Critical comparison of control techniques for a flight dynamics controller

A case study for implementation on a quad-rotor unmanned aerial vehicle.

Dissertation submitted for the degree

***Magister Ingeneriae* in Computer Engineering**

at the Potchefstroom campus of the North-West University

G. Otto

20120974

Supervisor: **Prof. J.E.W. Holm**

November 2011

Declaration

I, Gustav Otto, hereby declare that the dissertation entitled "**Critical comparison of control techniques for a flight dynamics controller**" is my own original work and has not already been submitted to any other university or institution for examination.



Gustav Otto

Student number: 20120974

Signed on the 18th day of November 2011 at Potchefstroom.

Acknowledgements

First and foremost I would like to thank my parents for their continued support in my studies. To my colleagues I owe great thanks for their assistance and support during this project. I would also like to thank Prof. J.E.W. Holm and Mr. H. Marais for their guidance in the research and management of this project.

Abstract

This dissertation covers the process of modelling and subsequently developing a flight dynamics controller for a quad-rotor unmanned aerial vehicle. It is a theoretical study that focusses on the selection of a controller type by first analysing the problem on a system level and then on a technical level. The craft is modelled using the Newton-Euler model, accounting for multiple reference frames to account for the interpretation of orientation as seen by on-board sensors. The quad-rotor model and selected controllers are characterized and compared. The model is verified through simulation by comparison to a validated model. A series of generic control loops are derived and used as reference for the implementation of the controllers. A Simulator is developed and used to do a comparative study of the various controller types and the control approach. Finally a full simulation is done to demonstrate the interaction between the controllers.

Keywords: *UAV, Quad-rotor, control, comparison, Newton-Euler model*

Samevatting

Hierdie verhandeling behels die proses van modellering en daaropvolgende ontwikkeling van 'n vliegdinamika beheerder vir 'n Quad-rotor helikopter onbemande vliegtuig. Dit is 'n teoretiese studie wat fokus op die seleksie proses om 'n beheertegniek te kies deur eers op 'n stelsel vlak na die probleem te kyk en dan op 'n tegniese vlak. Die vaartuig word gemodelleer met behulp van die Netwon-Euler model. Die model neem in ag dat daar meer as een verwysingsraamwerk is, om sodoende in ag te neem die interpretasie van die oriëntasie vir sensors op die tuig. Die model en geselekteerde beheerders word gekarakteriseer en vergelyk. Die model word geverifieer met simulasies deur dit te vergelyk met 'n bestaande model wat reeds geverifieer is. 'n Reeks generiese beheerlusse word afgelei en as verwysing gebruik om die beheerders te implementeer. 'n Simulator word ontwikkel en gebruik om die vergelykende studie te doen van die verskillende beheer tegnieke. Uiteindelik word n volledige simulasie gedoen om die samewerking van die beheerders te demonstreer.

Sleutelwoorde: *UAV, Quad-rotor, beheer, vergelyk, Newton-Euler model*

Contents

List of Figures **x**

List of Tables **xiv**

List of Acronyms **xv**

List of Symbols & Subscripts **xvii**

Preface **xxi**

1 Introduction **1**

1.1 Overview 1

1.2 Problem statement 3

1.3 Objectives 10

1.4 Methodology 14

1.5 Overview of Dissertation 18

2 The Quad-rotor **20**

2.1 Case Study Introduction 20

2.1.1 The Quad-rotor 21

2.1.2 Theory of Flight 21

2.1.3 Considerations Regarding Craft Dynamics 23

2.1.4 Flight Control . . . 25
2.2 Model . . . 30
2.2.1 Model assumptions . . . 31
2.2.2 The Newton-Euler Model . . . 33
2.2.3 Quad-rotor Forces . . . 37
2.2.4 The Hybrid Reference Frame . . . 42
2.3 Parametrization . . . 46
2.3.1 Parameter Summary . . . 46
2.3.2 Calculated Parameters . . . 47
2.3.3 Measured Parameters . . . 48
2.3.4 Parametrized Equations . . . 52

3 Controller Analysis **55**
3.1 Methodology for controller analysis . . . 55
3.1.1 Principles of Control Systems . . . 55
3.1.2 Typical Control Issues . . . 57
3.1.3 Controller Performance Evaluation . . . 61
3.1.4 Model Characterization . . . 67
3.1.5 Comparison Methodology . . . 71
3.2 Control Loops . . . 74
3.2.1 Flight Modes . . . 74
3.2.2 Attitude Control . . . 75
3.2.3 Trajectory Control . . . 80
3.2.4 Altitude Control . . . 82
3.2.5 Controller integration . . . 86
3.3 Controller Identification and Characterization . . . 87

3.3.1 Proportional, Integral and Derivative (PID) Controllers 87
3.3.2 Fuzzy Logic Control (FLC) 96
3.3.3 Artificial Neural Network (ANN) Control 101
3.3.4 Model Predictive Control (MPC) 104
3.4 Controller Comparison 107

4 Simulation **111**
4.1 Controllers 111
4.1.1 Orientation Controller 111
4.2 Comparative Simulator 115
4.2.1 Structure 116
4.2.2 Noise 117
4.3 Simulator (Full model) 120

5 Simulation Results **121**
5.1 Orientation Control 121
5.1.1 Parameter Scaling 121
5.1.2 Noise Performance 126
5.1.3 Reference Tracking 128
5.1.4 Angular control interpretation 129
5.2 Altitude Control 131
5.3 Full Model Simulation 135
5.4 Model Verification 141
5.4.1 Comparison Model 141
5.4.2 Verification 143

6 Conclusion **149**

6.1 Summary 149
6.2 Validation 150
6.3 Verification 150
6.4 Conclusions 151
6.5 Recommendations 152
6.6 Future Studies 152

Bibliography **154**

A The Newton-Euler Model **158**

A.1 Kinematics 159
A.2 Dynamics 163

B Aerodynamic Considerations **165**

List of Figures

1.1 Control level model overview 5

1.2 Research Methodology 14

2.1 The Quad-rotor 22

2.2 Typical Quad-rotor orientation 22

2.3 Quad-rotor orientation (as implemented) 22

2.4 Forces used to control flight dynamics 26

2.5 Axis used to define flight dynamics 27

2.6 Basic functional flow 28

2.7 Basic control flow 29

2.8 Reference frames 34

2.9 Experimental set-up to measure thrust characteristics 48

2.10 Rotor speed measurement results 49

2.11 Thrust measurement results 49

2.12 Torque measurement results 50

2.13 Actuator linear model 51

3.1 Components of a typical closed loop control system 56

3.2 Dampening ratio 65

3.3 Step response with performance evaluators 65

3.4 Controller behavioural characteristics 66
3.5 Control loop characteristics and signals 67
3.6 Comparison methodology 71
3.7 Pitch angle interpretation 76
3.8 Pitch controller 76
3.9 Roll angle interpretation 77
3.10 Roll controller 78
3.11 Yaw angle interpretation 78
3.12 Yaw controller 79
3.13 Attitude control integration 80
3.14 Effective thrust 81
3.15 Altitude Stabilizer 84
3.16 Attitude compensator 84
3.17 Hover controller 85
3.18 Altitude control integration 85
3.19 Control integration 86
3.20 Classic three term PID controller 87
3.21 Anti-windup protection of PID controller 90
3.22 Series PID controller 90
3.23 Normalized response of various Proportional, Integral and Derivative controller combinations 91
3.24 Fuzzy Logic controller structure 96
3.25 Fuzzification of input 97
3.26 Fuzzy Inference engine 98
3.27 Artificial Neural Network (ANN) structure 101
3.28 A discreet MPC scheme 104

4.1 PD controller structure 112
4.2 PID controller structure 112
4.3 PI+D controller structure 113
4.4 FLC membership functions 114
4.5 FLC response curve 114
4.6 Simulator structure 116
4.7 Measurement noise 118
4.8 Load disturbance 119
4.9 Actuator disturbance 120

5.1 PD Controller step response 122
5.2 PID Controller 123
5.3 PI+D Controller 124
5.4 Fuzzy Logic Controller step response 125
5.5 Noise performance comparison 127
5.6 Reference tracking comparison 128
5.7 PD Controller reference tracking 129
5.8 Yaw control without angle interpretation 130
5.9 Yaw control with angle interpretation 130
5.10 Unbounded altitude controller failure 131
5.11 Unbounded altitude controller destabilizing pitch control 132
5.12 Bounded altitude controller performance 132
5.13 Bounded altitude controller attitude change test 133
5.14 Bounded altitude controller actuation 133
5.15 Altitude control during craft inversion 134
5.16 Uncompensated altitude controller actuation 134

5.17 Attitude compensated altitude controller actuation 135
5.18 Altitude stabilizer actuation 135
5.19 Full model simulation 1: Orientation 137
5.20 Full model simulation 1: Position 138
5.21 Full model simulation 2: Orientation 139
5.22 Full model simulation 2: Position 140
5.23 Step response verification 144
5.24 Verification of angle interpretation 146
5.25 Verification of translational motion (versus time) 147
5.26 Verification of translation motion (observer views) 148

A.1 Reference frames 160

B.1 Laminar and Turbulent flow 166
B.2 Ideal thrust distribution along propeller blades 167
B.3 Blade Tip Vortex 168
B.4 Vortex ring state 169

List of Tables

2.1 Calculated Parameters 47

2.2 Measured Parameters 51

3.1 Various PID controller structures 91

3.2 PID parameters and influence 93

3.3 Controller comparison: System Level 109

3.4 Controller comparison: Technical Level 110

4.1 Fuzzy Logic Control Rules 113

5.1 PD Controller Performance 123

5.2 PID Controller Performance 123

5.3 PI+D Controller Performance 124

5.4 Fuzzy Logic Controller Performance 125

5.5 Controller step response performance comparison 126

5.6 Controller noise performance comparison 127

5.7 Controller reference tracking performance comparison 128

List of Acronyms

AD Airworthiness Directive

AI Artificial Intelligence

ANN Artificial Neural Network

CAD Computer Aided Design

CIA US Central Intelligence Agency

CoM Centre of Mass

CTOL Conventional Take-off and Landing

DoD US Department of Defence

DOF Degrees of Freedom

ESP Electronic Stability Program

FBW Fly-by-wire

FLC Fuzzy Logic Control

GPS Global Positioning System

IAS Indicated Air Speed

IAE Integral of the Absolute Error

ISE Integral of the Squared Error

ITAE Integral of the Time and Absolute Error

ITSE Integral of the Time and Squared Error

LQR Linear Quadratic Regulator

MATV Multi Axial Thrust Vectoring

MIMO Multiple Input, Multiple Output

MPC Model Predictive Control

NMPC Non-linear Model Predictive Control

PID Proportional, Integral and Derivative

PO Percentage Overshoot

PWM Pulse Width Modulation

QP Quadratic Problem

RPV Remote Piloted Vehicle

SISO Single Input, Single Output

SoS System of Systems

STOL Short Take-off and Landing

UAS Unmanned Air Systems

UAV Unmanned Aerial Vehicle

UCAV Unmanned Combat Aerial Vehicle

USAF United States Air Force

VTOL Vertical Take-off and Landing

List of Symbols & Subscripts

Subscripts

B	Body reference frame
E	Earth reference frame
H	Hybrid reference frame
Θ	Orientation

Reference frames

The reference frames are based on the right handed orientation and rotation.

$(\mathcal{O}_E, x_E, y_E, z_E)$	Earth reference frame
$(\overline{\mathcal{O}}_B, x_B, y_B, z_B)$	Earth reference frame
$\mathbf{R}_\Theta$	Rotation matrix
$\mathbf{T}_\Theta$	Transfer matrix
$\mathbf{J}_\Theta$	General conversion matrix

Earth reference frame

$\overline{\Gamma}^E$	Position of craft / Position of B-frame ($\overline{\mathcal{O}}_B$)	$[m]$
$\overline{\Theta}^E$	Orientation of craft / Orientation of B-frame	$[rad]$
ϕ	Roll (Rotation around X-axis)	$[rad]$
θ	Pitch (Rotation around Y-axis)	$[rad]$
ψ	Yaw (Rotation around Z-axis)	$[rad]$
$\overline{\xi}^E$	General position vector	$[m \quad rad]$
$\overline{V}^E$	Translational velocity vector	$[m.s^{-1}]$
$\overline{F}^E$	General forces vector	$[N]$

Body reference frame

$\overline{V}^B$	Translational velocity vector	$[m.s^{-1}]$
u	Forward translational velocity	$[m.s^{-1}]$
v	Lateral translation velocity	$[m.s^{-1}]$
w	Vertical translational velocity	$[m.s^{-1}]$
$\overline{\omega}^B$	Angular velocity vector	$[rad.s^{-1}]$
p	Roll rate	$[rad.s^{-1}]$
q	Pitch rate	$[rad.s^{-1}]$
r	Yaw rate	$[rad.s^{-1}]$
$\overline{\nu}$	Generalized velocity vector	$[m.s^{-1} \quad rad.s^{-1}]$
$\overline{F}^B$	General forces vector	$[N]$
$\overline{\tau}^B$	General torque vector	$[N.m]$
$\overline{\Lambda}$	Generalized force vector	$[rad.s^{-1}]$

Hybrid reference frame

$\overline{\zeta}$	Generalized velocity vector	$[m.s^{-1} \quad rad.s^{-1}]$

Model Parameters

m	Total mass of craft	$[kg]$
I_{XX}	Rotational inertia around X-axis (Roll)	$[N.m.s^2]$
I_{YY}	Rotational inertia around Y-axis (Pitch)	$[N.m.s^2]$
I_{ZZ}	Rotational inertia around Z-axis (Yaw)	$[N.m.s^2]$
g	Gravitation constant	$[m.s^{-2}]$
b	Aerodynamic propeller thrust constant	$[N.s^2]$
d	Aerodynamic propeller drag constant	$[N.m.s^2]$
l	Symmetric distance of motot to axis of rotation	$[m]$
J_{TP}	Total rotational moment of inertia around rotor axis	$[N.m.s^2]$

Derived parameters & Quad-rotor forces

$\mathbf{I}$	Inertia matrix of craft	$[N.m.s^2]$
$\mathbf{M}$	System inertia matrix	$[kg \quad N.m.s^2]$
$\overline{G}$	Gravitational force vector	$[N \quad N.m]$
$\mathbf{U}(\overline{\Omega})$	Movement force vector	$[N \quad N.m]$
$\mathbf{E}(\overline{\xi})$	Movement matrix	$[N \quad N.m]$
$\mathbf{C}(\overline{\zeta})$	Coriolis-centripetal matrix	$[N \quad N.m]$
$\mathbf{O}(\overline{\zeta})$	Gyroscopic rotor matrix	$[N \quad N.m]$
U_i	Subsystem of control forces	$[N]$ / $[N.m]$
$\overline{\Omega}$	Propeller speed vector	$[rad.s^{-1}]$
Ω_i	Propeller i speed	$[rad.s^{-1}]$
Ω_T	Net propeller speed	$[rad.s^{-1}]$

Conventions

$c_k = \cos k$

$s_k = \sin k$

$t_k = \tan k$

$$\mathbf{S}(\overline{k}) = -\mathbf{S}^T(\overline{k}) = \begin{bmatrix} 0 & -k_3 & -k_2 \\ k_3 & 0 & -k_1 \\ -k_2 & k_1 & 0 \end{bmatrix} \qquad \overline{k} = \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix}$$

Hadamard product (entry-wise matrix multiplication)

$$\mathbf{A} \circ \mathbf{B} \qquad \rightarrow \qquad (\mathbf{A} \circ \mathbf{B})_{i,j} = \mathbf{A}_{i,j} \times \mathbf{B}_{i,j}$$

Preface

---

*"**The saddest aspect of life right now is that science gathers knowledge faster than society gathers wisdom.**" - Isaac Asimov*

Automation technologies have been around since the commencement of the electronic age. Many of these technologies have been out of the public eye, but the recent increase in development and use of autonomous and unmanned vehicles in warfare has caught the world's attention. There is currently more than twelve thousand robotic systems deployed in Iraq, with the numbers growing daily [1]. These robotic machines have become an integral part of modern warfare and each generation is more advanced, intelligent, capable and lethal.

Many of these machines are remotely operated systems, with very little intelligence of their own. Even with the continuous creation of more advanced Artificial Intelligence (AI) systems, technology has yet to reach the point of creating artificial sentience. These AI technologies have led to the creation of complex intelligent systems that has caught the imagination of society. The problem with these bleeding edge technologies is that although they may be responsible for great leaps in technology, they are often impractical or inefficient in comparison to their human counterparts. An example of such a system is the Honda Asimo humanoid robot. Development started as early as 1986 and has yet to reach the point of developing a humanoid robot that can even hope to

match the performance of a human.

This however does not mean that artificial intelligence and control technologies are always impractical. There have been many implementations of control systems that not only outperform their human counterpart, but often work in symbiotic relationship with a human operator. In many of these situations the human operator may not even be aware of the control system. The level of intelligence may vary, but the impact of the system may be well beyond what meets the eye. These assistive systems include predictive algorithms that manage the resources of a personal computer to improve real-time performance, driver aids such as the Electronic Stability Program (ESP), through to automated call centres that rely on voice recognition. None of these systems require sentience, but are to various degrees intelligent allowing for automation of specific tasks. The automation of flight controls on modern combat aircraft has made it possible to design airframes that are aerodynamically unstable giving them increased aerodynamic performance with regard to manoeuvrability, but making them impossible to fly without computer assistance. These Fly-by-wire (FBW) computers make hundreds of corrections per second to canards in order to maintain flight stability, but their intelligence is limited. They aren't able to fly the aircraft without the aid of a pilot, just as the pilot is unable to maintain flight stability without the aid of the Fly-by-wire system.

The automation of vehicles has taken a predominant leap in the last few decades with the introduction of technologies such as Unmanned Aerial Vehicles (UAV). Although very few of these systems are fully automated, the level of automation and the impact of these technologies have led to great concern with regard to the safety, legal and moral implications of these systems, especially with regard to automated military technologies with lethal capabilities. This trend to incorporate lethal capabilities can be witnessed in the MQ1 Predator Unmanned Aerial Vehicle (UAV), which was originally developed for the US Central Intelligence Agency (CIA) for use as a reconnaissance drone back in the 1990s. As the project progressed and its usefulness became apparent, the United States Air Force (USAF) pushed for an armed version. This led to the integration of Hellfire anti-armour missiles into the MQ1A in 2001 with pres-

sure from both the USAF and the CIA's Counter Terrorist Centre. This later led to the development of the MQ9 Reaper and the MQ1C Sky Warrior, both combat oriented variants of the MQ1. Larger systems have also been deployed such as the RQ4 Global Hawk boasting a 35m wingspan, similar to that of a Boeing 737.

There is also the matter of social acceptance, as many people object to automation due to a lack of faith in or understanding of the technology. Very few people are aware that even the fly-by-wire system in modern airliners has the ability to overrule the pilot to prevent accidents. The philosophies behind these systems vary in regard to whether the system puts ultimate control in hands of the computer, or in the hands of the pilot, allowing the pilot to bypass the on-board computers. In other cases it's a matter of paradigm conflict as in the case of the US Air Force, which only launched their first series of development programs in 2009, as drafted in their document "Unmanned Air Systems (UAS) Flight Plan 2009 - 2047" [2]. Even though the US Air Force only recently accepted the use of unmanned aircraft as part of their primary doctrine, other branches of the US Department of Defence (DoD) have proven the efficiency in the use of unmanned aircraft. A decade ago, in 2000, the DoD had fewer than 50 unmanned aircraft in its inventory, a number which has increased to over 6800 by October 2009 [1].

These unmanned systems are not new, with the first demonstration of a remotely operated vehicle being done by Nicola Tesla, the father of wireless technology. During World War II , unmanned systems were used not only on land and sea, but also in the aerial battlefield [1]. The German army used unmanned *FL-7s* boats, packed with explosives to defend their coast line. In 1944, the Allied forces started a project called "Operation Aphrodite" where they modified bombers to be piloted remotely from other aircraft, then packed them with explosives and flew them into military targets. However due to political and other factors, these technologies were never pursued to their full potential until now.

Although both philosophies on automation can be justified, the deciding factor should be the reliability of the system. This is one of the reasons for the level of automation, as the actions of a well programmed system is far more predictable than the actions of a

human operator. Human operators only have a limited concentration span, compared to computers that can continue working indefinitely (not considering maintenance cycles). A human operator on the other hand has greater experience and the ability to find new and creative solutions to a problem. Hence the creation of a practical automation system requires the engineer to determine the level of automation and the required intelligence to create a mutualistic* relationship between the AI and the human operator. This means that the engineer must use the AI to compensate for the weaknesses or deficiencies of the human operator, while the operator compensates in the same way for the AI, hence extracting the advantages of both human and machine, to optimize the overall performance and reliability of the system. The use of proper systems engineering techniques makes it possible to accurately make this distinction and develop an optimal system.

*mutualism: A symbiotic relationship in which both parties benefits.

Chapter 1

Introduction

---

*"**A complex system that works is invariably found to have evolved from a simple system that works.**" - John Gaule*

1.1 Overview

The study of automation requires a thorough understanding of the types of controllers and the characteristics of each. It also requires that the problem be understood in sufficient detail to allow for the derivation of a model or at least a detailed functional analysis to aid in the development of the controller.

The hypothesis of this study is that the selection of the controller type should not only be done based on technical considerations for optimal control, but rather based on the characteristics of the controller best suited to the required problem. That is to say that if time to market is priority, the controller that is the quickest to implement should be considered, if parameters are variable an adaptive controller might be better suited

and if accuracy is the primary concern, the selected controller should be a close approximation of the inverse of the system.

In this study the case of flight automation is considered, for it not only provides a technically challenging problem, but also has a series op practical considerations and model interpretations that need to be addressed. Firstly the system operates in a dynamically unstable environment due to unpredictable external factors such as wind, changes in air pressure and temperature, turbulence and many other factors. These factors are usually omitted from the models to simplify the mathematics, but ultimately need to be taken into account as they are unavoidable during real world operation. Other changes in parameters to consider are variations in payload configurations and fuel loads, etc.

The case study is that of a quad rotor helicopter. Although discussed in greater detail in later sections of this document, the basic concept behind the quad rotor is that it consists of four fixed pitch rotors, where control of the flight dynamics is done by changing the rotor speeds in various combinations. Since the system has to be controlled in six degrees of freedom, i.e. three axes of linear and three axes of angular motion, using only four motors (inputs), the system is considered to be an under actuated system. The combination of being under actuated and operating in a dynamically unstable environment presents a significant technical challenge. However, through the use of systems engineering principles the interdependence of the dynamics can be characterized and the system simplified, resolving the under actuation issue. In essence this hypothesis states that the linear motion, or trajectory, is dependent on the angular motion, or orientation. Thus creating a sequence of systems where the trajectory (3 axes) is controlled by the orientation (3 axes), which in turn is controlled by the rotor speeds (4 inputs). It is clear that the system can no longer be considered under actuated. In fact, the orientation and trajectory are so fundamentally linked that they can not be manipulated without the one affecting the other.

1.2 Problem statement

Automation of flight does not necessarily imply the replacement of the pilot; it simply reduces the involvement of the pilot in maintaining flight stability or navigational control. This may be to improve safety or comfort on commercial airliners, or to reduce the workload on a pilot to improve mission capabilities. The latter can be witnessed in the F-15 Eagles Multi-stage Improvement Program [3], which consists of a major overhaul of the avionics to extend the service lifetime of the aircraft. In this case the aircraft was still capable with regards to aerodynamic performance, but the outdated avionics placed too large a load on the pilot to keep up with the pace of modern warfare. In fact this upgrade to extend the service life, has made the F-15 one of the most successful fighter aircraft of the 20th century. There are also cases where the pilot is removed from the aircraft for safety reasons, such as remote piloted Unmanned Combat Aerial Vehicle (UCAV). In other cases the pilot is removed for practical reasons, such as extending mission length (RQ-4 Global Hawk) and increasing payload capacity. This requires an increased level of complication with regards to Remote Piloted Vehicle (RPV) as the system is completely dependent on the automation system. If communication is lost with the ground control station, the aircraft must be able to compensate and implement a rudimentary strategy to regain communications or continuous operation. This increased requirement for independence has led to the search for a fully automated UAV, which does not require any human piloting, onboard or remote. The Global Hawk is equipped with a fully automated pilot-less system, where an operator only gives mission specific commands such as take-off, land, fly to a waypoint or perform aerial surveillance of a specified area [1].

Automation may not only be limited to flight control, but to other systems such as communications, environmental controls and life support, target acquisition, weapons management and diagnostics. These systems integrate to allow for even greater automation and have led to the creation of technologies such as hybrid damage adaptive flight controls in the latest generation of combat aircraft [4]. The integration of these systems and their individual complexity has led to the adaptation of the System of

Systems (SoS) approach in design. The System of Systems approach allows for large scale integration without being weighed down by the detailed internal working of each subsystem. The large system is divided into task-oriented or dedicated systems with five common characteristics [5]; i.e. operational independence of the individual systems, managerial independence of the systems, geographical distribution, emergent behaviour and evolutionary development. Thus the integration of the systems allows for interoperability and synergism, allowing for extensive automation. By identifying the correct tasks and interfaces required to control an aspect of the system, this aspect can be separated and automated, without affecting the rest of the larger system.

The process of flight automation can be divided into various tasks using the Systems Engineering approach, as indicated by the control level model in Figure 1.1.

On the lowest level is the Orientation controller, which controls the angular motion, i.e. the orientation or attitude of the aircraft, hence maintaining flight stability. This controller needs an exact understanding of the dynamics of the airframe, the kinetics of each canard (flight control surface) and the propulsion system. Hence it is specific to each aircraft and needs recalibration after any changes to the airframe design. At this level the change in orientation is related to a change in each canard or engine thrust. This is by default part of the fly-by-wire systems in modern aircraft.

The Trajectory controller is the next level of control, which is responsible for the linear motion of the aircraft. Controlling the linear motion requires an understanding of the effect the orientation has on the trajectory of the craft. This translates to an understanding of the combined use of canards to affect the trajectory of the aircraft. It should also be aware of the change in effect of each canard on the trajectory, dependent on the current orientation and aerodynamic status of the aircraft. The controller is only sensitive to the type of aircraft and layout of canards. The Orientation and Trajectory controller together control the flight dynamics of the aircraft and are purely based on the physics of flight, hence requiring a relatively low level of intelligence. At this level of automation it reduces the skill level required to pilot the aircraft, but the system does not have the intelligence to function independently. In the case of military aircraft the fly-

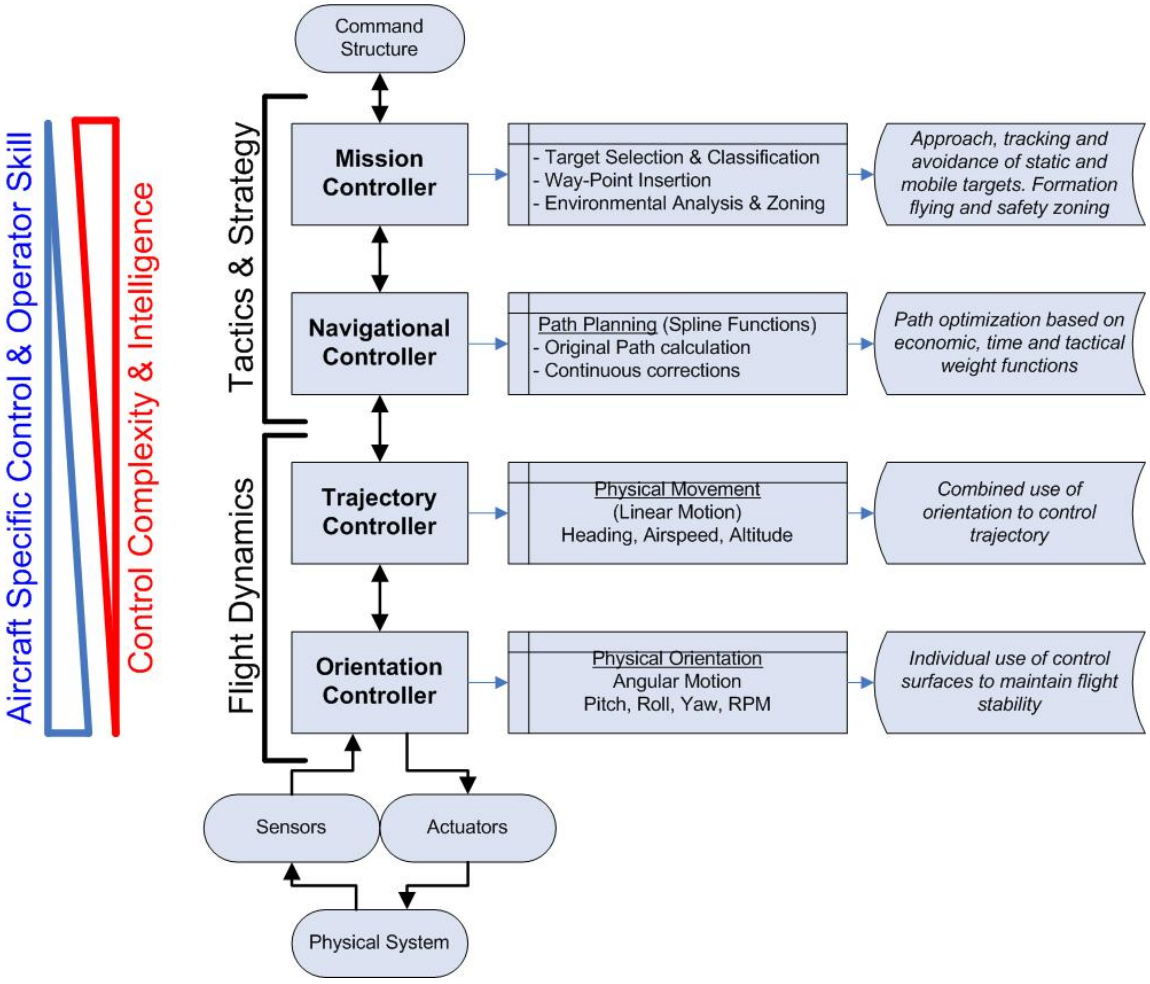


Figure 1.1: Control level model overview

by-wire system allows for the piloting of an aerodynamically unstable airframe with relative ease, reducing the load on the pilot while increasing aerodynamic performance with regard to manoeuvrability.

The next two levels of automation according to this model, is the *Navigational and Mission controllers*. These two levels combine to automate tactical and strategic aspects of the flight. The mission controller is used to identify way-points and classify them, either as destinations, obstacles or treats, etc. The navigational controller is responsible for determining the best path to the way-points, while avoiding obstacles and taking into account tactical considerations such as approach vectors. The navigational controller is only directly aware of the specialized flight characteristics of the aircraft with regard to the flight envelope. The increased level of intelligence needed is clear and

the predictability is reduced with each layer of automation. It is also worth noting that as the level of complexity increases, the required execution speed is reduced, i.e. the corrections to maintain flight stability might need to occur many times per second, but alterations to the path may only need to be performed once every few seconds.

The model is not limited to these four stages and can continue to branch. Higher levels may include communications for remote operation, collaboration levels to improve cooperation between various aircraft or even other entities, and to incorporate higher command structures. Lateral expansion might pertain to the inclusion of target identification and selection modules, weapons management, advanced tactical systems, diagnostic systems, surveillance systems or other support systems. The model also expands downward to incorporate sensors and the filtering techniques to analyse the measurements. Without these lower levels it wouldn't be impossible for the AI to perceive the physical world. A more detailed model is developed throughout this study.

The division of the layers are so that a pilot can be inserted at any level and perform the role of the higher levels. For example, if only the orientation controller is implemented, a skilled pilot is required to operate the craft. The pilot still needs to understand the effects of each canard and use them in the correct combination to control the trajectory. Another example would be an autopilot system, although the autopilot may have the ability to follow a programmed flight path, it may not be able to avoid obstacles. Thus the system is not independently in control of navigation and it cannot function without the trajectory and orientation controllers.

The interfaces between levels are there to transfer the requested state to the lower level and to pass the current status up to the higher levels. Certain aircraft specific performance parameters pertaining to the flight envelope also need to be passed to the higher levels, for example the navigation controller needs to be aware of the maximum turn rate and the cruising speed of the aircraft; the trajectory controller needs to understand the effectiveness of each canard and the maximum acceleration the airframe can handle. As the model is populated, the synergy becomes evident. Although each stage is assigned to a specific task or set of tasks and performs these tasks independently, the

various stages collaborate to achieve the goal of automating the flight. The independence of each stage refers to its indifference to changes in the design of other stages. Each subsystem is affected by surrounding subsystems to some extend, but as long as the interface specification is followed, any changes with regard to internal design should not affect the functioning of the other subsystems. It may affect overall system performance, but should not affect functionality. This division allows the higher levels to become more generic with regard to the airframe, while the lower level are less complex and less adaptable.

As stated above, the low level orientation controller is dependent on the exact aerodynamics of the aircraft, making it specific to each airframe. This controller understands the effect of each canard on the orientation of the aircraft and how its effect changes during various aerodynamic conditions. This includes the increase in force on each canard as the airspeed over the canard increases, requiring a scaling factor in the controller to compensate as the airspeed increases. This level can be developed to a more complex system that not only adapts to aerodynamic conditions, but adapts to the environmental conditions, such as turbulence and wind, as well as changes in efficiency of canards due to damage through the use of auto calibration techniques. With a very advanced adaptation system, it may be possible to gain enough control over the aircraft to improve safety, such as avoiding populated areas during a crisis when ground impact is unavoidable or getting a pilot to a safe zone before having to eject.

The trajectory controller should theoretically be able to function on any craft of similar configuration, from a small single engine private plane to a large commercial airliner. Although the performance will diminish due to calibration factors, the basic functionality should still be sufficient to maintain flight. If the trajectory controller has a thorough understanding of the flight dynamics, not only will the automated system be able to prevent a stall, but also recover from one, which a basic autopilot system is incapable of. By using a vector based control system, as opposed to a logic rule based system, the trajectory controller can always be aware of the effectiveness of each canard with regard to the current trajectory and its aerodynamic properties, possibly allowing for automated recovery from stalls, flat spins and other problematic aerody-

namic conditions. When this vector based technique is combined with an aircraft fitted with a thrust vectoring system, such as the F-16 Multi Axial Thrust Vectoring (MATV) demonstrator, the trajectory controller would even be capable of performing advanced manoeuvres, such as an controlled flat spin ("*the helicopter*") or a post stall loop ("*the hammer head*"). However this would require a very accurate and complex vector model of the aerodynamics of each canard and the dynamics of the thrust vectoring system. These vectors are passed up from the orientation controller as torques with regard to each axis. The trajectory controller uses these torques to find the optimal change to achieve the correct trajectory.

The navigational controller is only dependent on the manoeuvrability of the aircraft, referred to as the flight envelope. In essence it needs to understand the maximum and minimum airspeed, the maximum turn forces and maximum climb rates of the aircraft. This information translates into the maximum and minimum turn radius, maximum maintainable climb angle, hovering or circling capabilities, cruising speed etc. The navigation controller needs to understand the landing and take-off profiles of the aircraft. It distinguishes between Vertical Take-off and Landing (VTOL), Short Take-off and Landing (STOL) and Conventional Take-off and Landing (CTOL) fixed wing aircraft and helicopters. Helicopters are restricted to a specific height-velocity diagram to allow for auto-rotation as safety precaution. In a fixed wing aircraft for example, the plane needs to be aligned parallel with runway before touchdown, with a specific descend rate and airspeed. The optimal orientation for landing is also needed to avoid events such as tail strikes, a common problem on airliners. These characteristics are passed up from the trajectory controller. The mission controller serves as an intelligent integration point of various operational stacks in the system model. With regard to the flight stack, it might only be concerned with identifying accessible areas and viable approach trajectories based on the limitations of the aircraft.

On the subject of reliability, the operational independence is crucial. This does present a problem in regard to the interdependence of the subsystems. Each subsystem performs a task independently, but the performance of the entire system is dependent on the collaboration of these various subsystems. Any failures or deviations in perfor-

mance will affect the overall system; hence the design needs to compensate for this to the extent that it possible. This is usually done through the inclusion of redundancies. There are however realistic limitations to the independence of the subsystems, i.e. that the failure of lower subsystems will undoubtedly affect higher functions. For example, damage to any canard will affect the orientation, trajectory and navigation controllers, ultimately reducing mission effectiveness. This leads to the introduction of technologies such as the hybrid damage adaptive flight controller, allowing the system to compensate for the problem. However depending on the extent of the damage, this may not always be possible. Other cases may seem to be fatal, such as total hydraulic failure resulting in no control of all canards, yet modern airliners have a fly-by-wire system that can use the unbalances thrust of multiple distributed engines to steer the craft with limited manoeuvrability, but still enough to make a safe landing. A total loss of engine thrust for example can be compensated for in helicopters by switching the controller into an auto-rotation mode and in the case of planes into a glider mode, again with only limited control. It is still worth noting that the flight computers of the space shuttle allow for an un-powered re-entry with incredible accuracy.

This interdependence is the primary focus of the risk analysis. For this purpose the model uses the convention that in each vertical stack the above systems are dependent on the lower systems, and although lateral systems offer support they are not critical to the relevant stack. For example, failure of the weapons management system may affect the mission effectiveness of the aircraft, but it does not affect the flight control stack. The aircraft may lack the ability to engage, but can still perform other mission critical tasks such as surveillance, communications relay and more. There are cases where failure in other subsystems may result in introduction of limitations for safety, such as speed limitations should the landing gear fail to retract. This will still be visible in the model as the landing gear forms part of the actuator system, hence part of the flight control stack. Hence the control system for the landing gear should be part of the flight control stack, or at least present in one of the sub branches that is controlled by this stack. By keeping with the convention the process of risk analysis is simplified, making it easy to identify the cascade effect of each failure. These cascaded effects are

of great concern in any engineering problem, as even the smallest failure can have dire results. For this I can refer to the example of the Air France Flight 447 which went down June 1, 2009. The disaster was caused by a failure in the pitot tubes which resulted in an incorrect Indicated Air Speed (IAS). This was a known risk as the manufacturer made an announcement to replace the probes, but since it was a minor issue and not part of the Airworthiness Directive (AD) the airlines had the freedom to replace them at their own discretion. In the case of the Flight 447 the incorrect reading in airspeed combined with intense weather conditions, led to constant change in actual airspeed as the plane met alternating head and tail winds. The computers detected the disparity in the indicated air speed and disengaged the autopilot and auto-thrust systems, a standard action when the plane enters a stall. The pilots could not however recover from the stall and the stall prevention protocols in the fly-by-wire system were not functional due to the failure of the pitot tubes. This could have been prevented had there been sufficient backups for calculating airspeed, more accurate weather information, or perhaps some other techniques which have yet to be developed.

1.3 Objectives

During the course of this study, the various aspects of elementary flight automation will be analysed. The aircraft in question will be characterized based on its physical attributes and the fundamental flight dynamics. The field of application will be taken into consideration to produce the functional requirements and combined with physical analysis to produce the performance requirements to complete the requirements analysis. Then the functional analysis is compiled based on the actions required to control the various aerodynamic manoeuvres in reference to the user inputs. All the required parameters are documented, along with the performance parameters for benchmarking and optimization. The combination of the functional analysis and parameters provides the required information to define the architecture of the controller. Then a study is done of possible control techniques and they are compared to this derived architecture. The controller is also analysed based on required development

time and resources for implementation, such as availability of technical data, complexity and required engineering skills. The controllers are implemented in simulation and a log is kept of the process to develop each controller type. Each controller is rated based on its suitability with regard to the problem and then the optimal controller is selected for implementation. The next step is identification of functional and physical limitations of the aircraft to define safe operational limits. The implementation is then optimized and run through a thorough testing and evaluation cycle. A short risk analysis is done to identify problem areas and possible solutions are discussed. The study is then rounded of with a final reflection on the entire project life cycle.

Following the discussion of the previous section on level of control, only the two lower levels will be implemented, i.e. orientation and trajectory control, constituting a flight dynamics controller.

Using the standard tools of systems engineering, a concise summary of the study scope can be given as follows:

- Introduction to case study
- Requirements analysis
    - Characterization of airframe (Physical)
    - Performance requirements (Technical)
    - Resources
- Functional analysis
    - Operational
    - Functional
    - Technical
        - Physical parameters identification (Outputs)
        - Control parameters identification (Inputs)
        - Performance evaluation parameters identification (Monitors)
- Limitation (Flight envelope)
    - Physical
    - Operational
    - Safety
- Controller study & simulation
    - Develop mathematical model
    - Identify controller types
    - Parameters and controllability
    - Implementation resource dependence
    - Benchmarking
- Testing and Evaluation

  - T & E master plan
    - Stable hover
    - Basic movement
    - Complex movement
    - Model verification
    - Controller validation
  - Adjustment
  - Conclusion
- Life-cycle review
  - Study
  - Development
  - Evaluation
  - Conclusion

1.4 Methodology

Figure 1.2 shows the general research process and the various contributors that are relevant. These contributors are the inputs, the various constraints that limit the study and the resources required to complete the study.

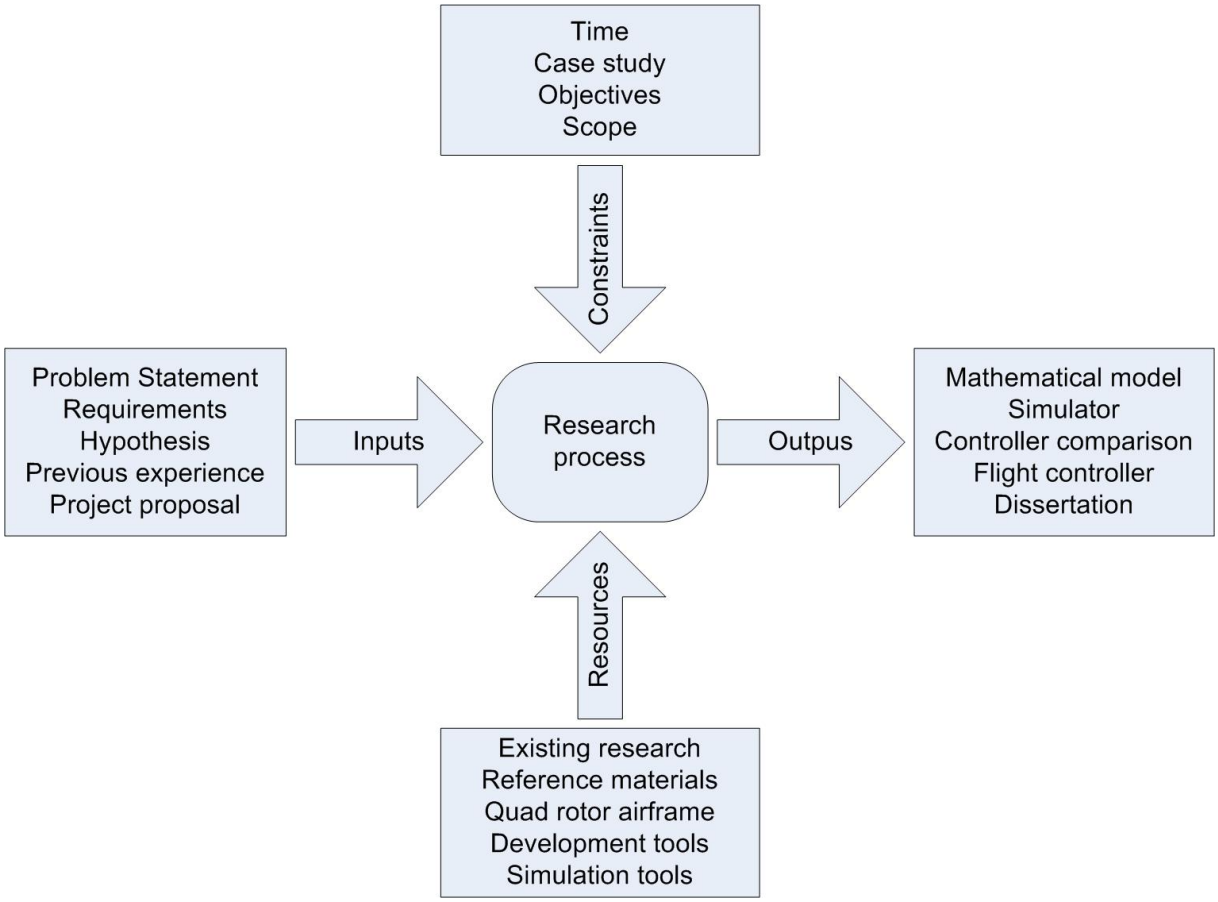


Figure 1.2: Research Methodology

Inputs

The initial inputs of the study are what defines the baseline of the study, the problem statement. The requirements, with regard to functionality, performance and maintainability, are also provided and used to derive the technical scope of the project. Gen-

erally the study direction is determined by some general hypothesis, providing a reference for the required solution. It is also good practice to review existing research on the problem, if available, and include it into the research. Finally the initial review is summarized in a project proposal, that if accepted provides the guidelines for the study.

For this study the inputs are as follows:

- Problem statement
- Requirements
- Hypothesis
- Previous experience
- Project proposal

Constraints

The constraints of the study are necessary to limit the structure and scope of the study. Without a proper scope the project may never reach completion. The problem statement is expressed as a series of objectives that need to be resolved and their individual priorities. This is used to define the scope of the project with regard to operational and functional considerations. Time and budget constraints are usually responsible for technical scope limitations.

The relevant constraints on this project are:

- Time: 2 year study period
- Financial: Study grant (if available)
- Case study: This study focuses only on the case of a quad-rotor helicopter
- Objectives (Discussed in previous section)
- Scope

- Mathematical modelling with simplifying assumptions
- Controller identification and comparison
- Simulation
- Implementation

Resources

Any project is dependent on various resources to ensure successful completion and these resources need to be identified and allocated at the start of the project. These resource can either be academic, technological, technical skills, development tools, simulation tools, Computer Aided Design (CAD) tools, logistical or financial.

The resources to be utilized during this study are:

- Existing research
  - General field of application
  - Various mathematical models and their assumptions
  - Various control techniques and results obtained
- Quad-rotor airframe
- Development, simulation and CAD tools

Outputs

The outputs of the projects are primarily related to the objectives, but not limited to them. Some outputs are produced that are needed for further steps, others are the results of unexpected events such as knowledge gained from failures or identification of new techniques. In general the project outputs include the project objectives (for example a product, plant or process), the relevant documentation and results.

The prescribed outputs of the study are:

- Mathematical model for quad-rotor UAVs
- Simulator implementing the mathematical model
- Comparison of various control techniques
- A simulated flight controller
- Dissertation

Research Process

The research process itself is designed to incorporate the inputs and resources, limited to the framework determined by the constraints to ultimately achieve the desired results. This constitutes a thorough understanding of the inputs, managing the resources based on the constraints to ensure that the achieved objectives meet the requirements.

The process followed for this study is discussed in detail in the next section. A brief summary would be that once the model is derived, a study is done on previous results obtained by implementing various controller types and subsequently compared. The most suitable controller is implemented to validate the hypothesis.

1.5 Overview of Dissertation

The dissertation, excluding chapters covered thus far, starts of with a literature study, Chapter (2.1), serving as an introduction to the case study. It gives the reader a thorough understanding of the principles behind quad rotor helicopters and the requirements to automate them. This section consists of discussions regarding the theory of flight, aerodynamic conditions of rotor craft and the framework of the craft.

Next, in Section (2.2) a mathematical model is derived to represent the quad rotor and aid in the development and simulation of the controllers. As with any model, certain assumptions need to be made to simplify the mathematics, but caution is taken to justify each assumption to maintain the accuracy of the model. The parameters for the model now needs to be calculated in order for the model to be representative of the airframe in use by this study. Section (2.3) now discusses the process of parametrization, where the method of calculating each parameter is discussed and each parameter is determined and documented.

With the model complete, it is now necessary to select a controller for the quad rotor. In Chapter (3) a thorough theoretical study is done to compare various control techniques, i.e. each controller is analysed not only based on technical performance but also with regard to characteristics. A table is compiled to list the characteristics of all the control techniques and the three controllers that are best suited to the problem are selected for simulations, shown in Section (4).

For the first set of simulations a single axis model is used, as the inclusion of multiple dimensions complicates the interpretation of the controller performance. This is due to, amongst other factors, the use of multiple reference frames. The results are analysed with regards to stability and sensitivity and the best controller is selected for implementation. The selection is first validated by performing a full multi-axial simulation of the controller. Then the results are given a thorough evaluation to validate the hypothesis. A brief conclusion on the results of the simulations are given.

The second to last Chapter (5) covers the simulation results where the controller is evaluated in the real world implementation and the model is verified. The final validation is done and compared to the simulation.

The final Chapter (6) contains the conclusions and reflections of the study. It identifies the problems encountered and suggests points for further study. The conclusion is followed by various appendices that cover various details that are too long to include in the main study, but are relevant to the process.

Chapter 2

The Quad-rotor

---

*"**The important thing in science is not so much to obtain new facts as to discover new ways of thinking about them.**" - Sir William Bragg*

2.1 Case Study Introduction

Considering the objectives of the study, an aircraft platform has to be selected to test the stability of the various controllers. Since the study is to focus on the control techniques, the platform should be as mechanically simple as possible, require little resources to implement and have vast amounts of information available about its dynamics. The case study used in this dissertation is the quad-rotor helicopter. Of all the various aircraft configurations, from fixed wing and rotary aircraft, the quad-rotor has the simplest mechanical configuration. The greatest drawback of designing a craft with VTOL (Vertical Take-off and Landing) capabilities is the level of complexity of the mechanical systems. Most rotor-craft make use of variable pitch rotor blades to control the craft,

which are very expensive and complex. However, a VTOL configuration does simplify the automation with regards to the flight envelope as the craft does not require complex take-off and landing procedures.

2.1.1 The Quad-rotor

The quad-rotor is a configuration that uses a simple mechanical system, based on four fixed-pitch rotors mounted symmetrically in the four corners of the craft as shown in Figure 2.1. All motion is controlled by changing the rotation speed of the individual rotors in various combinations. This design is very simple to construct and requires very little mechanical skills, making it a practical solution for a test platform. [6] The flight dynamics of the quad-rotor is also ideal for electronic control and it demonstrates the interdependence of the human and machine partnership, as the combination of control sequences and flight dynamics is too complex for a human operator to stabilize without the aid of a fly-by-wire system. It would be possible to construct an alternative version that utilizes a complex mechanical system to enable a human to pilot the system, as is done in many helicopters. This system is however very complex, expensive, fragile and adds unnecessary weight to the aircraft. The quad-rotor is relatively aerodynamically stable, which simplifies the implementation and reduces the need for optimization, making the quad-rotor the perfect test platform to safely analyse controller sensitivity regarding changes in physical parameters.

2.1.2 Theory of Flight

The flight dynamics of the quad-rotor is mechanically speaking quite simple. The four motors are statically mounted facing upward and control action is actuated by alternating the rotation speed of the rotors in various combinations. [6] The change in rotor speed affects the thrust and torque produced by set motor.

Typically the quad-rotor is orientated as shown in Figure 2.2, however due to sensor



Figure 2.1: The Quad-rotor

orientation in this design and the desire to reduce the lateral profile of the craft the orientation is rotated to the orientation shown in Figure 2.3. Note that in this orientation the craft's lateral profile (width) is reduced to improve indoor flight capabilities.

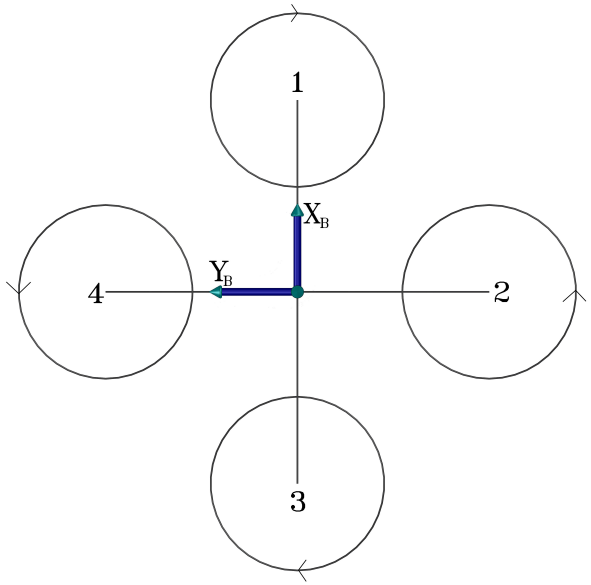


Figure 2.2: Typical Quad-rotor orientation

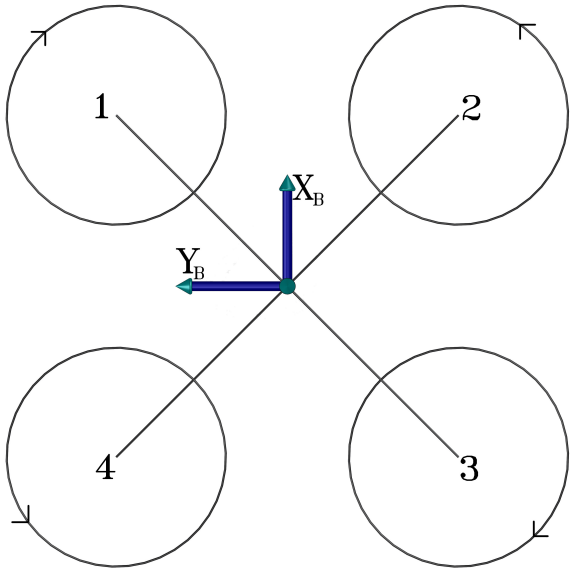


Figure 2.3: Quad-rotor orientation (as implemented)

Orientation control is achieved by the combination of rotor speeds and the resulting forces and torques. [6] Pitch and roll control is dependent on the combined change in angular velocity of the rotors on the one side of the axis of rotation while performing the opposite actuation regarding the motors on the opposing side of the axis of rotation. In order to achieve a rotation around the Y-axis (Pitch) in Figure 2.3, motors 1 & 2 have to be actuated while motors 3 & 4 are inversely actuated. Similarly roll (X-axis) is achieved by actuating motors 1 & 4 and inversely actuating 2 & 3. On the other hand, yaw control is achieved by manipulating the opposing blade pares, i.e. that increasing the rotation speed of motors 1 & 3 while decreasing 2 & 4, will offset the balance in rotational torque, while maintaining the balance in forces along the other axes. In this configuration, actuation can be used to affect rotation along one axis without affecting the other two axes, hence the craft can be manoeuvred in three dimensions using only the differential angular velocities of the four fixed pitch rotors. Vertical acceleration is dependent on the collective thrust of the four rotors. Horizontal forces are introduced by the vertical thrust as the orientation of the craft changes.

2.1.3 Considerations Regarding Craft Dynamics

The aerodynamic performance of the craft is dependent on various internal and external factors. These aerodynamic considerations are discussed in Appendix B. The rotational inertia is dependent not only on the dimensions and weight of the craft, but the way in which the weight is distributed, hence the position of the centre of mass. Changes in inertia affects the agility of the craft, while changes in atmospheric conditions can affect the stability and accuracy of control. A low inertia will increase agility, but also increases susceptibility to external disturbances such as wind.

For the sake of improved stability it is recommended that the craft be designed symmetrically. This serves to simplify modelling through symmetric inertia and provide an approximate position for the Centre of Mass (CoM) at the geometric centre of the craft. By placing the CoM at the geometric centre, below the height of the rotors, the craft approached a more stable configuration, reducing the amount of actuation needed to

stabilize the craft. The CoM is primarily dependent on the location of the batteries, motors and airframe, since they are the heaviest components. Since the airframe and motors are located symmetrically over the craft, the placement of the batteries should be near the desired CoM. By increasing the distance between the motors and the Centre of Mass (CoM), the rotational inertia is increase, but so is the rotational torque produced by the motor. It can therefore be said that an optimal geometric design can be derived for a quad-rotor to provide either agility or stability, or a good balance.

Formulating Rigid Body Kinetics

Kinematics is the study of the motion of a body without regard to the forces that cause the motion. The craft kinematics consists of two main factors, translational and angular motion. Both these are three dimensional, resulting in a full six Degrees of Freedom (DOF) model describing the motion of the craft. Applying a second order differential equation to each of these degrees of freedom provides a sufficiently detailed model of the motion of the craft. This includes the translational position, velocity and acceleration as well as the angular orientation, velocity and acceleration. Combined these six parameters and related functions for the system of equations describe the motion of the craft. A detailed model for the kinematics using the Euler model is derived in Appendix A.1.

Dynamics is the study of a craft's motion with regard to the forces that cause that motion. By using the Euler model for kinematics and combining it with Newtons laws of motion, a combined Newton-Euler model is formulated in Appendix A.2. The dynamics can be described in six degrees of freedom in relation to the various forces that act on the system [7]. The composition of the 6 DOF translation and rotation vectors, are three motion parameters in a Cartesian coordinate system, i.e. forward (x), lateral (y) and vertical (z). Whilst orientation is defined by the angular motion parameters, i.e. roll (ϕ), pitch (θ) and yaw (ψ). There exists a cross correlation between the translational motion and the orientation of the craft. In summary, the forward motion is dependent on the pitch (θ), the lateral motion on the roll (ϕ) and the heading of the

craft is dependent on the yaw (ψ).

The reference frames are based on the right handed orientation and rotation.

2.1.4 Flight Control

The detailed model of the flight dynamics is discussed in section 2.2, however an elementary understanding is required to complete the functional analysis. Firstly it is necessary to understand that the functional analysis constitutes multiple levels. At the top is the operational level, next the functional and finally the technical level. The operation level constitutes the user interactions and the technical level the physical actions to perform the various manoeuvres, while the functional level consists of the various processes to link the operational (user) to the technical (dynamics) level.

The principles behind the differential equations describing the flight dynamics are briefly discussed in the previous section. Figure 2.4 shows the various vectors reflecting the forces, and Figure 2.5 shows the parameters used to describe the motion (dynamics) of the craft.

Thrust Dynamics

Modelling the exact thrust dynamics can be complex. As discussed in Appendix B, the thrust is dependent not only on the exact geometry and aerodynamics of the rotor blades, but also on the atmospheric conditions including pressure, density, air flow, etc. A detailed model of the motors and related control circuitry is also needed to parametrize the torque characteristics of the motors. It is also necessary to know the inertial characteristics of the rotors, to model the actuation delays. Hence defining a mathematical model for the thrust dynamics can prove troublesome.

For the sake of simplification, a sequence of experimental set-ups can be used to derive a single model to describe the actuation characteristics that includes the speed

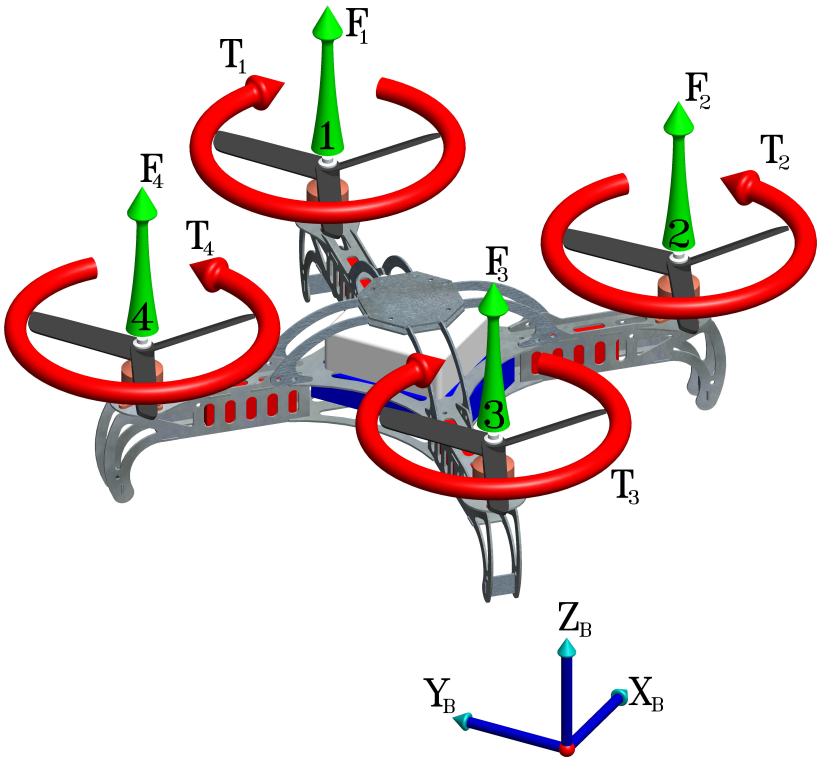


Figure 2.4: Forces used to control flight dynamics

controller, motor and rotor characteristics. This model measures the relationship between the Pulse Width Modulation (PWM) signal input of the speed controller to the resulting angular velocity of the rotor. The angular velocity is then related to the thrust and torque produced by the motor. Finally the results are expressed as a statistical regression model for each of the ratios. Since the parameters are so greatly affected by atmospheric conditions, the accuracy of the models are not critical to the development process, as the controller needs to be able to operate during these normal fluctuations.

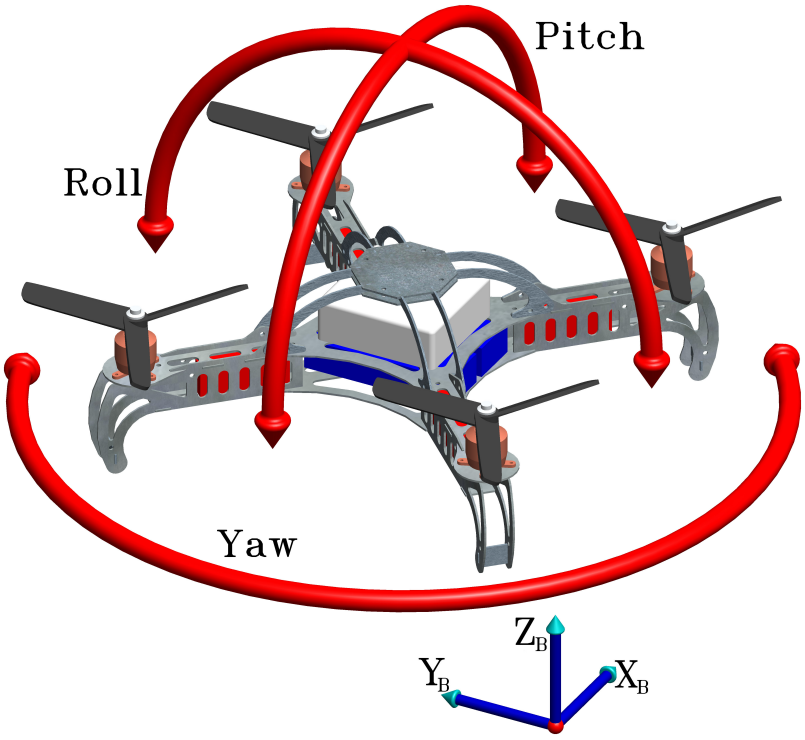


Figure 2.5: Axis used to define flight dynamics

Functional analysis

Operational level: At the operational level, the user inputs are interpreted to the required motion as expressed by the dynamic equations. Alternatively stated, the process is the conversion of the user instructions to the actionable motions.

Functional level: The Functional level is responsible for the translation of the motions to the required orientations that will result in the requested motion.

Technical level: The Technical level converts the required orientation to rotor speed combinations that will turn the craft to the required orientation.

Figure 2.6 shows the basic functional flow as described above, indicating only the major processes, not the detailed inner workings. Where Ω_i indicates the rotor speed of motor i.

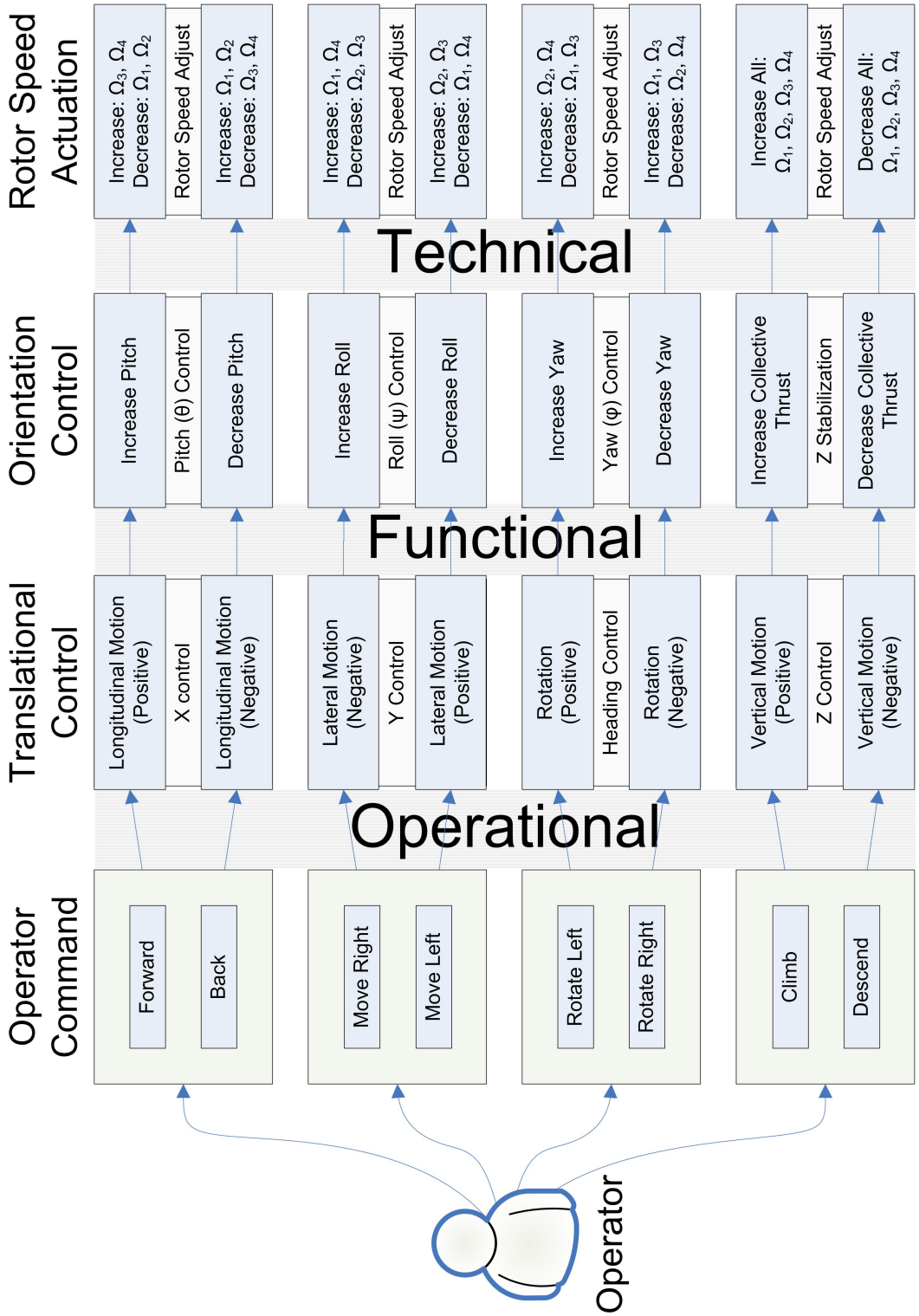


Figure 2.6: Basic functional flow

Control Framework

Based on the discussion in the previous section, the distinction between translational and angular motion results in the need for distinct controllers for both. As stated in chapter 1, the translational motion controller is referred to as the trajectory controller and the rotational motion controller as the orientation controller. The integration of these two controllers is shown in Figure 2.7, clearly indicating that the output of the trajectory controller is the required orientation to achieve the desired motion.

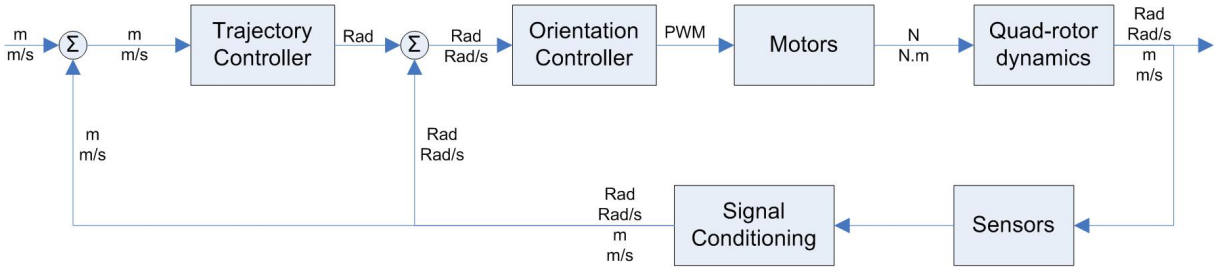


Figure 2.7: Basic control flow

2.2 Model

There are various modelling techniques available to model the dynamics of the craft. Each has various advantages, but also increases in complexity. The easiest of the methods is the use of Euler angles. Barclay [8] did a comparative study between various modelling techniques.

Euler angles can be used to represent the dynamics of the craft in a single reference frame, as discussed in previous works [6]. This is done by describing the forces as dependent on the orientation of the craft. Although this is sufficiently accurate, it does present a few problems regarding the presence of various interpretation errors and effects such as gimbal lock. It also does not account for the difference in perception between an external observer and one on-board the craft, i.e. the sensors. Salih et al. [9] shows how to derive a dynamic model using Euler angles. Similar uses of this modelling techniques can be seen in the works of Erginer et al. [10], Goel et al. [11], Altuğ et al. [12], [13], Petersen et al. [14]

The Newton-Euler method is slightly more complex than the simple Euler angle model, but simplifies the addition of other forces that act on the craft. It expresses the dynamics in matrix form as a sum of forces that act on the craft. This makes it easier to account for forces such as drag, gyroscopic effects, gravity and wind, etc. Kivrak [15], Wu [16], Balas [17], Bouabdallah [18] and Miller [19] show implementations of the Newton-Euler method in a single reference frame.

The addition of multiple reference frames allow for the proper interpretation of sensor data in a rotating reference frame. The use of this rotating reference frame does however make it necessary to account for the Coriolis effect. Although through proper interpretation the effect of gimbal lock can be avoided, it is not inherently removed as in the quaternion method. Implementation of the Newton-Euler method in multiple reference frames can be seen in the works of Altuğ et al. [20], Hamel et al. [21] and Bresciani [7].

The quaternion approach, which is similar to complex algebra, but in four dimensions, has a few advantages over the standard Euler angle based models. Although it is more complex, the addition of an extra redundant axis allows for the prevention of effects such as gimbal lock, since there is always three axes even when two axes align. Tayebi and MCGilvray [22] utilized the quaternion method to model the quad-rotor dynamics.

Another method that can be used is Lagrangian mechanics, which combines the conservation of momentum with the conservation of energy. Although this method is mathematically more complex, it simplifies a complex system based on constraints rather than a collection of equations. The Lagrangian method requires that all the losses be modelled in order to gain an accurate model. Salazar-Cruz [23] used the Lagrangian method to model the quad-rotor dynamics.

This study is structured in such a manner as to aid a beginner in the field, the model is based on Euler angles since they are easier to understand. The model discussed in Section 2.2.2 was derived from previous work [6] and converted to matrix form, accounting for multiple reference frames, using the work of Tommaso Bresciani [7]. Some modifications were made to account for the change in orientation. Additional notes are made to explain how the model can be expanded further and the assumptions are described in greater detail.

2.2.1 Model assumptions

Before the model can be derived the assumptions needed to simplify the mathematics and the conditions under which they hold true need to be documented. These assumptions are now listed and discussed. Note that the reference frames are based on the right-handed coordinate system for translational motion and that a left-hand rotation convention is used for angular motion, i.e. that rotation is counter-clockwise when looking in the positive direction of the axis.

Fixed body: The body of the quad-rotor is considered to be rigid, having no deformation or flex, for all forces applied. By omitting this deformation, the inertia matrix is simplified to a time-invariant matrix. The structural design and frame material needs to be chosen to approximate this property. The assumption of a fixed body also simplifies the translation of sensor data to the body reference frame.

Centre of Mass (CoM): The CoM is assumed to coincide with the origin $\overline{O}_B$ of the body reference frame that is also the centre of the fixed body frame. It is also assumed that the principle axis of inertia coincide with the axis of the *B-frame*, simplifying the inertia matrix to a purely diagonal matrix. To ensure this assumption is a valid approximation, the weight of the craft has to be distributed evenly around the symmetric centre of the body.

Body symmetry: The assumption that the centre of mass and the principle axis of inertia coincide with the body reference frame, allows for another simplification: the use of body symmetry. By assuming body symmetry, the drag and thrust characteristics of each motor is assumed to be the same. The position of each motor relative to the principle axis of inertia is assumed to be the same and hence the torque produced by each motor is equal. The rotational inertia around the pitch and roll axes are also equal. This reduces the number of parameters as it is not necessary to define the position of each rotor individually.

Vibration: The presence of vibration is not modelled due to the rigid body assumption, which means any vibration translates to the same motion throughout the body. The primary source of vibration is the motors and due to the high rotational speed, the frequency of the vibration is high enough to be filtered by a low pass filter without removing relevant sensor data.

Drag and wind: As the primary objective is to achieve stable hover in indoor conditions, the modelling of wind and drag is unnecessary. Drag is proportional to the airspeed of the aircraft, hence at hover where the airspeed is zero or at least very small, the drag is negligible.

2.2.2 The Newton-Euler Model

The model used to describe the dynamics of the quad-rotor is based on the generic six DOF Newton-Euler model for a rigid body object, as is explained in Appendix A. For the sake of readability, a quick review is in order.

There are two reference frames defined in the model, the earth reference frame (*E-frame*) and the body reference frame (*B-frame*). Equation (2.1) describes the kinematics of a generic 6-DOF rigid body object.

$$\dot{\overline{\xi}} = \mathbf{J}_{\Theta}\overline{\nu} \tag{2.1}$$

The generalized velocity vector $\dot{\xi}[+]$ in reference to the *E-frame*; can be derived from the generalized velocity vector $\nu[+]$ in reference to the *B-frame* by using the generalized conversion matrix $\mathbf{J}_{\Theta}[-]$ as defined in (2.2), with $\mathbf{R}_{\Theta}$ the rotation and $\mathbf{T}_{\Theta}$ the transfer matrix.

$$\mathbf{J}_{\Theta} = \begin{bmatrix} \mathbf{R}_{\Theta} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{T}_{\Theta} \end{bmatrix} \tag{2.2}$$

The generalized position vector $\xi[+]$ in reference to the *E-frame* constitutes translational $\Gamma^{E}[m]$ and angular $\Theta^{E}[rad]$ position components as indicated by (2.3).

$$\overline{\xi} = \begin{bmatrix} \overline{\Gamma}^{E} & \overline{\Theta}^{E} \end{bmatrix}^{T} = \begin{bmatrix} X & Y & Z & \phi & \theta & \psi \end{bmatrix}^{T} \tag{2.3}$$

Similarly the generalized velocity vector $\nu[+]$ in reference to the *B-frame* is composed of translational $\mathbf{V}^{B}[m.s^{-1}]$ and angular $\omega^{B}[rad.s^{-1}]$ velocity components as indicated in equation (2.4).

$$\overline{\nu} = \begin{bmatrix} \overline{\mathbf{V}}^{B} & \overline{\omega}^{B} \end{bmatrix}^{T} = \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^{T} \tag{2.4}$$

Figure 2.8 shows the relationship between the two reference frames as discussed in Appendix A. Note that the reference frames are based on the right handed orientation and rotation.

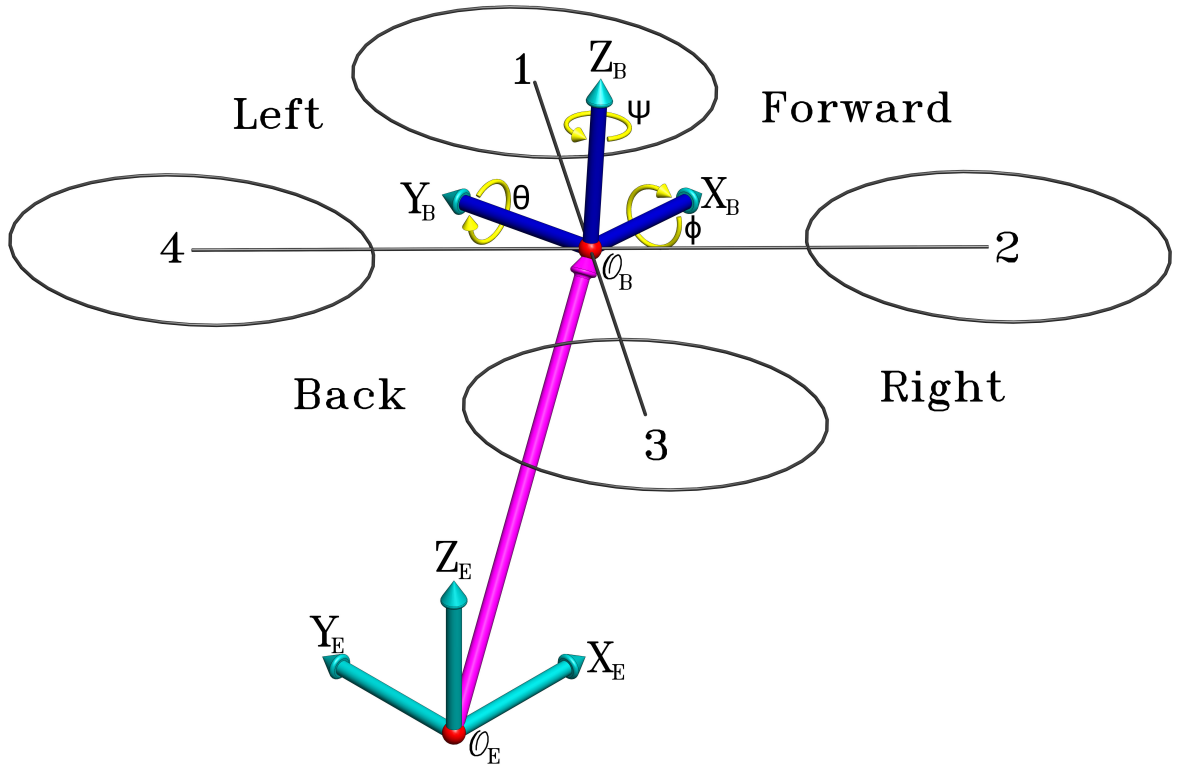


Figure 2.8: Reference frames

The dynamics of a generic 6-DOF rigid body is summarized in (2.5) in reference to the *B-frame*.

$$\begin{bmatrix} m\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \dot{\overline{V}}^B \\ \dot{\overline{\omega}}^B \end{bmatrix} + \begin{bmatrix} \overline{\omega}^B \times \left(m\overline{V}^B\right) \\ \overline{\omega}^B \times \left(\mathbf{I}\overline{\omega}^B\right) \end{bmatrix} = \begin{bmatrix} \overline{F}^B \\ \overline{\tau}^B \end{bmatrix} \tag{2.5}$$

In this equation $m[kg]$ is the mass of the body, $I[N.m.s^2]$ is the body inertia matrix, $F^B[N]$ is the net forces and $\tau^B[N.m]$ the net torques, all in reference to the *B-frame*. This entire model is generic for the 6-DOF dynamics of any rigid body; hence the forces vector is expanded to contain specific information to describe the quad-rotor dynamics in detail.

A generalized force vector Λ is defined in (2.6), representing the net forces and torques.

$$\overline{\Lambda} = \begin{bmatrix} \overline{F}^B & \overline{\tau}^B \end{bmatrix}^T = \begin{bmatrix} F_x & F_y & F_z & \tau_x & \tau_y & \tau_z \end{bmatrix}^T \tag{2.6}$$

Hence (2.5) can be simplified to matrix form as shown in (2.7).

$$\mathbf{M}_B\dot{\overline{\nu}} + \mathbf{C}_B\left(\overline{\nu}\right)\overline{\nu} = \overline{\Lambda} \tag{2.7}$$

In this equation $\dot{\nu}$ is the generalized acceleration in reference to the *B-frame*, $\mathbf{M}_B$ is the system inertia matrix (a constant diagonal matrix due to the rigid body assumptions) and $\mathbf{C}_B(\overline{\nu})$ is the Coriolis-centripetal matrix, all in reference to the *B-frame*. Note that the Coriolis-centripetal matrix $\mathbf{C}_B(\overline{\nu})$ is dependent on the generalized velocity vector $\overline{\nu}$. Equation (2.8) shows the system inertia matrix $\mathbf{M}_B$.

$$\mathbf{M}_B = \begin{bmatrix} m\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{XX} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{YY} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{ZZ} \end{bmatrix} \tag{2.8}$$

Equation (2.9) shows the Coriolis-centripetal matrix $\mathbf{C}_B(\overline{\nu})$. Note that the first three columns are all zero since the Coriolis Effect is only present to the angular motion, as it is an observed deflection in a rotating reference frame and not a translational force acting on the object.

$$\mathbf{C}_B(\overline{\nu}) = \begin{bmatrix} \mathbf{0}_{3\times3} & -m\mathbf{S}(\overline{V}^B) \\ \mathbf{0}_{3\times3} & -\mathbf{S}(\mathbf{I}\overline{\omega}^B) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & mw & -mv \\ 0 & 0 & 0 & -mw & 0 & mu \\ 0 & 0 & 0 & mv & -mu & 0 \\ 0 & 0 & 0 & 0 & I_{ZZ}r & -I_{YY}q \\ 0 & 0 & 0 & -I_{ZZ}r & 0 & I_{XX}p \\ 0 & 0 & 0 & I_{YY}q & -I_{XX}p & 0 \end{bmatrix} \tag{2.9}$$

The above equation uses the notation $\mathbf{S}(\cdots)[+]$ to designate the skew-symmetric matrix as explained in (2.10).

$$\mathbf{S}(\bar{k}) = -\mathbf{S}^T(\bar{k}) = \begin{bmatrix} 0 & -k_3 & k_2 \\ k_3 & 0 & -k_1 \\ -k_2 & k_1 & 0 \end{bmatrix} \qquad \bar{k} = \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} \tag{2.10}$$

Now given the generic form of the equation shown in (2.7), with the forces vector Λ describing the dynamics of the quad-rotor, it is necessary to identify the various forces that act on the quad-rotor to derive the specific dynamics of the craft.

2.2.3 Quad-rotor Forces

Gravity

The first force is that of gravity, dependent on the mass of the object and gravitational acceleration $g[m.s^{-2}]$. Gravity is constant in the earth reference frame (*E-frame*), but has to be converted to the *B-frame*. The gravitational vector $\overline{G}_B(\overline{\xi})$ is shown in (2.11).

$$\overline{G}_B(\overline{\xi}) = \begin{bmatrix} \overline{F}_G^B \\ \mathbf{0}_{3\times 1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_\Theta^{-1}\overline{F}_G^E \\ \mathbf{0}_{3\times 1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_\Theta^{-1} \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \\ \mathbf{0}_{3\times 1} \end{bmatrix} = \begin{bmatrix} mgs_\theta \\ -mgc_\theta s_\phi \\ -mgc_\theta c_\phi \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.11}$$

In this equation, $\overline{F}_G[N]$ is the gravitation force vector in regard to each reference frame. Since $\mathbf{R}_\Theta$ is an orthogonal normalized matrix, the inverse $\mathbf{R}_\Theta^{-1}$ is equal to the transposed $\mathbf{R}_\Theta^T$, shown in (2.12).

$$c_k = \cos k, \quad s_k = \sin k, \quad t_k = \tan k$$

$$\begin{aligned} \mathbf{R}_\Theta^{-1} &= \mathbf{R}_\Theta^T \\ &= \begin{bmatrix} c_\psi c_\theta & s_\psi c_\theta & -s_\theta \\ -s_\psi c_\phi + c_\psi s_\theta s_\phi & c_\psi c_\phi + s_\psi s_\theta s_\phi & c_\theta s_\phi \\ s_\psi s_\phi + c_\psi s_\theta c_\phi & -c_\psi s_\phi + s_\psi s_\theta c_\phi & c_\theta c_\phi \end{bmatrix} \end{aligned} \tag{2.12}$$

Gyroscopic effect

The second force acting on the craft is a result of the gyroscopic effect produced by the rotation of the rotors. To summarize the concept behind the gyroscopic effect, consider the example of when a rotating body has a rotation around the z-axis and then experiences a rotation around the x-axis, it will result in a rotation around the y-axis. The

torque produced by the gyroscopic effect is proportional to the angular velocity of the primary rotation. In the case of the quad-rotor, only the gyroscopic effect taken into account is due to the propeller rotation, as the angular rotation of the body is limited to the point of having no noticeable effect. Since two of the rotors turn clockwise and two counter clockwise, the gyroscopic effect is only present if the algebraic sum of their angular velocities is not equal to zero. The gyroscopic rotor matrix $\mathbf{O}_B(\overline{v})[+]$ is shown in (2.13).

$$\begin{aligned}
\mathbf{O}_B(\overline{v})\overline{\Omega} &= \begin{bmatrix} \mathbf{0}_{3\times 1} \\ -\Sigma_{k=1}^{4} J_{TP} \left(\omega^B \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right) (-1)^k \Omega_k \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3\times 1} \\ J_{TP} \begin{bmatrix} -q \\ p \\ 0 \end{bmatrix} \Omega_T \end{bmatrix} \\
&= J_{TP} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ q & -q & q & -q \\ -p & p & -p & p \\ 0 & 0 & 0 & 0 \end{bmatrix} \overline{\Omega}
\end{aligned} \tag{2.13}$$

$J_T P$ is the total rotation moment of inertia as calculated around the rotor axes. From the equation it is clear that the gyroscopic effect has no direct impact on the translational motion of the craft.

The overall propeller speed $\Omega_T[rad.s^{-1}]$ used in (2.13) is defined by the propeller speed vector $\overline{\Omega}[rad.s^{-1}]$ in (2.14), where $\Omega_i[rad.s^{-1}]$ is the speed of propeller i.

$$\Omega_T = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \qquad \overline{\Omega} = \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{bmatrix} \tag{2.14}$$

Movement forces

The third contribution is the movement forces produced by the rotors. This constitutes the translational forces and rotational torques produced by the rotors to control the craft. Based on the aerodynamic principles of propeller dynamics, both the thrusts and torques produced are proportional to the square of the propeller speed. Subsequently a movement matrix $E_B[+]$ can be compiled to parametrize this relation and then be multiplied with the square of the rotor speeds $\overline{\Omega}^2$ to get the movement vector $U_B(\overline{\Omega})[+]$ as shown in (2.15). A more detailed explanation of the various aerodynamic contributions can be found in Appendix B.

$$\mathbf{U}_B(\overline{\Omega}) = \mathbf{E}_B\overline{\Omega}^2 = \begin{bmatrix} 0 \\ 0 \\ U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ b\left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right) \\ bl\left(\Omega_1^2 - \Omega_2^2 - \Omega_3^2 + \Omega_4^2\right) \\ bl\left(-\Omega_1^2 - \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right) \\ d\left(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2\right) \end{bmatrix} \tag{2.15}$$

In this equation the parameter $l[m]$ is the perpendicular distance from the given axis to the given motor. Since the frame is squarely symmetric, all the distances are the same and can be represented by a single value. The aerodynamic parameters for thrust $b[N.s^2]$ and drag $d[N.m.s^2]$ are used to relate the force and torque to the speed of the rotors. For the sake of simplification the model for the acceleration of the rotors is kept as a linear system, neglecting the $\dot{\overline{\Omega}}$ components. Hence the order of the system is kept low to simplify the derivation of the control algorithms.

Consequently the constant matrix E_B can be divined as shown in (2.16).

$$\mathbf{E}_B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ b & b & b & b \\ bl & -bl & -bl & bl \\ -bl & -bl & bl & bl \\ -d & d & -d & d \end{bmatrix} \tag{2.16}$$

Now (2.7) can be re-evaluated to express these forces as part of the model. This is expressed in (2.17).

$$\mathbf{M}_B\dot{\overline{\nu}} + \mathbf{C}_B(\overline{\nu})\overline{\nu} = \overline{G}_B(\overline{\xi}) + \mathbf{O}_B(\overline{\nu})\overline{\Omega} + \mathbf{E}_B\overline{\Omega}^2 \tag{2.17}$$

By rearranging (2.17), the generalized acceleration $\dot{\overline{\nu}}$ can be expressed in reference to the *B-frame* as in (2.18).

$$\dot{\overline{\nu}} = \mathbf{M}_B^{-1}\left(-\mathbf{C}_B(\overline{\nu})\overline{\nu} + \overline{G}_B(\overline{\xi}) + \mathbf{O}_B(\overline{\nu})\overline{\Omega} + \mathbf{E}_B\overline{\Omega}^2\right) \tag{2.18}$$

This model can now be expressed as a system of equations as is done in (2.19).

$$\begin{aligned}
\dot{u} &= (vr - wq) + gs_\theta \\
\dot{v} &= (wp - ur) - gc_\theta s_\phi \\
\dot{w} &= (uq - vp) - gc_\theta c_\phi + \frac{U_1}{m} \\
\dot{p} &= \frac{I_{YY} - I_{ZZ}}{I_{XX}}qr - \frac{J_{TP}}{I_{XX}}q\Omega_T + \frac{U_2}{I_{XX}} \\
\dot{q} &= \frac{I_{ZZ} - I_{XX}}{I_{YY}}pr + \frac{J_{TP}}{I_{YY}}p\Omega_T + \frac{U_3}{I_{YY}} \\
\dot{r} &= \frac{I_{XX} - I_{YY}}{I_{ZZ}}pq + \frac{U_4}{I_{ZZ}}
\end{aligned} \tag{2.19}$$

U_i is given in the subsystem of control forces, dependent on the various input rotor speeds, given in (2.20).

$$\begin{aligned}
U_1 &= b\left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right) \\
U_2 &= lb\left(\Omega_1^2 - \Omega_2^2 - \Omega_3^2 + \Omega_4^2\right) \\
U_3 &= lb\left(-\Omega_1^2 - \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right) \\
U_4 &= d\left(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2\right) \\
\Omega_T &= -\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2
\end{aligned} \tag{2.20}$$

Further expansion of the model is possible to account for other forces that affect the motion of the craft. One such force is drag, a force produced by the motion of the craft through the atmosphere due to air resistance. This force is dependent on the velocity of the craft and is easiest to define in reference to the body frame. This force

is left unaccounted for as it is difficult to calculate the parameters and because of the low operating velocities in this study, it is not considered a major contributor. If the parameters for drag are known, it is simple to model the force of wind by expressing the force of wind as changes in airspeed, hence using the same process as drag. This is an accurate approach as airspeed is considered different from ground speed due to the presence of wind. The next step would be to model the variation in wind by some meteorological model where the direction and force of the wind is changed using a first or second order function (to prevent instantaneous changes during simulation). Wind is not considered as relevant to the scope of this study. It may also be relevant to account for the shift in the centre of mass depending on the type of payload used. Other considerations are discussed in Appendix B.

2.2.4 The Hybrid Reference Frame

The above equations are all defined in reference to the *B-frame* and although they simplify the definition of the control inputs, they fail to describe navigational considerations since the *B-frame* is coincident with the craft body and hence not stationary. The position is defined in the formulation of the *B-frame* origin. Translating the model to the *E-frame* would provide the necessary navigational information, but would not represent the reference frame of the sensors and controllers. To resolve this problem a hybrid reference frame (*H-frame*) is defined, using the translational parameters from the *E-frame* for navigation and the angular parameters from the *B-frame*. Doing this allows for navigational control using an earth reference sensor such as GPS and orientation control using body reference sensors such as gyroscopes. This greatly simplifies the implementation of control without the need to convert sensor data between the reference frames.

Consider equation (2.21) where $\overline{\zeta}[+]$ represents the general velocity vector.

$$\overline{\zeta} = \begin{bmatrix} \dot{\overline{\Gamma}}^E & \overline{\omega}^B \end{bmatrix}^T = \begin{bmatrix} \dot{X} & \dot{Y} & \dot{Z} & p & q & r \end{bmatrix}^T \tag{2.21}$$

Equation (2.22) expresses the dynamics of the quad-rotor in reference to the *H-frame*.

$$\dot{\overline{\zeta}} = \mathbf{M}_H^{-1} \left(-\mathbf{C}_H(\overline{\zeta})\overline{\zeta} + \overline{G}_H + \mathbf{O}_H(\overline{\zeta})\overline{\Omega} + \mathbf{E}_H(\overline{\xi})\overline{\Omega}^2 \right) \tag{2.22}$$

In this equation the quad-rotor generalized acceleration is $\dot{\overline{\zeta}}[+]$, i.e. in reference to the *H-frame*. It is now necessary to convert each of the parameter matrices to the *H-frame*. Note that only the translational components have to be converted as the angular components are still referenced to the *B-frame*.

The system inertia matrix $\mathbf{M}_H[+]$ remains unaffected, shown in (2.23).

$$\mathbf{M}_H = \mathbf{M}_B = \begin{bmatrix} m\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{XX} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{YY} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{ZZ} \end{bmatrix} \tag{2.23}$$

The Coriolis-centripetal matrix referenced to the *H-frame* $\mathbf{C}_H(\overline{\zeta})$ is different from $\mathbf{C}_B(\overline{\nu})$ since the translational components no longer observe the deflection of the rotating *B-frame*. Hence the translational components of $\mathbf{C}_H(\overline{\zeta})$ is now zero as shown in (2.24).

$$\mathbf{C}_H(\overline{\zeta}) = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & -\mathbf{S}(\mathbf{I}\overline{\omega}^B) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{ZZ}r & -I_{YY}q \\ 0 & 0 & 0 & -I_{ZZ}r & 0 & I_{XX}p \\ 0 & 0 & 0 & I_{YY}q & -I_{XX}p & 0 \end{bmatrix} \tag{2.24}$$

The gravitation vector in (2.25) is now simplified as it remains constant in reference to the *H-frame*.

$$\overline{G}_H = \begin{bmatrix} \overline{F}_G^E \\ \mathbf{0}_{3\times1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -mg \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.25}$$

Since the gyroscopic effects are only relevant to the angular components it is not af-

fected by the change to the *H-frame*. This is expressed in (2.26).

$$\mathbf{O}_H(\overline{\zeta})\overline{\Omega} = \mathbf{O}_B(\overline{v})\overline{\Omega} = \begin{bmatrix} \mathbf{0}_{3\times 1} \\ J_{TP} \begin{bmatrix} -q \\ p \\ 0 \end{bmatrix} \Omega_T \end{bmatrix} = J_{TP} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ q & -q & q & -q \\ -p & p & -p & p \\ 0 & 0 & 0 & 0 \end{bmatrix} \overline{\Omega} \tag{2.26}$$

There is however an increase in complexity when converting the movement matrix $\mathbf{E}_B$ to the *H-frame*. It might have been easier to define the control forces in reference to the *B-frame*, but relating the effect of these forces to the trajectory of the craft can prove problematic. Converting the $\mathbf{E}_B$ matrix to the *H-frame* can be done by converting the translational components by means of the rotation matrix $\mathbf{R}_\Theta$. The new movement matrix $\mathbf{E}_H(\overline{\xi})$ is shown in (2.27) and (2.28).

$$\begin{aligned} \mathbf{E}_H(\overline{\xi})\overline{\Omega}^2 &= \begin{bmatrix} \mathbf{R}_\Theta & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{I}_{3\times 3} \end{bmatrix} \mathbf{E}_B\overline{\Omega}^2 = \begin{bmatrix} \left(s_\psi s_\phi + c_\psi s_\theta c_\phi\right) U_1 \\ \left(-c_\psi s_\phi + s_\psi s_\theta c_\phi\right) U_1 \\ \left(c_\theta c_\phi\right) U_1 \\ U_2 \\ U_3 \\ U4 \end{bmatrix} \\ &= \begin{bmatrix} b\left(s_\psi s_\phi + c_\psi s_\theta c_\phi\right)\left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right) \\ b\left(-c_\psi s_\phi + s_\psi s_\theta c_\phi\right)\left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right) \\ b\left(c_\theta c_\phi\right)\left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right) \\ bl\left(\Omega_1^2 - \Omega_2^2 - \Omega_3^2 + \Omega_4^2\right) \\ bl\left(-\Omega_1^2 - \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right) \\ d\left(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2\right) \end{bmatrix} \end{aligned} \tag{2.27}$$

Note that U_1 now affects all three translational movement directions due to the rotation of the body, hence $\mathbf{E}_H(\overline{\xi})$ is now dependent on the orientation of the craft as shown in

(2.28).

$$\begin{aligned}
\Delta_1 &= \left(s_\psi s_\phi + c_\psi s_\theta c_\phi\right) \\
\Delta_2 &= \left(-c_\psi s_\phi + s_\psi s_\theta c_\phi\right) \\
\Delta_3 &= \left(c_\theta c_\phi\right) \\
\mathbf{E}_H(\overline{\xi}) &= \begin{bmatrix} b\Delta_1 & b\Delta_1 & b\Delta_1 & b\Delta_1 \\ b\Delta_2 & b\Delta_2 & b\Delta_2 & b\Delta_2 \\ b\Delta_3 & b\Delta_3 & b\Delta_3 & b\Delta_3 \\ bl & -bl & -bl & bl \\ -bl & -bl & bl & bl \\ -d & d & -d & d \end{bmatrix}
\end{aligned} \tag{2.28}$$

The system of equations for the model in reference to the *H-frame* is now shown in (2.29).

$$\begin{aligned}
\ddot{X} &= \left(\sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi\right)\frac{U_1}{m} \\
\ddot{Y} &= \left(-\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi\right)\frac{U_1}{m} \\
\ddot{Z} &= -g + \left(\cos\theta\cos\phi\right)\frac{U_1}{m} \\
\dot{p} &= \frac{I_{YY} - I_{ZZ}}{I_{XX}}qr - \frac{J_{TP}}{I_{XX}}q\Omega_T + \frac{U_2}{I_{XX}} \\
\dot{q} &= \frac{I_{ZZ} - I_{XX}}{I_{YY}}pr + \frac{J_{TP}}{I_{YY}}p\Omega_T + \frac{U_3}{I_{YY}} \\
\dot{r} &= \frac{I_{XX} - I_{YY}}{I_{ZZ}}pq + \frac{U_4}{I_{ZZ}}
\end{aligned} \tag{2.29}$$

$$\begin{aligned}
U_1 &= b\left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right) \\
U_2 &= lb\left(\Omega_1^2 - \Omega_2^2 - \Omega_3^2 + \Omega_4^2\right) \\
U_3 &= lb\left(-\Omega_1^2 - \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right) \\
U_4 &= d\left(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2\right) \\
\Omega_T &= -\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2
\end{aligned} \tag{2.30}$$

Once again the symbols U_i and Ω_T are used as defined in equation (2.30) to present the rotor speeds. These equations in (2.29) are the bases for the derivation of the control algorithms, where (2.30) are the inputs used to control the system.

2.3 Parametrization

This section covers the methodology used to calculate the parameters needed to populate the model. Firstly, a summary follows of the parameters needed.

2.3.1 Parameter Summary

Consider the generalized dynamics of the quad-rotor as observed in the hybrid reference frame.

$$\dot{\overline{\zeta}} = \mathbf{M}_H^{-1}\left(-\mathbf{C}_H(\overline{\zeta})\overline{\zeta} + \overline{G}_H + \mathbf{O}_H(\overline{\zeta})\overline{\Omega} + \mathbf{E}_H(\overline{\xi})\overline{\Omega}^2\right) \tag{2.31}$$

The parameters include:

Inertia Matrix ($\mathbf{M}_H$) *& Inverse* ($\mathbf{M}_H^{-1}$)

As a result of the assumed symmetry of the craft, the inertia matrix is simply a diagonal matrix consisting of the mass (m) and rotational inertia around each axis (I_{XX}, I_{YY}, I_{ZZ}). Once the inertia matrix has been determined, the inverse inertia matrix can be calculated. This is simplified due to the diagonal nature of the matrix.

Gravity Matrix ($\overline{G}_H$)

Gravity is only observed in the translational motion and not in the rotational motion; hence in the hybrid reference frame gravity is expressed as a constant downward acceleration of $g = 9.807 m.s^{-2}$ along the z axis.

Gyroscopic effect Matrix ($\mathbf{O}_H(\overline{\zeta})$)

The gyroscopic effect is only relevant to the rotational motion. It is dependent on the current rotational velocity and rotor speeds, with the only parameter to calculate being the net rotational inertia (J_{TP}) around the rotor axis.

Coriolis-centripetal Matrix ($\mathbf{C}_H(\overline{\zeta})$)

The Coriolis-centripetal matrix is dependent on the current rotational velocity and the inertia matrix.

Movement Forces Matrix ($\mathbf{E}_H(\overline{\xi})$)

The movement forces matrix has multiple parameters and is dependent on the general velocity vector. The translational forces are dependent on the current orientation, determined using the rotation matrix and the net trust. The rotational torques are dependent on the differential thrust and the symmetric distance (l) to the axis of rotation. The thrust is proportional to the rotor speed and thrust constant (b) of the given propeller. The Yaw rotational torque is dependent on the relationship between the rotor speed and the propeller drag constant (d).

Depending on the parameter, it may be easier to calculate or measure the value associated with it. The parameters to be determined theoretically include rotational inertia, net rotor inertia, the inverse inertia matrix, the Coriolis-centripetal matrix, the symmetric distance to axis of rotation and gravity. The mass is easy to measure whilst the drag and thrust constants are difficult to calculate and hence must be determined through some experimental set-up.

2.3.2 Calculated Parameters

Firstly, gravity is considered as a constant $g = 9.807 m.s^{-2}$. Rotational inertia and the symmetric distance to the axis of rotation is determined using the mass properties tools of Solidworks. The results are shown in table 2.1.

Table 2.1: Calculated Parameters

Parameter	Value	Unit
l	0.18	m
m	3.33	kg
I_{XX}	6.93×10^{-3}	$N.m.s^2$
I_{YY}	6.93×10^{-3}	$N.m.s^2$
I_{ZZ}	13.3×10^{-3}	$N.m.s^2$
J_{TP}	22.5×10^{-6}	$N.m.s^2$

2.3.3 Measured Parameters

The drag and thrust constants are difficult to calculate and hence must be determined through some experimental set-up. Figure 2.9 shows the experimental set-up to measure the thrust produced by the rotor and motor combination. The motor is attached to the end of a bar, whilst the other end is attached to a stationary pivot point. The thrust is directed upward so that the force directed downward into a scale used to measure the force. The force is measure in kilograms (kg) and needs to be converted to newtons (N), using a generalized ratio of $1kg \approx 10N$. Note that if the rotor is inverted, the rotational directional has to be inverted as well to ensure that the rotor turns in its intended direction, to maximize the rotors aerodynamic performance. If the airflow over the rotor is not in the direction intended, the efficiency is greatly reduced.



Figure 2.9: Experimental set-up to measure thrust characteristics

As stated before, in order to simplify the model, the actuation model is measured to include the characteristics of the speed controllers, motors and rotors into a single look-up table. The rotor speed is measured in relation to the speed controller PWM input, whilst the thrust and torque is measured in relation to the rotor speed. The results of the experimental set-up is shown in figures 2.10 through 2.12.

The relationship shown in Figure 2.10, can be expressed by the equation shown in (2.32), with $Duty\ \ cycle\ \ (\%)$ the PWM duty cycle percentage.

$$\Omega = \eta\sqrt{Duty\ \ cycle\ \ (\%)} \tag{2.32}$$

There is little variation in the measurement compared to the model, as the measure-

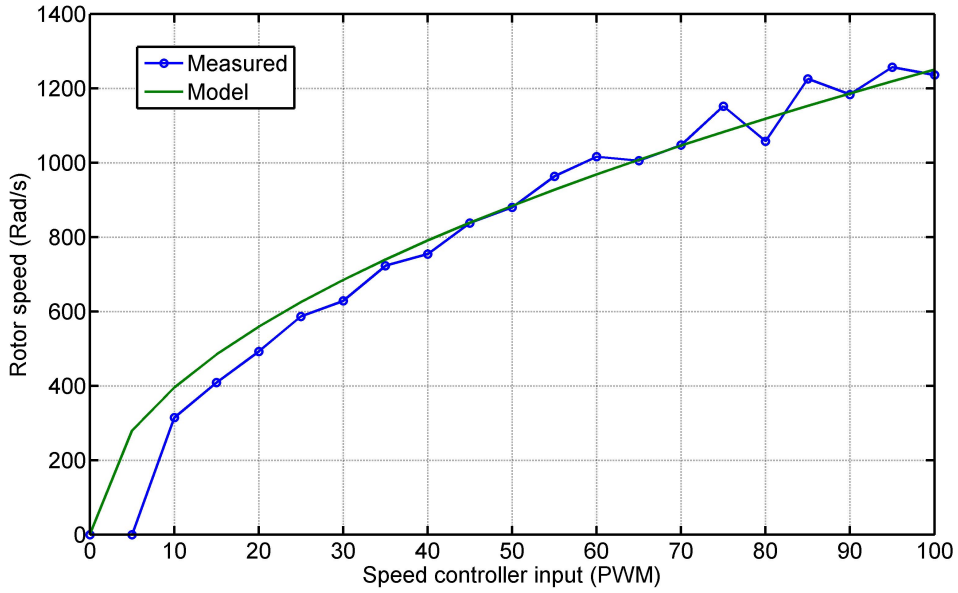


Figure 2.10: Rotor speed measurement results

ment method is relatively accurate. The variance noted at the initial conditions are due to the energy needed to overcome the motor's internal forces, while the disturbances at the higher angular velocities are due to vibrations because of slightly unbalanced rotor blades.

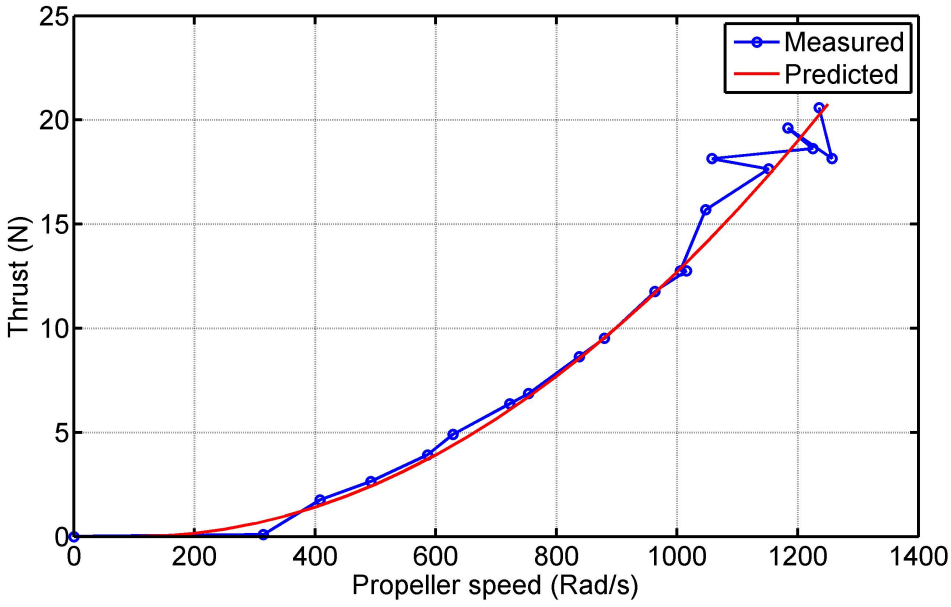


Figure 2.11: Thrust measurement results

Figure 2.11 shows the relationship between the thrust and the rotor speed as expressed by (2.33).

$$F_i = b\Omega_i^2 \tag{2.33}$$

The variations in the measured results at high angular velocities are due to fluctuations in air flow and vibrations.

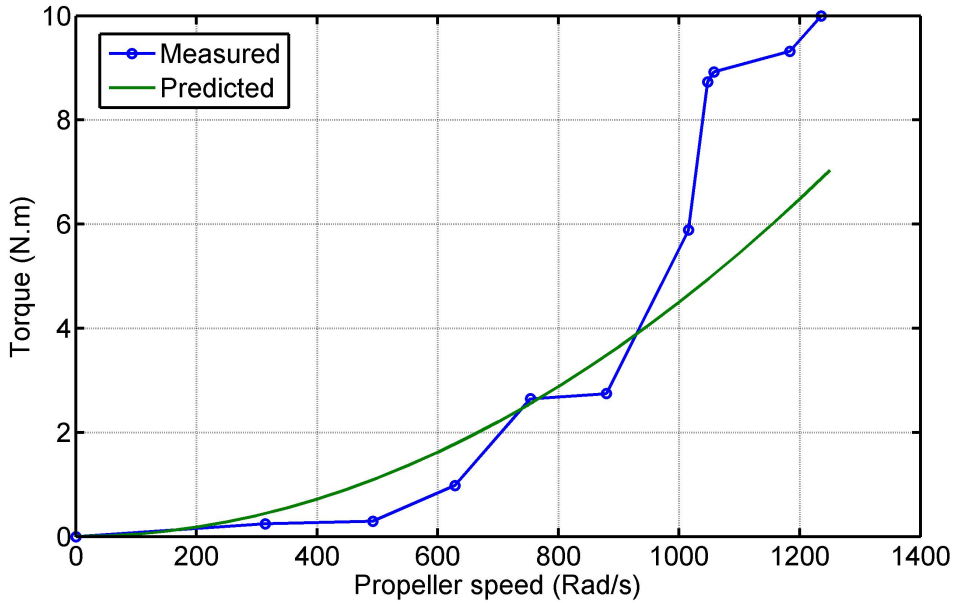


Figure 2.12: Torque measurement results

The ratio between the yaw torque and rotor speed is shown in Figure 2.12 and described by (2.34).

$$T_i = d\Omega_i^2 \tag{2.34}$$

The results show a larger deviation for this measurement, primarily due to the set-up of this experiment. The motor was turned on its side such that a rotational torque is produced into the scale. The variation is due to the vibration of the thrust pushing the blade away from the scale. Again it should be noted that the accuracy of these measurements are not paramount, as the controller should be designed to allow for a certain amount of deviation in their values. The variation in measurements can be reduced by directly attaching the motor to the scale and improving the balancing of the rotor blades. Ideally a test platform should be constructed that securely hold the

motor in place and measure the thrust and torque continuously with load cells as the angular velocity increases. This platform should also be designed so that it does not restrict the flow of the air through the rotors.

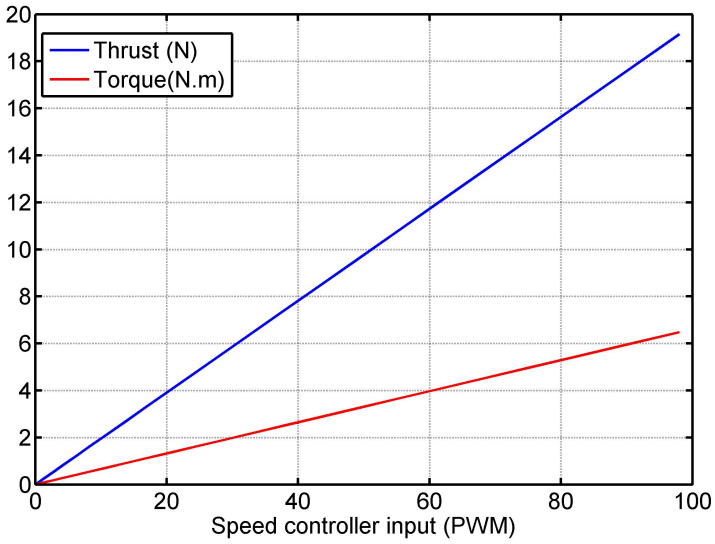


Figure 2.13: Actuator linear model

When the thrust and torque models are combined with the rotor speed model, relating the thrust and torque to the PWM duty cycle, it becomes clear that the speed controller is configured to linearise the response of the motors as shown in Figure 2.13. Table 2.2 shows the parameters determined from the measurements.

Table 2.2: Measured Parameters

Parameter	Value	Unit
Max Thrust	21	N
Max Torque	7.1	$N.m$
Max Rotor Speed	12000	RPM
b	13.3×10^{-6}	$N.s^2$
d	4.5×10^{-6}	$N.m.s^2$
η	125	Rad/s vs Duty Cycle

2.3.4 Parametrized Equations

Matrices

The resulting inertia matrix is shown in (2.35).

$$\begin{aligned}\mathbf{M}_H &= \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{XX} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{YY} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{ZZ} \end{bmatrix} \\ &= \begin{bmatrix} 3.33 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3.33 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3.33 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.007 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.007 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.013 \end{bmatrix}\end{aligned} \tag{2.35}$$

The inverse inertia matrix is calculated using algebraic arithmetic, based on the property of a diagonal matrix shown in (2.36).

$$\mathbf{M}_H^{-1} = \begin{bmatrix} \frac{1}{m} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{m} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{I_{XX}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{I_{YY}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{I_{ZZ}} \end{bmatrix} = \begin{bmatrix} 0.3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 143 & 0 & 0 \\ 0 & 0 & 0 & 0 & 143 & 0 \\ 0 & 0 & 0 & 0 & 0 & 77 \end{bmatrix} \tag{2.36}$$

The Coriolis-centripetal matrix derived from the inertia matrix as in (2.37).

$$\mathbf{C}_H(\overline{\zeta}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.013r & -0.007q \\ 0 & 0 & 0 & -0.013r & 0 & 0.007p \\ 0 & 0 & 0 & 0.007q & -0.007p & 0 \end{bmatrix} \tag{2.37}$$

The Gyroscopic effect matrix can be simplified as in (2.38).

$$\mathbf{O}_H(\overline{\zeta})\overline{\Omega} = \mathbf{O}_B(\overline{\nu})\overline{\Omega} = \begin{bmatrix} \mathbf{0}_{3\times 1} \\ 22.5 \times 10^{-6} \begin{bmatrix} -q \\ p \\ 0 \end{bmatrix} \Omega_T \end{bmatrix} \tag{2.38}$$

The Movement matrix and subsystem U_i is shown in (2.39).

$$\mathbf{E}_H(\overline{\xi})\overline{\Omega}^2 = \begin{bmatrix} \left(s_\psi s_\phi + c_\psi s_\theta c_\phi\right) U_1 \\ \left(-c_\psi s_\phi + s_\psi s_\theta c_\phi\right) U_1 \\ \left(c_\theta c_\phi\right) U_1 \\ U_2 \\ U_3 \\ U4 \end{bmatrix} \tag{2.39}$$

$$\begin{aligned} U_1 &= 13.3 \times 10^{-6} \left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right) \\ U_2 &= 2.4 \times 10^{-6} \left(\Omega_1^2 - \Omega_2^2 - \Omega_3^2 + \Omega_4^2\right) \\ U_3 &= 2.4 \times 10^{-6} \left(-\Omega_1^2 - \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right) \\ U_4 &= 4.5 \times 10^{-6} \left(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2\right) \\ \Omega_T &= -\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2 \end{aligned} \tag{2.40}$$

System of equations

The system of equations can be simplified to the system shown in equation (2.41), with U_i as shown in equation (2.40).

$$\begin{aligned}
\ddot{X} &= (\sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi)\,\frac{U_1}{m} \\
\ddot{Y} &= (-\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi)\,\frac{U_1}{m} \\
\ddot{Z} &= -9.807 + (\cos\theta\cos\phi)\,\frac{U_1}{m} \\
\dot{p} &= -0.92qr - 0.0032q\Omega_T + \frac{U_2}{0.007} \\
\dot{q} &= 0.92pr + 0.0032p\Omega_T + \frac{U_3}{0.007} \\
\dot{r} &= \frac{U_4}{0.013}
\end{aligned} \tag{2.41}$$

Chapter 3

Controller Analysis

---

*"**Everything should be made as simple as possible, but not simpler.**" - Albert Einstein*

3.1 Methodology for controller analysis

3.1.1 Principles of Control Systems

The first step toward designing a controller for any system is to understand the basic control loop. The control loop forms the basis of control theory and is a useful method for the analysis of the system and controller as a whole. In this section a basic discussion is given on this process. Figure 3.1 shows a diagram of a typical closed loop (feedback) control system. In the case of a open loop (feed-forward) control system, the transducer and signal conditioning blocks are removed as there is no state feedback to the controller. It is clear that there are many components and it is worth noting that each interface has unique characteristics in the system that affects the design. For example in most cases the process to be controlled is a real world analogue system, hence operating in continuous time domain, where as on the other hand the controller is usu-

ally a digital controller, hence operating in a discrete time domain. Interfacing these two systems can result in a few problems that have to be addressed if the controller is going to be robust.

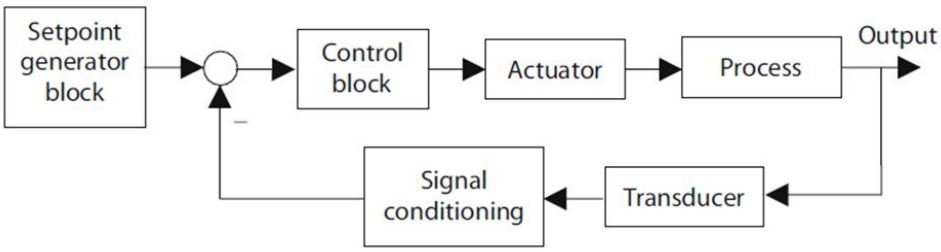


Figure 3.1: Components of a typical closed loop control system

[24] The control loop in the above figure can be characterized by grouping the block into the following:

Process:

The process block refers to the actual system being controlled, which is modelled based on its various inputs, outputs and characteristic transfer function.

Control:

The control block consists of the controller and associated processing to form a stable closed loop system that has to meet some predefined performance requirements. The controller is usually representative of the inverse of the process block, where the input is the error signal determined by calculating the difference between the desired set-point and the measured state.

Actuation:

Actuation is the process allowing the controller to interact with the actual system, generally converting the control signal into a mechanical action. Without the actuation the controller would be unable to affect the system.

Measurement:

Measurement is the process necessary to convert the status of the system in to electronic signals for use in the control block. Without some means to sense the status of the system it would be impossible to control the system. This is done through some type of sensor linked to a certain physical condition. The measurement process is usually the primary source of noise in the system and hence requires a lot of processing to filter the results, which is done in the signal conditioning block.

Communications:

The interlinking of blocks has to conform to the requirements of each of the blocks and hence introduce additional parameters. In some cases the blocks of the system are geographically distributed and hence some form of networking is required, thus introducing new delays that need to be taken into account to ensure good system performance.

3.1.2 Typical Control Issues

With any control system there are various issues that arise which are difficult to predict during the theoretical analysis of the system. These are due to many physical factors or the mathematical interpretation of the system.

System Non-linearity

One of the most challenging parts of control is with regard to non-linearity, making it very difficult to predict the behaviour of the system. Simply put, non-linear systems do not always respond the same way to the same inputs. Clearly it is counter intuitive to attempt control of a system if there isn't certainty regarding the effect an input would have. One method of resolving this issue is by identifying operational regions where the system is linear or approximates linearity. If there is only one linear

region, the solution could be bounded inputs and outputs. If multiple linear regions exist, it might be possible to design multiple controllers, one for each linear region, and some selection process to determine the current region. Note this is dependent on save transition between regions. In other cases the system might be approximated to a linear model based on various assumptions, as in this case study.* By omitting various factors, the system approaches linearity; however these factors are still present during real world operation. Generally systems behave linearly with small input changes and non-linearly to large input changes. Depending on the measure of non-linearity, the controller can focus on using the localized linear behaviour to stabilize the system.

The primary factors responsible for the non-linearity in the system are as follows:

Sensors:

The sensors have a limited measurement scope which if exceeded by the system results in non-linear observation. Ensure that the system does not exceed the sensors measurements range or is below the sensor sensitivity levels.

Actuators:

In any real world system the actuators have physical limitations. These limitations are non-linear behaviour and if the controller exceeds these limitations, a condition known as actuator saturation occurs. Consequently the controller no longer affects the system, resulting in an open-loop control system that will probably become unstable. Ensure that the controller is tuned and/or limited to the capabilities of the actuators.

Model assumptions:

"Essentially, all models are wrong, but some are useful." - George E P Box.

The use of modelling techniques are inevitably based on assumptions that simplify the mathematics by excluding factors that are difficult to model or calculate. These assumptions hold true under certain conditions, the challenge is matching these con-

*The quad-rotor model assumptions were discussed in section 2.2.1.

ditions with the actual operational conditions. A simple example of the effect of assumptions can be illustrated by looking at the work needed to move a boulder. Given a model that calculates the effort needed to roll a boulder based on the distance covered, it is true that gravity (i.e. the potential energy) can be omitted if the rock is moved in a horizontal plane. However if the same model is used on an incline plane, it would show that it takes the same amount of effort to move the rock up a hill as it does to move it down hill. This obviously is not the case, but the misinterpretation occurs because the required conditions for the original assumptions where not met. This can easily happen as the assumptions become more complex and if the conditions aren't well documented.

External factors:

The exclusion of external factors that are difficult to predict is common practice in modelling. In the case of flight automation, the controllers are designed (at least initially) by excluding factors such as wind and drag. Both wind and drag are dependent on the exact aerodynamic characteristics and atmospheric conditions. In the case of the aerodynamics, it is difficult to calculate, as even the texture of the paint used on the fuselage can affect the aerodynamic performance. Atmospheric conditions aren't mathematically difficult to model, but near impossible to predict. Simulated scenarios can be used to test performance during various conditions. The assumption is that during a stable indoor hover, neither wind nor drag is an issue. However when operating outdoors with increased speed, the non-linearity becomes prominent and aberrant behaviour is observed in the controller. Depending on the application, these factors need to documented and accounted for.

Other non-linearities include discontinuous transition points. Considering the case of rotational motion, when 360 degrees is reached the angle returns to zero. This is a discontinuity and hence non-linear behaviour. The instantaneous change in angle is also considered to be very large if determined by differentiation.

Input Disturbances

Input disturbances can be caused by factors such as batteries loosing charge, decreasing the efficiency of the actuators, loss of power or loss of user inputs (loss of communication). These possible disturbances need to be identified, accounted for and documented.

Output Disturbances

Loading disturbances on the process translates throughout the system and need to be documented. These disturbances can be due to noise and performance variations in the actuators, either due to changing environmental conditions, mechanical wear or damage.

Measurement noise disturbances

Any measurement technique has some form of noise depending on the method employed by the transducer. Understanding the source and nature of the noise is useful in the development of a signal processing method to remove it. Noise can be internal to the sensor, such as thermal noise present in all electronics, due to the discrete measurement methods or due to physical conditions such as vibration and electromagnetic noise. If proper discrete signal processing principles are not followed, problems such as aliasing can arise. Other processing related noise sources include differentiation that serves to amplify noise. It is necessary to understand the effect that any given processing method has on noise before it is applied to the measurement signal.

Inverse response

Inverse response leads to the controller having the opposite effect as desired. Inverse response can be caused by internal wind-up, energy build-up, or to state it more simply, momentum. However the lack of this energy can also lead to inverse response. It can also be caused when the system enters a non-linear state where the output is inversely proportionate to the input. In the case of flight automation this is observed during inverted flight. A controller programmed to pull the elevators up to increase altitude, would force an inverted plane intro a dive. To resolve the problem the controller has to be aware in the changes in system dynamics. One solution to this problem is the use of multiple controllers, tuned to work at different operating points.

3.1.3 Controller Performance Evaluation

Closed-loop requirements

The performance specification of the controller is usually expressed in terms of closed-loop requirements. Some more prominent examples of these evaluators include stability, reference tracking, supply rejection, load disturbance rejection, measurement disturbance rejection and robustness. The problem however is that optimizing one evaluator can destabilize another. Hence these requirements have to be prioritized.

Stability:

The stability of the system refers to factors such as settling time and steady-state error, the degree of oscillation at steady-state, etc.. This is a priority if the accuracy of the controller is paramount.

Reference tracking:

Reference tracking is similar to settling time, however the input is usually contin-

uous and not step or impulse functions. With reference tracking it is critical to keep the error signal at a near zero value. It is also important in this case to minimize the overshoot that occurs during sudden input changes.

Input disturbance rejection:

This reflects the sensitivity of the controller to changes in the supply of the system. This is critical in systems that behave as buffers, maintaining a stable output with an unstable input.

Output disturbance rejection:

Load distances are due to variations in the behaviour of the system. This factor is critical in non-linear systems where the output can appear to be unstable.

Measurement disturbance rejection:

Measurement noise produces by the sensors need to be removed to ensure the controller doesn't respond to non-existent behaviour. For example, the noise produced by frame vibration and that of wind sway, have distinctly different spectral characteristics.

Robustness:

Robustness is the overall ability of the control to continue operation during severe changes in parameters. This is a primary concern in safety critical control systems.

Performance evaluators

Section references: [25]
These performance evaluators describe the system behaviour in reference to unit step response.

Rise Time:

The rise time (T_r) is the time it takes for the system response to intersect the region within $\pm 10\%$ of the desired value for the first time since the transient event.

Peak Time:

The peak time (T_p) is the time it takes for the system response to reach the peak value of the transient response.

Percentage Overshoot:

The percentage overshoot (PO) is defined as the ratio between the peak value reached and the final or desired value.

$$PO = \frac{peak\ \ value - final\ \ value}{final\ \ value} \times 100\% \tag{3.1}$$

Steady State Error:

The steady state error (e_{ss}) of a system is the error observed in the system after the transient response has ceased, leaving only continuous response.

$$e_{ss} = \lim_{t \to \infty} e(t) = \lim_{s \to 0} sE(s) \tag{3.2}$$

Settling Time:

The settling time (T_s) is the time the system output takes to settle to a value within a given percentage (usually 2%) of the desired value.

Sensitivity:

The sensitivity of the system to changes in parameters is crucial to the robustness of the controller. The sensitivity is defined as the ratio of the change in the system transfer function to the change in the process transfer function in reference to a given parameter change.

$$S = \frac{\delta T/T}{\delta G/G} = \frac{\delta lnT}{\delta lnG} \tag{3.3}$$

Performance Indices:

With the advent of adaptive controller, the need arose for a quantitative measure of a control systems performance. Each performance index places a different emphasis on the system performance in regard to a specified characteristic. The multiplication of time reduces the weight of the large initial error and focussing on steady state error, while the squared error emphasized quick system response. The four performance indices listed below are the ones most commonly used, although others can be defined to better suit the emphasis of the optimization. A controller is considered to be optimal if the value of the given performance index reaches an extremum.

Integral of the Absolute Error (IAE)

$$IAE = \int_0^T |e(t)|dt \tag{3.4}$$

Integral of the Squared Error (ISE)

$$ISE = \int_0^T e^2(t)dt \tag{3.5}$$

Integral of the Time and Absolute Error (ITAE)

$$ITAE = \int_0^T t|e(t)|dt \tag{3.6}$$

Integral of the Time and Squared Error (ITSE)

$$ITSE = \int_0^T te^2(t)dt \tag{3.7}$$

Dampening ratio:

The dampening ratio (ζ) describes the dampening of oscillation in the systems response. Critical dampening ($\zeta = 1$) is the point between over ($\zeta > 1$) and under ($\zeta < 1$) dampening. Equation (3.8) shows the generic transfer function of a second order system, indicating the presence of the dampening ratio. In the equation ω_n is the natural frequency of the system and the resonance frequency is given by the equation $\omega_r = \omega_n\sqrt{1-\zeta^2}$. Figure 3.2 shows the effect of the dampening ratio.

$$T(s) \approx \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{3.8}$$

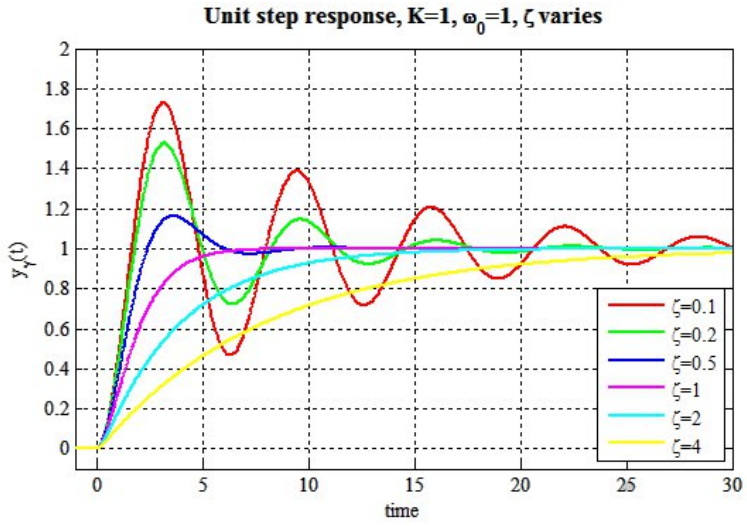


Figure 3.2: Dampening ratio

Delay:

The actuation delay in the system is the time it takes for a control action to affect the system. This is important to take note of as it increases the rise time and similar parameters, but cannot be reduced by tuning the controller.

Figure 3.3 shows the step response of a system and the various performance evaluators.

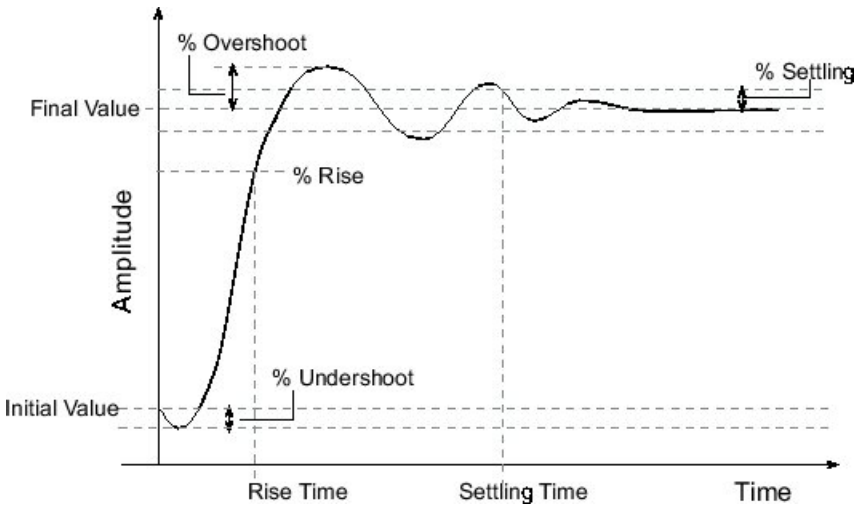


Figure 3.3: Step response with performance evaluators

Figure 3.4 shows the behavioural characteristics described by the performance param-

eters on the various response curves.

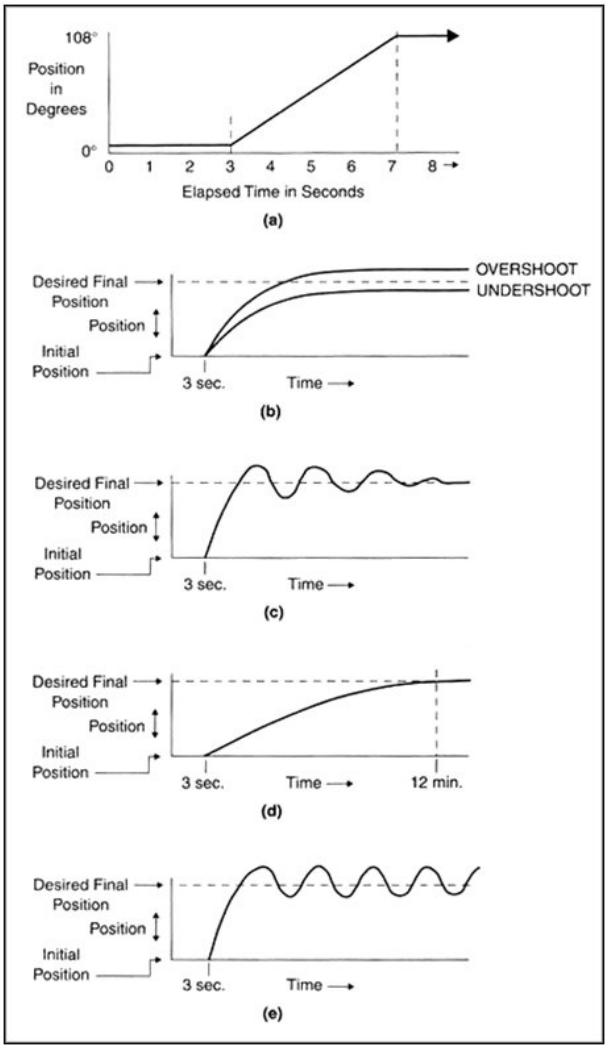


Figure 3.4: Controller behavioural characteristics

(a) Ideal response, (b) Steady state error, (c) Cyclic response with overshoot and oscillation that reaches stability, (d) Damped signal without overshoot, but with large settling time, (e) Unstable oscillating system

3.1.4 Model Characterization

Given the characteristics of control systems given in the previous section, it is now possible to characterize the quad-rotor system using the information gained from the model. Figure 3.5 shows the generalized control loop for the quad-rotor system, including the various disturbances present in the system. Note that the detailed working of the controller, including the distinction between trajectory and orientation controllers, is not expressed in the diagram and is represented by the control block.

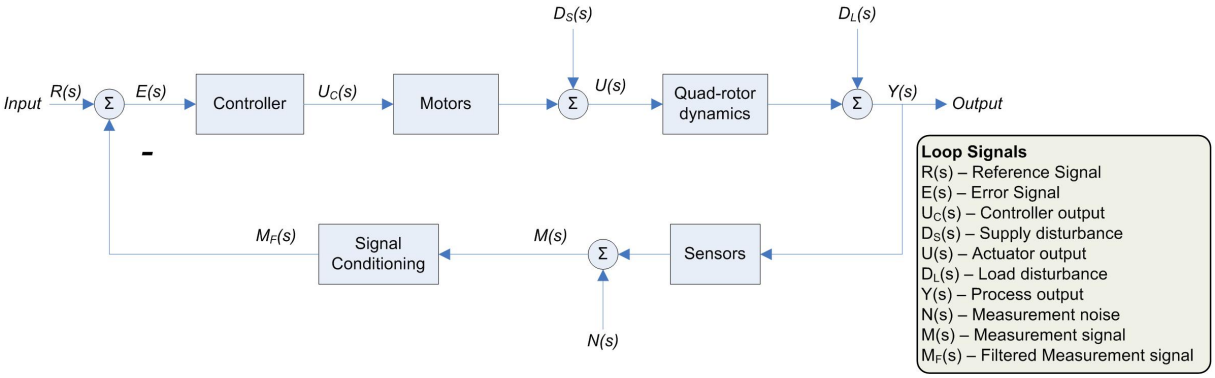


Figure 3.5: Control loop characteristics and signals

The system characteristics will now be discussed in reference to this control loop.

There are various *system non-linearities* present is this system as the system is in fact non-linear by nature and only simplified to have localized linear behaviour. The first simplification is the distinction of translational and rotational motion, where the trajectory can be considered to be a linearly related to the orientation of the craft, while the orientation can be considered to the linearly related to small changes in rotor speeds. Hence the non-linear system can be divided into a series of linear subsystems, thus linearising the system by cascading these linear subsystems that can each be controlled by bounded linear controllers.

Sensor and actuator limitations are dependent on the hardware and documented in later sections. Sensor arrays of various measurement ranges can be integrated to pro-

vide greater measurement range while providing good localized sensitivity. By measuring the rate of change directly, the errors introduced by integration and differentiation are omitted. Actuators, in this case brushless motor driven fixed rotor, have only positive response in a limited range. This range needs to be documented and related to the inertia characteristics of the quad-rotor to determine operational points. The efficiency of the rotors are also dependant on atmospheric conditions and hence vary, but within a given range. Determining this range provides a good reference to determine the flight envelope.

The assumptions made in this model were discussed at the beginning of section 2.2.1. The conditions needed for these assumptions are relatively easy to meet, but the presence of wind, or air circulation, is possible even during indoor flight. The processes to compensate for the variation in actuator efficiency (effective thrust) would be able to compensate for the presence for light winds. Variation in the forces caused by wind will increase the steady state error, magnitude of oscillation or sway during hover. This does not reflect the instability of the controller, merely the presence of variations of forces. The oscillation can be minimized, but minor deviations will always be present due to controller and actuation delays. As atmospheric conditions deteriorate, the inherent dynamic instability in the system will increase, increasing the steady state error. (Refer to Appendix B)

The presence of discontinuities in the mathematical interpretation of the dynamics results in non-linearities that can cause aberrant behaviour in the controllers. The interpretation of rotational motion is a circular movement from 0° to 360°, where 0° and 360° represent the same point. Hence the transition from 360° to 0° is a discontinuity that presents a few practical problems. If for example the controller is instructed to change angle from 350° to 10°, the controller will reverse and rotate through 340° when in fact a far more efficient action would be to implement a positive rotation through the transition of a mere 20°. This may only be inefficient regarding yaw control, but in the case of pitch and roll control the full rotation will result in the quad-rotor flipping over, causing an uncorrected altitude controller to accelerate the quad-rotor downward, attempting to correct for the loss in altitude by increasing thrust. A simple solution for

pitch and roll control is to change the angular interpretation from 0° through 360° to a model employing a rotation from -180° through 180°. Since the craft is expected to operate only in a predominantly upright position, the direct correction of the controller will be correct as passing through the $180^\circ / -180^\circ$ point is not recommended. In the case of yaw control, the use of 0° through 360° correlates to the standard heading definition, hence a more practical solution is to apply a modulus operation on the error signal, placing it in the -180° through 180° range.

The *input disturbances* that could affect the system are not all shown in the control loop. The power delivered to the motors are determined by the charge on the batteries, hence as the charge decreases the efficiency of the motor decrease, in turn reducing the effective thrust. Disturbances in the reference signals can be introduced by communication errors between the ground control station and the quad-rotor. The system has to validate the data received and also have a rudimentary action plan should communication be lost completely.

Output disturbances go hand in hand with the system non-linearities as they are primarily the result of aerodynamic disturbances, hence dependant on atmospheric conditions. Another source of output disturbances is vibration caused by imbalance in the rotors. The use of three blade rotors is done in the hope of reducing this vibration.

Measurement disturbances are caused by various processing, sampling, electronic and physical factors. Firstly, any sensor is plagued by internal noise due to the mechanisms employed by the transducer. Then there are various electromagnetic noise sources that affect the measurements. These are dependent on the circuit design and layout, and include factors such electromagnetic interference, ground level noise, etc. To minimize this and simplify the design, the use of digital sensors is recommended. The sampling of the measurements also has to take into account good sampling principles to avoid effects such as aliasing. The measurements signal also need to be filtered before any other processing/interpretation is done to prevent processing aberrations such as noise amplification through arithmetic such as differentiation.

The *closed loop requirements* of the case study are dependent on the application. Each parameters priority is discussed based on its effect of the application.

The stability of the system is ultimately one of the primary objectives of the controller; hence minimizing the oscillations in the frame is paramount. The size of the steady state error is paramount in applications such as aerial photography, but less critical if fast aggressive manoeuvring is necessary. The Percentage Overshoot (PO) is an important factor to minimize due to the cascade of controllers. If both trajectory and orientation controllers have a high overshoot, the total overshoot becomes unstable. For applications that require accurate manoeuvres, such as indoor flight, the percentage overshoot has to be the close to zero. On the contrary, for applications where fast response in necessary the settling time is the priority and overshoot might be more permissible. The balance between overshoot and response is the case where reference tracking is critical. For an automated UAV that employs automatic navigation, reference tracking is the primary focus of controller optimization. This requires good response with minimal overshoot. Typically the reference signal has specified characteristics that limit the rate of change. Reference tracking is not concerned with step response, only the accuracy of the output to follow the reference signal with minimal delay and error.

The ability of the controller to reject disturbances from multiple sources, including input, output and measurement noise, is critical in a dynamically unstable system such as this. As stated before, the various disturbances can be characterized as being in specific spectral bands determined by their physical characteristics. Other output disturbances can be corrected for by the use of techniques employing operational set-points. The robustness of the control system is its ability to maintain stability during adverse changes in parameters, which is the focus of this study.

Understanding the dynamics of the craft help identify possible inverse response points. As stated before, the biggest concern is in the interpretation of the orientation angles, but this problem can be corrected by using modulus operations. The altitude controller has to be designed to account for the fact that the actuators can only produce a positive

response, making it impossible to maintain altitude if the craft is upside down. The craft first has to be in an upright position to be able to regain altitude control.

3.1.5 Comparison Methodology

Now that characteristics for the generic controller and the case study have been determined, the process of comparing the various controllers to the parameters of the system can be defined. Figure 3.6 shows the basic framework of the selection process.

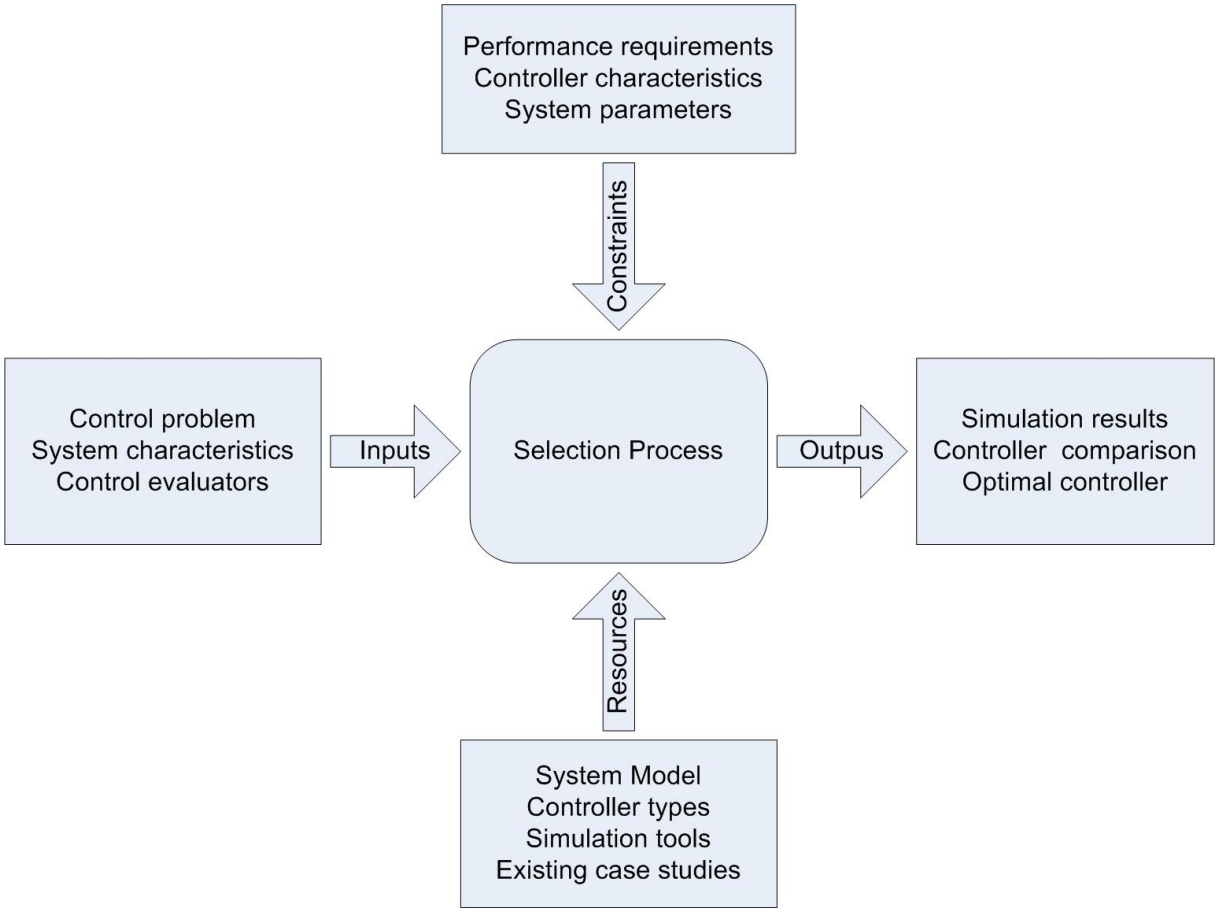


Figure 3.6: Comparison methodology

The inputs to the study, as discussed in the previous sections, are the control problem, system characteristics and control evaluators. The various constraints that guide

the process to the selection of the optimal controller include the performance requirements, controller characteristics and system parameters. The resources that are available to the process are the system model, the various controller types, simulation tools and existing case studies. The outputs of the process are simulation results, controller comparison and the selected optimal controller.

The comparison is done is two distinct steps, the first being a system level comparison and the second begin a technical level comparison. On the system level the system characteristics are compared to the controller characteristics to determine the suitability of the controller. The controller is evaluated based on the following characteristics: [26]

Maintainability: The ability to ensure continued operation during the operational life-cycle of the system.

Sustainability: The ability to ensure correct operation throughout the various design changes in the life-cycle of the system.

Testability: The ability to safely and easily test the controller during the development cycle.

Usability: The ease of implementation of the controller, reflecting the complexity, the required skill level and the compatibility of the controller.

Reliability: The ability of the controller to remain stable even during parameter changes as well as the risk of aberrant behaviour.

The technical level comparison makes use of technical performance evaluators to analyse the performance of the controller. The primary evaluators are the linearity, stability, robustness and adaptability of the controller. The control evaluators are prioritized based on the performance requirements for the given control problem. The robustness of the controller is evaluated based the change scope of the system parameters. Existing case studies regarding implementation of each of the controller types are evaluated to determine the complexity and suitability of each technique. Those control

techniques that are suitable for implementation are simulated using a simplified model and compared based on the control evaluators to find the optimal controller. The optimal controller is then simulated using the full complexity of the model and scope of the parameter changes to validate the selection of the controller.

If the hypothesis is validated the controller is implemented and the implementation is used to verify the accuracy of the model.

3.2 Control Loops

3.2.1 Flight Modes

The definition of various flight modes allows for the optimization of the controller in multiple operating points based on the requirements of each mode. The implementation of the various flight modes can be done through the use of multiple controllers or gain scheduling. Listed below are a few examples of possible flight modes that can be defined for a quad-rotor.

- Normal flight
  - Long range cruising
  - Standard operations
- Hover / Position hold
  - Aerial videography
  - Surveillance
- Accurate manoeuvrability
  - Take-off
  - Landing
  - Docking
  - Formation flying
- Aggressive/Agile manoeuvrability
  - Aerobatics
  - Combat (UCAVs)

3.2.2 Attitude Control

Attitude control is achieved by introducing differential rotor speed into the dynamics of the craft. These variations on rotors speeds introduce rotational torques that changes the attitude of the craft.

The overall attitude control can be simplified into three separate controllers, each responsible for a single axis. The cross correlation between the controllers can be characterized and integrated through simple summation, while the gyroscopic forces and Coriolis effect can be modelled as load disturbances. Since all motion is kept at low speeds, the forces produced by the gyroscopic and Coriolis effects are kept small and hence can be considered as load disturbances and not part of the primary dynamics that need to be predicted and controlled. If aggressive manoeuvring is required, it is recommended that controllers be implemented to compensate for these forces to ensure accurate operation.

Pitch control

Pitch is the forward and backward tilt of the quad-rotor, responsible for forward motion. The modelling and interpretation of the pitch angle is shown in Figure 3.7. The plant dynamics for the pitch controller is shown in (3.9).

$$\dot{q} = \frac{I_{ZZ} - I_{XX}}{I_{YY}} pr + \frac{J_{TP}}{I_{YY}} p\Omega_T + \frac{U_3}{I_{YY}} \tag{3.9}$$

Since it is assumed that the impact of the Coriolis effect and Gyroscopic forces are negligible compared to that of the motor forces, the pitch dynamics can be simplified to (3.10).

$$\dot{q} = \frac{U_3}{I_{YY}} = \frac{lb(-\Omega_1^2 - \Omega_2^2 + \Omega_3^2 + \Omega_4^2)}{I_{YY}} \tag{3.10}$$

The simplified single axis pitch controller is illustrated in Figure 3.8 with $\Delta\Omega$ indicating the differential angular velocity required. Note that the action of the controller is distributed evenly between the four rotors. The use of inverters simplifies the definition of the orientation and influence of each motor.

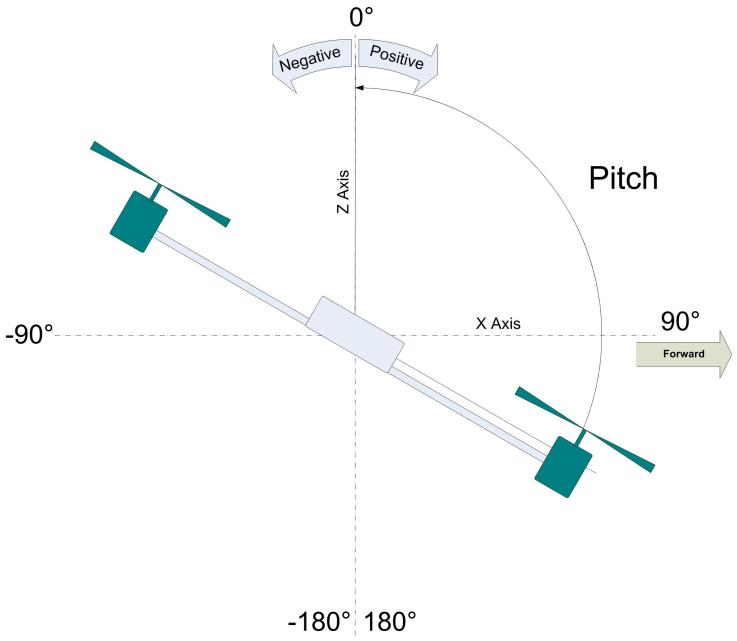


Figure 3.7: Pitch angle interpretation

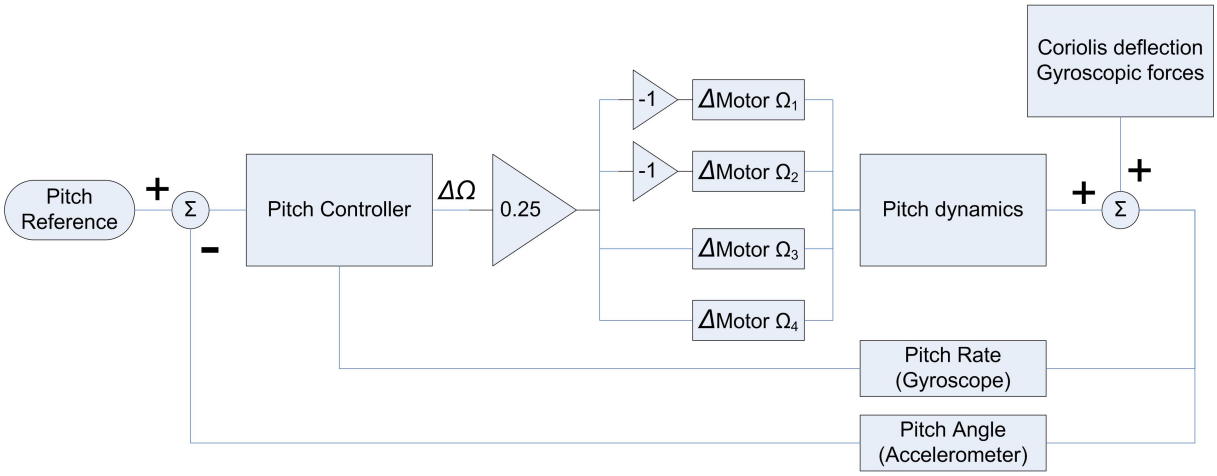


Figure 3.8: Pitch controller

Roll control

Roll is the left and right tilt of the quad-rotor, responsible for lateral motion. Figure 3.9 shows the orientation and interpretation of the roll angle. The plant dynamics for the

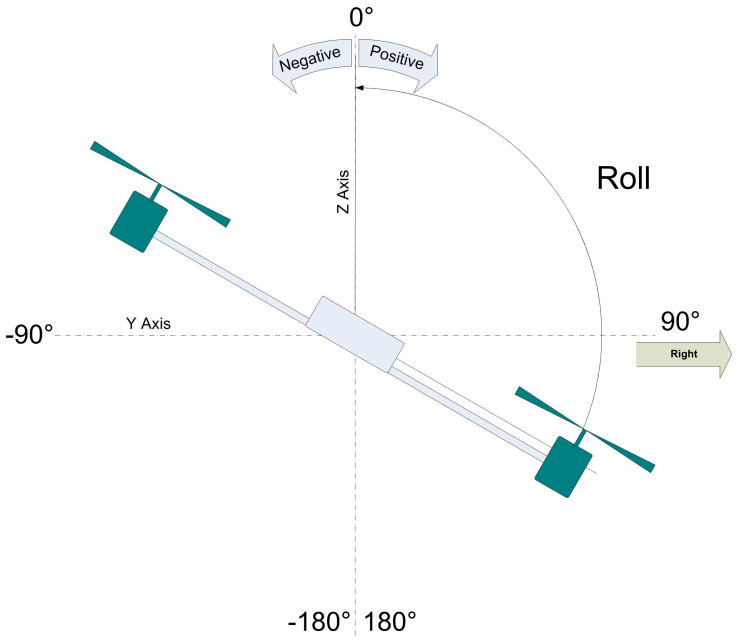


Figure 3.9: Roll angle interpretation

roll controller is shown in (3.11).

$$\dot{p} = \frac{I_{YY} - I_{ZZ}}{I_{XX}} qr + \frac{J_{TP}}{I_{XX}} q\Omega_T + \frac{U_2}{I_{XX}} \tag{3.11}$$

Since it is assumed that the impact of the Coriolis effect and Gyroscopic forces are negligible compared to that of the motor forces, the roll dynamics can be simplified to (3.12).

$$\dot{p} = \frac{U_2}{I_{XX}} = \frac{lb(\Omega_1^2 - \Omega_2^2 - \Omega_3^2 + \Omega_4^2)}{I_{XX}} \tag{3.12}$$

The simplified single axis roll controller is illustrated in Figure 3.10 with $\Delta\Omega$ indicating the differential angular velocity required. Layout is similar to the pitch controller, only with changes in the location of the inverters.

Yaw control

Yaw is the left and right horizontal rotation of the quad-rotor, responsible for the heading of the craft. Figure 3.11 shows the orientation and interpretation of the yaw angle. It is worth noting that the model's orientation is the opposite of the standard naviga-

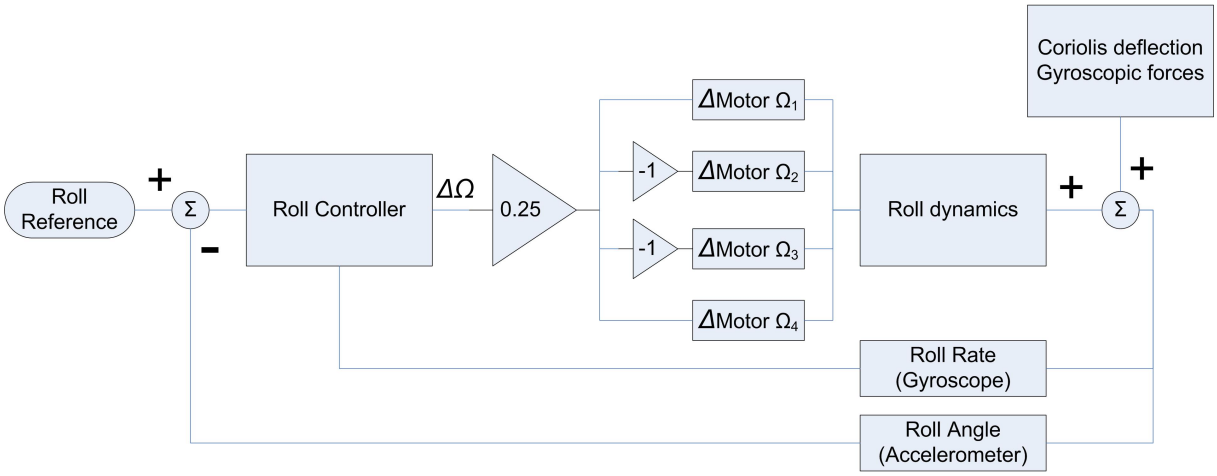


Figure 3.10: Roll controller

tional convention. Conversion can be done using (3.13).

$$magnetic\ \ heading = -1 \times yaw_angle + 360^{\circ} \tag{3.13}$$

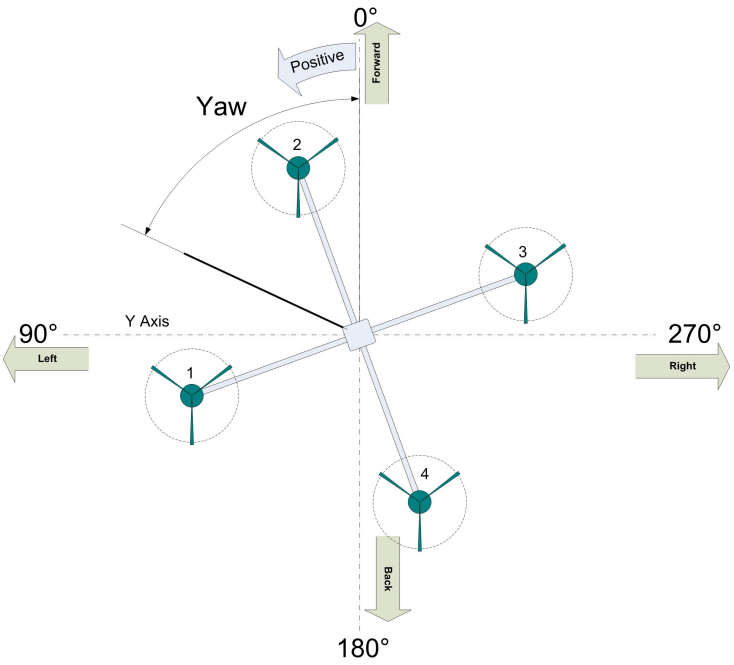


Figure 3.11: Yaw angle interpretation

The plant dynamics for the yaw controller is shown in (3.14).

$$\dot{r} = \frac{I_{XX} - I_{YY}}{I_{ZZ}} pq + \frac{U_4}{I_{ZZ}} \tag{3.14}$$

Since it is assumed that the impact of the Coriolis effect and Gyroscopic forces are negligible compared to that of the motor forces, the yaw dynamics can be simplified to (3.15).

$$\dot{r} = \frac{U_4}{I_{ZZ}} = \frac{d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2)}{I_{ZZ}} \tag{3.15}$$

The yaw controller is shown in Figure 3.12 with $\Delta\Omega$ indicating the differential angular velocity required. Once again the layout is similar to that of the previous controllers, with only the inverters in a new configuration.

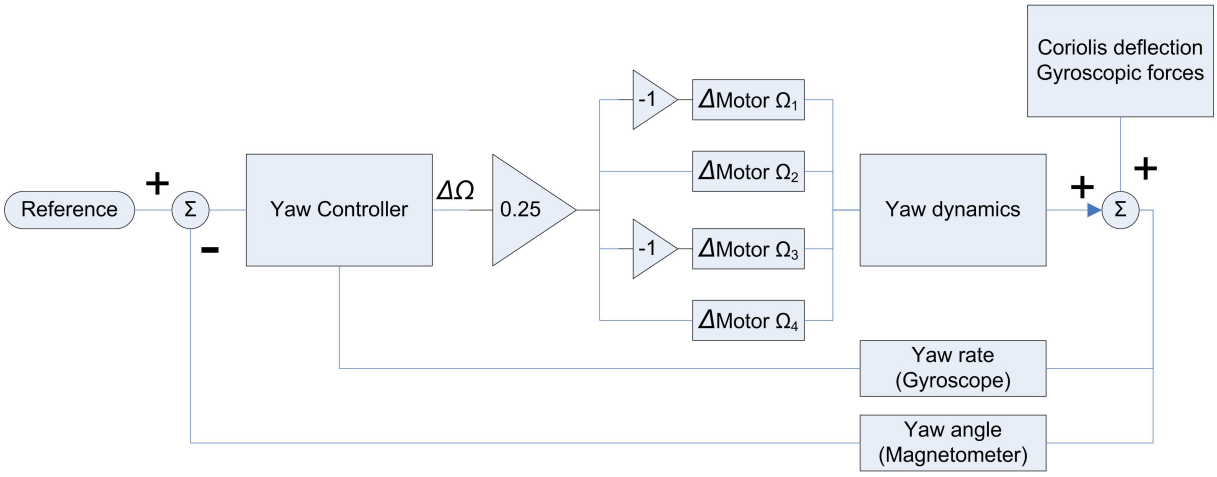


Figure 3.12: Yaw controller

Attitude control integration

As each controller only produces a differential rotor speed ($\Delta\Omega$), the integration of the various controllers requires a simple summation of the various differential rotor speeds. The summation provides the rotor speed combination that will result in the required orientation changes. It is only necessary for differential rotor speeds since only the relative difference in rotors speeds determine the rotational moments that the

rotors produce, and the total thrust is determined by the altitude controller. Figure 3.13 shows the integration of the various attitude controllers.

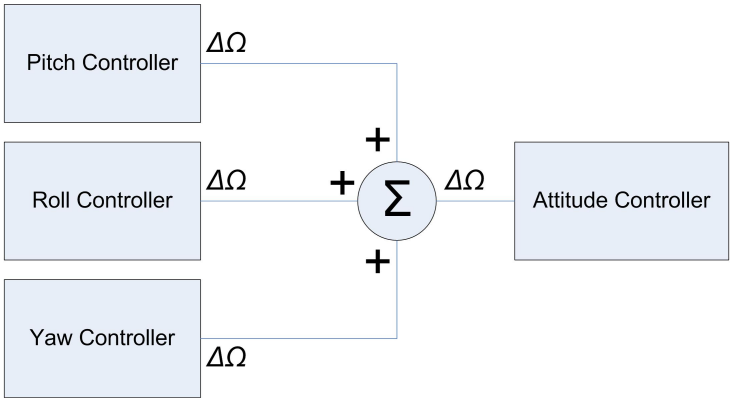


Figure 3.13: Attitude control integration

3.2.3 Trajectory Control

The trajectory controller, as stated in the initial model, is used to convert navigational commands into the required change in orientation that results in the requested motion. In order to achieve this, a thorough understanding is needed of the relationship between the orientation of the craft and the resultant translational motion. Considering the fact that the translational acceleration is proportional to the sine of the relevant orientation angle to the vertical, i.e. that the forward acceleration is proportional to the sinus of the pitch angle. Hence the maximum acceleration is experienced at a pitch of 90°. The problem however is that due to the nature of a helicopter, the vertical thrust component is proportional to the cosine of angle to the vertical axis. Thus given the maximum horizontal thrust at 90°, the vertical thrust component is reduced to zero. This is a critical problem, since without any vertical thrust the craft will effectively be in free fall and unable to maintain its altitude. Further more, if the angle exceeds 90° the vertical thrust is actually negative (pointing downward), causing the craft to accelerate towards the earth.

In order to prevent this, the orientation controller must be bounded to a safe flight

envelope, ensuring that the craft's angle to vertical never exceeds the point where the craft is unable to provide sufficient thrust to maintain altitude. If however this behaviour is required for improved manoeuvrability, it is the responsibility of the trajectory controller to manage this.

Due to this nature of rotor craft, the vertical trajectory controller or altitude controller is in fact controlling the collective thrust of the aircraft. In this case the actuation is balanced with the force of gravity, i.e. that the negative actuation is done by gravity whilst the positive actuation provided by the propellers. This balance only works if the craft is upright, as the direction of gravity is fixed. Even more extreme actuation can be achieved by allowing the craft to invert, but this requires some form of prediction to prevent extreme overshoots.

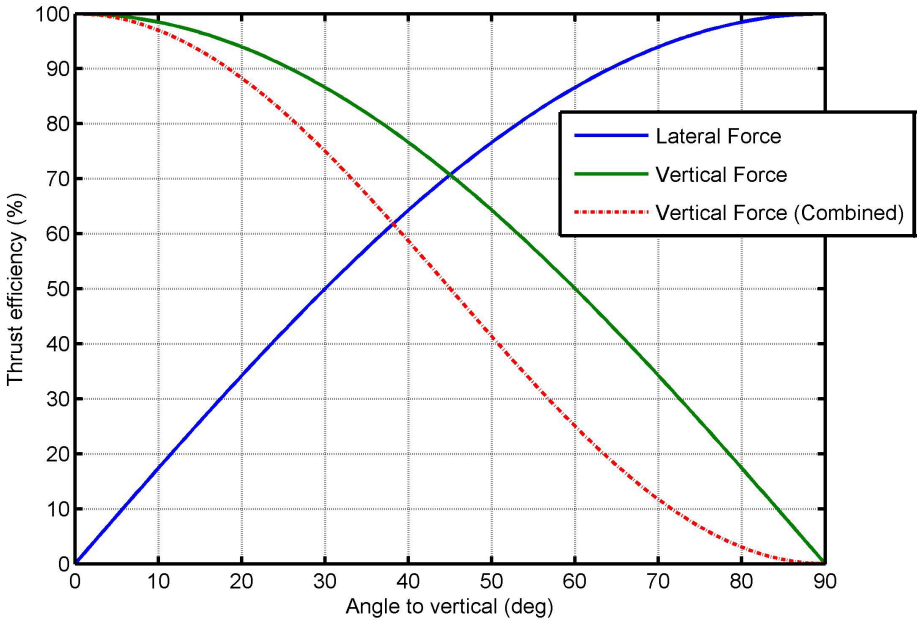


Figure 3.14: Effective thrust

Figure 3.14 shows the effective thrust for the given angle to vertical. It is clear from the graph that the optimal compromise would be to limit the orientation of the craft to 45°, as this allows both lateral and vertical thrust to maintain a 70% efficiency. When both the pitch (θ) and roll (ϕ) angle are non zero, they both affect the efficiency of the vertical thrust. The dotted line indicates the effective thrust given a combined pitch

and roll action simultaneously. The effective vertical thrust, as shown in equation 2.29, is determined by the product of the cosines of the two angles, thus giving an effective thrust of 50% at the 45° mark. It it therefore necessary to determine the limiting point based on the combined motion, rather than a single action.

Given the ratio of the maximum collective thrust versus the required lift thrust, the safe operating scope can be determined. If the ratio is 2:1, i.e. that the collective thrust is double that of the required lift thrust, the craft can be allowed to operate up to an angle of 60° where the vertical thrust efficiency is 50%. This does however mean that the craft has to max out the thrust to maintain altitude, limiting its ability to control orientation and hence decreasing stability. This will also waste a great deal of energy, be it battery power or fuel.

The horizontal trajectory controller used for this study is simply a translator of user commands. It linearly translates the requested translational velocity into a given orientation. To calculate the required orientation, the acceleration vector is first determined. Given the requested trajectory, the components are calculated and then converted into an optimal orientation as shown in (3.16).

$$Angle = \arcsin(\%Speed) \times \frac{Angle\ \ Limit}{90^\circ} \tag{3.16}$$

To simplify this cross-correlation, altitude control is done independently, thus the controllers automatically stabilize at the required operating point. This does however required proper dampening to prevent severe oscillations. Thus given the size of the requested translational velocity, a proportional angle is sent to the orientation controller. The heading is controlled by the yaw controller, although note that the rotational convention is inverted.

3.2.4 Altitude Control

Due to the various aspects of altitude control, the process is divided into three distinct processes. Once again the complex non-linear system can be approximated by a series

of linear approximations. The first is *altitude stabilization*, responsible for compensation of changes in thrust efficiency and disturbances. The second is an *attitude compensator*. Since the vertical component of the trust is dependent on the orientation of the craft, it is necessary to compensate for the loss in vertical thrust as the orientation changes.

Since the previous two controllers only respond to differential changes, the third is responsible for determining the collective thrust center point. This centre point is the reference value that is combined with the differential rotor speed changes to determine the final rotation speed of the rotors. As this is primarily dependent on the deployed weight of the craft, this controller is referred to as the hover controller. It determines, in real-time, the collective rotor speed needed to lift the craft for various payloads. Another factor to account for here is the change in propeller efficiency as the air pressure decreases with increased altitude. Although not relevant to this study, is worth noting that the configuration of the hover controller will detect and compensate for this phenomenon.

Altitude Stabilizer

The altitude stabilizer is a high speed compensator with limited actuation ability. It measures the error in altitude and compensates. The limited actuation is to prevent actuator saturation that will interfere with the attitude stabilizer. This also limits the crafts climb rate and the wastage of battery power on excessive actuation. Figure 3.15 shows the control flow of the altitude stabilizer.

Attitude compensator

The attitude compensator is another altitude controller with limited actuation. This controller compensates for the loss in vertical thrust due to changes in the attitude (orientation) of the craft. The primary contributors are the pitch and roll of the craft. Also note that the increased thrust, at a given angle, also increases the horizontal component of the thrust, hence increasing the translational acceleration. As the attitude exceeds a

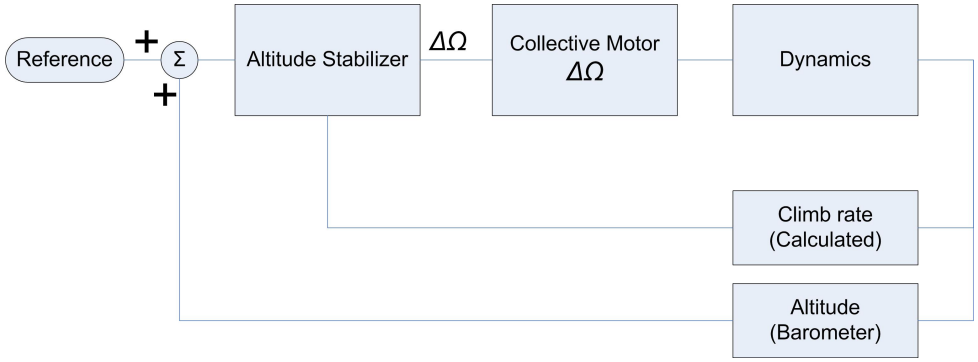


Figure 3.15: Altitude Stabilizer

certain angle, the horizontal component of the thrust exceeds the vertical component, eventually leading to actuator saturation and the inability to maintain control. Hence the compensator only operates for small angles that are needed to maintain a constant translational velocity. If the controller detects craft inversion, it will stop operation completely. Figure 3.16 shows the flow of the attitude compensator.

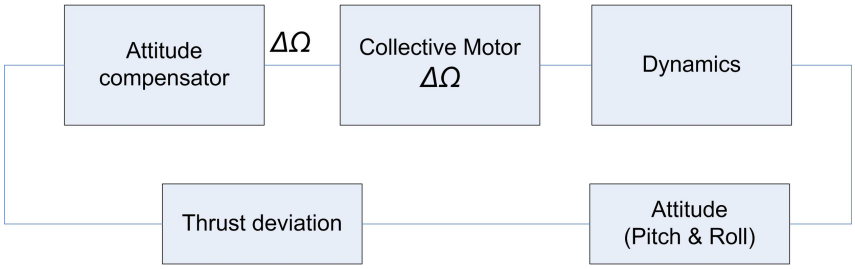


Figure 3.16: Attitude compensator

Hover controller

The final and most important of the altitude controllers is the hover controller. It calculates the thrust needed to lift the craft for the given payload. This calibration is initially done at take-off, by measuring the rotor speed at which the craft leaves the ground. Continuous calibration is possible by looking for a constant output (DC bias) from the altitude stabilizer. By realizing that the altitude stabilizer is continuously compensating in a given direction, the hover controller can adjust its calibration. Figure 3.17

shows the control loop of the hover controller.

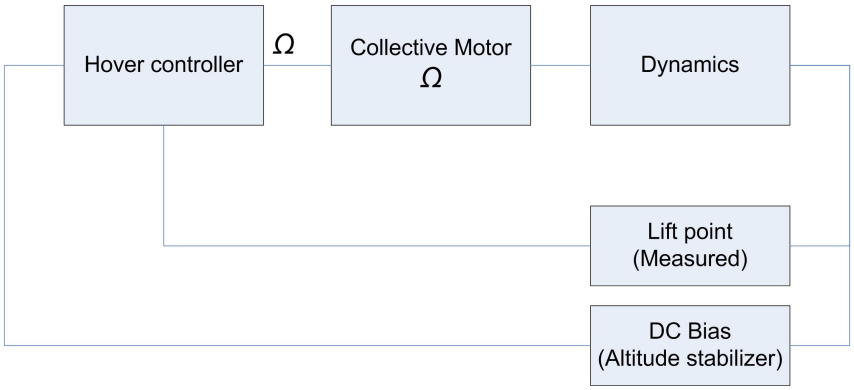


Figure 3.17: Hover controller

Altitude control integration

As with the attitude controller, the integration of the altitude controllers is done through summations. The difference however comes in that the hover controllers output is not differential, but the reference value that determines the central point for the rotor speeds. Figure 3.18 shows the integration of the altitude controllers.

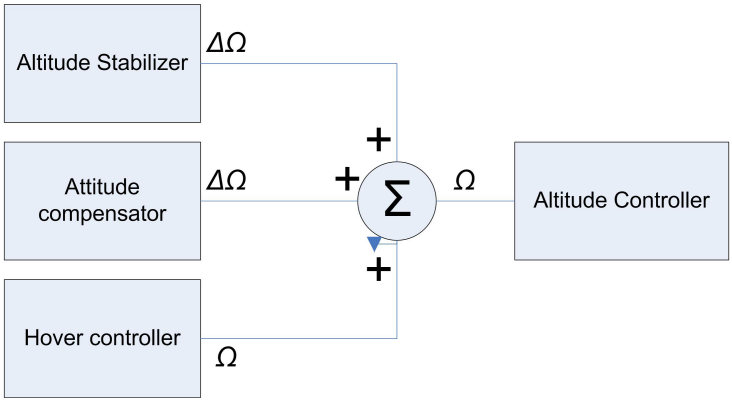


Figure 3.18: Altitude control integration

3.2.5 Controller integration

The final integration of the orientation controller, consisting of attitude control and altitude control, is shown in Figure 3.19.

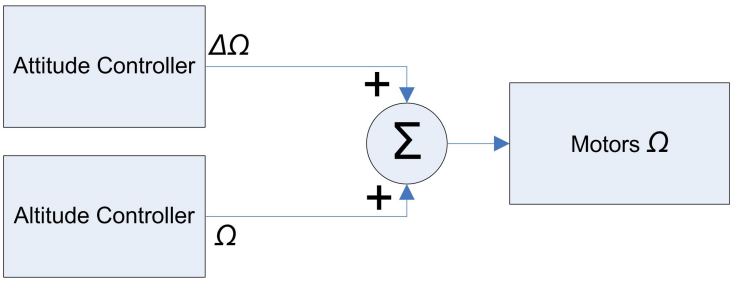


Figure 3.19: Control integration

The final summation provides the inputs to the movement matrix ($E_H(\overline{\xi})$), the control interface to the dynamics of the craft.

3.3 Controller Identification and Characterization

3.3.1 Proportional, Integral and Derivative (PID) Controllers

References: [24]

PID control is a Single Input, Single Output (SISO) linear feedback loop controller. This controller functions by minimising the error between the measured current state and desired states. The response is determined by the proportional (P), integral (I) and derivative (D) terms in the classic three term controller shown in Figure 3.20.

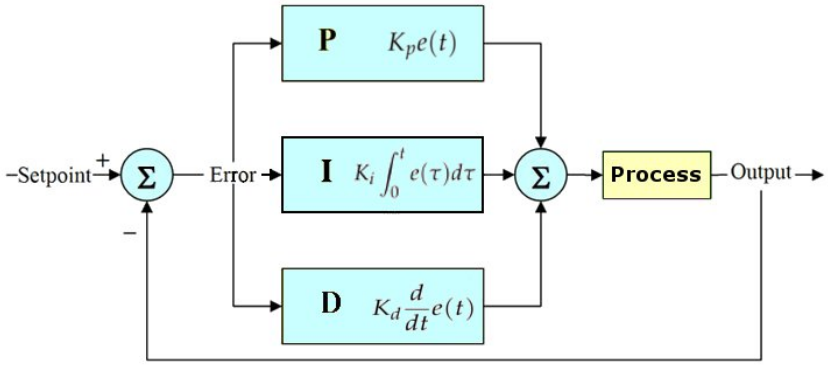


Figure 3.20: Classic three term PID controller

The classic PID controller is characterized by the time domain function shown in equation (3.17), with $e(t)$ the error function.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{d}{dt} e(t) \tag{3.17}$$

Or in Laplace domain form as in equation (3.18):

$$U(s) = \left[K_p + \frac{K_i}{s} + K_d s \right] E(s) \tag{3.18}$$

Each term provides a unique contribution to the overall performance of the controller. Since each term follows a different path, each parameter is decoupled from the others and can be adjusted without affecting the other paths.[†] The terms can be characterized based on their unique contributions. [24]

[†]Note that this does not hold true in certain custom configurations discussed later.

Proportional control is denoted by the P-term and is proportional to the size of the process error. This relationship is purely linear and is the primary control mechanism in any PID controller. The proportional control is represented by equation (3.19) in the time domain.

$$u(t) = K_p e(t) \tag{3.19}$$

Derivative control is denoted by the D-term and can be described as being proportional to the rate of change in the process error signal. By measuring the rate of change an element of prediction is introduced to the controller. This is particularly important in achieving stability in unstable systems as the derivative term introduces a measure of dampening. This compensation for non-linearity makes derivative control critical in many real-world applications. It should however be noted that derivative control is not necessarily the solution to stabilizing an oscillating system. If the control action is in the derivative region, i.e. the control action affects the rate of change in the error signal, then the proportional term provides dampening and an integral term is needed to stabilize the steady state error.[‡] Implementation of derivative control can however be problematic as calculating the rate of change results in noise amplification. Another problem that arises in pure derivative control is that when the system stabilizes at a constant error there is no rate of change and hence no derivative control. Derivative control is represented in equation (3.20) in the time domain.

$$u(t) = K_d \frac{d}{dt} e(t) \tag{3.20}$$

Integral control is denoted by the I-term and is described as being proportional to the integral of the process error signal. Integral control is used to compensate for any steady state offset from the reference signal. It overcomes the shortcomings of proportional control without the need for excessively large controller gain. The integral control term is shown in equation (3.21).

$$u(t) = K_i \int_0^t e(\tau) d\tau \tag{3.21}$$

[‡]This is a common misinterpretation in PID controllers. The differential shift in control action changes the effect each term has, hence it is necessary to understand the way in which control is actuated.

As with any controller, PID control has a few behavioural issues that need to be taken into account to prevent aberrant behaviour. Amongst these issues are *measurement noise*, *proportional* and *derivative kick*, *system non-linearity* and *negative process gain*. Significant *measurement noise* can lead to unstable operation. This can be caused by noise amplification from the derivative term or a lack of filtering on measurement signals. Noise is constant through the proportional term, attenuated by the integral term and amplified by the derivative term. Solutions include the use of a bandwidth limited derivative term or direct measurement of the rate of change.[§] The use of a low-pass filtering and derivative action on the measurement signal cancels each other out. A non-linear median filter is a better alternative, as it is well suited to non-linear feature extraction from noisy signals. Median filters can effectively suppress impulse noise while preserving edge information [27].

The presence of *proportional or derivative kick*, where the controller generates an excessively large response to step inputs due to the large process error, can cause dangerous over actuation in the system. Solutions include limiting the system output (but beware of integral windup) and alternatively moving the proportional and derivative terms into the feedback path, forming a I-PD controller. If the application allows it, another solution is limiting the rate of change in the reference signal by means of a first order hold function.

System non-linearity includes actuator saturation that leads to integral windup. Actuator saturation has a series of consequences, the most prominent is that the system no longer responds to control signal and behaves like an open-loop system. This can result in excessive overshoots or other critical failures. Anti-windup methods can be implemented to prevent this. *Negative process* gain and *model inversion* are both scenarios where the controller causes the opposite response in the system than is needed or expected. The use of multiple PID controllers for various operating points is a common solution to the problems including those of system non-linearity and negative process gain. Figure 3.21 shows the effect of an anti-windup method on a system experiencing

[§]Direct measurement of the rate of change means no differentiation, hence no noise amplification. A standard noise filter is sufficient in this case.

actuator saturation.

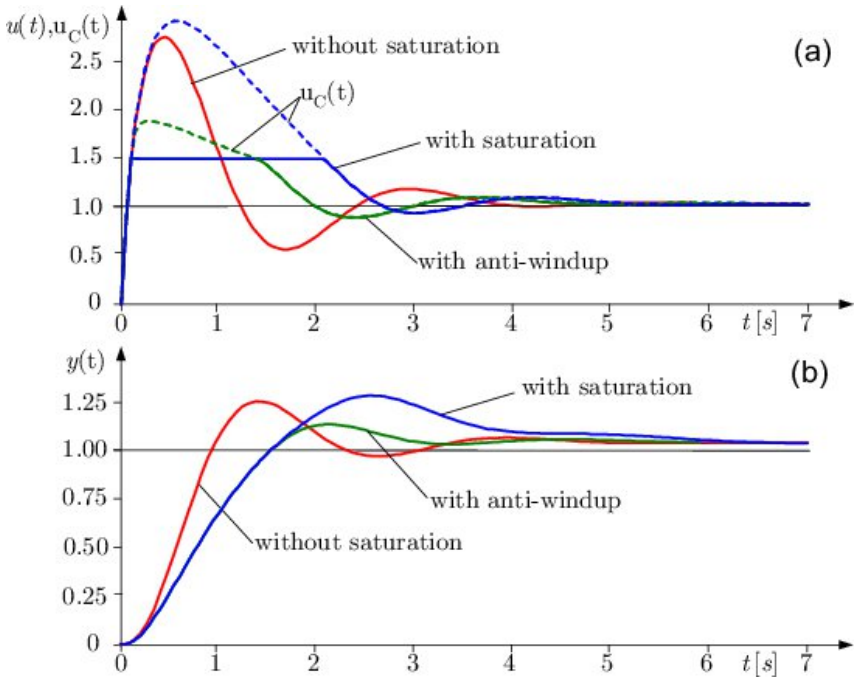


Figure 3.21: Anti-windup protection of PID controller
(a) Actuation, (b) System response

The three term PID controller can be organized into various configurations as shown in table 3.1. Other combinations include series PID controllers as shown in Figure 3.22 with transfer function (3.22).

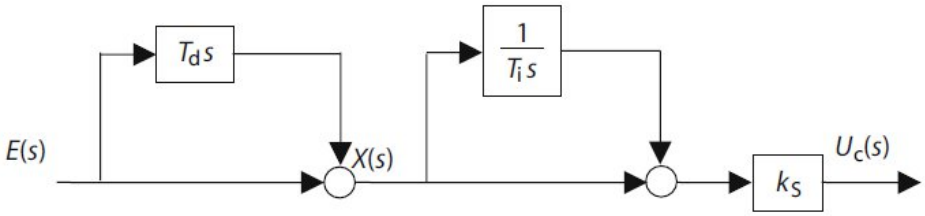


Figure 3.22: Series PID controller

$$U(s) = \left[k_s \left(1 + \frac{1}{T_i s}\right)(1 + T_d s)\right] E(s) \tag{3.22}$$

Table 3.1: Various PID controller structures

Structure	**General notation** $\rightarrow u(t)$	**Laplace notation** $\rightarrow \mathcal{L}\{u(t)\}$
P	$u(t) = K_p e(t)$	$U(s) = \left[K_p\right] E(s)$
I	$u(t) = K_i \int_0^t e(\tau)d\tau$	$U(s) = \left[\frac{K_i}{s}\right] E(s)$
PI	$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau$	$U(s) = \left[K_p + \frac{K_i}{s}\right] E(s)$
PD	$u(t) = K_p e(t) + K_d \frac{d}{dt} e(t)$	$U(s) = \left[K_p + K_d s\right] E(s)$
PID	$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{d}{dt} e(t)$	$U(s) = \left[K_p + \frac{K_i}{s} + K_d s\right] E(s)$
PI-D	$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau - K_d \frac{d}{dt} y(t)$	$U(s) = \left[K_p + \frac{K_i}{s}\right] E(s) - \left[K_d s\right] Y(s)$
I-P	$u(t) = K_i \int_0^t e(\tau)d\tau - K_p y(t)$	$U(s) = \left[\frac{K_i}{s}\right] E(s) - \left[K_p\right] Y(s)$
I-PD	$u(t) = K_i \int_0^t e(\tau)d\tau - K_p y(t) - K_d \frac{d}{dt} y(t)$	$U(s) = \left[\frac{K_i}{s}\right] E(s) - \left[K_p + K_d s\right] Y(s)$

Figure 3.23 shows the normalized response of a generic system to the various combinations of Proportional, Integral and Derivative control.

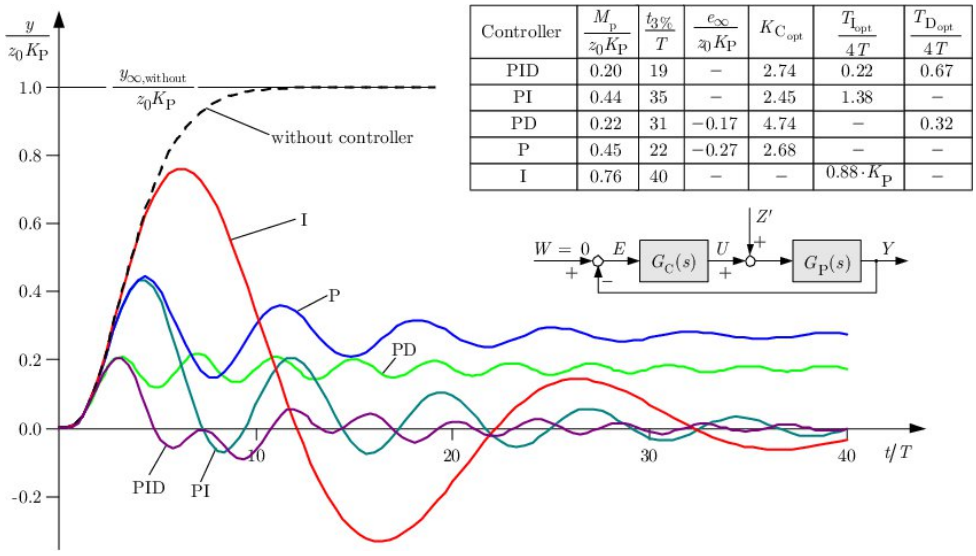


Figure 3.23: Normalized response of various Proportional, Integral and Derivative controller combinations

Improvements in PID controller include the combination of a feed-forward controller with the output of the PID controller. Through a extended knowledge of the system dynamics, the feed-forward controller can optimize performance by adding a sense of prediction in the behaviour of the system. *Gain scheduling* is a process through which PID parameters are changed to adapt to changing operating conditions. This allows for

changes in the response of the controller as the requirement and/or conditions change. Hybrid combinations with other controller types allow for the formation of non-linear PID controllers.

For practical implementation of PID controllers it is more practical to use the time constant form where

$$\tau_i = \frac{K_p}{K_i} \text{ and } \tau_d = \frac{K_d}{K_p}$$

thus changing the equation to the form shown in equations (3.23) and (3.24).[¶]

$$u(t) = K_p \left(e(t) + \frac{1}{\tau_i} \int_0^t e(\tau) d\tau + \tau_d \frac{d}{dt} e(t) \right) \tag{3.23}$$

$$U(s) = K_p \left[1 + \frac{1}{\tau_i s} + \tau_d s \right] E(s) \tag{3.24}$$

There are various methods for developing and implementing a PID controller, but generally the process consists of the following steps:

1. Choose control structure
2. Choose parameters through some selected method
3. Optimize performance

Tuning of PID controllers is usually done through a model, as this is required to determine the optimal parameters. This is however not a requirement for tuning as other methods include the Ziegler-Nicholas methods, training data, or runtime tuning. Many commercial PID controllers have auto-tuning functions and there are various software tools available that can derive the parameters from the model data without the need for mathematical calculation. Optimizing the parameters of PID controllers include the use of evaluation parameters such as overshoot, rise time, settling time, steady state error, stability and reference signal tracking. These parameters are related to the control function terms as shown in table 3.2.

[¶]Note that in the time constant form the control paths are no longer decoupled.

Table 3.2: PID parameters and influence

	Reference tracking		**Disturbance rejection tuning**	
	Step reference		**Constant load disturbance**	
	Transient	Steady State	Transient	Steady state
P	Increasing $K_p > 0$ speeds up response, increases overshoot.	Increasing $K_p > 0$ reduces but does not eliminate steady state error.	Increasing $K_p > 0$ speeds up response, increases overshoot.	Increasing $K_p > 0$ reduces but does not eliminate steady state error.
I	Introducing integral action $K_i > 0$ gives a wide range of response types. Slows response.	Introducing integral action $K_i > 0$ eliminates offset in the reference response. Increases settling time.	Introducing integral action $K_i > 0$ gives a wide range of response types. Minimizes noise disturbance.	Introducing integral action $K_i > 0$ eliminates steady state offsets.
D	Derivative action $K_d > 0$ gives a wide range of responses and can be used to tune response damping. (Decrease overshoot.)	Derivative action has no effect on steady state offset. Decreases settling time.	Derivative action $K_d > 0$ gives a wide range of responses and can be used to tune response damping. (Decrease overshoot.)	Derivative action has no effect on steady state offset.

A basic method to tune a PID controller is as follows [28]. Consider the time constant form shown in equation (3.23).

1. **Initial values**

   Start by setting $K_p = 0.1$ and $\tau_d = 0$. Set $\tau_i >> 0$. By making the integral time constant large, the integral function is effectively removed.

2. **Increase gain**

   Find the largest K_p value that does not result in an overshoot.

3. **Decrease integral time constant**

   Find the smallest τ_i without any overshoot.

4. **Introduce derivative control**
   If the system remains unstable with PI control, then consider introducing derivative control by increasing the value of $\tau_d > 0$.

A few case studies on the implementation of PID controllers on a quad-rotor that can be referred to are as follows. The work of Salih et al. [9] is an example of standard PID control of a Euler angle model. Most notably in the results, is the insufficient dampening resulting in excessive overshoots and oscillations.

Erginer et al. [10] used a PD control approach to control a landing sequence, also on an Euler angle model. Although the trajectory control performance was good, there is excessive oscillation in the orientation of the craft. Since there is no separation between orientation and trajectory control on a system level (regarding physical interpretation), the trajectory controller destabilizes the orientation to maintain the trajectory.

A better approach can be seen in the work of Goel et al. [11], whereby modifying the PID structure based on the function of the controller, the performance is improved. Using PD controller for orientation, PI controller for trajectory and PID for altitude control. This configuration allows for better stabilization of the orientation whilst maintaining good trajectory control.

Interestingly, Tayebi et al. [22] further divides the orientation controller, allowing for separate control of orientation and angular velocity, thus forming PD^2 control structure. This allows for better reference tracking performance. It is also worth noting that Tayebi utilizes a quaternion based model rather than the Euler angle model.

Bresciani [7] uses an modified PID structure that improves the stability of the craft by using the craft's angular rate for derivative control rather that the rate of change in the error signal. This prevents the craft from oscillating at small errors in orientation.

Miller [19] has a stable implementation of PD control for trajectory and PID control for orientation. Miller shows that the use of proper dampening aids in maintaining stability in the presence of various noise and disturbances sources. The effect of uncertainty

in model parameters was studied, but the effects not thoroughly suppressed by the controllers.

Balas [17] has a comprehensive study on the design of an optimal PID controller, as well as a comparison between PID, LQR, H_∞ control and various other methods.

Petersen et al. [14] did a comparative study between PID and LQR with results favouring the use of a PID controller.

Wu [16] did a comparative study using Feedback-Linearisation versus PD control. Wu implemented a modified PD controller that is band limited by including a low pass filter in the controller structure to limit the effects of high frequency noise.

Other papers on PID control include the works of Altu et al. [12], [13], [20].

Pros of PID control:

- Well established and proven control method with vast resources available.
- Easy to implement to functional point.
- Predictable linear behaviour.
- Multiple configurations with various performance characteristics.

Cons of PID control:

- Optimization can proof difficult.
- Optimization requires system model.
- Generally higher sensitivity to parameter changes.
- Limited control over response curve.

3.3.2 Fuzzy Logic Control (FLC)

References: [29], [30], [31]

Fuzzy logic controllers are derived from fuzzy set theory and operate on logic using approximate rather than precise values. The controller is implemented in three stages: *Fuzzification*, *Inference engine* and finally *Defuzzification*. *Fuzzification* is the process that converts the exact input values to logical terms and *Defuzzification* is the inverse. *Fuzzification* and *Defuzzification* is calculated based on the defined membership functions that relate the logic term's weight with the actual value of the variable. The *Inference engine* calculates the output based on a *Rule base* that defines the relation between the logic terms. Figure 3.24 shows the structure of a Fuzzy Logic controller.

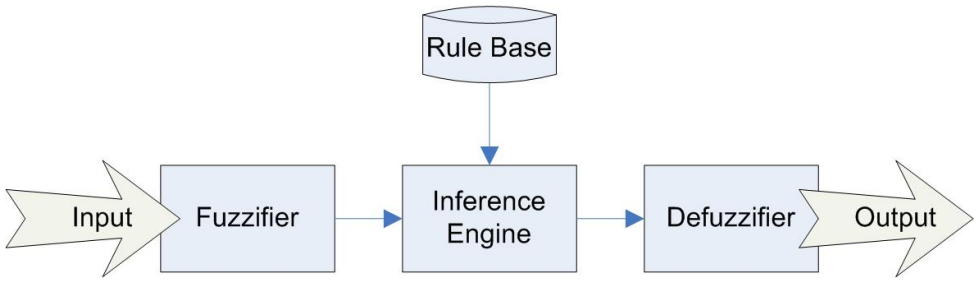


Figure 3.24: Fuzzy Logic controller structure

Fuzzification

Fuzzification is the process through which the crisp inputs are converted into the input fuzzy sets. Each input fuzzy set represents a linguistic variable, with the relationship defined by the input membership functions. The shape of the membership functions used, determine the linearity of the controller. There are various membership function types, as listed below:

- Linear
  - Singleton
  - Triangular
  - Trapezoidal
- Non-linear

- Quadratic
- Gaussian (exponential)
- Special such as cos-function

Figure 3.25 shows the fuzzification process.

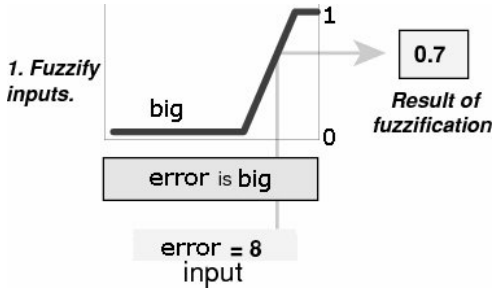


Figure 3.25: Fuzzification of input

Rule Base

The rule base describes the relationship between the various inputs and outputs in linguistic terms, defining a qualitative relationship between two or more variables. Fuzzy rules firing is the actual mathematical/computing process of relating the crisp input to the fuzzy sets, thus forming the antecedent part of the rule base. The fuzzy set is then 'clipped' and related to the consequent part to the extent of correlation defined in the rule base. Thus the output of each rule firing is a clipped or scaled fuzzy set of the output membership function, determined by the correlation between the crisp input and the linguistic fuzzy set.

Inference Engine

The processing of fuzzy rule firing can be done using one of two possible methods:

- Mamdani-type fuzzy processing
- Sugeno-type fuzzy processing

Fuzzy logic control has been around since its introduction in 1965 by Lotfi A. Zafeh, but it was Ebrahim Mamdani who in 1974 proposed a formalized method to process

the fuzzy rule sets. [32] Other inference methods can be used on fuzzy sets, but these two methods are the most popular in application on fuzzy logic controllers. Figure 3.26 shows the fuzzy rule firing within the inference engine.

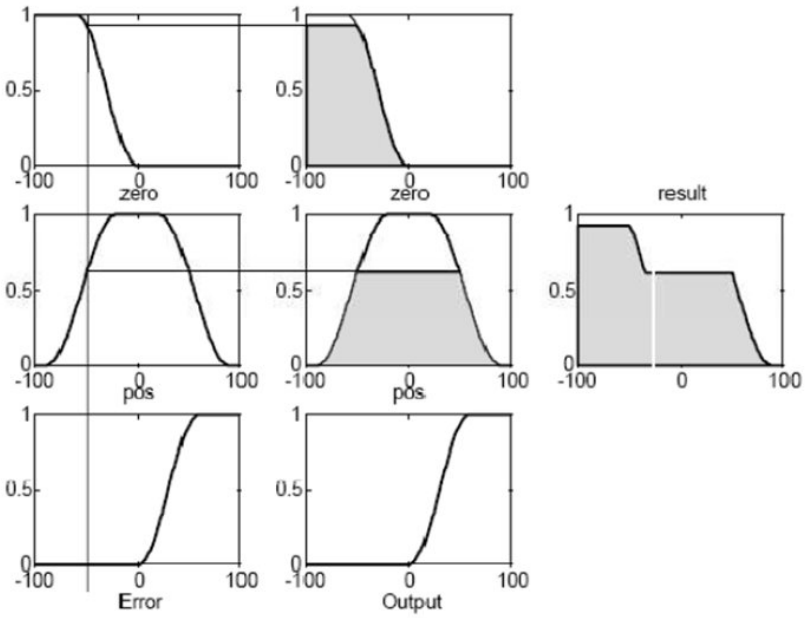


Figure 3.26: Fuzzy Inference engine

Mamdani-type inference engine is the most commonly used form of FLC. Mamdani proposed a variation that accounts for the change in processes error by measuring the change between the current and previous sample value. This provides a sense of prediction in the controller. If for example the process error is negative big, and the change in process error is also negative, the controller will actuate positive big. However if the process error is negative big with the change in process error positive, the actuation would be far less severe to prevent overshoot. The resultant output of the Mamdani controller is the union of each of the resultant fuzzy sets of each rule firing.

The recommended output membership function type for Mamdani type inference engine is singletons, as the improve the efficiency of the defuzzification process.

Sugeno-type inference engine is a variation on the Mamdani-type engine, exactly the same in regards to the fuzzy inference process and input fuzzification. The primary difference is that in the Sugeno controller the output is a direct mathematical function of the inputs, i.e. the output membership functions are either linear or constant. The output membership function is either a singleton or a mathematical function of one. Sugeno-type inference engine is computationally more efficient and ensures that the output surface is continuous.

Defuzzification

Defuzzification is the process of converting the resultant fuzzy set to crisp output values. There are various mathematical methods of defuzzification. With no real advantage of one technique over another, choosing which method to use is up to personal preference. No defuzzification is necessary in Sugeno-type fuzzy controllers. Here are a few defuzzification methods used in Mamdani-type controllers:

- Centre-of-area/gravity
- Centre-of-largest-area
- Mean of maxima
- First-of-maxima
- Middle-of-maxima
- Height

The linguistic variables are much easier to understand and modify that those of PID controllers. Although mathematical modelling is not required to design a fuzzy logic controller, it does help in optimization and simulation.

Thomas et al. [33] published a paper discussing various case studies of FLC, documenting the various pros and cons of FLC. Amongst the case studies of FLC on a quad-rotor is the works of Raza et al. [34] and Younes et al. [35].

Pros of Fuzzy Logic control:

- Easy implementation.
- Design from functional knowledge without model.
- Established control technique.
- Linear or Non-linear behaviour.
- Can mimic PID controller
- Exact control over system response curve.
- Robust behaviour.
- Control of unstable systems.
- MIMO: Multiple inputs, outputs and relationships.

Cons of Fuzzy Logic control:

- Optimization requires expert on system behaviour.
- Sacrifices precision for performance and solvability.

3.3.3 Artificial Neural Network (ANN) Control

References: [29]

An Artificial Neural Network (ANN) is a computational model that simulates the functioning of biological neural networks. It consists of a network of interconnected artificial neurons. Computation is done based on a connectionist approach, where each node is the determined by the sum of the nodes in the previous layer each scaled by a predetermined weight. An ANN is essentially a non-linear MIMO statistical data modelling tool. This results in the creation of adaptive or learning controllers.

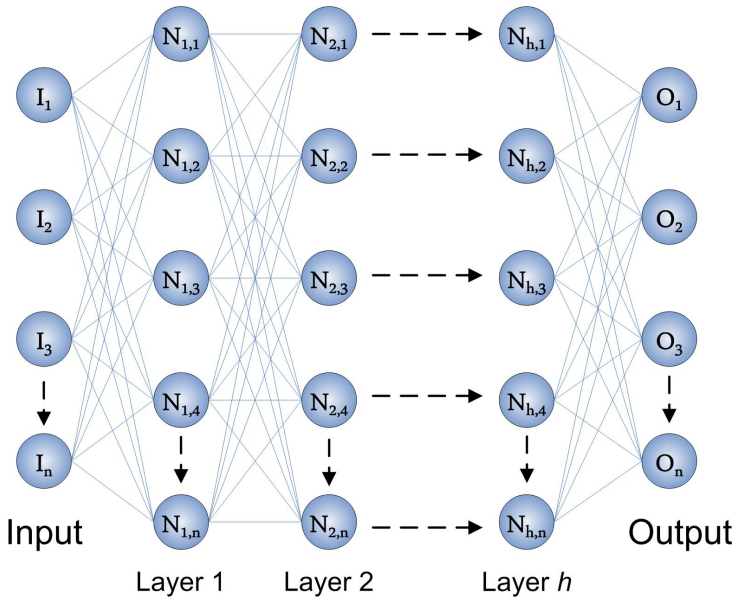


Figure 3.27: Artificial Neural Network (ANN) structure

Figure 3.27 shows the conceptual layout of a typical neural network. The ANN consists of a large number of processing elements, referred to as neurons, that are organized into layers. The response of the neural network is determined by the number of layers and neurons. There is no fixed method to determine the structure of the ANN, however as a general reference the order of the system is related to the number of layers, while the number on neurons determines the number of relationships. Each neuron in each layer is connected to every neuron in the adjacent layers, with each connection characterized by some numeric weight, determining its prevalence in the system. These values called node weights are determined through the learning pro-

cess. There are various paradigms regarding the learning process of neural networks. Neural networks are practical in controlling systems that are too complex to model mathematically, however they require training. Hence it is required to have some form of training data. The ANN can either be trained off-line, on-line or both, in which case it becomes an adaptive controller that continually learns. ANNs are suited to highly non-linear systems with a large number of variables and relationships.

There are various types of ANNs, characterized by learning either in a supervised or unsupervised fashion and being either Feedback nets or Feed-forward-only nets. Each category has multiple network types, defining everything from input and output types, how data is represented inside the network to how data is correlated.

Clearly the structure of ANNs can quickly become very complex, making it difficult to predict the system response. Since the ANN is based on a purely statistical system, there is no control over the associations made within the network. This can lead to aberrant behaviour because of the unbounded nature of ANNs. Hence it is not recommended that a pure ANN be used for an application such as flight control. ANNs can however be combined with other controllers in a strictly bound configuration, such as Neuro-fuzzy control, allowing for adaptive control.

Case studies of ANN being used to control quad-rotors are the works of Dierks et al. [36].

Pros of ANN control:

- Adaptive / learning controller.
- No model needed to develop.
- Multiple Input, Multiple Output (MIMO).
- Good performance in noisy systems.
- Low sensitivity to parameter changes (especially if adaptive).
- Can model highly non-linear systems.

Cons of ANN control:

- Requires training data.
- Can be unpredictable (especially if unbounded).
- Optimal network structure requires expert.
- Implementation can be difficult.

3.3.4 Model Predictive Control (MPC)

References: [37], [38]

MPC is a feed-forward MIMO algorithm for process control. Although generally implemented with feedback (for optimization), the controller functions based on an exciting knowledge of the system dynamics and reacts accordingly, making it inherently a feed-forward controller. MPC relies on a pre-calculated model of the process dynamics, knowing the step response of the system to each input. The variables are classified as either dependent or independent. The independent variables are the inputs to the system, while the dependent variables are the outputs. The model in the controller predicts the change in the dependent for each step change in each independent. Further, the independents are subdivided into control variables and feed-forwards. Control variables (CV) are the independents that are actuated by the controller, while the feed-forwards (FF) are independent that affect the system but can not be controlled by the controller. Feed-forwards can be thought of as the external disturbances to the system.

Figure 3.28 shows an example MPC scheme, showing the prediction and actuation.

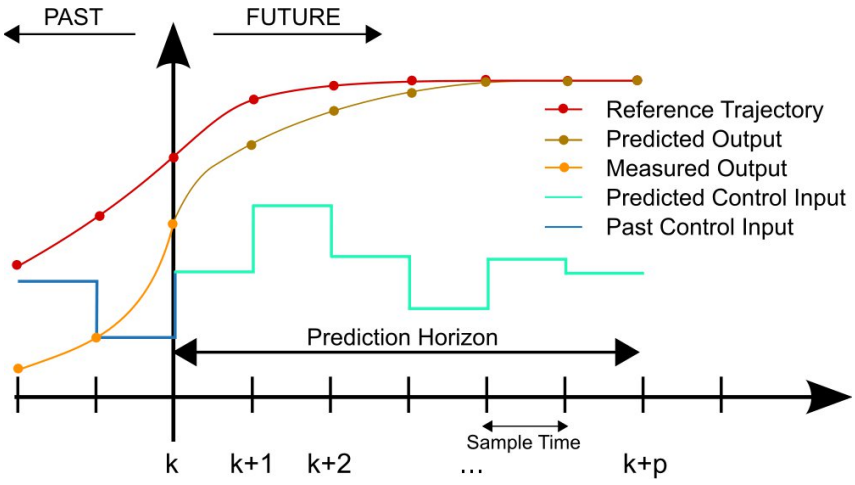


Figure 3.28: A discreet MPC scheme

Typically MPC is used for large systems that are difficult to model mathematically, such as oil refineries and chemical plants. MPC can easily be implemented using empirical models obtained through a statistical analysis of measured system dynam-

ics. Generally MPC is implemented as a linear controller or a series of localized linear controllers. The nominal MPC algorithm is formed as a highly structured convex Quadratic Problem (QP), stored as vectors representing the step response of the system. To calculate the results of a step actuation, the vector representing the response of the given action is added the current prediction.

Although Non-linear Model Predictive Control (NMPC) is possible, it requires a lot of processing since the QP in the predictive model has to be recalculated at each interval due to the adaptive feedback algorithm. This makes NMPC impractical for application as a flight dynamics controller that has to actuate at sub second intervals. It would be better suited for use in the higher control levels such as tactical control, where it is necessary for optimization of multiple cost functions, at a slower actuation speed. As is the case with ANNs, MPC can be combined with other controllers to utilize its advantages for optimization of the base controller, such as a Fuzzy logic controller.

The works of Kim et al. [39] and Dutka et al. [40] can be used as references for case studies on using MPC for flight automation.

Pros of MPC control:

- Predictable and constrained behaviour.
- Multiple Input, Multiple Output (MIMO).
- Adaptive (Can compensate for hardware failures).
- Handling of system time-delays.
- On-line optimization.
- Modelling of linear and quadratic systems.

Cons of MPC control:

- Limited to step actuation.
- Difficult to evaluate performance.

- Stability of optimization algorithm can be problematic.
- Feedback system can be very sensitive to noise.
- Generally not intended for dynamically unstable systems.

3.4 Controller Comparison

Considering the characteristics of the quad-rotor that has been identified and similarly those of the various controllers, a comparison can now be done to determine the suitability of the controller to the control problem.

The PID controller proved to be far more versatile than previously expected. Due to the various configurations and the extended capabilities through techniques such as gain scheduling, the controller can be tailored to the problem as needed. It is easy to implement, however the optimization can prove difficult without a model since the risk is too great to attempt optimization during flight. Although the controller does not perform well in non-linear systems, the cascading of controllers discussed *a priori* provides sufficient linearisation to resolve this issue. Given a thorough understanding of the characteristics of PID control and adaptation techniques, the controller can provide excellent performance and is highly recommended for use on the quad-rotor.

The Fuzzy Logic Controller, although slightly more difficult to implement, provides far greater control over the response curve of the system. It is capable of performing non-linear control and can function with multiple inputs and outputs. Implementation requires a functional understanding, although optimization could provide difficult without a model in this case. Given the customizable response of the controller, it is also recommended as a possible solution for the control of the quad-rotor.

The use of an Artificial Neural Network (ANN) is however not recommended since the response is too unpredictable. Providing accurate training data, implementing and sufficient bounding to maintain controller stability can prove difficult. Although it is possible to utilize this kind of controller for flight control as demonstrated by the case study, it is better suited for use in conjunction with another controller to provide adaptive capabilities.

A Model Predictive Controller can provide rather impressive performance with regard to optimization, but is not intended for use on dynamically unstable non-linear sys-

tems such as the quad-rotor. MPC does present some interesting possibilities for the higher control levels, but does not seem practical for control of the flight dynamics. Implementation of a MPC controller on such an unstable non-linear system would be very difficult.

Table 3.3 shows a summary of the system level comparison between the various controllers. Each parameter is given a weight and the evaluated, giving it an arbitrary score out of 10. The higher the score, the better suited the controller is considered to be.

Table 3.4 shows a summary of the technical level comparison between the various controllers. In this table the controllers are compared based on their technical characteristics.

Table 3.3: Controller comparison: System Level

Controller Type	Maintainability	Sustainability	Testability	Usability	Reliability	Score (10)
Weights:	$W_1 = 0.15$	$W_2 = 0.2$	$W_3 = 0.1$	$W_4 = 0.3$	$W_5 = 0.25$	$\Sigma = 1$
PID	7	8	5	8	7	7
FLC	8	8	5	9	9	8
ANN	5	7	6	3	4	5
MPC	5	7	7	4	5	5

Table 3.4: Controller comparison: Technical Level

Controller Type	Input Output Type	Linearity	Sensitivity to parameter changes	Noise Sensitivity	Requirements	Recommended?
PID	SISO	Linear	Medium - High	Medium	Model recommended	✓
FLC	MIMO	Linear / Non-linear	Medium	Medium	Functional analysis & Operational subspace	✓
ANN	MIMO	Non-linear	Low (Adaptive)	Low	Training data	×
Model Predictive Control	MIMO	Linear	High (Can be adaptive)	Medium	Model or Training data	×

Chapter 4

Simulation

---

*"**Simplicity is the ultimate sophistication.**" - Leonardo DaVinci.*

4.1 Controllers

Based on the results obtained in the previous section, the PID and Fuzzy Logic (FLC) controllers will now be simulated. Three variations of the PID controller is simulated with one variation of the FLC.

4.1.1 Orientation Controller

The orientation controller is used to demonstrate the method of selecting a controller based on technical performance. The controllers to be tested are the PD, PID, PI+D and Fuzzy logic.

PD Controller

The PD controller used in this system has a modified structure that measures the derivative action directly from the system, rather that determining the rate of change in the error signal. This allows for the better control of the crafts stability, improving the dampening of the angular velocities and reducing oscillations at steady state. Figure 4.1 shows the modified PD controller structure.

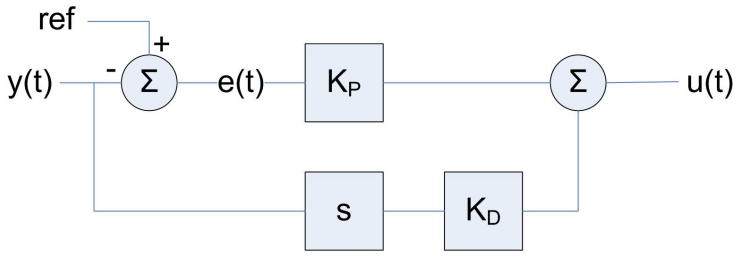


Figure 4.1: PD controller structure

PID Controller

The next controller is the PID controller, which a again uses the modified derivative structure. A saturation block is added to prevent integral windup from actuator saturation. The PID controller is actually implemented in the time constant form, simplifying the tuning of the controller. Figure 4.2 structure of the PID controller.

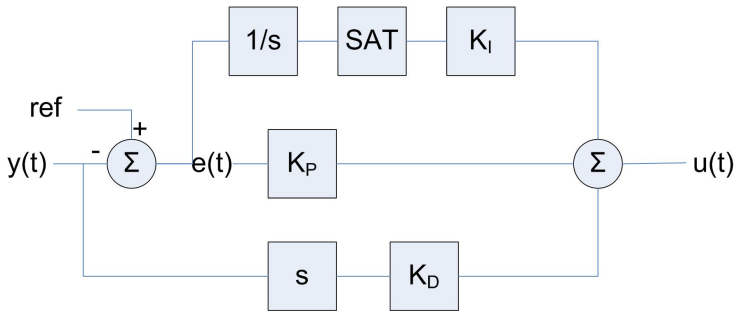


Figure 4.2: PID controller structure

PI+D Controller

The PI+D controller is similar to the PD controller, but with an additional integral path included after the proportional term. Again the controller has the modified derivative structure. Figure 4.3 shows the PI+D controller structure.

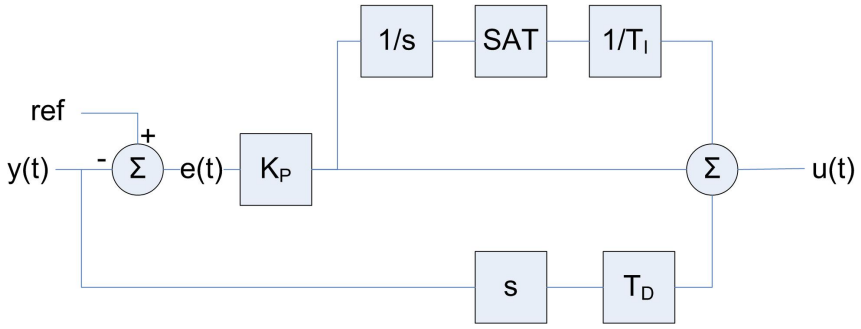


Figure 4.3: PI+D controller structure

Fuzzy Logic Controller

The Fuzzy Logic (FLC) controller used in this simulation is initially based on the PID controller and then modified to give a non-linear control behaviour. Table 4.1 shows the rules and Figure 4.4 shows the various membership functions of the FLC. Figure 4.5 shows the response curve of the FLC given the various angle error and angular rate in radians. The controller is configured to damp only at smaller angle errors.

Table 4.1: Fuzzy Logic Control Rules

If	(Constraint)	then	(Action)	(weight)
If	(Error is Negative)	then	(Control is Negative)	(1)
If	(Error is Zero)	then	(Control is Zero)	(0.5)
If	(Error is Positive)	then	(Control is Positive)	(1)
If	(Error is Zero) and (Rate is Positive)	then	(Control is Negative)	(0.55)
If	(Error is Zero) and (Rate is Negative)	then	(Control is Positive)	(0.55)

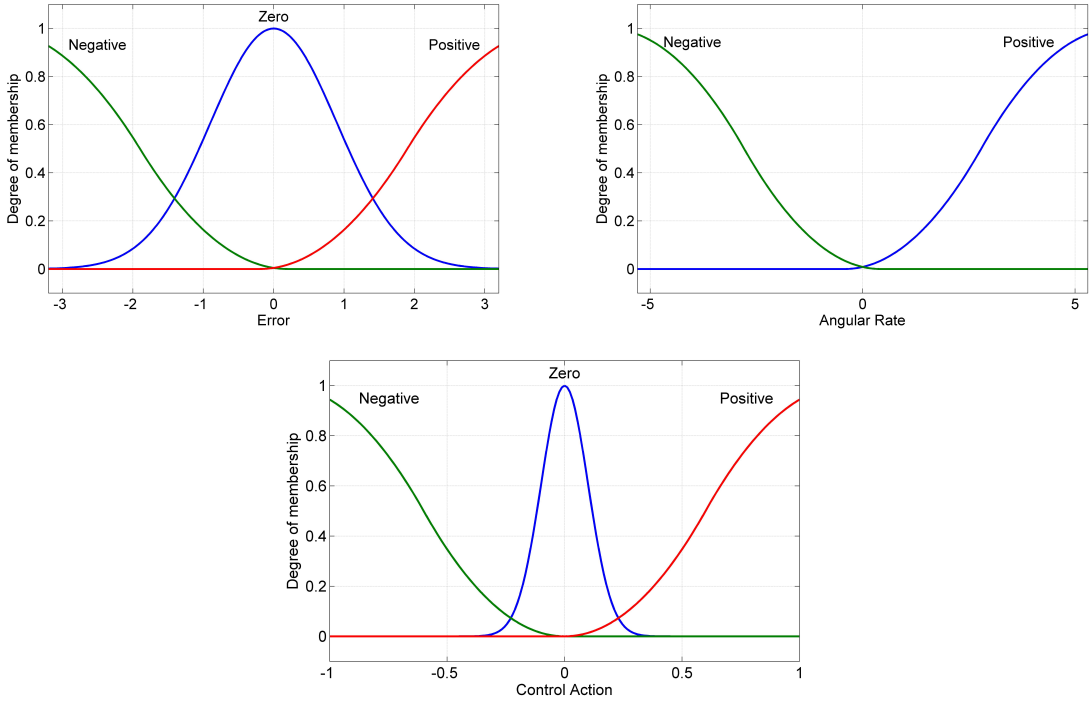


Figure 4.4: FLC membership functions

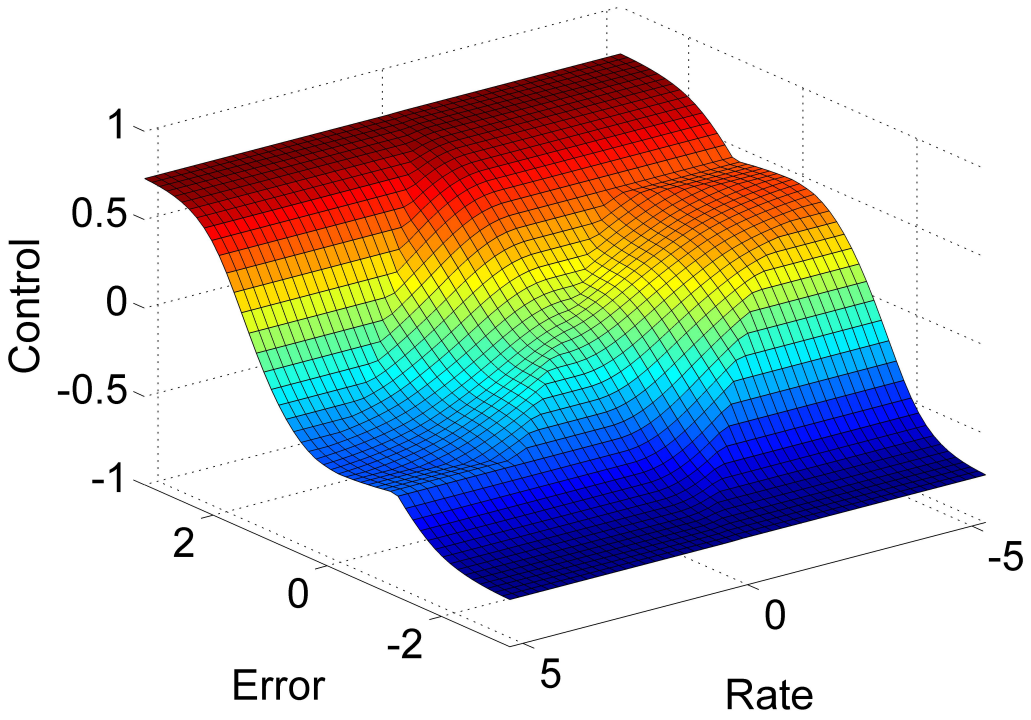


Figure 4.5: FLC response curve

4.2 Comparative Simulator

The comparative analysis of the controller performance is done using a single axis simulator as this simplifies the interpretation of the results. The pitch controller is used for this analysis. The additional forces such as the gyroscopic effects and Coriolis effect are omitted at this stage.

In order to evaluate the sensitivity of the controller the model parameters are adjusted and the simulation repeated. The deviation in response performance is used to measure the sensitivity. The mass properties, i.e. net mass and rotational inertia, is used as the evaluators. The effect of a parameter change is linearly related to the other parameters. Doubling the inertia has the same effect as halving the efficiency of the thrust, or halving the dimensions of the craft. The mass properties were chosen as they represent the parameters which are most likely to change during normal operation, since they are dependent on payload and fuel loads. Due to the linear nature of the relationships between the parameters, if the controller can continue to function with a given change in one parameter, it can be assumed that the controller can function with a similar change in any other parameter. There are however physical limitations, for example the mass can not exceed the lift capabilities of the craft, for this would defy any practical constraints. The scaling is also considered to be symmetric, i.e. that force and mass symmetry must remain to maintain the linearity of the model.

The next step is to test the sensitivity of the controllers to the various noise signals and disturbances present in the system. The simulator takes into account various disturbance sources, including measurement noise, actuator disturbances and load disturbances.

The final step in the comparative simulation is to evaluate the reference tracking performance. To ensure that the true performance is evaluated, the controller is presented with an extreme case on a non-linear sinusoidal signal, swinging from one extreme to the other.

4.2.1 Structure

The simulator is divided into various sections, similar to blocks in the control loop. Figure 4.6 shows the structure of the simulator.

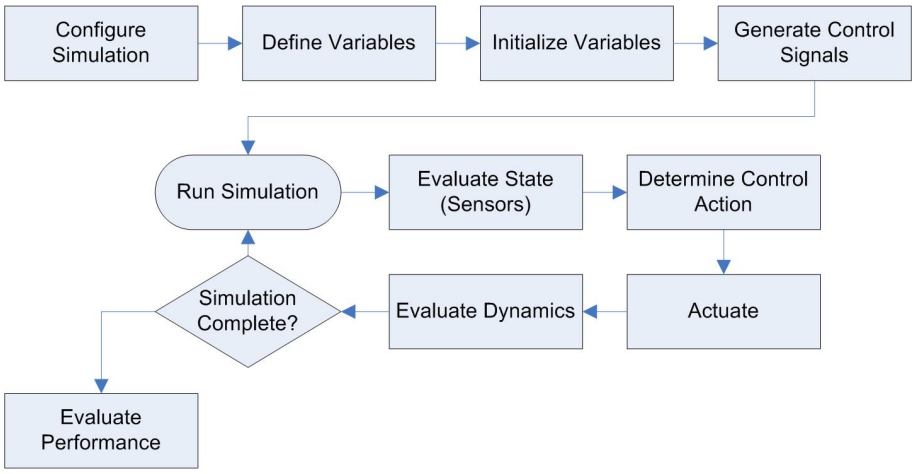


Figure 4.6: Simulator structure

The simulator consist of three major sections, the configuration, simulation and the evaluation.

The configuration section constitutes the following steps:
Configuration: This block does the initial configuration of the simulation, including run time, time step and control intervals.

Variable definition: In this section, the various model parameters are defined and the matrices generated.

Initiation of variables: The various variables used in the simulation are initiated here, defining their sizes and initial values. The noise signals are also generated in this block.

Control signal generation: In this block, the various reference signals are defined.

The simulation section consists of the following steps:
Evaluate State: This block is used to determine the state of the craft, i.e. orientation, velocity and position. This is used to calculate the error signals that are fed into the

controllers.

Determine control action: In this block the controllers and related algorithms are implemented and the actuation in determined.

Actuation: The actuation block is the physical simulation of the controller actuation instructions.

Evaluate Dynamics: The dynamics block simulates the systems response to the actuation.

The performance evaluation section consist of the visualization and analysis tools. The performance indices, step response characteristics, etc. are determined here.

4.2.2 Noise

Although the step response of the controller is important, the true measure of a controller is its robustness, which also includes its ability to withstand noise. In an application such as flight control there is many sources of noise as defined in Section 3.1.4.

The simulation uses low frequency Gaussian white noise to simulate the disturbances. The noise signals are directly introduced into the system without any filtering. The reason for this is to test the controller's robustness, rather than the efficiency of the filtering techniques. The amplitude of the the noise is therefore limited to a reasonable level.

The measurement noise is considered to be within 1.5 degrees. The measurement noise is representative of the noise present in the sensors and the errors introduced by body vibration etc. Figure 4.7 shows an example of the typical measurement noise as simulated.

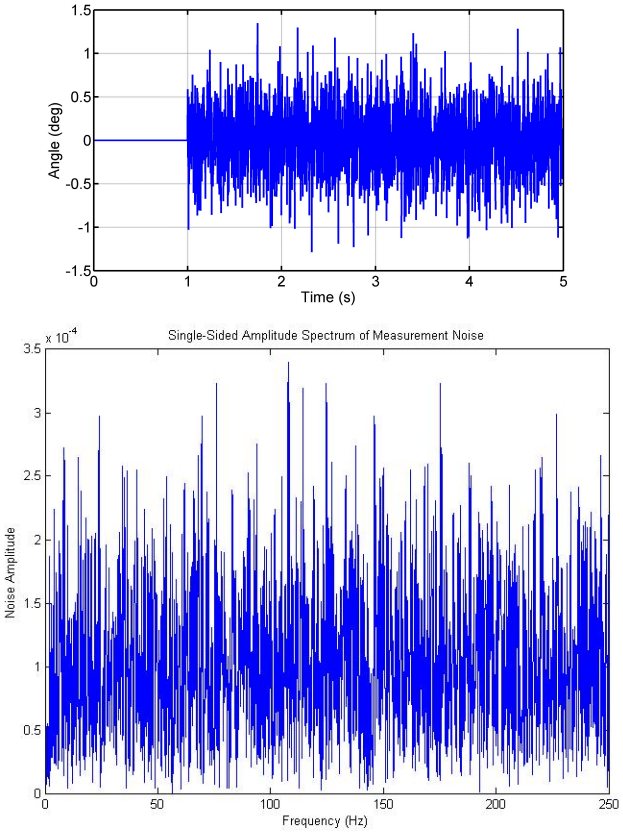


Figure 4.7: Measurement noise

The load disturbances are considered to be less than 0.5 degrees. This represents integration errors and accumulative effects. Figure 4.8 shows the typical load disturbances.

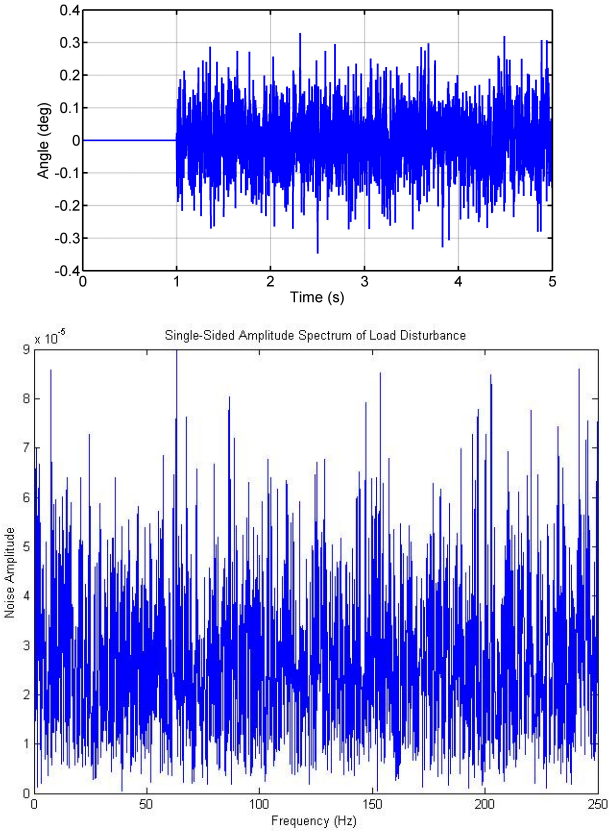


Figure 4.8: Load disturbance

The actuator disturbance on the other hand is very large, up to 30% of the thrust capability of the rotors. This was chosen so as to simulate the unpredictable nature of atmospheric conditions including wind, vibrations, and variations in motor efficiency. This also has an accumulative effect on the system. Figure 4.9 shows the typical actuator disturbance used in the simulation.

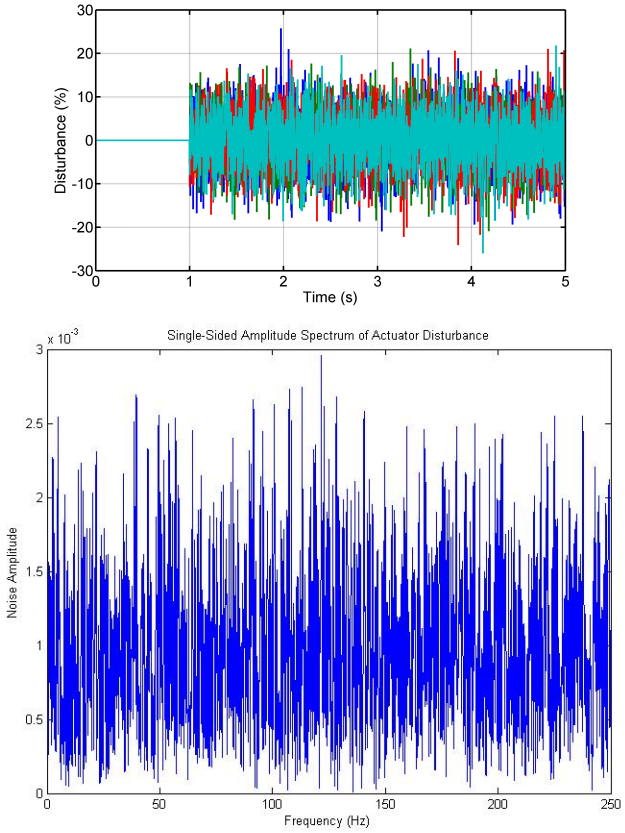


Figure 4.9: Actuator disturbance

4.3 Simulator (Full model)

The full model simulation shows the interaction of the various controllers. In this case a sequence of combined manoeuvres are done to demonstrate the collaboration of the controllers. The simulation is done only in the hybrid reference frame, with the position being calculated using the earth reference frame and the orientation being modelled in the body reference frame to account for angular interpretation of forces.

Chapter 5

Simulation Results

---

*"**Progress is man's ability to complicate simplicity.**" - Thor Heyerdahl*

5.1 Orientation Control

5.1.1 Parameter Scaling

To demonstrate the orientation control, the pitch controller is used. Given the fact that positive and negative actuation is possible, the controller can be allowed to have a small overshoot. The flight envelope is bounded to a pitch angle between -45° and 45°, giving an effective thrust of 70% for both vertical and horizontal components. The simulation is run given the extreme case of an angle change from one extreme to the other, thus a 90°, or 1.5708 *rad*, step angle change.

The performance requirements for a 90° step change are:

- Percentage overshoot $< 2\%$
- Rise time < 1s
- Settling time < 2s
- Steady state error < 0.2

Each of the controllers are simulated given a parameter change by a factor of 2, i.e. the parameter value can double or half that of the nominal value. The scale change is applied to the mass properties of the craft, i.e. both the mass and rotational inertia is scaled.

Figure 5.1 shows the 90° step response of a PD controller with gain constants $K_p = 0.4$ and $K_d = -0.00025$.

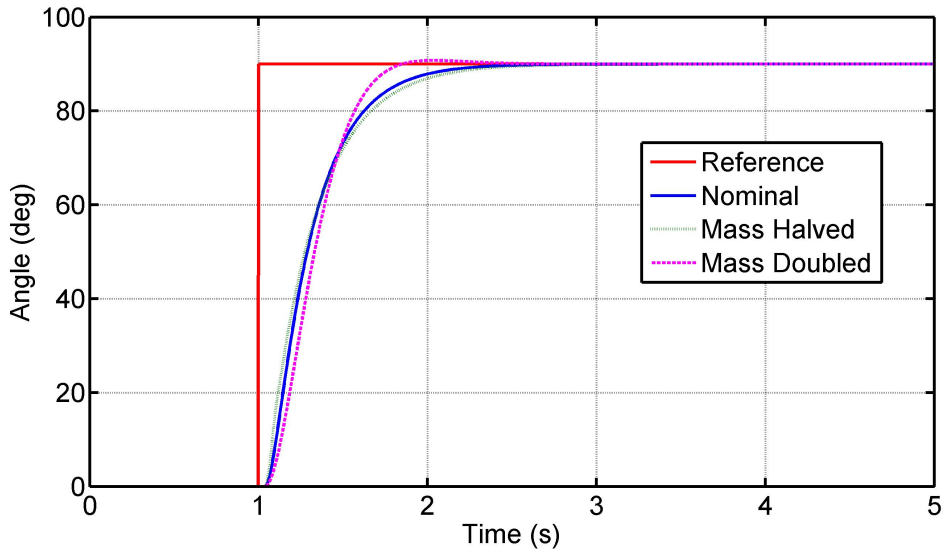


Figure 5.1: PD Controller step response

Table 5.1 shows the results of the performance analysis for the PD controller.

Table 5.1: PD Controller Performance

Scaling	T_r (s)	T_s (s)	PO (%)
0.5	0.698	1.152	0
1	0.650	1.042	0
2	0.584	0.746	0.8628

The controller stayed within specification for all three scaling factors, proving that the controller has a sufficiently low sensitivity to parameter changes.

Figure 5.2 shows the step response of the PID controller with gain constants $K_p = 0.3$, $t_i = 200$ and $t_d = -0.0008$.

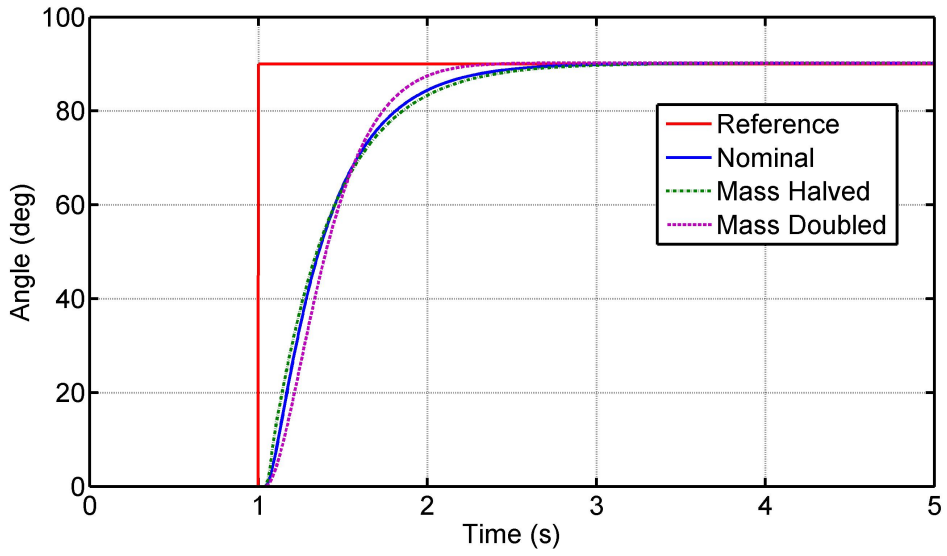


Figure 5.2: PID Controller

Table 5.2 shows the results of the performance analysis of the PID controller.

Table 5.2: PID Controller Performance

Scaling	T_r (s)	T_s (s)	PO (%)
0.5	0.892	1.458	0.2046
1	0.846	1.352	0.2058
2	0.760	1.056	0.2321

The PID controller also conforms to the sensitivity requirements.

Figure 5.3 shows the response of the PI+D controller, with gain constants $K_p = 0.4$, $Ki = 0.04$ and $Kd = -0.00025$.

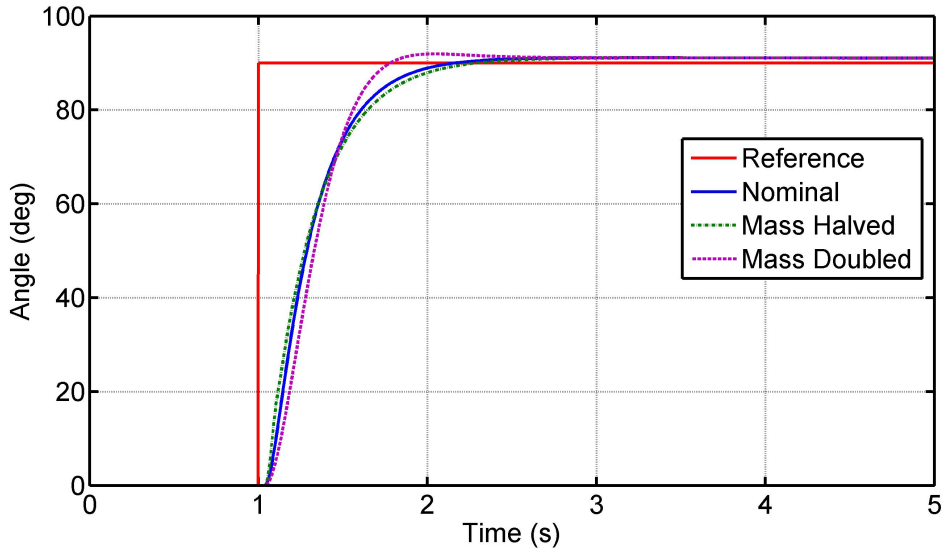


Figure 5.3: PI+D Controller

Table 5.3 shows the results of the performance analysis.

Table 5.3: PI+D Controller Performance

Scaling	T_r (s)	T_s (s)	PO (%)
0.5	0.674	1.024	1.2251
1	0.630	0.932	1.2395
2	0.572	1.176	2.1604

Given a mass double that of the nominal value, the PI+D controller just barely exceeds the specified overshoot. The controller also suffers from a steady-state error introduced by the series integral term. The controller will function (although with a steady state error), but given that the previous two controllers stayed within the specifications this controller would not be the ideal choice regarding stability.

Figure 5.4 shows the step response of the Fuzzy Logic controller. The FLC is configured to be non-linear, damping only when the error approaches zero. This can be seen in the change in rate of the response curve as the angle passes 60° on the graph, i.e. the error is less that 30°, alter which the dampening is in effect.

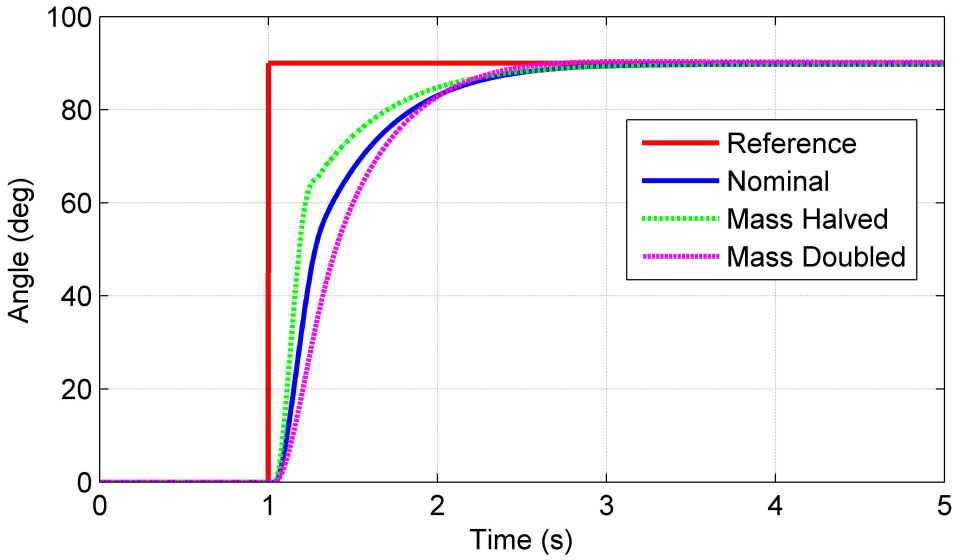


Figure 5.4: Fuzzy Logic Controller step response

Table 5.4 shows the results of the performance analysis of the FLC.

Table 5.4: Fuzzy Logic Controller Performance

Scaling	T_r (s)	T_s (s)	PO (%)
0.5	0.754	1.4980	0
1	0.896	1.538	0
2	0.928	1.362	0.3565

To compare the overall performance of the controller's step response, the IAE, ISE, ITAE and ITSE performance indices can be used. The smaller the performance index value, the better the controller performed. These values are calculated using the angular error in radians. Table 5.5 shows the step response performance indices of each controller for a nominal airframe.

According to these performance indices the best controllers are either the PD or PD+I controllers. The difference in step response performance is minimal. In this comparison the IAE and ISE performance indices take priority as the overall response is being considered.

Table 5.5: Controller step response performance comparison

Controller	IAE	ISE	ITAE	ITSE
PD	0.5207	0.5091	0.6545	0.5814
PID	0.6575	0.6183	0.8892	0.7307
PI+D	0.5536	0.5061	0.8047	0.5789
FLC	0.6310	0.5426	0.8842	0.6358

5.1.2 Noise Performance

The next simulation is to compare the performance of each controller when noise is introduced. The nominal mass is used and each controller is simulated using the exact same Gaussian white noise patterns. This simulation incorporates measurement noise, actuator disturbances and load disturbances. Figure 5.5 shows the results comparing the performance of each controller to the same noise.

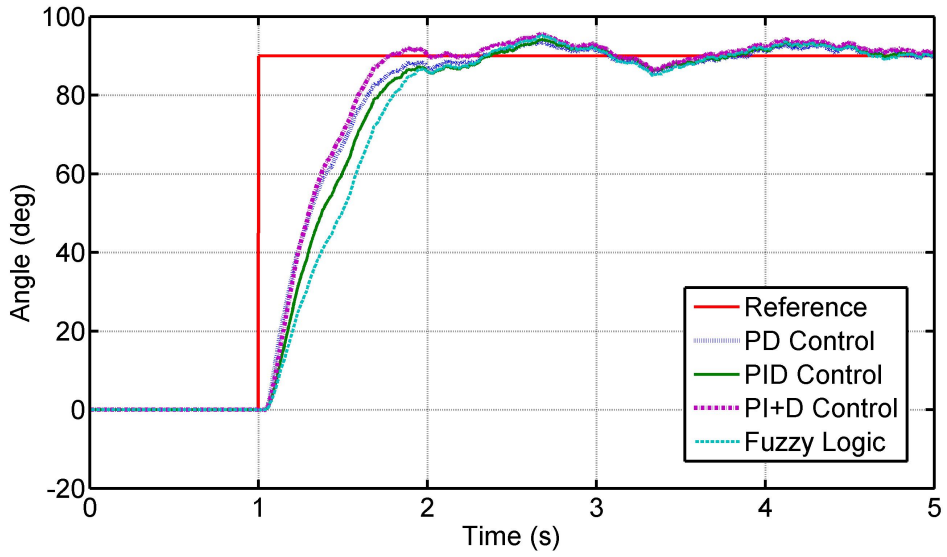


Figure 5.5: Noise performance comparison

Table 5.6 shows the results of the performance analysis of the controller performance in the presence of noise.

Table 5.6: Controller noise performance comparison

Controller	IAE	ISE	ITAE	ITSE
PD	0.7085	0.5488	1.2949	0.6699
PID	0.8273	0.6656	1.4802	0.8300
PI+D	0.7106	0.5468	1.3200	0.6714
FLC	0.9222	0.7900	1.6220	1.0027

The results of this simulations indicates the PD might be the better option for this case study. For this comparison the ITAE and ITSE performance indices take priority as the steady state response ($t \gg 0$) to the noise is the primary concern.

5.1.3 Reference Tracking

The final step in the simulation of the orientation controller is to evaluate the reference tracking performance of the controllers. For this simulation an extreme case is given to push the controller to its limits, making it easier to evaluate the performance of the controller. The reference signal is a non-linear sinusoidal function, making a 120° swing in 1.2 seconds. Figure 5.6 shows the reference tracking performance of the various controllers.

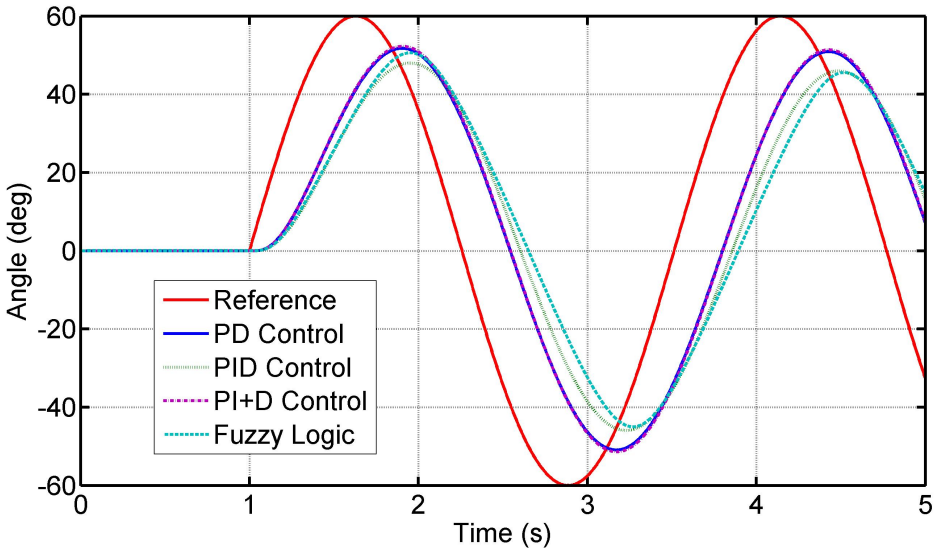


Figure 5.6: Reference tracking comparison

Table 5.7 shows the performance indices for the reference tracking comparison.

Table 5.7: Controller reference tracking performance comparison

Controller	IAE	ISE	ITAE	ITSE
PD	1.6828	0.8828	5.2114	2.8056
PID	1.9178	1.1517	5.9318	3.6505
PI+D	1.6962	0.9760	5.2599	2.8586
FLC	2.0444	1.3058	6.3561	4.1617

Again the results indicate that the PD controller is the optimal choice. In this com-

parison, the IAE and ISE performance indices are the most relevant, as the tracking performance is equally critical irrelevant of time. Figure 5.7 shows the reference tracking performance of the PD controller given a more reasonable reference signal. In this case the reference signal makes the same 120° swing, but over 2.4 seconds. There is a slight actuation delay, which could be reduced by decreasing the dampening in system. This would however increase the sensitivity of the controller and would not completely remove the delay.

Throughout all the simulations of the orientation controller, the PD controller performed best, thus making it the logical choice for the orientation controller.

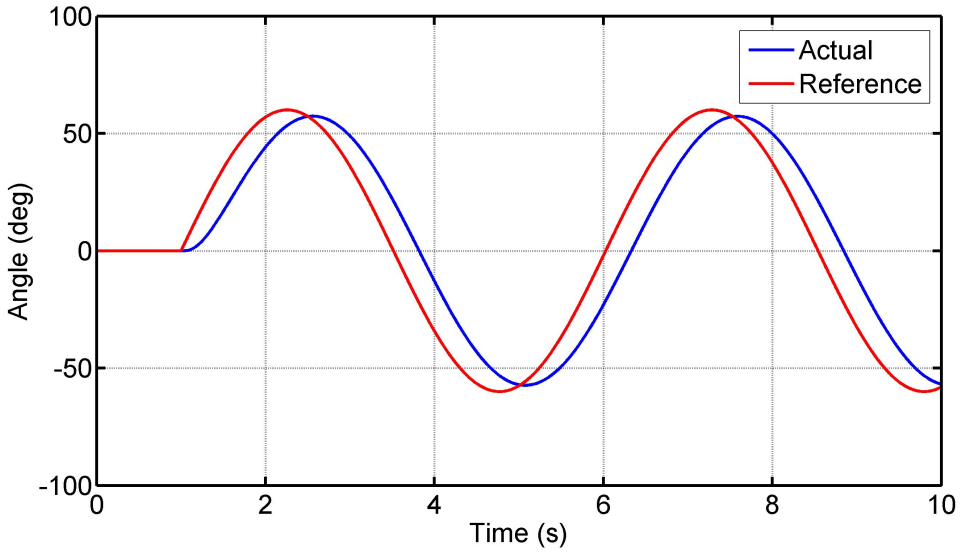


Figure 5.7: PD Controller reference tracking

5.1.4 Angular control interpretation

The issue of angle interpretation and the non-linearity of passing through the zero point was discussed in Section 3.1.4. In summary, to get the controller to perform the optimal actuation it is necessary to follow the shortest route for rotation, i.e. that the controller should never make a turn larger than 180° as turning in the other direction would then be shorter.

To demonstrate this, the yaw controller is used, as this is the most likely scenario to be encountered. Figure 5.8 shows the yaw controller without the angular interpreter. Figure 5.9 shows the yaw controller with the angle interpreter. Note that the controller now follows the shortest route, which requires that the controller pass through zero. In the first case the controller turns from 120° to −120° through a full 240°. In the second case the controller only has to turn through 120°.

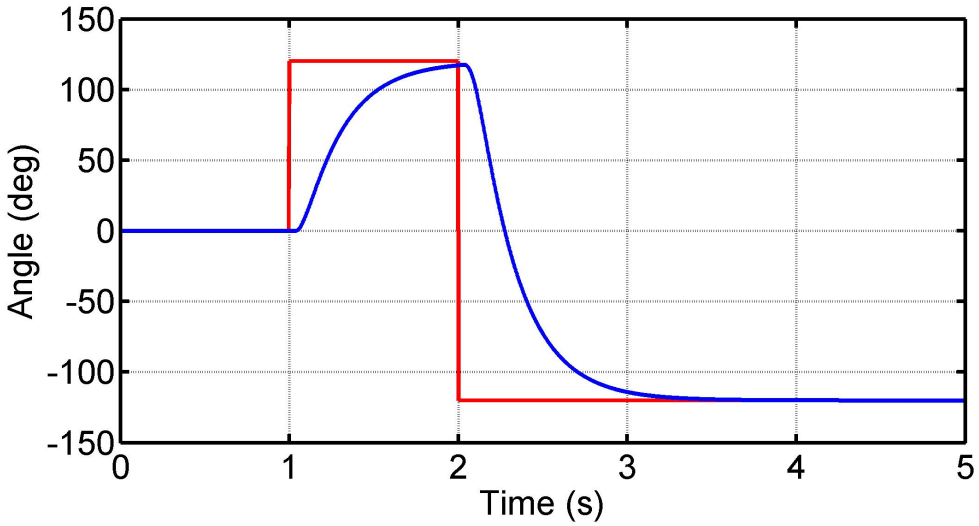


Figure 5.8: Yaw control without angle interpretation

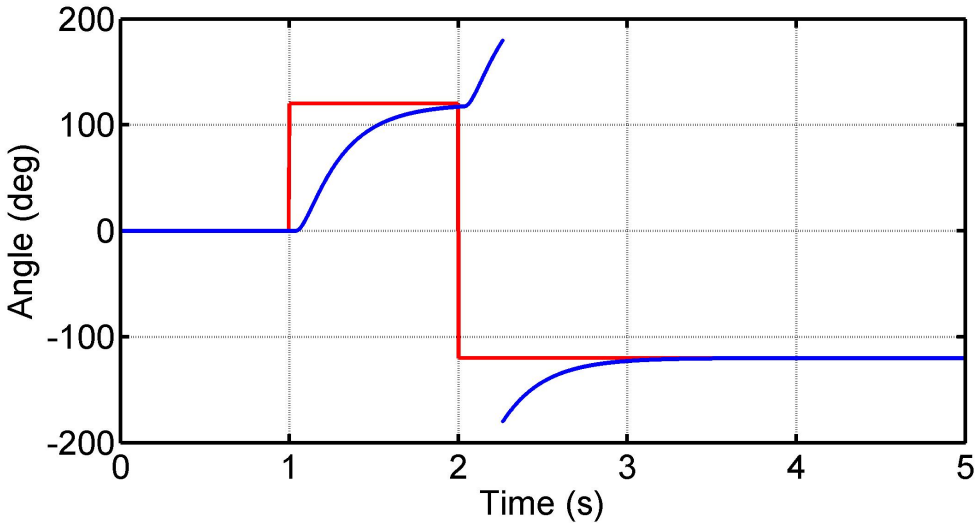


Figure 5.9: Yaw control with angle interpretation

5.2 Altitude Control

Single Altitude Controller

The single altitude controller performs well when analysing the step response, but once the controllers start to interact the limitations are clear. Figure 5.10 shows an unbounded altitude controller that saturates the actuators, resulting in a loss in orientation control as shown in Figure 5.11. The controller remains stable until the pitch angle exceeds 60°, resulting in an over actuation and subsequently actuator saturation.

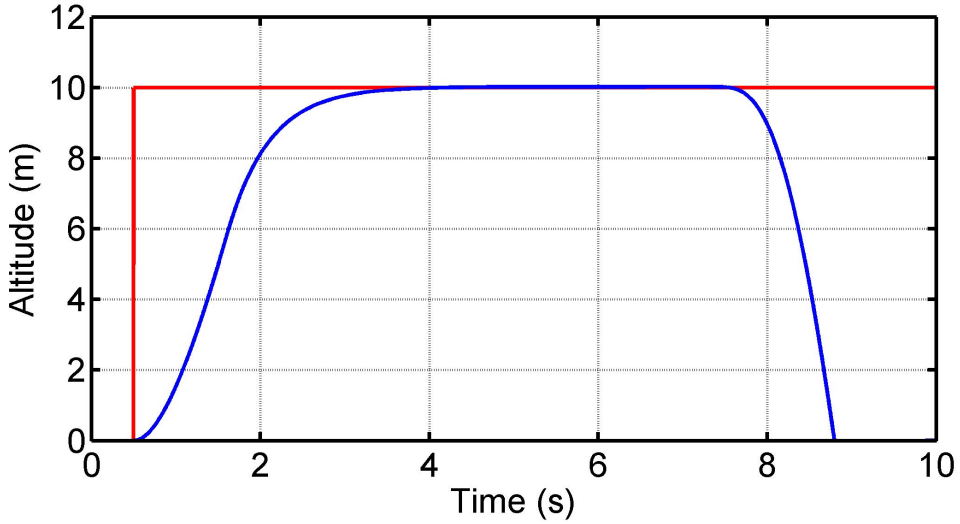


Figure 5.10: Unbounded altitude controller failure

Figure 5.12 shows the same controller, but with bounded actuation. With the orientation controller unaffected, the craft can recover with only limited altitude loss.

The controller is now tested with a sequence of orientation changes, first the pitch is increased to 45°, then to 60° and for the final step, both the pitch and roll are increased to 60°. According to the model the controller must be able to maintain altitude for the first two cases, but loose altitude on the final combined step. Figure 5.13 shows the altitude controller performance, clearly indicating the stability and altitude loss for the final step. The actuation of the controller can be seen in Figure 5.14, showing peaks when compensating for the changes in orientation.

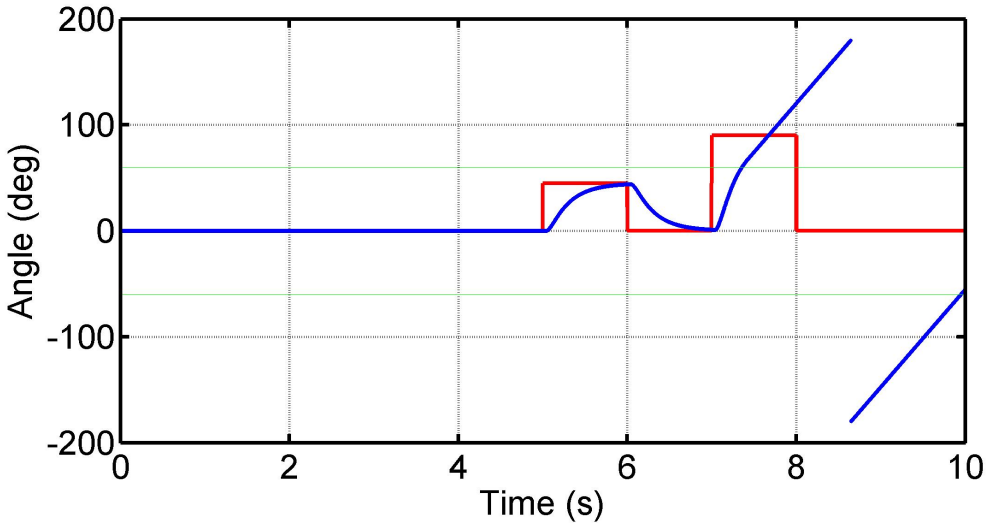


Figure 5.11: Unbounded altitude controller destabilizing pitch control

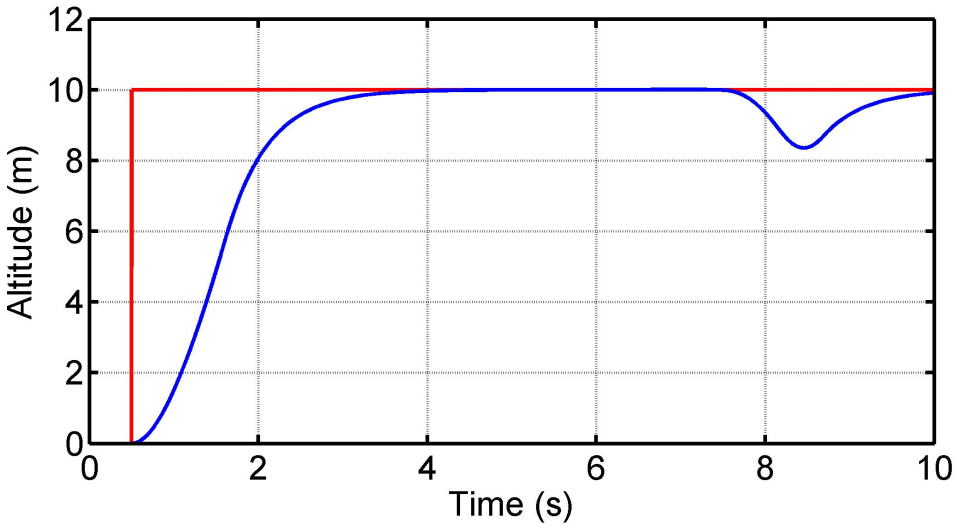


Figure 5.12: Bounded altitude controller performance

Attitude Compensator

The use of the altitude stabilizer in conjunction with the attitude compensator allows for more complex adaptation of model characteristics. The attitude compensator is a feed-forward controller that compensates for the change in orientation. This allows for individual tuning of the response characteristics of each controller, where the altitude stabilizer and hover controller has slow response characteristics, the attitude compen-

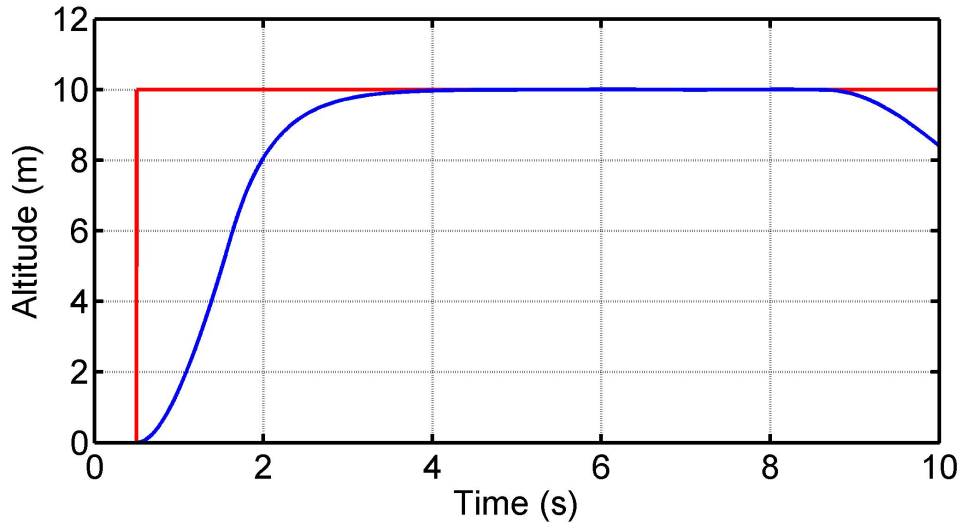


Figure 5.13: Bounded altitude controller attitude change test

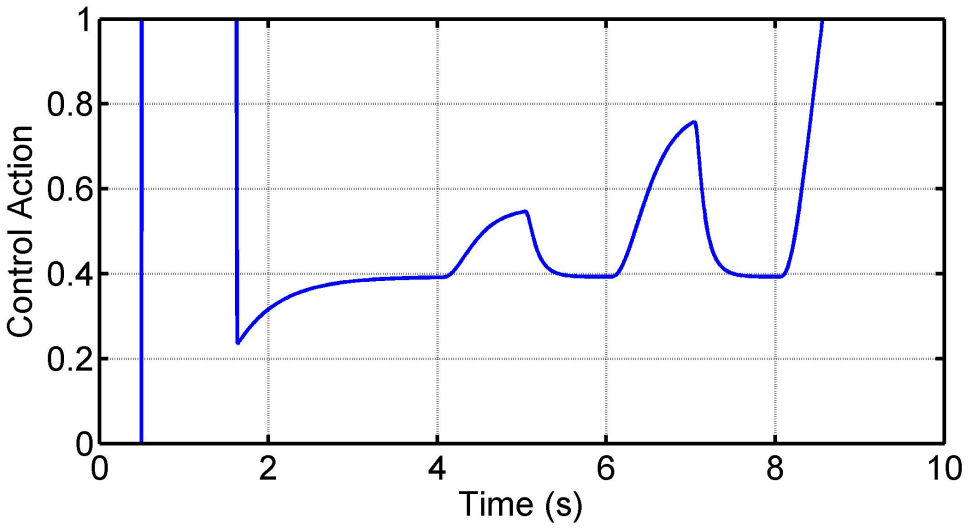


Figure 5.14: Bounded altitude controller actuation

sator allows for quick response to maintain altitude stability during fast orientation changes.

The attitude controller also incorporates an algorithm to minimize thrust during craft inversion, reducing the loss in altitude. This improved performance is shown in Figure 5.15.

Figure 5.16 shows the actuation of the uncompensated altitude controller with peaks

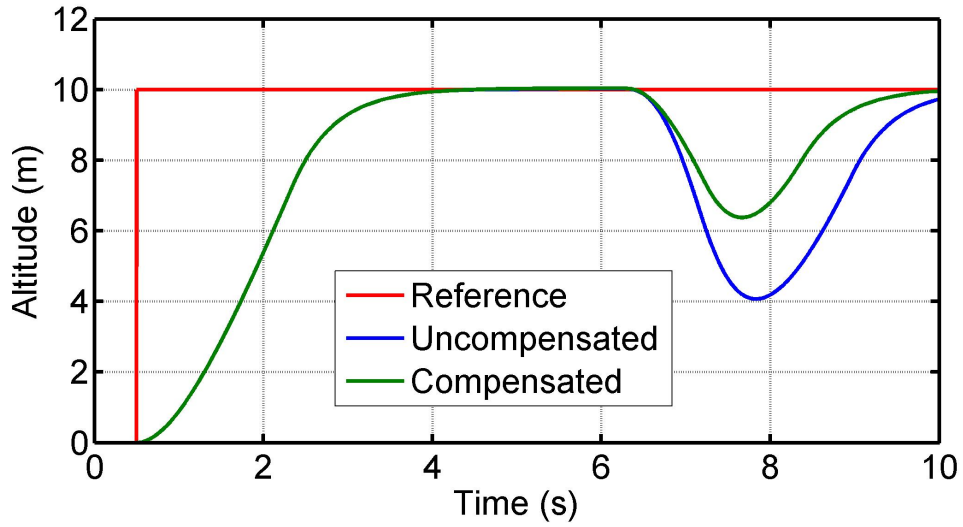


Figure 5.15: Altitude control during craft inversion

and full actuation during craft inversion.

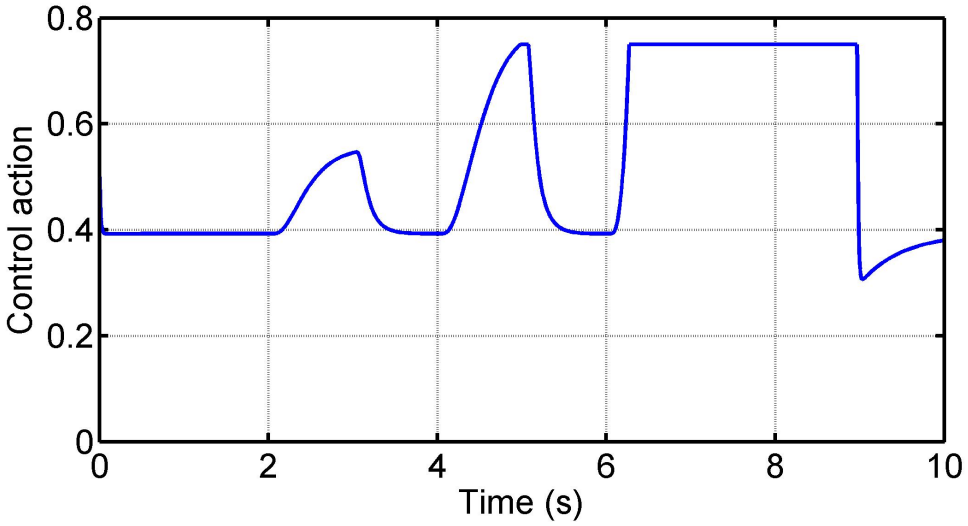


Figure 5.16: Uncompensated altitude controller actuation

Figure 5.17 shows the actuation of the altitude stabilizer combined with the attitude compensator. Note the peaks remain the same, but the actuation is reduced once the craft is inverted. Figure 5.18 shows the actuation of the altitude stabilizer, which no longer has to respond to the orientation changes, hence the absence of the first two peaks. The controller still actuates during extreme orientations such as inversion, but is now neutralized by the attitude stabilizer algorithm.

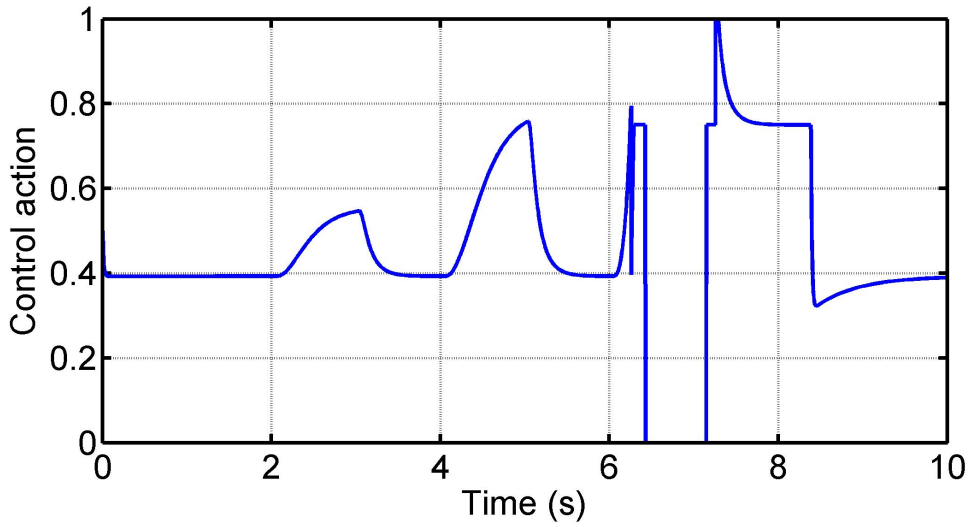


Figure 5.17: Attitude compensated altitude controller actuation

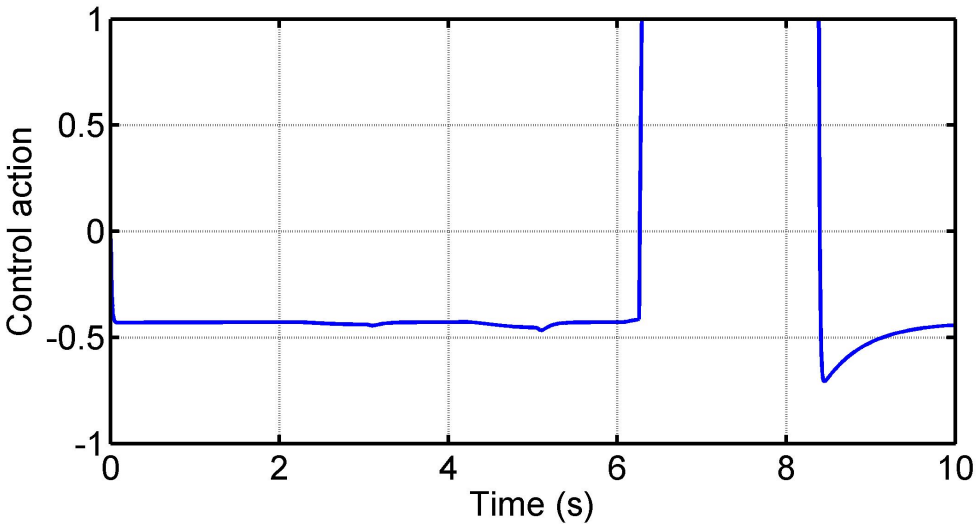


Figure 5.18: Altitude stabilizer actuation

5.3 Full Model Simulation

Given the full model with all the controllers implemented, a sequence of simulations are done to demonstrate control on the various levels. The simulation clearly indicates the interaction and cooperation between the various controllers.

Orientation controllers

At this level the craft is provided with a series of reference orientation commands. Thus only the attitude is controlled and the trajectory is only the consequence of these orientations, given that the altitude controller remains active to stabilize the altitude.

Figure 5.19 shows the orientation of the quad-rotor along with the reference signals and Figure 5.20 shows the position with reference altitude. The first set of graphs are the position versus time plots, whilst the last three are top and side views of the craft's motion.

The system was presented with the following commands:

- At $t = 1s$ change the pitch to 30^circ
- At $t = 5s$ change the yaw to 90^circ
- At $t = 7s$ change the yaw to 180^circ
- At $t = 12s$ change the roll to -30^circ
- At $t = 13s$ change the pitch to 0^circ
- At $t = 13s$ change the yaw to 270^circ

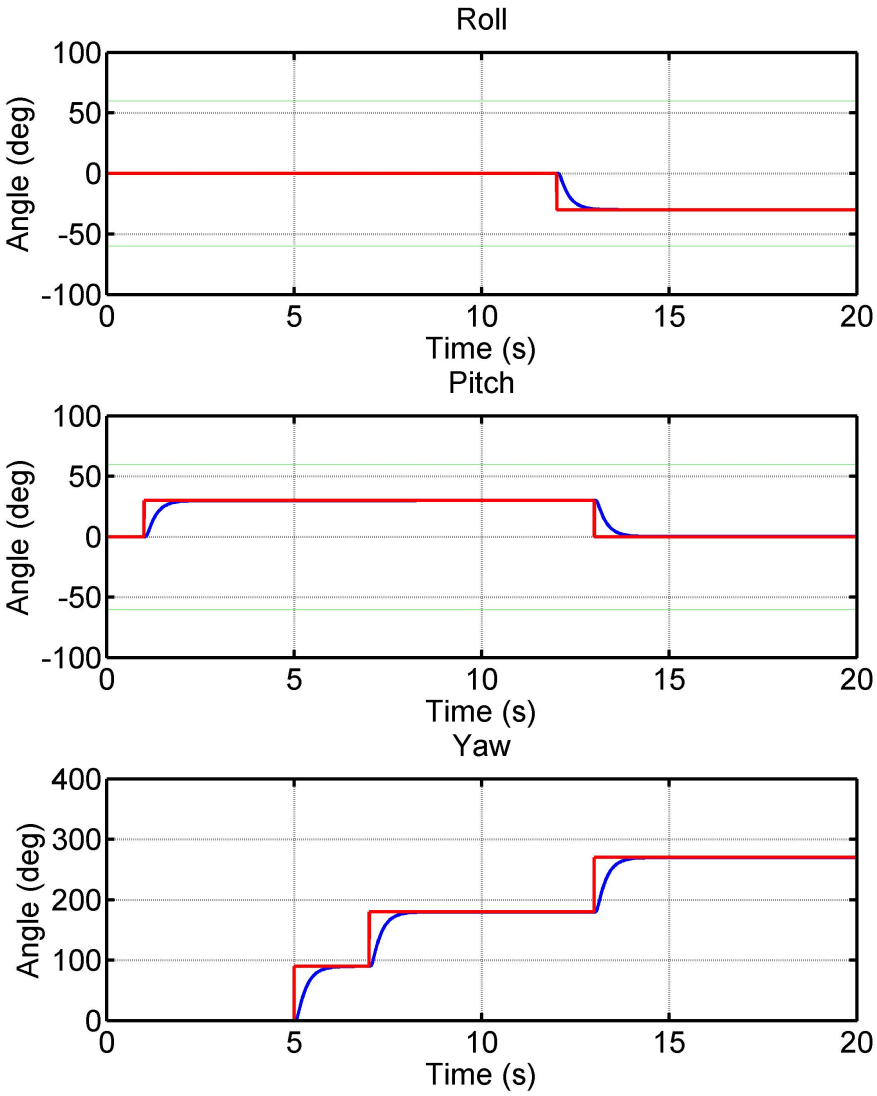


Figure 5.19: Full model simulation 1: Orientation

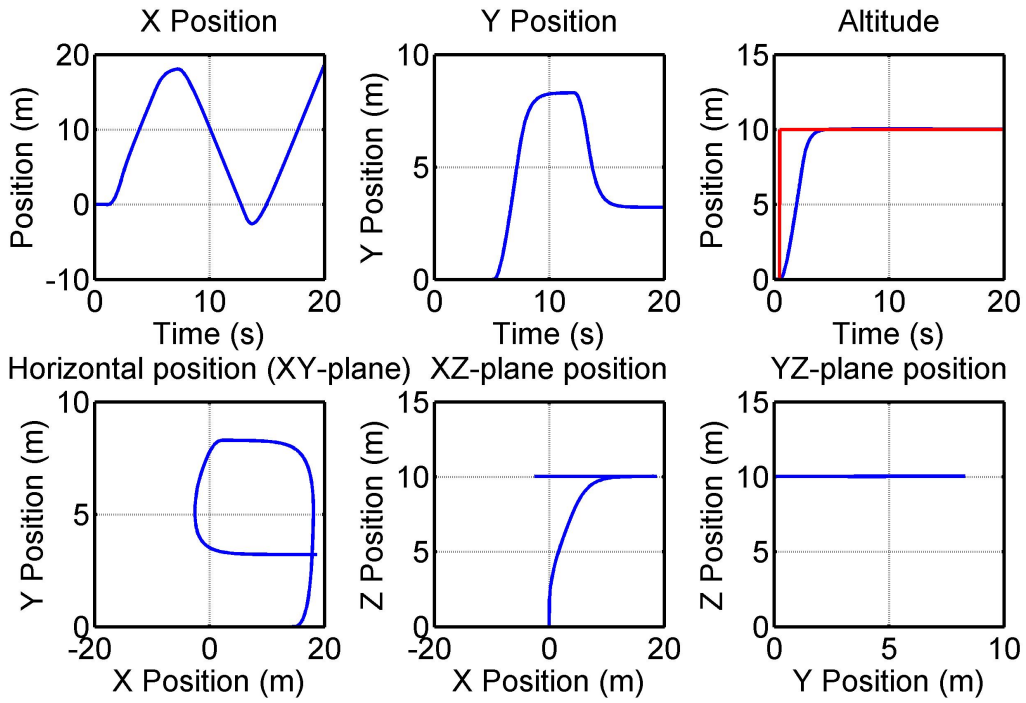


Figure 5.20: Full model simulation 1: Position

Trajectory controllers

At this level the craft is provided with a series of reference trajectory commands. The trajectory controller here is merely a feed-forward controller that calculates the angle required to maintain

Figure 5.21 shows the orientation of the quad-rotor along with the reference signals and Figure 5.22 shows the position. The first set of graphs are the position versus time plots, whilst the last three are top and side views of the craft's motion.

The system was presented with the following commands:

- At $t = 2s$ change the forward speed to 20%

- At $t = 4s$ change the lateral speed to -20%
- At $t = 6s$ change the forward speed to 0%
- At $t = 8s$ change the lateral speed to 0%

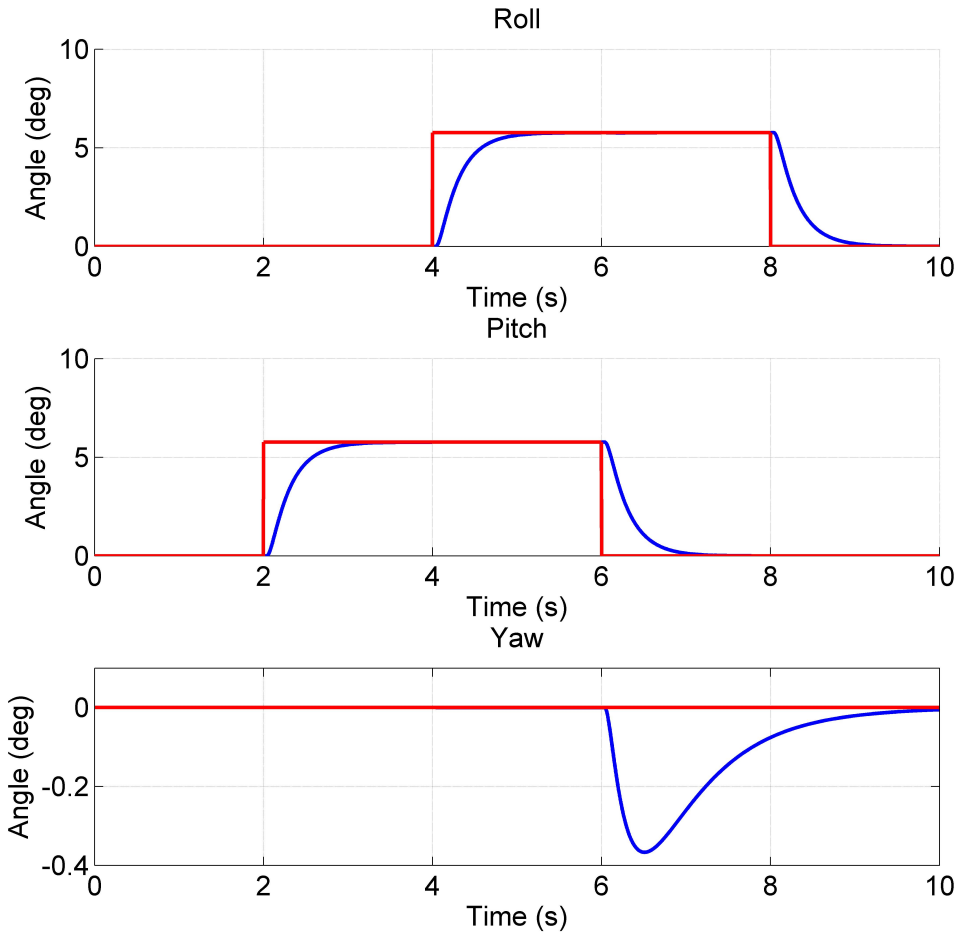


Figure 5.21: Full model simulation 2: Orientation

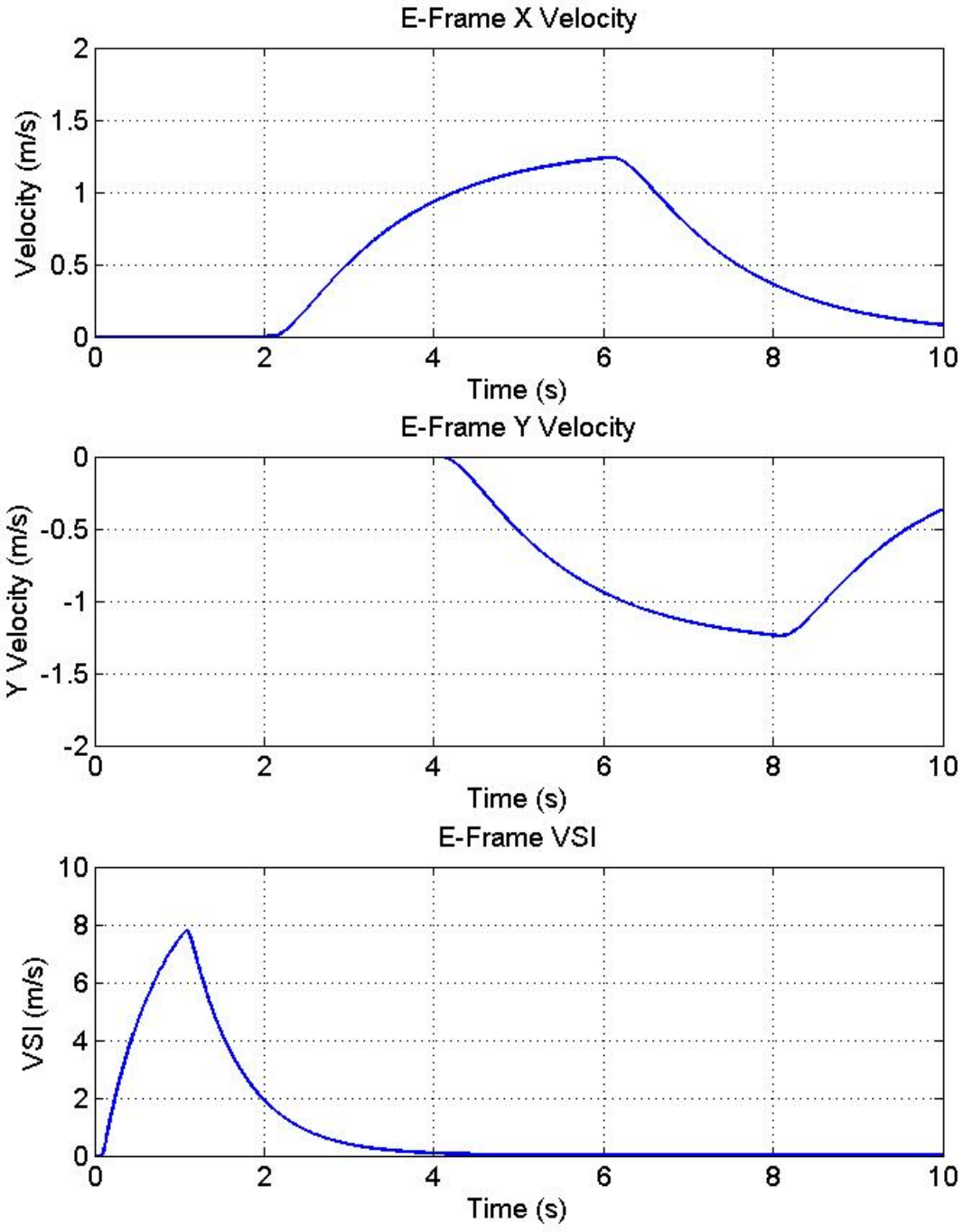


Figure 5.22: Full model simulation 2: Position

5.4 Model Verification

5.4.1 Comparison Model

The model discussed in the previous section has to be verified using either an experimental setup or through comparison to a model that has been verified a priori. The works of Altuğ et al. [12] [13] associated with the GRASP Laboratory of the University Of Pennsylvania are used for the verification. These references are selected because the references have verified models by an established and reputable research group in the field of quad-rotor automation.

The model makes the assumption that the forces introduced by the Coriolis effect are negligible compared to those of the rotors, i.e. that $F_i \gg m\overline{\omega}^B \times \overline{V}^B$ and $T_i \gg \overline{\omega}^B \times I\overline{\omega}^B$. The model also ignores the effects of gyroscopic forces due to the configuration of the quad-rotor, assuming that the net rotation velocity remains equal. This holds true unless a yaw control action is performed. If an imbalance is introduced by the yaw controller combined with a roll or pitch motion, a gyroscopic force is observed, although it remains very small.

Note that the original article differs from the equations given below due to differences in symbol assignments.

The model starts of with the interpretation of the angles from the body reference frame to the earth reference frame, as shown in (5.1).

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{5.1}$$

The system of equations (in the H-frame) governing this model is shown in (5.2). The variable K_i is the drag coefficient along each axis, dependant on the speed of the craft. This is a simplified approximation that assumes body symmetry as the drag is not

dependant on the orientation of the craft.

$$\begin{aligned}
\ddot{x} &= \left[\left(\Sigma_{i=1}^{4} F_i\right)\left(c_\psi s_\theta c_\phi + s_\psi s_\phi\right) - K_1\dot{x}\right]/m \\
\ddot{y} &= \left[\left(\Sigma_{i=1}^{4} F_i\right)\left(s_\psi s_\theta c_\phi - c_\psi s_\phi\right) - K_2\dot{y}\right]/m \\
\ddot{z} &= \left[\left(\Sigma_{i=1}^{4} F_i\right)\left(c_\theta c_\phi\right) - mg - K_3\dot{z}\right]/m \\
\dot{p} &= \left[l\left(F_1 - F_2 - F_3 + F_4\right) - K_4 p\right]/I_{XX} \\
\dot{q} &= \left[l\left(-F_1 - F_2 + F_3 + F_4\right) - K_5 q\right]/I_{YY} \\
\dot{r} &= \left[l\left(-T_1 + T_2 - T_3 + T_4\right) - K_6 r\right]/I_{ZZ}
\end{aligned} \tag{5.2}$$

The model now makes the assumption that the angles (ϕ, θ, ψ) are small such that the transfer matrix T_Θ approaches the identity matrix, i.e. that $\dot{\overline{\Theta}}^{E} = \overline{\omega}^{B}$. This simplifies the model to the system of equations shown in (5.3).

$$\begin{aligned}
\ddot{x} &= \left[\left(\Sigma_{i=1}^{4} F_i\right)\left(c_\psi s_\theta c_\phi + s_\psi s_\phi\right) - K_1\dot{x}\right]/m \\
\ddot{y} &= \left[\left(\Sigma_{i=1}^{4} F_i\right)\left(s_\psi s_\theta c_\phi - c_\psi s_\phi\right) - K_2\dot{y}\right]/m \\
\ddot{z} &= \left[\left(\Sigma_{i=1}^{4} F_i\right)\left(c_\theta c_\phi\right) - mg - K_3\dot{z}\right]/m \\
\ddot{\phi} &= \left[l\left(F_1 - F_2 - F_3 + F_4\right) - K_4\dot{\phi}\right]/I_{XX} \\
\ddot{\theta} &= \left[l\left(-F_1 - F_2 + F_3 + F_4\right) - K_5\dot{\theta}\right]/I_{YY} \\
\ddot{\psi} &= \left[l\left(-F_1 + F_2 - F_3 + F_4\right) - K_6\dot{\psi}\right]/I_{ZZ}
\end{aligned} \tag{5.3}$$

The simplifications are done so as to identify the controllable dynamics of the system. The simplified model is now shown in (5.4), with u_i the control forces described in (5.5).

$$\begin{aligned}
\ddot{x} &= \left[u_1\left(c_\psi s_\theta c_\phi + s_\psi s_\phi\right) - K_1\dot{x}\right]/m \\
\ddot{y} &= \left[u_1\left(s_\psi s_\theta c_\phi - c_\psi s_\phi\right) - K_2\dot{y}\right]/m \\
\ddot{z} &= \left[u_1\left(c_\theta c_\phi\right) - mg - K_3\dot{z}\right]/m \\
\ddot{\phi} &= \left(u_2 - K_4\dot{\phi}\right)/I_{XX} \\
\ddot{\theta} &= \left(u_3 - K_5\dot{\theta}\right)/I_{YY} \\
\ddot{\psi} &= \left(u_4 - K_6\dot{\psi}\right)/I_{ZZ}
\end{aligned} \tag{5.4}$$

$$\begin{aligned} u_1 &= F_1 + F_2 + F_3 + F_4 \\ u_2 &= l\left(F_1 - F_2 - F_3 + F_4\right) \\ u_3 &= l\left(-F_1 - F_2 + F_3 + F_4\right) \\ u_4 &= l\left(-T_1 + T_2 - T_3 + T_4\right) \end{aligned} \tag{5.5}$$

Since the verification model accounts for drag, whilst the developed model does not, the drag coefficient can be set to zero or alternatively the developed model can simply be expanded as shown in (5.6).

$$\begin{aligned} \ddot{X} &= \left(\sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi\right)\frac{U_1}{m} - \frac{K_1\dot{x}}{m} \\ \ddot{Y} &= \left(-\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi\right)\frac{U_1}{m} - \frac{K_2\dot{y}}{m} \\ \ddot{Z} &= -g + \left(\cos\theta\cos\phi\right)\frac{U_1}{m} - \frac{K_3\dot{z}}{m} \\ \dot{p} &= \frac{I_{YY} - I_{ZZ}}{I_{XX}}qr - \frac{J_{TP}}{I_{XX}}q\Omega_T + \frac{U_2}{I_{XX}} - \frac{K_4 p}{I_{XX}} \\ \dot{q} &= \frac{I_{ZZ} - I_{XX}}{I_{YY}}pr + \frac{J_{TP}}{I_{YY}}p\Omega_T + \frac{U_3}{I_{YY}} - \frac{K_5 q}{I_{YY}} \\ \dot{r} &= \frac{I_{XX} - I_{YY}}{I_{ZZ}}pq + \frac{U_4}{I_{ZZ}} - \frac{K_6 r}{I_{ZZ}} \end{aligned} \tag{5.6}$$

$$\begin{aligned} U_1 &= b\left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right) \\ U_2 &= lb\left(\Omega_1^2 - \Omega_2^2 - \Omega_3^2 + \Omega_4^2\right) \\ U_3 &= lb\left(-\Omega_1^2 - \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right) \\ U_4 &= d\left(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2\right) \\ \Omega_T &= -\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2 \end{aligned} \tag{5.7}$$

5.4.2 Verification

Step Response and Observable Dynamics

The first step in verification is to compare the step response of the orientation controllers for individual and simultaneous orientation changes. Figure 5.23 shows the

step response for both models, the developed model on the left and the verification model on the right. The systems are presented with the following series of commands:

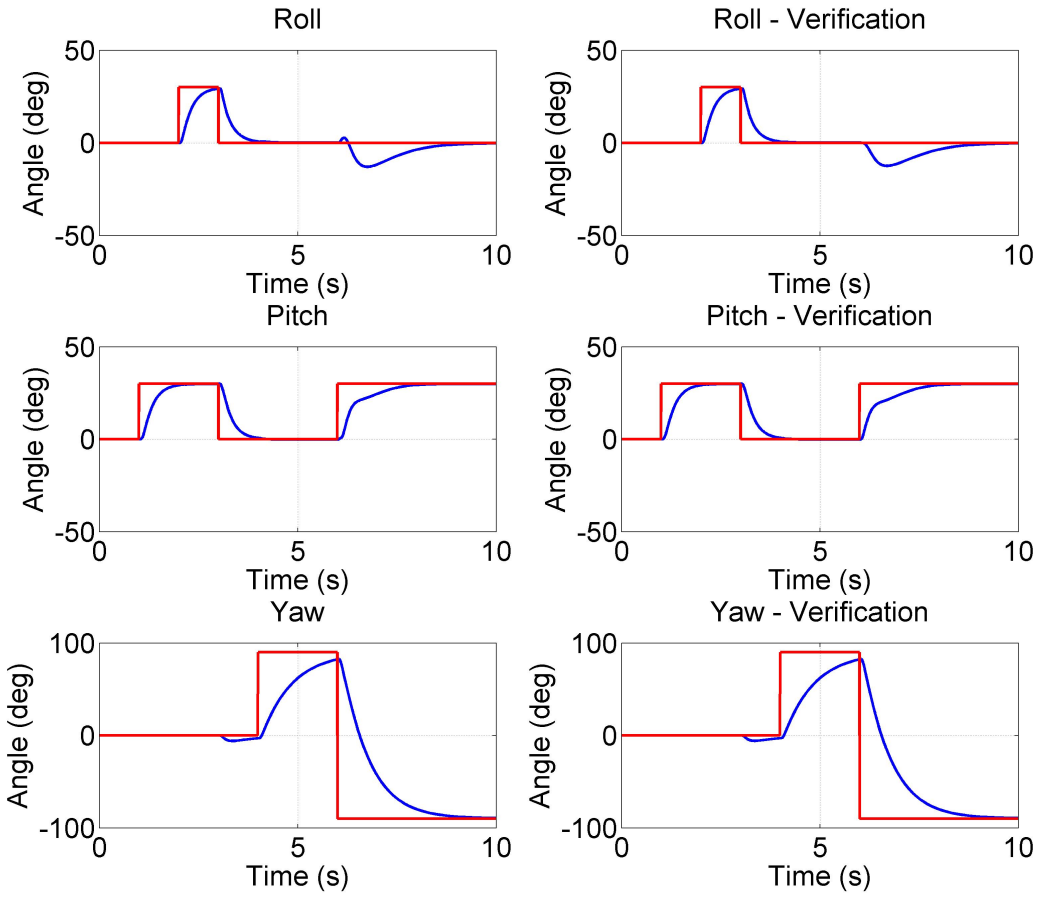


Figure 5.23: Step response verification

- At $t = 1s$, change pitch to $30°$
- At $t = 2s$, change roll to $30°$
- At $t = 3s$, change pitch and roll to $0°$
- At $t = 4s$, change yaw to $90°$
- At $t = 7s$, change pitch to $30°$ and yaw to $-90°$

The dynamics of the two models are very similar, with the slight difference observable in the roll at $t = 7s$ when in the model on the left the roll first peaks positive before turning negative. This is due to the gyroscopic effect present during the yaw control action and simultaneous pitch action. The verification model does not include gyroscopic forces and hence this effect is absent.

Angle Interpretation

Other notable dynamics is the deviation in yaw at $t = 3s$ and in roll at $t = 7s$ which is due to the interpretation of the rotation in three dimensions. As the orientation changes, the axis of rotation changes orientation as well (B-frame), hence the effect of the actuation forces are distributed along the three axis. This can be observed through the transfer matrix, as the effect of the forces (which are fixed in the body reference frame) changes along with the orientation of the craft. Figure 5.24 shows this effect clearly when the controller is commanded to change heading (yaw) while maintaining a constant forward pitch of 30°.

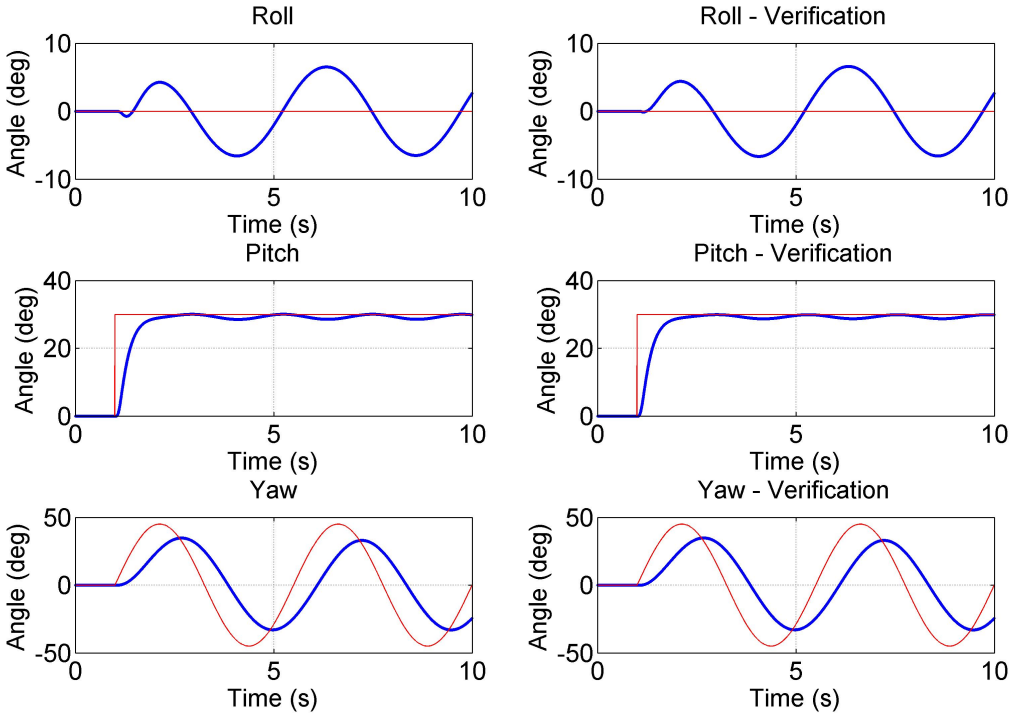


Figure 5.24: Verification of angle interpretation

The transfer of rotational effect between the angles is clear as the heading changes, but the scale of the effect is minimal as can be seen on the pitch graph. The only way to resolve this issue is to vectorize the control as discussed later.

Translational Motion

The translational motion of the craft is modelled in the earth reference frame (*E*-frame) and is identical for both the models. The translational forces are dependant on the net force and orientation of the craft. The figures below show the correlation between the developed and verification models.

Figure 5.25 is a representation of position versus time and Figure 5.26 is a representation as seen by an external stationary observer.

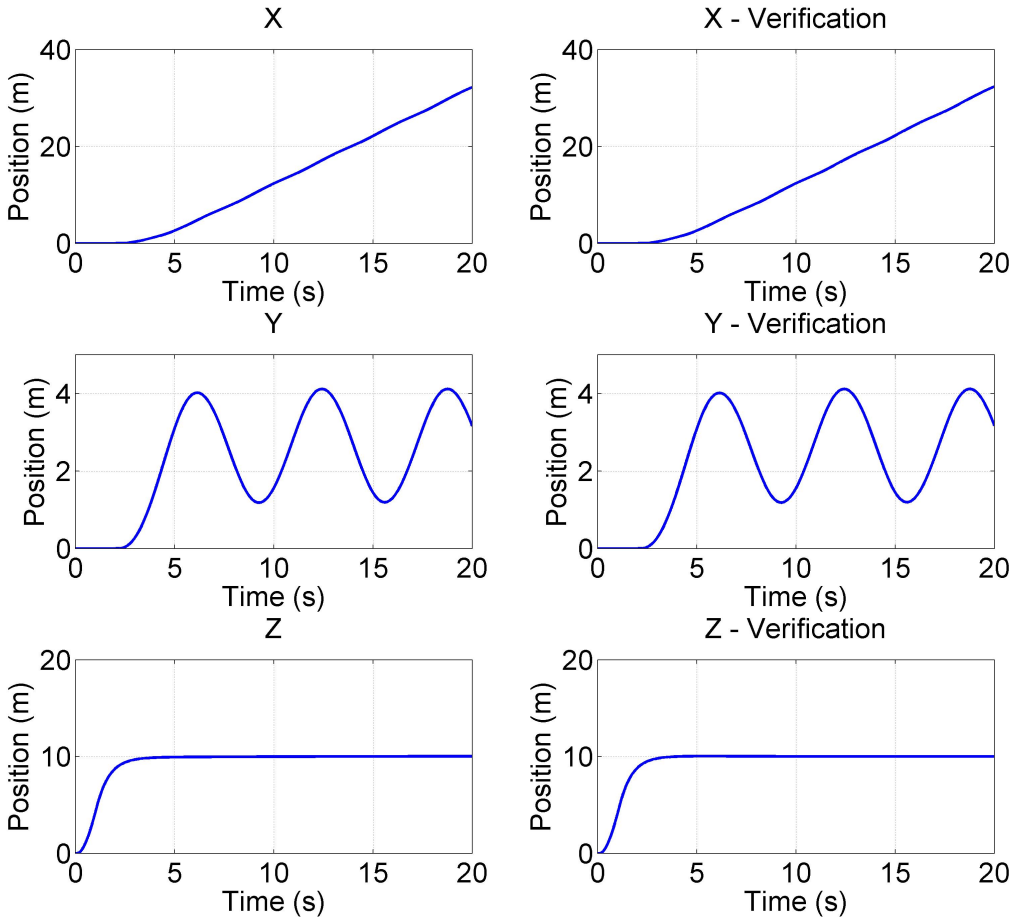


Figure 5.25: Verification of translational motion (versus time)

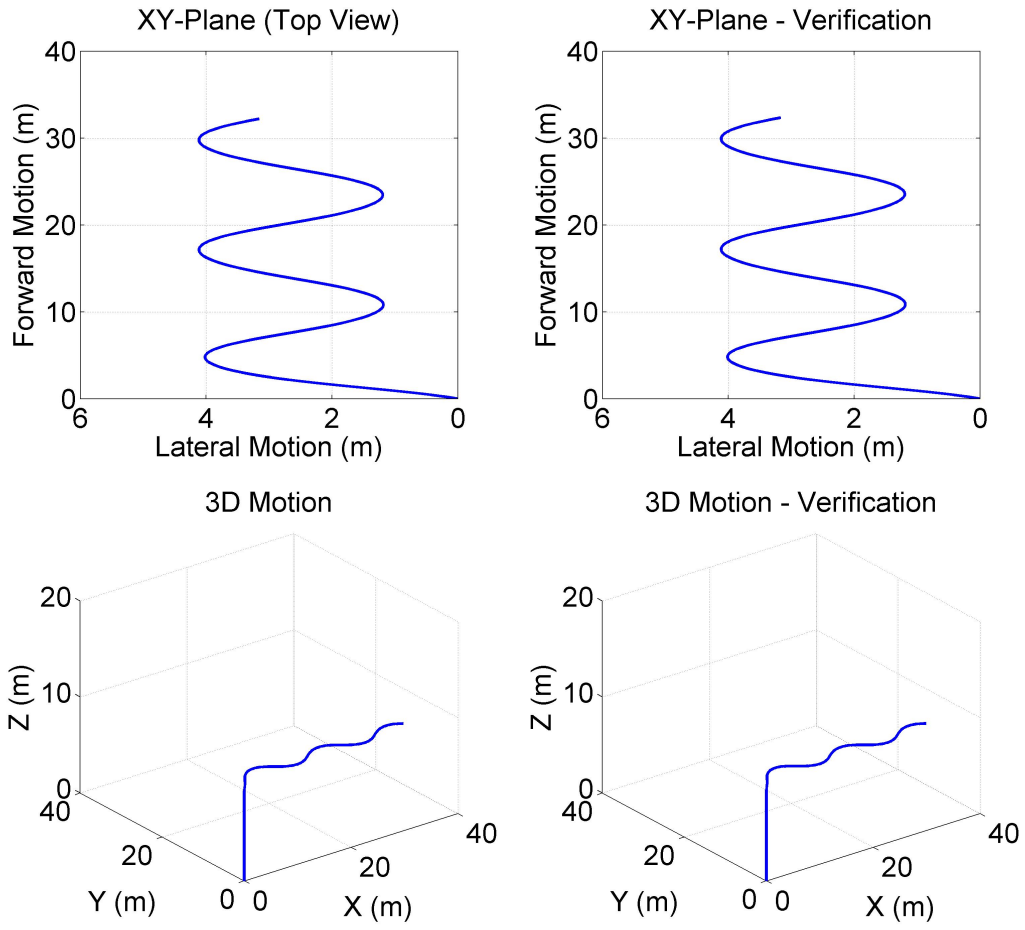


Figure 5.26: Verification of translation motion (observer views)

The respective responses of the two models are close to identical and hence the developed model is considered to be verified as both were simulated with the exact same PD controllers. This is an important point as the validation of the model relies on the verification of the model dynamics. The verification model was verified by comparison to a physical experiment with a quad-rotor and hence it can be concluded from the comparison that both models are representative of the real world system.

Chapter 6

Conclusion

---

*"**I have forced myself to contradict myself in order to avoid conforming to my own taste.**" - Marcel Duchamp*

6.1 Summary

During the course of this study the quad-rotor was characterised and a detailed model of the craft dynamics was derived. The model was derived using the experience of multiple other studies. The model was parametrised using a series of experiments and the results were verified through a correlation to other similar model. It was found that the parameter values did correlate to those of other studies and consequently were implemented into a simulator.

The simulator was implemented in Matlab in two distinct modes. The first was a single comparative simulator that was used to evaluate the technical performance of the various controllers for comparison. The second was a full model simulator to evaluate

the interaction of the controllers. Using the first simulator the technical comparison was done on the orientation controller, showing the differences in performance of various controller types. The importance of proper characterization of the problem was shown using the altitude controller. More complex techniques were shown such as the inclusion of a feed-forward controller to increase the predictive nature of the control system. The results were documented and the correlations done.

6.2 Validation

The simulations demonstrated that the initial hypothesis were indeed valid. The use of the system-level analysis helped to characterize both the control problem and the controller, allowing for a comparison to be made. Through this comparison the most suitable controllers were selected based on viability, then put through a technical comparison to determine the ideal controller. All the controllers identified in the system level analysis were in fact implemented on the model and performed as expected. The technical analysis revealed slight differences in the performance of the controllers, as expected. Subsequently the ideal controller was selected and implemented for the full model simulation. The results of the simulation serve as validation that the process for selecting a controller was in fact beneficial.

6.3 Verification

The verification of the results is done through a comparative study. As stated before, the model was derived using multiple models and the results were correlated. A detailed comparison was done to a model of the GRASP laboratory, a leading contributor in the field of quad-rotor research. The model performed exactly as expected and the results correlated with those found in other similar implementations.

6.4 Conclusions

The study started with a characterization of the problem from the perspective of a beginner in the field of flight automation, slowly growing in complexity. The controllers selected for this comparison were selected because they are well known and didn't require as much specialized knowledge and experience as some of the more advanced control techniques. Other techniques that were identified include Linear Quadratic Regulator (LQR), H_2 and H_∞ controllers, Feedback Linearisation, Sliding mode controllers, Lyopunov function control, Back stepping controllers and others. All of these controllers have successfully been implemented in control of quad-rotors, but were omitted as the focus of the study is the demonstration of the selection process. These controllers also required more specialized knowledge making them more difficult to implement, support and maintain in practice.

Similarly is the case with the model used, as there are better suited techniques such as quaternions or Lagrangian mechanics, both of which require the relevant mathematical knowledge. Thus it was decided that the Newton-Euler model would be used to make the study more comprehensible. Any of these models are sufficiently accurate, they only provide for better interpretation of the dynamics, thus allowing for more advanced control.

Ultimately the objectives of the study were successfully reached. The system level analysis correctly characterised the controllers and the quad-rotor, resulting in a valid suitability study. The system analysis allowed for the design of the generic control loops that were used as reference to implement the controllers. The technical comparison successfully identified the differences between the controllers. The controllers that were simulated were not optimal, but representative of a typical manual tuning process. The controllers all performed as expected and within the specified requirements. The PD controller proved to be the most efficient for the control of the orientation, whilst a PID controller was used for the altitude stabilizer. The attitude compensator and trajectory controller will simply feed-forward controllers that represent the inverse

of the dynamics they control.

6.5 Recommendations

Although all the controllers functioned as required, the full potential of each controller was still not achieved. The entire orientation controller could have been replaced by a single fuzzy logic controller. Similarly there are many documented cases of modified PID controllers that exhibit non-linear behaviour. A lot more work can be done to understand the full capabilities of these controllers, not to mention the ones that weren't included in the study. Although the model is capable of identifying the effects of angular shift between the two reference frames, these controllers do not thoroughly utilize this translation. Effects such as gimbal lock can be witnessed in the model, but it causes the controllers to destabilize. Hence it is recommended that a vectorized control method be derived that allows for continued control in these circumstances.

6.6 Future Studies

As stated before, there are various modelling techniques and a comparative study of these techniques might give more insight into the advantages of each technique. Another interesting observation made during this study was the use of hybrid controllers, where more than one controller type is combined to create a new controller with capabilities exceeding both the original controllers. Examples of such hybrid controllers include, Neuro-Fuzzy control, Model Predictive Fuzzy control, Fuzzy-PID control, Sliding mode PID control and many more. These hybrid controllers allow for some impressive behavioural characteristics and could possibly be truly optimal controllers. They allow for non-linear behaviour in traditionally linear controller, incorporate adaptive and predictive properties with high speed control and improved robustness beyond what is possible with a single controller, to mention but a few possibilities. It is therefore recommended that further studies be done on the use of these hybrid controllers.

Bibliography

[1] P. W. Singer, *Wired for war - The robotics revolution and conflict in the 21st century*. The Penguin Press, 2009.

[2] "United States Air Force Unmanned Aircraft Systems Flight Plan 2009-2047," Headquarters, United States Air Force, Tech. Rep., 18 May, 2009.

[3] D. King and D. S. Massey, "History of the F-15 program: A silver anniversary first flight remembrance," *Air Force Journal of Logistics*, 1997.

[4] N. Nguyen and K. Krishnakumar, "Hybrid damage adaptive flight control with model inversion adaptation," NASA Ames Research Center, Tech. Rep., 2007.

[5] P. H. Sydenham, *Systems approach to Engineering Design*. Artech House, Inc., 2004.

[6] G. Otto, "Autonomous quad rotor helicopter unmanned aerial vehicle," North-West University, Tech. Rep., 2008.

[7] T. Bresciani, "Modelling, identification and control of a quadrotor helicopter," Master's thesis, Department of Automatic Control, Lund University, October 2008.

[8] R. Barclay, "A generic simulator for quad-rotor unmanned aerial vehicles," Tech. Rep., May 2005.

[9] A. L. Salih, M. Moghavvemi, H. A. F. Mohammed, and K. S. Gaeid, "Flight PID controller design for a UAV quadrotor," *Scientific Research and Essays*, vol. 5(23), pp. 3660–3667, 2010.

[10] B. Erginer and E. Altuğ, "Modeling and PD control of a quadrotor VTOL vehicle," in *Proceedings of the 2007 IEEE Intelligent Vehicles Symposium*, 2007.

[11] R. Goel, S. M. Shah, N. K. Gupta, and N. Ananthkrishnan, "Modeling, simulation and flight testing of an autonomous quadrotor," in *Proceedings of ICEAE*, 2009.

[12] E. Altuğ, J. P. Ostrowski, and C. J. Taylor, "Control of a quadrotor helicopter using dual camera visual feedback," *The International Journal of Robotics Research*, vol. 24, no. 5, pp. 329–341, 2005.

[13] E. Altuğ, J. P. Ostrowski, and R. Mahony, "Control of a quadrotor helicopter using visual feedback," in *Proceedings of the 2002 IEEE International Conference on Robotics & Automation*, 2002.

[14] C. F. Petersen, H. Hansen, S. Larsson, L. B. T. Madsen, and M. Rimestad, "Autonomous hovering with a quadrotor helicopter," Aalborg University, Tech. Rep., 2008.

[15] A. Özgür Kivrak, "Design of control systems for a quadrotor flight vehicle equiped with inertial sensors," Master's thesis, Atilim University, December 2006.

[16] Y. Wu, "Development and implementation of a control system for a quadrotor UAV," Master's thesis, Hochschule Ravensburg-Weingarten University of Applied Science, March 2009.

[17] C. Balas, "Modeling and linear control of a quadrotor," Master's thesis, Cranfield University, September 2007.

[18] S. Bouabdallah and R. Siegwart, "Towards intelligent miniature flying robots," Autonomous Systems Lab, EPFL, Tech. Rep., 2005.

[19] K. Miller, "Path tracking control for quadrotor helicopters," Tech. Rep., July 2008.

[20] E. Altuğ, J. P. Ostrowski, and C. J. Taylor, "Quadrotor control using dual camera visual feedback," in *Proceedings of the 2003 IEEE International Conference on Robotics & Automation*, 2003.

[21] T. Hamel, R. Mahony, R. Lozano, and J. P. Ostrowski, "Dynamic modelling and configuration stabilization for an X4-Flyer," in *15th Triennial World Congress*, 2002.

[22] A. Tayebi and S. McGilvray, "Attitude stabilization of a VTOL quadrotor aircraft," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 3, 2006.

[23] S. Salazar-Cruz, A. Palomini, and R. Lozano, "Trajectory tracking for a four rotor mini-aircraft," in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, 2005.

[24] M. A. Johnson and M. H. Moradi, *PID Control: New Identification and Design Methods*. Springer, 2005.

[25] R. C. Dorf and R. H. Bishop, *Modern Control Systems*, 10th ed. Prentice Hall, 2005.

[26] B. S. Blanchard and W. J. Fabrycky, *Systems Engineering and Analysis. 4th ed.* Prentice Hall, 2006.

[27] M. Stork, "Median filters theory and applications," University of West Bohemia, Facultry of Electrical Engineering, Department of Applied Engineering, Tech. Rep., 2003.

[28] P. Steenkamp, "SACAC Workshop: Modern techniques and software for process control," October 2011.

[29] L. Reznik, *Fuzzy Controllers*. Newnes, 1997.

[30] M. Oosterom and R. Babuška, "Fuzzy gain scheduling for flight control laws," *IEEE International Fuzzy Systems Conference*, 2001.

[31] B. Vidolov, J. D. Miras, and S. Bonnet, "A two-rule-based fuzzy logic controller for contrarotating coaxial rotors UAV," in *IEEE International Conference on Fuzzy Systems*, July 2006, p. 7.

[32] E. H. Mamdani, "Application of fuzzy algorithms for the control of a simple dynamic plant," *Proc IEEE*, 1974.

[33] D. E. Thomas and B. Armstrong-Hélouvry, "Fuzzy logic control - A taxonomy of demonstrated benefits," *Proceeding of the IEEE*, vol. 86, no. 3, 1995.

[34] S. A. Raza and W. Gueaieb, "Intelligent flight control of an autonomous quadrotor," in *Motion Control*. Intech, 2010.

[35] Y. Al-Younes and M. Jarrah, "Attitude stabilization of quadrotor UAV using backstepping fuzzy logic & backstepping least-mean-square controllers," in *Proceedings of the 5t International Symposium on Mechatronics and its Applications*, 2008.

[36] T. Drieks and S. Jagannathan, "Output feedback control of a quadrotor UAV using neural networks," *IEEE Transactions on Neural Networks*, vol. 21, no. 1, 2010.

[37] A. A. Tyagunov, "High-performance model predictive control for process industry," Ph.D. dissertation, Technische Universiteit Eindhoven, June 2004.

[38] M. Rau and D. Schröder, "Model predictive control with nonlinear state models," Technical University Munich, Tech. Rep., 2002.

[39] H. J. Kim, D. H. Shim, and S. Sastry, "FLYING ROBOTS - Modeling, control and decision making," in *Proceedings of the 2002 IEEE International Conference on Robotics & Automation*, 2002.

[40] A. S. Dutka, A. W.Ordys, and M. J.Grimble, "Non-linear predictive control of 2 DOF helicopter model," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, 2003.

Appendix A

The Newton-Euler Model

The Newton-Euler model is Euler's axiom of Newton's laws of motion describing the motion of a rigid body object in six DOF. Any object moving in three dimensional (3D) space experiences translational motion in three dimensions and rotation around three axes, hence six dimensions are required to define the motion and subsequently referred to as the 6-DOF model. [7]

Firstly there are a few assumptions that need to be noted before the model can be formulated. The craft is modelled as a fixed body to simplify the equations. By assuming that there is no deformation and flex in the body the inertia matrix can be defined as time-invariant. Body symmetry can be used to simplify equations and control forces are more easily defined. Physical measurements by on-board sensors can also be easily translated to a fixed body frame. The next assumption is that the Centre of Mass (CoM) is coincident with the origin $\overline{\mathcal{O}}_B$ of the body reference frame (*B-frame*) and that this is also the centre of the fixed body symmetry. It is also assumed that the body principle axis of inertia coincide with the axis of the *B-frame*. This assumption reduces the inertia matrix to a diagonal matrix, greatly simplifying the equations.

A.1 Kinematics

Kinematics is the study of the motion of a fixed body without consideration of the forces and torques acting on it. To better describe the motion of a 6-DOF rigid body, it is necessary to define two reference frames:

- Earth inertial reference frame (*E-frame*)
- Body reference frame (*B-frame*)

The *E-frame* $(\mathcal{O}_E, x_E, y_E, z_E)$ is a fixed reference frame, i.e. an external reference viewpoint, using a right-hand convention. The axis x_E points toward north, y_E points toward the west and z_E points upward and $\mathcal{O}_E$ the origin of the *E-frame*. The translational position of the object with regard to the *E-frame* is given by the vector $\Gamma^E[m]$ and the angular orientation by the vector $\Theta^E[rad]$.

The *B-frame* $(\overline{\mathcal{O}}_B, x_B, y_B, z_B)$ is fixed to the body, where x_B points forward, y_B points toward the left, z_B points upward and $\overline{\mathcal{O}}_B$ is the origin of the *B-frame*. This frame also uses the right-hand convention. The *B-frame* is primarily used not to define position, but rather motion, forces and torques. The translational velocity $(\overline{V}^B[m.s^{-1}])$, the angular velocity $(\overline{\omega}^B[rad.s^{-1}])$, the forces $(\overline{F}^B[N])$and the torques $(\overline{\tau}^B[N.m])$are defined in this frame.

The translational position Γ^E of the object describes the position of the *B-frame* origin with regard to the E-frame origin in reference to the *E-frame*. This is expressed in (A.1).

$$\overline{\Gamma}^E = \begin{bmatrix} X & Y & Z \end{bmatrix}^T \tag{A.1}$$

Figure A.1 is a graphical representation of the translational relation between the two reference frames. The reference frames are based on the right handed orientation and rotation.

The angular orientation, referred to in aircraft as attitude, $\overline{\Theta}^E$ is defined as the orientation of the *B-frame*, and hence the fixed body, to the *E-frame*. This orientation is

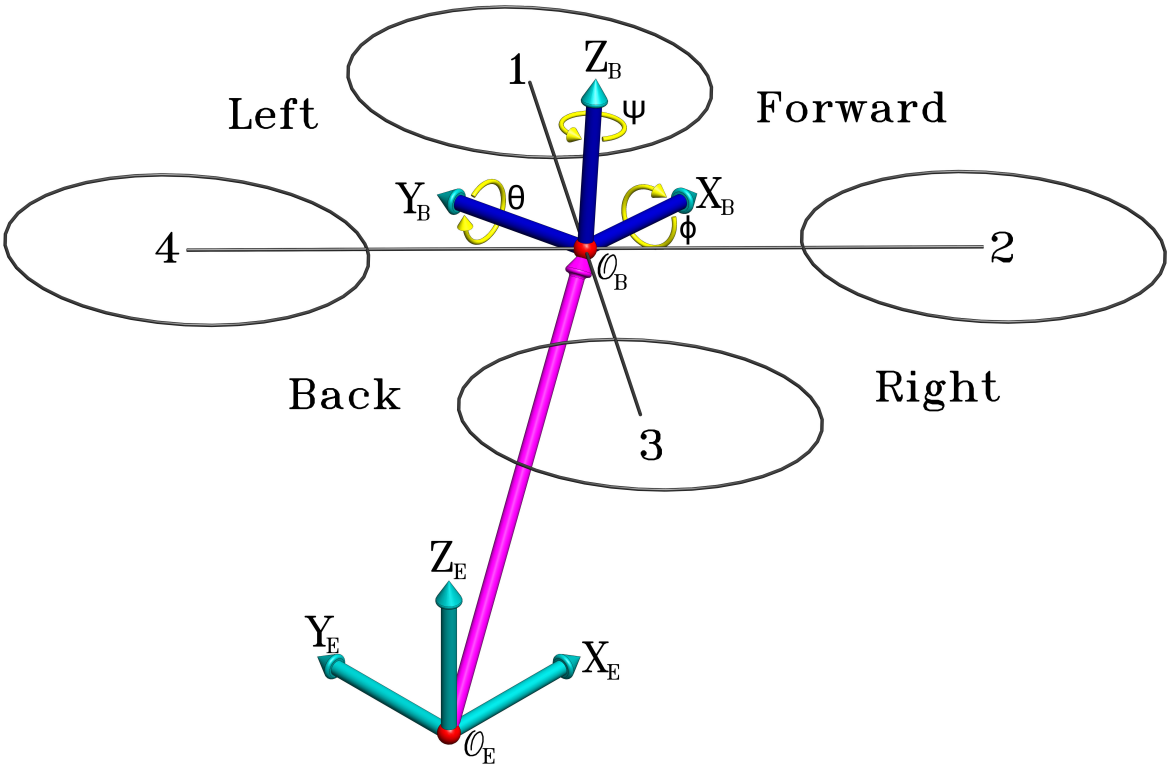


Figure A.1: Reference frames

defined as three sequential rotations around each of the main axes of the *E-frame*, subsequently referred to as roll, pitch and yaw. Roll is the rotation around the *x*-axis, pitch the rotation around the *y*-axis and yaw around the *z*-axis. A right-handed coordinate system is used, i.e. that rotation is counter-clockwise around the axis the axis when viewed from above. The orientation (*attitude*) vector is shown in (A.2).

$$\overline{\Theta}^{E} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^{T} \tag{A.2}$$

To convert any vector between the two reference frames, the rotation matrix $R_{\Theta}[-]$ is used. The rotation matrix is obtained by sequentially rotating the frame around each of the three axes.

Equation (A.3) shows the composition of the rotation matrix R_{Θ}, using the notation:

$$c_k = \cos k, \qquad s_k = \sin k, \qquad t_k = \tan k$$

$$
\begin{aligned}
R_\Theta &= R(\psi,z)R(\theta,y)R(\phi,x) \\
&= \begin{bmatrix} c_\psi c_\theta & -s_\psi c_\phi + c_\psi s_\theta s_\phi & s_\psi s_\phi + c_\psi s_\theta c_\phi \\ s_\psi c_\theta & c_\psi c_\phi + s_\psi s_\theta s_\phi & -c_\psi s_\phi + s_\psi s_\theta c_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix}
\end{aligned} \tag{A.3}
$$

The translational velocity $\overline{V}^B$ and angular velocity $\overline{\omega}^B$, as defined in reference to the *B-frame*, are shown respectively in (A.4) and (A.5).

$$
\overline{V}^B = \begin{bmatrix} u & v & w \end{bmatrix}^T \tag{A.4}
$$

$$
\overline{\omega}^B = \begin{bmatrix} p & q & r \end{bmatrix}^T \tag{A.5}
$$

The translational and angular quantities in each reference frame can be combined into a two 6 dimensional vectors. The generalized position vector $\overline{\xi}[+]$ in reference to the *E-frame* is defined in (A.6) and the generalized velocity vector $\overline{\nu}[+]$ in reference to the *B-frame* in (A.7).

$$
\overline{\xi} = \begin{bmatrix} \overline{\Gamma}^E & \overline{\Theta}^E \end{bmatrix}^T = \begin{bmatrix} X & Y & Z & \phi & \theta & \psi \end{bmatrix}^T \tag{A.6}
$$

$$
\overline{\nu} = \begin{bmatrix} \overline{V}^B & \overline{\omega}^B \end{bmatrix}^T = \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^T \tag{A.7}
$$

As stated before, using the rotation matrix the translational velocity in the *B-frame* can be expressed in reference to the *E-frame* by vector $\overline{V}^E[m.s^{-1}]$ as defined in (A.8).

$$
\overline{V}^E = \dot{\overline{\Gamma}}^E = \mathbf{R}_\Theta \overline{V}^B \tag{A.8}
$$

Similarly the angular velocity $\overline{\omega}^B$ can also be expressed by the vector $\dot{\overline{\Theta}}^E$ in reference to the *E-frame*. Equation (A.9) shows this relation using the transfer matrix $T_\Theta[-]$.

$$
\begin{aligned}
\overline{\omega}^B &= \mathbf{T}_\Theta^{-1} \dot{\overline{\Theta}}^E \\
\dot{\overline{\Theta}}^E &= \mathbf{T}_\Theta \overline{\omega}^B
\end{aligned} \tag{A.9}
$$

Determining the transfer matrix $\mathbf{T}_\Theta$ can be done by resolving the Euler rates in (A.10) in reference to the *B-frame* by rotating each rate to the same plane. The solutions are

shown in (A.11) and (A.12), using the same notation as before.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}(\phi, x)^{-1} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \mathbf{R}(\phi, x)^{-1}\mathbf{R}(\theta, y)^{-1} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} = \mathbf{T}_{\Theta}^{-1} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{A.10}$$

$$\mathbf{T}_{\Theta}^{-1} = \begin{bmatrix} 1 & 0 & -s_{\theta} \\ 0 & c_{\phi} & c_{\theta}s_{\phi} \\ 0 & -s_{\phi} & c_{\theta}c_{\phi} \end{bmatrix} \tag{A.11}$$

$$\mathbf{T}_{\Theta} = \begin{bmatrix} 1 & s_{\phi}t_{\theta} & c_{\phi}t_{\theta} \\ 0 & c_{\phi} & -s_{\phi} \\ 0 & s_{\phi}/c_{\theta} & c_{\phi}/c_{\theta} \end{bmatrix} \tag{A.12}$$

The entire 6-DOF model including the relationship between the two reference frames, as described in (A.6) and (A.7), can be summarized into a single equation shown in (A.13). The derivative of the generalized position in the E-frame $\dot{\overline{\xi}}[+]$ and the generalized velocity in the *B-frame* $\overline{\nu}$ can be related by using the generalized conversion matrix $\mathbf{J}_{\Theta}[-]$. Equation (A.14) describes the matrix $\mathbf{J}_{\Theta}$, using the notation $\mathbf{0}_{3x3}$ for a 3 by 3 matrix filled with zeros.

$$\dot{\overline{\xi}} = \mathbf{J}_{\Theta}\overline{\nu} \tag{A.13}$$

$$\mathbf{J}_{\Theta} = \begin{bmatrix} \mathbf{R}_{\Theta} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{T}_{\Theta} \end{bmatrix} \tag{A.14}$$

A.2 Dynamics

Having the generalized equations for the kinematics as above, it is easy to extrapolate the dynamics by including the forces in reference to the *E-frame*. Eulers axiom of Newtons second law of motion is shown in (A.15), where $m[kg]$ is the objects mass, $\ddot{\overline{\Gamma}}^E$ is the generalized translational acceleration and $\overline{F}^E$ the vector containing the sum of the forces in reference to the *E-frame*.

$$m\ddot{\overline{\Gamma}}^E = \overline{F}^E \tag{A.15}$$

When changing the reference frame to the *B-frame*, the equation changes to take into consideration the Coriolis Effect. This is necessary since the *B-frame* is a non-inertial reference frame, where a deflection is observed due to the rotation of the reference frame. This deflection is described by the Coriolis Effect. Equation (A.16) shows the translational dynamics equation in reference to the *B-frame*.

$$\overline{F}^E = \mathbf{R}_\Theta \overline{F}^B$$

$$m\left(\dot{\overline{V}}^B + \overline{\omega}^B \times \overline{V}^B\right) = \overline{F}^B \tag{A.16}$$

$\dot{\overline{V}}^B$ is the translational acceleration vector in the B-frame, $\overline{\omega}^B \times \overline{V}^B$ represents the deflection observed by the Coriolis effect and $\overline{F}^B$ the vector containing the sum of the forces in reference to the *B-frame*. Note that the symbol $\times$ denotes the vector product.

Similarly the rotational dynamics can be expressed as in (A.17).

$$\mathbf{I}\ddot{\overline{\Theta}}^E = \overline{\tau}^E$$

$$\mathbf{I}\dot{\overline{\omega}}^B + \overline{\omega}^B \times \left(\mathbf{I}\overline{\omega}^B\right) = \overline{\tau}^B \tag{A.17}$$

In the above equations, $I[N.m.s^2]$ is the body inertia matrix in the *B-frame*, $\ddot{\overline{\Theta}}^E$ is the angular acceleration in reference to the *E-frame*, $\dot{\overline{\omega}}^B$ is the angular acceleration in the *B-frame*, $\overline{\omega}^B \times \left(\mathbf{I}\overline{\omega}^B\right)$ again is the Coriolis deflection and $\overline{\tau}^B$ the net torques around each axis of the *B-frame*.

In summary, by combining (A.16) and (A.17), it is possible to describe the generic dynamics of any object using 6-DOF. Equation (A.18) shows the dynamics in matrix form.

$$\begin{bmatrix} m\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \dot{\overline{V}}^B \\ \dot{\overline{\omega}}^B \end{bmatrix} + \begin{bmatrix} \overline{\omega}^B \times \left(m\overline{V}^B\right) \\ \overline{\omega}^B \times \left(\mathbf{I}\overline{\omega}^B\right) \end{bmatrix} = \begin{bmatrix} \overline{F}^B \\ \overline{\tau}^B \end{bmatrix} \tag{A.18}$$

In (A.18) the notation $\mathbf{I}_{3x3}$ refers to an identity matrix of dimensions 3×3. Also note that the first matrix is diagonal and constant. This equation holds true for any rigid body object that meets the assumptions made initially. The constraints and dynamics needed to model the quad-rotor are discussed in Chapter 2.

Appendix B

Aerodynamic Considerations

The dynamics of the craft is subject to various aerodynamics conditions that cause deviation from the idealized model discussed in section 2.2. It may not be necessary to mathematically model these factors, but they need to be understood in principle so they can be identified during the analysis of the implementation results. It is also necessary to understand the effects these factors have on the controller and how to compensate for or prevent these conditions. [6]

Airframe Aerodynamics

In order to understand the effect of the body's shape on the aerodynamics it is first necessary to understand the difference between laminar and turbulent flow. Laminar flow is when the particle trajectories are parallel with no cross currents, eddies or swirls. Turbulent flow on the contrary is chaotic flow with rapid variation in pressure. Due to the formation of vortexes and other disturbances in turbulent flow, the drag experienced is much greater than in the case of laminar flow. Figure B.1 shows basic examples of both laminar and turbulent flow.

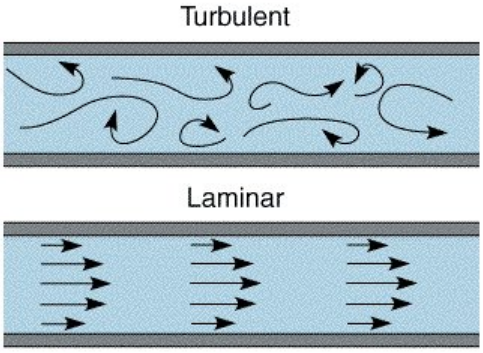


Figure B.1: Laminar and Turbulent flow

Since laminar flow produces less drag, it is the purpose of the aerodynamic design to attempt to produce only laminar flow over the aircraft body. This increases the efficiency of the craft as it reduces the losses due to air friction, i.e. drag. It should be noted that although a skeleton (open gaps) frame body may produce a lighter airframe, it results in turbulent flows through the body.

Rotor Thrust Model

The thrust dynamics of rotors are quote complex to model. The basic nature of thrust dynamics can be described as follows: The thrust produced by the rotor is a result of the acceleration of the air. Bernoullis principle simply states it as the force is a function of the change in flow rate or pressure. Thus any existing airflow through the rotor will affect the thrust produced.

Total Effective Thrust

The concept behind total effective thrust is a simple one. As the orientation of the craft deviates from zero attitude, the vertical component of the thrust is reduced by the cosine of the angle. For example at an angle of 60° the vertical thrust is reduced by half. When taking into account for three dimensional rotation, the math becomes

slightly more complex, but the principles remains. Further, as the angle exceeds the 90° angle, the craft becomes inverted. This means that the thrust now accelerates the craft downward and no longer maintains lift, constituting a model inversion.

Thrust Distribution

The distribution of thrust along the length of the rotor blade is not uniform, rather it is dependent on the difference in translational velocity of each blade section. Since the rotors are spinning around an axis at a given angular velocity, the translational velocity of the given blade section increases the further it is from the axis of rotation. Hence in the ideal case the majority of thrust is produced by the outer regions of the propeller as shown in Figure B.2.

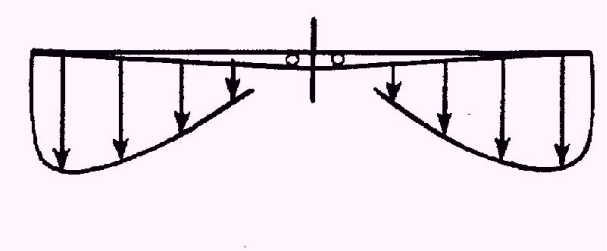


Figure B.2: Ideal thrust distribution along propeller blades

Blade Flapping

Another factor is that of blade flapping, a process that affects the effective thrust of the rotors as a result of airflow across the rotors. As the craft's horizontal velocity increases, the airflow over the rotors increase resulting in a difference in effective airspeed between the advancing and retreating blades. This difference in airspeed translates into a difference in effective thrust, as the one side produces more lift and thus causes a rotational torque around the axis pointing toward the direction of motion. Initially

the net lift remains unaffected, but the limiting factor is the crafts ability to counter this rotational torque without sacrificing attitude control. This is thus the primary limiting factor for the translational velocity of rotor craft. The quad-rotor configuration however is such that this effect is limited by the presence of opposing blade pairs.

Blade Tip Vortex

Blade tip vortex is the tendency of air to recirculate through the rotor at the rotor tips. Since the trust is produced by the acceleration of the air and the recirculating air has a higher entry speed, the effective thrust is reduced. The greater the recirculation the less effective the thrust. Figure B.3 shows the presence of blade tip vortex in a helicopter rotor. Note the two conditions, i.e. out of ground effect (OGE) and in ground effect (IGE), that affect the shape of the blade tip vortex.

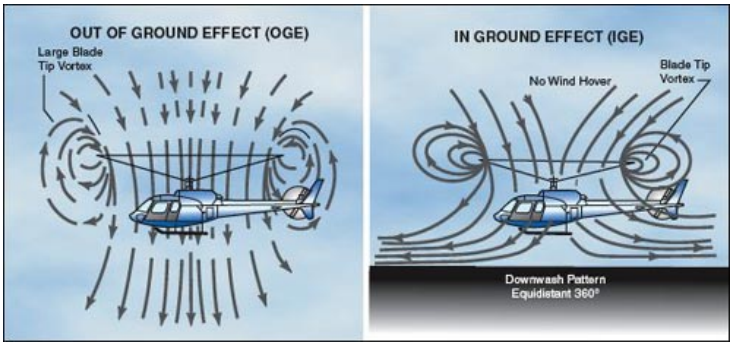


Figure B.3: Blade Tip Vortex

The in ground effect (IGE) reduces the vortex size, hence the effective thrust is higher near the ground. On the contrary, when hovering at higher altitudes, the vortex expands.

In extreme cases such as confined spaces or high climb rates, a vortex forms within the rotor as well. This recirculation causes an up draft of air through the rotor, dramatically decreasing the thrust and consequently causing the craft to loose altitude. This condition is know as settling with power or vortex ring state. Figure B.4 shows the

thrust distribution and air circulation during vortex ring state.

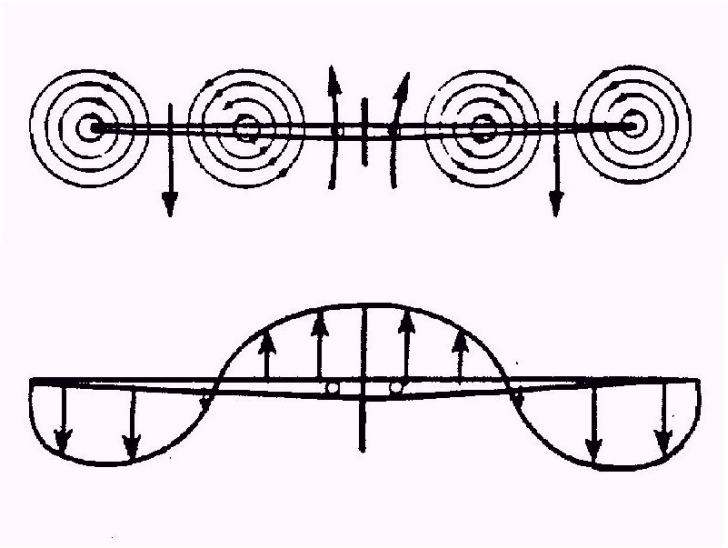


Figure B.4: Vortex ring state

When operating in confined spaces the risk of vortex formation is increased, making it difficult to design a rotor craft for indoor operation. One possible solution to minimize the formation of vortexes is to duct the rotors (ducted fans), which also serves to protect the rotors from strikes when operating in these confined spaces. Ducted fans will also aid to reduce blade flapping.

Wind: Air Speed vs Ground Speed

The airspeed of the craft is the speed of the airflow over the body and hence relevant to the flight envelope of the aircraft. Due to the presence of winds, the airspeed can vary from the ground speed and the actual distance covered. Thus the inverse can be used to model the winds affecting the craft. By defining the airspeed as separate from the ground speed, wind is simply added to the airspeed as this variance. The airspeed is used to calculate the aerodynamic drag and hence the force of the wind on the craft is taken into account.

Modelling

Modelling the aerodynamics conditions can prove to be problematic, however these factors discussed above can be modelled based on their characteristics and relevance. Airframe aerodynamics are modelled based on drag characteristics in a matrix dependent on the airspeed of the craft. Calculating the parameters for drag can be difficult though, hence the argument that if the craft operates at low speeds, the effect of drag can be neglected. The total effective thrust is a critical factor as it severely affects the ability of the craft to maintain altitude. It is easy enough to model and hence is included in the model. Thrust distribution, blade flapping and vortexes are neglected based on the assumption that operating speeds are limited and due to the small size of the craft and rotors, the effects are negligible. Regarding wind, once the drag parameters are known, it is easy to model wind. However if drag is neglected, there is no point in modelling wind.