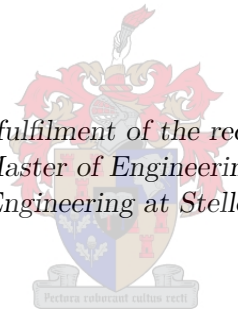
Development of a Close Quarters Collision-Protected Aerial Drone

by

Angus William Du Toit Steele



Thesis presented in partial fulfilment of the requirements for the degree of
Master of Engineering
in the Faculty of Engineering at Stellenbosch University

Supervisors:

Dr J.A.A. Engelbrecht Mr J Treurnicht

Department of Electrical and Electronic Engineering

December 2019

Declaration

1. I have read and understand the Stellenbosch University Policy on Plagiarism and the definitions of plagiarism and self-plagiarism contained in the Policy [Plagiarism: The use of the ideas or material of others without acknowledgement, or the re-use of one's own previously evaluated or published material without acknowledgement or indication thereof (self-plagiarism or text-recycling)].

2. I also understand that direct translations are plagiarism.

3. Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism.

4. I declare that the work contained in this assignment is my own work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment.

	A. Steele	***December 2019***
Student Number	Initials and Surname	Date

Abstract

This thesis presents the design, implementation and verification of a close quarters, collision-protected aerial drone. The ultimate goal of this work is to enable an unmanned aerial drone to navigate a given set of waypoints in a partially-known environment while avoiding collisions with unexpected obstacles.

The airborne platform was selected and the avionics system was designed to satisfy the operational requirements for CECAD (Confined Environment Capable Aerial Drone), a collision-protected aerial drone that could be used for mapping partially-known and potentially hazardous spaces found in an underground mining environment. Following a survey of existing rotorcraft designs, an overlapping quadrotor configuration was selected for the vehicle, since it was deemed to be the most suitable for flight in narrow confined spaces. The PixHawk, open-source flight controller was chosen due to its integrated sensors and communication ports, well-developed open-source flight control software and its large community of users. Ultrasonics were chosen as the proximity sensors used for obstacle avoidance.

Modelling and system identification of the actual vehicle were performed to create a representative mathematical model of the aircraft to be used for flight control design and verification. A complete flight control system was designed for the vehicle, and a waypoint navigation system with integrated obstacle avoidance was developed. The flight controllers were designed to provide tight position tracking and disturbance rejection, to enable stable flight and collision avoidance in a confined environment. A heading controller was added to keep the nose of the vehicle pointed generally in the direction of the vehicle's direction of travel. The waypoint navigation system schedules a sequence of position waypoints for the flight controllers, while the integrated obstacle avoidance function superimposes an obstacle avoidance velocity command on the waypoint navigation velocity command.

The system was implemented and verified in simulation using a simulation model that was created in Matlab and Simulink. Simulation models were created for the vehicle, the environment, the flight control system, and the waypoint navigation with obstacle avoidance. The simulation results show that the vehicle can successfully navigate waypoints in a partially-known environment while avoiding unexpected obstacles.

Uittreksel

Hierdie tesis beskryf die ontwerp, implementering, en verifikasie van 'n naby-kwartiere, botsing-beskermde hommeltuig. Die uiteindelike doel van die navorsing is om 'n onbemande hommeltuig in staat te stel om 'n gegewe stel wegpunte in 'n gedeeltelik-bekende omgewing te navigeer en terselfdertyd botsings met onverwagte hindernisse te vermy.

Die lugraam is gekies en die vlugelektronika is ontwerp om die operasionele vereistes te bevredig vir CECAD ("Confined Environment Capable Aerial Drone"), 'n botsing-beskermde onbemande hommeltuig wat gebruik kan word om gedeeltelik-bekende en potensieël gevaarlike ruimtes in 'n ondergrondse myn omgewing te karteer. Nadat 'n studie gemaak is van bestaande rotortuig ontwerpe, is 'n oorvleuelde, vier-rotor konfigurasie gekies as die mees geskikte konfigurasie vir vlug in smal inperkende ruimtes. Die PixHawk oopbron vlugbeheerder is gekies op grond van sy geïntegreerde sensore en kommunikasiepoorte, goed-ontwikkelde oopbron vlugbeheer sagteware, en sy groot gebruikersgemeenskap. Ultrasoniese sensore is gekies as die nabyheid sensore wat gebruik sal word vir hindernisvermyding.

Modellering en stelselidentifikasie van die werklike voertuig is uitgevoer om 'n verteenwoordigende wiskundige model van die onbemande voertuig te skep, sodat dit gebruik kan word vir vlugbeheer ontwerp en verifikasie. A volledige vlugbeheerstelsel is ontwerp vir die voertuig, en 'n wegpunt navigasie stelsel met geïntegreerde hindervermyding is ontwikkel. Die vlugbeheerders is ontwerp om goeie posisievolging en steurseinverwerping te verskaf, ten einde stabiele vlug en botsingvermyding in 'n inperkende omgewing moontlik te maak. 'n Gierhoek beheerder is bygevoeg om die neus van die voertuig gemiddeld in dieselfde rigting as die voertuig se rigting van beweging te hou. Die wegpunt navigasie stelsel skeduleer 'n reeks van posisie wegpunte vir die vlugbeheerders, terwyl die geïntegreerde hindernisvermyding funksie 'n hindernisvermyding snelheidsvektor superponeer op die wegpunt navigasie snelheidsbevel.

Die stelsel is geïmplementeer en geverifeer in simulasie deur gebruik te maak van 'n simulasiemodel wat geskep is in Matlab en Simulink. Simulink modelle is geskep vir die voertuig, die omgewing, die vlugbeheerstelsel, en die wegpunt navigasie stelsel met hindernisvermyding. Die simulasie resultate wys dat die voertuig suksesvol die wegpunte kan navigeer in 'n gedeeltelik-bekende omgewing terwyl dit onverwagte hindernisse vermy.

Contents

Abstract **iii**

Uittreksel **iv**

List of Figures **viii**

List of Tables **xi**

Nomenclature **xii**

Acknowledgements **xiv**

1 Introduction **1**
1.1 Project Background . . . 1
1.2 Problem Statement . . . 1
1.3 Application Requirements . . . 2
1.3.1 Controlled Indoor Flight . . . 2
1.3.2 Method of Expansion for Industrial Applications . . . 2
1.4 Project Scope . . . 3
1.5 Project Execution . . . 3
1.6 Thesis Outline . . . 4

2 Literature Review **5**
2.1 Flight Theory . . . 5
2.1.1 Momentum Theory and Thrust Basics . . . 5
2.1.2 Disk and Power Loading . . . 7
2.1.3 Electrical Power to Thrust . . . 8
2.2 Analysis of Conventional Rotor Wing Configurations . . . 8
2.2.1 Helicopter . . . 9
2.2.2 Coaxial Rotors . . . 10
2.2.3 Tandem Rotors . . . 10
2.2.4 Multirotor Designs . . . 11
2.3 Quadrotor Flight Dynamics . . . 11
2.3.1 Coordinate Systems, Rotations and Nomenclature . . . 12
2.3.2 Kinetics and Kinematics . . . 15
2.3.3 Mass Model and the Inertia Tensor . . . 15
2.3.4 Rotor Generated Forces and Moments . . . 16
2.3.5 Disturbances . . . 17
2.3.6 Instrumentation . . . 20
2.4 Review of Existing Flight Control Strategies . . . 21
2.4.1 Controller Architecture . . . 21
2.5 Collision Protection and Avoidance Techniques . . . 22
2.5.1 Impact Resistance . . . 23
2.5.2 Rolling Cage . . . 23
2.5.3 Sensing Techniques and Requirements for Collision Avoidance . . . 23
2.5.4 Collision Avoidance and Path Planning Algorithms . . . 24

3 Platform Design **28**
3.1 Design Considerations . . . 28
3.1.1 Physical Restrictions and Requirements . . . 28
3.1.2 Manoeuvring Decisions . . . 28
3.1.3 Disturbances . . . 29
3.1.4 Thrust Overhead . . . 29
3.1.5 Flight Time . . . 29
3.1.6 Discussion . . . 29
3.2 System Hardware Architecture . . . 29
3.2.1 Platform Construction . . . 30
3.2.2 Electronics Interface . . . 31
3.3 Platform Construction . . . 32
3.3.1 Concept 1 - The Overlapping Quad . . . 33
3.3.2 Concept 2 - The Unlike Size Quad . . . 34
3.3.3 Concept Comparison . . . 35
3.4 Electronic Design . . . 37
3.4.1 Flight Controller . . . 37
3.4.2 On Board Computer . . . 38
3.4.3 Location . . . 40
3.4.4 Object Avoidance . . . 40

4 Mathematical Modelling and System Identification **41**
4.1 Dynamic Flight Model . . . 41
4.2 System Identification . . . 41
4.2.1 Mass and Inertia . . . 42
4.2.2 Thrust and Moment Profiles . . . 42
4.2.3 Drag Coefficients . . . 43
4.2.4 Sensor Constants . . . 44
4.2.5 Wind Model . . . 44
4.3 Simulation Configuration . . . 45
4.3.1 Motor Mixer . . . 45

5 Controller Design **47**
5.1 Design Goals . . . 47
5.2 Flight Control Strategy . . . 47
5.3 Altitude Controller . . . 49
5.3.1 Heave Dynamics . . . 49
5.3.2 Heave Controller . . . 50
5.3.3 Climb Rate Controller . . . 52
5.3.4 Altitude Hold Controller . . . 55
5.4 Horizontal Control . . . 59
5.4.1 Roll and Pitch Rate Dynamics . . . 60
5.4.2 Roll and Pitch Rate Controllers . . . 60
5.4.3 Tilt Angle Controller . . . 64
5.4.4 Linear Velocity Controller . . . 68
5.4.5 Global Position Tracking Controller . . . 71
5.5 Heading Controller . . . 73
5.5.1 Yaw Rate Dynamics . . . 74
5.5.2 Yaw Rate Controller . . . 74
5.5.3 Yaw Angle Controller . . . 76

6 Flight Strategy and Obstacle Avoidance **82**
6.1 Waypoint Navigation and Heading Control . . . 82
6.1.1 Waypoint Navigation . . . 82
6.1.2 Align Yaw With Heading . . . 84
6.2 Implementation of the Proximity Sensors . . . 86
6.3 Obstacle Avoidance . . . 87

6.3.1 Obstacle Avoidance Implementation 87
6.3.2 Obstacle Avoidance Controller 89
6.3.3 Obstacle Avoidance Discussion 90

7 Simulated Flight Tests **93**
7.1 Test Objectives and Test Cases 93
7.2 Simulation Setup 94
7.2.1 Indoor Environment Model 94
7.2.2 Obstacle Distance Model 96
7.3 Simulated Flight Tests: Test Procedures and Simulation Results 97
7.3.1 Flight Control and Waypoint Navigation With Wind Disturbance and No Obstacles 97
7.3.2 Simple Obstacle Avoidance With Wind Disturbance 100
7.3.3 Waypoint Navigation and Obstacle Avoidance in a Basic Terrain 101
7.3.4 Waypoint Navigation and Obstacle Avoidance in a More Complex Terrain . 105
7.3.5 Limitations 108
7.4 Simulated Flight Tests Discussion 110

8 Conclusions and Recommendations **112**
8.1 Summary and Conclusions 112
8.2 Recommendations 113

Bibliography **114**

List of Figures

2.1 Visualisation of induced air flow through a rotor in free space (Taken from [4]) 6
2.2 Momentum theory in hover (Adapted from [4]) 6
2.3 Image representing, various Disk Loading values for varying rotorcraft (Taken from [4]) 7
2.4 Main components of a helicopter (Taken from [10]) 9
2.5 Image demonstrating the NOTAR system (Taken from[10]) 9
2.6 Different methods of lateral control in a Coaxial MAV (Adapted from [8]) 10
2.7 Quadrotor configuration 11
2.8 The inertial and body frames 12
2.9 Individual rotations around the X, Y and Z axes respectively. 13
2.10 Typical naming convention of body forces, moments and velocities for a quadrotor. . . 14
2.11 Forces and moments acting in the body frame on an X-Configuration quadrotor. . . . 16
2.12 Typical moments created by drag forces 18
2.13 Disturbances created by being in close proximity with a wall 19
2.14 Velocity components though the rotor for, no wall (left) and near wall (right) conditions (Taken from [2]) 20
2.15 Graph showing relationship between distance from the wall and moment felt be the craft (Taken from [2]) 20
2.16 Disturbance observer based controller structure [2] 22
2.17 Demonstration of the Bug2 algorithm (Image taken from [32]) 25
2.18 Local minima seen with potential field method 26
2.19 Example of RRT on a low resolution grid 27

3.1 System architecture 30
3.2 Rendering of initial concept of the unlike rotor size quadcopter (Left). Malloy Aeronautics hoverbike concept (Right) (Picture taken [41]). 33
3.3 Overlapping concept, visual representation of rotor pairs. Image modified from [41] . . 33
3.4 Graph representing the effects of overlapping rotors in a quadrotor 34
3.5 Unlike size quad visual representation of rotor pairs 34
3.6 Graphical representations of the thrust ratios for the unlike size quad 36
3.7 Pixhawk flight controller 38
3.8 Raspberry Pi 3 Model B 39
3.9 The Odroid XU4 39

4.1 Bifilar pendulum for inertia measurement 42
4.2 Thrust ranges for motor rotor pairs 43
4.3 Wind model 45
4.4 Motor mixer 45

5.1 High level control strategy 48
5.2 Heave controller - Control diagram 50
5.3 Heave controller - Root locus 51
5.4 Heave controller - Bode plots 51
5.5 Heave controller - Step response 52
5.6 Climb rate controller closed loop 53
5.7 Climb rate controller 53
5.8 Climb rate controller - Root locus 54

5.9 Climb rate controller - Open-loop bode plots . . . 54
5.10 Climb rate controller - Step response . . . 55
5.11 Altitude hold controller closed loop . . . 55
5.12 Altitude hold controller . . . 56
5.13 Altitude hold controller - Root locus . . . 56
5.14 Altitude hold controller - Bode plots . . . 57
5.15 Altitude hold P controller - Step response . . . 58
5.16 Altitude hold P controller - Step response with inner loop measurement offset . . . 58
5.17 Altitude hold P with limited I controller - Step responses . . . 59
5.18 Graph showing differences in ideal and non linear simulation responses for altitude . . . 59
5.19 Roll and pitch rate controller design . . . 61
5.20 Roll rate controller - Root locus . . . 61
5.21 Roll rate controller - Bode plot . . . 62
5.22 Pitch rate controller - Bode plot . . . 62
5.23 Pitch and roll rate controllers - Step responses . . . 63
5.24 Roll rate controller - Motor commands . . . 64
5.25 Pitch rate controller - Motor commands . . . 64
5.26 Tilt angle controller . . . 65
5.27 Conversion technique using dot and cross products . . . 66
5.28 Roll and pitch angle simplified closed loops . . . 66
5.29 Roll angle controller - Bode plots . . . 67
5.30 Roll and pitch angle controller - Step responses . . . 67
5.31 Roll angle controller - Motor commands . . . 68
5.32 Pitch angle controller - Motor commands . . . 68
5.33 North, East simplified closed loops . . . 69
5.34 North velocity controller - Bode plots . . . 70
5.35 North velocity controller - Root locus plot . . . 70
5.36 North velocity controller - Step responses with a disturbance . . . 71
5.37 North position controller - Bode plots . . . 72
5.38 North position controller - Step response . . . 72
5.39 North position controller - Large step response with and without a limiter . . . 73
5.40 Graph showing differences in ideal and non linear simulation responses for North position 73
5.41 Yaw rate controller - Control diagram . . . 75
5.42 Yaw rate controller - Root locus . . . 75
5.43 Yaw rate controller - Bode plots . . . 76
5.44 Yaw rate controller - Step responses . . . 76
5.45 Yaw angle PI closed loop system . . . 77
5.46 Yaw angle PI controller - Control diagram . . . 77
5.47 Yaw angle PID controller - Control diagram . . . 77
5.48 Yaw angle controller - Root locus (Left:PI, Right:PID) . . . 78
5.49 Yaw angle controller - Bode plots . . . 79
5.50 Yaw angle controller - Step responses . . . 79
5.51 Yaw angle controller - Step responses including inner loop measurement offset . . . 80
5.52 Yaw angle controller - impulses . . . 80
5.53 Graph showing differences in ideal and non linear simulation responses for yaw angle . 81

6.1 Waypoint navigator state machine . . . 83
6.2 Simple waypoint flight . . . 83
6.3 Simple waypoint flight - North, East, Down positions . . . 84
6.4 Yaw alignment controller . . . 85
6.5 Yaw alignment controller utilised for circular flight. Orange line represents the current heading . . . 85
6.6 Implementation of sensor noise . . . 86
6.7 Sensor placement for obstacle avoidance . . . 87
6.8 Visual descriptive aid for a virtual spring damper sensor system . . . 88
6.9 High level view of obstacle avoidance controller . . . 89
6.10 Individual sensor controller . . . 89

6.11 Obstacle avoidance demonstration - Position plot 90
6.12 Obstacle avoidance demonstration - Position and velocity command plots 91
6.13 Obstacle avoidance - Straight wall 91
6.14 Obstacle avoidance - Random object in flight path. Purple line represents the obstacle avoidance vector 92

7.1 Snapshot of simulation in the Simulink environment 94
7.2 Plot representing the creation of the roof and a side wall 95
7.3 Plot representing the creation of a room containing an obstacle 95
7.4 Obstacle detection demonstration 97
7.5 Step response with and without disturbance - North position plot 98
7.6 Waypoint flight with disturbance - Isometric and top position plot 99
7.7 Waypoint flight with disturbance - North, East and Down position plot 99
7.8 Right and top view of corridor flight with wind disturbance and overlayed obstacle avoidance vector 100
7.9 Corridor flight with disturbance - North, East, Down positions and obstacle avoidance velocity commands 101
7.10 Navigated flight in a wide corridor, showing avoidance vector 102
7.11 Navigated flight in a narrow corridor, showing avoidance vector 103
7.12 Navigated flight in wide corridor with yaw alignment, showing current heading of vehicle 104
7.13 Navigated flight in wide corridor with and without yaw alignment 105
7.14 Layout of environment and waypoints. 106
7.15 Yaw alignment plot of a generic flight test while utilising the heading alignment controller, showing avoidance vector. 107
7.16 Yaw alignment plot of a generic flight test while utilising the heading alignment controller, showing current heading of vehicle. 107
7.17 Plot of a generic flight test both with and without the yaw alignment active. 108
7.18 Limitations of the obstacle avoidance routine as a navigation algorithm. Straight wall in a wide space. 109
7.19 Limitations of the obstacle avoidance routine as a navigation algorithm. A wide open space leading into a narrow corridor. 110

List of Tables

2.1 Standard nomenclature 14
2.2 Examples of MAV weight distributions (Adapted from [9]) 15

3.1 Table representing the size comparison of the two concepts 36
3.2 Table representing the end comparison of the two concepts 37

4.1 Measured moments of inertia 42
4.2 Measured rotor thrusts 43
4.3 Drag coefficients 44
4.4 IMU sensor coefficients 44
4.5 GPS coefficients 44

5.1 Thrust headroom controller percentages 49
5.2 Heave controller limits 51
5.3 Climb rate controller limits 54
5.4 Altitude hold controller limits 57

Nomenclature

Abbreviations and Acronyms

MAV	Micro Aerial Vehicle
RPM	Revolutions per Minute
UAV	Unmanned Aerial Vehicle
CECAD	Confined Environment Capable Aerial Drone
CSIR	Council of Scientific and Industrial Research
US	University of Stellenbosch
GPS	Global Positioning System
SLAM	Simultaneous Localisation and Mapping
NED	North, East, Down
DCM	Direct Cosine Matrix
IMU	Inertial Measurement Unit
PID	Proportional, Integral, Derivative
DOBC	Disturbance Observer Based Control
TOF	Time of Flight

Greek Letters

α	Angle of Attack
ρ	Air Density
ϕ	Roll Angle
θ	Pitch Angle
ψ	Yaw Angle
η	Efficiency
τ	Lag Timing Constant

Lowercase Letters

b	Wing span
v	Velocity
m	Mass

q	Quaternion Representation

Uppercase Letters

X	Force vector along the body X-axis
Y	Force vector along the body Y-axis
Z	Force vector along the body Z-axis
L	Moment around body X-axis
M	Moment around body Y-axis
N	Moment around body Z-axis
U	Linear velocity along the body X-Axis
V	Linear velocity along the body Y-Axis
W	Linear velocity along the body Z-Axis
P	Angular velocity around body X-axis
Q	Angular velocity around body Y-axis
R	Angular velocity around body Z-axis
T	Thrust
A	Area
I	Inertia tensor
R_{LD}	Rotor lift to drag ratio
F_D	Force due to drag
M_D	Moment due to drag

Acknowledgements

It has become abundantly clear that undertaking any large portion of work requires a multitude of support. I would like to take this opportunity to thank the people who helped me in my endeavour and acknowledge their contributions.

Firstly a big thank you to the CSIR for funding my studies and providing resources and time. A special mention must go to my colleague and friend John Dickens for his insights into numerous aspects of this work.

Stellenbosch University for their incredible support throughout the project. Even as a part time student, working off campus I was shown dedication, care and support throughout this project.

My study leaders Japie Engelbrecht and Johann Treurnicht. The practical knowledge Johann offered has been a critical part to ensuring the success of this work, as well as to the growth of my control engineering knowledge. Japie's incredible insight into control theory and his uncanny ability of teaching difficult concepts has given me both confidence and understanding in control systems. The accessibility and time both have offered for editing and video conferencing has been critical during the final stages of this work. Thank you for the unwavering support and commitment.

My family has been the cornerstone to my state of being during this time. Completing a masters part time has been one of the most challenging experiences of my life and it would not be possible with out the support my family has offered. A special thanks to my mother Annette du Toit and my partner Dayle Nel for their unwavering support during this period.

Lastly, I must thank Bart Marsman. Initially a visiting student from the Netherlands and ultimately one of the most important crossroads of my masters journey. I had given up and you brought hope and enthusiasm. I dare not think what would have happened without your kindness, energy and knowledge. Friends are made during times of adversity and along with knowledge this work has produced a friend for life.

Chapter 1

Introduction

1.1 Project Background

Tracked and wheeled robots are beginning to reach their limitations, and society is in need of more complex and versatile vehicles. For a land robot to successfully navigate a complex or cluttered environment, the designer must look at creating a legged robot. Legged designs introduce complexity into any system due to the intensive control theory required. There has been some great progress in legged designs, such as Big Dog created by Boston Dynamics [1]. Nevertheless, deploying currently available legged platforms could cost valuable time with lengthy navigation routines. An alternative approach would be to use an aerial platform that could do overhead surveillance. A drone could complete the required task by flying over the floor-level complexities in the operating environment. However, with conventional flight techniques and platforms, most collisions would cause a failure and the system would not be able to complete its mission.

Many of the desired use cases are not open aired and the platform will be required to fly below and even through obstacles to complete its task. A good example of this is in search and rescue missions. An aerial vehicle will be required to navigate through damaged or even collapsed buildings. The same platform could be used in a mining environment to assist miners in assessing unexplored and potentially dangerous areas, which are currently assessed by humans. To improve safety in the mining environment, two South African research institutes have agreed to a joint collaboration in solving some of these aspects for an underground mine environment. This project involves both University of Stellenbosch (US) and the Council of Scientific and Industrial research (CSIR). A mining environment has many applications for a collision-protected drone including the mapping of unknown and potentially hazardous environments. The drone would be able to fly in, conduct a survey of the environment, and feed that information back to the miners, ensuring a safer work environment while minimising costly delays.

1.2 Problem Statement

There are many potential applications for a drone capable of close quarters flight. CECAD (Confined Environment Capable Aerial Drone) was seen as a potential safety platform for use in mines. Unsafe underground territories create a need for unmanned vehicles to do inspections. These areas are currently been mapped by trained professionals who risk their lives going into these unsecured regions. The use of land vehicles proves difficult and slow in complex terrains, creating a need for an alternative solution.

Designing any aerial drone introduces many complexities, including obtaining the required aerodynamics to achieve stable flight. There are modules that one can buy to stabilise the drone, but in a confined indoor space, a requirement is added that the drone must avoid collisions with the environment, and must maintain stable flight near floor, wall and ceiling surfaces as shown in [2], [3]. Several strategies will need to be investigated to assist the drone in navigating these confined environments. The drone should attempt to maintain a set distance from the walls, floors and other obstructions. For an indoor application it is important that the vehicle must be able to fly in an environment with no GPS links. Although this factor will not be solved in the scope of this

project, the design of CECAD should consider some of the factors involved to ensure expansion into that research can be done with relatively small changes to the work accomplished here.

To ensure that the platform can be extended to industrial applications, certain external factors and peripherals need to be considered. The drone will need to be able to complete some of these missions autonomously, especially when line of sight and communications are lost. The drone will need to be small to increase its ability to access confined spaces, the smaller size will limit payload and flight time. To complete a useful mission, the drone must carry a larger power source on board the aircraft to enable a sufficient flight time.

1.3 Application Requirements

The problem statement above briefly introduced certain user requirements that CECAD must satisfy. This section attempts to address each of these points and define them more specifically as key requirements for the system as a whole. This statement of requirements can be started by creating two high level objectives, achieving flight in a confined indoor space and providing the ability to expand into industrial applications.

1.3.1 Controlled Indoor Flight

The identified requirements of the system began by providing an aerial platform with the ability to fly in an enclosed, confined space such as a mining box hole. To do this, the platform must be able to position itself relative to any obstructions above, below or beside itself. This will require that the platform can measure its proximity to the surroundings in all directions. A typical mining box hole will be 3 m - 5 m in diameter, creating the need for proximity measurements with a range of at least 2.5 m and a vehicle which can comfortably fit inside this space. The flight controller must also therefore be able to use the additional sensor inputs in its control laws and other real time processes. When a new obstruction is detected it must have the ability to deterministically move away and reposition itself, while not straying too far off the mission plan.

In order for the platform to complete missions in this environment, the drone needs to withstand the disturbances introduced by a flight path close to obstructions. The vehicle ideally would be mechanically protected to prevent irreversible damage caused by a potential collision. The flight control law must be able to reject disturbances caused by the aerodynamic ground effect when the vehicle is in close proximity to walls, floors and other obstructions. Due to the nature of strong infrequent wind located in tunnels, the flight controller must also be able to reject constant wind disturbances.

1.3.2 Method of Expansion for Industrial Applications

If the above requirements are met, the platform could be expanded into an array of industrial and research applications. Most of these use cases will require additional flight modes, sensors and other peripherals. Although not all these different use cases will be explored in this project, they must be considered so that expansion into these realms can be done with minimal rework being needed on the platform.

Since most of these missions will require some level of autonomy, the chosen flight controller must include an autopilot flight mode that allows the user to switch between manual and automatic mode. There must be a method to send flight data back to a ground control station for real-time monitoring of the mission. The drone must provide a mechanism for mounting an array of different sensing equipment on board. This includes accounting for the extra thrust and electrical power requirements added by the sensors. Typical SLAM (Simultaneous Localisation and Mapping) equipment could vary between a set of stereo cameras or a depth sensor. Lightweight equipment would weigh less than 500 g and that can be set as the payload requirement for CECAD . Finally the drone must be able to stay in the air long enough to complete a mission. The CSIR has specified a minimum flight time of 30 minutes for the vehicle. With indeterminable mission lengths at this point, the drone must be able to expand its battery capacity to account for longer missions.

1.4 Project Scope

This section discusses the scope of the project. This project will identify an ideal platform configuration to be used in the work, including avionics and sensing. After the platform has been identified, a process of mathematical modelling will be done to create an accurate model of the chosen system. Once a complete simulation is designed, global position and heading controllers will be designed. These controllers will be verified in simulation. A flight strategy including obstacle avoidance will be designed to enable flight in a partially-known, confined environment. The full system will be tested and verified in simulation only.

1.5 Project Execution

This section briefly describes the work conducted and provides an overview of the project execution. The project included the following activities: the selection of an appropriate airborne platform, the design of the avionics system architecture, the selection of the hardware components, the mathematical modelling and system identification of the actual vehicle dynamics, the design of the flight controllers, the design of the waypoint navigation system with integrated obstacle avoidance, and finally the implementation and verification of the system in simulation.

A literature review on relevant theory and previous research was conducted to gain a better understanding of aerial vehicles and collision avoidance strategies.

The avionics architecture was then designed to satisfy all vehicle and system requirements. The considerations of the design included computing speed and weight of the flight control and computing hardware, onboard inertial measurement unit, and obstacle avoidance sensors. The design also catered for integration with an additional payload.

The airborne platform was selected to satisfy the operational requirements of CECAD. Following a survey of existing rotorcraft designs, an overlapping quadrotor configuration was selected for the vehicle, since it was deemed the most suitable for flight in narrow confined spaces. The overlap in the rotors optimises the thrust generation capabilities of the platform while reducing the overall size.

Once the platform and avionics had been selected, the mathematical modelling and system identification of the vehicle was performed to create an accurate simulation of the system, for verification purposes. This process included modelling the flight dynamics of the vehicle, performing system identification laboratory experiments with the real aircraft, and using data readily available in sensor datasheets. Noise was added to all the sensor measurements and update rates were limited to create a more realistic representation of the system.

Three separate controller subsystems were designed to control the altitude, the horizontal position, and the heading of the vehicle. Each flight control subsystem was designed to be robust against disturbances and to have tight position tracking to ensure stable flight inside a confined environment and to enable collision avoidance.

A flight strategy was developed to optimise the use of the vehicle in a narrow space similar to that encountered inside a mining environment. To enable autonomous flight, a waypoint navigator was designed to allow a user to input a set of waypoints based on partial knowledge of the environment. A heading alignment strategy was added to the existing controllers, ensuring that the drone maintains a set heading based on its current velocity. The final consideration was a thorough design and implementation of an obstacle avoidance routine. The system had to ensure no collisions would occur and ensure suitability for the implementation of a higher route planning strategy in future work.

A series of simulated flight tests were then designed to evaluate the operation of each component of the system. The first test is used evaluate the waypoint navigation and disturbance rejection capabilities of the vehicle, without obstacle avoidance. The second test is designed to evaluate the obstacle avoidance routine in the presence of a disturbance and a simple obstacle. The third test is designed to evaluate the interaction between the waypoint navigation, obstacle avoidance and the heading alignment routines in a narrow, winding corridor arrangement. The fourth test is designed to evaluate the full system in a more complex environment which includes narrow

corridors, unexpected obstacles and an unachievable waypoint. The fifth and final test is designed to show the limitations of the design.

1.6 Thesis Outline

The layout of the thesis is presented in this section.

In Chapter 1 the research topic was introduced. The chapter provided the background and motivation for the research, formulated the problem statement, and gave an overview of the project execution.

In Chapter 2 a literature review is performed. The chapter starts by providing an overview of important flight theory concepts that are used further in the text. A review of conventional rotorcraft is performed, followed by an investigation into quadrotor flight dynamics. A review of existing flight controller strategies is done which is followed by research into different methods for collision protection and avoidance.

In Chapter 3 the platform design is presented. The chapter begins by considering the design requirements and restrictions. From these considerations a system architecture is proposed. The design of the vehicle is then performed, first by making decisions on the mechanical construction of the vehicle and followed by the design of the avionics.

In Chapter 4 the mathematical modelling and system identification is performed. The chapter begins by creating a dynamic flight model of the vehicle and is followed by the system identification of the actual vehicle's characteristics. The chapter concludes by providing a brief overview of the simulation configuration.

In Chapter 5 the controller design is presented. The chapter begins by outlining the design goals and presenting a flight control strategy. The design of the controllers is performed in the following order: altitude controller, horizontal controller, heading controller.

In Chapter 6 the flight strategy and obstacle avoidance routine is designed. A strategy for waypoint navigation and heading alignment is performed first. The chapter then presents the implementation of the proximity sensors and finally the design of the obstacle avoidance routine.

In Chapter 7 the simulated flight tests are performed. The chapter begins by presenting the test objectives and cases, followed by a look at the simulation setup. Five sets of flight tests are then performed, each designed to verify different aspects of the system. The chapter concludes by discussing the flight tests.

In Chapter 8 the conclusions and recommendations are presented. The chapter begins by providing a summary of the work and the conclusions based on the work completed in each chapter. The chapter finishes by providing recommendations for future work.

Chapter 2

Literature Review

This chapter seeks to evaluate existing research in the field of rotorcraft design and obstacle avoidance strategies. To critically review some of the high level concepts in rotorcraft design, a brief evaluation is given of flight theory and how it effects design decisions for rotorcraft. After an understanding of flight theory is obtained, it is necessary to evaluate how this theory is utilised in creating rotorcraft. Armed with a better understanding of flight generation for rotorcraft, an analysis of traditional rotorcraft configurations is completed. Due to the hazardous nature of the mission environments, existing collision protection techniques are then discussed. The next step is to review some of the methods used to control multi-rotor platforms. Once stable flight methods have been evaluated and discussed, the researcher reviewed existing methods for obstacle avoidance as well as the requirements of implementing an on-board obstacle avoidance system.

2.1 Flight Theory

This section seeks to describe principles of flight that will be pertinent to the design of a rotorcraft optimised for prolonged flight in a confined space. The section begins by describing the characteristics of thrust generation and the influence these have on different rotor configurations.

2.1.1 Momentum Theory and Thrust Basics

The forces generated by rotors are induced by pushing air through the rotor disk. Initially consider a helicopter in a hovering state (Weight(W) = Thrust(T)). Figure 2.1, taken from [4], helps visualise the induced air flow by showing how the rotor smooths out the air by forcing it through the disk area. This more uniform air creates an edge known as the slipstream or wake boundary, with the surrounding air remaining dormant [4]. Disturbances in the air stream will affect these characteristics and thus the thrust generation characteristic.

Rankine-Froude's Momentum Theory looks at this induced velocity as well as the displacement of air through the propeller, and attempts to quantify the induced thrust. While figure 2.1 helps visualise the principle, the variable naming convention for the equations is shown in figure 2.2 below. Labels 0, 1, 2 and ∞ refer to the locations of quiescent flow, inflow directly before the rotor, airflow immediately after the disk and the slipstream[1] or far wake condition respectively. The velocities are shown as the induced velocity in and out the rotor (v_i), the far wake velocity (v_∞) and finally v_0 represents the zone with zero flow rate. There is no velocity jump across the rotor, the energy being fed into the system by the rotor is represented by a pressure change.

The mass flow rate of the air through the rotors can then be described by (2.1.1), where (ρ) is the density of air and A is the area of one full blade rotation. The rate at which this mass of air is displaced becomes a crucial variable in rotor dynamics and is directly proportional to thrust (T). This relationship can be quantified as shown in (2.1.2). Thrust can also be calculated by finding the difference in pressures over the rotor disk as in (2.1.3). Since acceleration is the difference in v_∞ and v_0, (2.1.4) can also be obtained.

[1] Generally far wake is considered as 1 full rotor diameter distance away [4].

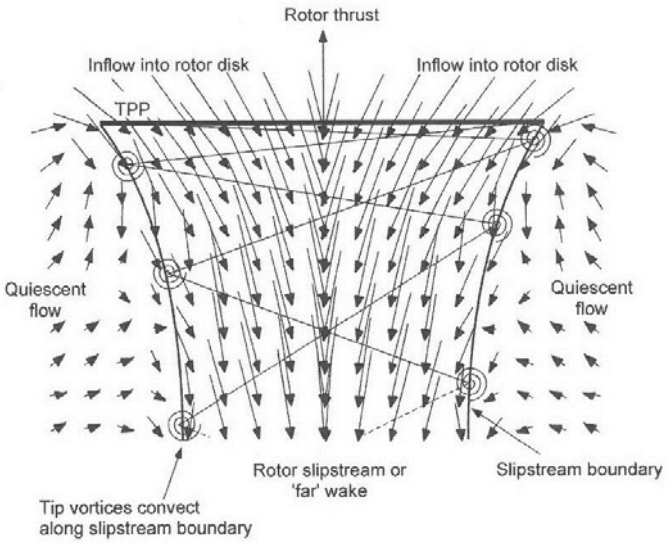


Figure 2.1: Visualisation of induced air flow through a rotor in free space (Taken from [4])

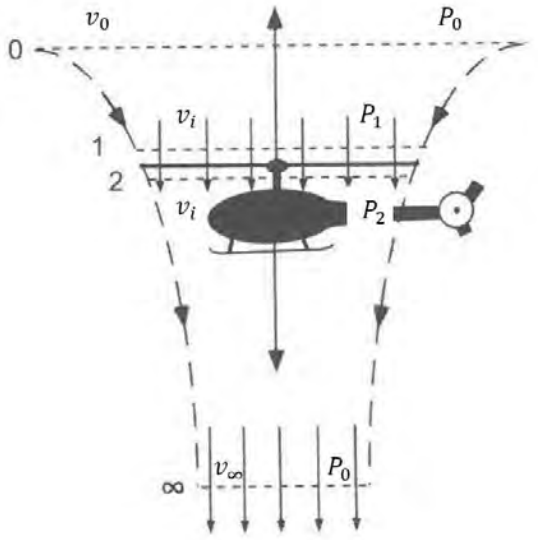


Figure 2.2: Momentum theory in hover (Adapted from [4])

$$\dot{m} = \rho A v_i \tag{2.1.1}$$
$$T = \dot{m} a \tag{2.1.2}$$
$$T = A(P_2 - P_1) \tag{2.1.3}$$
$$T = \rho A v_i (v_\infty - v_0) \tag{2.1.4}$$

Equation (2.1.4) demonstrates the negative effect of having active ambient air. This could be caused by surrounding turbulent airflow, wind or even translational movements.

2.1.2 Disk and Power Loading

2.1.2.1 Disk Loading

Disk loading (DL) is a term seen often in the world of rotorcraft, it is a simple but important ratio between thrust and the area a rotating disk makes. It is represented in (2.1.5). Since the pressure drop across each rotor is considered uniform, the disk loading for each rotor will equate to the pressure drop across that disk.

$$DL(\frac{N}{m^2}) = \frac{T}{A} = \frac{1}{2}\rho v_\infty^2 \tag{2.1.5}$$

For multi-rotor craft, the disk loading is assumed uniform across all rotors [4]. The overall disk loading of a single rotorcraft such as a traditional helicopter will be lower than that of a multi-rotor craft of a similar size [5]. Figure 2.3 shows some examples of disk loading values for a variety of rotor configurations, as shown disk loading is also a measure of hover efficiency.

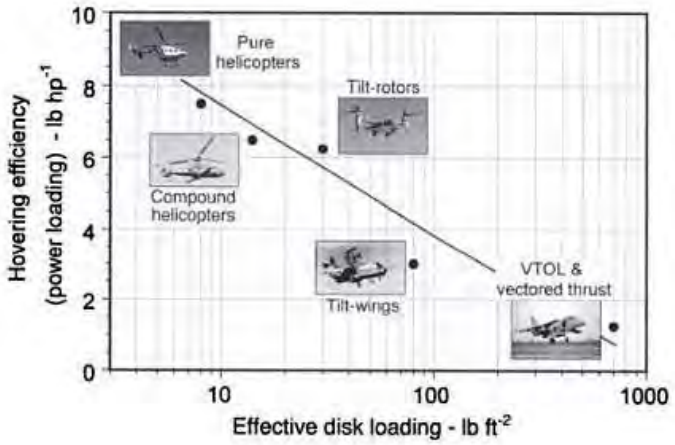


Figure 2.3: Image representing, various Disk Loading values for varying rotorcraft (Taken from [4])

A higher disk loading value results in larger values for induced velocities as well as the required power to hover. This means that the larger the blades, the higher the efficiency. More force will be generated by pushing large quantities of air slowly, than forcing small amounts of air through at high speeds. Of course with bigger blades, comes larger rotational inertia and geometry as well as the craft being less immune to gusts and interferences. A larger blade also creates faster tip velocities, which will limit the speed of the craft severely [4].

2.1.2.2 Power Loading

Power is given by the product of both thrust and the induced velocity at the blade. It can be written as shown in equation (2.1.6). What this ratio shows is that the ideal power is in cubic proportion to the induced velocity at the rotor. Therefore to reduce required power the rotor's induced velocity must be small, which can be accomplished by a significant increase in disk area [4].

$$P = 2\rho A v_i^3 \tag{2.1.6}$$

Another important ratio is between thrust and power, it is called power loading (PL) and is shown in equation (2.1.7). Power loading can be seen as a measure of craft efficiency.

$$PL(\frac{N}{kW}) = \frac{T}{P} \tag{2.1.7}$$

From equations (2.1.5) and (2.1.7) it can be shown that power loading is inversely proportional to disk loading. Therefore a craft with a lower disk loading will generally be a more efficient platform, as shown in Figure 2.3.

2.1.3 Electrical Power to Thrust

Equation (2.1.6) gives a quantitative approach to solving for aerodynamic power (P_i). If electrical power is taken as $P_e = VI$, where V is the applied voltage and I is the sourced current, with an efficiency of η then $P_i = \eta VI$. Noting that $P_i = Tv_i$ and using equation (2.1.6), a relationship between thrust and P_e can be formed and is represented in equation (2.1.8).

$$T = (2\rho A)^{\frac{1}{3}}(\eta P_e)^{\frac{2}{3}} \tag{2.1.8}$$

Equation (2.1.8) brings to light a very important relationship which states that thrust grows at a slower rate than the electrical input power to the rotor system.

$$T \propto P_e^{\frac{2}{3}}$$

2.2 Analysis of Conventional Rotor Wing Configurations

Some of the fundamental theories described above relate to the basics behind various rotor configurations and even varying flight techniques. Each different arrangement of blades introduces certain advantages and disadvantages to the system. Not every configuration will be applicable for all operations and it is important to determine what criteria are critical for the intended application. An analysis of varying rotor configurations is done below and follows a similar trend to that seen in [6], [7], [8] and [9]. The main weighted criterion for the discussion were listed in no particular order as:

1. Flight time and efficiency
2. Geometry and size
3. Drone Manoeuvrability
4. Control algorithms
5. Mechanical complexity

Before the analysis can be done, certain operating parameters of the different craft, surrounding the above mentioned criteria, need to be understood. There have been discussions regarding how rotor blades produce lift, this section discusses the real world implementation of those blades.

Typically a rotorcraft will be designed with either fixed pitched, or variable pitched rotors. A fixed pitched rotor is a rotor that has an optimally selected, unchangeable pitch and therefore a fixed angle of attack. The power requirements for either system are fairly similar, the advantages of a varying pitch is a single rotor has the potential for more dynamic force applications. The downfall however is the high level of complexity in the mechanical design. Both of these factors become pertinent in the final decision making of the platform design.

It is also known that any rotating member will produce a counter rotating torque to the static body, which means that any system with only one rotor will have inherent instability in the yaw axis. The end goal is to have a craft that can fly stably and accurately in 3 dimensions.

Having only a single, fixed pitched rotor allows only for control in the amount the craft flies up or down, as well as this fore mentioned instability. There are many different methods to obtain full six degrees of flight freedom. The following discussion tries to address each point listed above for different traditional methods.

2.2.1 Helicopter

A conventional helicopter is still the most widely used configuration for large rotorcraft [7]. It consists of a single main rotor, coupled with a smaller counter rotating rotor located in the tail as seen in figure 2.4, this is to counteract the developed counter torque.

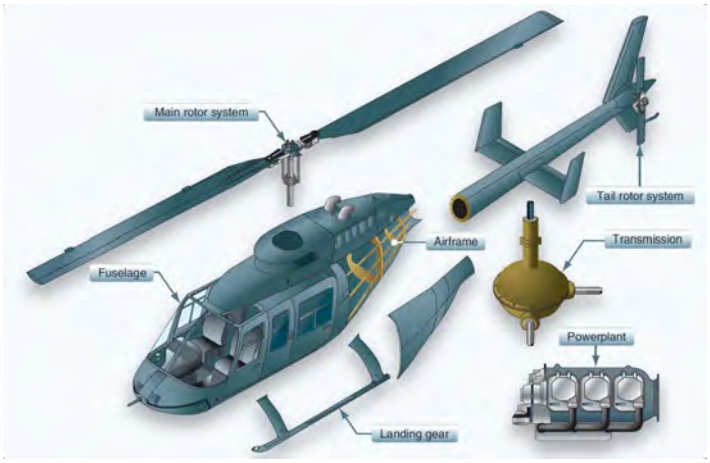


Figure 2.4: Main components of a helicopter (Taken from [10])

The main rotor of a standard helicopter has very low disk loading which gives it excellent hover efficiency. To achieve yaw stability this configuration makes use of a small tail rotor to counter act the induced moments. The extended tail rotor requires energy which it will draw from the motor while also adding a significant amount of length and weight to the craft. Cyclic control of the rotor's pitch allows the pilot to adjust the angle of attack of the rotor blades while they rotate, thus a forward pitch can be applied by increasing the lift on the left[2]. This set up is mechanically very complex and takes intensive control algorithms and laws to give stable control.

There are many different types of anti torque tail set ups. The ducted fan approach increases the efficiency of the tail rotor by channelling the air flow of the rotor. The NOTAR design [11] as seen in Figure 2.5 manipulates the airflow generated by the main rotor and directs it to counter act the induced torque. A tip-jet design eliminates the torque applied to the airframe and therefore no tail rotor is required [7].

Figure 2.5: Image demonstrating the NOTAR system (Taken from[10])

[2]This is true for an American style helicopter. The French design requires an increase of lift to the right

There have been many attempts at improving the standard helicopter design. These improvements have taken the form of adding rotors, designing hybrid aircraft and complex mechanical designs to harvest advantages of both the fixed wing and VTOL craft. Some have even tried to combine multiple features as Flanigan [12] did in his design of a tip-jet, compound, tilt rotor aircraft. In an attempt to keep the mechanical complexity to a minimum, not all configurations were investigated.

2.2.2 Coaxial Rotors

A coaxial configuration consists of two counter rotating blades located about the same centre of rotation that both use the same drive system. This eliminates the need for a tail rotor as the torque applied by both rotors cancel each other out. Localising the blades around a single point helps with the geometry of the craft as it is a more compact design. Using fixed pitched rotors, this platform will only give yaw and over all thrust control. Bohorquez et al in [8] attempted a number of lateral control methods, eventually settling on aerodynamic flaps to control the flow of the downwash, that and other methods are shown in figure 2.6. Briod et al in [13] also used the same set up in his team's design of the Gimball, a collision protected drone.

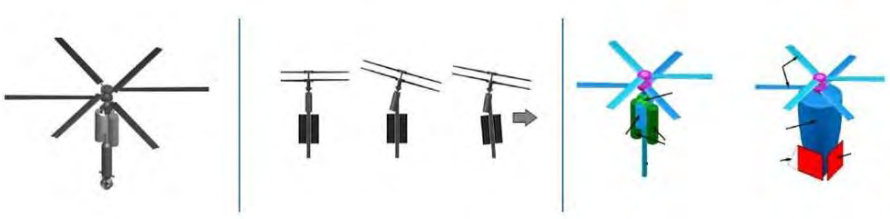


Figure 2.6: Different methods of lateral control in a Coaxial MAV (Adapted from [8])

The control flaps are the most common used form of lateral control for small coaxial MAVs. They introduce little mechanical complexity and do not require excessive power to use. The flaps do however decrease efficiency of the system by interfering with the rotor airflow. If designed correctly the flaps should only influence the system while in use. For hover and vertical flight the impact will be negligible. Each flap will require an actuator, this will increase total weight, power consumption and required mechanics [8].

The overlap of the rotors also induces an inefficiency into the system. Johnson in [14], says there is much debate in how the loss of power is calculated. He states two of his preferred methods, the method chosen has a better approximation for small overlaps and is shown in (2.2.1) [14]. ΔP is the interference power (considered here as a fraction of total power) and m is the overlap fraction and is calculated in (2.2.2) [14].

$$\frac{\Delta P}{P} = \left(\frac{2}{2-m}\right)^{1/2} - 1 \tag{2.2.1}$$

$$m = \frac{2}{\pi}\left[\cos^{-1}\frac{l}{2R} - \frac{l}{2R}\sqrt{1-\left(\frac{l}{2R}\right)^2}\right] \tag{2.2.2}$$

These quantities assume a small vertical separation so that the inflow rates of both rotors can be considered the same. To calculate the overlap function, the rotor radius R is needed as well as the separation distance l.

2.2.3 Tandem Rotors

A tandem rotorcraft is sometimes referred to as a dual rotor, as it consists of two blades to generate thrust and thereby decreasing disk loading and increase the lift capacity. In a tandem configuration the blades sit in the front and the rear of the craft. Tandems are often used in applications that require heavier loads than the traditional helicopter can effectively offer. The blades spin in opposite

directions to counteract the other's rotational torque. Pitch and Yaw control are readily available through manipulation of the rotor speeds, while roll control is not as easily accomplished with this design and generally require variable pitch rotors [15]. Using two smaller blades also decreases the effects of interferences such as gusts on the craft.

As described in (2.1.8) the thrust of the system increases slower than the electrical power input into the system. In a standard configuration, doubling the electrical power would only increase the thrust by a factor of ≈ 1.587. Where as doubling the amount of rotors being driven will double both the thrust and the electrical power. This gives the tandem arrangement the capability of lifting heavier loads with relatively low power consumption, as well as demonstrating low power consumption for hover and slow translatory flight. Having twin blades does increase the size of the craft, but the elimination of the tail rotor sees the size being similar to that of a classic helicopter.

2.2.4 Multirotor Designs

Drones have joined other remote controlled vehicles in the world of hobbyists. Of all the different designs the multirotor is the most popular. The four rotor design is generally chosen due to its incredible stability and manoeuvrability. Similar to the tandem, quadrotors have very good disk loading and thus can be used to lift heavy loads, there are even products that have 8 rotors to seriously increase the payload capability. This does however relate to a more power hungry system and a less efficient hover.

As shown in Figure 2.7, a quad rotor consists of two pairs of counter rotating propellers. Each shaft will be driven by its own motor and will contribute to the overall thrust and moment generation of the craft. Having the freedom to control each blade independently gives the pilot advanced manoeuvrability, with minimal mechanical complexities. This also reduces the complexity of the control algorithms as six degrees of freedom can be obtained by simply adjusting the speed of the motors. Besides the poor hover efficiency, the biggest downside of the multirotor designs is their size and weight. Each blade requires a drive system and space to rotate without interference. This generally limits the flight time of multirotors.

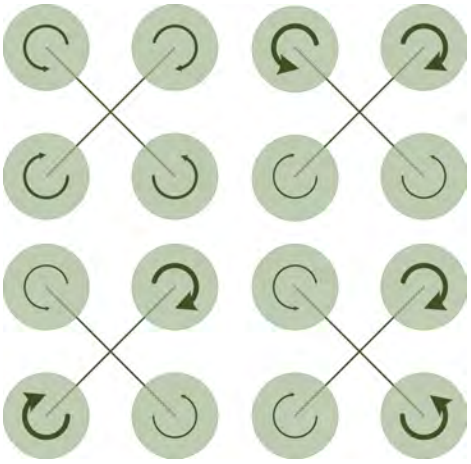


Figure 2.7: Quadrotor configuration

2.3 Quadrotor Flight Dynamics

This section will discuss some of the methods and limitations pertaining to modelling the flight dynamics of a rotorcraft. Most of the discussion will surround multirotors, specifically quadrotors, as the majority of the literature is based on these designs [16]–[19]. Due to the mechanical complexity of swash plate designs, the discussion is assuming a fixed pitched rotor set up.

Before control laws can be applied there must be a dynamic model of the craft. To create the model there must be a good understanding of the factors that effect these dynamics as well as the mathematical methods for deriving the equations. A brief introduction to the nomenclature

and axis systems is done and is followed by a discussion into modelling rotorcraft forces and moments. After the model can be obtained mathematically it is important to discuss the physical implementation of obtaining the data, and the instrumentation required. Unfortunately its very rare to have a flying environment that is void of disturbances, this section is closed with a discussion about the various disturbances that effect the flight dynamics of rotorcraft, including some specific environmental disturbances.

2.3.1 Coordinate Systems, Rotations and Nomenclature

As the rotorcraft manoeuvres through space, two separate frames are created. Each axis system is important and transforming easily between these frames is necessary. Some of these methods are described in this section, as well as the various means of representing these rotations. This section begins by describing these different frames, namely the inertial and body frames.

2.3.1.1 Inertial and Body Frame

The inertial, or North East Down (NED), frame aligns itself with the North and East directions on a compass. The third axis will align with gravity as a positive Z component. This frame assumes that the earth is flat and non-rotating and this frame's origin can be defined arbitrarily.

The body frame aligns itself with the body of the drone, with the front of the craft facing in the positive X direction and the Z axis is defined perpendicular to the rotor plane with thrust generated in a negative Z direction. The origin of the body frame is defined as the centre of mass for the drone.

Figure 2.8 is a visual representation of both frames.

Figure 2.8: The inertial and body frames

In order to relate the motion of the craft in the body frame to the inertial frame, it is necessary to be able to represent the rotation between these frames.

2.3.1.2 Euler Angles

The most intuitive way to represent the rotation between two frames, is by looking at the rotation between each corresponding axis. These are known as the Euler angles and are made up of roll (ϕ), pitch (θ) and yaw (ψ) angles. Euler angles provide a very intuitive understanding of the rotation between the different frames. This is best explained with Figure 2.9.

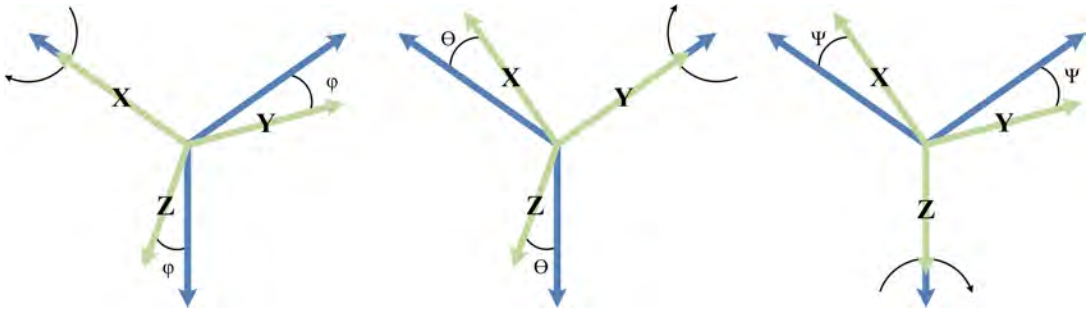


Figure 2.9: Individual rotations around the X, Y and Z axes respectively.

The yaw angle is defined as a pure rotation around the Z-Axis. Roll and pitch are defined as pure rotations around the X-Axis and Y-Axis respectively. The Euler angle representation does have limitations, such that any 3 Euler angles could represent a different rotation, based on the order it is applied. For this project, a Euler 3-2-1 sequence will be followed. There is also a chance of a singularity at extreme angles, this is not a concern for this project, as it will only be necessary to ever complete small rotations [19], [20].

2.3.1.3 Direct Cosine Matrix

The direct cosine matrix (DCM), provides a simple method for transforming vector references between two different frames. This is necessary for converting the NED frame to the body frame and vice versa. The DCM is calculated by following 3 individual rotations and multiplying their results together. A 3-2-1 Euler sequence will transform first using yaw then pitch and finally roll. Each transformation is represented as a 3x3 Matrix representing a rotation around one of the axes.

In the case of rotating from the body to the NED frame, the matrix takes the form as shown in equation (2.3.1) [16], [19] where $C_x = \cos(x)$ and $S_x = \sin(x)$. The matrix is also orthogonal, which means that $\mathbf{R}^{-1} = \mathbf{R}^T$. $\mathbf{R}^T$ would be the rotation from the inertial frame to the body frame [16], [19], [20].

$$\mathbf{R} = \begin{bmatrix} C_\psi C_\theta & C_\psi S_\theta S_\phi - S_\psi C_\phi & C_\psi S_\theta C_\phi + S_\psi S_\phi \\ S_\psi C_\theta & S_\psi S_\theta S_\phi + C_\psi C_\phi & S_\psi S_\theta C_\phi - C_\psi S_\phi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{bmatrix} \tag{2.3.1}$$

The DCM does provide a mathematically simple method for creating relationships between frames, however this method is computationally taxing as it is forced to recalculate the matrix and the multiplications on every loop.

2.3.1.4 Quaternions

The quaternion representation manages to minimise the computation required to calculate transformations, as well as remove the singularity found in the Euler representation [20]. One of the major downsides of quaternions is that they are difficult to interpret intuitively. A quaternion follows the form seen in (2.3.2) and contains a scalar value q_w and a vector component $[q_x\ q_y\ q_z]$. This representation is broken up into a rotation angle, and a rotation axis.

$$\mathbf{q} = \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix} \tag{2.3.2}$$

Quaternions come with their own set of mathematical rules and laws which will not be discussed here. However it should be noted that there are techniques that provide simple conversion from and to Euler angles and thus the DCM.

X, Y, Z	Force vector along the respective body frame axis
L, M, N	Moment around the respective body frame axis
U, V, W	Linear velocity along each body frame axis
P, Q, R	Angular velocity around each body frame axis

Table 2.1: Standard nomenclature

2.3.1.5 Nomenclature

The naming convention used, follows Moller's notation [19] and is shown in Table 2.1. It makes sense that the global position and velocity of the craft be described in the NED frame, however the forces and moments will be generated in the body frame. Since there is now a simple relationship between the two frames, it is possible to relate the body frame forces and movements, into earth frame translations. The variables shown in Table 2.1 are visualised in Figure 2.10. The variables are all defined in the body frame and are shown, along with their positive directions. The right hand rule was used to dictate direction.

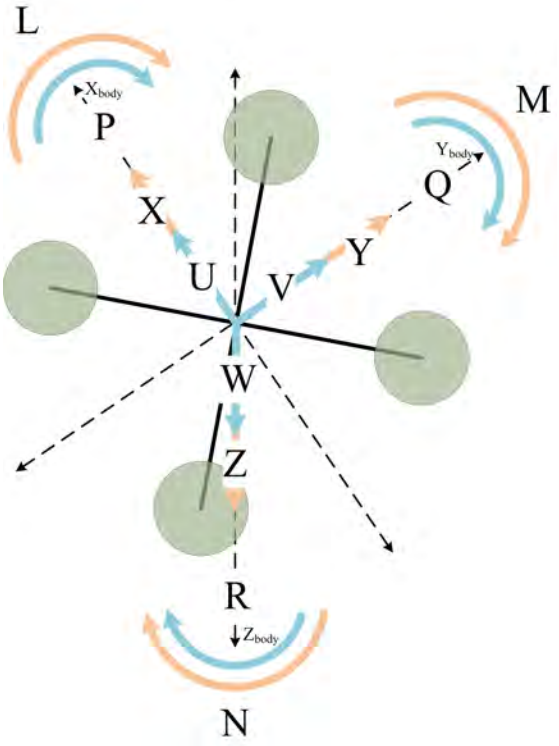


Figure 2.10: Typical naming convention of body forces, moments and velocities for a quadrotor.

The body frame forces, moments and velocities can be seen, and are described in (2.3.4) - (2.3.9). Where X, Y, Z are the forces in each body axis, with the rotor thrust being produced in the negative Z direction. L, M, N are the moments around the x, y, z axes respectively and U, V, W are the velocities aligned with the x, y, z axes respectively.

Using the rotation matrix described in (2.3.1), a relationship for North, East and Down velocities can be made and is described in (2.3.3).

$$\begin{bmatrix} \dot{N} \\ \dot{E} \\ \dot{D} \end{bmatrix} = \mathbf{R} \begin{bmatrix} U \\ V \\ W \end{bmatrix} \tag{2.3.3}$$

2.3.2 Kinetics and Kinematics

Assuming the system can be considered as a rigid body, allows for the use of normal Newtonian mechanics to create the equations of motion. This method will also use the Euler angles described above [16], [19], [21].

The derivations of these calculations are well documented in literature [22]

$$X = m(\dot{U} - VR + WQ) \tag{2.3.4}$$
$$Y = m(\dot{V} - UR + WP) \tag{2.3.5}$$
$$Z = m(\dot{Q} - UQ + VP) \tag{2.3.6}$$
$$L = \dot{P}I_{xx} + QR(I_{zz} - I_{yy}) \tag{2.3.7}$$
$$M = \dot{Q}I_{yy} + PR(I_{xx} - I_{zz}) \tag{2.3.8}$$
$$N = \dot{R}I_{zz} + PQ(I_{yy} - I_{xx}) \tag{2.3.9}$$

2.3.3 Mass Model and the Inertia Tensor

2.3.3.1 Mass Model

In any aerial vehicle, mass is always an important design criterion. Every aspect of the platform must be designed to be the lightest it possibly can. Having a light weight craft is one part of the design criterion, another would be ensuring that the weight is geometrically spread out correctly, as well as functionally distributed appropriately. The table below was adapted from [9] and demonstrates the latter point by showing the weight distribution of three separate crafts. Depending on the different criteria for the craft, different functional blocks will be allocated a certain percentage of weight. For example if the project requires a longer flight time, a higher percentage would be given to the power source and possibly less to the external payload. Generating a good mass model before designing helps better understand the requirements for the craft and could be a deciding factor in the construction.

Component	0.3kg Vehicle	1.8kg Vehicle	3.7kg Vehicle
Rotor System	11.0%	11.2%	13.9%
Tailboom Assembly	8.0%	9.1%	7.8%
Main Rotor Motor	15.4%	10.5%	8.1%
Fuselage/Structure	7.0%	15.1%	12.0%
Main Transmission	2.0%	3.4%	3.4%
Landing Gear	2.3%	3.4%	2.9%
Control System	5.7%	18.3%	9.3%
Avionics	29.4%	2.4%	1.6%
Power Source	19.2%	26.6%	41.0%

Table 2.2: Examples of MAV weight distributions (Adapted from [9])

2.3.3.2 Inertia Tensor

It was also mentioned that the weight needs to be geometrically positioned correctly, the point of this would be to create as much symmetry in the craft as possible. If this is done correctly the principle axes of inertia will align very closely with the body of the craft, simplifying calculations later on and helping find and define the principle axes. The inertia tensor is a matrix that is a representation of a rigid body's resistance to rotations in 3D space. For the general case the inertia tensor takes the form as shown in equation (2.3.10). The inertia tensor is very dependant on a craft's symmetry, and is symmetric itself. In other words, $I_{xy} = I_{yx}$, $I_{xz} = I_{zx}$ and $I_{zy} = I_{yz}$ and therefore if a craft is symmetric about the X, Y and Z axes, then the assumption can be made that $I_{xy} = I_{yx} = I_{xz} = I_{zx} = I_{yz} = I_{zy} = 0$ [16], [23].

$$\mathbf{I} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \tag{2.3.10}$$

In order to successfully model a rotorcraft, the inertia tensor must be known and will be defined around the centre of rotation of that rotorcraft. The method for obtaining the inertia tensor is described in the system identification of this project.

2.3.4 Rotor Generated Forces and Moments

The forces and moments generated by the rotors are discussed here. It is assumed that the rotors will only generate a force perpendicular to their plane while the moments are dependant on the placement of the rotors.

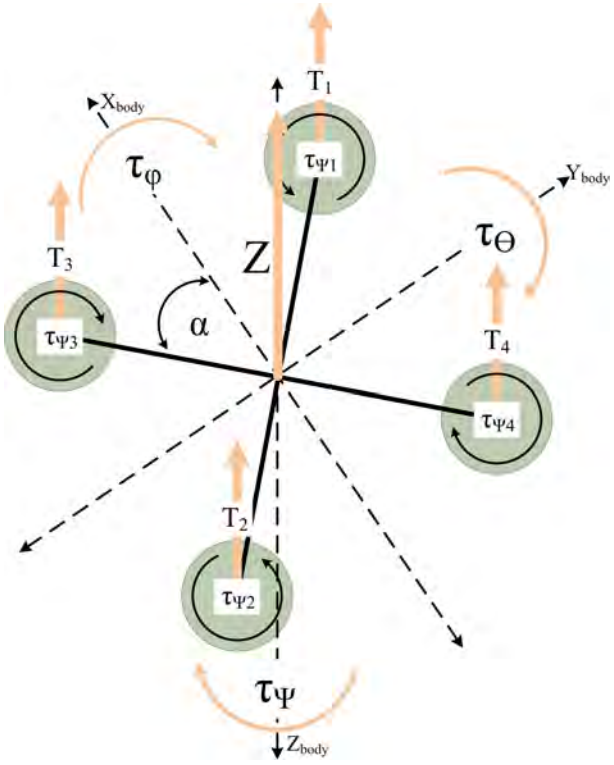


Figure 2.11: Forces and moments acting in the body frame on an X-Configuration quadrotor.

2.3.4.1 Actuators

As shown in Figure 2.11, all the forces generated by the quadrotor are a product of the four rotors. The rotors convert mechanical energy from the motors into aerodynamic power. The motors convert electrical energy into mechanical energy based on the motor commands sent from the controller. Both the motors and rotors can not react instantly to new commands, this lag introduces a timing delay constant into the system [19].

If the lag timing constant is defined as τ, thrust generated by motor x as T_x and the command sent to that motor as T_{xR}. Then (2.3.11) can be created and applies to all four motors.

$$T_x = -\dot{T}_x \tau + T_{xR} \tag{2.3.11}$$

2.3.4.2 Controlled Body Forces

Figure 2.11 assumes that all of the rotors lie in the same plane, and only provide a unidirectional force. This assumption allows the easy creation of a total force Z, which is shown in (2.3.12) as the sum of all four motor thrusts.

$$Z = (T_1 + T_2 + T_3 + T_4) \tag{2.3.12}$$

To command this value, a virtual actuator can be created δ_Z which commands all four rotor thrusts. Equation (2.3.11) demonstrates the lag to generate these thrusts and the same lag dynamics will apply to δ_Z, thus creating (2.3.13).

$$Z = -\dot{Z}\tau + \delta_Z \tag{2.3.13}$$

2.3.4.3 Controlled Body Moments

A quadrotor generates a moment around its own axes through varying the speed of each motor. The torque generated is also dependant on the spacing for the type of quadrotor used. A standard X-Configuration quadrotor is shown in Figure 2.11 and was used for this analysis. To induce a torque around the X-axis, the sum of the two left rotors subtracted from the sum of the two right rotors must be non-zero. Similarly the front and back rotor summations must not be equal to induce a torque around the Y-axis. As shown in 2.11, each rotor also creates a moment around the Z-axis. This induced torque is a product of the rotors lift to drag ratio and the chord length and is represented in (2.3.14).

$$\tau_{\psi x} = \frac{r_D}{R_{LD}} \times T_x \tag{2.3.14}$$

Assuming that each rotor has the same characteristics and are spaced evenly, these moments can be mathematically expressed as shown in (2.3.15) - (2.3.17), where l is the distance from the centre of the rotor to the centre of gravity, r_D is the chord length and R_{LD} is the lift to drag ratio for the rotors.

$$\begin{aligned} L &= \frac{r_D}{R_{LD}} \times (T_3 + T_4 - T_1 - T_2) && (2.3.15)\\ M &= (T_1 + T_3 - T_4 - T2) \times lcos(\alpha) && (2.3.16)\\ N &= (T_2 + T_3 - T_1 - T4) \times lsin(\alpha) && (2.3.17) \end{aligned}$$

Virtual actuators can be created to command these moments, namely δ_ϕ, δ_θ and δ_ψ. However these commands will be subject to the same time delay experienced by the rotors. Therefore (2.3.18) - (2.3.20) can be used to represent the relationship between these commanded values and the actual moment [19], [21].

$$\begin{aligned} L &= -\dot{L}\tau + \delta_\phi && (2.3.18)\\ M &= -\dot{M}\tau + \delta_\theta && (2.3.19)\\ N &= -\dot{N}\tau + \delta_\psi && (2.3.20) \end{aligned}$$

2.3.5 Disturbances

2.3.5.1 Drag

Drag is a damping force whose quantity is relative to the speed of the object, and always opposes the direction of motion. Drag is defined here in the body frame and the equations for three

dimensional drag can be calculated. As shown in (2.3.21) - (2.3.23), the effect of drag can be reduced through mechanical design and flight strategy, by reducing the area of the plane facing towards the direction of the motion.

$$F_{Dx} = C_{DX}(\frac{1}{2}\rho U^2)A_{YZ} \tag{2.3.21}$$

$$F_{Dy} = C_{DY}(\frac{1}{2}\rho V^2)A_{XZ} \tag{2.3.22}$$

$$F_{Dz} = C_{DZ}(\frac{1}{2}\rho W^2)A_{XY} \tag{2.3.23}$$

Due to an offset between the centre of gravity and the centre of pressure, the drag forces can also create undesired moments. Equations (2.3.24) - (2.3.26) can be derived from the Figure 2.12, where d_x, d_y, d_z are the offsets of the centre of pressure. F_{Dx}, F_{Dy}, F_{Dz} are the forces generated by drag acting along the coinciding body axis. M_{Dx}, M_{Dy}, M_{Dz} are the moments generated by the drag forces and the offset of the centre of pressure, they act around the coinciding axis. A_{YZ}, A_{XZ}, A_{XY} are the surface areas facing the corresponding plane in the body frame with C_{DX}, C_{DY}, C_{DZ} as the corresponding drag coefficients.

$$M_{Dx} = F_{Dz} \times d_y - F_{Dy} \times d_z \tag{2.3.24}$$

$$M_{Dy} = F_{Dx} \times d_z - F_{Dz} \times d_x \tag{2.3.25}$$

$$M_{Dz} = F_{Dy} \times d_x - F_{Dx} \times d_y \tag{2.3.26}$$

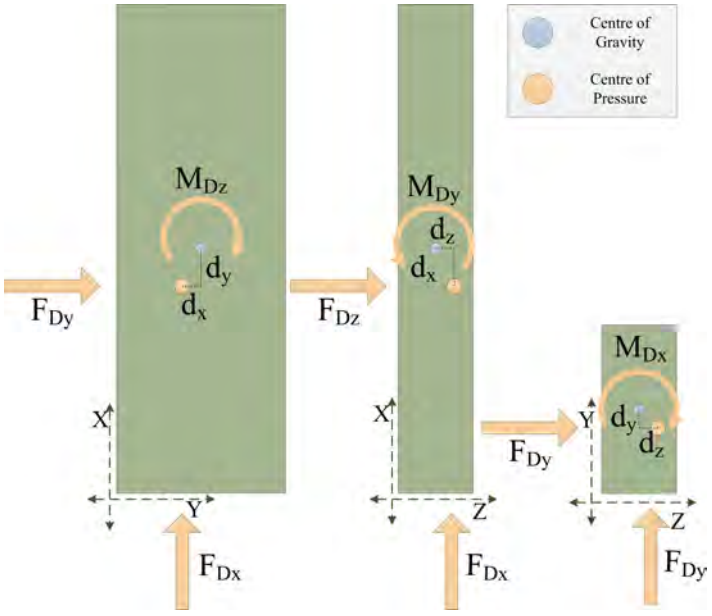


Figure 2.12: Typical moments created by drag forces

2.3.5.2 Airflow Characteristics

In the preceding section on flight theory, the importance of air density, pressure and the creation of rotor wake boundary are discussed. The negative effects of disrupting airflow as well as the need for controlling this disturbance has been well documented in literature [2], [18], [24].

Using Figures 2.1 and 2.2 as references Airflow can be seen as the stream of air from v_0 to v_∞ through v_i. Equation 2.1.4 states that thrust is directly proportional to the relationship between these velocities and any deviation in these velocities will vary the thrust of the rotor in question.

v_0 is only zero when the craft is in a state of pure hover, completely stationary, and there is no wind. Increasing the speed of the craft will increase the v_0 component creating a variation in the overall thrust, the same can be said for any condition that contains a tangible wind factor.

As investigated by [18] mechanical intrusions will have an effect on the far wake velocity, thus also effecting the generated thrust. In the design of STARMAC by Hoffmann et al [18] the frame was designed to be very configurable so that the effects of the mechanical design could be quantified. Originally the rotors were shrouded and quite close to the centre of gravity of the craft. The shrouds were a distance of 5% rotor radius and when removed the yaw tracking improved from $\pm 10°$to $\pm 3°$. When not included in the dynamic model the obstruction in the air stream will cause lower and less stable values of thrust, affecting the accuracy of the model.

When the rotors were located close to the centre of gravity they obtained some attitude disturbances that were eliminated by moving the rotors further away. It was also observed that any object that lies close to the rotor tip, created intense arbitrary disturbances and should be avoided [18].

[2] attempts to model some of the disturbances for a single rotor craft hovering near wall, but as stated by [24] it is not viable to accurately quantify these disturbances, however their presence must not be neglected. As Figure 2.13 shows, these can be modelled as a disturbance to the input force and moments.

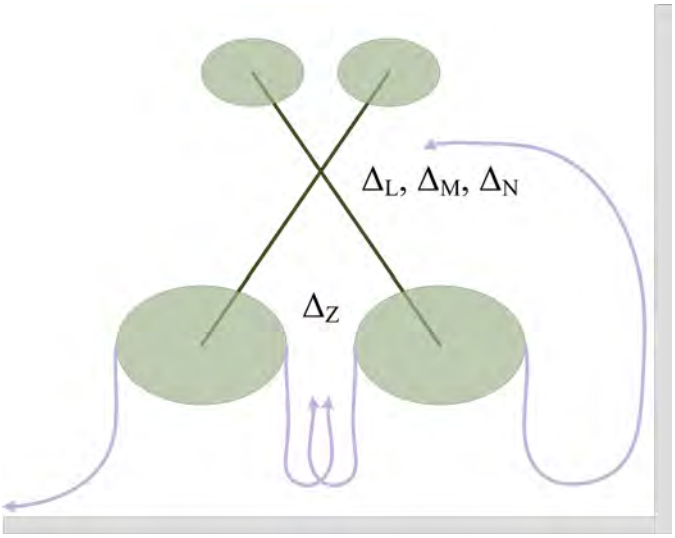


Figure 2.13: Disturbances created by being in close proximity with a wall

As demonstrated by [2], there is also an induced moment acting on the rotor as the rotor approaches the wall. Figure 2.14 is an image generated by [2], it demonstrates the change in airflow on a rotor close to a wall region.

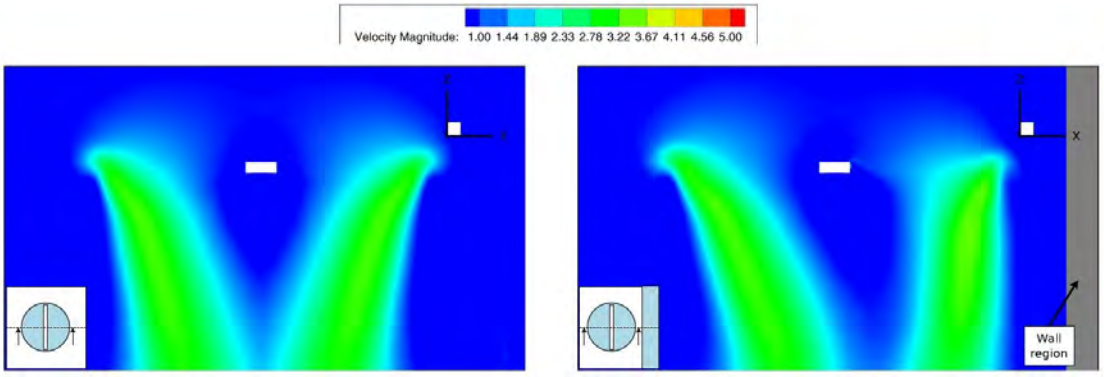


Figure 2.14: Velocity components though the rotor for, no wall (left) and near wall (right) conditions (Taken from [2])

In [2], Robinson et al used the script c as their unit of measure for distance to the wall, c is chord length of the airfoil. The graph shown in Figure 2.15 shows how the moment felt by the craft varies with the distance to the wall. The direction of the moment will be perpendicular to the wall. As above, this disturbance can also be modelled as a variation to the input moments to the system.

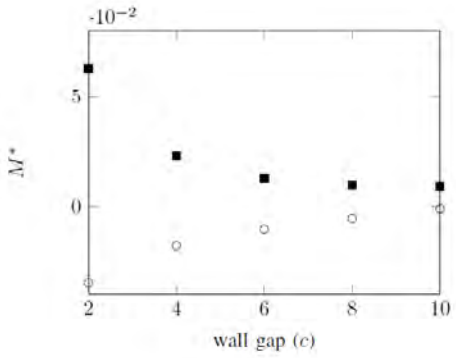


Figure 2.15: Graph showing relationship between distance from the wall and moment felt be the craft (Taken from [2])

In [2] they conclude their paper by describing a proposed method using a disturbance based controller, which will be investigated further in this text. This dynamic flight model requires sensing of specific craft variables, typical sensing methods and requirements are discussed in the following section.

2.3.6 Instrumentation

The need for a well instrumented craft is intuitive and well documented in literature [18], [19]. With modern advancements in sensing technology there is now a number of methods to obtain the required information. However, some of these are very costly and require specific operational environments. For this purpose, only the inertial measurement unit (IMU) and the global positioning system (GPS) are discussed here.

2.3.6.1 Inertial Measurement Unit

Traditionally an IMU will consist of an accelerometer, gyroscope and possibly a magnetometer. The accelerometer has the ability to measure accelerations in the body axis. It should be noted that an accelerometer in rest, sitting along the inertial axis will provide a reading of $[0\ 0\ g]^T$. The gyroscope measures the rotational accelerations of the craft, thus measuring 6 degrees of

freedom in total. Unfortunately gyroscopes suffer from sensor drift and needs compensation. The accelerometer's gravity offset can be used to align with the vertical axis, however a magnetometer is required to account for drift in the horizontal plane. Various filtering and fusion techniques are used to combine these readings, the most popular of which is a Kalman Filter [18], [19].

2.3.6.2 Global Positioning System (GPS)

The most common method of measuring a drone's position is with the use of a GPS. The GPS readings can also be used to create an estimate of the craft's linear velocity in the inertial frame. When combined, multiple GPS units can be used to obtain higher precision. A major downside of a GPS is the dependence on a satellite link, this dependence severely limits the operational environments for this sensor. Other methods of localisation include using known maps and proximity sensing to provide the rotorcraft with position estimates. As discussed, these techniques are out of scope for this project and will not be considered. This project will assume an earth position and velocity estimate are present, subject to filtered noise provided through band limited white noise [19].

2.3.6.3 Measurement Noise

From the discussion it is evident that sensors will not be without error. The noise associated with the measurements will be different for both varying sensor types and manufacturers and will depend on the chosen sensors. The characterisation of the noise for this project is discussed in the system identification chapter. The environment in which these sensors operate also has a significant effect on operation. Moller characterises this measurement noise as a random band limited white noise block and adds a low pass filter to the GPS measurements [19].

2.4 Review of Existing Flight Control Strategies

The object of this section is to gain a better understanding of successful controller designs and implementations. A talented pilot could fly a rotorcraft with minimal assistance from an on board controller, however to successfully create an autonomous vehicle capable of flight in a confined environment, a robust controller design is required. The discussion is broken into two sections, first a look into high level controller architectures is done after which a look into some successful real world implementations is investigated. The different axis systems used are discussed in a previous section and transition between these axes becomes a crucial part of the controller architecture.

2.4.1 Controller Architecture

Traditional tracking control for a rotorcraft is distributed between three controllers, namely an altitude, heading and horizontal controller [19], [25]. The specific loop structure is determined by which feedback is available to close the loops. These loops are made up of combinations of traditional control laws such as PID controllers and Lead-Lag compensators.

The altitude controller is straight forward, dynamics are derived directly from Newton mechanics. A traditional structure is the use of three cascaded loops. The most inner loop can control the linear acceleration of the craft in the body frame and its loop can be closed by use of an accelerometer. The next loop is used to control the climb rate of the craft in the earth frame, with the final loop controlling the final position of the craft. The altitude loop is closed by some form of altimeter, traditionally a barometer can be used to estimate altitude. For more precise altitude control a sensor can be used to give a relative height above the ground. The velocity loop is generally closed by some approximation of velocity through differentiation of absolute position. In some cases optical flow sensors could be used to obtain a velocity measurement directly.

The heading controller is responsible for controlling the orientation of the craft. Traditional quadrotors can fly omni-directionally irrespective of their heading, allowing for less stringent design requirements on this controller. Orientation is traditionally described as an Euler angle off of a given axis obscuring the axis systems and need for rotations. The heading controller can be simply broken into two loops, an inner rate loop and an outer angle loop. Both of which can be closed by the use of an IMU combining magnetometers and gyroscopes.

The final and most complex controller structure is the horizontal control. This is the only controller that is required to control both angular and translational control loops introducing the need for inferring desired angles from a given translational reference. The controller generally is also required to work in both axis systems, requiring some form of geometric transformation. The translational portion of the controller is split up into North and East controllers, while the inner angular system is broken up into roll and pitch controllers. The translational portion is further broken down into an outer position loop which feeds setpoints to an inner linear velocity loop. Both of these loops can be closed by a GPS unit. The angular portion of the controller is then also split into an inner rate loop which is potentially closed by a gyroscope and an angular position loop which utilises feedback from an IMU.

Tracking control is used to obtain stable flight and reject disturbances and unmodelled errors. However certain disturbances might hinder the performance of these controllers and produce instability that traditional control law can not account for without negatively affecting the tracking control. Feed forward control could be utilised if the disturbances can be predicted and their effects have been quantified, this however is rarely the case.

2.4.1.1 Disturbance Based Observer Controller

Another method of rejecting extreme disturbances, while limiting the adverse effects on the tracking control, is to estimate the disturbance and then control based on that estimation. In an investigation of the near wall effects, Robinson et al propose a disturbance based observer which can be used to predict and counteract these disturbances [2], [26]. Figure 2.16 lays out the structure of such a controller, and demonstrates how it is interfaced with traditional tracking control. The main benefit of this structure is the loop is only activated when the state estimations do not match the state outputs.

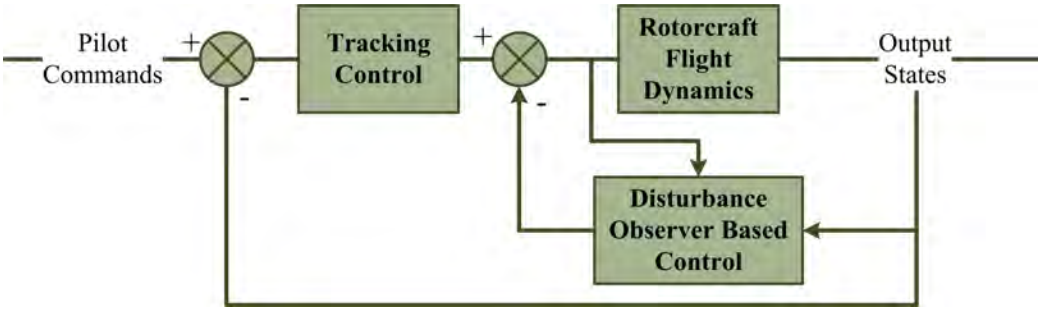


Figure 2.16: Disturbance observer based controller structure [2]

For a system like this to be effective, the observer will need to measure the disturbances in real time requiring fast computation. Lee et al implemented an adaptive control strategy which uses an IMU and a Vicon system to estimate the disturbances and feeds them into the system [24]. This requires a complex lab set up and can not be used in any environment, however their experiments did validate the effectiveness of such a system.

2.5 Collision Protection and Avoidance Techniques

This section discusses the implementation of collision protection for an aerial vehicle. The discussion looks at various techniques that can be namely broken up into collision protection and collision avoidance. Collision protection is an addition to any platform that will assist during an impact. An advanced enough collision protection technique could negate the risk of impact in a confined environment. Any obstruction in a rotor's path will either destroy the part, or stop it from spinning thus disabling the thrust generation capacity of that rotor. Collision protection in its simplest form is a simple shroud that can protect the rotors. Collision avoidance is a technique that require additional sensors that enables the vehicle to navigate away from obstacles and avoid collisions altogether. Collision and obstacle avoidance sensing techniques can also be used to aid

in a higher route planning algorithms. For successful collision avoidance, the craft must be able to collect data about its surroundings.

This section begins by describing some of collision protection techniques. The sensing requirements and techniques for collision avoidance are then discussed followed by some literature on collision avoidance and path planning algorithms.

2.5.1 Impact Resistance

Impact resistance is a technique used by a variety of fields in the world today. Included in this is something as simple as the shocks or suspension in a car, they are designed to allow the automotive to withstand sudden impacts. Generally these techniques use a component that has some tangible spring constant that can be used to absorb the energy felt during an impact.

Klaptocz et al investigated such a system for an aerial vehicle [27]. Using a combination of drop tests and forced collisions it was validated that an impact resistance shroud could limit the force felt by the platform. Light weight materials can be used to create the design however there inevitably will be additional weight and size added to the platform.

2.5.2 Rolling Cage

Although impact resistance can limit the effect of a collision Briod et al set out to create a system truly robust to collisions [3], [13], [28]. Using a coaxial rotorcraft they developed a system whereby, using a three axis gimbal, an external shell will passively rotate around the craft on impact. Thus almost negating the effect of the collision completely. Although a proven solution to confined indoor flight, the system is optimised for a coaxial rotorcraft limiting the payload and flight time capabilities of the design.

2.5.3 Sensing Techniques and Requirements for Collision Avoidance

Traditionally the collision avoidance sensing techniques can be broken into two main categories. Namely some form of proximity measurement can be done, or a higher level more intelligent image processing system can be applied [29]. This discussion analyses each of these designs separately.

2.5.3.1 Proximity Measurement

The first and most common would be using a time of flight (TOF) sensor, such as an ultrasonic transducer. Very similar to bats, a transmitter module will emit a pulse and based on the time it takes for the signal to return, the distance can accurately be measured. A sonar is dependant on a lot of variables and would require calibration for the environment it is in. Due to the method with which the modules acquire information the drone will receive a speed limitation. Since the sonar is dependant on the density of the medium it travels through, disturbance created by the rotors could severely affect the performance of the sensors.

Another option in the same category would be to use an infra-red (IR) transceiver in the same configuration. The system uses strobed IR pulses to monitor the distance of an object, the system depends on the ratio of reflection for the IR spectrum. If an object has a high refraction or absorption ratio the signal can get lost. Ambient light can also cause interference, although most modules will have some form of built in filter. Both these methods are also sensitive to the angle at which they are measuring.

[30] demonstrates the usefulness of combining both ultrasonic and infra-red measurements.

Using the same base technology TOF cameras have been developed such as Microsoft's Kinect. Instead of a single point, a TOF camera projects an IR grid into the area it wishes to explore. By measuring deviations in the expected grid structure, distance information can be inferred. In an IR rich environment, TOF cameras can get overwhelmed and produce unreliable results. Due to the projection, the power requirements and weight of such a system can also be high.

2.5.3.2 Image Processing

The term image processing has been used here to discuss a system that can extract data through a video feed. The process is often referred to as machine vision and can be trained to detect objects present in the image feed. A general method is to do edge detection, allowing the craft to differentiate between objects and safe flight zones. These configurations generally require a lot of processing power and can add considerable weight and electrical power requirements to an aerial vehicle.

Although more processing is required, very good results can be obtained from this set up. Generally these systems are slow and have a high power draw due to the multiple required modules. However, reliability from a robust image processing unit will be more reliable and consistent than a different system. Barry et al developed a system that minimises some of these negative effects [31]. Using this design Barry could obtain object avoidance at speeds of 31mph, proving the effectiveness of the object avoidance system.

It is also well known that a camera feed can provide other advantages, such as allowing the pilot to fly without line of sight. Stereo vision can also produce 3D information from multiple 2D images, creating useful data for the operators. Optical flow algorithms can also be applied to a camera feed to estimate the current velocity of the craft. A major downfall of optical systems is the requirement for well illuminated areas. A dimly lit area, or even a dust filled environment could produce faulty results hindering the performance of the collision avoidance system.

2.5.4 Collision Avoidance and Path Planning Algorithms

The theory behind obstacle avoidance algorithms is well documented and understood in literature. An obstacle avoidance routine will allow the craft to deviate off of the planned motion in order to avoid collisions. A successful implementation will attempt to limit this deviation as much as possible. Some of the main considerations for these algorithms are the amount of computational power required and the speed at which these computations can be done, as well as the required sensing information.

Some of the typical sensors used in collision avoidance have been described above. A combination of these sensors can also be constructed to give even more robust and reliable results. The investigation into existing collision avoidance algorithms can be limited by proposing a chosen sensing technique. Due to the complexities involved in image processing implementations, this work will only investigate techniques using traditional proximity measurements.

2.5.4.1 Bug Algorithm

The bug algorithm was developed by Lumelsky in [32]. The principle behind the bug algorithm is to create a desirable path to the set point, referred to as the main line (M). The algorithm allows the robot to deviate off of the desired path based on immediate proximities to objects. The new path will follow the obstacle boundary until the main line is found again. The robot will continuously create new points along the path that either fall on the main line or the obstacle contour. This allows the robot itself to contour around an object while still attempting to get back onto the correct path. Once the path is found, the robot will continue along, until another object is detected. The Bug2 algorithm uses a state machine defined below.

1. Move along the straight line segment drawn from start point to end point until one of the following occurs
    a) The final point is reached and the mission is completed
    b) An obstacle is detected and the location is stored, the algorithm then moves on to step 2
2. Follow the obstacle contour until one of the following occurs
    a) The final point is reached and the mission is completed

b) The main line is met again at a distance to the end point less than that of the original divergence and the new path does not cross an obstacle. Return to step 1.

c) The robot returns to the original point of divergence indicating that the target cannot be reached. The mission is stopped.

Figure 2.17 is an image showing the Bug2 algorithm at work with the craft being commanded from location S to location T. As shown, when the vehicle is in close proximity to an object it will create a contour around it until the desired path is rediscovered.

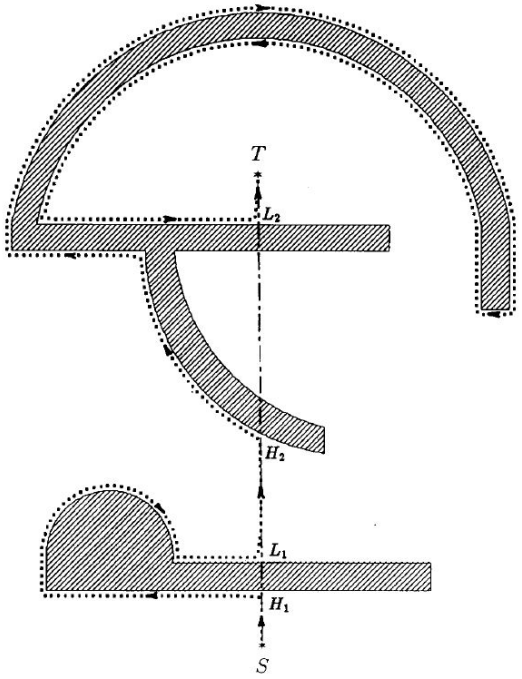


Figure 2.17: Demonstration of the Bug2 algorithm (Image taken from [32])

This method is useful, but limited. The robot will not attempt to optimise the route and this can be an inefficient method of solving the problem. Subsequently the bug algorithm has been developed further and multiple variations in the principle have been adapted. These adaptations are based on the bug principle, however utilise more advanced contour creation and state transitions, allowing for more versatility and optimised route planning.

2.5.4.2 Potential Field Method

Another widely used and documented algorithm is the potential field method [33]. This method works on the basis of creating attractive potentials, in the form of goals and unattractive potentials, which can be the obstacles or no fly zones. A simple way of describing this method is to make a comparison with a charged particle moving through a potential field. In this case, the aircraft is considered a positive charge and the end goal or position is considered a negative charge. The goal will provide an attraction force based on the robot's proximity to the end point. Obstacles, in this analogy, can then be thought of as other positive particles which will produce a negative, or repelling force to the platform. The algorithm will then calculate the attractive and repelling forces based on immediate sensing data. The sum of all of these forces can then be given to the craft and used as a command to allow for successful navigation in an unknown environment.

An important consideration is the generation of these forces based on simple proximity measurements. Equations (2.5.1) - (2.5.3) are used to represent typical implementations of these calculations [33]. Traditionally there will be a single goal which will attract with a parabolic like force, that increases quadratically as the robot moves further away from the goal. This set up will also

ensure that the force felt by the robot decreases as the setpoint is approaching. Equation (2.5.1) represents the attractive force with $K_{attract}$ as a scaling factor and d_{goal} as the distance from the set point. Although the platform can expect one goal, there might be multiple objects in its vicinity all creating repelling forces seen in (2.5.2).

The equation assumes that the obstacle influence threshold (d_o) is less than the minimal considered distance (d_{obs}). K_{repel} is the repelling scaling factor. q is the current position of the aircraft with q_{obs} the obstacle location. The repelling force is then calculated by summing all of these forces from k number of obstacles, as seen in the creation of the total force in (2.5.3).

$$F_{attract} = K_{attract} \times d_{goal}^2 \tag{2.5.1}$$

$$F_{repel} = K_{repel} \times (\frac{1}{d_{obs}} - \frac{1}{d_o}) \times \frac{1}{d_{obs}^2} \frac{q - q_{obs}}{d_{obs}} \tag{2.5.2}$$

$$F_{total} = F_{attract} + \sum_{n=1}^{k} F_{repel(n)} \tag{2.5.3}$$

A major limitation of the potential field method is the assumed holonomy of the platform. A non-holonomic vehicle will struggle to implement a potential field method as the direction of the induced forces may not be reachable for the platform in given scenarios. Another downfall of this method is the local minima created when the sum of repulsion forces equals to the sum of attractive forces. Figure 2.18 is used to illustrate a scenario where this local minima is created. Careful sensor placement can limit the occurrence of these situations, however it will be virtually impossible to remove this risk completely.

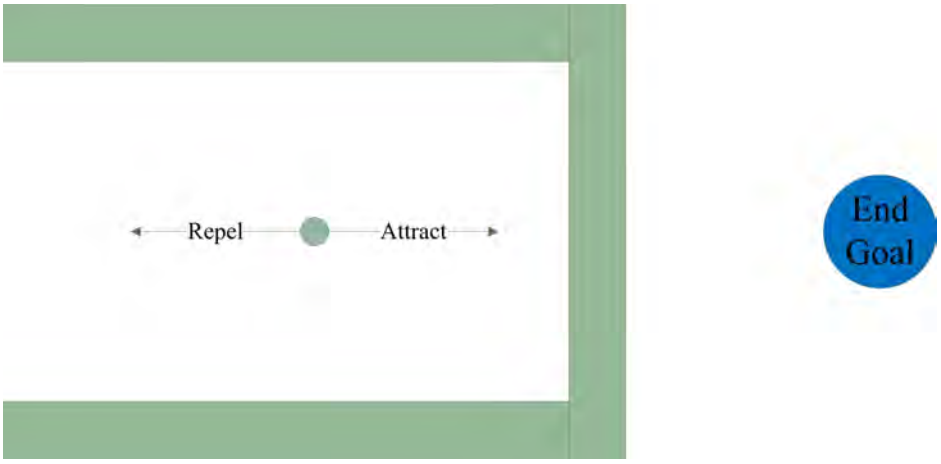


Figure 2.18: Local minima seen with potential field method

2.5.4.3 Path Planning Algorithms

The final step to achieve a fully autonomous platform would be the integration of a more robust, optimised path planning algorithm. Such an algorithm would feed setpoints into the platform's tracking control. Although not in the scope of this work it is important to understand the function and requirements of these algorithms. Many studies involving motion planning have been done and generally include an overview of some more popular techniques [34]–[36]. Two popular groups of algorithms are Grid Search and Sample Based algorithms.

As the name suggests, a grid search algorithm creates a grid-like configuration space for the robot. As the robot occupies a grid point it evaluates available adjacent spots until the end point is met. Each transition to a new block is assigned a cost and thus each route to the goal can be given a total tally, allowing for the optimal path to be chosen. For low dimensional problems the grid based solutions work adequately, however can become computationally taxing as the growth of

grid points increases exponentially as the path progresses. One of the most popular grid search algorithms is the A*, introduced in 1968 [37]. The A* algorithm is characterised by the inclusion of a heuristic to optimise the search algorithm.

Sample based algorithms are currently the leading innovation in motion planning. These methods manage to solve the problem of higher dimensional problems by only considering a known number of random sample points. The relatively low computation required allows for real time implementation of these algorithms.

Two of the most recognised and utilised sample based methods are the Probabilistic Roadmap Method (PRM) [38], [39], and the Rapidly-exploring Random Tree (RRT) [40].

Unlike A*, the PRM algorithm generates a random node in the configuration space and will attempt to link this node to its closest neighbours. This algorithm requires a method of collision detection to ensure two nodes can be linked via a non-collision path. Once the configuration space has been sufficiently sampled, the start and end nodes will attempt to link to the closest nodes, thus completing the roadmap. Once the roadmap is complete a path can be found using any graph based planning algorithms.

Where the PRM algorithm creates a roadmap of the sample space, the RRT method creates a tree of nodes by linking each new random node to only one parent node in the existing tree. The travel distance between nodes is limited by a configurable parameter in the algorithm. Another benefit of the RRT is the ability to grow two trees simultaneously, one from the start point and one form the end node. Both the PRM and the RRT are probabilistically complete, which means that as the number of nodes tends to infinity the probability of finding the goal tends to one, so long as a route exists. Figure 2.19 demonstrates the basic output of an RRT algorithm implemented on a low resolution grid.

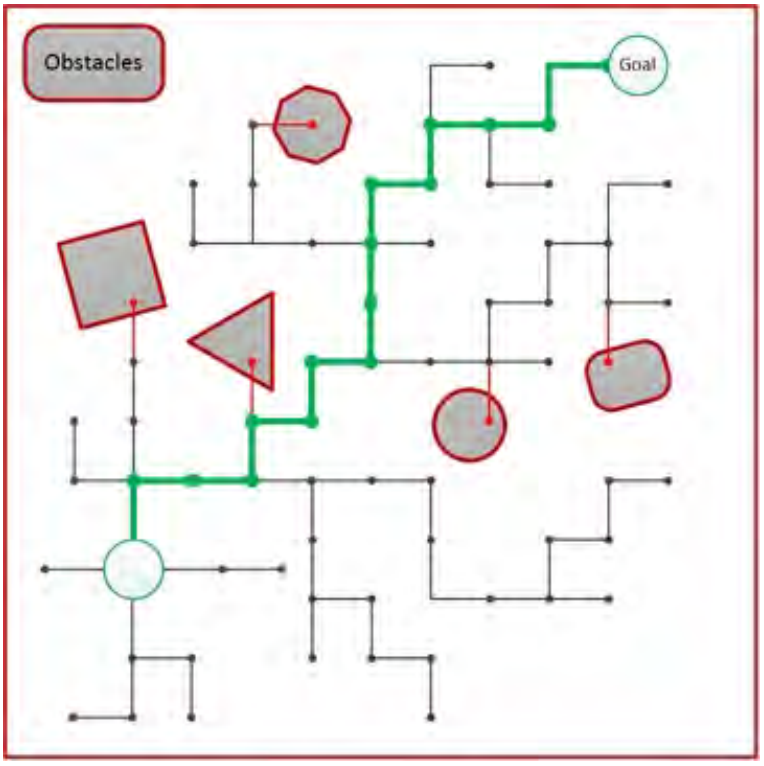


Figure 2.19: Example of RRT on a low resolution grid

Chapter 3

Platform Design

This chapter contains the details of a vehicle for use in a confined space. The design considerations of the vehicle are first discussed before deciding on the chosen platform. The hardware design decisions are preceded by giving an overview of the total system architecture design. After the high level system design has been done, this chapter looks into different platform designs and rotor configurations. Based on analysis of empirical data and knowledge learnt through an analysis of conventional rotors in Section 2.2, the best suited rotor configuration is identified. Once a suitable platform has been chosen, a look into the required electronics is done. This includes all the electronics required for stable flight, as well as sensor requirements for operations and flight strategy. The chapter is concluded with a summary of the platform decisions made and used in the simulation.

The final design will be modelled and used to validate the proposed outcomes for this work. Although this chapter will discuss the various aspects of a rotorcraft, it will focus on the components required for creating an accurate simulation of the system.

3.1 Design Considerations

Before an appropriate design can be done, specific design criteria need to be outlined; this includes parameters such as the required flight time, physical size of the vehicle, expected disturbances and manoeuvring decisions.

3.1.1 Physical Restrictions and Requirements

One of the major problems CECAD will have to overcome is to navigate these unknown areas and not only survive collisions, but also reject the disturbances introduced by being close to the walls. Since mine shafts are predominantly long and narrow, the same approach will be looked at for the design of the platform. To optimise the size of rotors that can be used the platform will also be designed to be long and narrow. Since the vehicle will be required to navigate very confined spaces, the smaller the vehicle the better. The minimum size of the vehicle is limited by the need for adequate flight time as well as payload capacities which require a higher disk loading.

3.1.2 Manoeuvring Decisions

The manoeuvring decisions are dependant on the type of environment and type of missions required of the platform. These decisions influence the final design of the vehicle. The end use case for CECAD will include mapping of partially-known areas. To complete this, it will simplify the procedure if the vehicle keeps its orientation during flight. Due to the nature of the environment, fast speeds will not be used regularly. Therefore CECAD will be designed to have slow, steady and controlled movements.

3.1.3 Disturbances

Apart from the difficulty of navigating and manoeuvring through a partially-known area there are other disturbances introduced into the system. Some common disturbances found were identified in Section 2.3.5. Due to the nature of the tunnels, wind gusts are created that funnel through these passageways. These winds will produce large undesired drag moments and forces. The effect of coming close to a wall or floor has been discussed in the literature study. Since the areas will only be partially known and complex, collisions and bumps are extremely likely. The flight strategy will try and ensure the drone does not collide with the environment.

3.1.4 Thrust Overhead

The total thrust overhead of a rotorcraft is a percentage above the thrust required for hover. This value determines the vehicle's ability to manoeuvre, with a higher value giving it more freedom and a greater ability to resist disturbances. With these benefits the system does become very sensitive and more difficult to control and stabilise. Since the vehicle will be in confined narrow passages, the vehicle does not need to be fast moving, rather a "slow and steady" pace will be approached. The vehicle does however need enough power to counteract the disturbances described above. These considerations lead to a value of 150%, with 100% being enough thrust to hover with a payload.

3.1.5 Flight Time

Flight time is dependant on efficiency and power requirements of the system as well as the capacity of the on-board power source. This becomes a typical optimisation problem. By adding a larger power source the weight is increased and therefore the platform requires more power to keep itself aloft. Weight is a determining factor for any aerial system and influences flight time, for this reason the weight of every subsystem must be optimised. To ensure the vehicle can complete a mission it will need sufficient flight time. Initial discussions with the CSIR have set 30 minutes as the bottom limit. The original platform might not be able to reach this goal, but once the platform is performing adequately adjustments can be made to the system to optimise weight and power consumption.

3.1.6 Discussion

The above specifications led to the decision of using a multirotor. The high disk loading, manoeuvrability and disturbance rejection of a multirotor make it the ideal platform for this design. The chapter continues the discussion on the basis of a multirotor design. The system design section describes a typical system architecture for a multirotor platform.

3.2 System Hardware Architecture

The following section gives an overview behind the system design of CECAD . The section begins by working through the hardware system architecture shown in Figure 3.1 and identifying the objectives and roles of each subsystem in the design. The first consideration is the size and mounting needs of the system, provided by the mechanical platform. This gives a good start to elaborate on the needed capabilities of the proposed method of propulsion. The next considerations are the electronic modules required to run all the necessary peripherals and obtain all the required sensor data.

The system architecture for CECAD is laid out below in Figure 3.1. The hardware system comprises of 3 main subsystems, namely Mechanical Construction, Propulsion and Electronics. The role each subsystem plays in meeting the system requirements is discussed in more detail below.

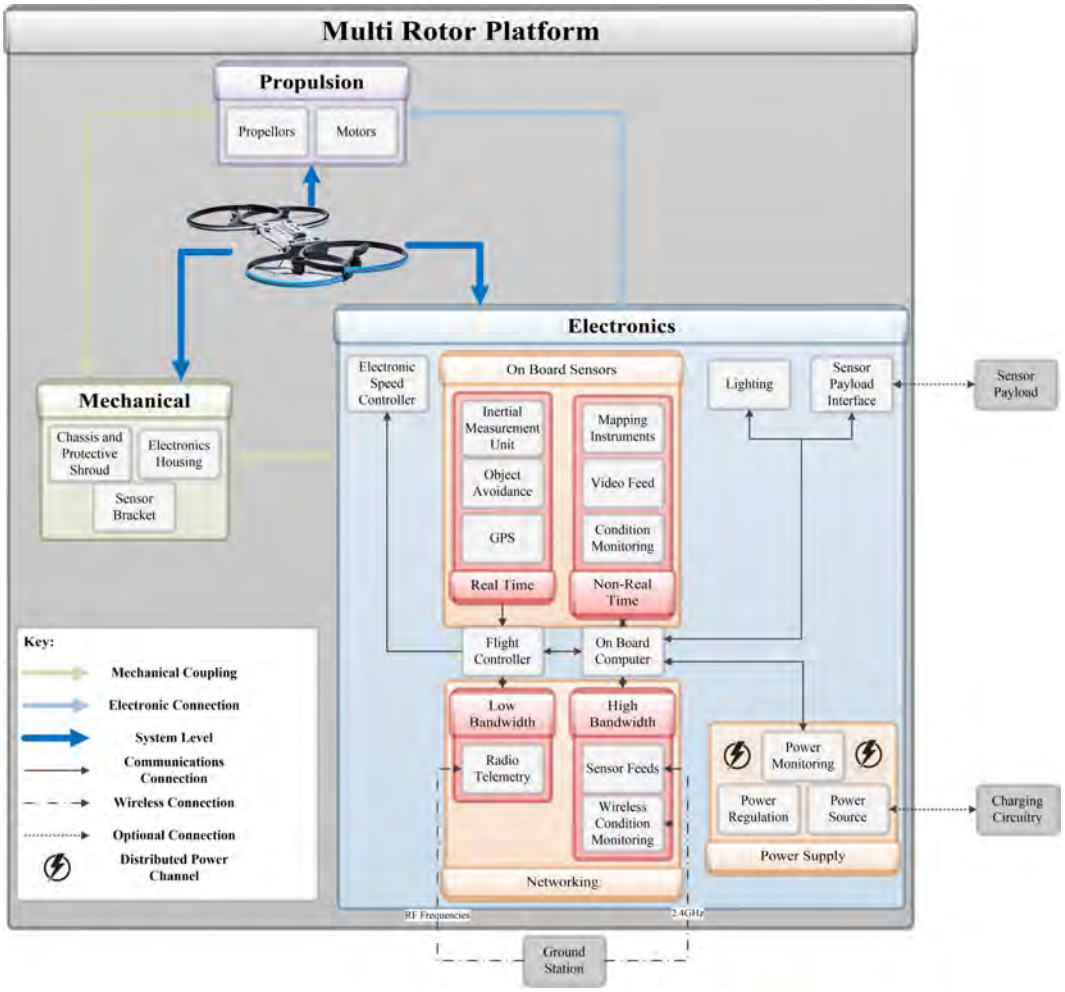


Figure 3.1: System architecture

3.2.1 Platform Construction

The platform is the physical embodiment of the vehicle. In this context this includes the mechanical construction of the vehicle, as well as the flight generating components.

3.2.1.1 Mechanical Construction

The mechanical construction of the system introduces size and weight restrictions for the rest of the system. It should provide adequate mounting for the necessary peripherals described in the sections following, while remaining as light weight as possible. The structure needs to protect the rotors from collisions by means of a protective shroud and include lightweight landing gear. The chassis must also be such that the weight is distributed as symmetrically as possible. The propulsion system must be housed and provided with rigidity for steady flight control.

3.2.1.2 Propulsion and Flight Characteristics

The propulsion system comprises of the motors and propellers. This subsystem needs to supply enough thrust and overhead for steady control, as well as enough lifting capacity to carry any additional, mission sensors. The craft should be able to handle an external sensor payload. To ensure capable disturbance rejection, the total maximum thrust should include sufficient overhead above the hover requirement.

The craft must efficiently hover and fly at low speeds. The craft should have multiple control surfaces to accurately counter disturbances. Due to the intended use case in narrow corridors, the

vehicle will fly with a set orientation and thus does not require the abilities for sharp turns of manoeuvres.

3.2.2 Electronics Interface

At the core of any reliable robotic platform is a thoroughly planned avionics system, for CECAD this is the most complex subsystem containing the most modules. It is responsible for all intelligence and power generation. This section breaks this subsystem into more discrete parts. It begins by separating the two computing modules into real time and non-real time components. The real time components include some of the on board sensors which are required for stable and fast control, which all feed into the flight controller. For the drone to successfully autonomously navigate an environment, it will need to better understand its surroundings. The flight controller will analyse this data and based on flight software will control the electronic speed controllers (ESCs). The on board computer (OBC) will process and handle all non flight critical sensor information and can be considered as the mission computer. The OBC must also have provision to connect to a sensor pack via a predetermined interface. Another useful and necessary function, will be the ability to illuminate a dark area.

The network design will also consist of 2 discrete wireless links. Each link is dedicated to one of the computing modules. The real time node will be interfaced through a low bandwidth, high range and high reliability interface. With the non-real time node linking to the high bandwidth interface which will have lower range and reliability requirements. Both of these interfaces will link to a ground station of some sort.

No system can operate without a sufficient power source. In the case of any robotic system it is important that every aspect of the design is optimised. The capacity of the power supply is limited by weight and should be optimised to the lifting capacity of the platform. Once the battery chemistry has been decided upon, work can be done to design a robust power management system. Limiting the leakage of every subsystem as well as choosing efficient modules are both part of an effective power management system. The analysis of the power system is important since it generally contributes a very high percentage to the overall mass of the vehicle.

3.2.2.1 Flight Controller

The flight controller is responsible for handling all the flight critical sensor data. Using a motor mixing algorithm the flight controller will then output signals to the electronic speed controllers. There are multiple options for flight controllers, all tailored to different applications, platforms and flying styles; each option has their own level of reconfigurability. For this application it is important that the chosen flight controller allows access to both the inner control loops as well as allows additional sensor data to be added into the architecture. The flight controller is also responsible for handling the pilot commands and waypoints sent via the radio link, as well as transmitting in flight diagnostic information back to the ground control station.

The flight controller's computing time capabilities have a direct influence on the system. This timing needs to be understood and introduces requirements on the controller designs. Any additional computing the vehicle needs should be handled by the OBC and not the flight controller. This will reduce the risk of unmodelled timing delays and unrealistic controller design goals.

3.2.2.2 Electronic Speed Controller

The above mentioned flight controller sends a motor control command to the electronic speed controller (ESC). The ESCs then in turn directly control the speed of each motor. Each motor should have a dedicated ESC. This part needs to be chosen based on the maximum amount of current it can handle. At 100 % throttle the current draw of the motor should not exceed 75 % of the ESC's limit. Another important consideration is the ESC's refresh rate and computing speed. The flight controller will be rapidly sending data to the ESC. The quicker the module can respond, the more robustly the platform will be controlled. Possible timing delays can be accounted for by ensuring the controllers have sufficient phase margins.

3.2.2.3 On Board Computer

Where the flight controller ensures the vehicle maintains steady flight, the On Board Computer (OBC) handles all higher level processing and flight strategy. This will include interfacing to the rest of the on board sensors, the sensor pack as well as handling the high bandwidth networking to the ground station. The OBC should not be required for flight purposes, and can thus be considered as a dedicated mission computer. The OBC should be as light weight and low power as possible while being sufficiently powerful to do real time analysis of camera data for future missions. The OBC must have multiple standard communication ports to easily interface to other sensors. There should also be a highly reliable, and fast interface between the OBC and the flight controller, thus allowing more complex flight strategies to be put in place, while not risking the robustness of the tracking control.

3.2.2.4 On Board Sensors

The on board sensors can be split up into two distinct categories. The first set of sensors are to enable stable flight of the vehicle and are used to close the control loops. Examples of such sensors would be inertial measurement units (IMU), global positioning units (GPS) and other relative velocity measuring sensors.

The second set of on board sensors are required for enabling more intelligent autonomous flying strategies. Examples of these sensors include proximity measurements to enable obstacle avoidance and localisation. Other types of sensors from this set could include condition monitoring of the vehicle's batteries.

3.2.2.5 Sensor Pack

The actual sensor pack in question will remain generic. So that the platform can cater for an array of sensing devices and applications. To ensure compatibility, the sensor pack needs to be powered and communicated to. It will have access to interface directly to the OBC. Typical examples of sensor pack application would be mapping equipment and stored video feed for post flight inspections.

3.3 Platform Construction

The traditional configurations of drones struggle to handle disturbances introduced by being in a confined space. Traditional configurations are also not optimised for fitting inside a long narrow space. For these reasons a few unique designs were considered and are discussed below. The comparison surrounds the hover point and it was assumed that thrust to RPM and RPM to current are linear relationships around this working point.

Two concepts were selected as final candidates. This next section walks through some of the important factors considered and ultimately, why certain decisions were made. The naming convention used is shown in Figure 3.2. On the left of Figure 3.2 represents the *"Unlike Size Quad"* and the right of Figure 3.2 is the *"Overlapping Quad"*.

Figure 3.2: Rendering of initial concept of the unlike rotor size quadcopter (Left). Malloy Aeronautics hoverbike concept (Right) (Picture taken [41]).

3.3.1 Concept 1 - The Overlapping Quad

The overlapping quad is a concept pursued by Malloy Aeronautics [41]. They used the design in an initial concept of their hover bike personnel carrier, which requires a large payload capacity. The design uses an H-formation for its rotors, except the rotors are brought in to limit the width of the vehicle to the point that they overlap, as shown in Figure 3.2. Each overlapping pair will have both spin directions, this feature is shown in Figure 3.3.

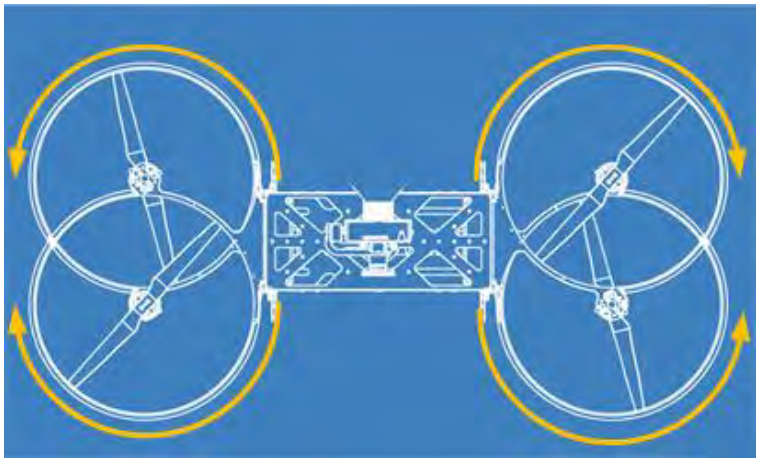


Figure 3.3: Overlapping concept, visual representation of rotor pairs. Image modified from [41]

Overlapping rotors introduce an inefficiency into the system. Figure 3.4 illustrates how the percentage overlap affects the total efficiency of the rotors. The point highlighted is the overlap seen in the Malloy Aeronautics implementation.

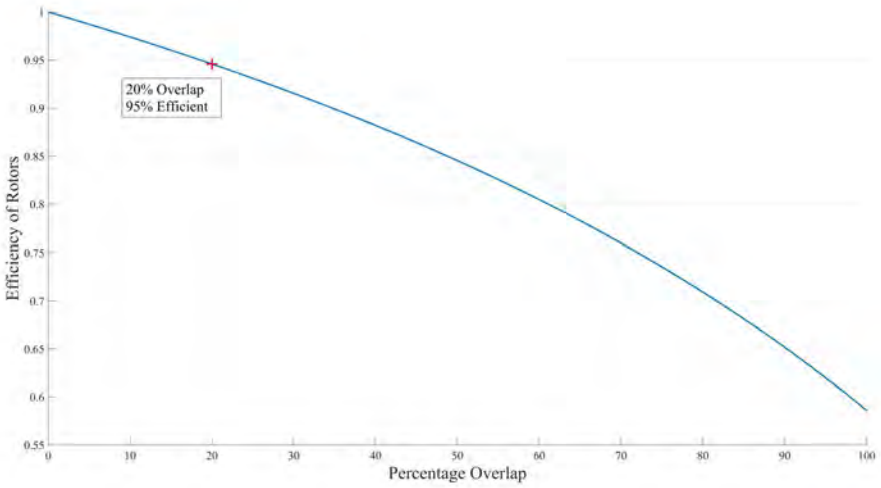


Figure 3.4: Graph representing the effects of overlapping rotors in a quadrotor

A large benefit to this design is the power to size ratio. This benefit can be utilised through handling larger payloads, or a larger battery pack increasing total flight time.

3.3.2 Concept 2 - The Unlike Size Quad

The unlike size quad is an original design that uses the standard cross formation, except it has two pairs of different size rotors. This means that the counter rotating pairs will be set up as shown in Figure 3.5, with each rotation direction having one big and one small rotor.

To maintain a common disk loading in the system, the thrust requirement will be lower on the smaller blades and larger on the bigger blade. The smaller the side rotors get, the higher the thrust requirements of the larger rotors become, this limits the rotor size ratio. Initial calculations, factoring in thrust overhead, overall size of the vehicle and minimum thrust allowed to rotors, set the ratio between $65\% - 80\%$. When approaching the lower bound, the thrust requirement for hover alone leaves very little room for manoeuvring or disturbance rejection. The upper bound reaches a point where the size difference is so negligible the design's narrow intent is lost.

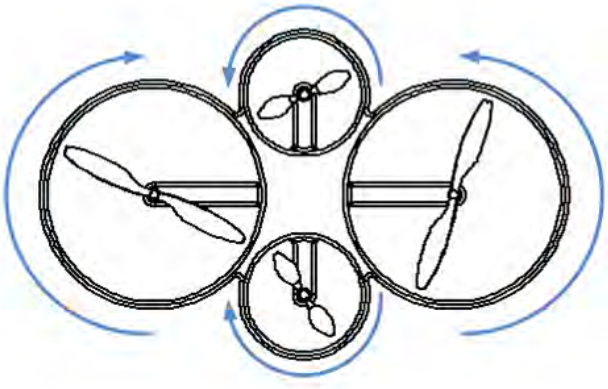


Figure 3.5: Unlike size quad visual representation of rotor pairs

The second important choice is how far to put the rotors away from the centre. As the vehicle gains translational speed, the air doesn't come in directly from the top any more as shown in Figure

2.1. Instead it now starts to come in at an angle. As the vehicle then manoeuvres and changes its orientation, the air starts coming in at more extreme angles. If the electronics housing has a lot of height it can interfere with the wake boundary and create an inefficiency. This inefficiency is also based on how far the rotors are from the housing. Therefore, before this decision can be made the limits of how small the centre electronics housing needs to be decided.

Through discussions and observations of current systems a minimum limit of 75mm $\times$ 75mm was set. To avoid overlap, the distance from the centre of the big rotor and the centre of the vehicle will be at the very least slightly greater than R (rotor radius). Including space for the housing increases this.

Since the vehicle needs to remain narrow, the side rotors can be brought in as well as shrunk. This would require pushing the two larger rotors slightly more out. Bringing the side rotors in closer to the middle housing will reduce efficiency, so a compromise must be chosen between width and efficiency. Since the side rotors don't contribute as much to the system as their bigger counterparts, they have the option of a lower separation distance. The lower separation distance can also be justified by the lower need and use of roll moments and side translations.

3.3.3 Concept Comparison

The comparison of the two concepts includes hover efficiency, thrust, electrical power requirements, size and manoeuvrability. The final decision will need to be made with certain assumptions in mind. These assumptions as well as the method of comparison are described below.

3.3.3.1 Assumptions and Method

Since both designs use 4 rotors they can be compared relative to a well known design, the standard quadrotor. For each configuration certain parameters need to be decided before a comparison can be done.

If hover is a thrust of 100%, the overhead was set at 50% of that, to a total of 150%. The mass in question includes provision for a sensor pack of undecided mass.

The unlike size quad had its small rotors set at 75% of the larger ones, this decision is explained further in the text. To give a quantifiable reference, R was set at 254 mm. With that assumption the unlike size quad moves its side rotors in to a distance of 300 mm and the larger rotors were moved to 508mm $= 2R$ away. The overlapping quad set a separation distance of 350 mm which created an overlap factor of 20%.

3.3.3.2 Rotor Area

If a rotor size of R is assumed for the rotors[1] then the total area for a standard quad will be $A_{std} = 4\pi R^2$. The reduction in radius of the two smaller rotors leads of the unlike size quad leads to a decrease in area of 78.13% of A_{std}.

3.3.3.3 Thrust and Power Considerations

The decision to set the smaller rotors to 75% was based on observation of thrust ratios between the rotors. The final value of thrust required from each rotor as a percentage is shown in Figure 3.6. The points marked are at minimum, hover and maximum. The total thrust available to the unlike size quad is $\approx$ 78% of the thrust available to the standard design. This reduced value also comes at a weight reduction. The overlapping quad has an equal total rotor area but an inefficiency is introduced by the overlap as according to (2.2.1). Therefore the overlapping quad has 95% of the total thrust capacity, without the weight reduction.

[1] The big rotors in the case of the unlike size quad

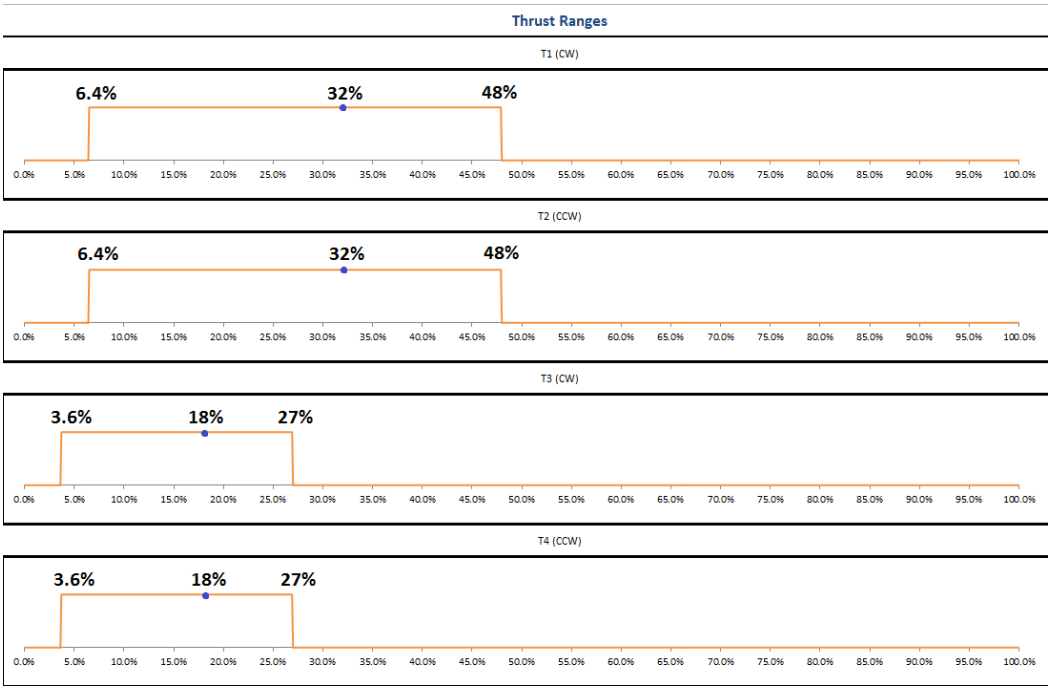


Figure 3.6: Graphical representations of the thrust ratios for the unlike size quad

The values for electrical power were calculated according to how much energy would be needed to obtain the same thrust as the standard design. The inefficiency introduced by the overlap relates to a reduction in thrust of $\Delta T_{overlap} = 5.36\%$, therefore $\Delta P_{overlap} = 14.21\%$ is needed to overcome this loss, based on (2.1.8).

For the unlike size quad, the reduction in rotor size leads to a substantial loss in aerodynamic power, even with a small reduction in inertia. To regain that power, the rotors need to be pushed harder, this leads to an increase in electrical power. A value of $\Delta P_{unlikesize} = 18.5\%$ is calculated.

3.3.3.4 Size and Manoeuvrability

The size was calculated as though the drone made a rectangular box and with the rough values above, Table 3.1 puts those values in a tabular format.

Concept	Length (mm)	Width (mm)
Unlike Size	1524	981
Overlapping	**1308**	**858**
Standard	1524	1524

Table 3.1: Table representing the size comparison of the two concepts

As shown both crafts are similar in size, with the overlapping design being slightly shorter as well as more narrow.

3.3.3.5 Discussion

The quantifiable values are culminated below in Table 3.2. The winner of each parameter is written in bold.

Concept	Disk Loading	Total Thrust	Electrical Power	Length (mm)	Width (mm)
Unlike Size	**78.13%**	78.13%	118.57%	1524	981
Overlapping	80.18%	**94.64%**	**114.21%**	**1308**	**858**
Standard	100%	100%	100%	1524	1524

Table 3.2: Table representing the end comparison of the two concepts

The overlapping quadrotor introduces a significant size reduction to the standard quadrotor while losing minimal thrust to inefficiencies. This allows for a larger more powerful drone to be deployed in tight spaces. The extra required electrical power can be nullified by the ability to carry a larger power source thus increasing flight time.

Malloy Aeronautics sell what is called a *Drone-3 Kit*, it includes various accessories to help developers use the platform. Based on the above analysis and conclusions the overlapping quadrotor design will be used as the design going forward.

3.4 Electronic Design

At the core of any reliable system is a thoroughly planned electronics system. There are multiple aspects this subsystem must handle. Due to the rapidly increasing market and interest in multirotors there has been a wave of developers and designers creating devices for these specific purposes. The first and foremost is monitoring and controlling flight dynamics.

The on board computer will handle the data streams and control the non-real time peripherals. For the drone to successfully navigate an environment it will need to better understand its surroundings.

3.4.1 Flight Controller

There are multiple options for flight controllers, all tailored to different applications and flying styles. This section identifies some possible options

3.4.1.1 Custom Design

Given the time and resources most final products will look at custom designing some hardware and electronics. The benefits of a custom design include complete control over the operation and functionality of the design as well as reduced cost when scaling up. The development of a custom design however can be very taxing and costly.

It is also common for a research lab to dedicate time into developing such modules. By using a custom board designed by the Massachusetts Institute of Technology, Cutler in his masters dissertation demonstrates the effectiveness of custom electronics [42]. *The University of Stellenbosch's Electronics System Lab* produced a custom design of their own which is currently under redesign and unfortunately cannot be used for this project.

3.4.1.2 Pixhawk

Considering time and budget of the project, the need for a commercial flight controller becomes evident. Due to the growing hobbyist community, some flight controllers are difficult to modify and are designed for use as a *"plug and play"* module. Fortunately there is also a large designer community which has created the need for more configurable modules.

Figure 3.7 shows the *Pixhawk*, which is marketed as an autopilot module for fixed wing and rotor wing aircraft. It is specifically tailored for research and is listed as open-hardware[2]. Due to the open platform it has created an experienced community with good documentation and other forms of assistance.

[2]https://store.3dr.com/products/3dr-pixhawk

Figure 3.7: Pixhawk flight controller

It features a 32-bit STM32F427 processor, running at 168MHz with 256Kb of RAM and 2Mb of Flash memory. It comes equipped with a full inertial measurement unit (IMU), consisting of a gyroscope, accelerometer and magnetometer. The Pixhawk also includes an integrated barometer and has an additional 32-bit co-processor that acts as a failsafe. There are multiple interfacing capabilities as well as a built in power protection unit[3].

The Pixhawk has been designed for robotic applications thus is light weight and power efficient. It operates on a real time operating system called *NuttX* which has Unix characteristics but is much lighter than a different operating system such as Linux. This provides a lot of reconfigurability which is needed in this project.

3.4.2 On Board Computer

The On Board Computer (OBC) is required to handle all higher level processing. In an aerial application weight and power consumption are both important considerations. Based on current available commercial products a few were selected. The advantages and disadvantages of each are described below.

3.4.2.1 Raspberry Pi 3 Model B

The Raspberry Pi has become a well known and respected piece of hardware. It has generated a large community and thus resources are readily available and the device can be bought locally. The new version of the device runs a 1.2GHz 64-bit quad-core ARMv8 CPU and includes built in Bluetooth and WiFi modules, an image from the Raspberyy Pi website is shown in Figure 3.8[4].

[3] https://pixhawk.org/modules/pixhawk
[4] https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

Figure 3.8: Raspberry Pi 3 Model B

It has a large 40 pin GPIO connector, 4 USB connections and an Ethernet port. The Pi's main advantage would be the access to the online community constantly updating Raspberry Pi resources and forums. With the community also comes example projects and large variety of compatible hardware and open source software. With a maximum current of 2.5A at 5V, the 12.5W computer is relatively low powered which suits this application. However there are more powerful machines that can run more intense algorithms at the cost of more power.

3.4.2.2 Odroid XU4

Hardkernel has designed a compact high processing power unit called the Odroid XU4. It has gained respect in some developer communities due to its incredible processing power. It can run both Android, Ubuntu and other similar Linux based operating systems. Hardkernel has generated a large amount of documentation and wiki pages, all available on their website[5]. They also have a team of developers creating new devices to interface with the computers.

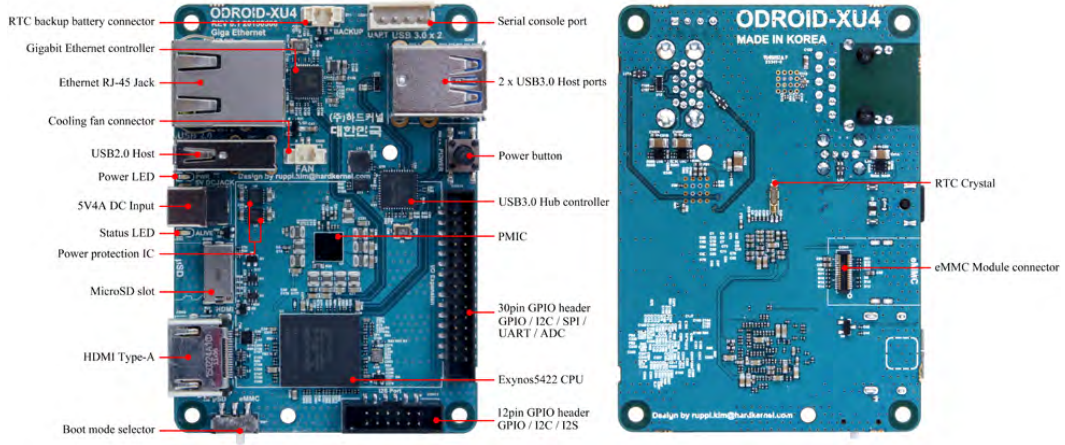


Figure 3.9: The Odroid XU4

The XU4 hosts a Samsung Exynos5422 Cortex-A15 processor running at a 2GHz clock speed and an additional Cortex-A7 Octa core processing unit. This allows for incredible processing capabilities and speed, an image from Hardkernel's Website is shown in Figure 3.9. The rated power supply

[5] http://www.hardkernel.com/

for the unit is a 4A at 5V module. Including a 20W module on board will add a significant drain on the power source. It comes with 2 stable USB 3.0 ports and an additional USB 2.0 connection.

The Odroid is also frequently used at the CSIR, opening up experience and knowledge on the embedded computer.

3.4.3 Location

An important part of any robotic application is localisation. As stated earlier, an underground environment limits the use of traditional GPS. Stellenbosch University as well as the CSIR are both funding research into localisation in a GPS constrained environment. Until such a time that these technologies are readily available, this project will work with traditional GPS devices.

As discussed previously, this project will not be designing for a GPS constrained environment. Fortunately there are many commercial GPS modules available for purchase. Each with their own advantages and disadvantages. For this project the module needs to be lightweight and low power. The GPS needs to create speed and translation data, this information is crucial when generating maps of unknown areas. The PixHawk website recommends the 3DR uBlox GPS kit[6]. Although many alternatives exist, the open source hardware status creates ease of integration and support. With these considerations, this module has been chosen.

It weighs a total of 16.8g and as has a low 8.5mm profile. This module will perform adequately and fulfil the design needs of the project.

3.4.4 Object Avoidance

Due to the nature of the project, object avoidance is an important attribute to include in the design. Obstacle avoidance requires that the drone has some level of an understanding of the environment. This information can then be used in higher processing nodes to actively avoid objects and plan routes. A few different sensor options are observed below, each with a set of advantages and disadvantages.

3.4.4.1 Laser Range Finders

Laser range finders can be found in a multitude of high end robotic platforms. They exhibit high accuracy and can be bought as full 360° modules. Laser range finders are expensive and heavy making them less suitable for this application. Single point laser range finders are also readily available, but have a very narrow beam, reducing their coverage.

3.4.4.2 Ultrasonic Proximity Sensors

Ultrasonic sensors are already used in aircraft, generally as downward facing to accurately predict altitude. Relatively speaking ultrasonic sensors are cheap and light weight and consume little power. They are however prone to measurement errors when used against non-uniform surfaces and require additional processing and time to ensure that multiple sensors do not influence each other. These sensors produce a measurement cone, rather than a directed pinpoint measurement providing a large coverage area per sensor. For this work ultrasonic proximity measurement will be used.

[6] https://store.3dr.com/products/3dr-gps-ublox-with-compass

Chapter 4

Mathematical Modelling and System Identification

This chapter follows the process of mathematically modelling the quadrotor system. The nomenclature for the kinetics and kinematics has been discussed in Section 2.3. The chapter begins with the dynamic flight model of the vehicle, focusing on all forces and moments generated by the rotors. The chapter then continues to describe the process of system identification followed to generate the required constants and system parameters for an accurate model. The external disturbances are discussed next, specifically the wind models used in this work. The chapter concludes with a brief overview of the non-linear simulation generated from the discussion.

4.1 Dynamic Flight Model

The dynamic flight model of the vehicle must cater for all six of the degrees of freedom the vehicle desires. The dynamics of this system is modelled as three rotational and three translational degrees of freedom [19]. To continue deriving the equations of motion, the following assumptions are made: the aircraft is a rigid body, the aircraft has constant mass and I_{xy}, I_{xz}, I_{yz} are all negligibly small.

The Newton-Euler method of defining the accelerations uses the inertial frame to define the linear accelerations, and the body frame to define the rotational accelerations. Using Newton's first law and the rotation matrix described in (2.3.1), the expression for the linear acceleration can be developed and is shown in (4.1.1). $\mathbf{R}$ is the rotation matrix.

$$\begin{bmatrix} \ddot{N} \\ \ddot{E} \\ \ddot{D} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \mathbf{R} \begin{bmatrix} 0 \\ 0 \\ \frac{\mathbf{T}}{m} \end{bmatrix} \tag{4.1.1}$$

The rotational accelerations of the vehicle can be similarly described using the moments and the simplified inertia tensor. These rotational rates are described in (4.1.2)

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} I_{xx}N \\ I_{yy}M \\ I_{zz}L \end{bmatrix} \tag{4.1.2}$$

4.2 System Identification

In order to correctly model the system, a thorough system identification needs to be completed. This entails real world measurements of the chosen platform or substantiated evidence from literature for aircraft of similar size and characteristics. The methods and results from these experiments are covered in this section.

4.2.1 Mass and Inertia

Using a calibrated scale the mass of the rotorcraft measured at 3.352Kg.

To calculate moments of inertia, the Bifilar Pendulum method was used. The method is thoroughly described in literature and involves tying the drone from the ceiling allowing it rotate around one axis. Since it is desired to measure the inertias along three axes, three separate test set-ups were required. Images of the test set up for a single axis is shown in Figure 4.1.

Figure 4.1: Bifilar pendulum for inertia measurement

Each axis was measured 10 times and the values were averaged out to obtain the final values represented in Table 4.1. To give a representation of measurement variance, the standard deviation is provided along side.

Parameter	Averaged Measured Value	Standard Deviation
I_{xx}	0.025027578	0.001063842
I_{yy}	0.169260024	0.000142928
I_{zz}	0.170196714	0.000527406

Table 4.1: Measured moments of inertia

4.2.2 Thrust and Moment Profiles

In order to correctly validate the thrust characteristics of the drone, each motor rotor pair needed to be evaluated. Each pair was marked and coupled to a load cell. The ESCs were configured to send varying PWMs to the motors. The commands sent to the ESCs and the measured thrust values are plotted together in Figure 4.2. Table 4.2 gives the exact maximum and minimum values of each rotor motor pair.

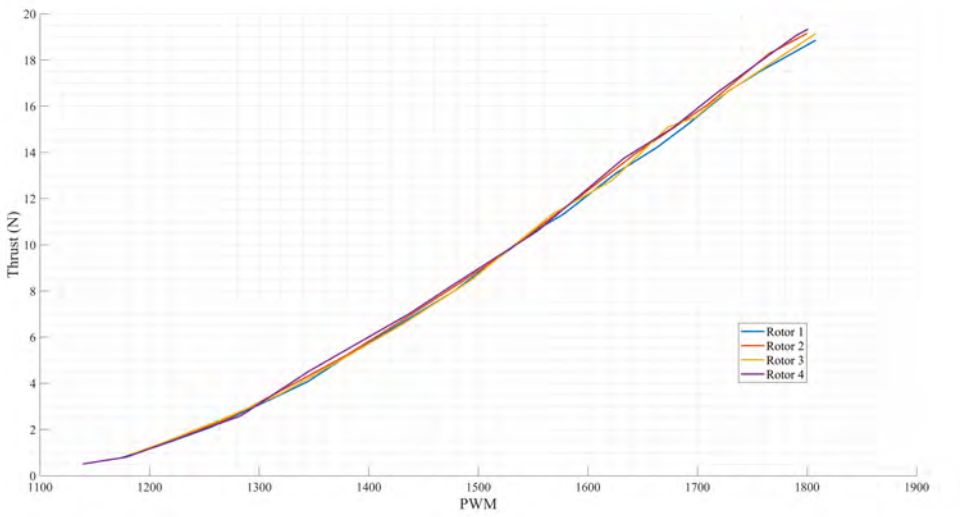


Figure 4.2: Thrust ranges for motor rotor pairs

Pair	Max Thrust	Min Thrust
1	18.8558 N	0.7852 N
2	19.1489 N	1.1889 N
3	19.1434 N	0.9511 N
4	19.3369 N	0.5087 N

Table 4.2: Measured rotor thrusts

The motor lag constant for a similar sized vehicles and rotors was investigated in [19] and found to be 0.125s. Each rotor is also subject to high frequency noise caused by undesired vibrations. The high frequency noise is modelled as a bandwidth limited white noise applied to each motor separately.

4.2.3 Drag Coefficients

The drag coefficients chosen are that typically expected for a flat plate with an effective frontal area, as that seen in the fuselage of the vehicle. A C_D of 1 is chosen.

The surface area of the drone was calculated by taking physical measurements of the drone and calculating the surface area from the measurements. Table 4.3 records the calculated surface areas to be used in the calculation of each drag component. The last value in the table is the centre of gravity (COG) which was calculated to be located at the centre of the X and Y axis indicating symmetry along those axes, with a 3.5 cm offset in the Z-Axis. The centre of pressure (COP) is assumed as the centre of the vehicle, creating the required offset for the drag moments described in 2.3.5.1.

Parameter	Value
A_x	$0.0354\,\mathrm{m}^2$
A_y	$0.0693\,\mathrm{m}^2$
A_z	$0.4\,\mathrm{m}^2$
C_D	1
COG	$[0\ 0\ 0.035]\,\mathrm{m}$
COP	$[0\ 0\ 0]\,\mathrm{m}$

Table 4.3: Drag coefficients

4.2.4 Sensor Constants

Investigation into the sensors used by the PixHawk, the proposed flight controller module, gives an indication of the sensor accuracy, resolution, noise and speed the drone measurements will be subject to.

The PixHawk uses a combination of gyroscopes, accelerometers and magnetometers. The sensors used for the characterisation are the 6 DOF BMI055 which contains a 3-Axis accelerometer and a gyroscope along with the IST8310, 3-Axis magnetometer. Table 4.4 lists the extracted data.

Sensor	Offset	Resolution	Noise	Maximum Bandwidth
Accelerometer	$\pm 70\,\mathrm{mg}$	$0.98\,\mathrm{mg}$	$150\,\mu\mathrm{g}/\sqrt{Hz}$	1000 HZ
Gyroscope	$\pm 1\,^\circ/\mathrm{s}$	$0.1\,^\circ/\mathrm{s}$	$0.014\,^\circ/\mathrm{s}/\sqrt{Hz}$	1000 HZ
Magnetometer	$\pm 0.3\,\mu\mathrm{T}$	$0.3\,\mu\mathrm{T}/\mathrm{LSB}$	NA	200 Hz

Table 4.4: IMU sensor coefficients

For location and velocity measurements the GPS sensor used is the Neo-M8N developed by UBlox. Table 4.5 lists the relevant information obtained from the sensor datasheet.

Sensor	Accuracy	Maximum Bandwidth
Position	$2.5\,\mathrm{m}$	10 Hz
Velocity	$0.05\,\mathrm{m/s}$	10 Hz

Table 4.5: GPS coefficients

From the information gathered the sensors can now be modelled accordingly in Simulink. The IMU is modelled using the 3-Axis accelerometer block and the 3-axis gyroscope block in Simulink. The noise is set by using the above documented noise ratios. The GPS model utilises the Band-Limited White Noise Block in Simulink to create appropriate sensor noise.

4.2.5 Wind Model

The wind model is broken into two portions, a constant wind and a more erratic, unpredictable wind both of which flow in the NED frame. The constant wind is defined as a configurable constant in the North East Down frame. The gust component is modelled as shown in Figure 4.3. The band limited white noise block is passed through a first order low pass filter to create a more realistic dynamic wind. The gain of the filter can be adjusted to observe effects of larger wind gusts on the vehicle.

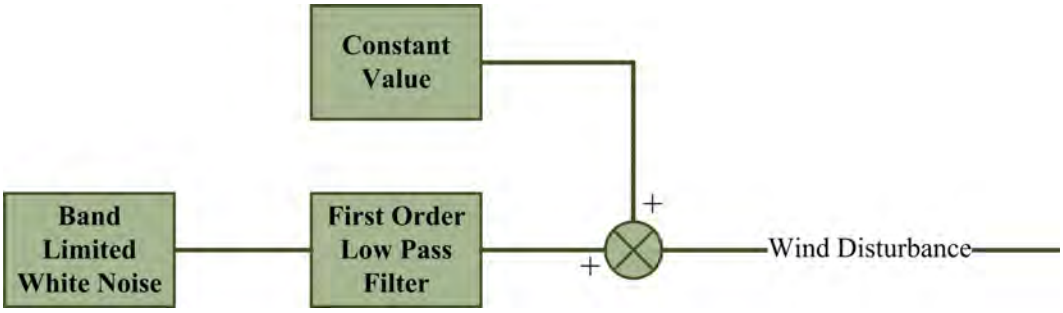


Figure 4.3: Wind model

The direction is commanded in the NED frame and rotated to apply the applicable forces to the body frame. The direction is modelled with a constant offset and varied by adding a small random number, smoothed by a low pass filter to create a more realistic wind direction variance.

4.3 Simulation Configuration

This section describes the process of creating the non-linear simulation of the aircraft. The drone was modelled in Simulink using a combination of blocks included in the Aerospace and Control Toolboxes, as well as a set of custom blocks. The model is required to successfully predict the position of the vehicle after being subjected to forces and moments. Using the dynamic flight model created above the bodies linear and rotational accelerations can be calculated by inputting forces and moments into the system. These accelerations can be subsequently integrated to achieve rates and positions. No model is completely accurate, however the information recorded in this chapter, when included, should create a good estimation of the real world vehicle dynamics.

4.3.1 Motor Mixer

The motor mixer is an important part of the control software running on the onboard computer and thus should be accurately modelled. This portion of code commands the ESCs of the drone to create the desired moments and forces. Section 2.3.4 explains how the rotors have control of four virtual actuators, these are δ_Z, δ_θ, δ_ψ and δ_ϕ. The motor mixer is responsible for converting these desired moments and forces into actual values of thrust for each rotor.

As Figure 4.4 describes, each motor thrust value is made up of a summation of four thrust values, T_Z, T_θ, T_ψ and T_ϕ. Each value describes the rotor's contribution to the generation of that specific virtual actuator.

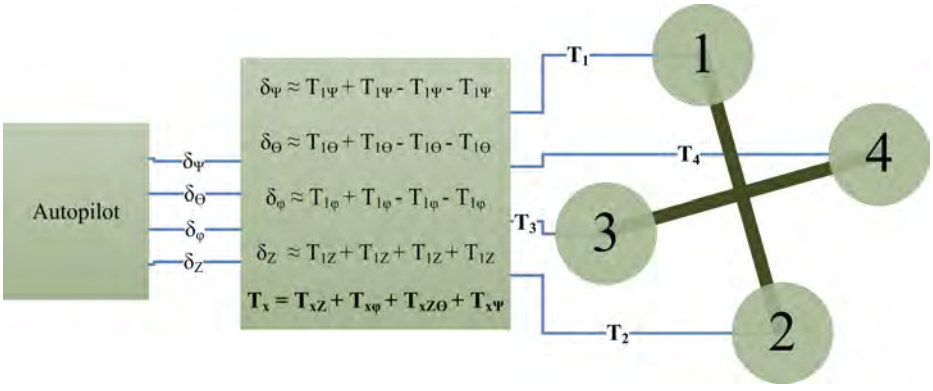


Figure 4.4: Motor mixer

The motor mixer mathematics is based on the contribution provided by each rotor's thrust to the relevant virtual actuator. Assuming that each rotor only contributes a thrust aligned with the

Z-Axis, it can be said that each rotor's thrust contributes 100% of its power to the virtual actuator δ_Z. The roll and pitch virtual actuators are moment contributions and are calculated by evaluating the distance off the centre of gravity. Using the rotor arm length (l) and the angle (α) a simple trigonometric relationship dictates the roll and pitch contributions. Finally the yaw contribution is calculated by taking into consideration the drag created by each rotor as it generates thrust. This relationship includes the lift to drag ratio (R_{LD}) of each rotor and the effective chord length (r_D) where the drag force is applied. Using all of these relationships the motor mixer's contribution matrix can be created and is shown in (4.3.1). The signs in the matrix are dictated by the body frame notation of positive directions and the placement of the rotors. The yaw contributions are dependant on the rotation direction of the rotors.

$$\text{Contribution Matrix} = \begin{bmatrix} -1 & -1 & -1 & -1 \\ l\sin\alpha & -l\sin\alpha & l\sin\alpha & -l\sin\alpha \\ -l\cos\alpha & l\cos\alpha & l\cos\alpha & -l\cos\alpha \\ -\dfrac{r_D}{R_{LD}} & -\dfrac{r_D}{R_{LD}} & \dfrac{r_D}{R_{LD}} & \dfrac{r_D}{R_{LD}} \end{bmatrix} \tag{4.3.1}$$

The values for thrust can subsequently be calculated by multiplying the desired forces and moments by the inverse of the contribution matrix.

Chapter 5

Controller Design

This chapter follows the stages of designing a controller for a quadrotor and begins by discussing the overall strategy. An analysis of existing rotorcraft control systems has been done in Section 2.4. The system must be able to follow waypoint commands, containing a position reference in the North, East and Down frame. As well as align with a desired heading, this reference will be in the form of a Euler angle. The detailed design goals are discussed before the controller design is begun.

5.1 Design Goals

The design goals for the control system are presented here. The controllers must ensure that the system is capable of meeting all of the prescribed requirements.

1. The controller must be able to track a position reference in the North, East and Down frame. The limited space in which the vehicle must fly requires a final system with sufficient damping and minimal overshoot.
2. The transient response of the velocity control system must be rapid to ensure fast responses to changing setpoints caused by unexpected obstacles.
3. The controllers must be robust to unmodelled system errors. To accomplish this sufficient phase margin is required to handle unmodelled timing delays and sufficient gain margin is required to handle unmodelled errors in actuation and plant gains.
4. The output of the controllers must never exceed the thrust capabilities of the system.
5. The controllers should be designed as to not demand large angles for the vehicle.
6. The velocity of the vehicle must be limited for flight inside a confined environment.
7. The controllers must be capable of rapidly rejecting disturbances.
8. The designed controller bandwidth must not exceed the bandwidth capable of the rotor motor system.

The stipulated design goals are appropriate for a vehicle expected to fly in a confined space and differ from traditional flight which requires high speed and fast position tracking.

5.2 Flight Control Strategy

Figure 5.1 represents the high level control strategy for this project. This discussion will break the system into three main components: an altitude controller, a horizontal flight controller and a heading controller.

The altitude controller starts by getting a position reference in the earth frame. This reference is fed through a Proportional Integral (PI) controller to create a desired climb rate. The climb

rate reference is converted into the body frame and is then controlled using a Proportional (P) controller to produce a body acceleration reference. The heave controller can only produce a force perpendicular to the rotor, thus the Z-axis acceleration component is taken and fed through an additional PI heave controller. The output of this inner heave controller would be the virtual actuator δ_Z.

The horizontal controller gets both a North and an East position reference and uses PI controllers to create velocity setpoints in the earth frame. These setpoints are converted to the body frame where the linear velocity P controller works and outputs acceleration references in the body frame. Using the body acceleration references, roll and pitch rate setpoints can be created using a tilt angle controller. These rate set points are fed into the inner rate loop lead-lag controllers which output the virtual actuators δ_ϕ and δ_θ.

Lastly the heading controller is discussed. Correct design of the tilt angle controller will decouple the vehicle's horizontal controller from a dependency on the heading of the vehicle. This allows seamless implementation of a heading controller. The heading controller is broken down into two parts: a yaw angle and a yaw rate controller. The angle loop uses a PI control architecture while rate loop utilises a simple P controller. The output of the yaw rate controller is the virtual actuator δ_ψ.

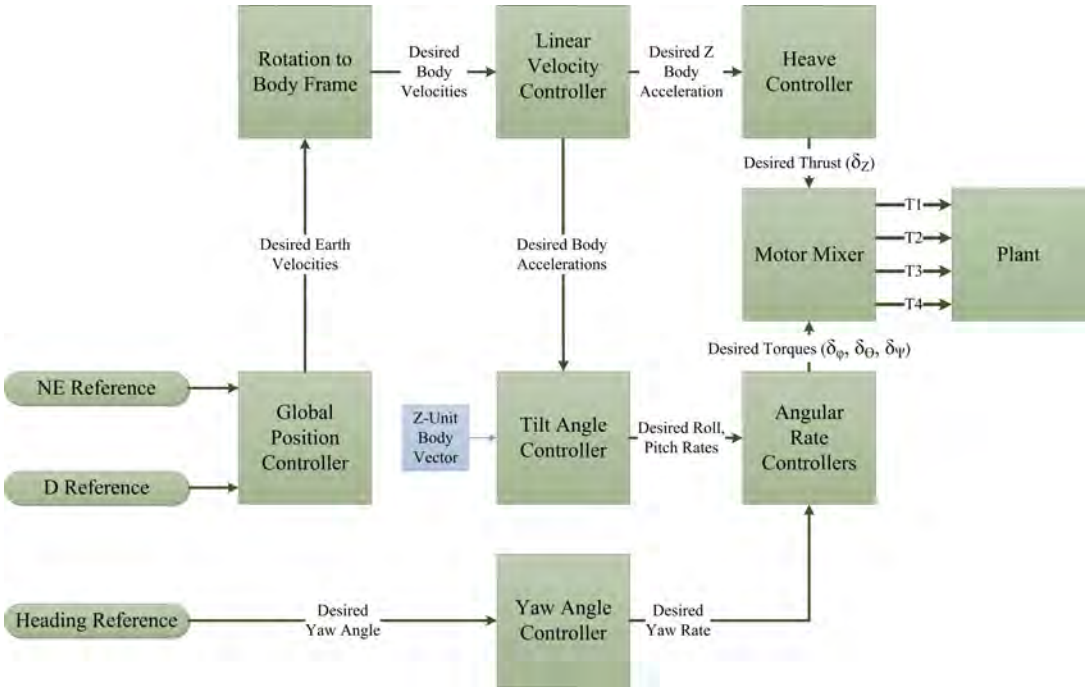


Figure 5.1: High level control strategy

Each controller utilises the same rotor system to produce their outputs. This dependency on a single generation system can create interference between the controllers. To ensure that no loop can saturate another, each system is given a percentage of headroom in which it can work, as shown in Table 5.1. The controller design must ensure that the thrust commanded during a step response is within those limits.

Controller	Percentage	Allowed Thrust Per Rotor
Altitude	64%	12.16N
North	8%	1.52N
East	12%	2.28N
Heading	6%	1.14N
Safety Factor	10%	1.9N

Table 5.1: Thrust headroom controller percentages

5.3 Altitude Controller

This section discusses the design and implementation of the altitude controller which is responsible for controlling the desired height of the vehicle. The vehicle is required to fly in confined spaces and must be able to track a setpoint with zero steady state error and negligible overshoot. In order for the altitude system to handle disturbances, an integrator term will be required. The system must be able to respond quickly to commands, however it must not exceed its thrust utilisation percentage. To do this the system will require an upper limit which should not be reached. A lower limit is then introduced so the vehicle does not descend too quickly.

The overall altitude controller is structured as a set of cascaded control loops, with the most inner loop controlling the aircraft's acceleration and the most outer loop controlling the desired altitude in the earth frame. The system must be able to respond to disturbances quickly. Therefore, the inner heave loop utilises a PI controller to follow a desired vertical acceleration reference. The climb rate P controller is responsible for generating these acceleration references and is fed a desired linear vertical velocity reference by the altitude hold P controller. In order to reject measurement errors in the inner loops, the altitude hold controller makes use of a limited integrator that does effect the bandwidth of the system. Before the controllers can be designed, an analysis of the system's heave dynamics must first be performed.

5.3.1 Heave Dynamics

Using Newton mechanics at near hover conditions for the aircraft, the heave dynamics can be derived and are shown in (5.3.1). Where $\dot{W}$ is the current acceleration of the vehicle in the Z-axis and m is the vehicle's mass. Z is defined as the current instantaneous force being produced by the rotors.

$$\dot{W} = \frac{Z}{m} \tag{5.3.1}$$

The state variable of the system is chosen as Z with the output of this plant being $\dot{W}$. Using the transfer function for motor-rotor lag dynamics seen in (2.3.11) and the dynamics seen in (5.3.1), the state space equation for the system can be derived and is shown in (5.3.2) and (5.3.3).

$$[\dot{Z}] = -[\frac{1}{\tau}]\,[Z] + [\frac{1}{\tau}][\delta_Z] \tag{5.3.2}$$

$$[\dot{W}] = -[\frac{1}{m}]\,[Z] \tag{5.3.3}$$

Subsequently the transfer function can be calculated and the result is shown in (5.3.4). The negative gain of the transfer function must be noted and is caused by the direction of the defined axes, with the rotors producing a negative Z-Axis force.

$$G(s)_{heave} = \frac{\frac{-1}{m\times\tau}}{s + \frac{1}{\tau}} \tag{5.3.4}$$

The rotor motor lag of 0.125 s stipulated in the system identification chapter produces the pole at $\frac{-1}{\tau} = -8$ and indicates the maximum response capabilities and timing constant of the rotor system.

5.3.2 Heave Controller

The heave controller is responsible for commanding the δ_Z virtual actuator to achieve a desired Z-axis acceleration in the body frame. The heave controller is the fastest controller in the altitude system and should utilise as much bandwidth as the rotor-motor system allows. The altitude controller wishes to reject disturbances quickly and thus a PI architecture was initially chosen as shown in Figure 5.2. The system should be stable and exhibit a reasonable phase margin and therefore be able to compensate for margin loss when the outer loops are closed. The phase margin in the open loop system should be at least 70°.

The integrator is used to reject disturbances, while the most left limiter shown in Figure 5.2 was added to stop integrator wind up and is not considered during the linear controller design. The proportional gain is used to move the closed loop poles and achieve the desired bandwidth. The dynamic response of the system can be investigated using the root locus and bode plots shown in Figure 5.3 and 5.4.

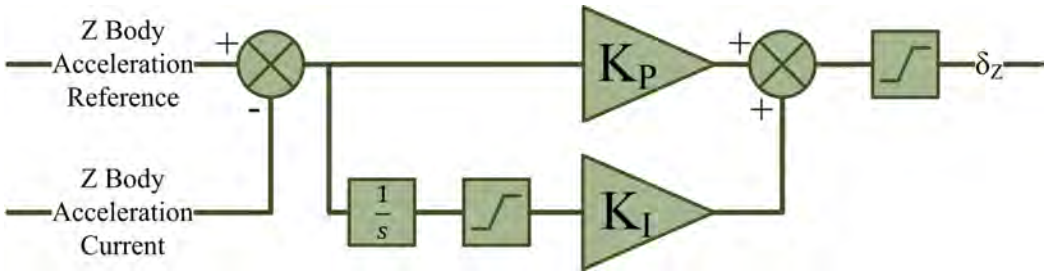


Figure 5.2: Heave controller - Control diagram

Figure 5.3 shows the root locus of the system with the PI controller included. The controller introduces a new open loop pole at the origin. To maintain a first order response, the zero is placed close to the plant pole. This placement will attenuate the open loop, plant pole's response. Finally the gain is varied until the closed loop responses are closely aligned with the naturally occurring open loop pole. The final closed loop poles are a set of complex poles and are located at $-7.55 \pm 3.64i$. The frequency response can be evaluated using the bode plot in Figure 5.4.

The final cross over frequency is shown on the bode plot for the heave controller in Figure 5.4. The gain plot shows the controller adjust the crossover frequency to 7.99 rad/s which is close to the limit of the system. The controller also increases the phase of the system and has a final phase margin of 84°.

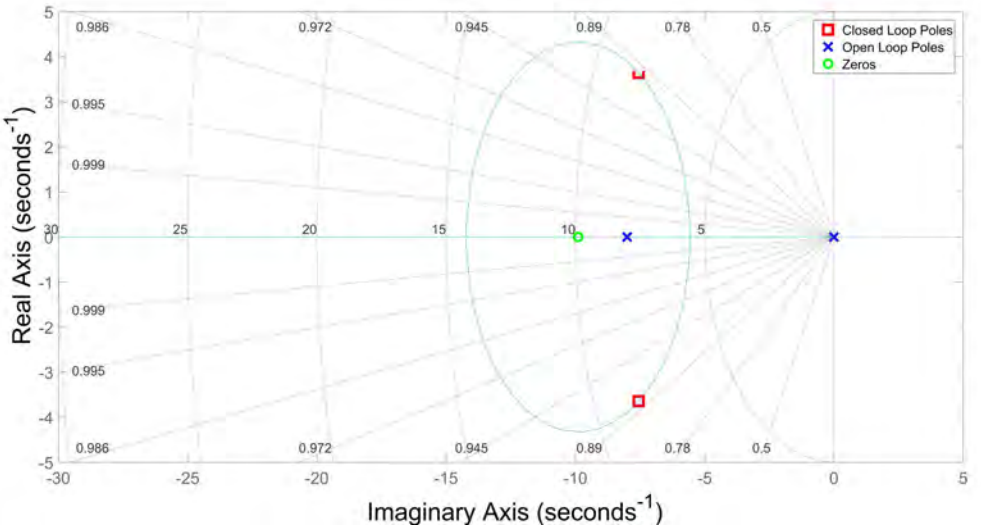


Figure 5.3: Heave controller - Root locus

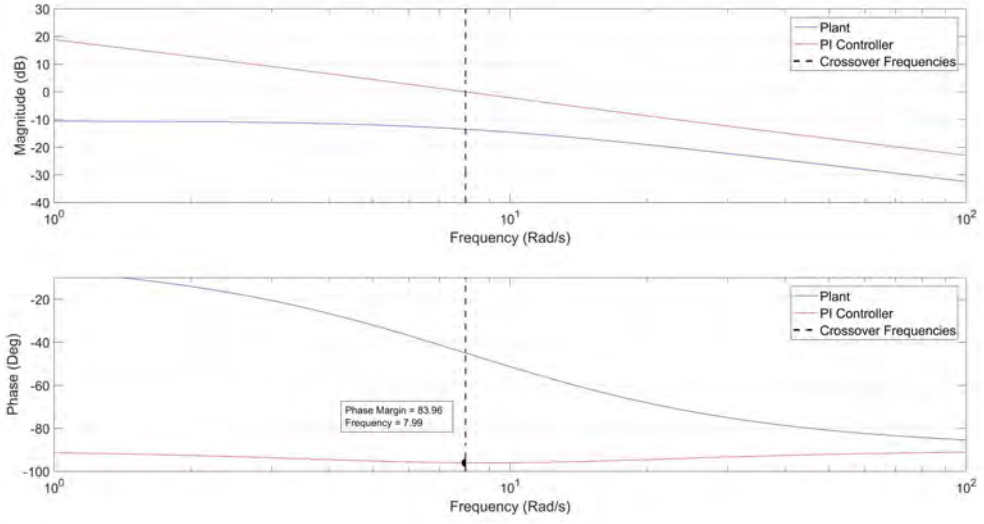


Figure 5.4: Heave controller - Bode plots

An additional non-linear element in the form of a limiter is brought into the system to limit the maximum and minimum thrust commands. The maximum limit is used to ensure the heave controller does not saturate the motors, thereby creating headroom for the angular rate controllers. The lower limit is used to ensure the vehicle always descends at a steady pace. The maximum thrust allowances are displayed in Table 5.1, the final limits chosen are shown in Table 5.2.

Limit Name	Min	Max
Integrator Wind Up	-1.5	1.5
Thrust Command	10	48.64

Table 5.2: Heave controller limits

5.3.2.1 Heave Controller Discussion

Now that the system presents stable dynamic results in the frequency and Laplace domains, using the non-linear simulation, the time domain responses can be discussed in brief. The resultant step response, including the PI controller, is shown in Figure 5.5. To demonstrate the disturbance rejection capabilities of the design, a force of $10N$ is applied to the drone at $2.5s$.

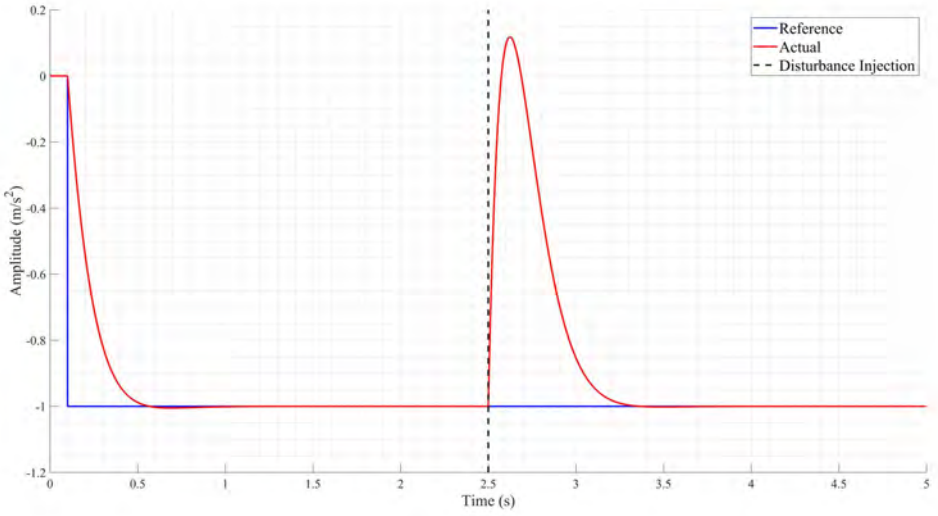


Figure 5.5: Heave controller - Step response

The heave controller, as the most inner loop, limits the response for the rest of the altitude control system. The proposed design brings the heave loop response close to the limits of the plant, thus producing a similar (but slower) timing constant to that of the motor-rotor system. The system reaches and settles within 5% of the reference by $0.31s$, is critically damped and presents negligible overshoot. The system also shows to be capable of tracking an acceleration setpoint with zero steady state error. As shown in Figure 5.5 the system can also respond quickly to a large, sudden and constant disturbance. The maximum rotor thrust commanded during this run is 3.35 N. Gravity will add on offset of mg to the acceleration setpoint, this needs to be rotated into the body frame as this controller provides force in the Z-body axis.

5.3.3 Climb Rate Controller

The climb rate controller is responsible for controlling the vertical velocity of the aircraft, in the earth frame. This introduces the need for rotating either the reference or the command into the body frame. The decision can be made by considering the frame in which the sensing information is provided. Most aircraft make use of some form of global positioning system and using differentiation can calculate speed. At near hover conditions the plant can be linearised and the rotation can be excluded. The architecture for the climb rate controller can be is outlined in Figure 5.6.

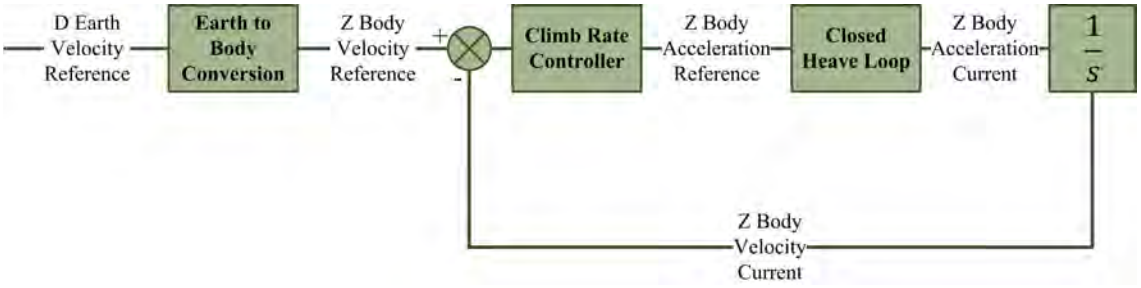


Figure 5.6: Climb rate controller closed loop

Figure 5.7 shows the simplified climb rate controller architecture. The computational time required for rotating the references can be considered by ensuring the controller design provides a reasonable phase margin. The speed of the climb rate controller is limited by the inner heave control loop. The controller must react quickly but there must still be a sufficient bandwidth ratio between the inner and outer loop. The controller must be able to track a setpoint with zero steady state error, but is not required to reject disturbances. The vehicle is required to produce a steady approach to position targets, the climb rate controller should then exhibit a damped first order response.

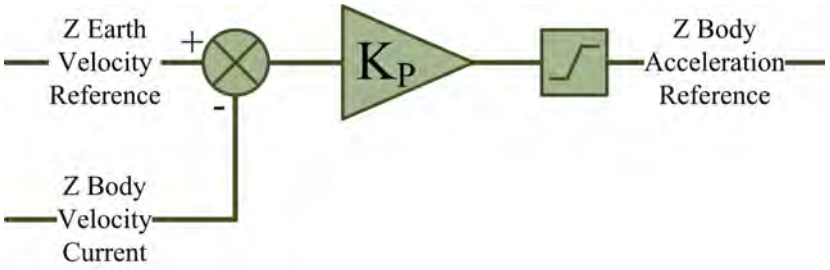


Figure 5.7: Climb rate controller

The open loop poles of the climb rate system are located at the closed loop pole positions of the inner heave system, while the mathematical relationship between acceleration and velocity yields an additional open loop pole at the origin. The free integrator in the plant ensures the system will track a step response with zero steady state error while the proportional gain is used to speed up the system and achieve the desired bandwidth. The dynamic response of the system is evaluated using the root locus and bode plots shown in Figure 5.8 and 5.9.

Figure 5.8 shows the location of the three final closed loop poles. There is a non dominant complex pair which is placed at $-5.11 \pm 3.76i$. The dominant pole is critically damped and located on the imaginary axis at -4.89.

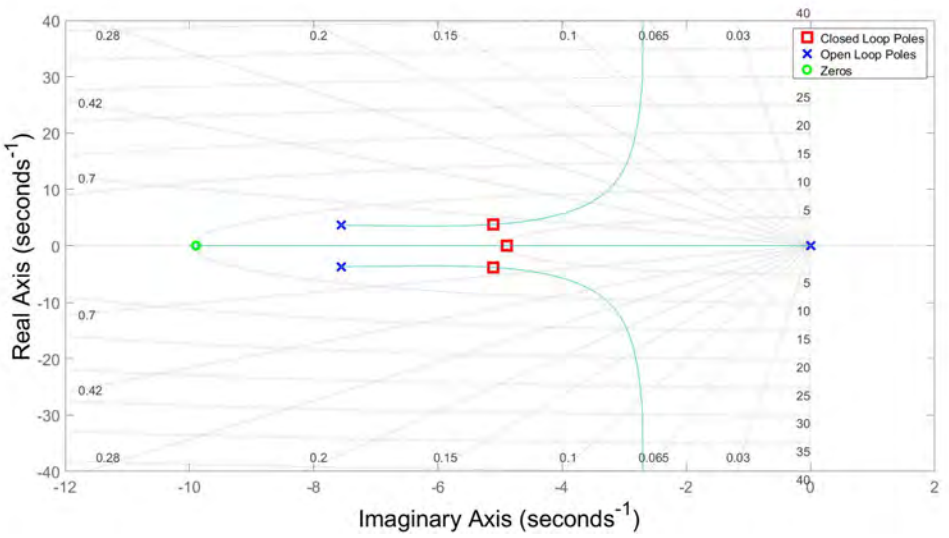


Figure 5.8: Climb rate controller - Root locus

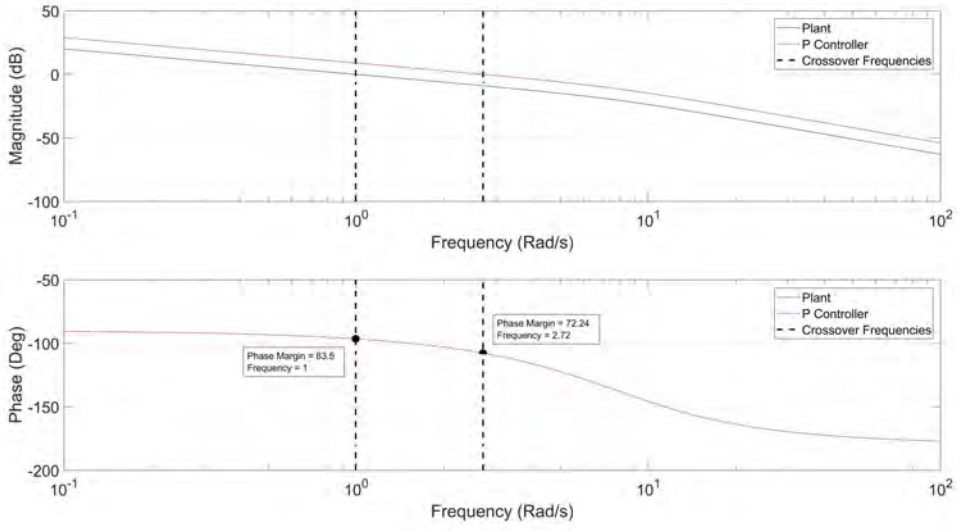


Figure 5.9: Climb rate controller - Open-loop bode plots

The bode plot shown in 5.9 shows zero change in phase due to the controller architecture. The gain however is increased and moves the crossover frequency to 2.72 rad/s. The ratio of inner and outer loop crossover frequencies is then 2.91, providing enough bandwidth between the inner and outer loops. The phase margin can then be calculated to be 72°.

As shown in Figure 5.7 there is a limiter applied to the acceleration commands. This limit is present due to the confined operational environment and ensures that the climb rate controller does not saturate the horizontal velocity controllers. The final limits are shown in Table 5.3

Limit Name	Min	Max
Acceleration Command	$-4\frac{\mathrm{m}}{\mathrm{s}^2}$	$4\frac{\mathrm{m}}{\mathrm{s}^2}$

Table 5.3: Climb rate controller limits

5.3.3.1 Climb Rate Controller Discussion

The dynamic response shows sufficient phase margin to handle unmodelled timing delays. While the ratio between the inner controller ensures this controller will not be influenced by the inner loop. The step response of the closed loop system is shown in Figure 5.10. The system has a 5% settling time of $0.747s$ and shows negligible overshoot. At $5s$ a disturbance of $10N$ is placed on the rotors and the system demonstrates the ability to continue tracking the desired setpoint with zero steady state error.

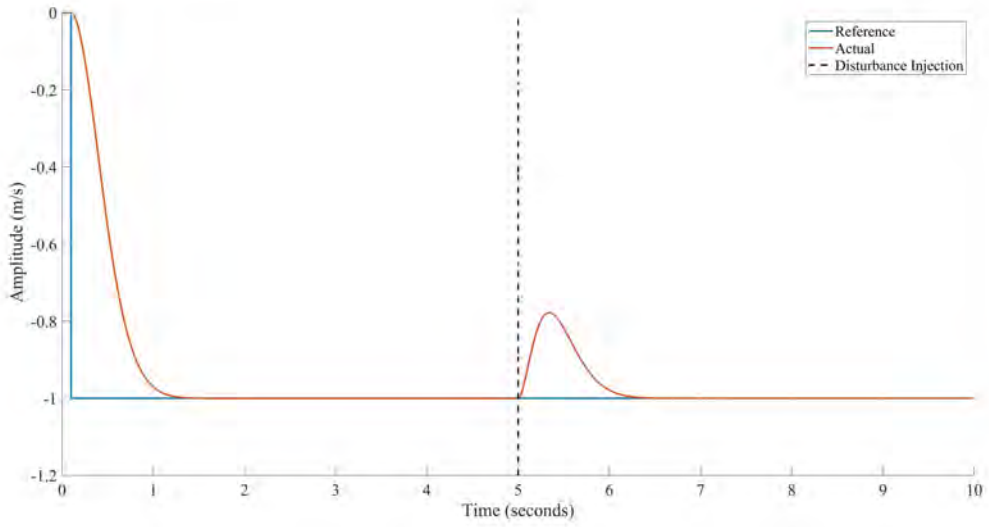


Figure 5.10: Climb rate controller - Step response

5.3.4 Altitude Hold Controller

The final stage of the vertical control system is the altitude hold controller. This controller receives a desired altitude in the earth frame and outputs a reference velocity, also in the earth frame. The closed-loop control block diagram is shown in Figure 5.11.

The altitude hold controller must be able to reject measurement errors in the inner loops, this can be achieved by adding an integrator into the controller. The system must also be able to track a set point with zero steady state error and must show a damped response with little overshoot. The system must be able to react quickly to commands, but is limited by the bandwidth of the climb rate system. The final bandwidth of this loop must be such that this controller is not influenced by the inner climb rate loop. To ensure this, the altitude controller bandwidth should be a magnitude of at least 2 smaller than the climb rate controller. As the most outer loop, this system will have unmodelled errors, the controller must be robust and exhibit sufficient gain margin and a phase margin of at least 60°.

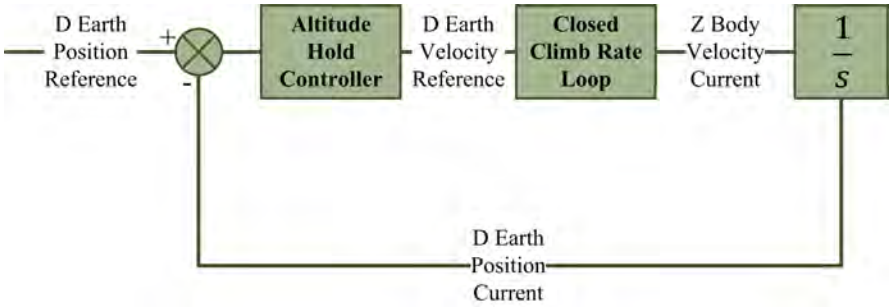


Figure 5.11: Altitude hold controller closed loop

The chosen controller architecture is shown in Figure 5.12. The proportional gain is used to vary the bandwidth to be within the limits of the system. A limited integrator is added to reject measurement errors in the inner loops, this component is represented by the faded integrator shown in Figure 5.12 and is not considered during linear analysis. The integrator shall be limited in such a way as to limit the interference of the proportional gain. This approach reduces the maximum disturbance rejection this controller can handle. To increase the bandwidth of the disturbance rejection capabilities, a PID controller architecture was also considered and the analysis was done for both control laws.

The system's dynamic response is analysed using the root locus shown in Figure 5.13 and the bode plot shown in Figure 5.14.

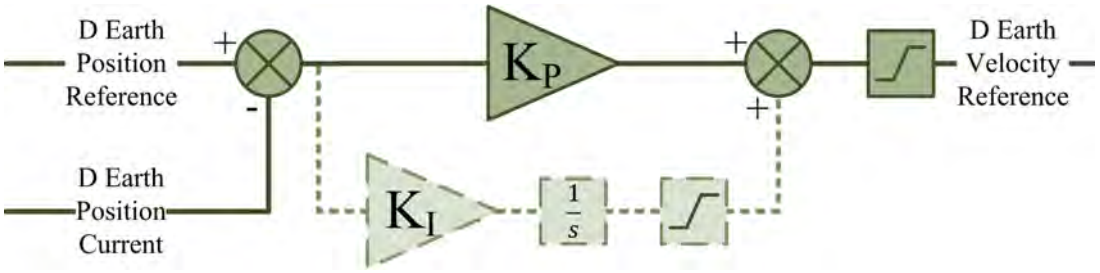


Figure 5.12: Altitude hold controller

Figures 5.13 and 5.14 evaluate a P controller against a PID controller. The dominant closed loop poles of the P controlled system are placed at $-2.03 \pm 0.58i$ and are slightly under damped with a damping ratio of 0.96.

The PID controller adds an additional pole and two additional zeros into the system. The closed loop poles of the PID controlled system are located at -6.64, $-3.59 \pm 5.03i$ and $-0.64 \pm 0.47i$. The two new zeros are placed at -0.57 and -2.06.

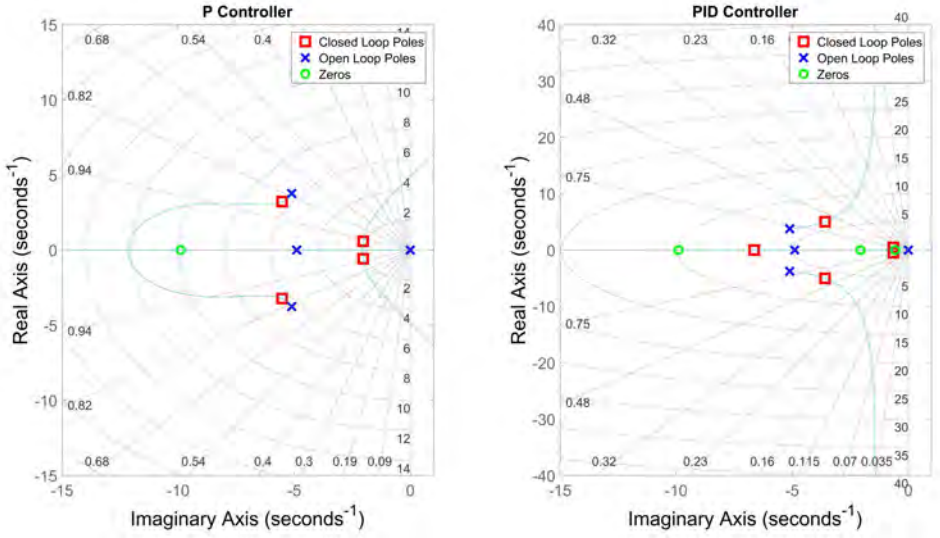


Figure 5.13: Altitude hold controller - Root locus

The bode plot shows the PID controller producing a final cross over frequency of 1.90rad/s, this response is too fast for the inner climb rate system and will need to be redesigned or discarded. The P controller exhibits a cross over frequency of 0.91rad/s, this produces a ratio of 3.03 between the

Limit Name	Min	Max
Integrator Wind Up	-0.09	0.09
Climb Rate Command	-5	5

Table 5.4: Altitude hold controller limits

inner and outer loops and a phase margin of 71.5°. The phase of the system using a P controller crosses the 180° mark at 4.85rad/s and has a gain margin of 8.6dB.

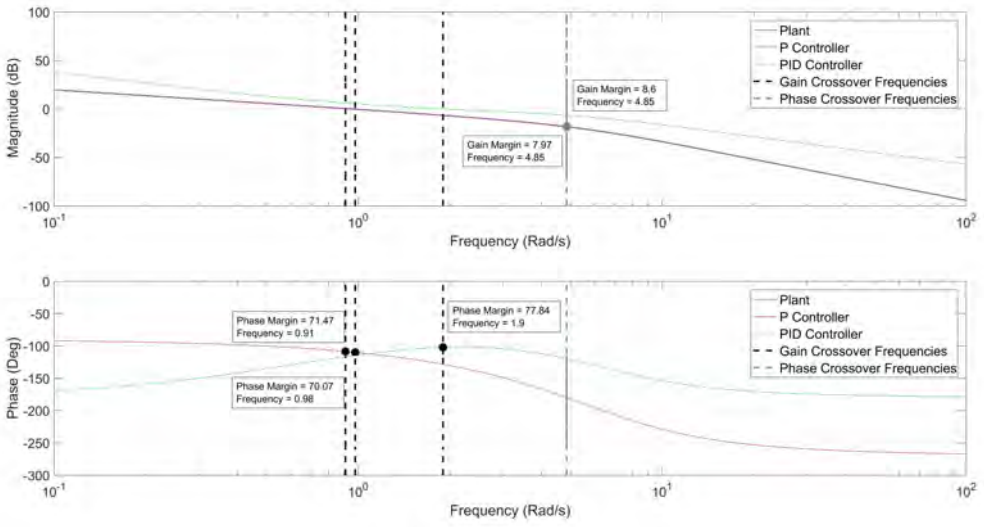


Figure 5.14: Altitude hold controller - Bode plots

To finalise the design, the two limiters are discussed. The first limiter is used to limit the effect of the integrator on the system as well as stop integrator wind up. The second limiter is used to limit the climb rate commands sent to the inner controllers. Both sets of limits are shown in Table 5.4

5.3.4.1 Altitude Hold Controller Discussion

Although both the P and PID controllers exhibit stable dynamic responses, the PID controller exhibited too fast a response and will be influenced by the inner controllers. The system which includes only a P controller exhibits a step response as shown in Figure 5.15, a disturbance of 10N is applied to the rotors at 10s. The system has a 5% settling time of 2.29s and tracks the set point with zero steady state error. The system handles the disturbance with a maximum overshoot of 0.01m and is critically damped.

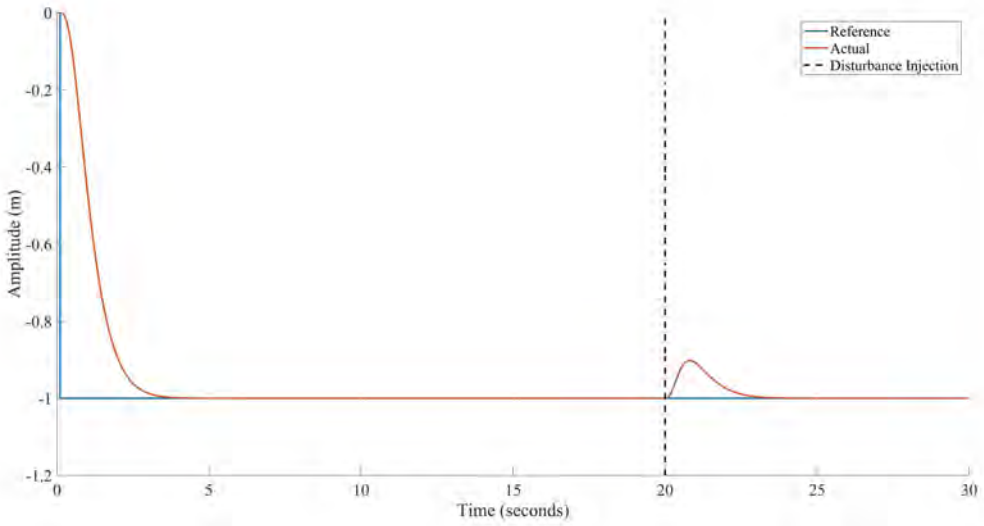


Figure 5.15: Altitude hold P controller - Step response

However, if a measurement disturbance is present in the inner loops, this system will not track a setpoint with zero steady state error. To demonstrate this a constant offset of 0.05m/s is placed on the Z-Axis velocity measurement, Figure 5.16 shows the current system cannot account for this disturbance. As shown in 5.12, a limited gain integrator is introduced into the system to help the system track steady state error.

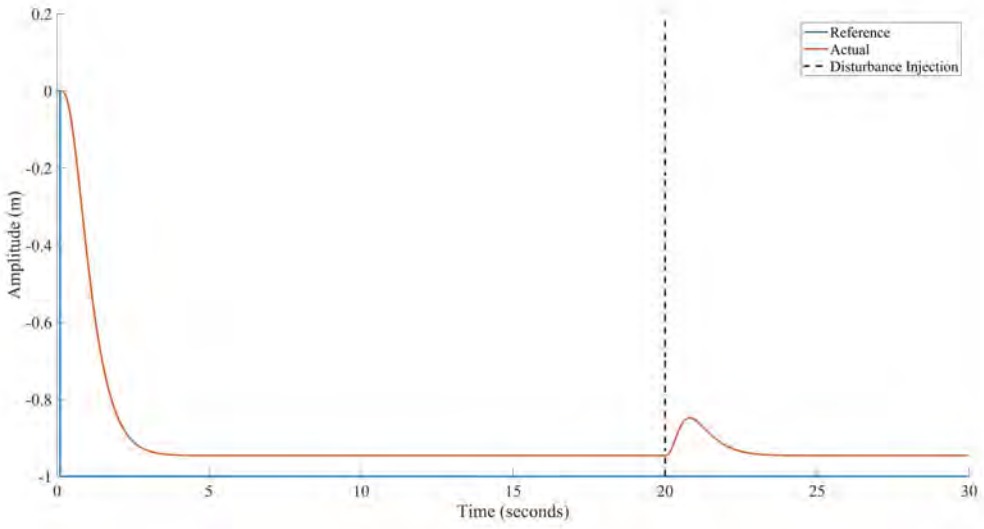


Figure 5.16: Altitude hold P controller - Step response with inner loop measurement offset

The new controller must be limited in such a way as to exhibit a similar transient response as the existing P controller. Figure 5.17 shows the step response of the new system both with and without the 0.05m/s offset in the velocity measurement. As shown, the P controller including a limited I component introduces more overshoot into the system. The limits are designed to ensure the new controller introduces less than 10% overshoot into the system.

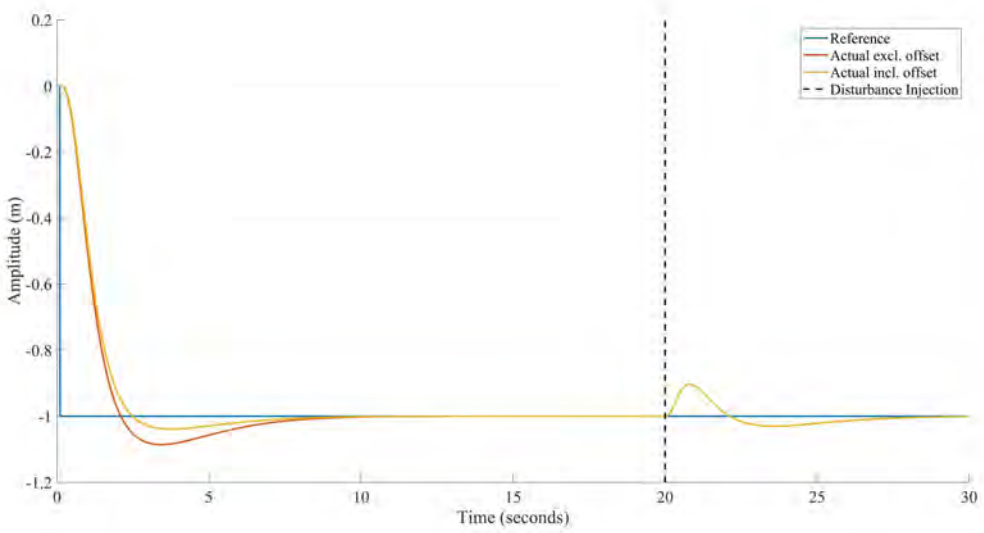


Figure 5.17: Altitude hold P with limited I controller - Step responses

All the preceding images use the ideal characteristics of the platform. Figure 5.18 demonstrates how the non-linear simulation deviates from the ideal case. The noise added to the feedback by the sensors combined with the actuation noise seen on the rotors creates small deviations from the ideal dynamics.

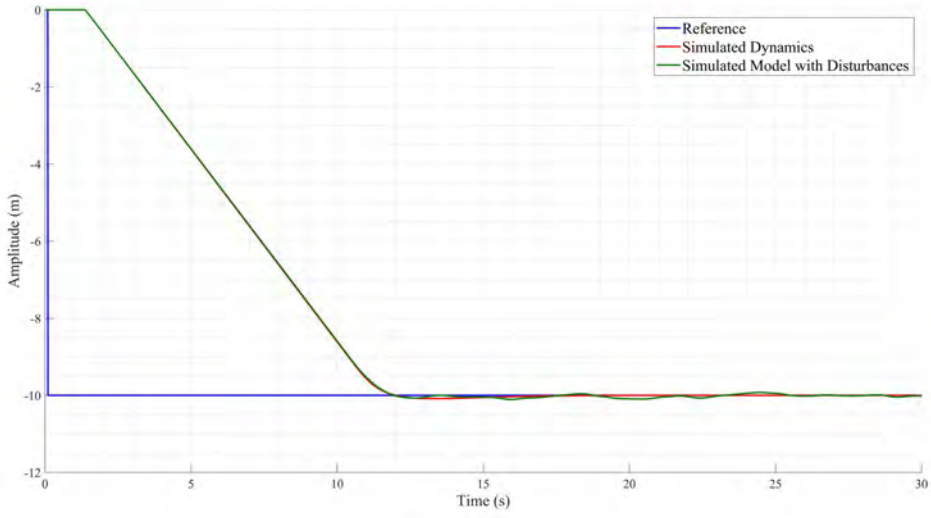


Figure 5.18: Graph showing differences in ideal and non linear simulation responses for altitude

5.4 Horizontal Control

This section describes the horizontal controller. This system is responsible for controlling the vehicle's North and East position in the earth frame, to do this the controller's most inner loop commands the pitch and roll rates of the vehicle. The narrow, confined spaces in which the vehicle must fly means it is very important for the horizontal controller to respond quickly to commands and disturbances. The limited space also limits the amount of overshoot, requiring a well damped final system. The system must ensure it stays within the thrust limits as to not affect the other controllers. The controller will need to be able reject disturbances caused by wind, sensor

offsets and unbalanced rotors. To handle these disturbances and other, the system will require an integrator in the controller. The integrator should be fast enough as to ensure the system stays within its narrow, permissible flight region, even during disturbances.

The horizontal controller is designed as two sets of four cascaded control loops, one set for roll and one set for pitch. The most inner loop controls either the roll or pitch rate of the vehicle by commanding the virtual actuators δ_ϕ and δ_θ respectively. The desired angular rates are in turn commanded by the tilt angle controller. The tilt angle controller is responsible for converting desired translational accelerations into desired roll and pitch angles. These acceleration setpoints are commanded by the linear velocity controller which receives its setpoint from the most outer global position controller. The global position controller will receive its setpoint from the waypoint generation method described in a proceeding section. This section begins by deriving the plant dynamics for roll and pitch.

5.4.1 Roll and Pitch Rate Dynamics

The roll and pitch acceleration dynamics can be derived using Newton mechanics at near hover conditions and the vehicle's inertia around the X-axis (I_{xx}) and the Y-axis (I_{yy}) respectively, the result is shown in (5.4.1) and (5.4.2).

$$\dot{p} = \frac{L}{I_{xx}} \tag{5.4.1}$$

$$\dot{q} = \frac{M}{I_{yy}} \tag{5.4.2}$$

The rotors introduce an additional time delay into the dynamics, as shown in (2.3.11). The state space equation for both systems can be derived using the current angular rates (p & q) and the current angular moments (L & M) as the system states. The state space representation for roll is shown in (5.4.3) and (5.4.4). The transfer function for roll acceleration can subsequently be calculated from the state space representation. Integrating the result produces the transfer function for roll rate as shown in (5.4.5). The same approach is followed for deriving the pitch rate dynamics shown in (5.4.6).

$$\begin{bmatrix} \dot{L} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} \frac{1}{\tau} & 0 \\ \frac{1}{I_{xx}} & 0 \end{bmatrix} \begin{bmatrix} L \\ p \end{bmatrix} + \begin{bmatrix} \frac{1}{\tau} \\ 0 \end{bmatrix} \delta_\phi \tag{5.4.3}$$

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} L \\ p \end{bmatrix} \tag{5.4.4}$$

$$G(s)_{roll} = \frac{\frac{1}{\tau I_{xx}}}{s(s + \frac{1}{\tau})} \tag{5.4.5}$$

$$G(s)_{pitch} = \frac{\frac{1}{\tau I_{yy}}}{s(s + \frac{1}{\tau})} \tag{5.4.6}$$

The roll and pitch plants both have a naturally occurring integrator, an open loop pole at $-\frac{1}{\tau}$ and no naturally occurring zeros. As shown, the plant gain is inversely proportional to the specific axis inertia. The design of the vehicle creates a smaller pitching plant gain than rolling plant gain. The design however, gives the pitching moment a longer torque arm, creating a larger actuation torque.

5.4.2 Roll and Pitch Rate Controllers

The roll and pitch rate controllers are the most inner loops of the horizontal controller and they command the δ_ϕ and δ_θ virtual actuators respectively. As the most inner controllers the outer loops of the horizontal controller are limited by the response and bandwidth of this system. Both

these systems should then utilise as much bandwidth as the physical motor-rotor system allows. Section 2.3.5 describes some of the expected disturbances where induced moments can be caused in multiple scenarios. Including being in near proximity with a wall or having unbalanced rotor pairs. The final system must track a set point with zero steady state error. To meet the horizontal controller's requirement of disturbance rejection, the roll and pitch rate controllers, as the fastest controllers, should include an integrator.

The integrator will slow the system down, which can be subsequently sped up with a lead compensator. However, the final commanded motor thrusts must be validated against the limits provided in Table 5.1. The final controller architecture is shown in Figure 5.19. The controller gain values must be chosen such that the inner rate system is robust to unmodelled errors and have a sufficient gain and phase margin.

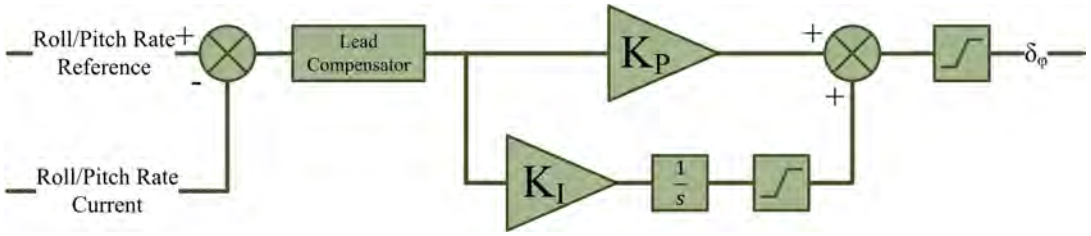


Figure 5.19: Roll and pitch rate controller design

First, the dynamic response of the controlled roll system is evaluated using the root locus shown in Figure 5.20. To maintain good damping, the two dominant poles are kept close to the imaginary axis and have a final damping ratio of 0.9. Where the placement of the slower zero dictates how much influence the integrator can have on the system.

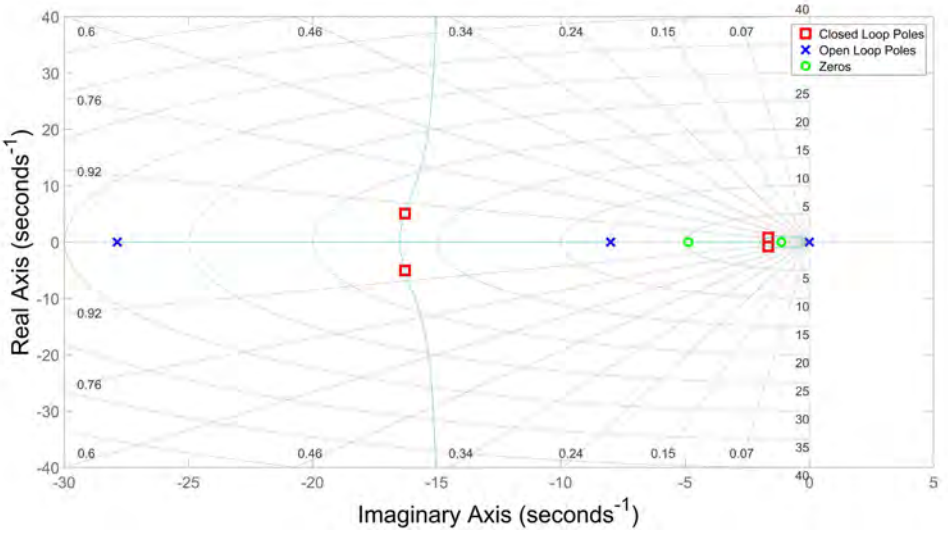


Figure 5.20: Roll rate controller - Root locus

The frequency response of the roll system is then investigated using the Bode plot shown in Figure 5.21. Unity feedback is compared against the chosen controller. The high natural gain of the rolling system gives unity feedback a very fast result with too much bandwidth for the physical system to match. There is also an insufficient phase margin of 25° in the system and no offset disturbance rejection. The controller adds an integrator to reject disturbances, this however reduces phase

even more and slows the system. The lead compensator is then used to increase the phase and bandwidth and reach a final phase margin of 80.6° with a crossover frequency of 4.75 rad/s.

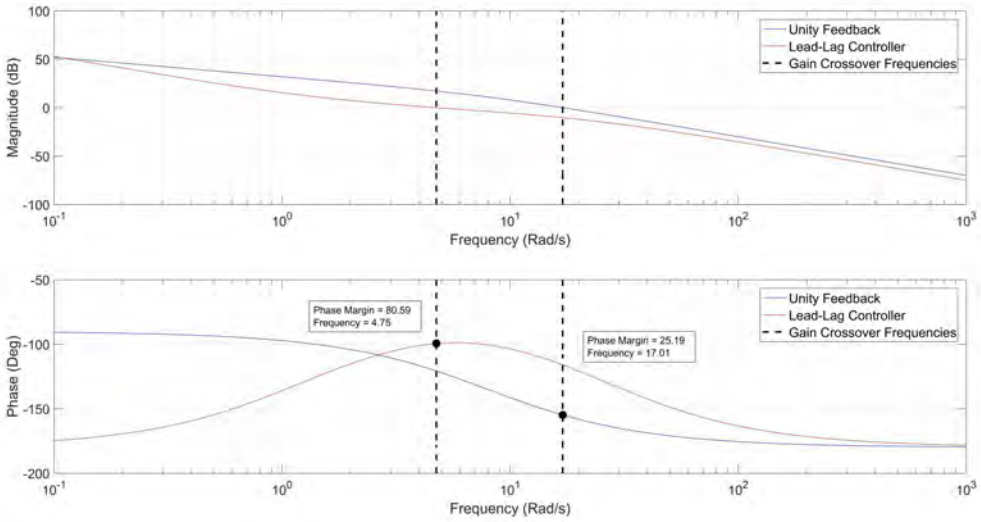


Figure 5.21: Roll rate controller - Bode plot

Next the pitch rate system is evaluated. The loci and closed loop poles of the controlled pitch system can be shown to be similar to the roll system. However, the pitching plant is naturally slower and has less gain than the rolling system. The pitch rate controller is thus required to have more gain than the roll rate controller. The Bode plot shown in Figure 5.22 is used to evaluate the frequency response of the pitch rate system. Unity feedback is compared against the implemented Lead-Lag controller. As shown the natural system with unity feedback produces a much lower crossover bandwidth than the natural roll system. As with the roll rate controller, the integrator included in the controller reduces phase and bandwidth in the system but also enables disturbance rejection. To speed up the system, a similarly placed lead compensator is used. This increases the phase and bandwidth to reach a final phase margin of 82.2° with a crossover frequency of 4.72 rad/s.

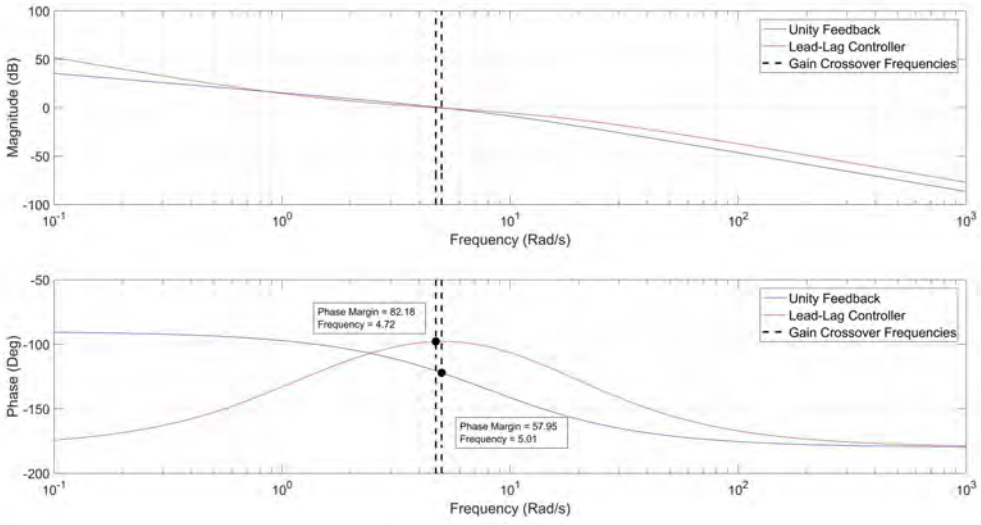


Figure 5.22: Pitch rate controller - Bode plot

5.4.2.1 Roll and Pitch Rate Controller Discussion

The dynamic responses of the roll and pitch rate systems are shown to be robust, damped and they both utilise the full bandwidth allowed by the system characteristics. The integrator term in both controllers will ensure that the rate loop can handle steady state disturbances. To stop integrator wind up however, the controllers include a saturation on the integral term. Although both systems have the same controller architecture, the physical design of the vehicle means the roll system will have a larger plant gain. It is desired that the roll and pitch rate systems have similar closed loop responses which means the pitch rate controller needs to have increased gain compared to the roll rate controller. This unfortunately leaves the roll rate controller to be more susceptible to disturbances. The flight strategy should take this into consideration and negate rolling disturbances as much as possible.

This characteristic can be shown in the time domain using step responses and the maximum impulse required of the motors. To enable a comparison between the systems, the step response of both the roll and pitch rate system is shown in Figure 5.23. To simulate a disturbance, a 0.05 Nm loss in torque is applied to the roll system at 5 s. The pitching system experiences a disturbance of 0.2 Nm. Both disturbances are calculated and equivalent to two rotors, on the same side, instantaneously losing 0.25 N of thrust.

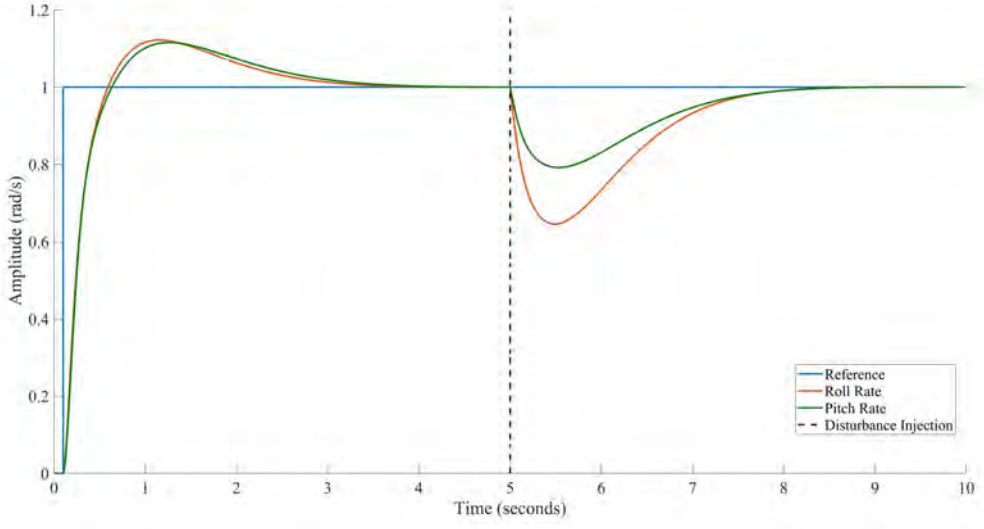


Figure 5.23: Pitch and roll rate controllers - Step responses

Both the roll and pitch closed loop systems have similar transient responses. The pitch rise time of 0.33 s is very similar to the roll rise time of 0.32 s. The pitch 5 % settling time is measured at 2.2 s which is also very close to the rolling settling time of 2.1 s. Both systems are similarly damped and have overshoot of 12 %. Limiting the integral term can reduce the overshoot however, this will also limit the disturbance rejection capabilities of the system. Both the roll and pitch systems handle the disturbance successfully and settle back within 5 % of the setpoint in 2.6 s. As expected, the roll system has more difficulty handling the disturbance.

The commands sent to the rotors during the roll step are shown in Figure 5.24 with a maximum commanded thrust of 0.49 N. Similarly, the pitching motor outputs are shown, in Figure 5.25, to have a maximum thrust of 0.69 N.

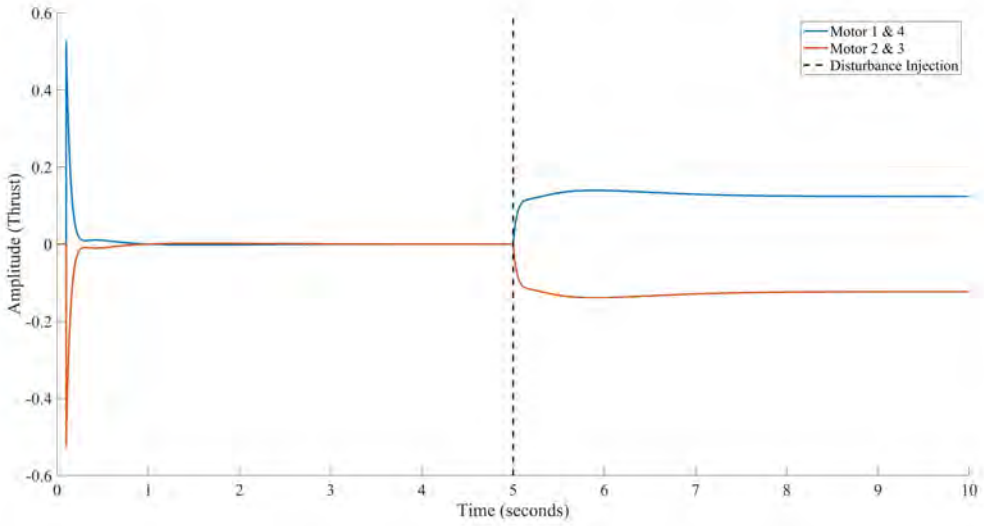


Figure 5.24: Roll rate controller - Motor commands

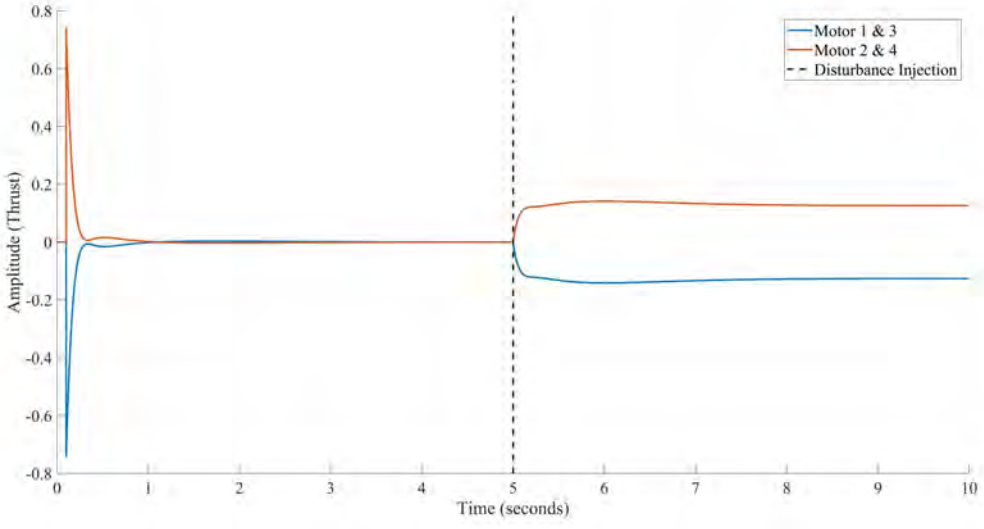


Figure 5.25: Pitch rate controller - Motor commands

Intuitively it can be strange that the pitching step response produces larger motor outputs than the rolling step. The longer pitch actuator arm would lead one to believe that the pitch system will command lower values of thrust. This is only true for a similar moment. The differing inertias entails that for a rate step response the rolling plant will produce a lower moment impulse as compared to the pitching system. To quantify the effect of the arm length versus the effect of the inertias both can be represented as ratios. The ratio between the roll and pitch arm lengths is 3.73 where the ratio between the inertias is 6.76. Therefore the pitch system has to work, ratio metrically, 1.81 times harder than the rolling system resulting in larger motor thrust outputs.

5.4.3 Tilt Angle Controller

The tilt angle controller is responsible for controlling the desired roll and pitch angles of the vehicle. The controller does this by commanding angular rates it calculates from a translational

acceleration reference in the earth frame. Using the current angular position, this earth frame acceleration reference is converted to the body frame and used to calculate the error in angular positions for roll and pitch. This error is then fed through a Proportional gain. This section makes reference to Figure 5.26 and begins by explaining the method used for converting the acceleration reference into desired angular rates.

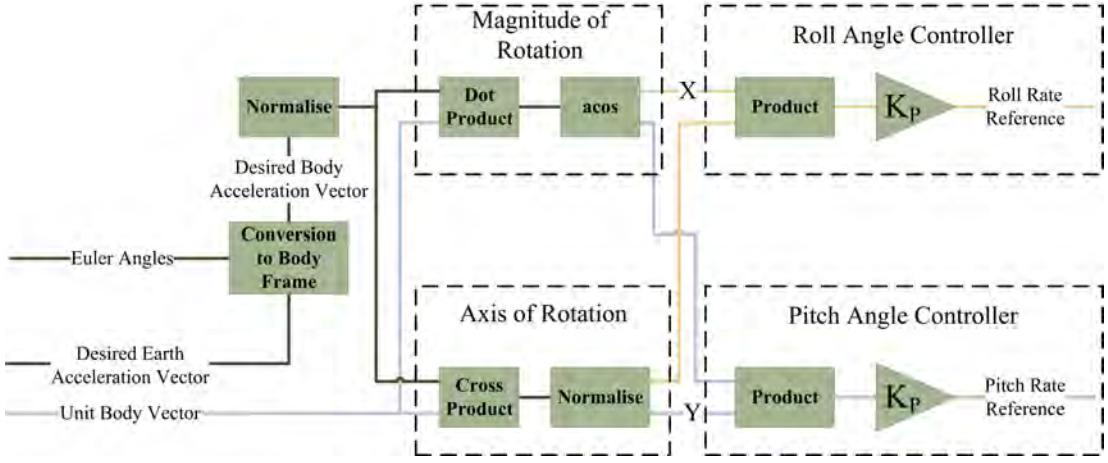


Figure 5.26: Tilt angle controller

5.4.3.1 Method of Conversion

The first step to calculating the desired roll and pitch angles is to convert the earth frame set point into a body frame reference. To enable the transformation, a rotation matrix is calculated from the current Euler angles as seen in (2.3.1). It is important to mention at this point that in order for accurate alignment, the desired earth acceleration vector must include a gravity component. The desired, now body, acceleration vector is normalised and then compared with a unit body vector. To remove any dependency on yaw, a unit Z body vector is created, which is perfectly aligned with the Z-Axis and thrust generation of the vehicle. Utilising the dot product shown in (5.4.7) the magnitude of the rotation can be calculated. Unit vectors are used, so simply taking the arc cosine of the result will produce the magnitude of rotation. The axis of rotation can subsequently be calculated by using the cross product shown in (5.4.8) and normalising the output to remove any magnitude. Figure 5.27 is used a visual aid for the preceding description.

$$\vec{a} \cdot \vec{b} = |ab| \cos \alpha \tag{5.4.7}$$

$$\vec{a} \times \vec{b} = \vec{c} \tag{5.4.8}$$

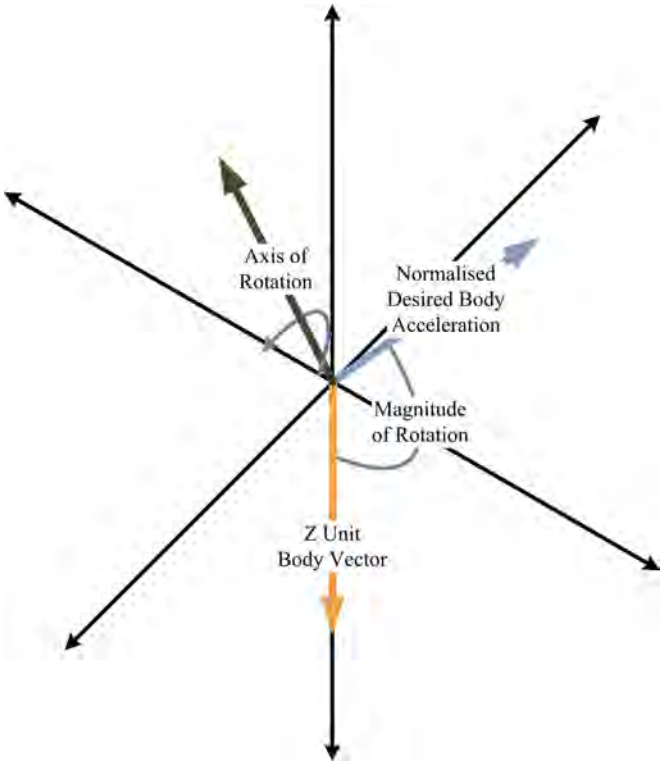


Figure 5.27: Conversion technique using dot and cross products

5.4.3.2 Roll and Pitch Angle Controllers

The linear analysis of the tilt angle controller is done by simplifying the system as shown in Figure 5.28. The additional time required to calculate the setpoints and the rotation matrix can be considered in the design by ensuring sufficient phase margin. As mentioned in the rate controller section, the outer controllers are limited by the inner loop bandwidth. The roll and pitch angle controllers must account for this by having a slower system with less bandwidth. For practical systems, the ratio between the inner and outer loop should be in the region of $2-4$. The inner rate controller includes an integrator term and can handle disturbances, thus allowing for a less complex angle control law.

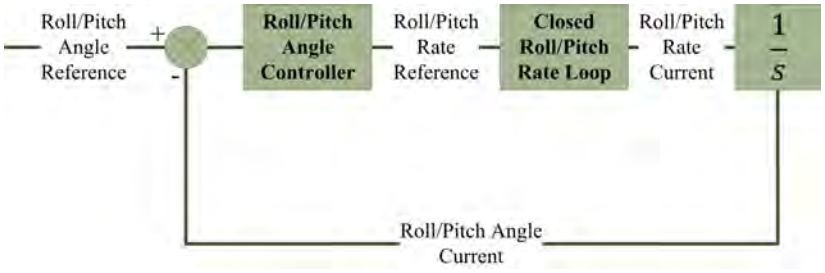


Figure 5.28: Roll and pitch angle simplified closed loops

The roll angle loop's frequency response is shown in the Bode plot in Figure 5.29. Unity feedback is compared against the chosen controller. The integration between rate and position increases the phase in the lower frequencies producing sufficient phase, allowing for a simple Proportional (P) control law in the angle loop. The final phase margin is 78°. The controller adds a bit more gain than unity feedback and increases the bandwidth while pushing the crossover frequency to 1.26 rad/s. There is now a ratio of 3.8 between the inner and outer loop. From observation there is

still more room for a faster system and increased bandwidth. However, the damping decreases as the system is pushed harder creating a need for more complex control law, with little gain benefit.

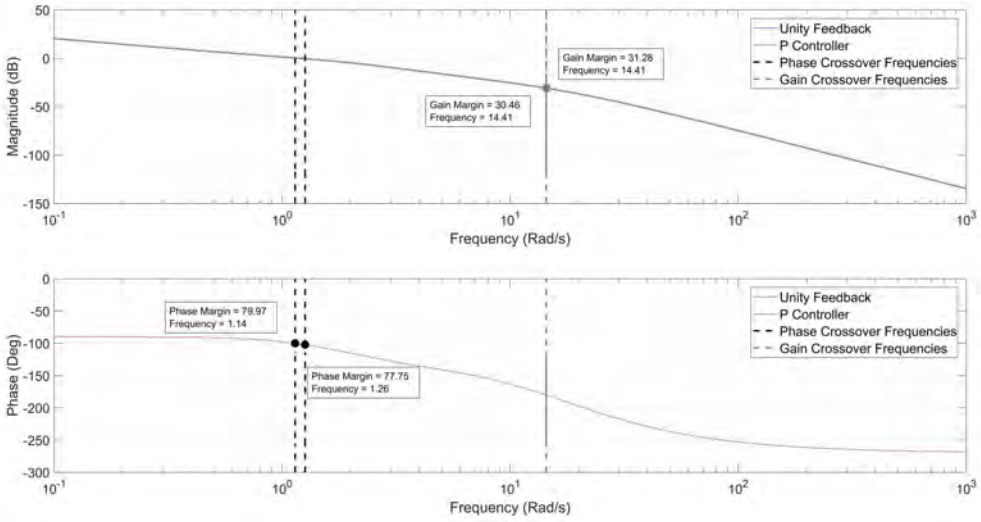


Figure 5.29: Roll angle controller - Bode plots

5.4.3.3 Tilt Angle Controller Discussion

The time domain response of the system is evaluated and discussed next. To draw a comparison between the roll and pitch systems their step responses are plotted together in Figure 5.30. As desired, the roll and pitch angle transient responses are almost identical. The final rise time for both systems is 1.65 s and they both successfully have zero steady state error. The same disturbances used in the rate loops are applied to this system at 10 s. As shown, both systems handle the disturbance successfully. Although, as expected, the pitch system deviates less from the reference.

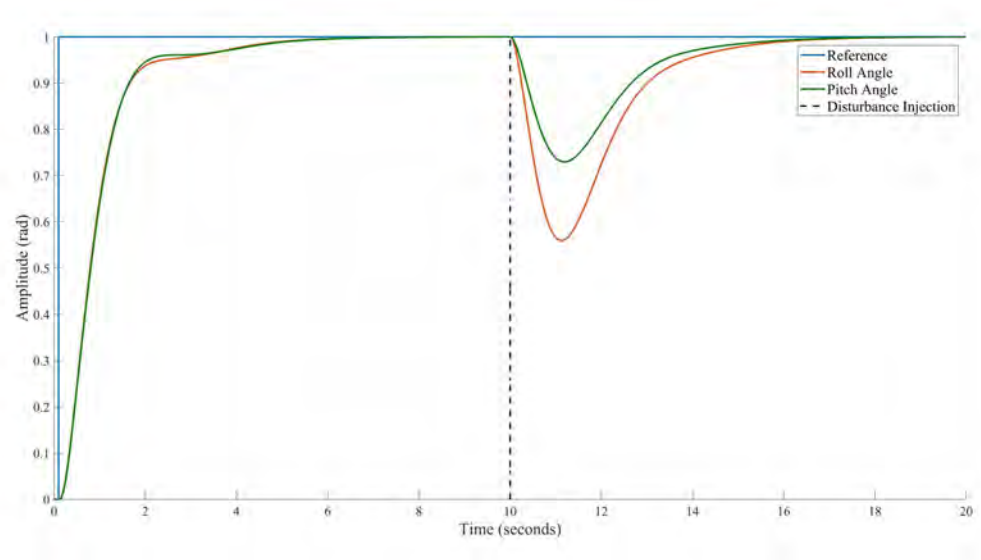


Figure 5.30: Roll and pitch angle controller - Step responses

Finally the commands sent to the motors are evaluated in Figures 5.31 and 5.32. The maximum thrust commanded by the pitch system is just less than 0.8 N. As expected the roll system commands a lower maximum of 0.57 N.

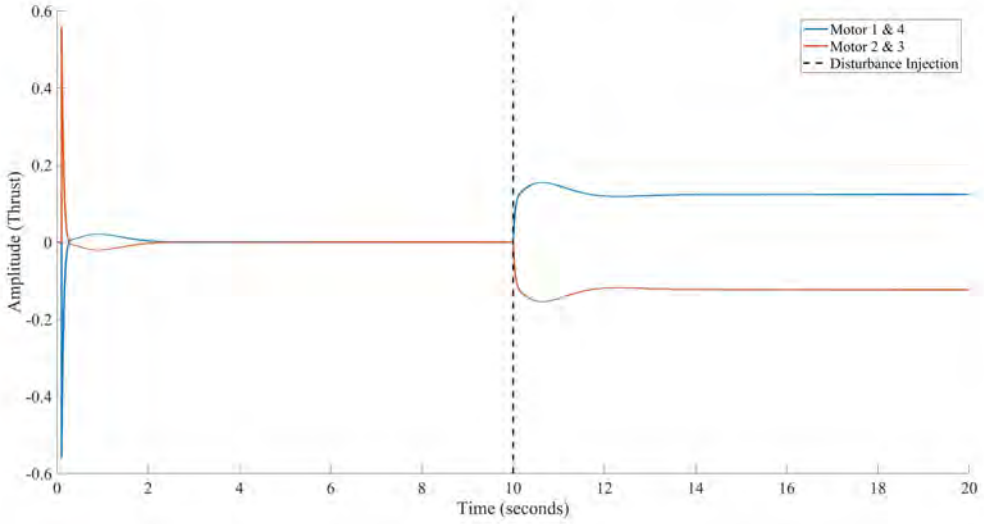


Figure 5.31: Roll angle controller - Motor commands

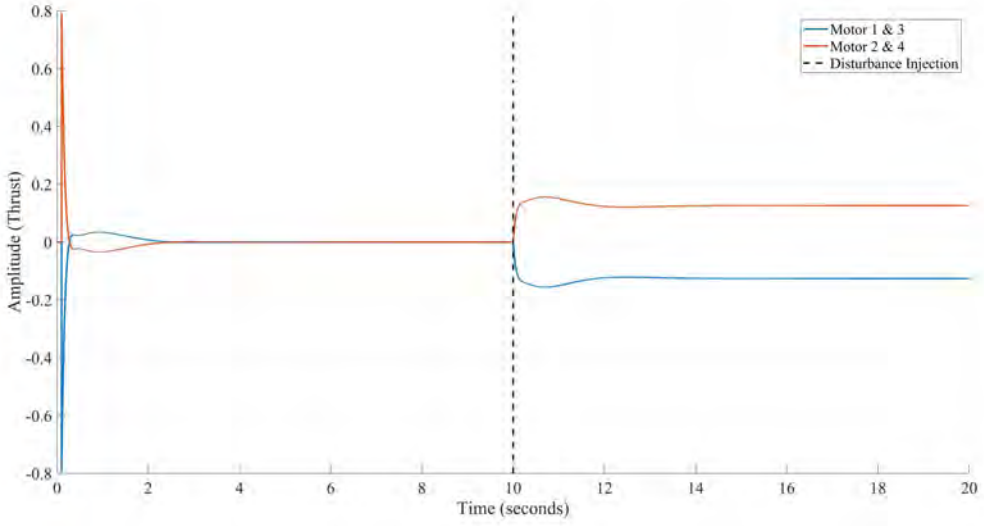


Figure 5.32: Pitch angle controller - Motor commands

5.4.4 Linear Velocity Controller

This section follows the design of the linear velocity controller. This controller is responsible for controlling the translational velocities of the vehicle along the North and East axis. This controller will receive a reference from the outer position loop and feed an acceleration command to the tilt angle controller. The tilt angle controller implementation successfully abstracts the angular position from the acceleration reference. However, the relationship between the pitch angle of the vehicle and North acceleration reference still requires a linearisation for the controller design.

For simplification the vehicle is assumed to be travelling at a maintained height in a Northern direction, with zero heading. The relationship between Northern acceleration of the vehicle can then be defined using the trigonometry of the pitch angle of the vehicle as seen in (5.4.9).

$$\ddot{N} = g \times \tan\theta \tag{5.4.9}$$

At low angles, which is expected for the vehicle, $\tan\theta$ can be approximated to θ allowing for the linearisation seen in (5.4.10) and (5.4.11). The closed loop diagram can then also be simplified as shown in Figure5.33.

$$\ddot{N} \approx g \times \theta \tag{5.4.10}$$

$$\theta \approx \frac{\ddot{N}}{g} \tag{5.4.11}$$

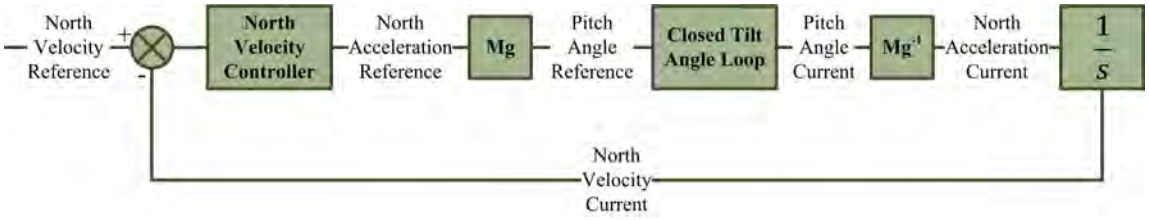


Figure 5.33: North, East simplified closed loops

The allowed bandwidth of the linear velocity controller is limited by the bandwidth of the tilt angle controller. The free integrator in the linear velocity loop will ensure that the system will track a set point with zero steady state error. However, there are expected disturbances which require more complex control than proportional control to reject. The Bode plots in Figure 5.34 assist with the design by allowing easy analysis of phase and gain in the system.

A traditional PI architecture increases the low frequency gain, however was not suitable due to the loss in phase and damping. Instead a lag compensator could be designed to limit the overshoot while enabling some disturbance rejection. The process of designing the lag compensator under went the following steps. First a proportional controller is designed to achieve the desired bandwidth, ω_{des}. The zero of the compensator is then placed far enough to negate any effect on the bandwidth. The pole has been placed to optimise both limiting overshoot and enabling disturbance rejection.

As desired the P and lag controlled systems exhibit the same crossover frequency bandwidth and negligibly different high gain profiles. Both the lag compensator and the PI controller increase the low bandwidth gain at the cost of some phase. However, the phase benefits of the lag compensator compared to the PI controller can be seen clearly. The final system is designed to have a crossover frequency of 0.42 rad/s and a phase margin of 65°. This bandwidth is a ratio 2.7 slower than the slowest loop in the tilt angle controller.

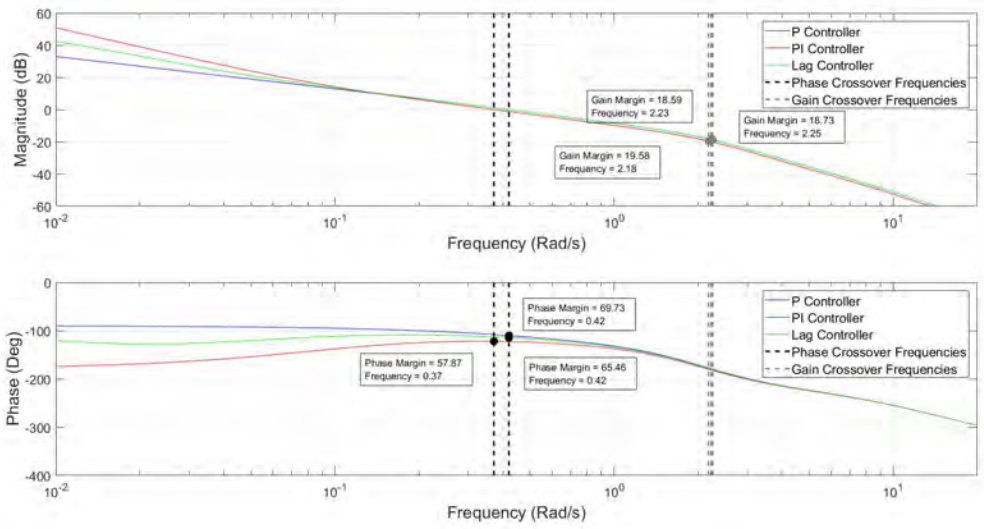


Figure 5.34: North velocity controller - Bode plots

The final placement of the lag compensator can be shown on the root locus in Figure 5.35. The compensator zero has been placed at $\frac{\omega_{des}}{10}$ with the pole a factor of 4 slower.

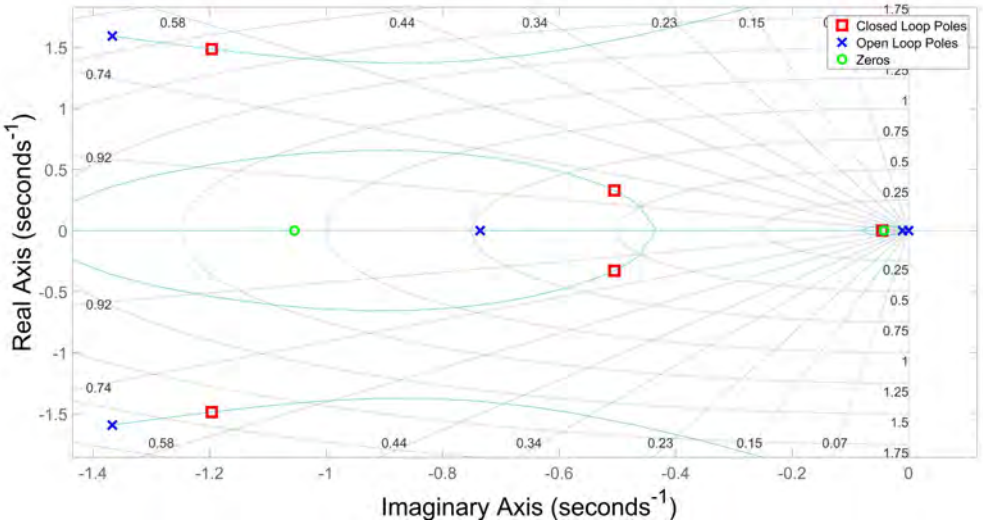


Figure 5.35: North velocity controller - Root locus plot

5.4.4.1 Linear Velocity Controller Discussion

The three controller designs all exhibit a stable dynamic response, the differences in gain and phase were identified and discussed in the Bode plot. The time domain responses and differences can now be evaluated and discussed. The step response of the P, PI and lag controllers are shown in Figure 5.36.

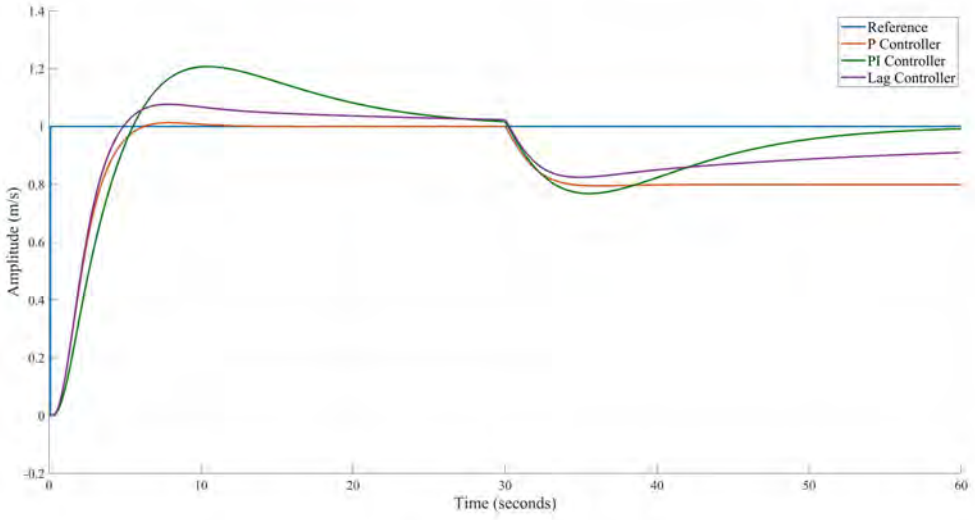


Figure 5.36: North velocity controller - Step responses with a disturbance

The proportional controller has a fast transient response with a rise time of 3.3 s. The P controller exhibits good phase margin and shows little overshoot. The lose in phase of the PI controller presents itself as a large overshoot of 20%. The integrator introduces a long tail into the system and slows the transient response to a rise time of 4.9 s. The lag compensator has some overshoot and a very similar transient response to the P controller. The benefit of the lag compensator over the P controller can be seen when a disturbance is introduced into the system. Figure 5.36 is also used to show the effect of a constant disturbance in the system by adding an external force at 30 s. The P controller is unable to reject the disturbance. The increased low bandwidth gain of the lag compensator manages to reduce the disturbance and as expected the PI controller successfully rejects the disturbance completely.

5.4.5 Global Position Tracking Controller

The position controller is the most outer loop of the horizontal controller and will be fed a reference from a waypoint generator or some other, high level flight strategy. There is sufficient disturbance rejection in the inner loops allowing the position controller to use a simple proportional controller. The final bandwidth should utilise the full potential of the inner loop velocity system. The vehicle should approach a set point steadily and with little overshoot, the final system should thus be well damped.

The Bode plot in Figure 5.37 is used for the design as it easily shows the phase and gain margins of the system. The plant is on the edge of stability and is compared with a controlled system utilising a P controller. The proportional gain is adjusted until there is sufficient phase margin of 69°and gain margin of 17 dB. The final cross over frequency is 0.16 rad/s, creating a ratio of 2.6 between the inner and outer loop.

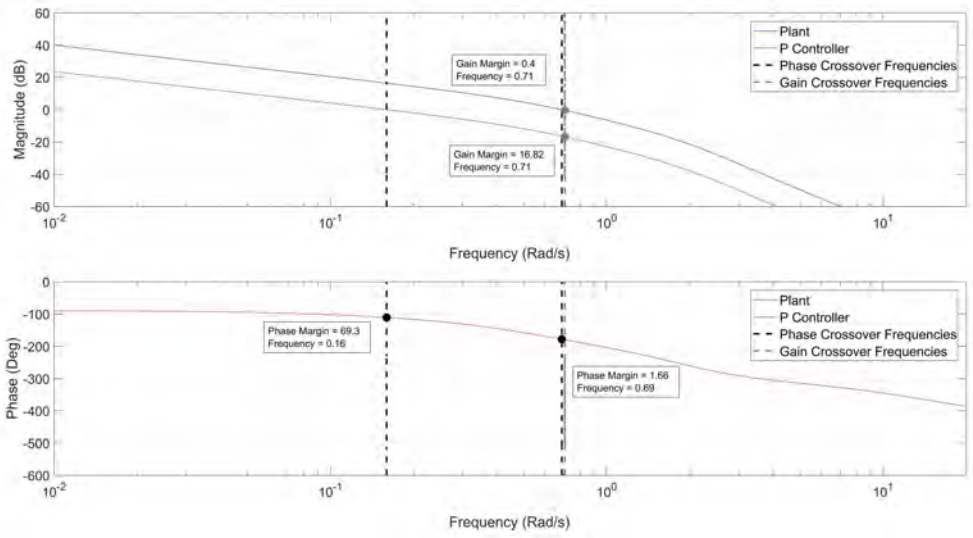


Figure 5.37: North position controller - Bode plots

5.4.5.1 Position Controller Discussion

The proportional gain increases stability in the system and reduces the crossover frequency. Figure 5.38 represents the step response of the system. The system is shown to be well damped with no oscillatory motion in the response.

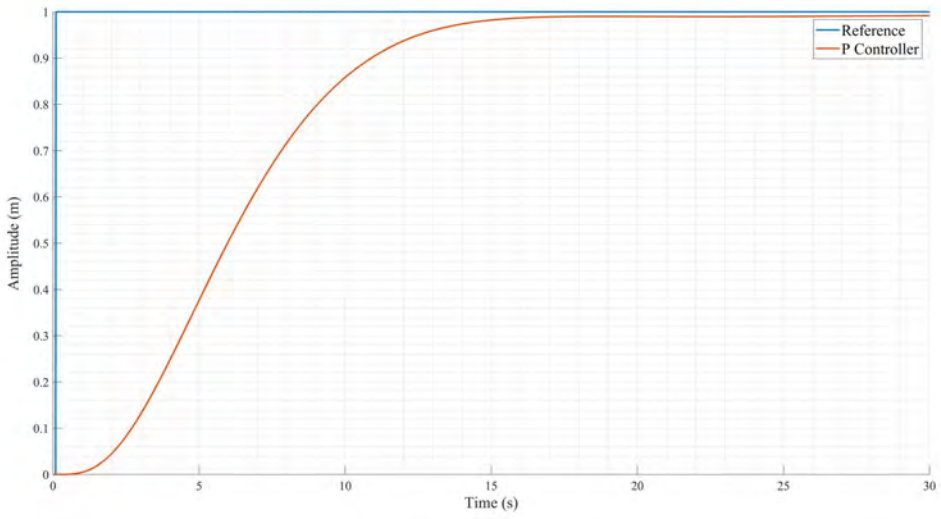


Figure 5.38: North position controller - Step response

The position controller could be commanded with large step values. To prohibit commanding large velocity values a limiter is used. Figure 5.39 is used to show the effect the saturation has for a large step input command.

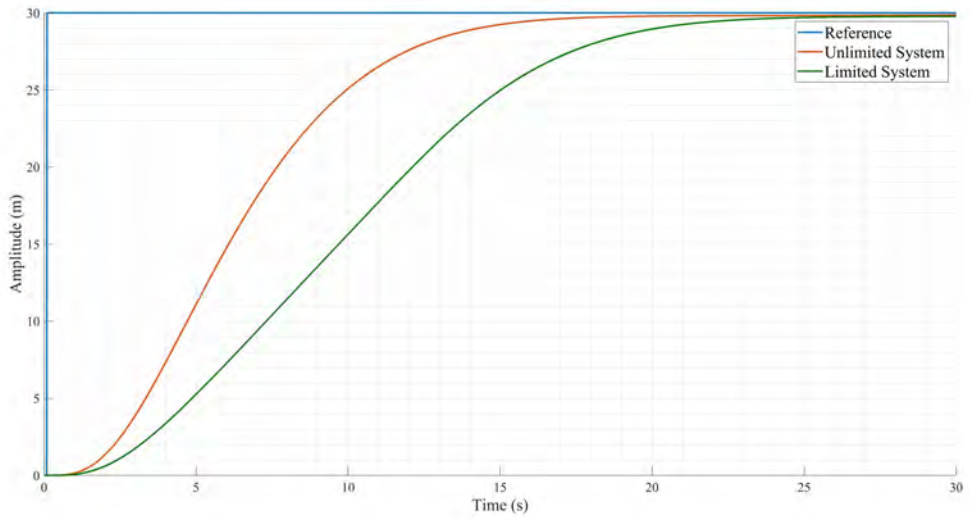


Figure 5.39: North position controller - Large step response with and without a limiter

The responses of the position controller have been shown for the ideal case. The non linear simulation generated for the project will deviate from this ideal through sensor and system noise. Figure 5.40 compares the response of the ideal dynamics to the one generated by the simulation.

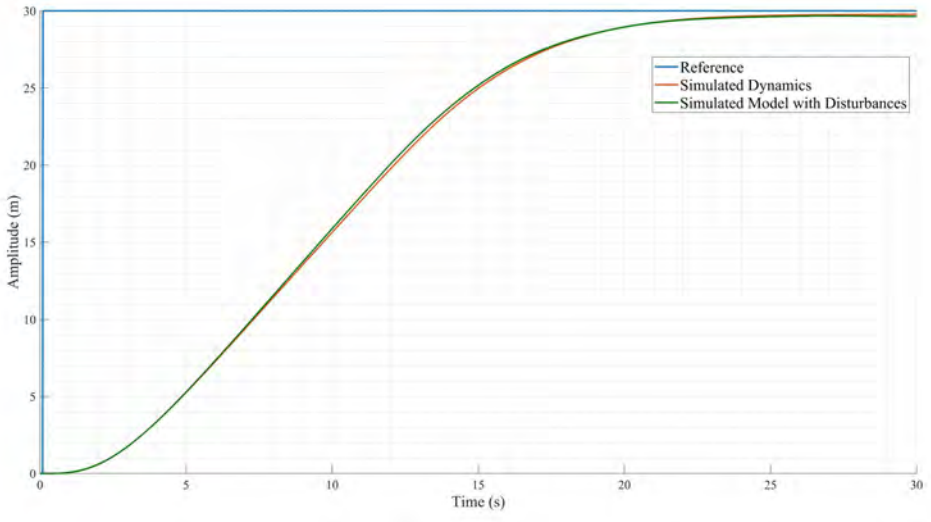


Figure 5.40: Graph showing differences in ideal and non linear simulation responses for North position

As shown, the simulated dynamics and the modelled system exhibit extremely similar responses.

5.5 Heading Controller

This section describes the heading controller which is responsible for aligning the vehicle with a desired yaw angle reference. An angle reference is given to a yaw angle controller which outputs a yaw rate reference. The inner yaw rate controller commands the yawing moment around the Z-Axis of the vehicle. Both the vertical and the horizontal controllers have been designed to operate

independently of the heading. The vehicle is however, expected to fly down a narrow channel. This calls for a method of aligning the body axis of the vehicle with a given heading in the earth frame. This heading could be given by a higher flight planning strategy.

The vehicle must be able to follow a heading setpoint with zero steady state error and have a reasonably damped response. The heading controller must also be able to reject disturbances and will require an integrator in the control law. The yaw controller must also consider that the yaw torque generation has a reduction gain due to the lift to drag ratio. The design of the vehicle also means the yaw rate dynamics produce the lowest plant gain with the largest inertia component. The juxtaposition of a low actuation torque and lower plant gain leads to the heading controller typically being slower and exhibit less bandwidth when compared to the other controllers. Before a controller can be designed, the plant dynamics must be derived.

5.5.1 Yaw Rate Dynamics

Using Newton mechanics at near hover conditions, the yaw dynamics for the vehicle can be derived, the result is shown in equation (5.5.1). $\dot{r}$ is the rotational acceleration of the vehicle and N is the instantaneous moment experienced by the vehicle around the Z-Axis.

$$\dot{r} = \frac{N}{I_{ZZ}} \tag{5.5.1}$$

$\dot{r}$ is chosen as the output of the system with the state variable chosen as N. From (5.5.1) and the rotor lag, the state space equation for the system can be derived and is shown in (5.5.2) and (5.5.3).

$$[\dot{N}] = -[\frac{1}{\tau}]\,[N] + [\frac{1}{\tau}][\delta_\psi] \tag{5.5.2}$$

$$[\dot{r}] = -[\frac{1}{I_{ZZ}}]\,[N] \tag{5.5.3}$$

$$G(s)_{yaw} = \frac{\frac{1}{\tau I_{ZZ}}}{s(s+\frac{1}{\tau})} \tag{5.5.4}$$

From the state space representation, the transfer function for the yaw acceleration can be calculated. Integrating the result produces the transfer function for yaw rate, introducing a new pole into the system, the result is shown in (5.5.4). This plant now has two open loop poles, the first pole is due to the lag introduced by the motor rotor system, and lies at $\sigma = -\frac{1}{\tau} = -8$ with the second due to the integration of yaw acceleration to rate.

5.5.2 Yaw Rate Controller

The yaw rate controller receives a yaw acceleration reference in radians per second (rad/s) and outputs the virtual actuator δ_ψ. The yawing moment is generated by air pressure on the rotors as they generate thrust, the reduction gain introduces the possibility of saturating the other controllers for large step inputs. However, as the most inner of the two heading loops, the yaw rate system limits the bandwidth of the outer yaw angle loop. The yaw rate controller must then produce enough bandwidth for the yaw angle controller, while ensuring it is not commanding thrust values above the limits described in Table 5.1. The yawing moment torque generation is also less accurately modelled and the system must exhibit high stability with large gain and phase margins. The free integrator in the yaw rate system will produce zero steady state error. The design for this controller can then be designed as a simple P controller with a non-linear saturation as shown in Figure 5.41.

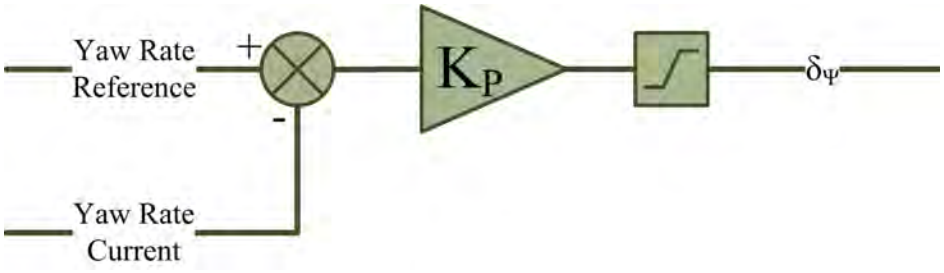


Figure 5.41: Yaw rate controller - Control diagram

The dynamic response of the proportional (P) controller is evaluated using the root locus in Figure 5.42. The controller adds no new poles or zeros. The proportional gain is used to move the closed loops and achieve the desired bandwidth. The final closed loop poles sit at $-4 \pm 1.93\,\text{rad/s}$, the poles are slightly under damped with a damping ratio of 0.9.

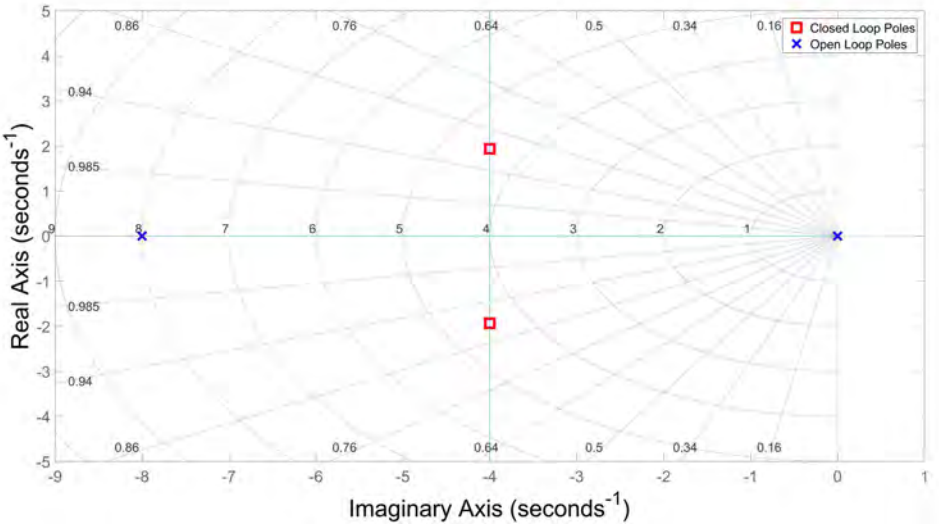


Figure 5.42: Yaw rate controller - Root locus

The bode plot in Figure 5.43 is used to evaluate the frequency response of the system against unity feedback. As expected the controller introduces no phase change into the system. Unity feedback produces a crossover frequency of $4.99\,\text{rad/s}$ which is too fast for the heading system. Reducing the gain of the system increases the phase margin to $74°$ and reduces the crossover frequency by nearly half to $2.37\,\text{rad/s}$.

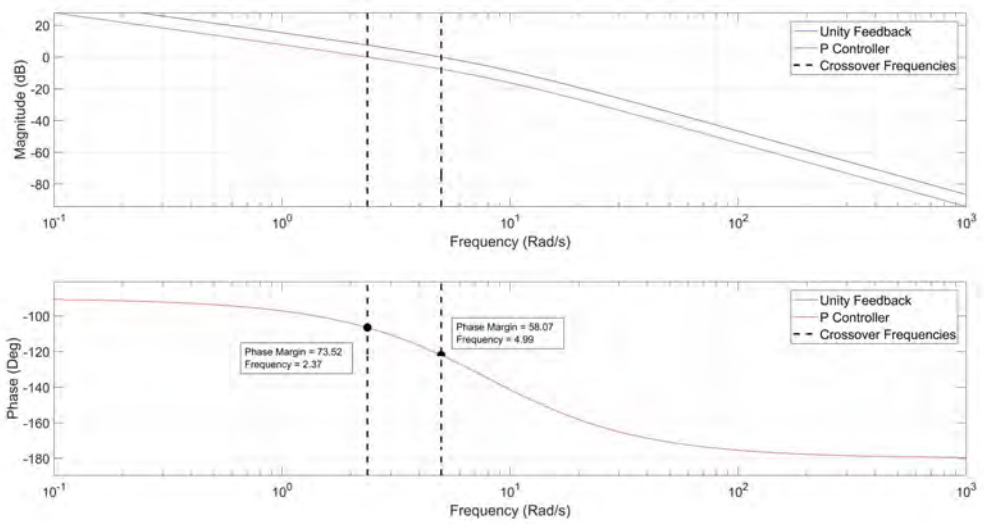


Figure 5.43: Yaw rate controller - Bode plots

5.5.2.1 Yaw Rate Controller Discussion

The controller increases stability in the system and produces the step response shown in Figure 5.44. The system has a 5% settling time of $0.9s$ and negligible overshoot. The maximum thrust commanded per motor using this system is 0.26 N, which falls in the limits of this system. The low gain of the system makes this system susceptible to disturbances and calls for an outer angle loop.

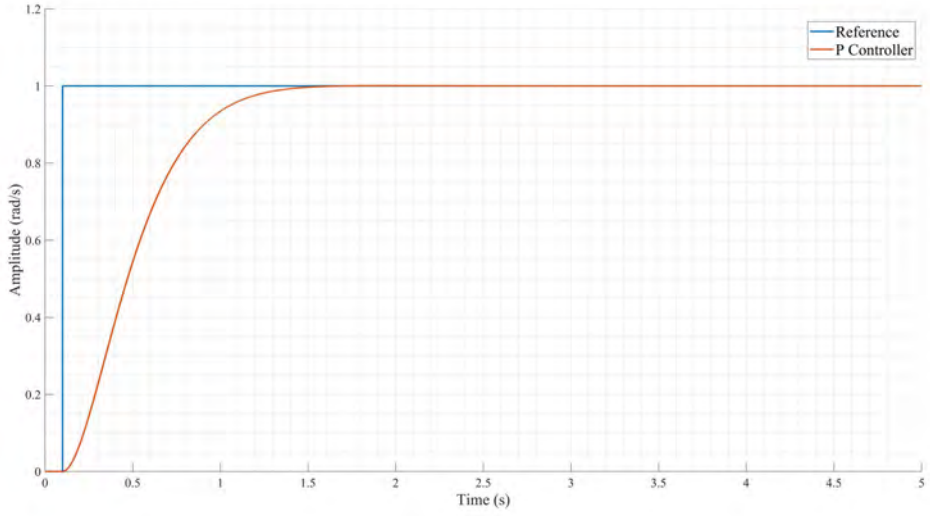


Figure 5.44: Yaw rate controller - Step responses

5.5.3 Yaw Angle Controller

The yaw angle controller receives a heading reference in radians and outputs a yaw angle rate reference in radians per second to the inner rate controller as shown in Figure 5.45. This controller is limited by the inner loop and must ensure significant bandwidth between the inner and outer loops. The system must be able to reject disturbances and requires an integrator in the system. The system must be reasonably damped with limited overshoot.

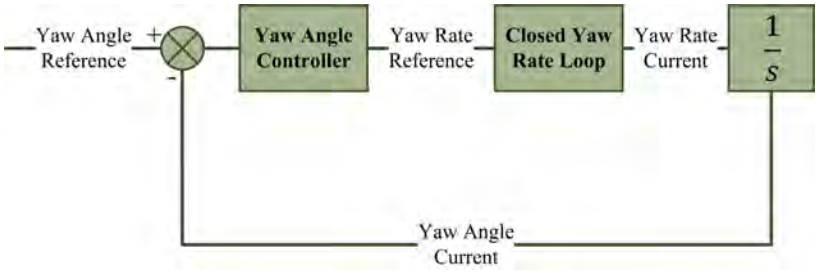


Figure 5.45: Yaw angle PI closed loop system

Initially a Proportional Integral (PI) controller was considered as shown in Figure 5.46. The proportional gain is used to move the closed loop poles and achieve the desired bandwidth. The integral term is introduced to account for expected disturbances as well as measurement errors in the rate loop. The PI controller adds a new zero and a new pole into the system.

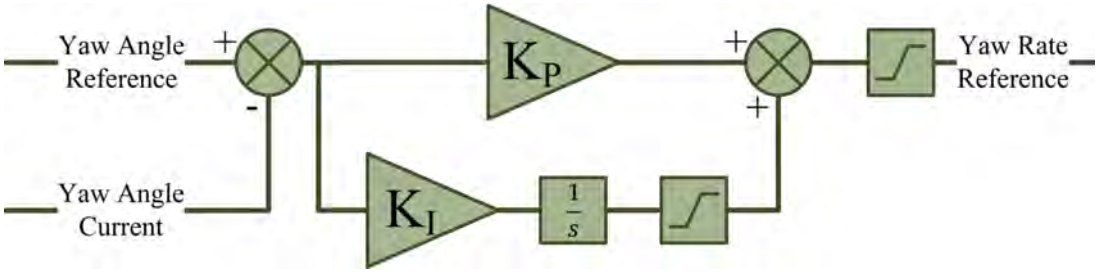


Figure 5.46: Yaw angle PI controller - Control diagram

The second scheme was designed as a Proportional Integral Derivative (PID) controller as shown in Figure 5.47, the differential term is introduced to increase the phase of the system and adds an additional zero. The differential command is fed through a low pass filter to reduce noise on the command. As shown both controllers contain non linear elements that are not considered during the linear design. The components excluded during the analysis are all the limiters as well as the low pass filter seen in the differentiator portion of the PID leg.

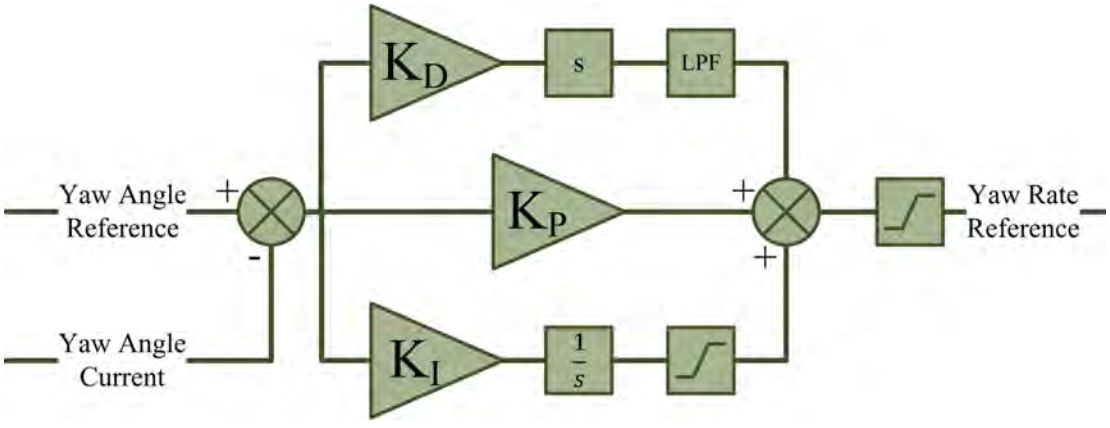


Figure 5.47: Yaw angle PID controller - Control diagram

The dynamic response of each system can be evaluated using the root locus diagrams seen in Figure 5.48. The plant has two open loop poles at the closed loop yaw rate pole locations, as well as a new pole at the origin introduced by the mathematical relationship between speed and position.

Both controllers introduce one new open loop pole at the origin and at least one zero. The PID controller introduces an additional zero into the system.

The final closed loop pole positions for the PI controller all lie on the imaginary axis and are located at -3.62, -3.26, -0.88 and -0.25. The PID controller has a dominant complex pair of closed loop poles at $-0.52 \pm 0.51i$ with the other non-dominant closed loop pole pair sitting at $-3.48 \pm 2.54i$. The PI controller has critically damped dominant poles where as the dominant poles for the PID controller are under damped with a damping ratio of 0.71.

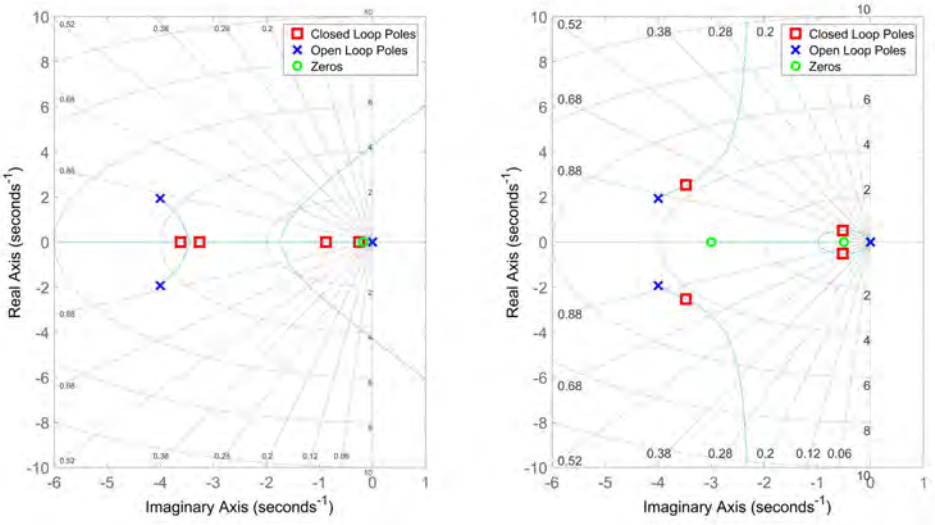


Figure 5.48: Yaw angle controller - Root locus (Left:PI, Right:PID)

The frequency response of both systems is evaluated next. Using the open loop bode plots shown in Figure 5.49, unity feedback is compared with a PI and PID controller. The first zero in both cases is placed close to the origin to limit the effect the new zero has on the system. The PID controller's second zero is placed to increase the phase of the system, allowing for more bandwidth in each leg of the controller. This additional phase allows for larger and more aggressive disturbance rejection, but will result in larger setpoints for the inner yaw rate loop.

The gain of each system has similar bandwidth around the cross over point. The extra zero in the PID controller reduces the gradient of the gain slope off and increases the total phase of the system. The PID controller exhibits the second largest phase margin of 61° which is found at 1.12 rad/s, the fastest of the three crossover frequencies. The PI controller achieves the desired bandwidth and has the slowest crossover frequency of 0.75 rad/s, this however relates to a lower phase margin of 59°. The final crossover frequency of the yaw rate system was 2.37 rad/s, resulting in a ratio with the PID controller of 2.12 and a ratio of 3.16 with the PI controller. A larger ratio implies less risk of attenuation for the outer loop.

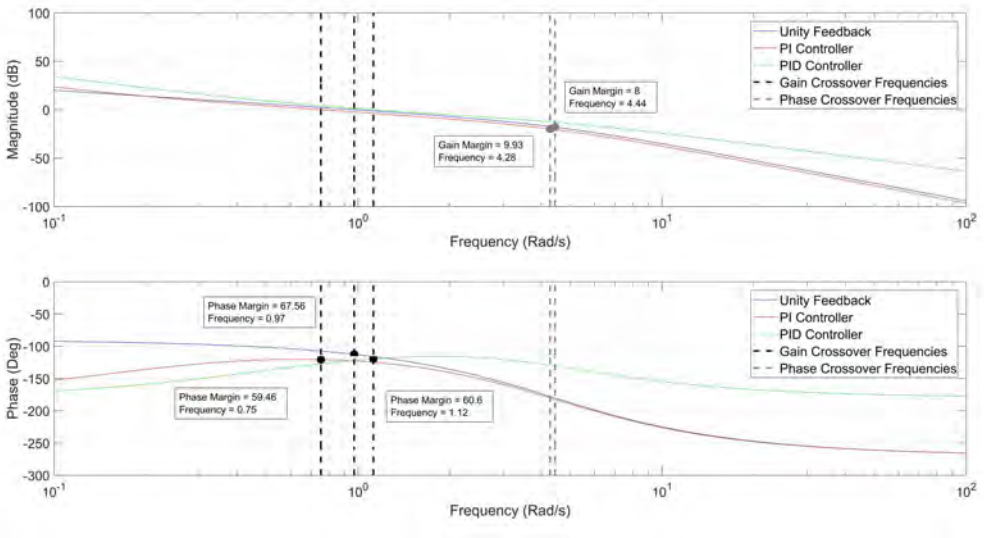


Figure 5.49: Yaw angle controller - Bode plots

5.5.3.1 Yaw Angle Controller Discussion

Each controller was added to the non-linear simulation and evaluated in the time domain including the non linearities previously unconsidered. Figure 5.50 shows the results of both controllers with and without the limits as well as the additional low pass filter on the differential gain. The PID controller without any limits or low pass filtering had a 5% settling time of 6.37s, adding in the limits and filter decreases that time to 4.9s. The PI controller had a settling time of 10.3s with no limits which was decreased to 8.8s by the addition of the limit on the integrator.

All three systems exhibit some overshoot. The limiters must be carefully chosen to reduce overshoot while also still allowing for substantial disturbance rejection. The linear PI and PID systems produce overshoot of 17% and 29% respectively. The limits for the integrators on both systems was set to ± 0.1 rad/s. This limit also becomes the maximum offset this controller can successfully correct for. Both systems had significant overshoot reduction, the PI controller now only had a 12% overshoot, with the limited PID system showing only 6% overshoot.

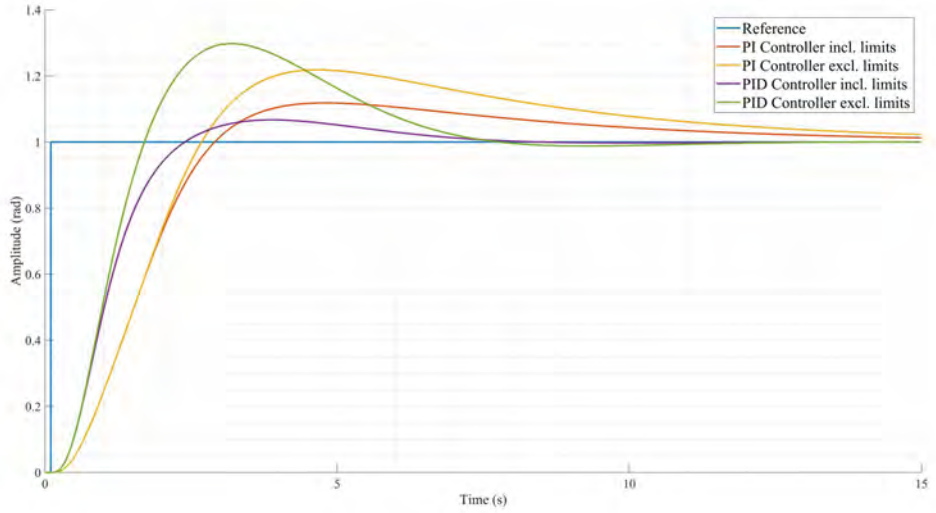


Figure 5.50: Yaw angle controller - Step responses

As mentioned the limits introduce the maximum disturbance rejection capability of each system. Figure 5.51 shows how each limited system handles a measurement offset of 0.1 rad/s in the yaw rate loop.

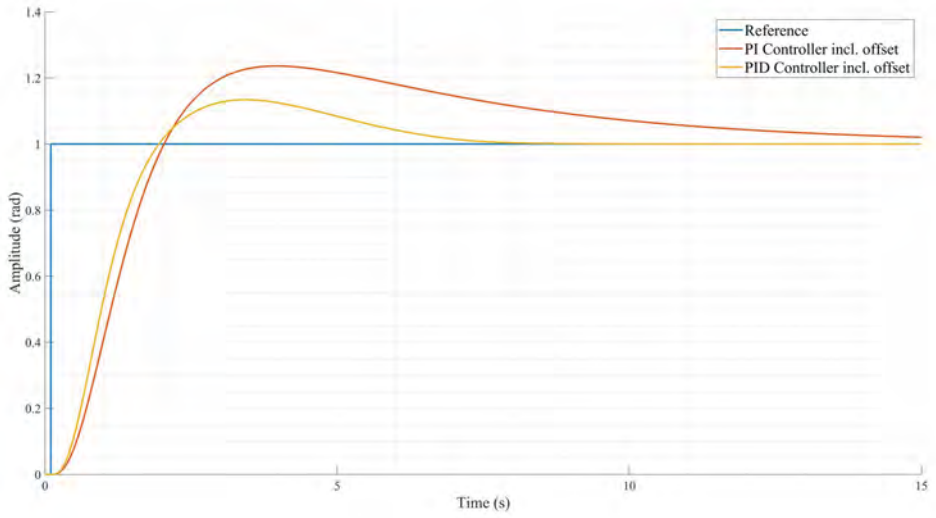


Figure 5.51: Yaw angle controller - Step responses including inner loop measurement offset

The final consideration is the impulse each system creates for the yaw rate controller, Figure 5.52 demonstrates both limited systems impulse responses. As seen the PID controller commands a larger initial setpoint, close to double that of the PI controller.

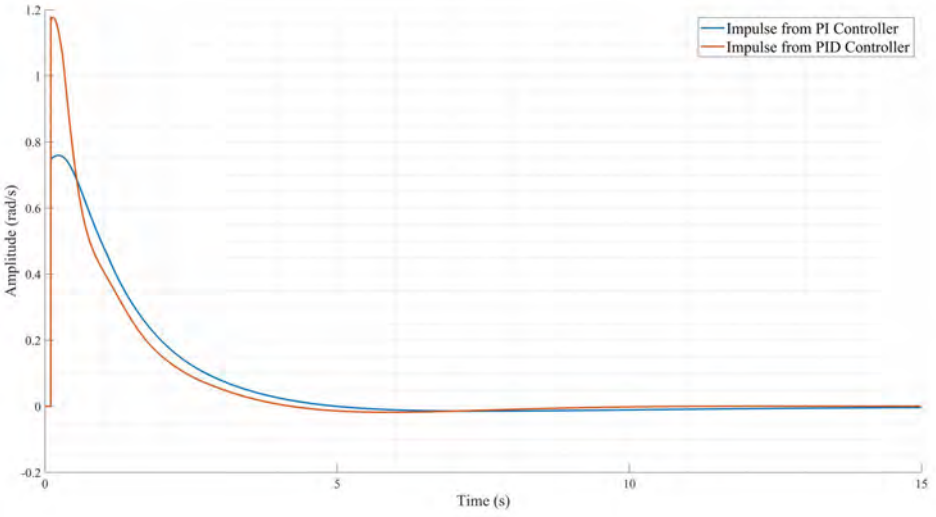


Figure 5.52: Yaw angle controller - impulses

The PI controller is chosen as the final controller implementation.

The responses of the yaw controller have been shown for the case using the ideal system dynamics. The sensor and actuation noise causes the non linear simulation to have result which varies from the ideal. Figure 5.53 compares the response of the ideal dynamics to the one generated by the simulation.

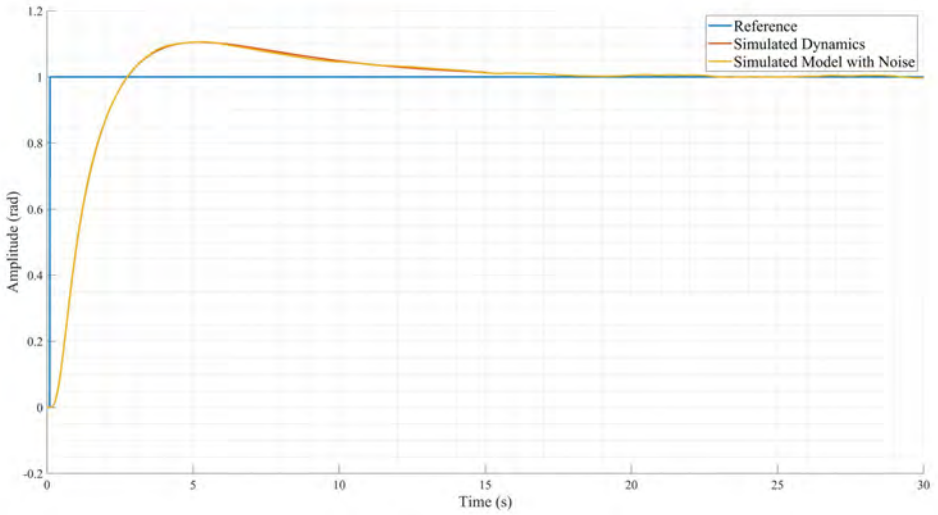


Figure 5.53: Graph showing differences in ideal and non linear simulation responses for yaw angle

Chapter 6

Flight Strategy and Obstacle Avoidance

This chapter describes the proposed and implemented flight strategy, including the obstacle avoidance technique. Now that stable flight has been achieved a more advanced flight routine can be implemented. To enable the aircraft to fly autonomously in open spaces a strategy is required to step through preprogrammed setpoints. These setpoints will be provided by a higher level path planning algorithm, for this work it is assumed these setpoints will come in the form of location waypoints. These waypoints are fed into the outer position controllers as a set of NED position references. The purpose of this system is to navigate these waypoints while avoiding obstacles. The final system thus implements a waypoint navigation algorithm and an obstacle avoidance routine that work together. The obstacle avoidance routine attempts to ensure a minimum separation distance from obstacles in the environment. The routine does this by using feedback from the proximity sensors and commanding velocities that interact with the output of the position controllers. Once the vehicle can successfully navigate to various waypoints conflict free, a flight strategy is implemented for missions in a confined and narrow space. A yaw alignment strategy is implemented to enable the vehicle to align the front with the current flight path.

The chapter begins by describing the implementation of the waypoint navigation and the heading alignment routine. The obstacle avoidance methodology is then discussed, including sensor place-ment. The obstacle avoidance routine is based on proximity measurements to the aircraft's immediate surroundings.

6.1 Waypoint Navigation and Heading Control

This section describes the flight strategy proposed for flight inside a confined narrow corridor. Although not in the scope of this work, the first step to creating a mission will be to implement a higher level route planning algorithm or waypoint system. This work creates a waypoint navigator which allows for mission locations to be preloaded into the vehicle and followed sequentially. The platform design has been previously discussed and was chosen to be a narrow, elongated design. The elongated design of the aircraft can fully be utilised in a narrow corridor if the heading and yaw is aligned as it traverses down the corridor. The section first describes the waypoint system proceeded by the strategy for yaw alignment.

6.1.1 Waypoint Navigation

The waypoints are loaded into the system as a set of North, East and Down position step references. The references are preloaded into the drone at mission start through a ground station. Once a mission is started the set points are fed into the relevant position controllers.

The ground station can program an acceptable position error limit for each waypoint. Once the limit in all three axes is reached, the waypoint navigator begins a timer which, when lapses, steps on to the next waypoint in the list. The ground station also provides the user with the option to move onto the next waypoint prematurely if desired. Figure 6.1 presents the state machine used for this function.

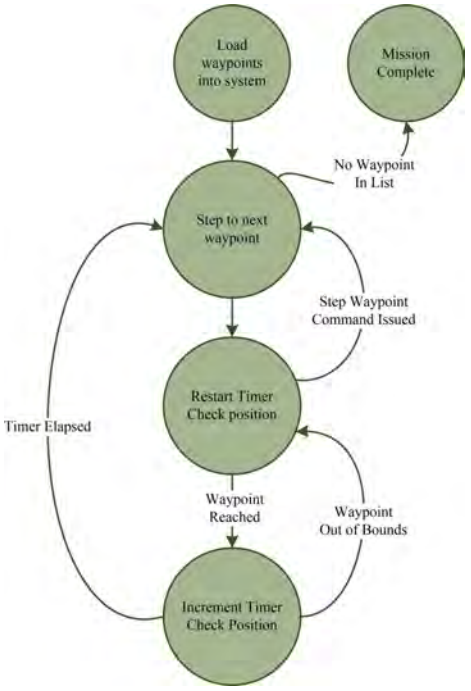


Figure 6.1: Waypoint navigator state machine

The system assumes that all references are reachable by the vehicle and will continue to pursue the target. A more advanced path planning algorithm can be used to detect unreachable positions and create alternative waypoints during flight.

Figure 6.2 shows a simple followed flight path set up by the waypoint generator. The blue line shown is the flown path, while the cyan dots represent the waypoints.

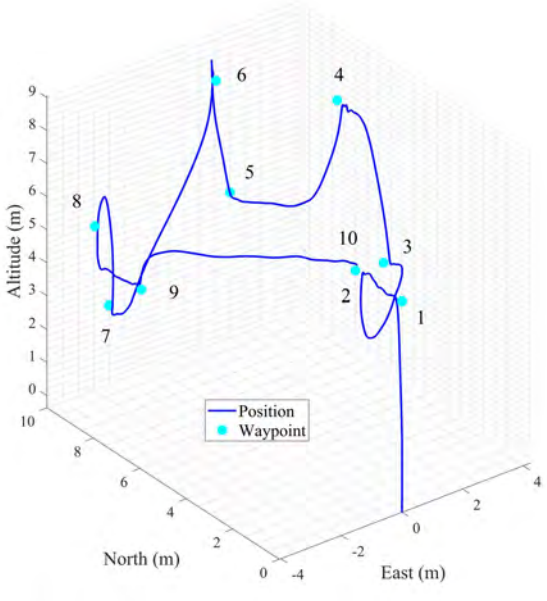


Figure 6.2: Simple waypoint flight

Figure 6.3 shows the response of the above flight, plotting North East and Down versus time.

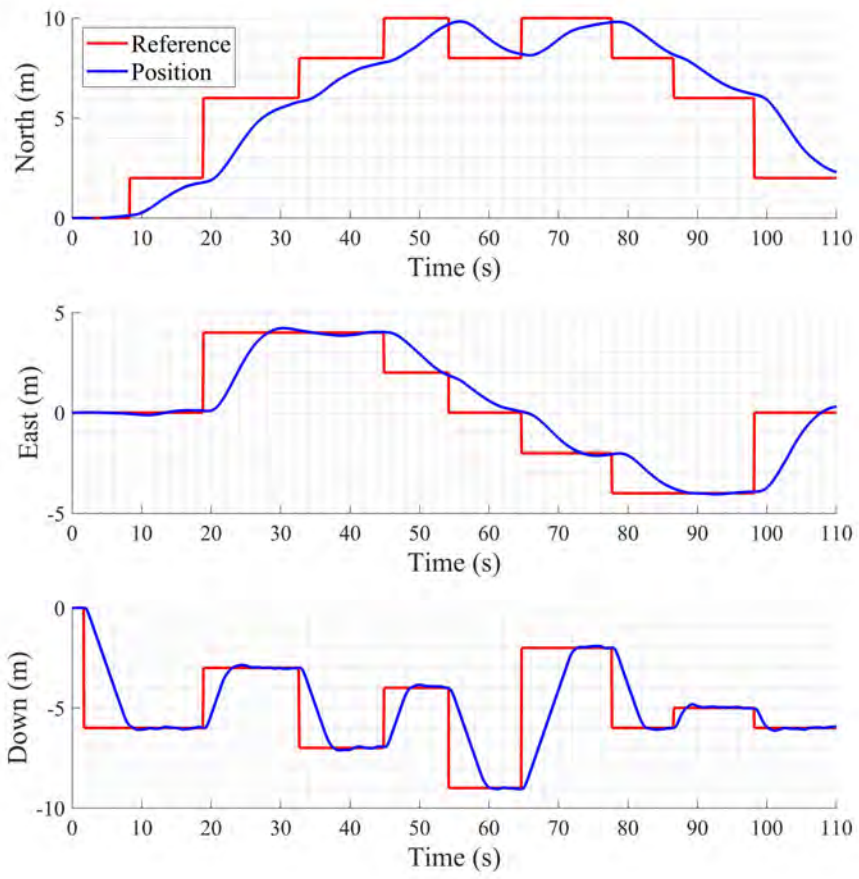


Figure 6.3: Simple waypoint flight - North, East, Down positions

6.1.2 Align Yaw With Heading

The strategy for yaw alignment is to align the X-Axis of the drone's body frame with the current actual velocity vector of the vehicle. This achieves a constantly forward facing drone assisting in tight spaces as well as a more predictable sensor feedback for missions requiring additional sensors.

The yaw angle controller designed in Section 5.5.3 works on the basis of a yaw angle error. The heading alignment controller would be required to calculate a yaw error to be controlled by the yaw angle controller. In a similar way to the tilt angle controller, this angle error can be calculated using the dot and cross product between two vectors. The first vector is the current velocity in the body frame, while the second vector is the desired alignment vector. For alignment of the X-Axis the vector would be a unit vector containing only an X component. The X and Y components of the body velocity are extracted to calculate the magnitude of rotation where the axis of rotation should always be around the Z-Axis. The cross product is utilised to calculate the direction of the rotation.

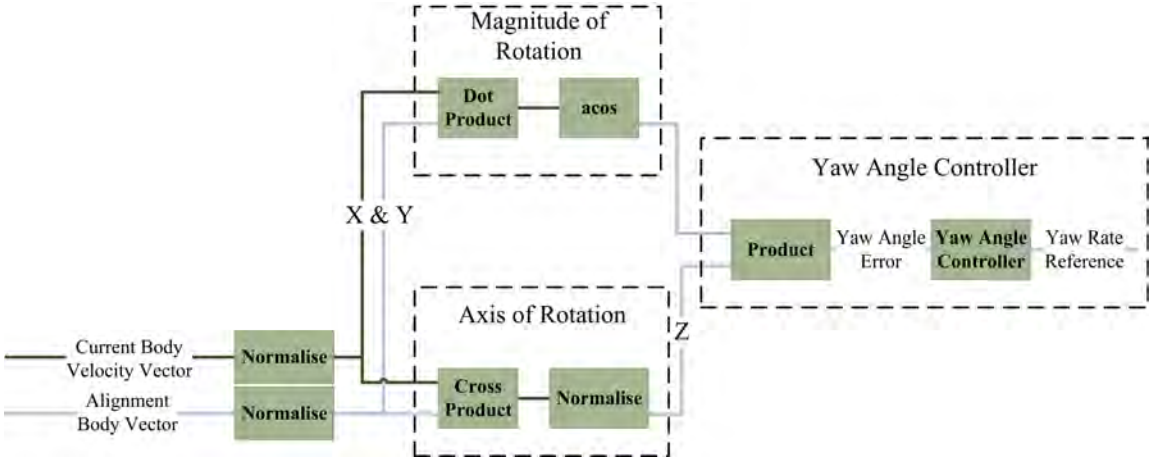


Figure 6.4: Yaw alignment controller

6.1.2.1 Yaw Alignment Discussion

At low speeds, while approaching the next position reference, the proposed yaw alignment method can produce widely varying results. For this reason a minimum velocity magnitude is set, where below that limit the current yaw angle will be maintained. To demonstrate the effectiveness of the heading alignment, the drone is commanded to fly in a circle by commanding a North sine and East cosine velocity reference. Figure 6.5 is the output from that experiment. The orange line represents the current heading of the vehicle. As seen, the vehicle begins facing due North and as the vehicle starts to fly its path, the vehicle follows and aligns itself accordingly.

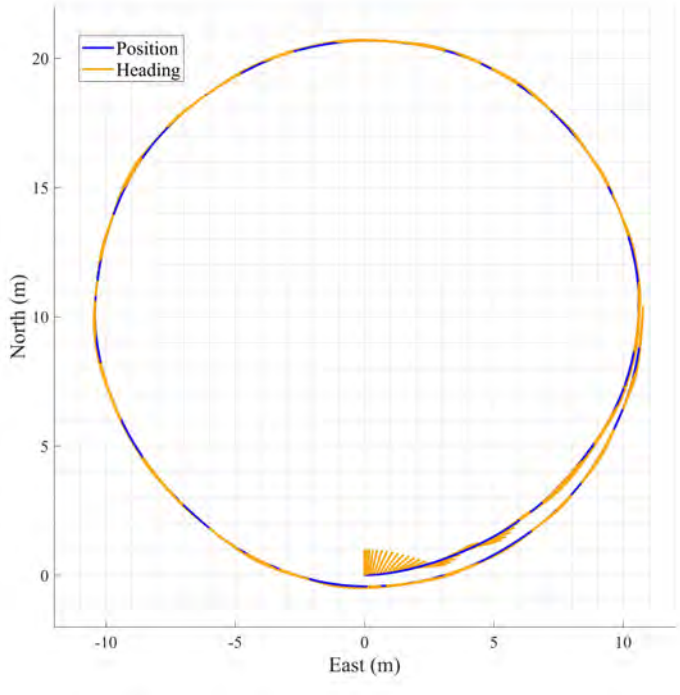


Figure 6.5: Yaw alignment controller utilised for circular flight. Orange line represents the current heading

6.2 Implementation of the Proximity Sensors

Typical sensors used in this kind of application would be either a form of proximity sensor, or in a high end application, a sweeping, high-rate laser range finder. This work assumes an ultrasonic sensor configuration and is modelled accordingly.

The measurements from ultrasonic sensors are subject to low amplitude noise. Appropriate low pass digital filtering can be applied to the sensor by the data collection algorithm. There is still expected variations and errors in the sensor measurement due to uneven surfaces and large angles of deflection, corrupting the data from the sensors. A low amplitude band limited white noise block can be used to represent these variations as shown in Figure 6.6.

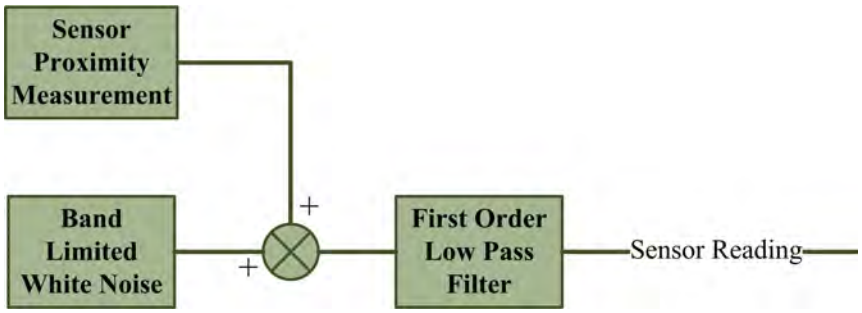


Figure 6.6: Implementation of sensor noise

The sensor transmits a sound wave signal and waits for the signals return to measure distance. The sensors utilise a time-out to ensure a desired bandwidth for the sensors but also limits the range. Anything beyond this sensor's maximum range will return as that maximum value. A typical ultrasonic sensor will have a measurement rate of around 10 Hz - 30 Hz.

The proximity sensor placement is of critical importance to the functioning of the obstacle avoidance routine. The steps between proximity measurements are blind spots and can lead to the sensors missing an obstacle and subsequent crashing of the vehicle. The placement can be broken up into two planes, namely the horizontal (XY) Plane and the vertical (Z) plane.

Figure 6.7 shows the horizontal placement of sensors used in this work. There are eight sensors, all placed at 45° from each other. It is evident from the image that there are large blindspots in the combined sensors field of view. However, a real ultrasonic sensor measures in a cone, thus increasing the view of each sensor.

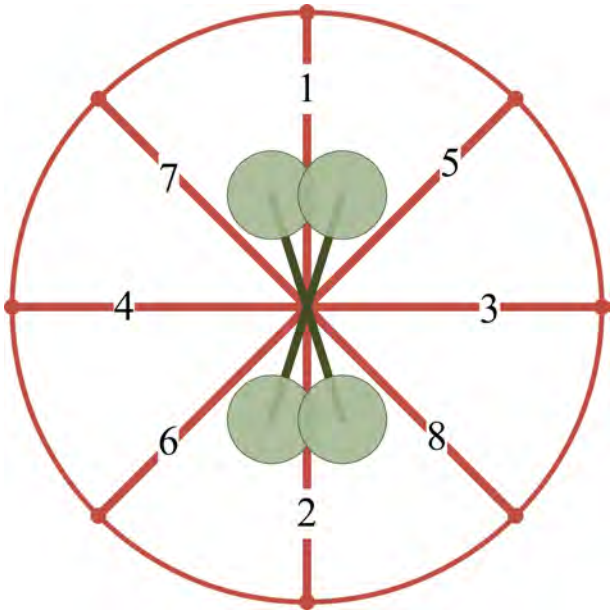


Figure 6.7: Sensor placement for obstacle avoidance

The vertical sensors used are placed in a simple up, down formation placed in the middle of the vehicle. In practice it would be beneficial to have a large cone for the vertical sensors, as to cover the widest possible field of view.

6.3 Obstacle Avoidance

The craft is required to navigate in an unknown environment without collision. The obstacle avoidance routine is responsible for ensuring the vehicle maintains a minimum set distance away from walls and other obstructions. When not in close proximity to any obstructions the obstacle avoidance routine should have a negligible effect on the vehicle and the effect should increase as the vehicle approaches a collision condition. The limited flight range of the vehicle dictates that the obstacle avoidance method should minimise deviation off of the desired path and allow the vehicle to reach the waypoints set by the waypoint generator when possible. The chosen obstacle avoidance controller should not degrade the performance of the normal flight controller.

6.3.1 Obstacle Avoidance Implementation

The requirements for the obstacle avoidance routine are mentioned above while different methodologies for collision avoidance were investigated in Section 2.5.4. This section describes the methodology behind the chosen obstacle avoidance routine. The system works on the basis of the potential field method investigated in Section 2.5.4.2 and draws inspiration from a traditional spring damper system.

A typical spring and damper create forces governed by equations (6.3.1) and (6.3.2). The spring force is calculated by taking the displacement distance (x_{dis}) of the spring from rest and multiplying it by the spring's constant (K_{spring}), which is a measure of the spring's stiffness. The damping force is proportional to the rate at which the damper is being displaced and similarly, is multiplied by a damping coefficient (K_{damper}). The damper will increase damping in the system and reduce oscillations caused by the spring.

$$F_{spring} = -K_{spring} \times x_{dis} \tag{6.3.1}$$

$$F_{damper} = -K_{damper} \times \frac{dx_{dis}}{dt} \tag{6.3.2}$$

Using the maximum sensor distance as the resting distance, a virtual spring and damper can be created as depicted in Figure 6.8.

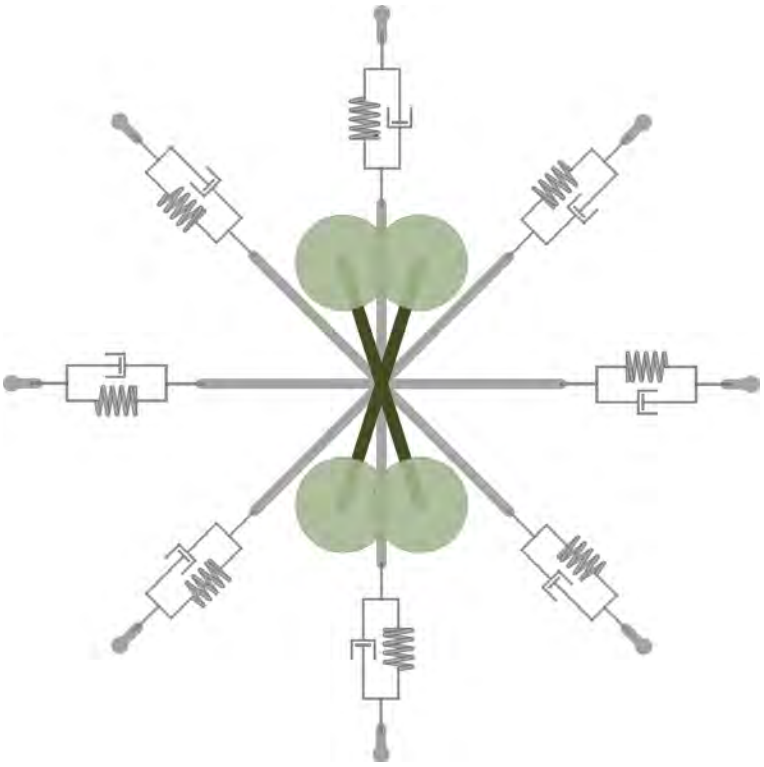


Figure 6.8: Visual descriptive aid for a virtual spring damper sensor system

The spring force can be used to push the drone away, but should not pull the vehicle towards an obstacle. The damper is added to reduce the oscillations caused by the spring as it comes in and out of contact with the wall.

The named sensor error is calculated by subtracting the current sensor measurement from the maximum sensor range and squaring the result. This quadratic relationship causes the obstacle avoidance controller to have a large effect when close to obstacles and less of an effect when obstacles first come into measuring distance. The effectiveness of using higher order systems is prevalent in the potential field method.

The corrective sensor values are summed together to create an obstacle avoidance vector. This provides abstraction to the actual sensor placement used in practice. As long as the angle of each sensor is known a single vector can be created and used in the obstacle avoidance controller. The controller attempts to drive each of the combined contributions to zero. This condition is met when there are no obstacles, or when all opposing sensors measure the same result, maintaining a equilateral distance from each wall. Equations (6.3.3) - (6.3.5) show how each sensor is combined. The sensor naming convention seen is specified in Figure 6.7 and α is the angle between the forward facing and sensor directions.

$$X_{prox} = \sum_{n=1}^{8} Sensor_n \cos \alpha_n \tag{6.3.3}$$

$$Y_{prox} = \sum_{n=1}^{8} Sensor_n \sin \alpha_n \tag{6.3.4}$$

$$Z_{prox} = Sensor_{down} - Sensor_{up} \tag{6.3.5}$$

To mimic the existing position controller structure a North, East and Down velocity commands are required. Since the sensors act in the body frame it is required to rotate the final proximity vector in to the NED frame. Figure 6.9 demonstrates how the controller feeds into the existing overall controller structure. As shown, the obstacle avoidance controller subtracts from the position controller's velocity set point to create a new velocity set point. Both set points are limited before being subtracted from one another. To ensure that the obstacle avoidance takes control to avoid collisions, the position controller is limited to a smaller value than that of the obstacle avoidance controller.

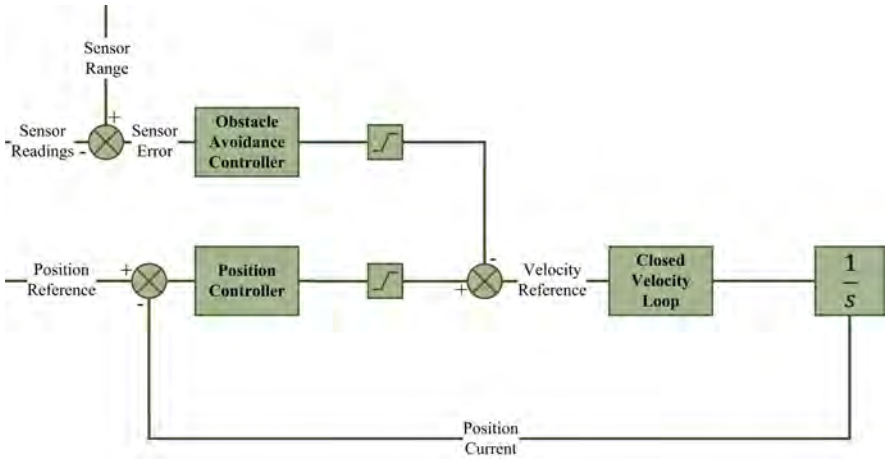


Figure 6.9: High level view of obstacle avoidance controller

6.3.2 Obstacle Avoidance Controller

As shown in Figure 6.9, the obstacle avoidance controller generates a velocity setpoint reference and works in conjunction with the existing position controller. A 3-axis stable position controller has been designed in the controller chapter. The allowed gain and bandwidth of the obstacle avoidance controller can be inferred from the existing controllers. To limit the oscillations when approaching a wall, a derivative controller component is added. This component is calculated by taking the velocity along each sensor arm and applying a gain to each calculated velocity. It must be ensured that the derivative controller component is only active when the specific sensor is in measurement range. Figure 6.10 shows the implementation of the controller.

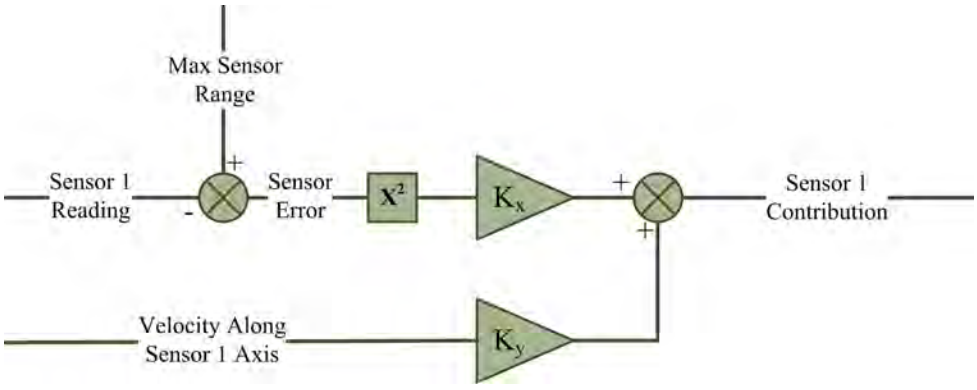


Figure 6.10: Individual sensor controller

A final consideration is the arm length of the sensor and the gains used to create the signal. The sensor maximum range should allow the vehicle to fly with none of the sensors active, but should also provide enough time for the system to respond. The gain of the controller should be considered

at worst case, this occurs when the sensor is at a distance of close to zero, creating a sensor error equal to the square of the maximum sensor distance. To maintain a stable system it is intuitive that the K_x gain of the sensor controller will decrease quadratically as the arm length increases. Based on expected corridor sizes, an ideal arm length range of 2.5 m - 4 m was set as an initial value for consideration. Increasing the derivative gain increases the total damping in the system allowing for a higher K_x gain.

The selection of the saturations, arm length and gains will decide how close the vehicle will be allowed to a given obstacle.

6.3.3 Obstacle Avoidance Discussion

The discussion is required to assess the feasibility and stability of the proposed obstacle avoidance routine. A simple method to assess this is by disabling the position controller and allowing the obstacle avoidance routine complete control of the aircraft's velocity reference commands. To create this scenario wall segments are placed at -0.5 m and 4 m North and East, with a floor at 0 m and roof at -6 m Down. The horizontal sensor range is set at 3.5 m with the vertical sensor range set to 2 m. The craft is started at position (0, 0, 0), in the bottom left corner of the four walled room.

The images shown in Figure 6.12 show the North, East and Down position respectively of the vehicle as it is controlled by the obstacle avoidance controller. The images on the right show the velocity set points sent by the controller to the inner velocity loop. As shown the vehicle is stable and moves steadily to the centre of the room with the velocity commands tending towards zero.

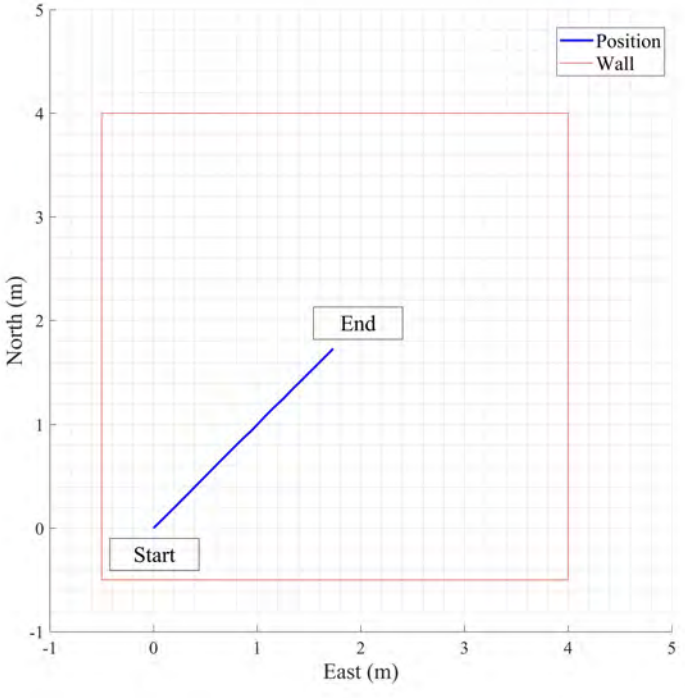


Figure 6.11: Obstacle avoidance demonstration - Position plot

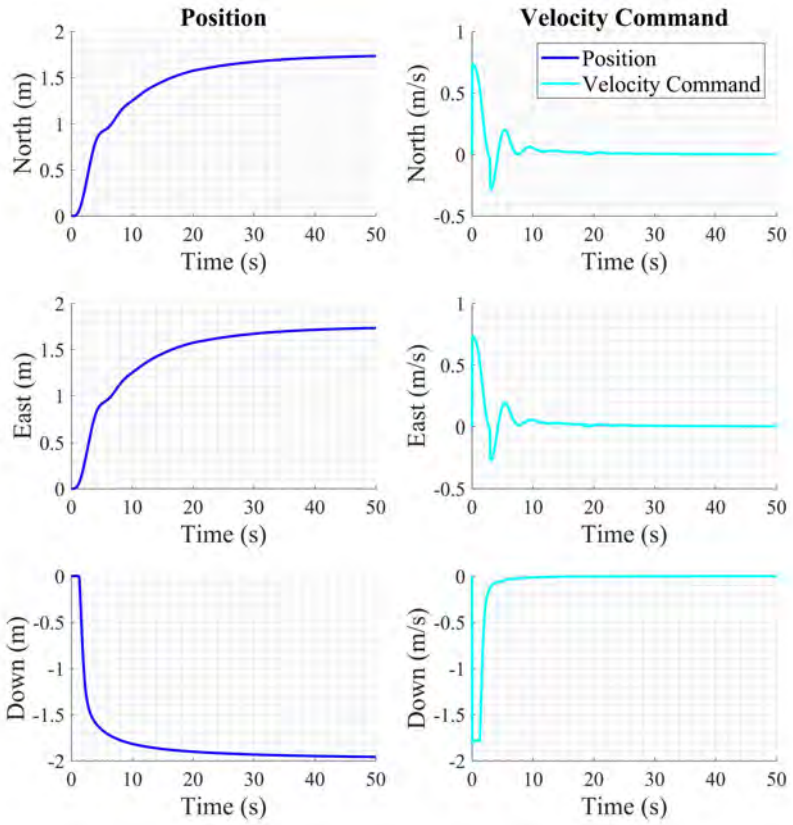


Figure 6.12: Obstacle avoidance demonstration - Position and velocity command plots

To demonstrate the interaction of the obstacle avoidance routine with the position controller a wall is set up at 10 m and the vehicle is commanded to go to 17 m. This test will ensure that the obstacle avoidance controller works when the vehicle is travelling at maximum speed. Figure 6.13 shows the North position of the vehicle in blue, the commanded position in red and the dotted line represents the position of the wall. As seen, the vehicle stops within 0.5 m of the wall.

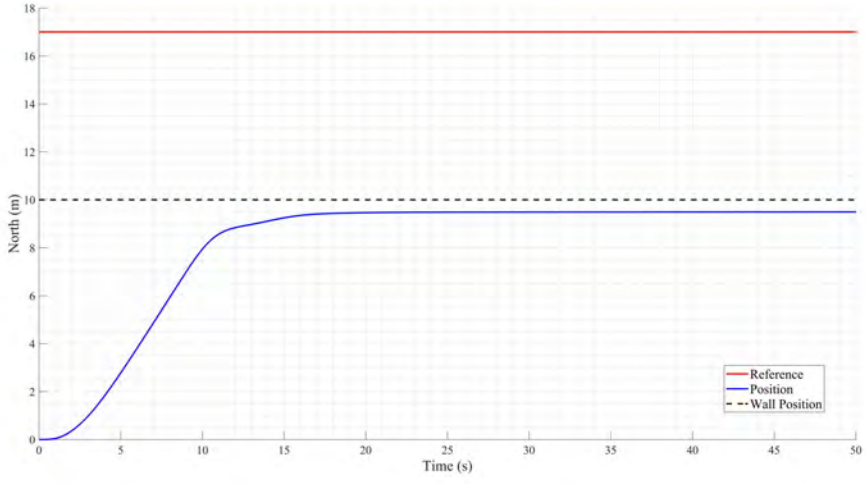


Figure 6.13: Obstacle avoidance - Straight wall

Figure 6.14 demonstrates the obstacle avoidance capabilities with a random obstacle in the drone's flight path. As seen, the vehicle will deviate off the path to avoid the object and detour around it. The purple lines represent the obstacle avoidance vector, which are interpreted as velocity commands. It will be noted that at some point the obstacle avoidance vector seems to be pulling the vehicle towards the obstacle. This occurs due to the vehicle having a large velocity away from the obstacle and the derivative component of the controller reducing the overshoot away from the obstacle.

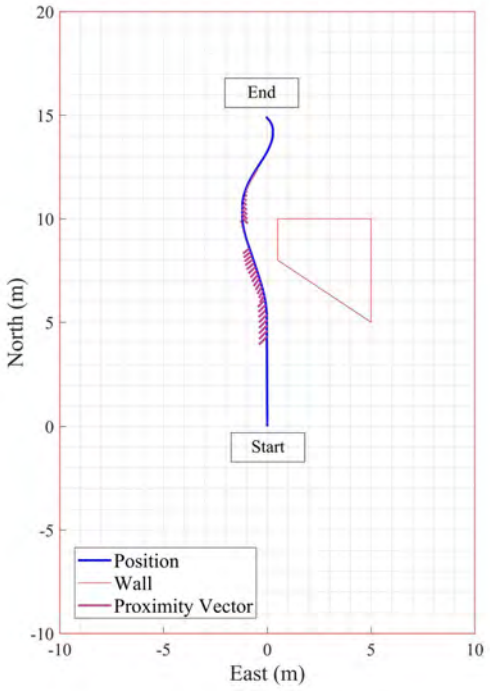


Figure 6.14: Obstacle avoidance - Random object in flight path. Purple line represents the obstacle avoidance vector

Although these examples show the method as an effective means of obstacle avoidance, this method has limitations that are explored in the following flight tests chapter.

Chapter 7

Simulated Flight Tests

This chapter presents the simulation tests that were performed to verify the functioning and performance of the waypoint navigation and collision avoidance algorithm. A simulation setup was created using the quadrotor flight dynamics model from chapter 4, the flight controllers from chapter 5 and the waypoint navigation and collision avoidance system designed in chapter 6. The quadrotor flight dynamics use the realistic physical parameters identified from the real quadrotor vehicle that were identified in chapter 4. The indoor environment for the quadrotor is modelled using flat planes and straight line segments to represent the floor, roof, walls and obstacles.

The first part of the chapter lays out the test objectives and cases. The second part of the chapter provides an overview of the simulation setup that was used for the simulated flight tests. The third part of the chapter describes the simulated flight tests that were performed, including the test objective, the test procedure, and the test results for each test.

The simulated flight tests show that the waypoint navigation and collision avoidance system successfully navigates a number of example indoor environments, and also demonstrates the limitations of the system.

7.1 Test Objectives and Test Cases

The objective of this work was to design an aerial vehicle that is capable of flight inside a confined space and narrow corridor. The first goal is to create a vehicle that continues to function in the presence of system and sensor noise. The vehicle must be shown to perform this task in the presence of disturbances.

Once the vehicle is shown to reject of disturbances using the designed flight controllers, the obstacle avoidance system needs to be tested. The first test must prove that in a simple environment, with disturbances the system ensures there is no collision. The next test must be to evaluate the drone navigating in a simple environment with the assistance of the obstacle avoidance controller. The difference with these tests will be to prove that the obstacle avoidance routine will navigate around obstacles and not simply just avoid them. After the drone can successfully navigate a simple environment, a more complex environment needs to be tested creating the need for more complex manoeuvres. Finally the limitations in the proposed method need to be shown, an environment where the vehicle will not be able to complete its desired mission must be shown to understand where improvements and future work can be aimed.

The aims of the each test is listed below for ease of reference. The independent tests need to be designed to show the listed outcomes.

1. General waypoint flight in the presence of disturbances
2. Obstacle avoidance routine in the presence of a disturbance
3. Navigation in a basic environment using the obstacle avoidance routine both with and without yaw alignment

4. Navigation in a complex environment using the obstacle avoidance routine both with and without yaw alignment

5. Limitations of the proposed flight routine

7.2 Simulation Setup

The testing and validation is done using a model generated in Simulink. Figure 7.1 gives a high level snapshot of the all the components in the final simulation. The user can input commands into the waypoint stepper which creates commands for the tracking control system. The tracking control system comprises of all the controllers designed in Chapter 5. The obstacle avoidance routine and the heading alignment strategy are discussed in Chapter 6 and feed into the tracking control block. The desired moments and forces generated by the tracking control are converted to thrust commands and fed into the plant model along with the modelled disturbances. The absolute states calculated by the vehicle flight dynamics are then fed through a sensor block to add noise and create values expected in a real system. The sensor values were chosen by extracting values from the sensor datasheets. These values included bandwidth limitations, sensor offsets and accuracies. The simulated sensors include an IMU, a GPS and the proximity sensors.

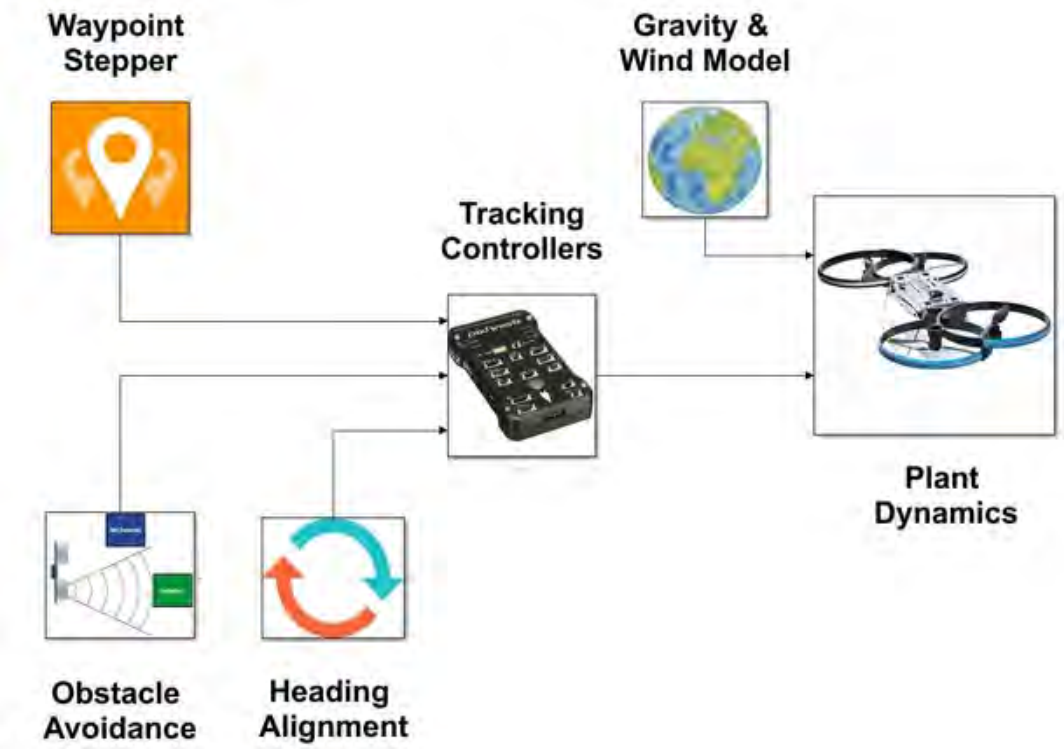


Figure 7.1: Snapshot of simulation in the Simulink environment

7.2.1 Indoor Environment Model

This section describes the modelling of the roof, floor, walls and obstacles used to verify the obstacle detection and avoidance algorithms. To simplify the implementation of the environment in simulation, the environment was modelled using horizontal and vertical planar surfaces to represent the floor, roof, walls and obstacles of an indoor environment. This choice also simplifies the calculation of the simulated proximity measurements. However, this does not mean that the waypoint navigation and obstacle avoidance system developed in chapter 6 is limited to a planar environment. The simple environment can be replaced with more complex environmental models in the future.

The floor will be modelled as a solid plane in the NE plane. The roof will be modelled similarly, however in certain scenarios the roof will be varied for different values of North and East, as shown in Figure 7.2.

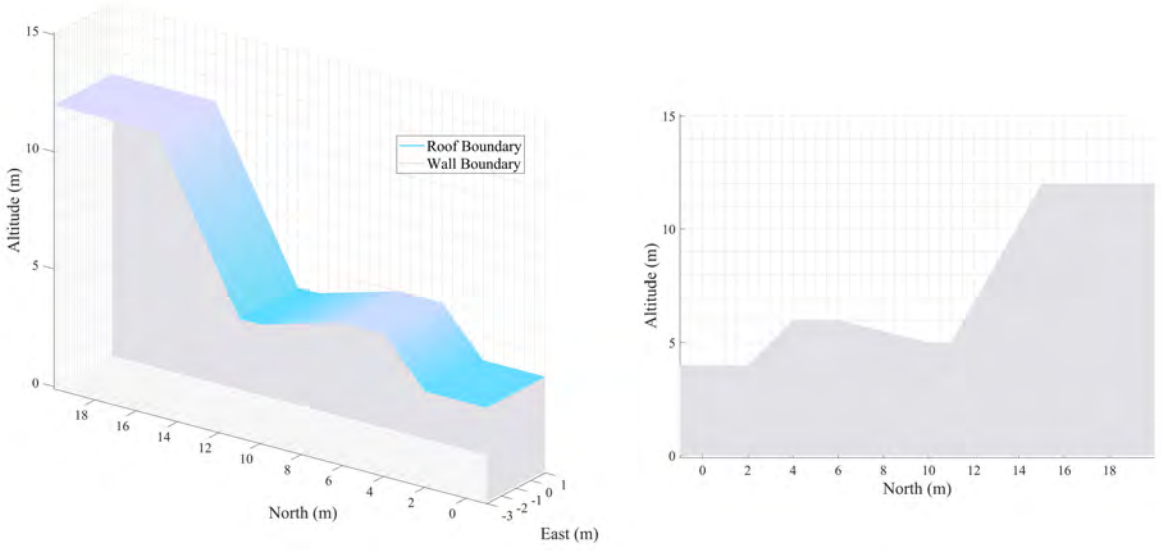


Figure 7.2: Plot representing the creation of the roof and a side wall

The walls and obstacles will all be modelled as sets of straight line segments which are the same value for all values of Z. Figure 7.3 shows a top view for the creation of a room with an obstacle present.

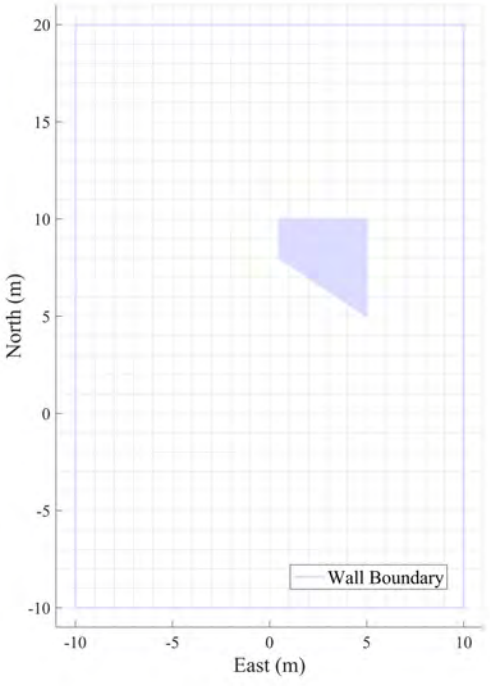


Figure 7.3: Plot representing the creation of a room containing an obstacle

7.2.2 Obstacle Distance Model

To simulate the proximity sensor measurements, the distance from the vehicle to the nearest obstacle in each sensor direction must be calculated for each sensor. The proximity sensor measurements are calculated slightly differently for horizontal proximity sensors and vertical proximity sensors.

For a horizontal proximity sensor, the horizontal obstacle distance is calculated by determining the intersection between the sensor beam line, and the line segment of the vertical plane that represents a wall or obstacle. The obstacle distance is the distance measured between the the vehicle position and the position of the sensor beam's intersection with the line segment of the wall or obstacle. Sensor noise is added to create the final proximity sensor measurement.

If the sensor beam intersects with multiple vertical obstacles, then the distance to the closest intersection is used. If the sensor beam does not intersect a vertical obstacle, or if the obstacle distance is beyond the maximum measurement range of the proximity sensor, then the maximum distance measurement is returned. Finally, if the incident angle of the sensor beam with the obstacle line segment is too large, then the intersection is ignored. This is to model the fact that proximity sensors do not receive a strong reflected signal when the incident angle of the sensor beam on the obstacle surface is too large, causing the sensor to not register the obstacle.

For the vertical proximity sensors, the obstacle distance is calculated as the difference between the vertical position of the vehicle and the vertical positions of the horizontal planes representing the floor or the ceiling. The proximity sensor measurement is calculated as the obstacle distance plus sensor noise.

The simulation will produce ten simulated proximity sensor measurements, one for each of the eight horizontal proximity sensors, and one for each of the two vertical proximity sensors. The simulated proximity sensor measurements are provided to the Waypoint Navigation and Obstacle Avoidance System, which will then use them to perform obstacle avoidance.

For the benefit of the reader, the equation used to calculate the intersection of two line segments is described below. A line segment is considered as a finite portion along a line, dictated by two points existing on that line. The equations of the line segments can be represented mathematically as shown in (7.2.1) - (7.2.2). p_1 and p_2 are the end and start points of the first line segment with p_3 and p_4 belonging to the second line segment. The equations of each line segment hold true for the case of $0 \leq t \leq 1$ expressing a position along the defined segment.

$$p_1 = t(p_2 - p_1) \tag{7.2.1}$$
$$p_3 = t(p_4 - p_3) \tag{7.2.2}$$

The definition of an intersection is where these two lines are equal, breaking the equation into x and y components then creates equations (7.2.3) - (7.2.4). Where t_a and t_b are the offsets for each respective line segment.

$$x_1 + t_a(x_2 - x_1) = x_3 + t_b(x_4 - x_3) \tag{7.2.3}$$
$$y_1 + t_a(y_2 - y_1) = y_3 + t_b(y_4 - y_3)) \tag{7.2.4}$$

The final step is to solve for both t_a and t_b. If both t_a and t_b exist between 0 and 1, then it is known that the line segments intersect. If either t_a or t_b exist outside of this range then the general lines intersect outside of the described segment constraints. If the lines are collinear, no intersection of the general lines, the result for t_a and t_b will be unsolvable. There might exist a possibility where the sensor line segments intersect with multiple walls, in this case the sensor would only return the closest obstacle. As mentioned, distance measurement devices struggle to return accurate results when the angle between the object and the sensor is very large. In this case the intersection is not recorded as to mimic a realistic sensor. In the case that no intersection is measured, the sensor would return its maximum range.

Figure 7.4 shows the typical result of a single obstacle detection frame. The red dots show the calculated intersection between the sensor beams and the environment. The corresponding distances are summed into a single proximity vector with the negative of that vector representing an obstacle avoidance vector shown in violet. The calculation of the proximity vector is not part of the environment model, but is a calculation performed by the onboard collision avoidance algorithm

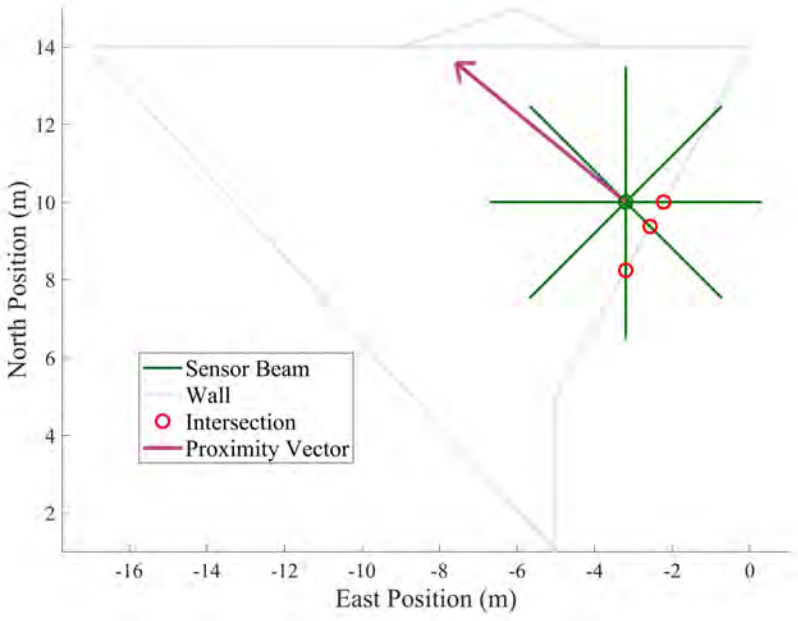


Figure 7.4: Obstacle detection demonstration

7.3 Simulated Flight Tests: Test Procedures and Simulation Results

This section describes the simulated flight tests in more detail. For each simulated flight test, the test objectives are stated, the test procedure is described, and the test results are presented and discussed. The testing is structured to follow a set of logical pattern of validations.

7.3.1 Flight Control and Waypoint Navigation With Wind Disturbance and No Obstacles

The purpose of this test is to ensure the correct operation and performance of the flight control and waypoint navigation in the presence of wind disturbances, but with no obstacles present. The results of this test will serve as the baseline for subsequent tests where obstacle avoidance is active.

First the vehicle is commanded to execute a position step in the presence of wind. Next, the vehicle is commanded to navigate a predefined set of waypoints to verify the guidance system in the presence of wind. The wind disturbance is chosen to be a $5 \pm 1\,\mathrm{m/s}$ wind blowing in a north-westerly direction, with a heading of $45° \pm 10°$off of the North axis.

To pass the first part of the test, the vehicle must perform the horizontal position steps exhibiting a transient response and steady state tracking error that meets the design specifications. The vehicle must also reject the external wind disturbance. To pass the second part of the test, the vehicle must navigate the waypoints with minimal cross-track deviation from the given path, and must also reject the external wind disturbances.

Figure 7.5 shows the response for a 3 m North and East position step. Note that the North position step is in the same direction as the North component of the wind, while the East position step is opposing the direction of the East component of the wind.

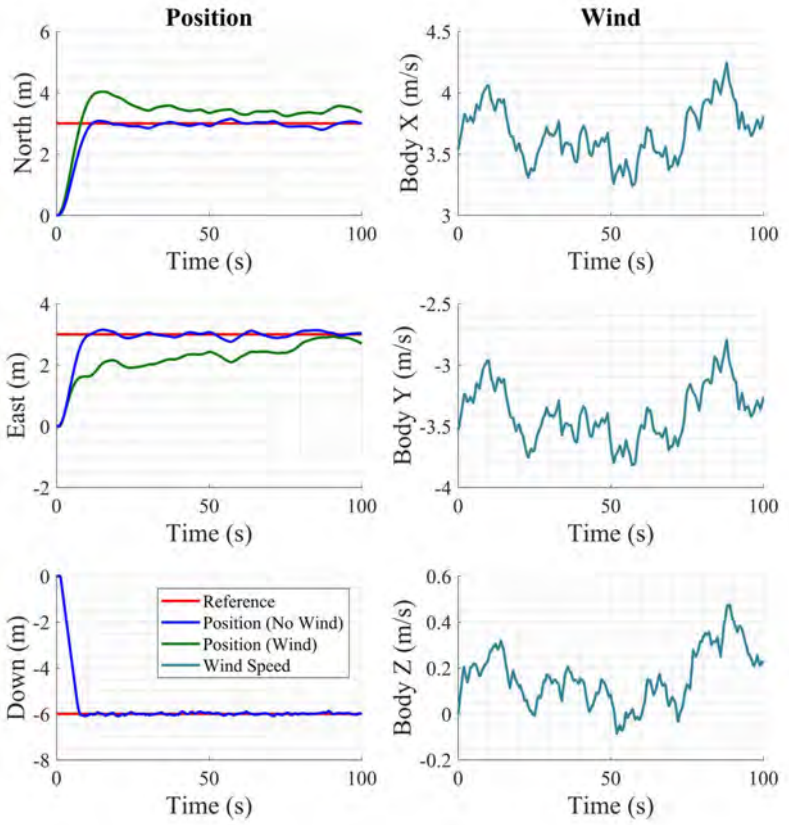


Figure 7.5: Step response with and without disturbance - North position plot

The North position step response exhibits an increased overshoot due to the external wind disturbance, that is in the same direction as the position step. The disturbed response has a rise time of 7.2 s for the 3 m step, while the undisturbed system has a rise time of 9.6 s. The East position step response exhibits a slight undershoot and a slower rise time due to the wind disturbance that opposes the direction of the position step. The undisturbed system exhibits a rise time of 8.3 s where the disturbed system has a steady state offset of 0.5 m.

The disturbed responses exhibits a non-zero steady state error, due to the use of a lag controller and not a pure integrator in the velocity controller.

Next, the waypoint generator is loaded with the same waypoints seen in Chapter 6. The large wind is angled to push the drone North and West and is present from the beginning of the simulation. Figure 7.6 shows both the isometric and the top view of the flight. The undisturbed flight is plotted alongside the disturbed flight for comparison.

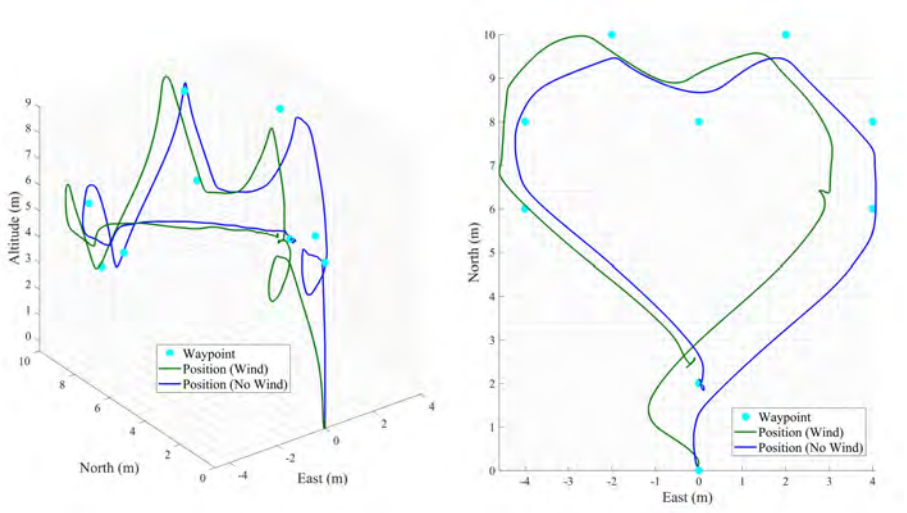


Figure 7.6: Waypoint flight with disturbance - Isometric and top position plot

Figure 7.7 shows the North, East, Down responses for the disturbed flight. The East position is initially offset by the wind and is commanded to fly against the wind before returning with the wind pushing it away from the setpoint. The North position reaches within 0.5 m in both cases. The seemingly linear portion of the curve entails that the disturbance is causing the drone to hit the upper saturation limit of the velocity command.

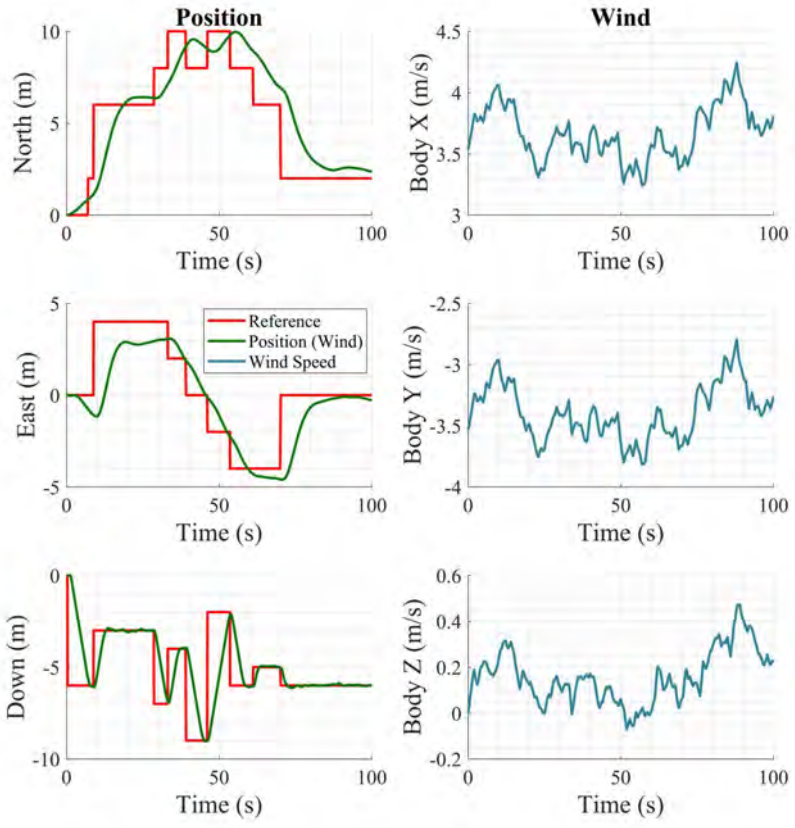


Figure 7.7: Waypoint flight with disturbance - North, East and Down position plot

The East position has more difficulty handling the disturbance. The non symmetry of the vehicle causes a larger surface area in that plane, which leads to a larger drag caused by wind disturbances. The East position is initially pushed 1 m off of its setpoint and settles to just less than a metre off the final desired position. The test results show that the flight controllers are able to control the vehicle to follow position steps, and the guidance system is able to control the vehicle to successfully navigate a given set of waypoints, even in the presence of wind disturbances.

7.3.2 Simple Obstacle Avoidance With Wind Disturbance

The next test is designed to evaluate the effectiveness of the obstacle avoidance routine in the presence of a disturbance. The test is designed to see if the vehicle can maintain a set distance away from a wall while the tracking system and a disturbance push it towards the wall. The vehicle is placed in a straight tunnel with a varying roof height, and is then given a combined horizontal and vertical position step command, that would normally cause the vehicle to collide with one of the tunnel walls, as well as with the tunnel ceiling. A constant 5 m/s wind was applied to the vehicle facing due East, pushing the vehicle into the wall. The cyan dot in the image represents the waypoint at (18, 0, -10).

To pass the test, the vehicle must perform the combined horizontal and vertical position step while the obstacle avoidance function simultaneously prevents the vehicle from colliding with either the side walls or the ceiling of the tunnel.

There are two images in Figure 7.8, the image on the left shows the view from the right, while the image on the right adds the top view. Both of these figures overlay the obstacle avoidance vector which is displayed in a violet coloured line.

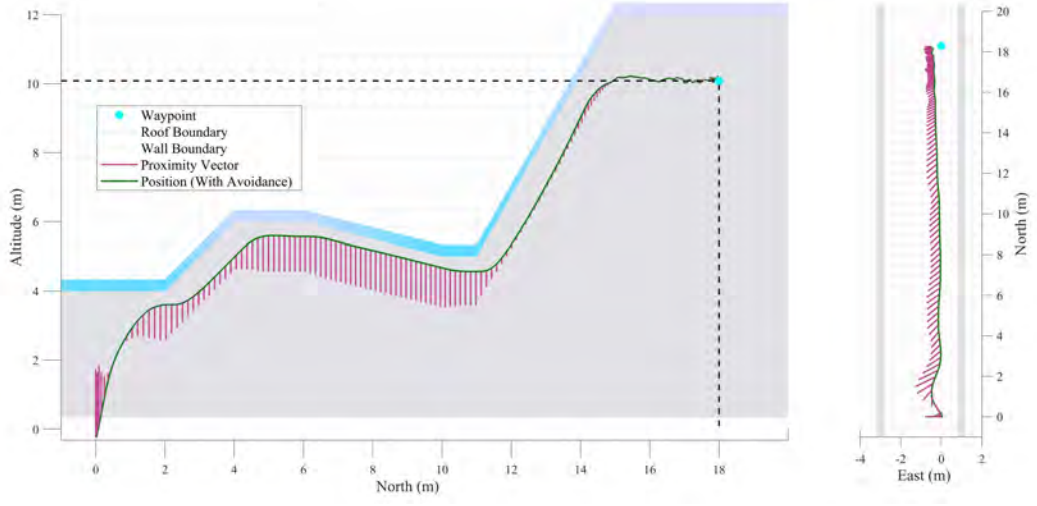


Figure 7.8: Right and top view of corridor flight with wind disturbance and overlayed obstacle avoidance vector

The drone is commanded to an East position of 0 m while the wind is attempting to force the drone directly East as well. The obstacle avoidance vector successfully steers the vehicle away from the East wall and maintains an average distance of 1.5 m off of the wall with a standard deviation of 0.15 m. The obstacle avoidance vector shown in the violet line of the right image shows the direction in which the drone is being pushed by the obstacle avoidance controller. The proximity to the East wall pushes the drone in a Westward direction, with a South facing component. The derivative portion of the controller creates the South facing component of the vector as it moves in a Northerly direction.

The figures shown in 7.9 show the same flight with each position plotted against time. The green line shows the path flown in a scenario where the obstacle avoidance controller is activated. The

red line is the set point with the blue line being the flown path with obstacle avoidance deactivated. The grey line in the East plot represent the wall at 1 m.

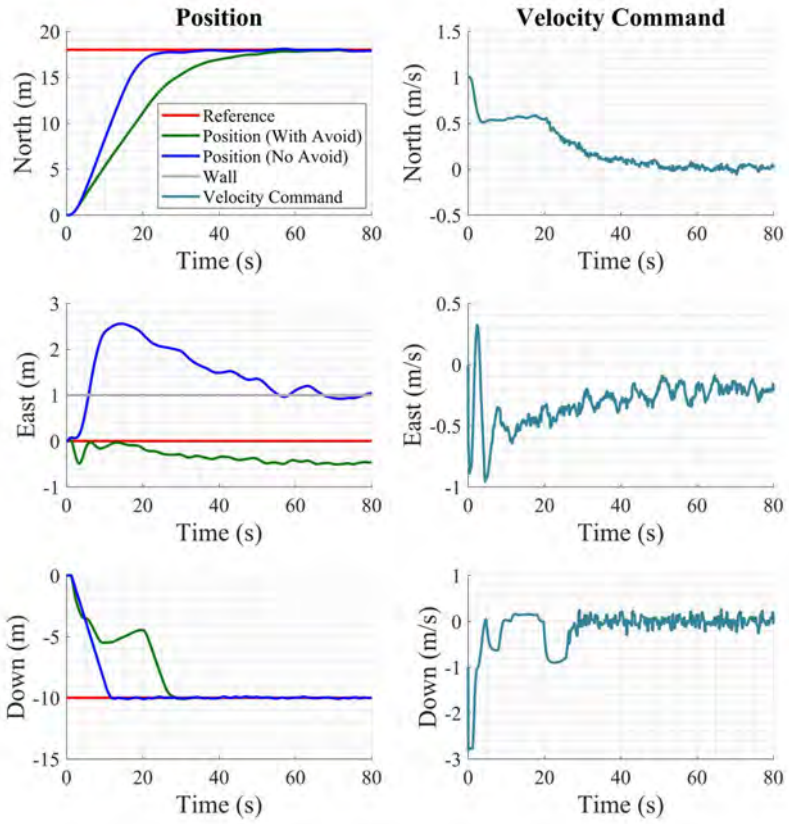


Figure 7.9: Corridor flight with disturbance - North, East, Down positions and obstacle avoidance velocity commands

The obstacle avoidance controller slows down the North settling time of the vehicle, this is due to the derivative component of the obstacle avoidance controller which opposes the direction of travel.

As expected from the previous test, without the obstacle avoidance controller the drone will collide with the wall at 1 m. The obstacle avoidance controller pushes the vehicle off of its setpoint to ensure it maintains a set distance from the wall.

The horizontal obstacle avoidance has been the main focus this far. In this scenario a varying roof was placed at positions between 4 m and 12 m with a floor at a constant 0 m. The setpoint of -10 m is above the roof until late in the flight. The vehicle maintains a minimum distance of 0.5 m from the roof.

The obstacle avoidance system has shown its effectiveness by avoiding a collision in the presence of a disturbance. The routine successfully kept the drone away from the walls by maintaining a set distance. The routine also ensures that the vehicle flies slowly and steady when in the presence of obstacles.

7.3.3 Waypoint Navigation and Obstacle Avoidance in a Basic Terrain

The purpose of the next series of tests is to verify the correct operation and performance of the system when waypoint navigation and obstacle avoidance are integrated. A basic environment is

used to remove anomalous behaviour due to specific properties of the environment. Simulations are performed both with and without yaw alignment to investigate the effect of the yaw alignment on the waypoint navigation and obstacle avoidance algorithms. The tests are performed using two different environments, namely a wide corridor and a narrow corridor, both of which are configured in a sawtooth shape. A series of three tests are performed. For all three tests the drone starts at the South end of the corridor and is given a destination waypoint at the North end of the corridor

The first test uses the wide corridor environment, where the obstacle avoidance function will only activate part of the time because the corridor is wider than the maximum distance of the sensor beam. The second test uses the narrow corridor environment, where the obstacle avoidance routine will be activated constantly, in multiple directions. The third test investigates the effect of the yaw alignment on the waypoint navigation and obstacle avoidance routines. To pass this set of tests, the vehicle must navigate the corridor with minimal cross-track deviation from the waypoint path while the obstacle avoidance function simultaneously prevents the vehicle from colliding with either the side walls or the ceiling of the tunnel.

7.3.3.1 Wide Corridor

The simulation results for the first test using the wide corridor and no yaw alignment are shown in Figure 7.10. The walls of the corridor are represented by grey lines. The vehicle starts at the South end of the corridor and is given a destination waypoint at the North end of the corridor (34 m North). The flown path is shown in green with the obstacle avoidance vector shown with the violet lines. The obstacle avoidance vector is interpreted by the vehicle as a velocity command.

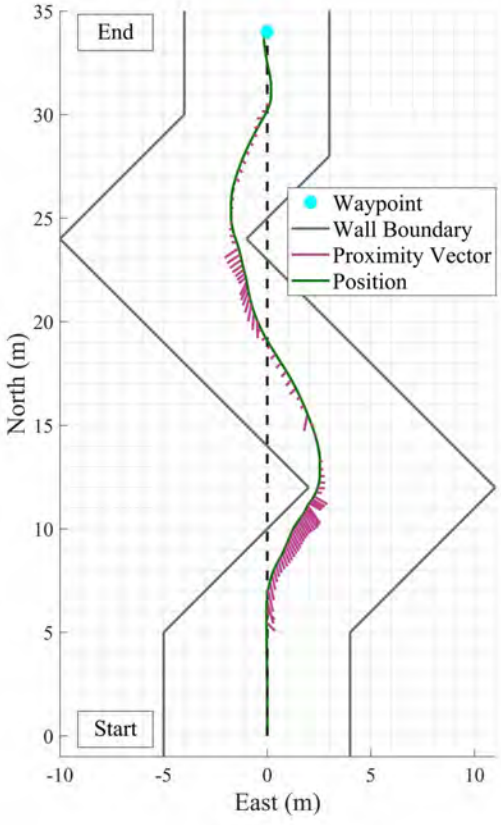


Figure 7.10: Navigated flight in a wide corridor, showing avoidance vector

The vehicle navigates the corridor to reach the goal with minimal cross-track deviation from the waypoint path, while maintaining a distance of 0.5 m away from the walls. The obstacle avoidance is only active when in close proximity to the walls. Although the vehicle successfully navigated the given path, the simulation results do show the importance of the density for sensor placement. The sharp corners created by the wall only activate one of the sensors, which creates a relatively small obstacle avoidance vector. Having a more dense sensor placement would both increase scanned area and create a larger obstacle avoidance vector, which would keep the vehicle further away from the corners.

7.3.3.2 Narrow Corridor

The simulation results for the second test using the narrow corridor and no yaw alignment are shown in Figure 7.11. Once again, the vehicle starts at the South end of the corridor and is given a destination waypoint at the North end of the corridor (17 m). The test illustrates the vehicle's ability to fly in a confined space where the obstacle avoidance function is always activated from multiple sides. The colour coding follows the same convention as the first test.

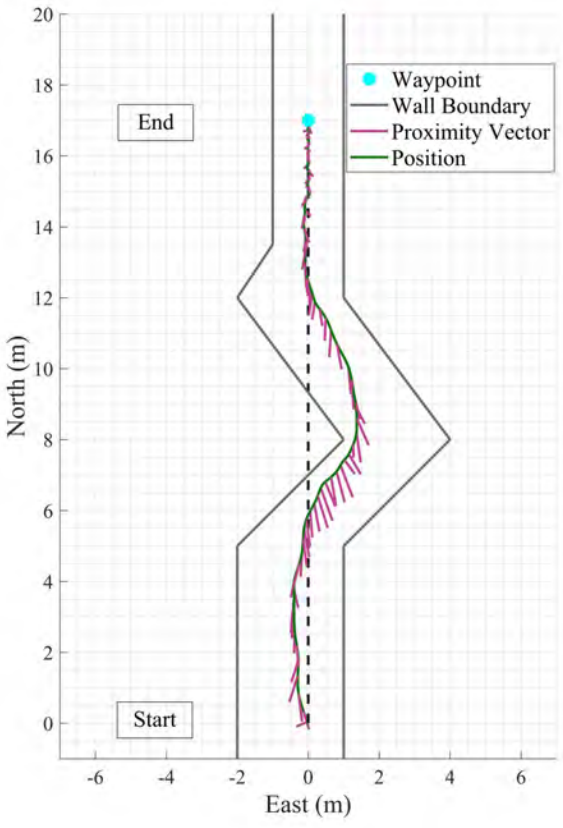


Figure 7.11: Navigated flight in a narrow corridor, showing avoidance vector

The vehicle navigates the corridor to reach the goal with minimal cross-track deviation from the waypoint path, while maintaining a minimum distance of 0.5 m from the walls. The test demonstrates the importance of the derivative action of the obstacle avoidance vector, specifically in a confined space. Although all the sensors are active due to close proximity to multiple walls, the walls which the drone is approaching takes precedence.

7.3.3.3 Yaw Alignment in a Confined Space

The simulation results for the third test using a wide corridor and integrating the yaw alignment routine is shown in Figure 7.12. The vehicle starts at the South end of the corridor and is given a destination at the North end of the corridor (34 m). In this test the vehicle is also expected to maintain a heading that aligns with the direction in which the vehicle is flying. The test illustrates the vehicle's ability to maintain a set heading without interfering with the vehicle's waypoint and obstacle avoidance functions. The red line is used to indicate the vehicle's position and the orange line is used to indicate the current heading of the vehicle.

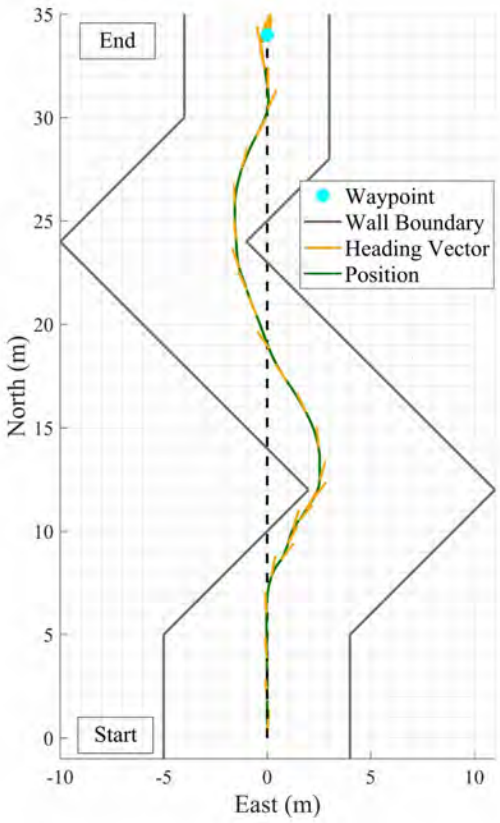


Figure 7.12: Navigated flight in wide corridor with yaw alignment, showing current heading of vehicle

The yaw controller has a lower bandwidth than the horizontal velocity controller, which results in a transient yaw alignment error when there is a sudden change in velocity heading. The yaw angle successfully tracks the velocity heading, but with some lag when the velocity heading is varying. When the velocity heading is constant the yaw angle tracks the velocity heading with zero steady-state error.

The average yaw alignment error for the flight is only 2° with a standard deviation of 21.5°. The heading controller therefore successfully keeps the nose of the vehicle pointed generally in the direction of the vehicle's direction of travel.

Figure 7.13 shows the simulation results for the waypoint navigation with obstacle avoidance in the wide corridor, both with and without yaw alignment.

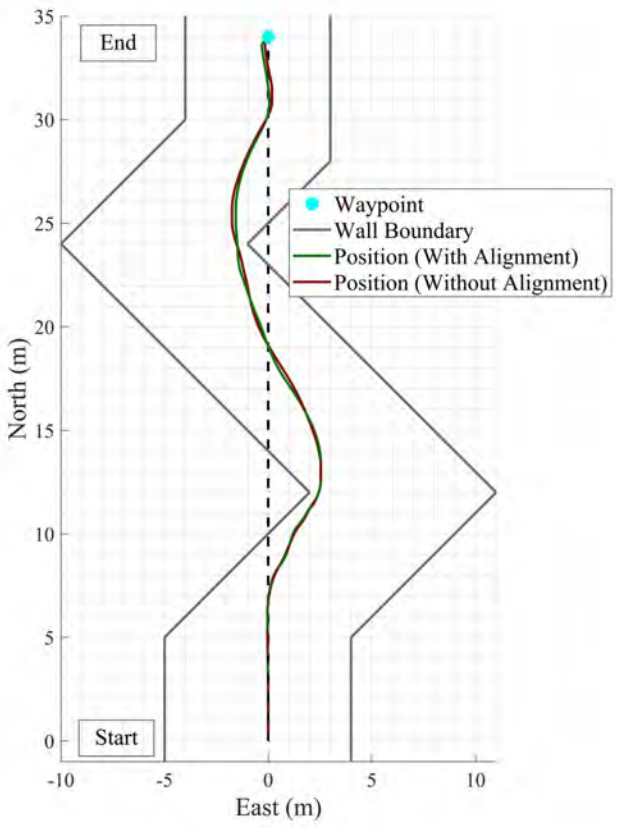


Figure 7.13: Navigated flight in wide corridor with and without yaw alignment

The executed flight trajectory with yaw alignment is very similar to the executed flight trajectory without yaw alignment. The simulation results verify that the yaw alignment action has a minimal adverse effect on the waypoint navigation and obstacle avoidance.

7.3.4 Waypoint Navigation and Obstacle Avoidance in a More Complex Terrain

The purpose of this test is to verify the correct operation and performance of the waypoint navigation with obstacle avoidance in a more complex environment, where the vehicle is given a sequence of waypoints to navigate, and where the environment contains a combination of open spaces, narrow corridors and unexpected obstacles. The test is again performed both with and without yaw alignment control. For the test, the waypoints are placed to make the vehicle explore the environment, but with the need to navigate narrow corridors and avoid obstacles unassisted. This simulates the practical scenario where the user gives the vehicle a sequence of waypoints that are chosen using incomplete knowledge of the environment map, and the vehicle must navigate the environment while mapping it out in more detail. The test environment and the sequence of waypoints are shown in Figure 7.14.

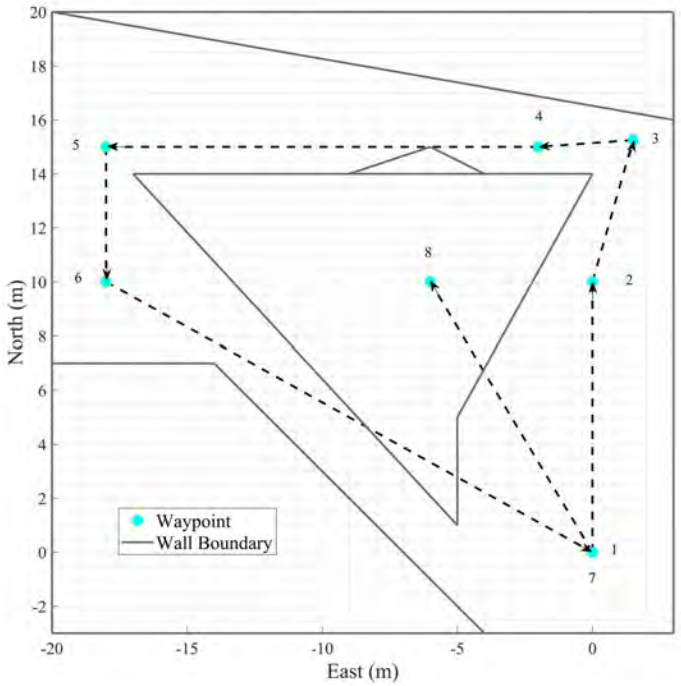


Figure 7.14: Layout of environment and waypoints.

The vehicle starts in the Southeast corner of the environment, and is given a sequence of waypoints to navigate, which are numbered from 1 to 8. After reaching the desired flight altitude, the vehicle must travel Northward through the open space (waypoint 1 to 2), then around a sharp corner through a narrow space (waypoints 2, 3, and 4), then westward along a wall (waypoint 4 to 5) where it will encounter an unexpected protrusion halfway, then southward around another corner (waypoint 5 to 6), and then diagonally through a long, narrow corridor in a Southeastern direction (waypoint 6 to 7). Finally, it is given an impossible waypoint which is deep inside an obstacle (waypoint 8). Also notice that the vehicle cannot fly in a straight line from waypoint 6 to waypoint 7, but will have to navigate the narrow corridor while avoiding the side walls.

The simulation results for waypoint navigation and obstacle avoidance with yaw alignment active is shown in Figures 7.15 and 7.16. Figure 7.15 shows the executed flight trajectory with the obstacle avoidance vector indicated, while Figure 7.16 shows the same executed flight trajectory but with the vehicle heading alignment vector indicated.

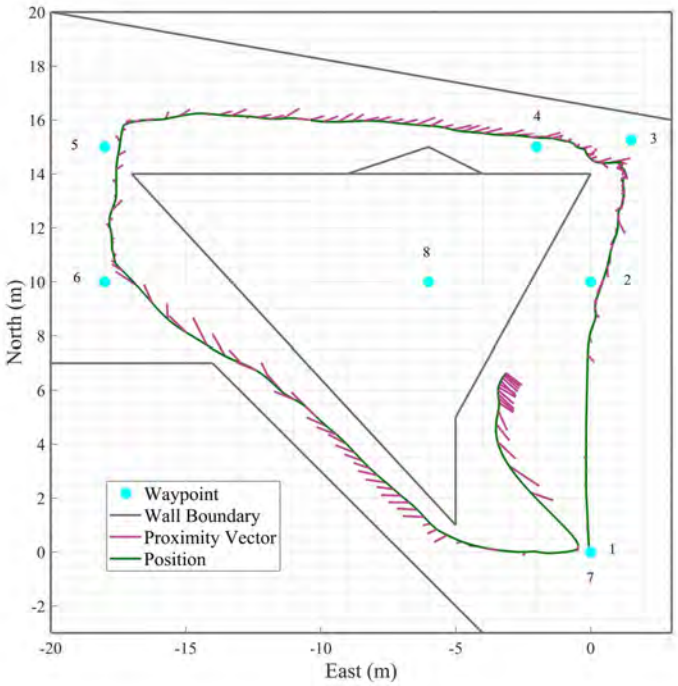


Figure 7.15: Yaw alignment plot of a generic flight test while utilising the heading alignment controller, showing avoidance vector.

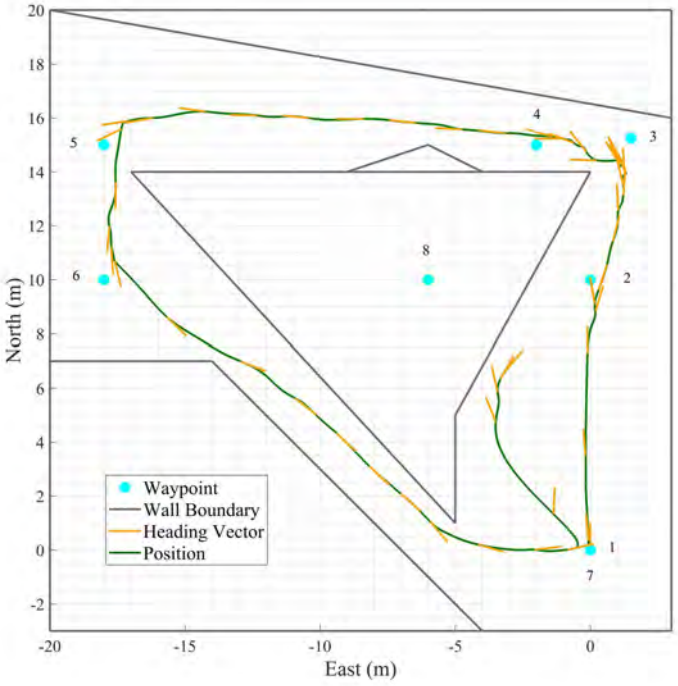


Figure 7.16: Yaw alignment plot of a generic flight test while utilising the heading alignment controller, showing current heading of vehicle.

The simulation results show that the vehicle successfully navigates the given waypoints while simultaneously avoiding the unexpected obstacles. The obstacle avoidance is not active while the vehicle traverses the open space between waypoint 1 and 2. As the vehicle travels around the first sharp corner and through the narrow space (waypoints 2, 3, and 4), the obstacle avoidance activates to avoid the sharp corner. As the vehicle travels westward along the wall (waypoint 4 to 5), the obstacle avoidance activates to avoid the unexpected protrusion from the wall. As the vehicle travels between waypoint 6 and 7, the obstacle avoidance compensates for the fact that the vehicle cannot fly in a straight line, and activates to guide the vehicle along the long, narrow, diagonal corridor while avoiding the side walls. Finally, the obstacle avoidance activates when the vehicle is given a waypoint which is inside an obstacle (waypoint 8) and maintains a safe separation distance from the wall. The simulation results also show that the vehicle successfully aligns its yaw angle with the velocity heading while navigating the environment. The yaw angle successfully tracks the velocity heading, but with some lag when the velocity heading is varying. When the velocity heading remains constant, for example during long stretches of straight path, then the yaw angle controller has more time to settle, and tracks the velocity heading with zero steady-state error. For this simulation, the yaw angle tracks the velocity heading with an average error of 3.6° with a standard deviation of 5.5°.

The simulation results with and without yaw alignment are shown and compared in Figure 7.17. The green line shows the results without yaw alignment, while the red line shows the results with yaw alignment.

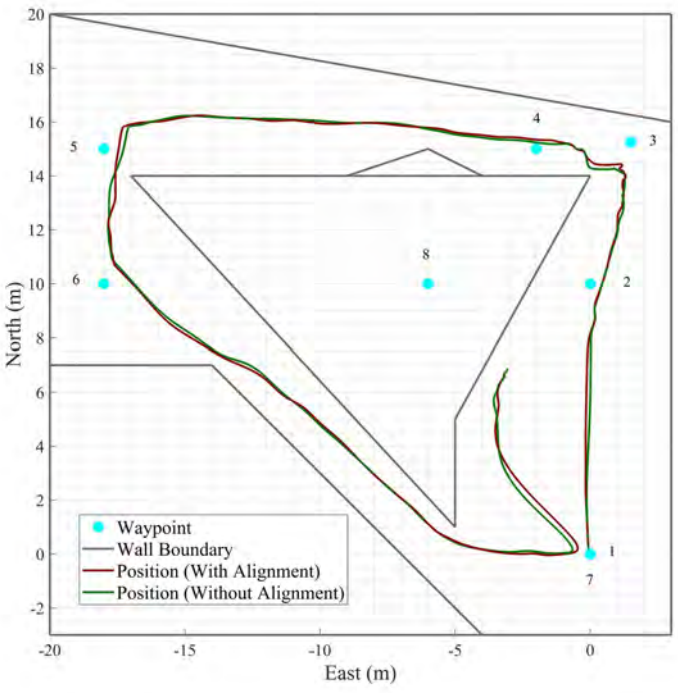


Figure 7.17: Plot of a generic flight test both with and without the yaw alignment active.

The results show that the yaw alignment has very little effect on the waypoint navigation and obstacle avoidance, with the largest trajectory deviation occurring when there are very sharp velocity heading changes, such as the sharp turn created between waypoints 6, 7, and 8.

7.3.5 Limitations

The purpose of the last set of tests is to illustrate some limitations of the waypoint navigation and obstacle avoidance system. The system is not a path planning algorithm, and only attempts

to follow a given sequence of waypoints while simultaneously avoiding obstacles. The system does not change or replan the given waypoints if they were poorly chosen or if the environment changes. If the waypoints are chosen poorly by the user, perhaps due to incomplete knowledge of the environment, then situations may arise where the vehicle will not be able to navigate the complete waypoint path, and may get stuck due to obstacle avoidance actions.

The first test illustrates that the vehicle will get stuck if it encounters a wide obstacle that crosses its intended path perpendicularly. The second test illustrates that the vehicle will get stuck if its intended path leads it into a narrowing corridor with open space leading to the corridor opening.

Figure 7.18 shows the simulation setup and the simulation results for the scenario where the vehicle encounters a wide obstacle across its intended path. The vehicle starts at a location in the southern end of the environment and is given a destination waypoint in the northern end. However, there is a wide obstacle oriented from west to east across its intended path. Since the obstacle has a finite width, the vehicle could theoretically just fly around it to reach the destination waypoint.

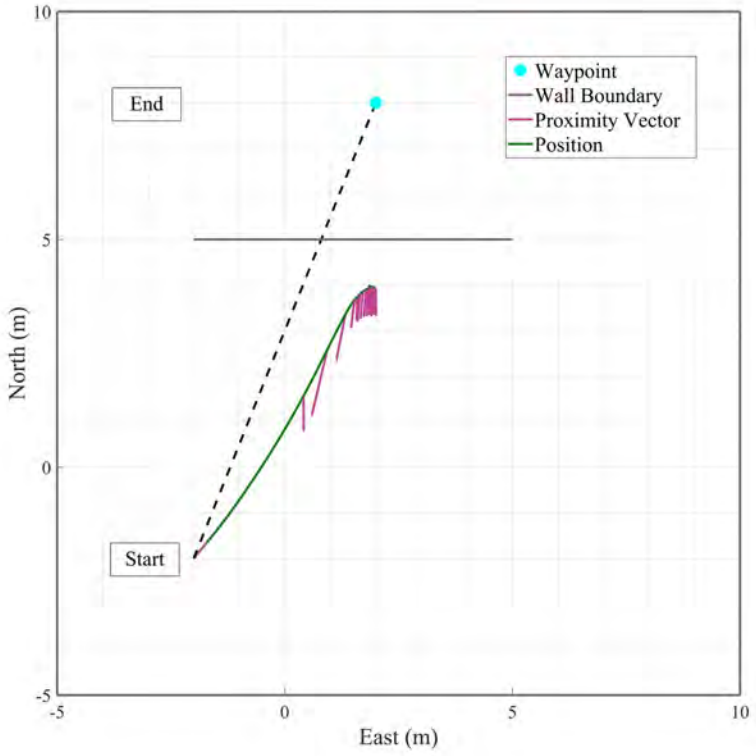


Figure 7.18: Limitations of the obstacle avoidance routine as a navigation algorithm. Straight wall in a wide space.

As the vehicle approaches the wall, it is detected by the proximity sensors and the obstacle avoidance routine is activated. The obstacle avoidance function commands an avoidance action southward, while the waypoint navigation commands an action northward to reach the waypoint. Due to the lower limit on the position controller, the vehicle stops at the location where the two opposing actions are equal, and the vehicle maintains a safe separation distance from the wall. Since the obstacle lies across the intended path perpendicularly, the proximity sensor measurements that are mounted diagonally forward and to the sides measure equal distances, and there is no resulting avoidance action either westwards or eastwards. Therefore the vehicle cannot move along the obstacle and gets stuck behind it. Although the vehicle successfully avoids the collision with the obstacle, it does not successfully complete the mission specified by the waypoints.

Figure 7.19 shows the simulation setup and simulation results where the vehicle's intended path leads into a narrowing corridor. The vehicle starts in a wide space in the southern end of the environment, and is given a waypoint inside a narrow corridor towards the north. The approach has two diagonal walls to either side of the vehicle, and the environment gradually narrows from the wide space to the narrow corridor.

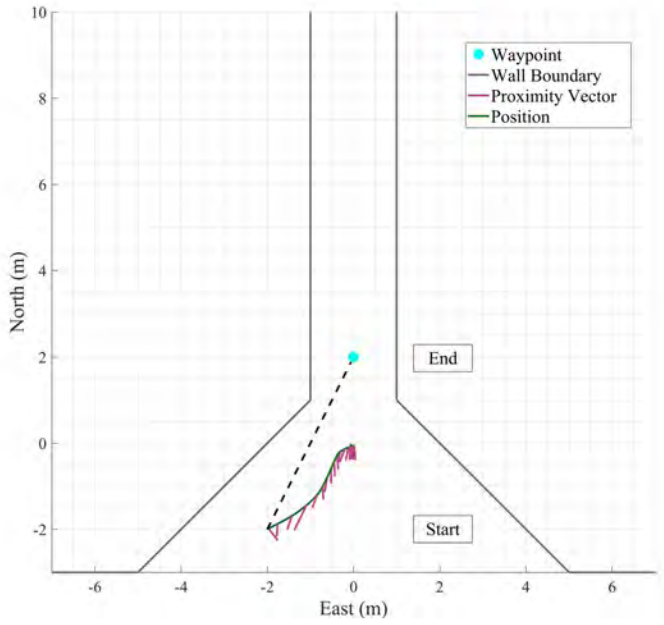


Figure 7.19: Limitations of the obstacle avoidance routine as a navigation algorithm. A wide open space leading into a narrow corridor.

As the vehicle approaches the narrow corridor, the proximity sensors that point diagonally forward and to the side both detect the diagonal walls. Note that the proximity sensor that points directly forward does not detect an obstacle. The westward and eastward components of the obstacle avoidance vector therefore cancel out, while the southward components add together. The obstacle avoidance function therefore commands a southward avoidance action, due to the diagonal walls to the side, even though there is no obstacle directly in front of the vehicle. Meanwhile, the waypoint navigation function commands a northward action to reach the waypoint. The vehicle therefore stops at the location where the two opposing actions are equal, and the vehicle comes to a halt in the funnel shape leading to the corridor. As seen, the vehicle will also tend to centre itself in the east-west direction. In this scenario there was no physical obstacle directly along the vehicle's path, but it was still unsuccessful in completing the mission specified by the waypoints.

Both of these scenarios illustrate the limitations of the waypoint navigation and obstacle avoidance system, and indicate that path planning and re-planning must be still be performed at a higher-level, either by a human operator or an autonomous path planning algorithm.

7.4 Simulated Flight Tests Discussion

A series of simulated flight tests were designed to assess the functioning of the controllers designed in chapter five and their robustness against disturbances. The tests were then designed to assess the functions designed in chapter six, namely the waypoint, yaw alignment and obstacle avoidance systems.

The vehicle was shown to be able to follow waypoints in the presence of a wind disturbance. Next, it was shown that the obstacle avoidance routine is capable of ensuring that there is no collision

while following a waypoint in the presence of a disturbance in a narrow space. The vertical and horizontal obstacle avoidance routines ensured that the vehicle did not hit the roof of varying height nor did it hit any of the side walls.

The next series of tests placed the simulated vehicle in both a wide and a narrow corridor. The corridors were both designed in a sawtooth shape. The vehicle successfully reached the waypoint in both cases, including the final test where the yaw alignment was activated. In the narrow corridor, the obstacle avoidance routine was active from multiple sides for the duration of the flight and proved the effectiveness and requirement for the derivative component of the obstacle avoidance controller. The yaw alignment strategy was shown to be effective while having negligible effect on the operation of the obstacle avoidance controller and the waypoint scheduler.

The final set of tests was designed to assess all of the vehicle's capabilities in a more complex environment. The test setup included narrow corridors, open spaces, unexpected obstacles and an unreachable waypoint. The vehicle successfully navigated the environment and avoided all collisions. Although the vehicle was unable to reach the final waypoint, the obstacle avoidance routine ensured a safe distance for the vehicle from all of the side walls.

To complete the testing, specific scenarios were designed to show the limitations of the design. Although the obstacle avoidance routine ensures the vehicle does not collide with any obstacles, situations exist where the routine will stop the vehicle from reaching a destination. A higher level input from a human operator or a more intelligent replanning algorithm will be required.

Chapter 8

Conclusions and Recommendations

The final chapter presents a summary of the work done in this project, discusses the main conclusions, and recommends future areas that should be focussed on to achieve the goal of complete autonomous flight in a confined environment.

8.1 Summary and Conclusions

In this project, a close quarters, collision-protected aerial vehicle, capable of obstacle avoidance and basic navigation inside a partially-known, confined environment, was successfully designed, implemented, and verified. The proposed platform, controllers and flight strategy were verified in simulation and were able to maintain a distance of 0.5 m away from walls and obstacles while maintaining stable flight.

The craft design of the vehicle was accomplished through analysis of conventional rotor wing configurations and flight theory. The vehicle required high payload capabilities for additional sensor packs and larger power sources to increase flight time. This was accomplished by choosing a design that optimised thrust capabilities in a narrow space. The design differs from a traditional quadcopter by having a 20% overlap of the front and rear rotor sets. The required avionics system architecture was designed by identifying the computing and sensing needs of the vehicle. Specific hardware was chosen to be as lightweight and high speed as possible.

The vehicle was modelled through a process of mathematical modelling and system identification. Real world measurement were taken from the vehicle and the onboard avionics as well as from manufacturer datasheets. A simulation model of the system and external disturbances was created in Matlab and Simulink.

Three separate controller subsystems were designed to control the proposed platform in six degrees of freedom. The three subsystems were broken up into an altitude, horizontal and heading controller. All three subsystems were shown to be capable of rejecting disturbances and providing stable control. The altitude control system controls the height of the craft by commanding a climb rate which in turn controls the acceleration of the craft in line with the body Z-Axis. The horizontal controller is responsible for controlling the North and East position and velocity of the craft. This was accomplished by relating the North and East accelerations to relative pitch and roll angles for the craft which in turn command the pitch and roll angular rates. The heading controller is responsible for controlling the yaw angle of the craft by commanding a yaw rate. Each controller fed their setpoints into a motor mixer which created the correct thrust outputs for each motor.

To enable autonomous flight, a waypoint navigator was created which enables the aircraft to automatically step between position set points, predefined by a user based on existing knowledge of the environment. The heading controller was used to keep the nose of the vehicle pointed generally in the direction of the vehicle's direction of travel. This ensured that the craft's longer axis is always in the direction of flight, minimising drag and proximity to narrow corridors.

An investigation into existing collision avoidance techniques led to the successful generation of a proximity-based obstacle avoidance routine. The method chosen requires a proximity measurement relative to the craft in the X, Y and Z body axes and utilises the potential field method of obstacle

avoidance. This allowed the craft to avoid obstacles by maintaining a set distance from obstructions in all three axes.

The simulation showed that the proposed flight strategy and controllers could be used for navigation in a partially-known, confined environment. The platform was designed to ensure sufficient thrust capabilities for a larger power source and additional sensor payload making it suitable for expansion into industrial applications.

8.2 Recommendations

The following recommendations are proposed to improve the viability of the system as an autonomous platform and expand the work to create a real world implementation.

- The proposed obstacle avoidance routine is not a path planning algorithm, and only attempts to follow a given sequence of waypoints while simultaneously avoiding obstacles. The obstacle avoidance does not change or replan the given waypoints if they were poorly chosen or if the environment changes. If the waypoints are chosen poorly by the user, perhaps due to incomplete knowledge of the environment, then situations may arise where the vehicle will not be able to navigate the complete waypoint path, and may get stuck due to obstacle avoidance actions. A higher-level path planning algorithm should therefore be developed to perform planning and re-planning to autonomously generate the set of mission waypoints for the waypoint navigator.
- The system was only verified in simulation on a high-fidelity simulation model of the vehicle, and was not verified with practical flight tests. To finalise the validation of the proposed system, real world flight tests should be conducted using the real platform. It is recommended that prior to any implementation of the flight strategy or obstacle avoidance, the mechanical construction should be verified to be robust to ensure a good flight setup, limiting risk during flight testing.
- Additional flight modes should be created to allow the pilot control of the inner loops. This work creates a waypoint generator that feeds velocity commands. Situations exist, specifically during initial testing, that require the pilot to command the inner-loop controllers.
- The literature review included an investigation into the aerodynamic ground effect and associated disturbances experienced by the vehicle when flying near walls. More detailed modelling and system identification experiments should be performed to characterise these disturbances, to ensure that the flight control system can successfully reject the disturbances.
- A robust state estimator would reduce noise and error on the measurements and allow for the implementation of a disturbance observer based control algorithm. Such an algorithm could assist with successful rejection of larger disturbances while limiting the effect on the tracking control.

Bibliography

[1] M. Raibert, K. Blankespoor, G. Nelson and R. Playter, "Bigdog, the rough-terrain quadruped robot", *The International Federation of Automatic Control*, vol. 17, pp. 6–9, 2008 (Cited on page 1).

[2] D. Robinson, H. Chung and K. Ryan, "Computational investigation of micro rotorcraft near-wall hovering aerodynamics", in *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, May 2014, pp. 1055–1063 (Cited on pages 1, 18–20, 22).

[3] A. Klaptocz, A. Briod, J.-c. Zufferey and D. Floreano, "An indoor flying platform with collision robustness and self-recovery", *2010 IEEE International Conference on Robotics and Automation*, pp. 3349–3354, 2010 (Cited on pages 1, 23).

[4] J. Leishman, *Principles of Helicopter Aerodynamics*, ser. Cambridge Aerospace Series. Cambridge University Press, 2002 (Cited on pages 5–7).

[5] F. A. Administration, *Rotorcraft Flying Handbook*, ser. FAA Handbooks Series. Aviation Supplies & Academics, Incorporated, 2001 (Cited on page 7).

[6] A. Steele and J. Treurnicht, "Review of standard rotor configurations for a micro unmanned aerial system", in *2015 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech)*, Nov. 2015, pp. 123–129 (Cited on page 8).

[7] Y. Cao, D. Li, Q. Zhang and H. Bian, "Recent development of rotorcraft configuration", *Recent Patents on Engineering*, vol. 1, no. 1, pp. 49–70, 2007 (Cited on pages 8, 9).

[8] F. Bohorquez, P. Samuel, J. Sirohi, D. Pines, L. Rudd and R. Perel, "Design, analysis and hover performance of a rotary wing micro air vehicle", *Journal of the American Helicopter Society*, vol. 48, no. 2, p. 80, 2003 (Cited on pages 8, 10).

[9] L. Young, E. Aiken, J. Johnson, J. Andrews, J. Klem and R. Demblewski, "New concepts and perspectives on micro-rotorcraft and small autonomous rotary-wing vehicles", *20th AIAA Applied Aerodynamics Conference*, vol. 2002-2816, p. 13, 2002 (Cited on pages 8, 15).

[10] F. A. Administration, "Helicopter components, sections, and systems", in *Helicopter Instructor's Handbook*, US Department of Transportation, 2012, ch. Chapter 5, p. 183 (Cited on page 9).

[11] A. H. Logan and R. E. Moore, "Us patent:4200252 - helicopter antitorque system using circulation control", 1980 (Cited on page 9).

[12] K. W. Flanigan, "Us patent:7147182 - gas powered tip-jet-driven tilt-rotor compound vtol aircraft", 2006 (Cited on page 10).

[13] A. Briod, A. Klaptocz, J. Zufferey and D. Floreano, "The airburr: A flying robot that can exploit collisions", in *2012 ICME International Conference on Complex Medical Engineering (CME)*, 2012, pp. 569–574 (Cited on pages 10, 23).

[14] W. Johnson, *Helicopter Theory*, ser. Dover Books on Aeronautical Engineering Series. Dover Publications, 1994 (Cited on page 10).

[15] P. Y. Oh, M. Joyce and J. Gallagher, "Designing an aerial robot for hover-and-stare surveillance", *IEEE Advanced Robotics*, pp. 303–308, 2005 (Cited on page 11).

[16] T. Luukkonen, "Modelling and control of quadcopter", *Journal of the American Society for Mass Spectrometry*, pp. 1134–45, 2011 (Cited on pages 11, 13, 15).

[17] P. Pounds, R. Mahony and P. Corke, "Modelling and control of a large quadrotor robot", *Control Engineering Practice*, vol. 18, pp. 691–699, 2010 (Cited on page 11).

[18] G. Hoffmann, H. Huang, S. Waslander and C. J Tomlin, "Quadrotor helicopter flight dynamics and control: Theory and experiment", *AIAA Guidance, Navigation and Control COnference Exhibit*, Aug. 2007 (Cited on pages 11, 18–21).

[19] P. Möller, "Automated landing of a quadrotor unmanned aerial vehicle on a translating platform", Master's thesis, Stellenbosch University, 2015 (Cited on pages 11, 13–17, 20, 21, 41, 43).

[20] K. Großekatthöfer and Z. Yoon, "Introduction into quaternions for spacecraft attitude representation", *TU Berlin*, pp. 1–16, 2012 (Cited on page 13).

[21] L. R. García Carrillo, A. E. Dzul López and C. Lozano Rogelioand Pégard, "Modeling the quad-rotor mini-rotorcraft", in *Quad Rotorcraft Control: Vision-Based Hovering and Navigation*. Springer London, 2013, pp. 23–34 (Cited on pages 15, 17).

[22] J. Blakelock, *Automatic Control of Aircraft and Missiles*, ser. A Wiley-Interscience publication. Wiley, 1991 (Cited on page 15).

[23] P. Garcia, R. Lozano and A. Dzul, *Modelling and control of mini-flying machines*, ser. Advances in Industrial Control. Springer London, 2005 (Cited on page 15).

[24] D. Lee, A. U. Awan, S. Kim and H. Jin Kim, "Adaptive control for a vtol uav operating near a wall", in *AIAA Guidance, Navigation, and Control Conference 2012*, Aug. 2012 (Cited on pages 18, 19, 22).

[25] D. Mellinger, N. Michael and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors", *International Journal of Robotic Research - IJRR*, vol. 31, Apr. 2012 (Cited on page 21).

[26] D. C. Robinson, "Modelling and estimation of aerodynamic disturbances acting on a hovering micro helicopter in close proximity to planar surfaces", PhD thesis, Monash, 2016 (Cited on page 22).

[27] A. Klaptocz, A. Briod, L. Daler, J.-c. Zufferey and D. Floreano, "Euler spring collision protection for flying robots", in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 1886–1892 (Cited on page 23).

[28] A. Briod, D. Floreano, P. Kornatowski and J.-C. Zuffery, "A collision-resilient flying robot", *Journal of Field Robotics*, vol. 31, no. 4, pp. 496–509, 2014 (Cited on page 23).

[29] J. Green, A. Marnewick and J. H. Pretorius, "Prototyping during the requirements elicitation process in the development of an underground unmanned aerial system", *Proceedings of the 2015 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference, PRASA-RobMech 2015*, pp. 60–65, 2015 (Cited on page 23).

[30] N. Gageik, P. Benz and S. Montenegro, "Obstacle detection and collision avoidance for a uav with complementary low-cost sensors", *IEEE Access*, vol. 3, pp. 599–609, 2015 (Cited on page 23).

[31] A. J. Barry, H. Oleynikova, D. Honegger, M. Pollefeys and R. Tedrake, "Fast onboard stereo vision for uavs", *IROS Workshop*, p. 7, 2015 (Cited on page 24).

[32] V. Lumelsky and T. Skewis, "Incorporating range sensing in the robot navigation function", *IEEE Transactions on Systems, Man and Cybernetics*, vol. 20, no. 5, pp. 1058–1069, 1990 (Cited on pages 24, 25).

[33] M. I. Ribeiro, "Obstacle avoidance", *Instituto de Sistemas e Robótica, Instituto Superio Técnico*, p. 1, 2005 (Cited on page 25).

[34] D. Delling, P. Sanders, D. Schultes and D. Wagner, "Engineering route planning algorithms", in *Algorithmics of large and complex networks. Lecture notes in computer science*, Springer, 2009 (Cited on page 26).

[35] W. Van den Aardweg, "Robust sampling-based conflict resolution for commercial aircraft in airport environments", Master's thesis, Stellenbosch University, 2015 (Cited on page 26).

[36] C. Beyers, "Motion planning algorithms for autonomous navigation for a rotary-wing uav", Master's thesis, Stellenbosch University, 2013 (Cited on page 26).

[37] P. E. Hart, N. J. Nilsson and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths", *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, 1968 (Cited on page 27).

[38] L. E. Kavraki, P. Svestka, J. Latombe and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces", *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996 (Cited on page 27).

[39] L. E. Kavraki, M. N. Kolountzakis and J. Latombe, "Analysis of probabilistic roadmaps for path planning", *IEEE Transactions on Robotics and Automation*, vol. 14, no. 1, pp. 166–171, 1998 (Cited on page 27).

[40] S. M. Lavalle, "Rapidly-exploring random trees: A new tool for path planning", May 1999 (Cited on page 27).

[41] M. Aeronautics, "Drone 3 hoverbike (http://www.hover-bike.com/ma/product/drone-3/)", *Mally Aeronautics Website*, 2015 (Cited on page 33).

[42] M. Cutler and J. How, "Design and control of an autonomous variable-pitch quadrotor helicopter", PhD thesis, Massachusetts Institute of Technology, 2012 (Cited on page 37).