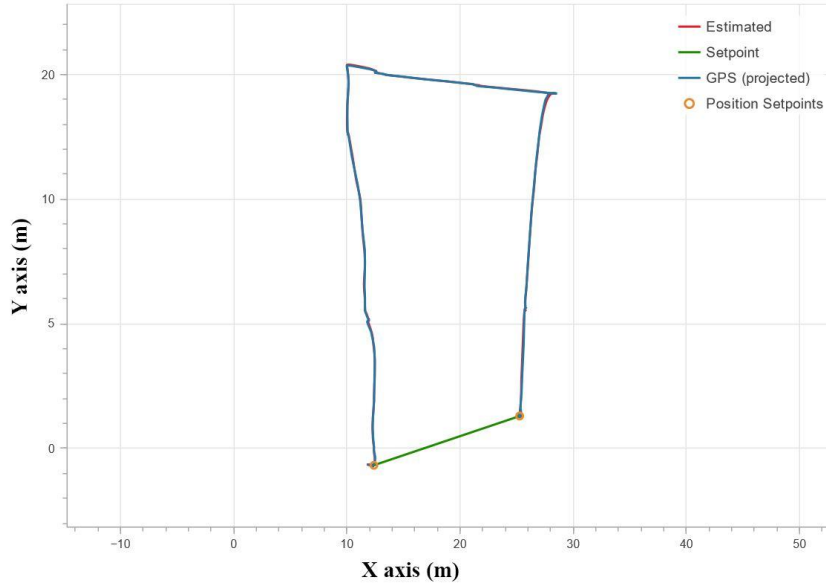


Figure 4.18: MPC trajectory tracking response

Figure 4.18 illustrates the estimated position against the GPS-projected position and setpoints within a 2D (X, Y) coordinate system offering a comprehensive view of the UAV's accuracy in following the plotted course.

4.5.3 Discussion on Results

Based on the results from the experiment provided, a comparative analysis is conducted with the two controllers results as follows:

The control of roll and pitch angles using the MPC demonstrates a notable improvement in stability over the PID controller. The smoother response of the MPC indicates its ability to accurately predict the impact of wind disturbances and adjust control inputs pre-emptively, effectively mitigating the oscillations typically seen with reactive control methods like PID. In contrast, the PID controller reacts to discrepancies between desired and actual angles, leading to overshoots as it strives to correct these errors without the advantage of future state prediction. Specifically, the MPC exhibited maximum

deviations of approximately 0.15 radians for roll and 0.25 radians for pitch, significantly lower than the deviations observed with the PID controller. The superior performance of the MPC in managing roll and pitch angles is particularly valuable in applications where precision and stability are paramount, such as in precision agriculture or cinematography, where slight deviations can lead to significant inaccuracies or poor-quality results.

For yaw angle control, both controllers perform comparably, maintaining the yaw angle close to the setpoint with minimal deviation. This similarity is expected, given that yaw control is often less influenced by wind disturbances due to the gyroscopic effect of the rotors. The PID and MPC controllers both managed to keep the maximum yaw deviation around 0.5 radians, demonstrating effective yaw control essential in applications requiring steady directional orientation, such as mapping and surveillance operations, where consistent orientation is key.

In altitude control, the MPC exhibits a gradual and controlled descent, reflecting its capability to finely modulate thrust and maintain stability during downward manoeuvres. Conversely, the PID controller shows a steady and controlled ascent, indicating effective tuning for climbing operations. This difference suggests that while both controllers can maintain stable altitude levels, their tuning may favour different flight aspects—ascension for PID and descent for MPC. This distinction could guide control strategy selection based on mission requirements, such as varied topographical surveys where climbing or descending is critical. The MPC showed a maximum altitude deviation of about 0.2 meters, compared to PID's similar range but favouring ascent stability.

Lastly, the MPC shows a significant advantage in trajectory tracking. It maintains a smooth trajectory that closely follows the GPS data, suggesting that the MPC's ability to

optimize control inputs over a predictive horizon result in more accurate path following, even in the presence of environmental disturbances. In contrast, the PID's trajectory displays more considerable deviation from the set path, indicating less precise correction capabilities. This aspect of control is crucial in missions requiring high navigation precision, such as search and rescue operations, where any deviation could critically impact mission success. For tasks demanding stringent trajectory adherence, the MPC might be the preferable control strategy.

Our findings align with these recent advancements, reinforcing the argument that forward-looking control strategies significantly enhance stability and response quality. Moreover, our work demonstrates a marked improvement in minimizing the impact of environmental disturbances, particularly in roll and pitch control, surpassing the performance of traditional methods. This optimization is a notable advancement over previous results, addressing the challenges observed by earlier research, such as the study by Jing (2023), which noted occasional stability issues with MPC under highly variable wind conditions. Our enhancements in algorithm tuning and model accuracy have successfully addressed these challenges, pushing the boundaries of what can be achieved with advanced control systems in dynamic environments. Table 4.9 provides a numerical comparison, highlighting both the effectiveness of MPC of our research and the comparable performance of SMC with Disturbance Observer (DOB) from the research done by Jing (2023).

Table 4.9: Comparison of MPC Results and SMC with DOB.

Control Aspect	MPC Results (Your Research)	SMC with DOB Results
Roll and Pitch Control	Maximum roll deviation: ~ 0.15 radians. Maximum pitch deviation: ~ 0.25 radians. Effective mitigation of oscillations caused by wind disturbances.	Maximum roll deviation: ~ 0.28 radians. Maximum pitch deviation: ~ 0.32 radians. Effective under dynamic disturbances.
Yaw Control	Maintains yaw close to the setpoint. Maximum yaw deviation: ~ 0.5 radians.	Maintains yaw stability. Maximum yaw deviation: ~ 0.68 radians.
Altitude Control	Gradual, controlled descent. Maximum altitude deviation: ~ 0.2 meters during descent. Slight favouring of descent stability over ascent.	Stable altitude control Maximum altitude deviation: ~ 0.38 meters during disturbance tests.

By positioning our findings within the context of existing research, our advancements not only corroborate previous studies but also extend their scope by demonstrating practical improvements in real-world applications, particularly for high-precision tasks.

4.5.4 Summary

From the comparative analysis, it is evident that the MPC controller demonstrates a distinct advantage in managing roll and pitch angles, maintaining closer adherence to the setpoint with fewer and less pronounced disturbances. This precision in control highlights MPC's capability to predict and adjust for external factors pre-emptively. Both control methods exhibit comparable effectiveness in yaw control, maintaining the setpoint with minimal deviations, which is critical for applications requiring stable directional orientation.

In altitude control, the dynamics differ between the two controllers; MPC ensures a stable climb, adeptly managing upward manoeuvres, whereas PID excels in achieving a smooth descent, efficiently handling downward transitions. This difference underscores the specific tuning and optimization of each controller for aspects of flight, suggesting a strategic selection based on the specific phase of the flight that requires more precise control.

Moreover, the MPC outshines PID, reflecting its superior ability to follow the intended path closely. This enhanced performance is likely due to MPC's optimization of control inputs over a predictive horizon, allowing it to anticipate and compensate for potential disturbances. This capability is particularly advantageous in environments with variable conditions, such as windy scenarios, where maintaining a precise trajectory is challenging but critical for mission success.

However, it's notable that while MPC generally shows greater resilience and precision, it exhibits a higher susceptibility to disturbances in trajectory tracking, with a greater deviation from the setpoint compared to the PID controller. This observation could indicate that while the predictive nature of MPC provides significant advantages in stability and precision, it may also lead to overcompensation or heightened sensitivity to changes in environmental conditions. This analysis highlights the nuanced performance of MPC and PID in complex control tasks, suggesting that the choice of controller can be crucially informed by the specific requirements and conditions of the application, balancing between precision, resilience, and adaptability to disturbances.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

This research has delved into the complexities of creating and applying advanced control strategy for UAV quadcopters, with a special emphasis on operating within intricate and challenging terrains such as those found in mangrove forests. The culmination of this research is the formulation of a mathematical model which incorporate external disturbances and uncertainties, firmly rooted in the Newton-Euler framework. This model has been instrumental in unravelling the intricate dynamics that govern the interaction between UAV quadcopters and the myriad environmental variables present in such demanding landscapes.

A cornerstone of this research has been the development and validation of a robust mathematical model, grounded in the Newton-Euler framework, to accurately capture the complex dynamics of UAV quadcopters operating in challenging environments. This model incorporates external disturbances and uncertainties, providing a critical foundation for understanding UAV-environment interactions and serving as a basis for designing advanced control strategies.

Building upon this foundation, the study introduced a cascade MPC algorithm, which was rigorously tested and validated through comprehensive simulations and experimental evaluations. The integration of the MPC algorithm with the mathematical model highlights the synergy between predictive control and dynamic modelling, enabling precise and adaptive UAV operations. This research underscores the potential of advanced

control systems to address the complexities of navigating unpredictable terrains, such as mangrove forests, while maintaining operational stability and efficiency.

By seamlessly combining theoretical modelling with practical validation, this thesis bridges the gap between academic research and real-world UAV applications. The insights gained from this work not only enhance the understanding of UAV behaviour in adverse conditions but also lay the groundwork for future innovations in autonomous systems, particularly in scenarios demanding high levels of adaptability and reliability.

The practical effectiveness of the proposed control algorithm has been tested and validated through a series of experimental evaluations and comprehensive simulation studies. In these tests, the MPC controller consistently outperformed the PID controller across various parameters: achieving faster rise times by approximately 15%, reducing overshoot by about 10%, and improving settling times by up to 50%. Furthermore, the MPC controller demonstrated better response and stability with less oscillatory behaviour. These practical tests have not only demonstrated the algorithm's superior performance but have also provided a platform for juxtaposing theoretical models with real-world scenarios. This synergy between theoretical insights and empirical data has been invaluable, shedding light on the real-world applicability and constraints of UAV operations in hostile environments.

Moreover, the research journey has unveiled a rich tapestry of insights into the nuanced behaviours and potential limitations of UAVs when faced with adverse environmental conditions. By seamlessly blending sophisticated theoretical models with hands-on experimental data, this thesis has successfully bridged the often-wide chasm that separates academic research from tangible, real-world applications. This holistic approach

has not only validated the effectiveness of the proposed MPC algorithm but has also paved the way for future innovations in UAV technology, particularly in the realm of autonomous operations in challenging and unpredictable environments.

5.2 Recommendations

Building upon the foundational work presented in this thesis, several recommendations have been proposed to guide future research and practical advancements in UAV technology, particularly in manoeuvring through predefined 3D trajectories, obstacle avoidance, and swarm UAV control:

1. Advanced Control Algorithms for 3D Manoeuvring and Obstacle Avoidance

Future research should focus on the development of control algorithms capable of precise 3D trajectory tracking and real-time obstacle avoidance. Enhancing the computational efficiency of MPC algorithms while incorporating adaptive techniques such as machine learning could enable UAVs to anticipate and react to environmental changes dynamically. Machine learning models trained on diverse datasets could further improve path planning and decision-making in cluttered 3D environments.

2. Integration of Advanced Sensors for Enhanced Situational Awareness

The adoption of advanced sensory technologies is critical for improving UAV capabilities in 3D navigation. Incorporating sensors such as LiDAR, hyperspectral imaging, and advanced radar systems could enable UAVs to create detailed 3D maps of their surroundings, facilitating accurate trajectory following and collision avoidance. The fusion of sensory data with control algorithms would ensure UAVs maintain high situational awareness, even in complex and dynamic environments.

3. Swarm UAV Control for Coordinated 3D Operations

Research into swarm UAV systems holds immense potential for collaborative missions. Developing algorithms for coordinated manoeuvring in 3D space, with a focus on obstacle avoidance and maintaining formation integrity, could unlock capabilities in areas like disaster relief, large-scale environmental monitoring, and search-and-rescue operations. Swarm control strategies should emphasize decentralized decision-making, ensuring fault tolerance and scalability.

By addressing these recommendations, future research can drive significant advancements in UAV technology, unlocking new capabilities in 3D navigation, obstacle avoidance, and swarm coordination. The integration of intelligent control systems, as demonstrated in this thesis, is essential for navigating complex and dynamic environments. These efforts will contribute to innovative UAV applications in scientific exploration, environmental monitoring, and humanitarian operations.

REFERENCE

- Ahmad, F., Kumar, P., & Patil, P. P. (2018). Modeling and simulation of a quadcopter with altitude and attitude control. *Nonlinear Studies*, 25(2), 287–299.
- Abdelmaksoud, S. I., Mailah, M., & Abdallah, A. M. (2021). Improving disturbance rejection capability for a quadcopter UAV system using self-regulating fuzzy PID controller. In *2020 international conference on computer, control, electrical, and electronics engineering (ICCCEE)* (pp. 1-6). IEEE.
- Alameri, S., Lazic, D., & Ristanovic, M. (2018). A Comparative study of PID, PID with tracking, and FPID controller for missile canard with an optimized genetic tuning method using Simscape modelling. *Tehnicki Vjesnik - Technical Gazette*, 25(Supplement 2). <https://doi.org/10.17559/tv-20171207130458>
- Alawad, W., Halima, N. B., & Aziz, L. (2023). An unmanned aerial vehicle (UAV) system for disaster and crisis management in smart cities. *Electronics*, 12(4), 1051.
- Alexis, K., Nikolakopoulos, G., & Tzes, A. (2011). Switching model predictive attitude control for a quadrotor helicopter subject to atmospheric disturbances. *Control Engineering Practice*, 19(10), 1195–1207. <https://doi.org/10.1016/j.conengprac.2011.06.010>
- Alexis, K., Tzes, A., & Nikolakopoulos, G. (2012). Model predictive quadrotor control: attitude, altitude and position experimental studies. *IET Control Theory & Applications*, 6(12), 1812–1827. <https://doi.org/10.1049/iet-cta.2011.0348>
- Alifah, Z. N. (2022). Control design of quadcopter using linear quadratic gaussian (LQG). In *2022 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM)* (pp. 192-196). IEEE.

- Arrieta, O., Campos, D., Rico-Azagra, J., Gil-Martínez, M., Rojas, J. D., & Vilanova, R. (2023). Model-based optimization approach for PID control of pitch–roll UAV orientation. *Mathematics*, *11*(15), 3390.
- ÅströmK. J., & WittenmarkB. (2011). *Computer-controlled systems: theory and design*. Dover Publications.
- Ayad, R., Nouibat, W., Zareb, M., & Bestaoui Sebanne, Y. (2019). Full control of quadrotor aerial robot using fractional-order FOPID. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, *43*, 349-360.
- Berbra, C., Leseq, S., & Martinez, J. J. (2008). A multi-observer switching strategy for fault-tolerant control of a quadrotor helicopter. In *2008 16th Mediterranean Conference on Control and Automation* (pp. 1094-1099). IEEE.
- Bohn, C., & Atherton, D. P. (1995). An analysis package comparing PID anti-windup strategies. *IEEE Control Systems Magazine*, *15*(2), 34-40.
- Borbolla-Burillo, P., Sotelo, D., Frye, M., Garza-Castañón, L. E., Juárez-Moreno, L., & Sotelo, C. (2024). Design and real-time implementation of a cascaded model predictive control architecture for unmanned aerial vehicles. *Mathematics*, *12*(5), 739.
- Boudjedir, H., Bouhali, O., & Rizoug, N. (2014). Adaptive neural network control based on neural observer for quadrotor unmanned aerial vehicle. *Advanced Robotics*, *28*(17), 1151–1164. <https://doi.org/10.1080/01691864.2014.913498>
- Bouffard, P., Aswani, A., & Tomlin, C. (2012). Learning-based model predictive control on a quadrotor: onboard implementation and experimental results. In *2012 IEEE International Conference on Robotics and Automation* (pp. 279-284). IEEE

- Bouabdallah, S., Murrieri, P., & Siegwart, R. (2005). Towards autonomous indoor micro-VTOL. *Autonomous Robots*, 18, 171-183.
- Bouabdallah, Samir. *Design and control of quadrotors with application to autonomous flying*. No. 3727. EPFL, 2005.
- Bowen, B. M., Baars, J. A., & Stone, G. L. (2000). Nocturnal wind direction shear and its potential impact on pollutant transport. *Journal of Applied Meteorology and Climatology*, 39(3), 437-445.
- Bresciani, T. (2008). Modelling, identification, and control of a quadrotor helicopter. *MSc Thesis*.
- Brescianini M.D, Dario; Hehn, Nonlinear quadrocopter attitude control technical report.<https://www.research-collection.ethz.ch/bitstream/handle/20.500.11850/154099/eth-7387-01.pdf>, 2013.
- Castillo, P., Lozano, R., & Dzul, A. E. (2005). *Modelling and control of mini-flying machines*. Springer Science & Business Media.
- Cavanini, L., Ippoliti, G., & Camacho, E. F. (2021). Model predictive control for a linear parameter varying model of an UAV. *Journal of Intelligent & Robotic Systems*, 101(3), 57.
- Chunyu, W., & Litong, H. (2024). Study on Modeling and control for a novel active suspension. *International Journal of Automotive Technology*, 1-8.
- Chen, C., & Tan, J. (2023). Path following for UAV using nonlinear model predictive control. *Journal of Physics Conference Series*, 2530(1), 012021–012021. <https://doi.org/10.1088/1742-6596/2530/1/012021>

- Choi, J. W., & Lee, S. C. (2009). Anti windup strategy for PI-type speed controller. *IEEE Transactions on Industrial Electronics*, 56(6), 2039-2046.
- Das A, A., Subbarao, K., & Lewis, F. (2009). Dynamic inversion with zero-dynamics stabilisation for quadrotor control. *IET control theory & applications*, 3(3), 303-314.
- Djizi, H., Zahzouh, Z., & Bouzaouit, A. (2023). Quadcopter prototype stability assessment with Pid controller and euler-lagrange Approach. *The Scientific Bulletin of Electrical Engineering Faculty*, 23(1), 15-20.
- Dyer, E. (2018). *Modelling and Control of an Omni-directional UAV. Ms Thesis*
- El Hamidi, K., Mjahed, M., El Kari, A., & Ayad, H. (2019). Neural network and fuzzy-logic-based self-tuning PID control for quadcopter path tracking. *Studies in Informatics and Control*, 28(4), 401-412.
- Espinoza, J., Hakim, N., Tan, D., Wilson, T., Bingi, K., Khan, E., & Masrura, S. (2023). Fractional-order PID control of quadrotor drone. In *2023 Innovations in Power and Advanced Computing Technologies (i-PACT)* (pp. 1-6). IEEE.
- Faessler, M., Fontana, F., Forster, C., Mueggler, E., Pizzoli, M., & Scaramuzza, D. (2016). Autonomous, vision-based flight and live dense 3D mapping with a quadrotor micro aerial vehicle. *Journal of Field Robotics*, 33(4), 431-450.
- Fernando, H. C. T. E., De Silva, A. T. A., De Zoysa, M. D. C., Dilshan, K. A. D. C., & Munasinghe, S. R. (2013). Modelling, simulation and implementation of a quadrotor UAV. In *2013 IEEE 8th International conference on industrial and information systems* (pp. 207-212). IEEE.

- Fessi, R., & Bouallègue, S. (2019). LQG controller design for a quadrotor UAV based on particle swarm optimisation. *International Journal of Automation and Control*, 13(5), 569-594.
- Gaspari, F., Ioli, F., Barbieri, F., Belcore, E., & Pinto, L. (2022). Integration of UAV-lidar and UAV-photogrammetry for infrastructure monitoring and bridge assessment. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43, 995-1002.
- Gui, Y. (2023). Comparison of several PID controls for quad-rotor UAV under wind disturbance. *Applied and Computational Engineering*, 5(1), 139–145. <https://doi.org/10.54254/2755-2721/5/20230549>
- Haidong, Z., Qiuyu, C., Chongfa, Z., Yajie, D., Yufeng, M., & Jun, Y. (2021). Stability research of quadcopter UAV under unstable wind. In *2021 IEEE 7th International Conference on Control Science and Systems Engineering (ICCSSE)* (pp. 114-118). IEEE.
- Hochard, J. P., Hamilton, S., & Barbier, E. B. (2019). Mangroves shelter coastal economic activity from cyclones. *Proceedings of the National Academy of Sciences*, 116(25), 12232-12237.
- Irizarry, J., & Costa, D. B. (2016). Exploratory study of potential applications of unmanned aerial systems for construction management tasks. *Journal of Management in Engineering*, 32(3), 05016001.
- Iskandarani, M., Hafez, A. T., Givigi, S. N., Beaulieu, A., & Rabbath, C. A. (2013). Using multiple quadrotor aircraft and linear model predictive control for the encirclement

- of a target. In *2013 IEEE International Systems Conference (SysCon)* (pp. 620-627). IEEE.
- Jardin, M. R., & Mueller, E. R. (2007). Optimized measurements of UAV mass moment of inertia with a bifilar pendulum. In *Proceeding of AIAA Guidance, Navigation and Control Conference and Exhibit, Hilton Head, SC, USA*.
- Jiang, F., Pourpanah, F., & Hao, Q. (2019). Design, implementation, and evaluation of a neural-network-based quadcopter UAV system. *IEEE Transactions on Industrial Electronics*, 67(3), 2076-2085.
- Jiang, B., Li, B., Zhou, W., Lo, L. Y., Chen, C. K., & Wen, C. Y. (2022). Neural network-based model predictive control for a quadrotor UAV. *Aerospace*, 9(8), 460.
- Jing, Y., Mirza, A., Sipahi, R., & Martinez-Lorenzo, J. (2023). Sliding mode controller with disturbance observer for quadcopters; experiments with dynamic disturbances and in turbulent indoor space. *Drones*, 7(5), 328.
- Kathiresan, K., & Bingham, B. L. (2001). Biology of mangroves and mangrove ecosystems.
- K. Ogata, *Modern Control Engineering*, 5th ed. Upper Saddle River, NJ: Prentice Hall, 2010. pp. 567–595.
- Kumar, A., Gupta, S., Maheshwari, S., Shah, S. V., & Tripathy, N. S. (2023). Trajectory tracking by quadcopter using MPC in presence of obstacles and external disturbances. In *Proceedings of the 2023 6th International Conference on Advances in Robotics* (pp. 1-6).

- Khosravian, E., & Maghsoudi, H. (2019). Design of an intelligent controller for station keeping, attitude control, and path tracking of a quadrotor using recursive neural networks. *International Journal of Engineering*, 32(5), 747-758.
- Lammers, D. T., Williams, J. M., Conner, J. R., Baird, E., Rokayak, O., McClellan, J. M., Bingham, J. R., Betzold, R., & Eckert, M. J. (2023). Airborne! UAV delivery of blood products and medical logistics for combat zones. *Transfusion*. <https://doi.org/10.1111/trf.17329>
- Lee, D., Burg, T. C., Dawson, D. M., Shu, D., Xian, B., & Tatlicioglu, E. (2009). Robust tracking control of an underactuated quadrotor aerial-robot based on a parametric uncertain model. In *2009 IEEE international conference on systems, man and cybernetics* (pp. 3187-3192). IEEE.
- Liao, J., & Song, X. (2023). Mathematical modelling and model predictive controller design of a quad tiltrotor UAV. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 237(18), 4163-4174.
- Liu, Y., Ma, Y., Wu, B., & Liu, Y. (2022). Design and experiment of UAV variable spray control system based on RBF-PID. In *The International Conference on Image, Vision and Intelligent Systems (ICIVIS 2021)* (pp. 945-954). Singapore: Springer Nature Singapore.
- Luis, C., & Ny, J. L. (2016). Design of a trajectory tracking controller for a nano-quadcopter. *arXiv preprint arXiv:1608.05786*.

- Mahony, R., Kumar, V., & Corke, P. (2012). Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE robotics & automation magazine*, 19(3), 20-32.
- Mamdouh, M., Ibrahim, M., Sarhan, A., & Ashry, M. (2023). Path following controller and obstacles avoidance using nonlinear model predictive control. In *2023 International Telecommunications Conference (ITC-Egypt)* (pp. 701-706). IEEE.
- Martínez-León, A. S., Jatsun, S., & Emelyanova, O. (2024). Investigation of wind effects on UAV adaptive PID based MPC control system. *Enfoque UTE*, 15(2), 36-47.
- Martini, S., Sönmez, S., Rizzo, A., Stefanovic, M., Rutherford, M. J., & Valavanis, K. P. (2022). Euler-lagrange modeling and control of quadrotor UAV with aerodynamic compensation. In *2022 International Conference on Unmanned Aircraft Systems (ICUAS)* (pp. 369-377). IEEE.
- Mellinger, D., Lindsey, Q., Shomin, M., & Kumar, V. (2011). Design, modeling, estimation and control for aerial grasping and manipulation. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 2668-2673). IEEE.
- Miranda-Moya, A., Castañeda, H., & Wang, H. (2023). Fixed-time extended observer-based adaptive sliding mode control for a quadrotor uav under severe turbulent wind. *Drones*, 7(12), 700.
- Miranda-Moya, A., Castaneda, H., & Wang, H. (2024). Fixed-time differentiator-based adaptive non-singular fast terminal image-based visual servoing for a quadrotor UAV Subject to turbulent wind. *IEEE Transactions on Aerospace and Electronic Systems*. vol. 60, no. 3, pp. 2807-2818, June 2024.

- Mohan, M., Richardson, G., Gopan, G., Aghai, M. M., Bajaj, S., Galgamuwa, G. A. P., Vastaranta, M., Arachchige, P. S. P., Amorós, L., Corte, A. P. D., de-Miguel, S., Leite, R. V., Kganyago, M., Broadbent, E. N., Doaemo, W., Shorab, M. A. B., & Cardil, A. (2021). UAV-supported forest regeneration: current trends, challenges and implications. *Remote Sensing*, 13(13), 2596.
- Mohammadi, N., Tayefi, M., & Zhu, M. (2023). Vertical take-off and hover to cruise transition for a hybrid UAV using model predictive controller and MPC allocation. *Aircraft Engineering and Aerospace Technology*, 95(10), 1642-1650.
- Mohammed, M. J., Rashid, M. T., & Ali, A. A. (2014). Design optimal PID controller for quad rotor system. *International Journal of Computer Applications*, 106(3), 15-20.
- Muliadi, J., & Kusumoputro, B. (2018). Neural network control system of UAV altitude dynamics and its comparison with the PID control system. *Journal of Advanced Transportation*, 2018, 1-18.
- Naidoo, Y., Stopforth, R., & Bright, G. (2011). Quadrotor unmanned aerial vehicle helicopter modelling & control. *International Journal of Advanced Robotic Systems*, 8(4), 45.
- Nguyen, A. T., Xuan-Mung, N., & Hong, S. K. (2019). Quadcopter adaptive trajectory tracking control: A new approach via backstepping technique. *Applied Sciences*, 9(18), 3873.
- Nobahari, H., & Sharifi, A. (2019). Wind compensation in trajectory tracking of a fixed wing uav using a nonlinear model predictive controller based on the particle swarm optimization. In *2019 7th International Conference on Robotics and Mechatronics (ICRoM)* (pp. 138-143). IEEE.

- Noormohammadi-Asl, A., Esrafilian, O., Arzati, M. A., & Taghirad, H. D. (2020). System identification and H_∞ -based control of quadrotor attitude. *Mechanical Systems and Signal Processing*, 135, 106358.
- Novak, D., & Tebbani, S. (2023). Nonlinear MPC for the multi-UAV system with allocated priority for collision avoidance. In *2023 31st Mediterranean Conference on Control and Automation (MED)* (pp. 7-12). IEEE.
- Nugroho, H. A., Cahyadi, A. I., & Ardiyanto, I. (2023). Trajectory tracking control of uav bicopter using linear quadratic gaussian. *arXiv preprint arXiv:2309.08226*.
- Pan, M., Chen, C., Yin, X., & Huang, Z. (2021). UAV-aided emergency environmental monitoring in infrastructure-less areas: LoRa mesh networking approach. *IEEE Internet of Things Journal*, 9(4), 2918-2932.
- Patel, K., & Barve, J. (2014). Modeling, simulation and control study for the quad-copter UAV. In *2014 9th International Conference on Industrial and Information Systems (ICIIS)* (pp. 1-6). IEEE.
- Prach, A., & Kayacan, E. (2018). An MPC-based position controller for a tiltrotor tricopter VTOL UAV. *Optimal Control Applications and Methods*, 39(1), 343-356.
- Peixoto, A. J., Serrantola, W. G., & Lizarralde, F. (2024). Dynamic smooth sliding control applied to UAV trajectory tracking. *IEEE Access*, 12, 44306–44324. <https://doi.org/10.1109/access.2024.3380918>
- Raffo, G. V., Ortega, M. G., & Rubio, F. R. (2010). An integral predictive/nonlinear H_∞ control structure for a quadrotor helicopter. *Automatica*, 46(1), 29–39. <https://doi.org/10.1016/j.automatica.2009.10.018>

- Rinaldi, M., Primatesta, S., & Guglieri, G. (2023). A comparative study for control of quadrotor UAVs. *Applied Sciences*, 13(6), 3464.
- Saccani, D., Cecchin, L., & Fagiano, L. (2022). Multitrajectory model predictive control for safe UAV navigation in an unknown environment. *IEEE Transactions on Control Systems Technology*, 31(5), 1982-1997.
- Salma, N. M., & Osman, K. (2020). Modelling and PID control system integration for quadcopter DJIF450 attitude stabilization. *Indonesian Journal of Electrical Engineering and Computer Science*, 19(3), 1235-1244.
- Say, M. F. Q., Sybingco, E., Bandala, A. A., Vicerra, R. R. P., & Chua, A. Y. (2021). A genetic algorithm approach to PID tuning of a quadcopter UAV model. In *2021 IEEE/SICE International Symposium on System Integration (SII)* (pp. 675-678). IEEE.
- Shao, X., Sun, G., Yao, W., Liu, J., & Wu, L. (2022). Adaptive sliding mode control for quadrotor UAVs with input saturation. *IEEE/ASME Transactions on Mechatronics*, 27(3), 1498–1509. <https://doi.org/10.1109/TMECH.2021.3094575>
- Sharifi, F., Mirzaei, M., Gordon, B. W., & Zhang, Y. (2010). Fault tolerant control of a quadrotor UAV using sliding mode control. *2010 Conference on Control and Fault-Tolerant Systems (SysTol)*. <https://doi.org/10.1109/systol.2010.5675979>
- Shen, S., Michael, N., & Kumar, V. (2011). Autonomous multi-floor indoor navigation with a computationally constrained MAV. In *2011 IEEE International Conference on Robotics and Automation* (pp. 20-25). IEEE.
- Shi, Y., Thomasson, J.A., Murray, S., Pugh, N.A., Rooney, W., Shafian, S., Rajan, N., Rouze, G., Morgan, C.L., Neely, H., Rana, A., Bagavathiannan, M., Henrickson,

- J.V., Bowden, E., Valasek, J., Olsenholler, J.A., Bishop, M.P., Sheridan, R.D., Putman, E.B., Popescu, S.C., Burks, T., Cope, D.A., Ibrahim, A., McCutchen, B.F., Baltensperger, D., Avant, R.V., Vidrine, M., & Yang, C. (2016). Unmanned aerial vehicles for high-throughput phenotyping and agronomic research. *PLoS ONE*, *11*.
- Singh, B. K., & Kumar, A. (2023). Attitude and position control with minimum snap trajectory planning for quadrotor UAV. *International Journal of Dynamics and Control*, *11*(5), 2342-2353.
- Singh, B. K., & Kumar, A. (2023). Model predictive control using LPV approach for trajectory tracking of quadrotor UAV with external disturbances. *Aircraft Engineering and Aerospace Technology*, *95*(4), 607-618.
- Singh, B. K., Kumar, A., & Kumar, G. V. (2023). Comprehensive review of various control strategies for quadrotor unmanned aerial vehicles. *FME Transactions*, *51*(3), 298-317.
- Sridhar, S., Gupta, G., Kumar, R., Kumar, M., & Cohen, K. (2019). Tilt-rotor quadcopter explored: hardware-based dynamics, smart sliding mode controller, attitude hold & wind disturbance scenarios. In *2019 American Control Conference (ACC)* (pp. 2005-2010). IEEE.
- Srivastava, A., Indu, S., & Sharma, R. (2024). Design and flight testing of LQRi attitude control for quadcopter UAV. *arXiv preprint arXiv:2404.12261*.
- Suraj, S. K., George, V. I., Thirunavukkarasu, I., & Valsa Paul, T. (2021). Design and simulation of a sliding mode-based control design for a quadrotor UAV. In *Smart Sensors Measurements and Instrumentation: Select Proceedings of CISCON 2020* (pp. 495-507). Springer Singapore.

- Svacha, J., Paulos, J., Loianno, G., & Kumar, V. (2020). Imu-based inertia estimation for a quadrotor using newton-euler dynamics. *IEEE Robotics and Automation Letters*, 5(3), 3861-3867.
- Tabassum, A., Vuppala, R. K., Bai, H., & Kara, K. (2021). Variance reduction of quadcopter trajectory tracking in turbulent wind. *IFAC-PapersOnLine*, 54(20), 102-107.
- Tongue, B. (2002). Principles of Vibration. Oxford University Press. ISBN 9780195142464. Available at: <https://books.google.co.za/books?id=wAGqXVImUjYC>
- Velagić, J., Osmić, N., Klovo, V., & Lačević, H. (2022). Design of LQR controller for 3D trajectory tracking of octocopter unmanned aerial vehicle. In *2022 8th International Conference on Control, Decision and Information Technologies (CoDIT)* (Vol. 1, pp. 63-68). IEEE.
- Walid, M., Slaheddine, N., Mohamed, A., & Lamjed, B. (2018). Modelling, identification and control of a quadrotor UAV. In *2018 15th International Multi-Conference on Systems, Signals & Devices (SSD)* (pp. 1017-1022). IEEE.
- Wang, B. H., Wang, D. B., Ali, Z. A., Ting Ting, B., & Wang, H. (2019). An overview of various kinds of wind effects on unmanned aerial vehicle. *Measurement and Control*, 52(7-8), 731-739.
- Wang, P., Man, Z., Cao, Z., Zheng, J., & Zhao, Y. (2016). Dynamics modelling and linear control of quadcopter. In *2016 International Conference on Advanced Mechatronic Systems (ICAMechS)* (pp. 498-503). IEEE.

- Wang, X., & Mao, X. (2024). Non-overshooting sliding mode for UAV control. *arXiv preprint arXiv:2405.01087*.
- Yuan, X. (2024). Achieving stable trajectory tracking in complex environments using an adaptive PID control strategy-based quadcopter drone. *Applied and Computational Engineering*, 34, 212-225.
- Zhao, L., & Zhu, M. (2023). MS-YOLOv7: YOLOv7 based on multi-scale for object detection on UAV aerial photography. *Drones*, 7(3), 188.
- Z. Zhang, X. Pan, Z. Zhang, and Z. Zhou, Research on autonomous mobile landing method for quadcopter UAV based on fuzzy PID, *2023 3rd International Conference on Electrical Engineering and Control Science (IC2ECS)*, Hangzhou, China, 2023, pp. 751-754, doi: 10.1109/IC2ECS60824.2023.10493837.
- Zhang, G., He, Y., Dai, B., Gu, F., Han, J., & Liu, G. (2019). Robust control of an aerial manipulator based on a variable inertia parameters model. *IEEE Transactions on Industrial Electronics*, 67(11), 9515-9525.
- Zhi, Y., Li, G., Song, Q., Yu, K., & Zhang, J. (2017). Flight control law of unmanned aerial vehicles based on robust servo linear quadratic regulator and kalman filtering. *International Journal of Advanced Robotic Systems*, 14(1), 1729881416686952.

APPENDICES

Appendix A

Both wind turbulence and gusts significantly impact UAV operations, but the sudden and extreme nature of gusts often presents the greater challenge, demanding careful consideration in planning and executing UAV missions in environments like mangrove forests.

To better understand the nature of Wind in mangrove forest, we conducted a one-month measurement of Wind Speed and Direction in the nearest Mangrove Forest situated in Kuching Wetlands National Park as shown in Figure B1.



Figure B1: Kuching Wetlands National Park

The results gotten after one month period of measurement is shown below in figures (B2 and B3):

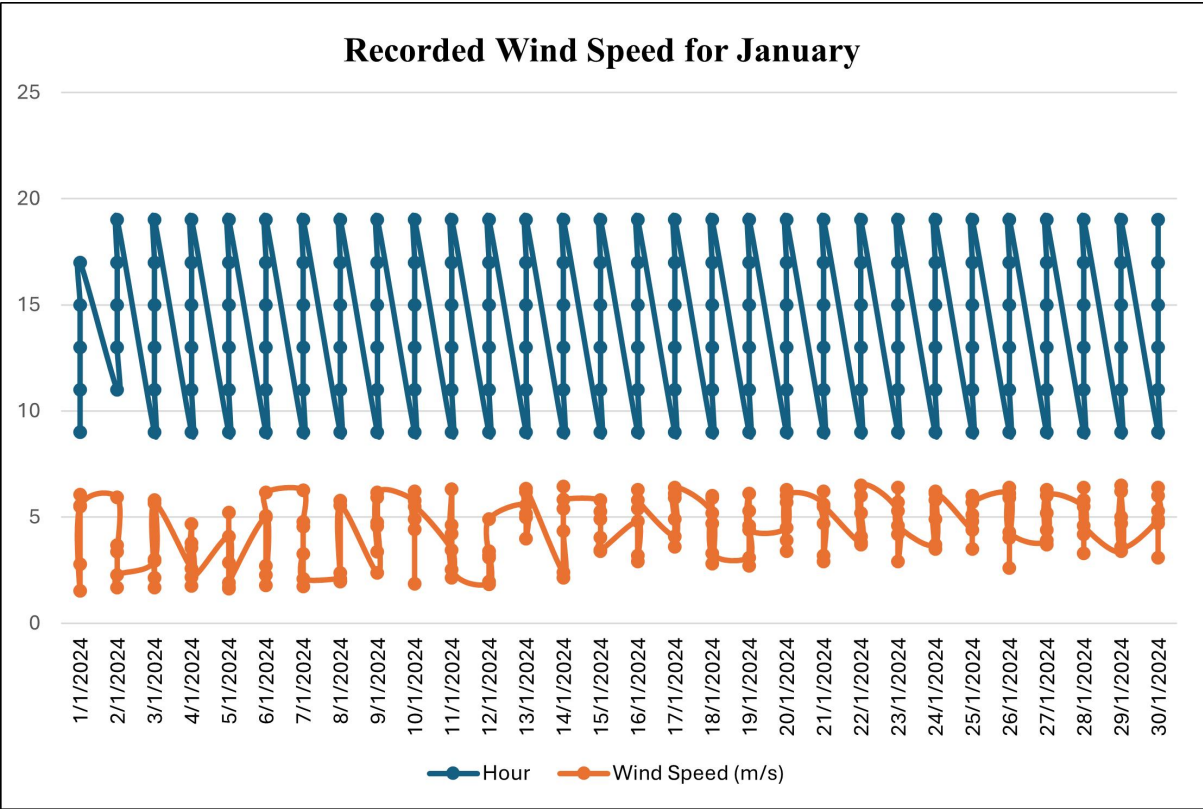


Figure B2: Recorded Wind Speed for January

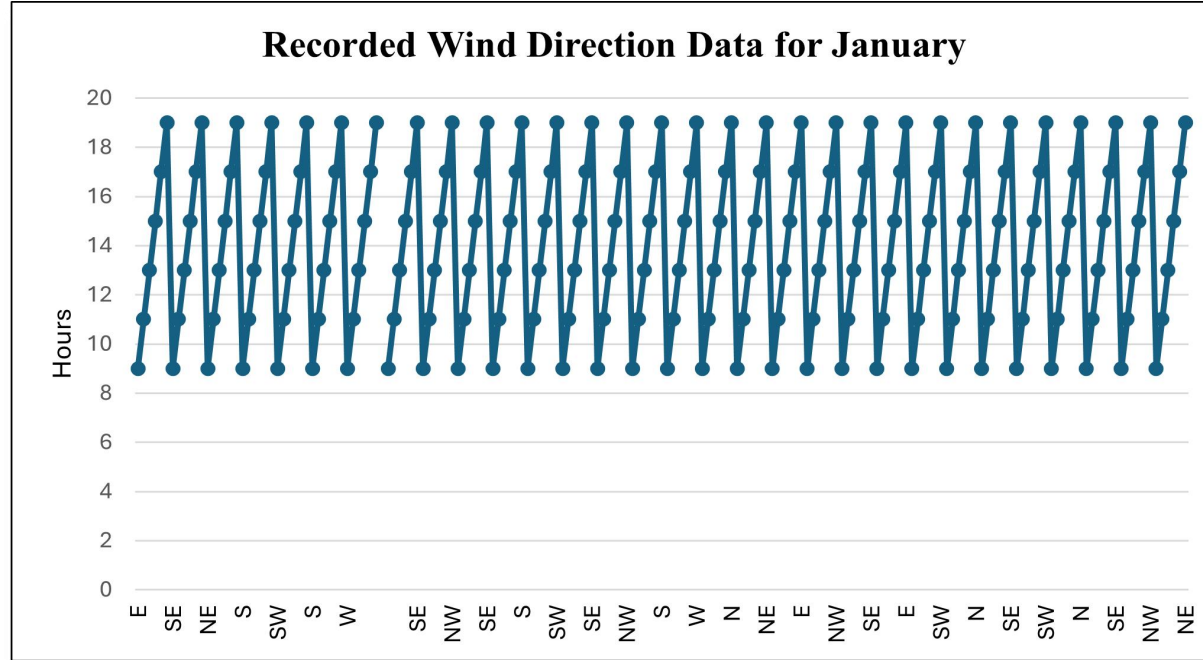


Figure B3: Recorded Wind Direction for January

Figure B2 shows wind speed fluctuations ranging from 2 m/s to 6 m/s throughout the month. The consistent pattern of peaks and troughs suggests a daily cycle with moderate variations in wind speed, likely influenced by weather conditions. The upper series indicates that measurements were taken consistently at the same time each day. In contrast, Figure B3 illustrates frequent shifts between cardinal and intercardinal directions (e.g., E, NE, NW, SE), reflecting significant variability in wind movement. These directional changes suggest dynamic weather systems, causing shifting wind patterns over time. Together, the figures highlight stable wind speeds but unpredictable directional changes.

The measurement was carried out using Vane anemometer placed at the top of the tree (approximately 10.2 m), also recognized as wind vanes or aerovanes, serve as critical instruments in fields such as meteorology, aviation, environmental science, and various industrial sectors for measuring wind speed and direction. These devices comprise a rotating element like a propeller or cups to gauge wind speed, alongside a tail vane that aligns with wind direction, thus enabling the simultaneous assessment of both metrics. The core of a vane anemometer includes a horizontal shaft with a propeller or rotor at one end and a tail vane at the other, ensuring the propeller faces the wind. Wind speed is determined by the rotation speed of the propeller, measured either mechanically through gears and a dial or electronically via sensors. The tail vane's alignment with the wind is used to deduce wind direction, typically through electronic means such as potentiometers or optical encoders, which translate the vane's orientation into an electrical signal.

Measurement Inaccuracy

The following are the most common reasons why an erroneous measurement is produced:

- There is sensor noise, resulting in erroneous measurements of the collected data.
- The sensor's efficiency deteriorates, resulting in inaccessible states being reached. Observers are commonly used to tackle this problem, as detailed in the section on Berbra et al., (2008).

Appendix B

1. Testo 416 - Digital 16 mm vane anemometer

The compact 16 mm vane anemometer testo 416 with telescopic probe and App connection is ideal for simple, fast and precise determination of flow velocity and volume flow in ventilation ducts of air conditioning and ventilation systems. The connection to the testo Smart App enables configuration, display (also as second screen) and storage as well as documentation of the measured values directly in the App.



Figure B4: vane anemometer testo 416

2. Technical data

Velocity - Vane

Measuring range	0.6 to 40 m/s
Accuracy	$\pm(0.2 \text{ m/s} + 1.0 \% \text{ of mv})$
Resolution	0.1 m/s

Figure B5: Velocity Vane Data

General technical data

Weight	323 g
Dimensions	135 x 60 x 28 mm
Operating temperature	Probe: -10 to +70 °C -10 to +50 °C
Product-/housing material	ABS + PC / TPE
Cable length	1.5 m
Diameter probe shaft	16 mm
Protection class	IP20
Length probe	329 mm
Battery life	60 h
Battery type	3x AA
Storage temperature	-10 to +50 °C

Figure B6: General Technical Data

Appendix C

Hardware Technical Data and Calibration

1.1 Holybro Pixhawk 6C

Pixhawk 6C[®] is the latest update to the successful family of Pixhawk[®] flight controllers designed and made in collaboration with Holybro[®] and the PX4 team.

It is equipped with a high performance H7 Processor, and comes with IMU redundancy, temperature-controlled IMU board, and cost-effective design, delivering incredible performance and reliability.



Figure B7: Pixhawk 6C

- Technical Specification
- Processors & Sensors

- FMU Processor: STM32H743
 - 32 Bit Arm® Cortex®-M7, 480MHz, 2MB memory, 1MB SRAM
- IO Processor: STM32F103
 - 32 Bit Arm® Cortex®-M3, 72MHz, 64KB SRAM
- On-board sensors
 - Accel/Gyro: ICM-42688-P
 - Accel/Gyro: BMI055
 - Mag: IST8310
 - Barometer: MS5611
- Electrical data
- Voltage Ratings:
 - Max input voltage: 6V
 - USB Power Input: 4.75~5.25V
 - Servo Rail Input: 0~36V
- Current Ratings:
 - TELEM1 Max output current limiter: 1.5A
 - All other port combined output current limiter: 1.5A

- Mechanical data
- Dimensions: 84.8 * 44 * 12.4 mm
- Weight: 59.3g
- Interfaces
- 16- PWM servo outputs (8 from IO, 8 from FMU)
- 3 general purpose serial ports
 - TELEM1 - Full flow control, separate 1.5A current limit
 - TELEM2 - Full flow control
 - TELEM3
- 2 GPS ports
 - GPS1 - Full GPS port (GPS plus safety switch)
 - GPS2 - Basic GPS port
- 1 I2C port
 - Supports dedicated I2C calibration EEPROM located on sensor module
- 2 CAN Buses
 - CAN Bus has individual silent controls or ESC RX-MUX control
- 2 Debug ports:

- FMU Debug
 - I/O Debug
- Dedicated R/C input for Spektrum / DSM and S.BUS, CPPM, analog / PWM RSSI
- Dedicated S.BUS output
- 2 Power input ports (Analog)
- Other Characteristics: Operating & storage temperature: -40 ~ 85°C

1.2 Accelerometer Calibration

The accelerometer must be calibrated on first use or if the flight controller orientation is changed. Otherwise, it should not need to recalibrate.

Performing the Calibration

QGroundControl will guide you to place and hold your vehicle in several orientations (you will be prompted when to move between positions). The calibration steps are:

- Start *QGroundControl* and connect the vehicle.
- Select "**Q**" icon > **Vehicle Setup** > **Sensors** (sidebar) to open *Sensor Setup*.
- Click the **Accelerometer** sensor button.

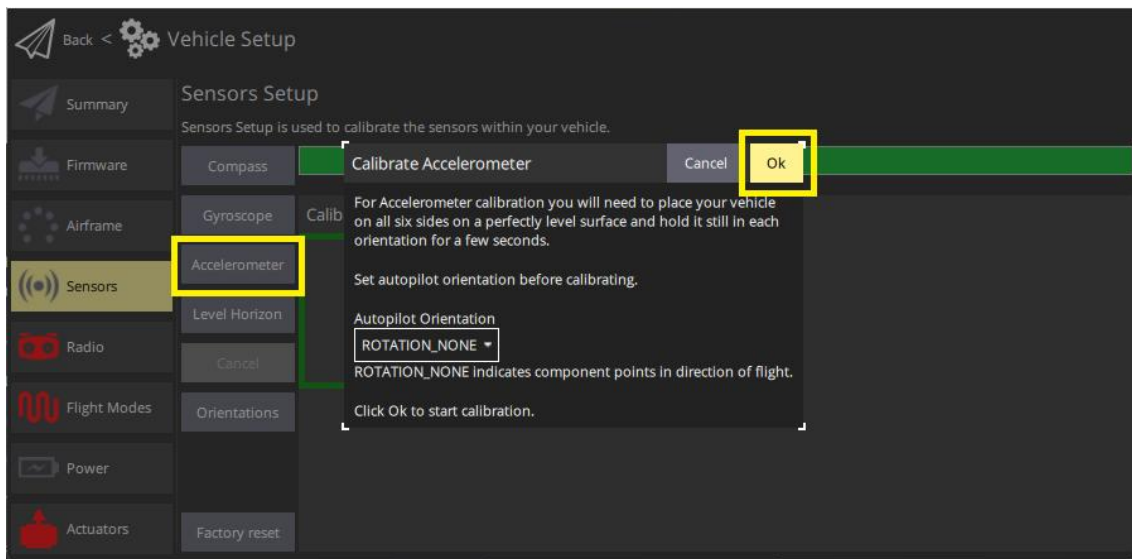


Figure B8: Accelerometer Calibration Starting Process

Position the vehicle as guided by the *images* on the screen. Once prompted (the orientation-image turns yellow) hold the vehicle still. Once the calibration is complete for the current orientation the associated image on the screen will turn green.

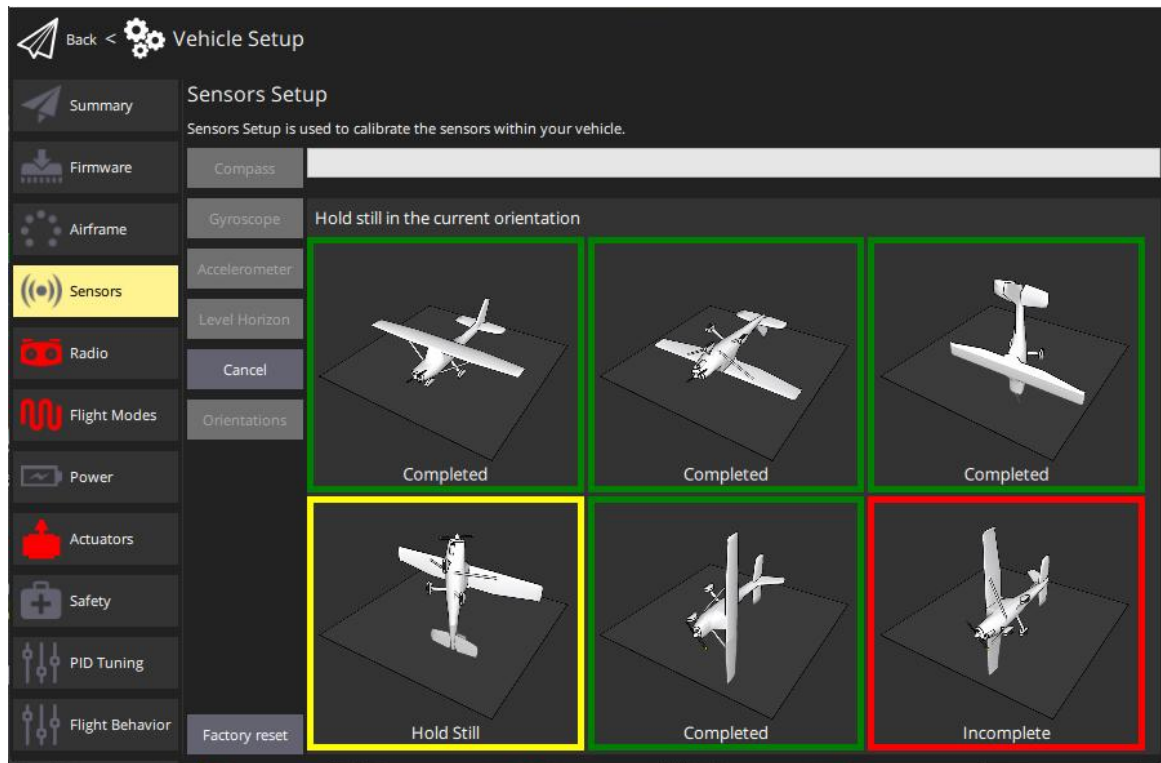


Figure B9: Accelerometer Calibration Ending Process

Repeat the calibration process for all vehicle orientations.

Once you've calibrated the vehicle in all the positions *QGroundControl* will display *Calibration complete* (all orientation images will be displayed in green, and the progress bar will fill completely). You can then proceed to the next sensor.

1.3 Gyroscope Calibration

Performing the Calibration

The calibration steps are:

- Start QGroundControl and connect the vehicle.
- Select "Q" icon > Vehicle Setup > Sensors (sidebar) to open Sensor Setup.
- Click the Gyroscope sensor button.

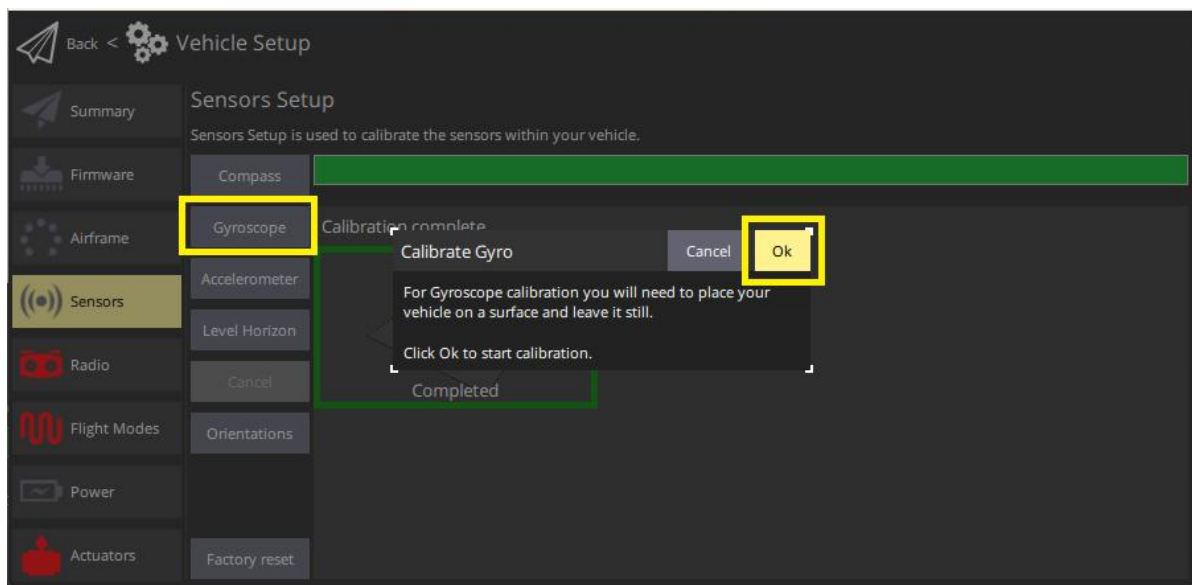


Figure B10: Gyroscope Calibration Starting Process

- Place the vehicle on a surface and leave it still.
- Click Ok to start the calibration.
- The bar at the top shows the progress

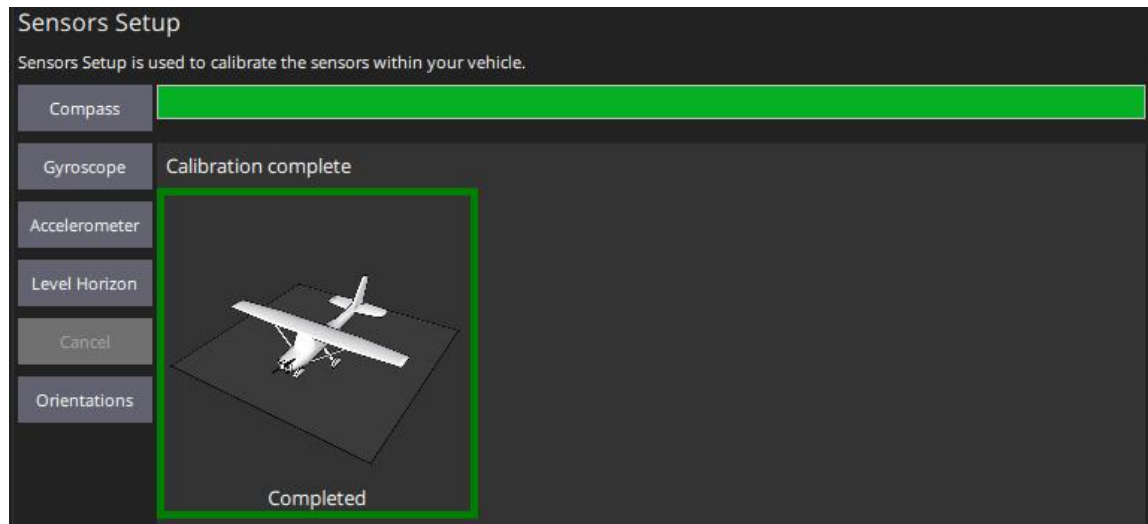


Figure B11: Gyroscope Calibration Ending Process

When finished, *QGroundControl* will display a progress bar *Calibration complete*

1.4 Compass Calibration

The compass calibration process configures all connected internal and external magnetometers. *QGroundControl* will guide you to position the vehicle in several set orientations and rotate the vehicle about the specified axis.

Performing the Calibration

Before starting the calibration:

- Choose a location away from large metal objects or magnetic fields.
- Connect via telemetry radio rather than USB if possible. USB can potentially cause significant magnetic interference.
- If using an external compass (or a combined GPS/compass module), make sure it is mounted as far as possible from other electronics in order to reduce magnetic interference, and in a *supported orientation*.

i. Complete Calibration

The calibration steps are:

- Start *QGroundControl* and connect the vehicle.
- Select **"Q" icon > Vehicle Setup > Sensors** (sidebar) to open *Sensor Setup*.
- Click the **Compass** sensor button.

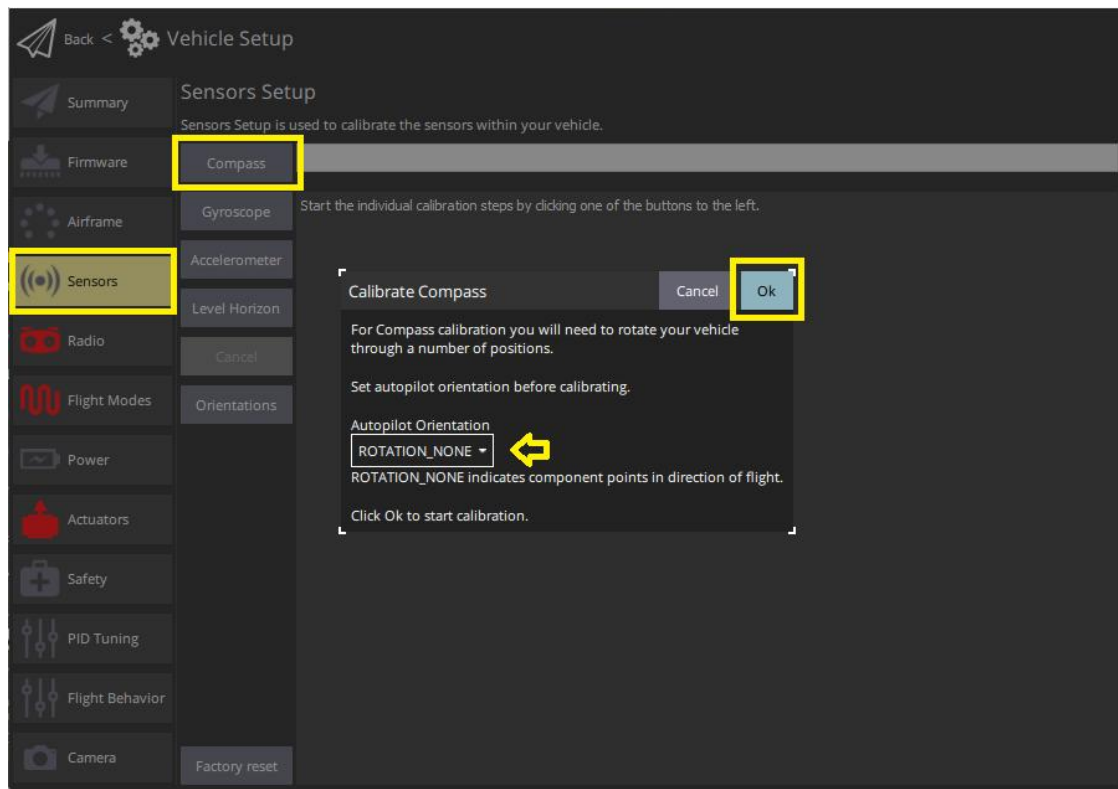


Figure B12: Compass Calibration Starting Process

1. Click OK to start the calibration.
2. Place the vehicle in any of the orientations shown in red (incomplete) and hold it still. Once prompted (the orientation-image turns yellow) rotate the vehicle around the specified axis in /both directions. Once the calibration is complete for the current orientation the associated image on the screen will turn green.

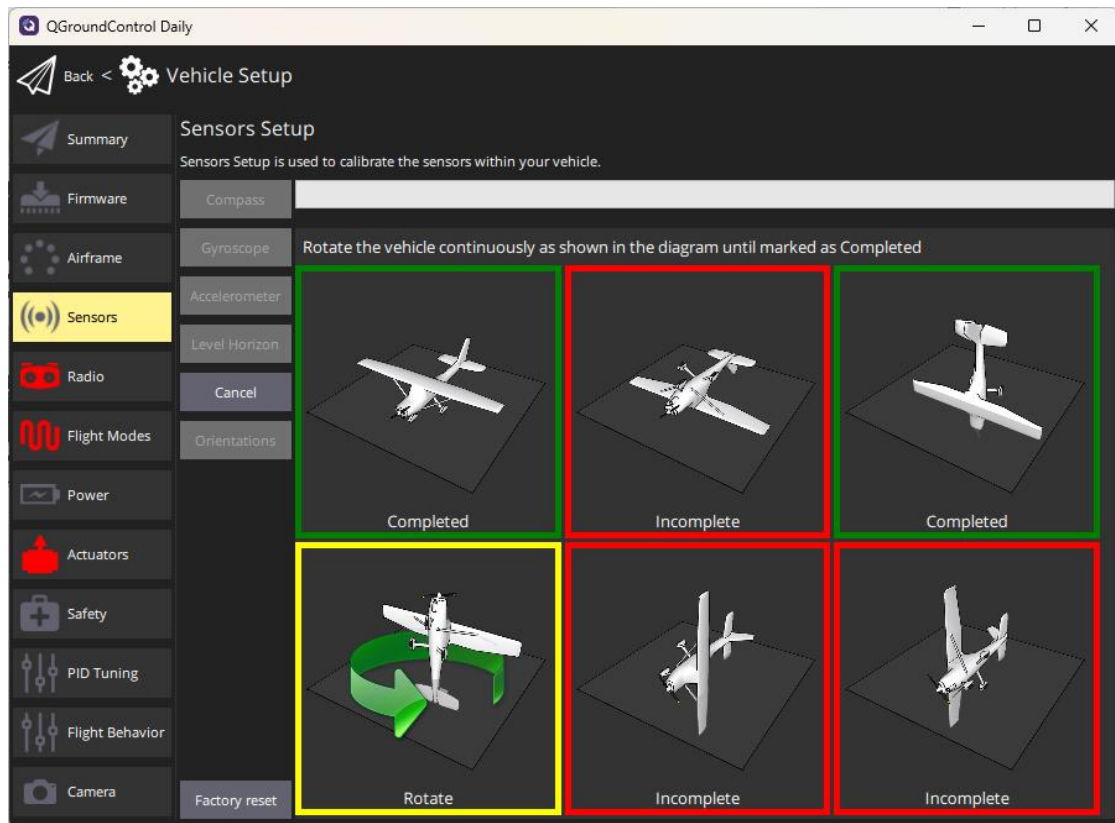


Figure B13: Compass Calibration Starting Ending Process

3. Repeat the calibration process for all vehicle orientations.

Once you've calibrated the vehicle in all the positions *QGroundControl* will display *Calibration complete* (all orientation images will be displayed in green, and the progress bar will fill completely). You can then proceed to the next sensor.