



UNIVERSITEIT • STELLENBOSCH • UNIVERSITY
jou kennisvennoot • your knowledge partner

A Comparison of Control Systems for the Flight Transition of VTOL Unmanned Aerial Vehicles

by

Steven Cornelius Kriel



*Thesis presented at the University of Stellenbosch in
partial fulfilment of the requirements for the degree of*

Master of Science in Electrical & Electronic Engineering

Department of Electrical and Electronic Engineering
University of Stellenbosch
Matieland, South Africa

Study leader: Prof T. Jones

March 2008

Copyright © 2008 University of Stellenbosch
All rights reserved.

Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

Signature:
SC Kriel

Date:

Abstract

This thesis details the development of linear control systems that allow a vertical take-off and landing unmanned aerial vehicle to perform transitions between vertical and horizontal flight. Two mathematical models are derived for the control system design. A large non-linear model, describing all the dynamics of the aircraft, is linearised in order to perform optimal control using linear quadratic regulator theory. Another model is decoupled using time scale separation to form separate rigid body and point mass dynamics. The decoupled model is controlled using classical control techniques. Simulation results are used to judge the relative performance of the two control schemes in several fields including: Trajectory tracking, sensitivity to parameters, computational complexity and ease of use.

Opsomming

Hierdie tesis ondersoek die ontwerp van 'n vliegbeheerstelsel vir 'n vertikaal opstyg en land outonome onbemande vliegtuig. Die stelsel reguleer die vliegtuig tydens die oorgang van vertikale vlug na konvensionele horisontale vlug. Twee wiskunde modelle word afgelei vir die beheerstelsel ontwerp. 'n Groot nie-lineêre model, wat al die dinamika van die vliegtuig beskryf, word gelineariseer sodat optimale beheer toegepas kan word deur van 'n lineêre kwadratiese reguleerder gebruik te maak. Die nie-lineêre model word ook vir simulaties gebruik. 'n Tweede model word ontkoppel op 'n tyd-skaal vlak om onafhanklike starre liggaam en punt massa dinamika te vorm. Die ontkoppelde model word beheer deur klassieke beheer tegnieke. Simulasie resultate word gebruik om die beheer skemas teenoor mekaar op te weeg in verskeie areas, onder andere trajek volging, sensitiwiteit vir veranderende parameters, berekenings kompleksiteit en gemak van gebruik.

Acknowledgments

I would like to extend my gratitude to everybody who assisted me in the completion of this project. In particular I would like to thank the following people and organizations.

- My supervisor, Prof. Thomas Jones, for getting me involved in this line of research and giving me the freedom to make the project my own.
- Iain Peddle for providing insight into various topics, specifically the Time Scale Decoupled controller.
- The National Research Foundation for helping me fund the project.
- My fellow UAV researchers, Carlo van Schalkwyk, Emile Rossouw and John Wilson for the numerous discussions (arguments) *mostly* concerning our research.
- Everybody who helped with hardware, software and other (coffee machine) problems, specifically Niel Müller and Christiaan Brand.
- Suzaan de Stadler for her support throughout the busiest and most stressful times.

Deze scriptie is opgedragen aan mijn grootvader, Steven Cornelis Eijdenberg

Contents

Declaration	ii
Abstract	iii
Opsomming	iv
Acknowledgments	v
Contents	vii
Nomenclature	ix
List of Figures	xiii
List of Tables	xvi
1 Introduction	1
1.1 Overview	1
1.2 Problem Statement	1
1.3 Outcomes	4
1.4 Thesis Outline	4
2 Aircraft Modelling	6
2.1 Axis System	7
2.2 Six Degree of Freedom	8
2.3 Force Model	13
2.4 LQR Model	18
2.5 Time Scale Decoupled Model	23
3 Linear Quadratic Regulator	28
3.1 Introduction to LQR	28
3.2 Controller Structure	30
3.3 Trajectory Creation	34
3.4 Weighting Selection	42
3.5 Results	54

4 Time-Scale Decoupled	59
4.1 Introduction	59
4.2 Inner Loop Design	61
4.3 Outer Loop Design	67
4.4 Results	78
5 Sensitivity Analysis	86
5.1 Methodology	86
5.2 Results	86
6 Conclusions and Recommendations	90
6.1 Overview	90
6.2 Comparison	91
6.3 Recommendations	93
Appendices	94
A Axis System Transformations	95
A.1 Vector Notation	95
A.2 Coordinate Transformations	96
B Feedback Linearisation	99
B.1 Overview	99
C Aircraft Parameters	101
C.1 Physical Properties	101
C.2 Aerodynamic	102
D Simulation Environment	106
D.1 Non-Linear Aircraft Simulation	106
D.2 Six Degree of Freedom	107
D.3 Sensor Model	107
D.4 Extended Kalman Filter	107
D.5 Servo Model	107
E Additional Figures	109
E.1 TSD three and four second transitions	109
E.2 Sensitivity Analysis Loci	111
Bibliography	116

Nomenclature

Physical:

b	Wing Span
c	Mean Aerodynamic Chord
m	Mass
S	Surface Area
A	Aspect Ratio
e	Efficiency
I	Moment of Inertia Matrix
I_{xx}	Moment of Inertia around roll axis
I_{yy}	Moment of Inertia around pitch axis
I_{zz}	Moment of Inertia around yaw axis

Natural Constants:

ρ	Air Pressure
g	Gravitational Acceleration

Aerodynamic:

q	Dynamic Pressure
C_L	Aerodynamic Lift Coefficient
C_D	Aerodynamic Drag Coefficient
C_l	Aerodynamic Roll Coefficient
C_m	Aerodynamic Pitch Coefficient
C_n	Aerodynamic Yaw Coefficient
C_x	Aerodynamic Axial Force Coefficient
C_y	Aerodynamic Lateral Force Coefficient
C_z	Aerodynamic Normal Force Coefficient

Linear Quadratic Regulator:

J	Cost Function
Q_1	State weighting matrix
Q_2	Actuator weighting matrix

Position and Orientation:

P	Position Vector
N	North Position
E	East Position
D	Down Position
h	Height
P_X	Axial Position Error
P_Y	Lateral Position Error
P_Z	Normal Position Error
α	Angle of Attack
β	Angle of Side slip
q_{1-4}	Quaternions
ϕ, θ, ψ	Euler Angles
i, j, k	Basis Vectors
DCM	Direction Cosine Matrix

Velocity and Rotation:

V	Velocity Vector
\bar{V}	Airspeed
u	Axial Velocity
v	Lateral Velocity
w	Normal Velocity
ω	Angular Velocity
p	Roll Rate
q	Pitch Rate
r	Yaw Rate

Forces, Moments and Accelerations:

M	Moment Vector
L	Roll Moment
M	Pitch Moment
N	Yaw Moment
F	Force Vector

X	Axial Force
Y	Lateral Force
Z	Normal Force
Σ	Specific Acceleration Vector
A	Axial Specific Acceleration
B	Lateral Specific Acceleration
C	Normal Specific Acceleration

Actuation:

T_C	Thrust Command
T	Thrust State
τ_T	Thrust Time Constant
ω	Induced Airspeed
A	Propeller Disc Surface Area
δ_e	Elevator Deflection
δ_a	Aileron Deflection
δ_r	Rudder Deflection
δ_{fc}	Collective Horizontal Flap Deflection
δ_{fd}	Differential Horizontal Flap Deflection
δ_v	Vertical Flap Deflection

System:

A, F	Continuous System Matrix
B, G	Continuous Input Matrix
C, H	Output Matrix
Φ	Discrete System Matrix
Γ	Discrete Input Matrix
T_S	Sampling Time
K	Gain Matrix

Subscripts:

B	Coordinated in Body Axes
E	Coordinated in Earth Axes
W	Coordinated in Wind Axes
G	Gravitational force or acceleration
T	Thrust force or acceleration
v	Vertical Flight

l	Level Flight
0	Static or Initial value
t	Trajectory
p	Perturbation

Superscripts:

BE	Body relative to Earth
BW	Body relative to Wind
WE	Wind relative to Earth

List of Figures

1.1	The Tilt-Wing aircraft built in [4]	2
1.2	(L) The Lockheed XFV tail-sitter prototype. (R) Tail-sitter UAV under construction.	3
2.1	Simplified Aircraft Model	6
2.2	Earth axis system, derived from [19]	7
2.3	Body Axis System, derived from [19]	8
2.4	Definition of Body Coordinated Position Errors	11
2.5	Simplified Airframe showing Actuation Surfaces	14
2.6	Propeller Thrust [9]	16
2.7	Induced Airspeed Vector Diagram [9]	17
2.8	Time scale separation principle	23
3.1	Model Predictive Control uses Expected Future Models	30
3.2	LQR Controller Block Diagram	34
3.3	Diagram showing the notations used for trajectory state functions	36
3.4	Velocity Trajectory	36
3.5	Angle of Attack Trajectory	37
3.6	Pitch Angle Trajectory	38
3.7	Pitch Rate Trajectory	39
3.8	North Position Trajectory	40
3.9	Height Trajectory	40
3.10	Thrust Trajectory	41
3.11	Elevator Deflection Trajectory	41
3.12	Normal step during level flight	45
3.13	Lateral step during level flight	46
3.14	Normal step during vertical flight	47
3.15	Lateral step during vertical flight	48
3.16	Level Flight Pole Locations	49
3.17	Vertical Flight Pole Locations	50
3.18	Gain Settling - Previous Project [5]	52
3.19	Gain Settling - Level Flight	53
3.20	Trajectory Tracking of the LQR system	57

3.21	Trajectory Tracking of the LQR system	57
3.22	Trajectory Tracking of the LQR system	58
4.1	Inner Loop Block Diagram	60
4.2	Outer Loop Block Diagram	61
4.3	Integrator Windup can cause instability	62
4.4	Axial specific acceleration controller with Anti-Windup	63
4.5	Integrator Anti-Windup prevents system from going unstable	63
4.6	Aircraft rolls so that commanded acceleration remains in the ZX-plane	68
4.7	Open Loop Guidance Dynamics	71
4.8	Closed Loop Guidance Controller	71
4.9	Small Normal Specific Acceleration problem	73
4.10	Diagram showing the notations used for kinematic trajectory	74
4.11	Specific Acceleration Vector Diagram	75
4.12	Specific Acceleration Trajectory	76
4.13	Velocity Trajectory	77
4.14	Position Trajectory	77
4.15	Normal step during level flight	80
4.16	Lateral step during level flight	81
4.17	Normal step during vertical flight	82
4.18	Lateral step during vertical flight	83
4.19	Decoupled System Response	84
4.20	Decoupled System Response	85
4.21	Decoupled System Response	85
5.1	LQR Sensitivity to Pitch Moment parameters	87
5.2	TSD Sensitivity to Pitch Moment parameters	87
C.1	Moment of Inertia Measurement Setup	102
C.2	Aircraft Geometry in AVL	104
E.1	Decoupled System Response	109
E.2	Decoupled System Response	110
E.3	Decoupled System Response	110
E.4	LQR Sensitivity to Lift parameters	111
E.5	LQR Sensitivity to Side Force parameters	111
E.6	LQR Sensitivity to Roll Moment parameters	112
E.7	LQR Sensitivity to Pitch Moment parameters	112
E.8	LQR Sensitivity to Yaw Moment parameters	112
E.9	LQR Sensitivity to Physical parameters	113
E.10	TSD Sensitivity to Lift parameters	113
E.11	TSD Sensitivity to Side Force parameters	113
E.12	TSD Sensitivity to Roll Moment parameters	114
E.13	TSD Sensitivity to Pitch Moment parameters	114

LIST OF FIGURES

xv

E.14 TSD Sensitivity to Yaw Moment parameters	114
E.15 TSD Sensitivity to Physical parameters	115

List of Tables

3.1	First Iteration Maximum Deviations	43
3.2	Longitudinal Modes during Level Flight	51
3.3	Lateral Modes	51
3.4	Final Max Deviations	55
4.1	Mixed Normal Actuator Aerodynamic Coefficients	64
4.2	Mixed Lateral Actuator Aerodynamic Coefficients	66
4.3	Chosen Pole Locations	78
4.4	Actual Pole Locations	79
5.1	Sensitivity to Various Parameters	88
C.1	Moments of Inertia	102
C.2	Parameters for Calculating Drag	103
C.3	Stability Derivatives	104
C.4	Control Derivatives	105
D.1	Standard Deviations of Noise on Sensors	108

Chapter 1

Introduction

1.1 Overview

Most conventional UAVs (and larger craft) can be divided into two categories, CTOL (Conventional Take-off and Landing) or VTOL (Vertical Take-off and Landing). CTOL craft, such as most fixed wing aircraft, have several advantages in the areas of efficiency, speed and mission radius. In contrast, VTOL craft are restricted in these areas but have the tremendous benefit of being able to take off and land nearly anywhere. This project, a continuation of previous work done at the Electronic Systems Laboratory, seeks to bridge these two regimes by creating a craft that has the flexibility offered by VTOL while maintaining the benefits of a fixed wing aircraft.

1.2 Problem Statement

This thesis addresses the problem of moving from horizontal to vertical flight. During this transition the model of the aircraft changes significantly. It is therefore not possible to use a single linear model over the entire flight envelope. Multiple control techniques are implemented to solve this problem. A Linear Quadratic Regulator will be implemented to provide optimal control. An innovative Time-Scale Separation technique based on the PhD work[2][3] of Iain Peddle at the University of Stellenbosch is tested for this application. The performance of the two control methods is compared to determine the best candidate.

1.2.1 History of Project

One Masters project [4] has already been dedicated to this endeavour. The project investigated various aircraft configurations as candidates for VTOL and efficient forward flight. Ultimately the Tilt-Wing design was found to offer the best solution.



Figure 1.1: The Tilt-Wing aircraft built in [4]

An Ultra Stick RC plane was extensively modified to create a tilt-wing craft. Two motors were added to the wings to provide lift during take-off and airflow over the new actuation surfaces. The existing ESL avionics was expanded and a simple controller was developed for hover and tested in simulations.

1.2.2 Change to Tailsitter

Early in this project it was realized that the tilt-wing configuration would have to be reconsidered. Several factors contributed to this decision.

- The coupling between the fuselage and the wings is very difficult to model. The models of both the wings and fuselage depend on the position of the wings, but must be simulated in a combined manner. The leads to a complex interaction between the two models.
- The center of gravity (CG) should be placed on the bar connecting the wings to the aircraft, but it is to be expected that the CG will move. The question arises whether a stable hover can be achieved with an offset CG. Through simulation it has been shown that there is a point of stable hover if the CG is offset, although a constant flap deflection is required. It is still unknown whether the flaps are able to provide a large enough moment to correct expected amount of CG-offset.
- During takeoff the propellers are very close to the ground and induce a significant ground effect. This effect is not easily modeled and is currently unknown.
- All the aerodynamic calculations in this project are done under the assumption that the wing aerodynamics operate in the linear domain, far away from the stall point. During transitions the fuselage and, more critically, the tail will experience a very high angle of attack. High

angles of attack produce non-linear effects which are very difficult to model or control.

- The additional weight added by the tilt-wing mechanism, avionics and motors would require the aircraft to fly at very high speeds or high angles of attack to produce enough lift to stay airborne.
- In the original design the batteries were mounted on the wings, behind the motors. This increases the moment of inertia in the yaw and roll axes making it more difficult to effectively control all motions of the aircraft with the available actuation forces.

It was clear that the tilt-wing design could not be used and that an entirely new approach was called for. The decision to create a tail-sitter was reached when it was realized that there is no need for the fuselage to stay level, as with a commercial tilt wing craft. In fact, the fuselage is not necessary at all. A tail-sitter is a plane with a large tail section capable of supporting the entire aircraft. In this standing position the thrust is directed upwards and maximum thrust is sufficient to accelerate the craft against gravity. Once the plane is in forward flight the required steady state thrust will be comparable to any fixed wing craft since most of the lift would be provided by the wings. The plane would *sit* on its tail and take off vertically before making the transition to forward flight without any mechanical configuration changes.

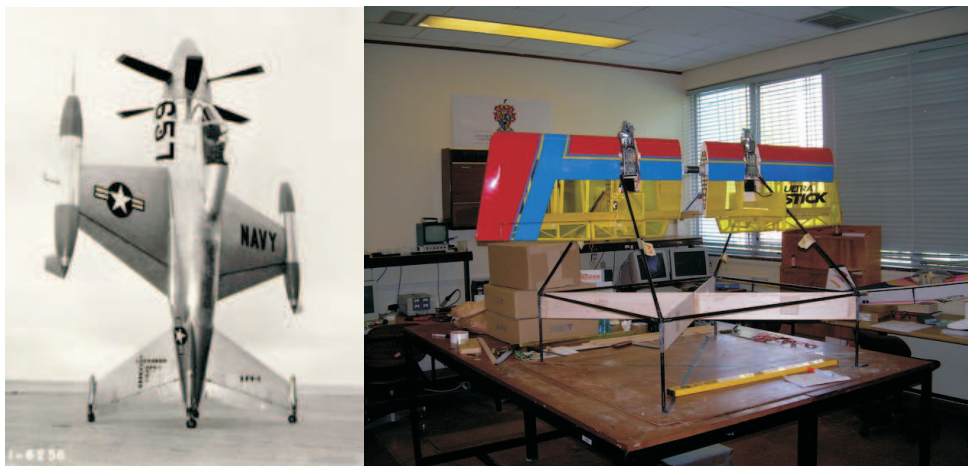


Figure 1.2: (L) The Lockheed XFV tail-sitter prototype. (R) Tail-sitter UAV under construction.

1.3 Outcomes

The construction of an aircraft that is aerodynamically viable for the intended flight range is difficult and requires thorough aerodynamic knowledge that is beyond the scope of this project. During construction it was not known whether the chosen design would be able to complete the intended trajectories. Without the landing gear to allow a safety pilot to land the aircraft in a conventional manner the prospect of transitions during test flights was deemed too risky for initial experiments

The outcomes of this project are therefore mainly theoretical. The main aim is to show, through simulations, that the intended design is viable and can be controlled using one or more of the proposed control schemes. To determine the best candidate these schemes were compared using the following criteria.

Implementation A good control system should be relatively simple to implement and use, giving the designer control over design parameters in a transparent and straightforward way.

Robustness Some techniques are more sensitive to small changes in the model. A well controlled system should remain stable over a large range of parameter uncertainty.

Trajectory Tracking Ultimately the performance of the control system will be judged on how well it can follow the proposed trajectory. A faster system with a higher bandwidth will be able to complete the trajectory faster. A higher bandwidth can move poles closer to instability and therefore a compromise will be necessary.

Computational Complexity Throughout the development of a control system the computational abilities of the OBC must be kept in mind. Ultimately the control system must be able to perform well on a relatively low-power CPU.

1.4 Thesis Outline

This thesis will detail the entire design process for Linear Quadratic Regulator and Time-Scale Decoupled control systems. Chapter 2 focuses on the derivation of the mathematical models used throughout the project. In Chapter 3 the implementation of an LQR control system is discussed and its performance is evaluated using non-linear simulations. Chapter 4 follows a similar outline as Chapter 3 for a Time Scale Decoupled control system. The sensitivity of the two control systems is investigated in Chapter 5 while conclusions

and recommendations are given in Chapter 6. The purpose of this project is to focus on the design of the control systems and as such several very important topics are only briefly mentioned in the main body of the text. These topics relate to work done in previous projects and are discussed in more detail in the appendices.

Chapter 2

Aircraft Modelling

When creating simulations, most elements should be made as general and widely applicable as possible. Modular design aids in this by isolating the specific aircraft model from the rest of the simulation. This is especially important when working in a team where objects are shared and version control is important. To this end the aircraft simulation is split into two sections as shown in figure 2.1. The general section takes forces and moments acting on the body as inputs and calculates the object's path in three dimensional space. This Six Degree of Freedom block is applicable to any object. The aircraft specific section uses aerodynamic information to convert velocities and actuations to rigid body forces and moments. These two sections are combined to form the full non-linear state model. The non-linear model is then linearised to allow control techniques to be applied.

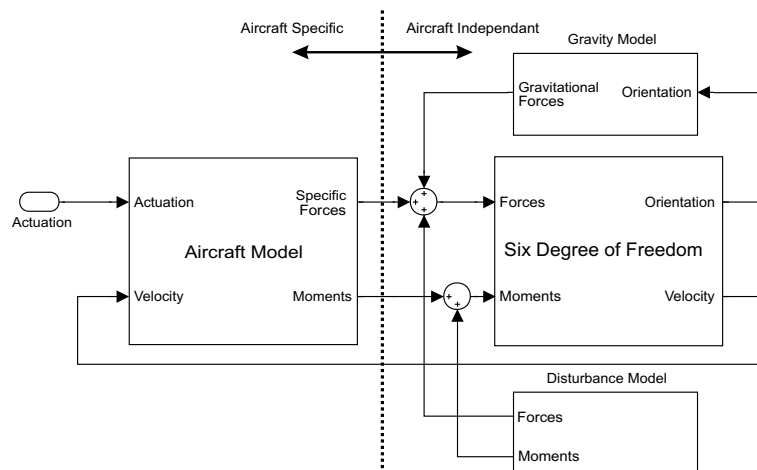


Figure 2.1: Simplified Aircraft Model

2.1 Axis System

In order to create a state space model vector quantities must be coordinated into an axes system as explained in appendix A. Three axis systems are used throughout this project.

2.1.1 Earth

The earth axis system constitutes an inertial reference frame in which Newton's Laws can be applied. The assumption is made that the Earth is flat and non-rotating. Considering the distance and duration of flight this is a valid assumption. A fixed point is chosen as the origin of the system, usually on the runway or some other convenient point. The X-axis is chosen to lie in the Northward direction while Y lies East and Z points down. Quantities coordinated in this system will also be referred to as NED quantities.

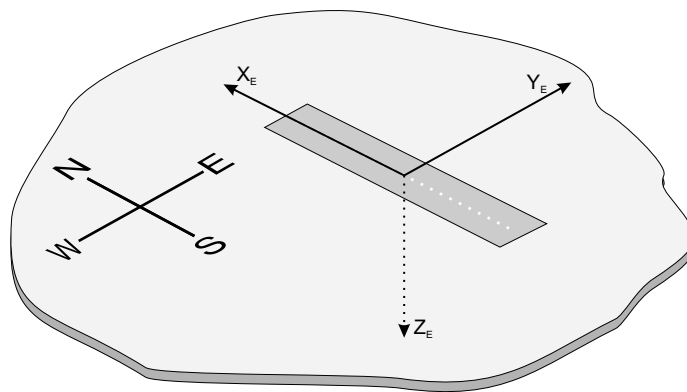


Figure 2.2: Earth axis system, derived from [19]

2.1.2 Body

The body axis system is attached to the airframe of the craft with the origin chosen as the CG. The X-axis is chosen along the axis of symmetry in the forward direction. The Y-axis points along the wing to the right and the Z-axis points down in the plane of symmetry. Rotations around these axes are called Roll, Pitch and Yaw respectively and are denoted by ϕ , θ and ψ . Since the origin of the axis system moves with the aircraft, it is not possible to define an absolute position in the body axes. It is sometimes useful to coordinate a position error in the body axes in which case the error vector can be separated in longitudinal and lateral errors. This separation makes it possible to decouple the system.

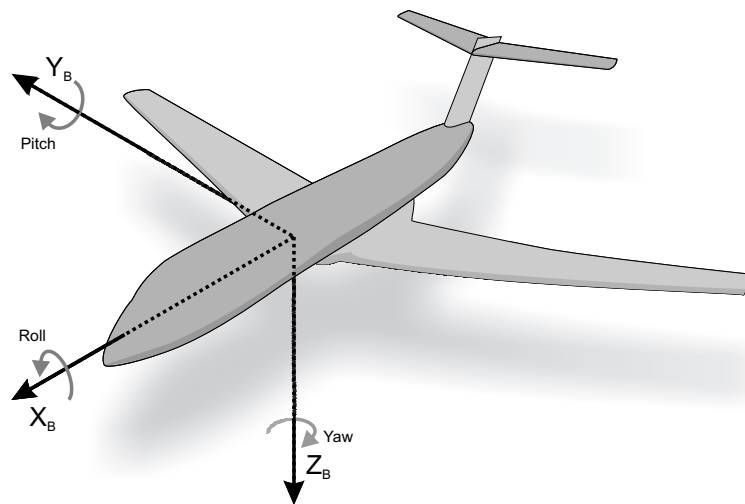


Figure 2.3: Body Axis System, derived from [19]

2.1.3 Wind

The wind axes system has the same origin as the body axes, but differs in orientation. The wind axes system is aligned to correspond to the aircraft's motion rather than its physical construction. This simplifies the calculation of aerodynamic forces since stability derivatives are coordinated in the wind axis system. The X-axis points into the incoming air stream. The Z-axis points down and lies in the plane of symmetry. The Y-axis is chosen perpendicular to the other two axes.

2.2 Six Degree of Freedom

2.2.1 Attitude Descriptions

As described in appendix A there are several ways to describe the attitude of an aircraft. Two techniques were implemented in this project: Euler angles and quaternions. Each method introduces some complexity to the control system. This section will describe the dynamics of the attitude parameters and the impact that each scheme will have on the design process.

Euler Angles Any Euler representation exhibits a discontinuity at a certain orientation as shown in [7]. Considering the wide flight envelope (compared to conventional aircraft) there is no Euler set that will be suitable for all expected orientations. It is necessary to switch between sets of Euler angles during flight.

To make this switch as seamless as possible a quaternion representation is used whenever possible, due to its lack of discontinuities. Euler angles

are only used to calculate the linearized matrix and the corresponding LQR gains. Therefore, two state equations are used throughout the flight envelope and is defined in [7] as

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}_{(321)} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix}_{(321)} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.2.1)$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}_{(231)} = \begin{bmatrix} 1 & \sin \phi & \cos \phi \tan \theta \\ 0 & \cos \phi & -\cos \phi \\ 0 & \cos \phi \sec \theta & -\sin \phi \sec \theta \end{bmatrix}_{(231)} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.2.2)$$

The subscripts 321 and 231 denote the state vectors using the two different Euler sequences. It should be noted that the ϕ , θ and ψ angles represent totally different physical quantities in the two state vectors. Another benefit of Euler angles is that the resulting mathematical model is very easy to decouple since each angle only describes movement on a single plane. Decoupling the dynamics results in a dramatic performance boost since all the matrix inversions associated with LQR are performed substantially faster.

Quaternions Quaternions have the advantage of not having any discontinuities. This allows the same attitude description to be used over the whole flight envelope. According to [6] the quaternion dynamics can be expressed as

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.2.3)$$

Unfortunately the use of Quaternions also give rise to the following problems.

1. Quaternion based models can often not be decoupled as easily as those based on Euler angles, as is illustrated in appendix A. A non-decoupled system forces the designer to use the full state matrix for LQR. Inverting the full state matrix is a computationally intensive operation that limits the amount of time steps that can be used when solving the Riccati equation.
2. When using LQR the designer needs to attribute weightings to each state. Euler angles are easy to visualize and a deviation in a certain angle has a clear and definite effect. Quaternions are not as intuitive and this makes it difficult to attribute weightings to the aircraft's orientation. When using Quaternions it is problematic to control a pitch angle error more heavily than a yaw angle error, since a single rotation will affect two or more quaternions.

3. Due to quaternion conditioning (quaternion vector has a size of 1) their effect is non-linear, [6]. A quaternion increasing from 0.9 to 1.0 may represent a much larger change in angle than an increase from 0.0 to 0.1. Since LQR minimizes a quadratic cost function the weightings on quaternions will deliver inconsistent results.
4. Consider the dynamic equation for q_4 during forward flight with a zero angle of attack and angle of side slip. The quaternions have been initialized for level flight in the North direction so that $\mathbf{q}_{1-4} = [0 \ 0 \ 0 \ 1]$.

$$\dot{q}_4 = \frac{1}{2}(-q_1 p - q_2 q - q_3 r) \quad (2.2.4)$$

When this equation is linearised all partial derivatives taken to angular velocities (p, q and r) will be zero. q_4 is thus uncontrollable. Logically one can reason that this does not cause a problem. As soon as the craft deviates from its intended orientation the first three quaternions will no longer be zero and q_4 will become controllable. This will enable the controller to steer the aircraft back to its trajectory with q_4 becoming less controllable closer to the steady state. When calculating LQR gains several matrices are inverted. States that are close to uncontrollable cause the matrix to be close to singular, or *ill-conditioned*. An ill-conditioned matrix may cause numerical methods to become unreliable [18]. This results in LQR never reaching steady state and raises serious doubts about the guaranteed stability of LQR.

2.2.2 Kinematics

Velocity and position can be coordinated in each of the axis systems defined earlier. These representations are suited to different situations. Since position is usually coordinated in the earth axes it is important to be able to easily transform the velocity to find the displacement derivative equations.

Earth When velocity is coordinated in the earth axes it has North, East and Down components. Notice that the displacement can be calculated independently of the aircraft's orientation. It is useful to coordinate velocity in the earth axes when controlling position since the controller does not need to be aware of the orientation of the aircraft. Velocity and position can be expressed as

$$\mathbf{V} = \dot{N}\mathbf{i}^E + \dot{E}\mathbf{j}^E + \dot{D}\mathbf{k}^E \quad (2.2.5)$$

$$\dot{\mathbf{P}} = \dot{N}\mathbf{i}^E + \dot{E}\mathbf{j}^E + \dot{D}\mathbf{k}^E \quad (2.2.6)$$

Body The velocity has components in each body direction. This system is advantageous when dealing with forces that act on the body, but are independent of orientation. Such forces exist primarily in hover, with actuations

causing body-fixed forces and accelerations. Forces created by aerodynamic effects, however, will need to be transformed since aerodynamic derivatives are coordinated in the wind axes [12]. To calculate displacement in the inertial reference frame a transformation matrix is needed, resulting in

$$\mathbf{V} = u\mathbf{i}^B + v\mathbf{j}^B + w\mathbf{k}^B \quad (2.2.7)$$

$$\dot{\mathbf{P}} = [i^E \ j^E \ k^E] [\mathbf{DCM}^{BE}]^T \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.2.8)$$

If the system is to be decoupled position errors must be coordinated in the body axis system. When a derivative is taken with respect to a rotating frame Coriolis's equation must be used, resulting in

$$\left. \frac{d\mathbf{P}_B}{dt} \right|_E = \left. \frac{d\mathbf{P}_B}{dt} \right|_B + \boldsymbol{\omega}_B^{BE} \times \mathbf{P}_B \quad (2.2.9)$$

$$\begin{bmatrix} \dot{P}_X \\ \dot{P}_Y \\ \dot{P}_Z \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} - \begin{bmatrix} qP_Z - rP_Y \\ rP_X - pP_Z \\ pP_Y - qP_X \end{bmatrix} \quad (2.2.10)$$

Figure 2.4 illustrates how body coordinated position errors are defined.

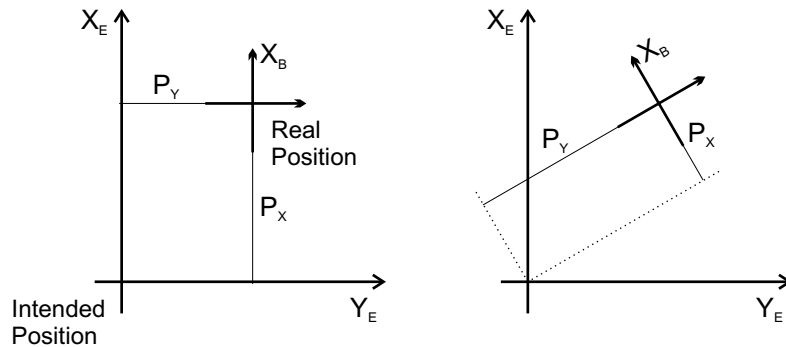


Figure 2.4: Definition of Body Coordinated Position Errors

Wind In the wind axes the X-axis is chosen into the incoming air stream, resulting in a velocity vector with only a X component. Two angles are defined to describe the difference between the wind and body axis: The angle of attack α and angle of side slip β .

The angle of attack is the angle at which air hits the airframe (body axes) projected on the plane of symmetry. The angle of side slip is the angle between the air stream and the plane of symmetry. These two angles are very important when dealing with aerodynamic forces such as lift. This makes the

wind axis the preferred system when working in a regime where such forces dominate the dynamics.

As with body velocities an axes transformation is needed to find the derivative of position in the earth axes.

$$\mathbf{V} = \bar{V} \mathbf{i}^W \quad (2.2.11)$$

$$\alpha = \arctan\left(\frac{w}{u}\right) \quad (2.2.12)$$

$$\beta = \arcsin\left(\frac{v}{V}\right) \quad (2.2.13)$$

$$\dot{\mathbf{P}} = \begin{bmatrix} i^E & j^E & k^E \end{bmatrix} [\mathbf{DCM}^{BE}]^T [\mathbf{DCM}^{WB}]^T \begin{bmatrix} \bar{V} \\ 0 \\ 0 \end{bmatrix} \quad (2.2.14)$$

2.2.3 Equations of Motion

Newton's equations for translational and rotational movement state that

$$\mathbf{F}_B = \left. \frac{d}{dt} m \mathbf{V}_B \right|_I \quad (2.2.15)$$

$$\mathbf{M}_B = \left. \frac{d}{dt} \mathbf{I}_B \boldsymbol{\omega}_B \right|_I \quad (2.2.16)$$

The axis center has been chosen as the center of gravity and aligned with the body axes so that

$$\mathbf{I}_B = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xz} & I_{yy} & I_{yz} \\ I_{xz} & I_{yz} & I_{zz} \end{bmatrix} \approx \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (2.2.17)$$

Note that the derivatives are taken with respect to the inertial reference frame. The state vectors however are coordinated in the body axes. The Coriolis equation describes the relationship between these two derivatives. Given a vector \mathbf{R} ,

$$\left. \frac{d}{dt} \mathbf{R} \right|_I = \left. \frac{d}{dt} \mathbf{R} \right|_B + \boldsymbol{\omega}^{BI} \times \mathbf{R} \quad (2.2.18)$$

By substituting equation (2.2.18) into (2.2.15) and (2.2.16) and assuming that the earth axes constitutes an inertial frame, the six equations of motion are given in [6] as

$$X = m(\dot{u} - vr + wq) \quad (2.2.19)$$

$$Y = m(\dot{v} - wp + ur) \quad (2.2.20)$$

$$Z = m(\dot{w} - uq + vp) \quad (2.2.21)$$

$$L = \dot{p}I_{xx} + qr(I_{zz} - I_{yy}) \quad (2.2.22)$$

$$M = \dot{q}I_{yy} + pr(I_{xx} - I_{zz}) \quad (2.2.23)$$

$$N = \dot{r}I_{zz} + pq(I_{yy} - I_{xx}) \quad (2.2.24)$$

2.3 Force Model

2.3.1 Actuators

Actuators apply forces and moments to the body by diverting the airflow over them. Airflow can originate from forward velocity or induced airflow created by the propellers. The deflection of a control surface is defined in radians with a *positive* deflection causing a *negative* moment. The modified airframe contains the conventional control surfaces (elevator, aileron and rudder) in addition to two additional actuators.

- The ailerons are mounted on the outside of the wing surface and lie in the airflow caused by forward velocity. The ailerons are deflected *differentially* to provide a rolling moment.
- The four flaps on the cross-shaped tail are combined to simulate the effect of an elevator and rudder. The mixing is illustrated in figure 2.5. The elevator creates a negative pitching moment and a positive lift (negative Z-axis) force. The rudder creates a negative yaw moment and a positive side force. The moment arms are long enough to ensure that moments are substantially larger than the forces. Unfortunately the forces are still significant and cannot be ignored.
- The horizontal flaps are mounted on the wings behind the propellers. Since the flaps lie directly in the airflow created by the propellers they can be used to manoeuvre during hover. The flaps can be actuated collectively or differentially. Collective deflection will result in a negative Z-force. The flaps are very close to the CG, resulting in a very small negative pitching moment. Differential deflection of the flaps causes a roll moment, but considering the effectiveness of ailerons during forward flight differential deflection is not used outside of hover.
- The vertical flaps lie in the induced air stream, perpendicular to the wing surface. This actuator will create a lateral force in the positive Y direction along with a small negative yaw moment.

It will be shown that the collective paddles and vertical flaps can be used to combat the non-minimum phase introduced by the forces exerted by the elevator and rudder.

2.3.2 Aerodynamic Forces

All aerodynamic forces are calculated using a general equation given in [1] as,

$$F = \frac{1}{2} \rho V^2 S C_F \quad (2.3.1)$$

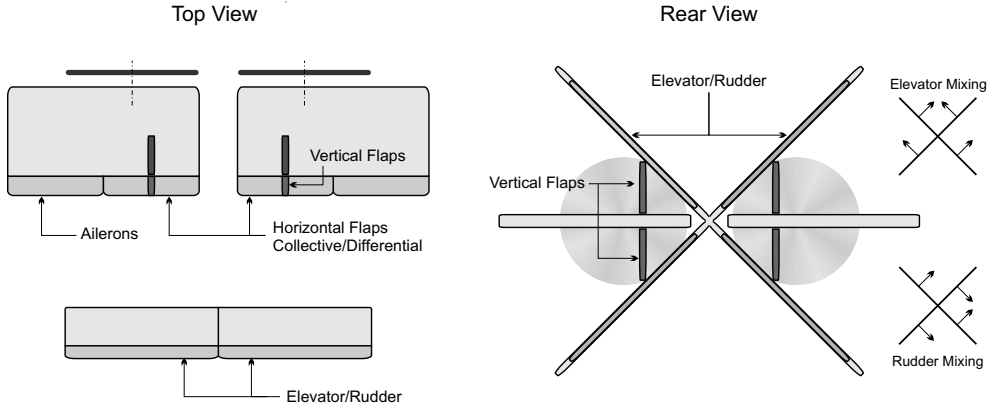


Figure 2.5: Simplified Airframe showing Actuation Surfaces

with ρ , V , S and C_F being the air density, air stream velocity, surface area and non-dimensional aerodynamic coefficient respectively. Moments are calculated in a similar manner using

$$M = \frac{1}{2}\rho V^2 S l C_M \quad (2.3.2)$$

with moment arm l .

To calculate the aerodynamic coefficients of an aircraft reference quantities are used. This is done so that the same surface area and moment arms can be used for all calculations, [12]. The total wing area is used as a reference surface, the mean-aerodynamic chord is used as a reference moment arm for pitching moments and the wingspan is used a reference moment arm for rolling and yawing moments.

2.3.2.1 Body Axes

The aerodynamic coefficients are obtained from simulation in AVL and are coordinated in the wind axes. When calculating the acceleration of the aircraft body-forces are required. For small α and β angles the wind axes aerodynamic coefficients can be coordinated in the body axes, given in [8] as

$$\begin{aligned} C_x &= C_x^W - C_z^W \alpha & C_l &= C_l^W - C_n^W \alpha \\ C_y &= C_y^W & C_m &= C_m^W \\ C_z &= C_x^W \alpha + C_z^W & C_n &= C_l^W \alpha + C_n^W \end{aligned} \quad (2.3.3)$$

2.3.2.2 Derivatives

When linearising a model the derivatives of aerodynamic coefficients are used. These were calculated using an AVL simulation as shown in appendix

C. The notation used is as follows:

$$C_{F_\epsilon} = \frac{\partial C_F}{\partial \epsilon} \text{ so that } C_F = \frac{\partial C_F}{\partial \epsilon} \epsilon \quad (2.3.4)$$

In words, C_{F_ϵ} is the change introduced in C_F for a change in ϵ , where F represents a force or moment and ϵ represents a normalized kinematic state. The simulation software returns these derivatives as dimensionless quantities [12], making it possible to directly compare the derivatives of aircraft that have different dimensions and fly at different speeds. To regain the correct units derivatives taken with respect to rates must be multiplied by $\frac{b}{2V}$ or $\frac{c}{2V}$. Derivatives with respect to angles do not need to be scaled since radians are dimensionless. The general equation for force becomes

$$F = \frac{1}{2} \rho V^2 S \left[C_{F_0} + \left(\frac{b}{2V} \right) C_{F_p} p + \left(\frac{c}{2V} \right) C_{F_q} q + \dots + C_{F_\alpha} \alpha \right] \quad (2.3.5)$$

It is possible to express the wind axes coefficients in terms of the dimensionless stability derivatives given in [8] as

$$C_x^W = -C_D \quad (2.3.6)$$

$$C_y^W = C_{y_\beta} \beta + \frac{b}{2V} [(C_{y_p} p + C_{y_r} r) + \alpha (C_{y_r} r - C_{y_p} p)] + C_{y_{\delta_a}} \delta_a + C_{y_{\delta_r}} \delta_r + C_{y_{\delta_v}} \delta_v \quad (2.3.7)$$

$$C_z^W = -C_L + \frac{c}{2V} C_{L_q} q \quad (2.3.8)$$

$$C_l^W = C_{l_\beta} \beta + \frac{b}{2V} [(C_{l_p} p + C_{l_r} r) + \alpha (C_{l_r} r - C_{l_p} p)] + C_{l_{\delta_a}} \delta_a + C_{l_{\delta_r}} \delta_r + C_{l_{\delta_{fd}}} \delta_{fd} \quad (2.3.9)$$

$$C_m^W = C_{m_0} + C_{m_\alpha} \alpha + \frac{c}{2V} C_{m_q} q + C_{m_{\delta_e}} \delta_e + C_{m_{\delta_{fc}}} \delta_{fc} \quad (2.3.10)$$

$$C_n^W = C_{n_\beta} \beta + \frac{b}{2V} [(C_{n_p} p + C_{n_r} r) + \alpha (C_{n_r} r - C_{n_p} p)] + C_{n_{\delta_a}} \delta_a + C_{n_{\delta_r}} \delta_r + C_{n_{\delta_v}} \delta_v \quad (2.3.11)$$

Note that roll and yaw rates are transformed to the wind axis system using a small angle approximation for angle of attack as shown in [8]. C_L and C_D represent the coefficients of lift and drag respectively, defined in [8] as

$$C_L = C_{L_0} + C_{L_\alpha} \alpha + C_{L_{\delta_e}} \delta_e + C_{L_{\delta_{fc}}} \delta_{fc} \quad (2.3.12)$$

$$C_D = C_{D_0} + \frac{C_L^2}{\pi A e} \quad (2.3.13)$$

C_{D_0} is the parasitic drag consisting of skin friction and form drag. The term $C_L^2 / \pi A e$ represents the induced drag due to lift. Since the aircraft has symmetrical wings the static lift C_{L_0} is zero. The lift generated by the fuselage is very small and cannot be calculated accurately.

2.3.3 Gravity

Gravity is fixed in the earth axes and to coordinate it in the body axis system the *DCM* is used.

$$\mathbf{F}_G = \begin{bmatrix} X_G \\ Y_G \\ Z_G \end{bmatrix} = [DCM^{BE}] \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \quad (2.3.14)$$

2.3.4 Thrust

The aircraft is equipped with two brushless DC motors. The motors are counter rotating to ensure a zero net moment. The motors provide thrust only in the positive X Body direction.

$$\mathbf{F}_T = \begin{bmatrix} X_T \\ Y_T \\ Z_T \end{bmatrix} = \begin{bmatrix} T \\ 0 \\ 0 \end{bmatrix} \quad (2.3.15)$$

Thrust Delay When thrust is commanded the force is not applied immediately. Various factors contribute to the delay, the main component being the inertia of the propellers. There are several methods to model thrust delay. A Padé approximation is usually used to model a pure delay. In this project the delay is modeled as a first order transfer function with time constant τ_T as used in [5] and [2]. The resulting dynamics can be expressed as

$$\dot{T} = -\frac{1}{\tau_T}T + \frac{1}{\tau_T}T_c \quad (2.3.16)$$

The time constant can be approximated through a practical experiment.

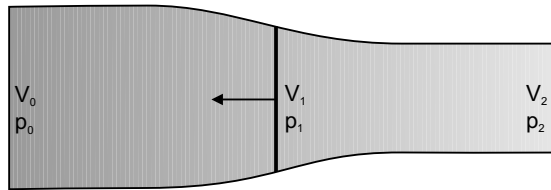


Figure 2.6: Propeller Thrust [9]

Induced Velocity According to [9] the thrust T generated by a propeller given by

$$T = \rho AV_1(V_2 - V_0) \quad (2.3.17)$$

where V_0 , V_1 and V_2 represent the airspeed far ahead, immediately behind and far behind the propeller respectively as illustrated in figure 2.6. In addition V_1 is the average of V_0 and V_2 , i.e. $V_1 = \frac{1}{2}(V_2 + V_0)$. The difference between V_1 and V_0 is called the induced velocity w . Keeping in mind that the induced airflow covers some of the actuation surfaces it becomes important to calculate w . Equation (2.3.17) can be rewritten in terms of V_0 and w as

$$T = \rho A(V_0 + w)2w \quad (2.3.18)$$

This makes sense since $\rho A(V_0 + w)$ is the mass rate of flow through the propeller disc and $2w$ is the total change in flow velocity.

During forward flight the thrust exerted by the propellers is relatively low. This causes the induced velocity to be negligible and only the airspeed is needed for aerodynamic calculations. In contrast during hover the forward speed is low and thrust is very high. The majority of airflow diverted by actuation surfaces is generated by the propellers and not forward airspeed. It is therefore necessary to calculate this increase in airspeed in order to accurately simulate actuation during hover or low speed vertical flight.

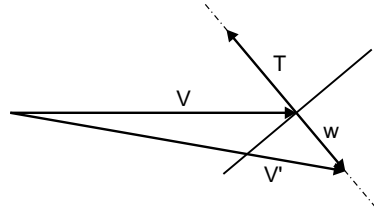


Figure 2.7: Induced Airspeed Vector Diagram [9]

$$T = \rho A V' 2w \quad (2.3.19)$$

$$V' = \sqrt{(V \cos \alpha + w)^2 + (V \sin \alpha)^2} \quad (2.3.20)$$

This leads to a fourth order equation in w :

$$w^4 + (2V \cos \alpha)w^3 + (V^2)w^2 = \left(\frac{T}{2\rho A} \right)^2 \quad (2.3.21)$$

Solving this equation provides the induced velocity. V_2 is then used to calculate forces exerted by actuation surfaces that lie within the air stream of the propellers. Newton's iterative method is used to solve the induced velocity when the system is relinearised.

2.4 LQR Model

2.4.1 Non-Linear State Equations

All the preceding differential equations are now combined to form a non-linear state space model linear in control vector \mathbf{u} , given in [5] as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u} \quad (2.4.1)$$

$$\mathbf{f}(\mathbf{x}) = [f_V(\mathbf{x}) \quad f_\alpha(\mathbf{x}) \quad \dots \quad f_T(\mathbf{x})]^T \quad (2.4.2)$$

$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} g_{V_{T_c}}(\mathbf{x}) & g_{V_{\delta_e}}(\mathbf{x}) & \dots & g_{V_{\delta_v}}(\mathbf{x}) \\ g_{\alpha_{T_c}}(\mathbf{x}) & g_{\alpha_{\delta_e}}(\mathbf{x}) & \dots & g_{\alpha_{\delta_v}}(\mathbf{x}) \\ \vdots & \vdots & \ddots & \vdots \\ g_{T_{T_c}}(\mathbf{x}) & g_{T_{\delta_e}}(\mathbf{x}) & \dots & g_{T_{\delta_v}}(\mathbf{x}) \end{bmatrix} \quad (2.4.3)$$

$$\mathbf{g}(\mathbf{x})\mathbf{u} = [g_V(\mathbf{x}) \quad g_\alpha(\mathbf{x}) \quad \dots \quad g_T(\mathbf{x})]^T \quad (2.4.4)$$

An LQR control system was developed for each of the attitude descriptions discussed. The system using Euler angles was decoupled into lateral and longitudinal modes that are controlled separately.

Full State Model using Quaternions:

$$\mathbf{x} = [V \quad \alpha \quad \beta \quad p \quad q \quad r \quad q_1 \quad q_2 \quad q_3 \quad q_4 \quad N \quad E \quad h \quad T]^T$$

$$\mathbf{u} = [T_c \quad \delta_e \quad \delta_a \quad \delta_r \quad \delta_{f_c} \quad \delta_{f_d} \quad \delta_v]^T$$

Full State Model using Euler Angles:

$$\mathbf{x} = [V \quad \alpha \quad \beta \quad p \quad q \quad r \quad \phi \quad \theta \quad \psi \quad N \quad E \quad h \quad T]^T$$

$$\mathbf{u} = [T_c \quad \delta_e \quad \delta_a \quad \delta_r \quad \delta_{f_c} \quad \delta_{f_d} \quad \delta_v]^T$$

Decoupled Model using Euler Angles and Body Coordinated Position Errors:

$$\mathbf{x}_{long} = [V \quad \alpha \quad q \quad \theta \quad P_X \quad P_Z \quad T]^T$$

$$\mathbf{u}_{long} = [T_c \quad \delta_e \quad \delta_{f_c}]^T$$

$$\mathbf{x}_{lat} = [\beta \quad p \quad r \quad \phi \quad \psi \quad P_Y]^T$$

$$\mathbf{u}_{lat} = [\delta_a \quad \delta_r \quad \delta_{f_d} \quad \delta_v]^T$$

The kinematics and force model are combined to form the complete non-linear model. When setting up the non-linear equations it is easier to not be restricted to a specific attitude description. Attitude states are only introduced into the equations through the DCM. To ensure generality (and simplify switching between descriptions) only the elements of the DCM will be

used according to the following notation:

$$[\mathbf{DCM}^{BE}] = \begin{bmatrix} e_{11}^{BE} & e_{12}^{BE} & e_{13}^{BE} \\ e_{21}^{BE} & e_{22}^{BE} & e_{23}^{BE} \\ e_{31}^{BE} & e_{32}^{BE} & e_{33}^{BE} \end{bmatrix} \quad (2.4.5)$$

Velocity

$$\begin{aligned} f_V = & \frac{\rho S b C_{y_p}}{4m} (V \alpha \beta r + V \beta p) - V \alpha^2 \beta r + \frac{\rho S C_{y_\beta}}{2m} V^2 \beta^2 \\ & - \left(\frac{\rho S C_{D_0}}{2m} + \frac{\rho S C_{L_0}^2}{2m \pi A e} \right) V^2 - \frac{\rho S C_{L_0} C_{L_\alpha}}{m \pi A e} V^2 \alpha \\ & - \frac{\rho S C_{L_\alpha}^2}{2m \pi A e} V^2 \alpha^2 + g (\alpha e_{11}^{BE} + \beta e_{12}^{BE} + e_{13}^{BE}) + \frac{1}{m} T \end{aligned} \quad (2.4.6)$$

$$\begin{aligned} f_\alpha = & q - \frac{\rho S C_{L_0}}{2m} V - \frac{\rho S C_{L_\alpha}}{2m} V \alpha - \alpha \beta r \\ & - \beta p - \frac{g}{V} (\alpha e_{13}^{BE} + e_{33}^{BE}) - \frac{\alpha T}{m V} \end{aligned} \quad (2.4.7)$$

$$\begin{aligned} f_\beta = & \frac{\rho S b C_{Y_p}}{4m} (p + \alpha r) + \alpha p - r \\ & + \left(\frac{\rho S C_{Y_\beta}}{2m} + \frac{\rho S C_{D_0}}{2m} + \frac{\rho S C_{L_0}^2}{2m \pi A e} \right) V \beta \\ & \frac{\rho S C_{L_0} C_{L_\alpha}}{m \pi A e} V \alpha \beta + \frac{\rho S C_{L_\alpha}^2}{2m \pi A e} V \alpha^2 \beta \\ & \frac{g}{V} (-\beta e_{13}^{BE} + e_{23}^{BE} - \alpha \beta e_{33}^{BE}) \end{aligned} \quad (2.4.8)$$

Angular Rates

$$\begin{aligned}
f_p = & -\frac{I_{zz} - I_{yy}}{I_{xx}}qr + \frac{\rho S b C_{l_\beta}}{2I_{xx}}V^2\beta - \frac{\rho S b C_{n_\beta}}{2I_{xx}}V^2\alpha\beta \\
& + \frac{\rho S b^2 C_{l_p}}{4I_{xx}}(Vp + V\alpha r) - \frac{\rho S b^2 C_{n_p}}{4I_{xx}}(V\alpha p + V\alpha^2 r) \\
& + \frac{\rho S b^2 C_{l_r}}{4I_{xx}}(Vr - V\alpha p) + \frac{\rho S b^2 C_{n_r}}{4I_{xx}}(V\alpha^2 p - V\alpha r)
\end{aligned} \quad (2.4.9)$$

$$f_q = \frac{\rho S c C_{m_0}}{2I_{yy}}V^2 - \frac{I_{xx} - I_{zz}}{I_{yy}}pr + \frac{\rho S c C_{m_\alpha}}{2I_{yy}}V^2\alpha + \frac{\rho S c^2 C_{m_q}}{4I_{yy}}Vq \quad (2.4.10)$$

$$\begin{aligned}
f_r = & -\frac{I_{yy} - I_{xx}}{I_{zz}}pq + \frac{\rho S b C_{n_\beta}}{2I_{zz}}V^2\beta + \frac{\rho S b C_{l_\beta}}{2I_{zz}}V^2\alpha\beta \\
& + \frac{\rho S b^2 C_{n_p}}{4I_{zz}}(Vp + V\alpha r) + \frac{\rho S b^2 C_{l_p}}{4I_{zz}}(V\alpha p + V\alpha^2 r) \\
& + \frac{\rho S b^2 C_{n_r}}{4I_{zz}}(Vr - V\alpha p) + \frac{\rho S b^2 C_{l_r}}{4I_{zz}}(V\alpha r - V\alpha^2 p)
\end{aligned} \quad (2.4.11)$$

Attitude Quaternions:

$$f_{q1} = \frac{1}{2}(q_4 p - q_3 q + q_2 r) \quad (2.4.12)$$

$$f_{q2} = \frac{1}{2}(q_3 p + q_4 q - q_1 r) \quad (2.4.13)$$

$$f_{q3} = \frac{1}{2}(-q_2 p + q_1 q + q_4 r) \quad (2.4.14)$$

$$f_{q4} = \frac{1}{2}(-q_1 p - q_2 q - q_3 r) \quad (2.4.15)$$

Euler 3-2-1:

$$f_\phi = (q \sin \phi + r \cos \phi) \tan \theta + p \quad (2.4.16)$$

$$f_\theta = (q \cos \phi - r \sin \phi) \quad (2.4.17)$$

$$f_\psi = (q \sin \psi + r \cos \phi) \sec \theta \quad (2.4.18)$$

Euler 2-3-1:

$$f_\phi = (-q \cos \phi + r \sin \phi) \tan \theta + p \quad (2.4.19)$$

$$f_\theta = (q \sin \phi + r \cos \phi) \quad (2.4.20)$$

$$f_\psi = (q \cos \psi - r \sin \phi) \sec \theta \quad (2.4.21)$$

Position Earth Coordinated Position Errors:

$$f_N = Ve_{11}^{BE} + V\beta e_{21}^{BE} + V\alpha e_{31}^{BE} \quad (2.4.22)$$

$$f_E = Ve_{12}^{BE} + V\beta e_{22}^{BE} + V\alpha e_{32}^{BE} \quad (2.4.23)$$

$$f_h = Ve_{13}^{BE} + V\beta e_{23}^{BE} + V\alpha e_{33}^{BE} \quad (2.4.24)$$

Body Coordinated Position Errors:

$$f_{P_X} = V \cos \alpha \cos \beta + qP_Z - rP_Y \quad (2.4.25)$$

$$f_{P_Y} = V \sin \beta + rP_X - pP_Z \quad (2.4.26)$$

$$f_{P_Z} = V \sin \alpha \cos \beta + pP_Y - qP_X \quad (2.4.27)$$

Thrust

$$f_T = -\frac{1}{\tau_e} T \quad (2.4.28)$$

Input equation

Velocity

$$g_{V_{\delta_a}} = \frac{\rho S C_{Y_{\delta_a}}}{2m} V^2 \beta \quad g_{V_{\delta_r}} = \frac{\rho S C_{Y_{\delta_r}}}{2m} V^2 \beta$$

$$g_{V_{\delta_v}} = \frac{\rho S C_{Y_{\delta_a}}}{2m} V^2 \beta$$

Angle of Attack

$$g_{\alpha_{\delta_e}} = -\frac{\rho S C_{L_{\delta_e}}}{2m} V \quad g_{\alpha_{\delta_v}} = -\frac{\rho S C_{L_{\delta_{fc}}}}{2m} V$$

Angle of Sideslip

$$g_{\beta_{\delta_a}} = \frac{\rho S C_{Y_{\delta_a}}}{2m} V \quad g_{\beta_{\delta_r}} = \frac{\rho S C_{Y_{\delta_r}}}{2m} V$$

$$g_{\beta_{\delta_v}} = \frac{\rho S C_{Y_{\delta_a}}}{2m} V$$

Roll Rate

$$g_{p_{\delta_a}} = \frac{\rho S b}{2I_{xx}} (C_{l_{\delta_a}} - C_{n_{\delta_a}} \alpha) V^2 \quad g_{p_{\delta_r}} = \frac{\rho S b}{2I_{xx}} (C_{l_{\delta_r}} - C_{n_{\delta_r}} \alpha) V^2 \quad (2.4.29)$$

$$g_{p_{\delta_v}} = \frac{\rho S b}{2I_{xx}} (C_{l_{\delta_v}} - C_{n_{\delta_v}} \alpha) V^2 \quad g_{p_{\delta_{fd}}} = \frac{\rho S b}{2I_{xx}} (C_{l_{\delta_{fd}}} - C_{n_{\delta_{fd}}} \alpha) V^2$$

Pitch Rate

$$g_{q_{\delta_e}} = \frac{\rho S c C_{m_{\delta_e}}}{2I_{yy}} V^2 \quad g_{q_{\delta_{fc}}} = \frac{\rho S c C_{m_{\delta_{fc}}}}{2I_{yy}} V^2$$

Yaw Rate

$$g_{r_{\delta_a}} = \frac{\rho S b}{2I_{xx}} (C_{n_{\delta_a}} + C_{l_{\delta_a}} \alpha) V^2 \quad g_{r_{\delta_r}} = \frac{\rho S b}{2I_{xx}} (C_{n_{\delta_r}} + C_{l_{\delta_r}} \alpha) V^2$$

$$g_{r_{\delta_v}} = \frac{\rho S b}{2I_{xx}} (C_{n_{\delta_v}} + C_{l_{\delta_v}} \alpha) V^2 \quad g_{r_{\delta_{fd}}} = \frac{\rho S b}{2I_{xx}} (C_{n_{\delta_{fd}}} + C_{l_{\delta_{fd}}} \alpha) V^2$$

Thrust

$$g_{T_{T_c}} = \frac{1}{\tau_c}$$

2.5 Time Scale Decoupled Model

The model created for the LQR control system contained position, attitude and velocity in a single system. The decoupling of the system was done along a Longitudinal/Lateral boundary. The time scale decoupled system offers a different approach by separating the fast and slow dynamics of a system as set out in [2].

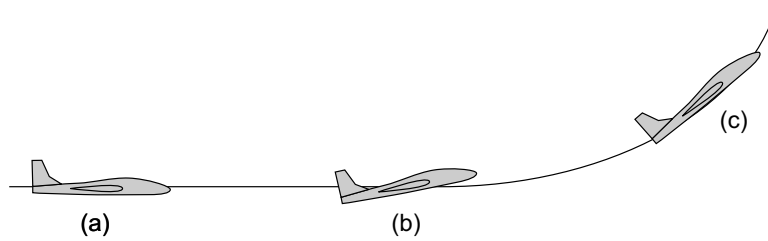


Figure 2.8: Time scale separation principle

When observing an aircraft in acrobatic flight it becomes clear that the position of the *body axis relative to the wind axis* changes significantly faster than the *wind axis relative to the inertial frame*. This is reflected by the trajectories flown in this project and illustrated in figure 2.8. When the aircraft pitches upward the angle of attack increases first (b) before the velocity vector changes direction (c). It is seen that the change in the body-states generates the acceleration that drives the change in the wind axis system.

To exploit this natural time scale separation in a control system it must be captured mathematically. The system will be separated at the wind axes accelerations that connect the fast and slow dynamics. Using techniques similar to those discussed in the previous section two separate sets of dynamics are created. The fast dynamics capture the movement of the body axis relative to the wind axis. Since the slow dynamics receives accelerations as inputs it is not aircraft specific and is referred to as the point mass dynamics and tracks the position of the aircraft through space.

2.5.1 Point Mass Dynamics (Slow)

Using basic kinematic equations and Newton's laws the relationship between position, velocity, acceleration and force can be expressed as

$$\frac{d}{dt}\mathbf{P}^{WE} = \mathbf{V}^{WE} \quad (2.5.1)$$

$$\frac{d}{dt}\mathbf{V}^{WE} = \mathbf{A}^{WE} \quad (2.5.2)$$

$$\mathbf{F} = m\mathbf{A}^{WE} \quad (2.5.3)$$

with the earth axes being assumed to be inertial. For the purposes of this controller position must be coordinated in an inertial frame while velocity and acceleration are expressed in the wind axis system. Keep in mind the inclusion of the Coriolis equation when a derivative is taken in a rotating reference frame.

$$\left. \frac{d}{dt} \mathbf{V}^{WE} \right|_E = \left. \frac{d}{dt} \mathbf{V}^{WE} \right|_W + \boldsymbol{\omega}^{WE} \times \mathbf{V}^{WE} = \mathbf{A}^{WE} \quad (2.5.4)$$

$$\Rightarrow \dot{\mathbf{V}}_W^{WE} + \mathbf{S}_{\boldsymbol{\omega}_W^{WE}} \mathbf{V}_W^{WE} = \mathbf{A}_W^{WE} \quad (2.5.5)$$

Substituting these into the kinematic equations yields

$$\dot{\mathbf{P}}_{WE}^E = [\mathbf{DCM}^{WE}]^T \mathbf{V}_W^{WE} \quad (2.5.6)$$

$$\dot{\mathbf{V}}_W^{WE} = -\mathbf{S}_{\boldsymbol{\omega}_W^{WI}} \mathbf{V}_W^{WE} + m^{-1} \mathbf{F}_W \quad (2.5.7)$$

$$\frac{d}{dt} [\mathbf{DCM}^{WE}] = -\mathbf{S}_{\boldsymbol{\omega}_W^{WE}} [\mathbf{DCM}^{WE}] \quad (2.5.8)$$

that describes the motion of a point mass through space with force as a driving input. The equations are rewritten in matrix form to form

$$\begin{bmatrix} \dot{e}_{11}^{WI} & \dot{e}_{12}^{WI} & \dot{e}_{13}^{WI} \\ \dot{e}_{21}^{WI} & \dot{e}_{22}^{WI} & \dot{e}_{23}^{WI} \\ \dot{e}_{31}^{WI} & \dot{e}_{32}^{WI} & \dot{e}_{33}^{WI} \end{bmatrix} = - \begin{bmatrix} 0 & -R_W & Q_W \\ R_W & 0 & -P_W \\ -Q_W & P_W & 0 \end{bmatrix} \begin{bmatrix} e_{11}^{WI} & e_{12}^{WI} & e_{13}^{WI} \\ e_{21}^{WI} & e_{22}^{WI} & e_{23}^{WI} \\ e_{31}^{WI} & e_{32}^{WI} & e_{33}^{WI} \end{bmatrix} \quad (2.5.9)$$

$$\begin{bmatrix} \dot{\bar{V}} \end{bmatrix} = \frac{1}{m} [\mathbf{X}_W] \quad (2.5.10)$$

$$\begin{bmatrix} \dot{P}_x \\ \dot{P}_y \\ \dot{P}_z \end{bmatrix} = \begin{bmatrix} e_{11}^{WI} \\ e_{12}^{WI} \\ e_{13}^{WI} \end{bmatrix} \bar{V} \quad (2.5.11)$$

with the constraints

$$\begin{bmatrix} Q_W \\ R_W \end{bmatrix} = \frac{1}{m\bar{V}} \begin{bmatrix} -Z_W \\ Y_W \end{bmatrix} \quad (2.5.12)$$

Note that the velocity is coordinated in the wind axes and only has a X component. The attitude representation is kept in general to illustrate that any method can be used to record the orientation.

2.5.2 Rotational Dynamics (Fast)

The rotational dynamics capture the movement of the body axes relative to the wind axes. First consider the rotation of the body axis in inertial space (earth axes). From Newton's law of rotational motion with a constant moment of inertia it follows that

$$\mathbf{M} = \mathbf{I}^B \left. \frac{d}{dt} \boldsymbol{\omega}^{BE} \right|_E \quad (2.5.13)$$

Coordinated the latter in the appropriate axis system yields

$$\mathbf{M}_B^B = \mathbf{I}_B^B \dot{\boldsymbol{\omega}}_B^{BE} + \boldsymbol{\omega}_B^{BE} \times (\mathbf{I}_B^B \boldsymbol{\omega}_B^{BE}) \quad (2.5.14)$$

By making $\boldsymbol{\omega}_B^{BE}$ the subject of the formula the dynamics of the wind axis angular velocity can be expressed as

$$\dot{\boldsymbol{\omega}}_B^{BE} = \mathbf{I}_B^B^{-1} \left[-\mathbf{S}_{\boldsymbol{\omega}_B^{BE}} \mathbf{I}_B^B \boldsymbol{\omega}_B^{BE} + \mathbf{M}_B^B \right] \quad (2.5.15)$$

To capture the dynamics of α and β the rotational of the body axes relative to the wind axes is investigated. The wind axes are defined such that the Z-axis lies in the aircraft's plane of symmetry, resulting in

$$\mathbf{k}^W \cdot \mathbf{j}^B = 0 \quad (2.5.16)$$

This condition must hold for all time and the derivative is therefore also equal to zero. The dot product is a scalar quantity, so the derivative can be taken with respect to any axis system.

$$\left. \frac{d}{dt} \mathbf{k}^W \cdot \mathbf{j}^B \right|_W = 0 \quad (2.5.17)$$

$$\mathbf{k}^W \cdot \left. \frac{d}{dt} \mathbf{j}^B \right|_W + \left. \frac{d}{dt} \mathbf{k}^W \right|_W \cdot \mathbf{j}^B = 0 \quad (2.5.18)$$

$$\mathbf{k}^W \cdot \left[\left. \frac{d}{dt} \mathbf{j}^B \right|_B + \boldsymbol{\omega}^{BW} \times \mathbf{j}^B \right] = 0 \quad (2.5.19)$$

$$\mathbf{k}^W \cdot \left[\boldsymbol{\omega}^{BW} \times \mathbf{j}^B \right] = 0 \quad (2.5.20)$$

This constraint holds when $\boldsymbol{\omega}^{BW}$ lies in the plane spanned by \mathbf{j}^B and \mathbf{k}^W . Therefore $\boldsymbol{\omega}^{BW}$ can be written as

$$\boldsymbol{\omega}^{BW} = \boldsymbol{\omega}^{BE} - \boldsymbol{\omega}^{WI} = a\mathbf{j}^B + b\mathbf{k}^W \quad (2.5.21)$$

The constraint is enforced through an appropriate selection of P_W [2]. From the definition of angle of attack and angle of side slip equation 2.5.21 can be expressed as

$$\boldsymbol{\omega}_B^{BW} = \boldsymbol{\omega}_B^{BE} - \boldsymbol{\omega}_B^{WI} = \dot{\alpha}\mathbf{j}_B^B - \dot{\beta}\mathbf{k}_B^W \quad (2.5.22)$$

when coordinated in the body axes. $\boldsymbol{\omega}_B^{WI}$ and \mathbf{k}_B^W are transformed to the wind axis system resulting in

$$\boldsymbol{\omega}_B^{BW} = \boldsymbol{\omega}_B^{BE} - \left[\mathbf{DCM}^{WB} \right]^T \boldsymbol{\omega}_W^{WI} = \dot{\alpha}\mathbf{j}_B^B - \dot{\beta} \left[\mathbf{DCM}^{WI} \right]^T \mathbf{k}_W^W \quad (2.5.23)$$

When written in matrix form this becomes

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} - \begin{bmatrix} C_\alpha C_\beta & -C_\alpha S_\beta & -S_\alpha \\ S_\beta & C_\beta & 0 \\ S_\alpha C_\beta & -S_\alpha S_\beta & C_\alpha \end{bmatrix} \begin{bmatrix} P_W \\ Q_W \\ R_W \end{bmatrix} = \begin{bmatrix} 0 & S_\alpha \\ 1 & 0 \\ 0 & -C_\alpha \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \end{bmatrix} \quad (2.5.24)$$

The equations can be rearranged to make the vector $[\dot{\alpha} \ \dot{\beta} \ P_W]^T$ the subject of the formula. Q_W and R_W are replaced by the constraints derived in the previous section. The complete rotational dynamics are formed by combining the angular velocity dynamics (2.5.15) and the α and β dynamics (2.5.24) to form

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -C_\alpha T_\beta & 1 & -S_\alpha T_\beta \\ S_\alpha & 0 & -C_\alpha \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \frac{1}{m\bar{V}} \begin{bmatrix} C_\beta^{-1} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} Z_W \\ Y_W \end{bmatrix} \quad (2.5.25)$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \mathbf{I}_B^{-1} \left(- \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \mathbf{I}_B \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \begin{bmatrix} L \\ M \\ N \end{bmatrix} \right) \quad (2.5.26)$$

$$P_W = \begin{bmatrix} C_\alpha C_\beta^{-1} & 0 & S_\alpha C_\beta^{-1} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \frac{1}{m\bar{V}} \begin{bmatrix} -T_\beta & 0 \end{bmatrix} \begin{bmatrix} Z_W \\ Y_W \end{bmatrix} \quad (2.5.27)$$

with the constraint on P_W ensuring that equation (2.5.21) remains valid.

2.5.3 Linear Force Model

The force model describes all the specific forces generated by the motion and form of the aircraft and is based on work done in [8]. Delay effects such as downwash lag have been ignored. The thrust is modeled as a body fixed force along the axial vector with a first order delay.

The rotational dynamics require forces in the wind axes and moments in the body axes. Aerodynamic coefficients are coordinated for the wind axes and so no additional calculations are required for the forces. Moments can be transformed to body axes by using a DCM, but it can be shown that for small incidence angles the discrepancy between wind and body moments are smaller than the inherent uncertainty of the aerodynamic coefficients [2].

$$\begin{bmatrix} X_W \\ Y_W \\ Z_W \end{bmatrix} = \frac{1}{2}\rho\bar{V}^2 S \begin{bmatrix} -C_D \\ C_y \\ -C_L \end{bmatrix} + \begin{bmatrix} \cos\alpha \cos\beta \\ -\cos\alpha \sin\beta \\ -\sin\alpha \end{bmatrix} T \quad (2.5.28)$$

$$\begin{bmatrix} L_W \\ M_W \\ N_W \end{bmatrix} = \frac{1}{2}\rho\bar{V}^2 S \begin{bmatrix} b & 0 & 0 \\ 0 & \bar{c} & 0 \\ 0 & 0 & b \end{bmatrix} \begin{bmatrix} C_l \\ C_m \\ C_n \end{bmatrix} \quad (2.5.29)$$

The aerodynamic coefficients can be calculated as follows.

$$C_D = C_{D_0} + \frac{C_L^2}{\pi A e} \quad (2.5.30)$$

$$\begin{aligned} \begin{bmatrix} C_y \\ C_L \end{bmatrix} &= \begin{bmatrix} 0 \\ C_{L_0} \end{bmatrix} + \begin{bmatrix} 0 & C_{Y\beta} & \frac{b}{2V} C_{Yp} & 0 & \frac{b}{2V} C_{Yr} \\ C_{L\alpha} & 0 & 0 & \frac{\bar{c}}{2V} C_{Lq} & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ p \\ q \\ r \end{bmatrix} \\ &+ \begin{bmatrix} C_{Y\delta_A} & 0 & C_{Y\delta_R} \\ 0 & C_{L\delta_E} & 0 \end{bmatrix} \begin{bmatrix} \delta_A \\ \delta_E \\ \delta_R \end{bmatrix} \end{aligned} \quad (2.5.31)$$

$$\begin{aligned} \begin{bmatrix} C_l \\ C_m \\ C_n \end{bmatrix} &= \begin{bmatrix} 0 \\ C_{m_0} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & C_{l\beta} & \frac{b}{2V} C_{lp} & 0 & \frac{b}{2V} C_{lr} \\ C_{m\alpha} & 0 & 0 & \frac{\bar{c}}{2V} C_{mq} & 0 \\ 0 & C_{n\beta} & \frac{b}{2V} C_{np} & 0 & \frac{b}{2V} C_{nr} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ p \\ q \\ r \end{bmatrix} \\ &+ \begin{bmatrix} C_{l\delta_A} & 0 & C_{l\delta_R} \\ 0 & C_{m\delta_E} & 0 \\ C_{n\delta_A} & 0 & C_{n\delta_R} \end{bmatrix} \begin{bmatrix} \delta_A \\ \delta_E \\ \delta_R \end{bmatrix} \end{aligned} \quad (2.5.32)$$

2.5.4 Complete Model

To form the inner loop dynamics the force model (equations (2.5.28) to (2.5.32)) is substituted into the rotational dynamics (equations (2.5.25) and (2.5.27)). The model can easily be decoupled into axial, normal and lateral components as shown in [2]. The decoupled models will be given in chapter 4.

Chapter 3

Linear Quadratic Regulator

3.1 Introduction to LQR

3.1.1 Feedback Basics

The need for feedback control arises from various unknowns entering the system. A perfectly defined noiseless system can be controlled using open loop control, but unfortunately this is never the case in practice. The immense complexity of real life systems coupled with the presence of noise on sensors and actuators rules out an exact mathematical representation. Feedback makes it possible to control a system containing an acceptable amount of unknowns.

3.1.2 Linear Quadratic Regulator

The design of an LQR control system is based on minimising a quadratic cost function [13]. The cost function is constructed by assigning a certain weighting to each state and input corresponding to their relative *importance*, resulting in

$$J = \frac{1}{2} \sum_{k=0}^N \left[\mathbf{x}(k)^T Q_1 \mathbf{x}(k) + \mathbf{u}(k)^T Q_2 \mathbf{u}(k) \right] \quad (3.1.1)$$

Q_1 and Q_2 represent the state and control weighting matrices respectively. The matrices must be positive definite and are usually diagonal. It is difficult to intuitively understand the effect of off diagonal weightings and therefore they are seldom used. The control input $\mathbf{u}(k)$ is chosen to minimize the cost function J over the period $k = 0 \dots N$. As shown in [13] the solution takes the form of a difference equation known as the discrete matrix Riccati equation. The solution to this equation is suited to feedback control since it has the form of full state feedback.

$$\mathbf{u}(k) = -K(k)\mathbf{x}(k) \quad (3.1.2)$$

To find the feedback matrix K the following iterative method is applied for $k = N \dots 0$ with initial condition $S(N) = Q_1$

$$M(k) = S(k) - S(k) \left[Q_2(k) + \Gamma(k)^T S(k) \Gamma(k) \right]^{-1} \Gamma(k)^T S(k) \quad (3.1.3)$$

$$S(k-1) = \Phi(k)^T M(k) \Phi(k) + Q_1 \quad (3.1.4)$$

$$K(k-1) = \left[Q_2(k) + \Gamma(k)^T S(k) \Gamma(k) \right]^{-1} \Gamma(k)^T S(k) \Phi(k) \quad (3.1.5)$$

Notice that the feedback matrix K is not required for the iterative method to continue and so to minimize the computation time K is only calculated in the final time step.

3.1.3 Prediction Horizon

When calculating the gain a prediction horizon (N) needs to be chosen to specify the number of time steps for which the Ricatti equation is solved i.e. $k = N \dots 0$. Several aspects need to be considered when this choice is made, mainly relating to the conversion speed and computation time. A previous project, [5], used a horizon of 10 time steps and calculated new gains at every step. In this project a different approach is used. The dynamics of the system does not change on the same timescale as the controller operates. Therefore it is possible to use the same gains over a longer period. In fact LQR is only provably stable if feedback gains are kept constant [13]. Relinearizing the system for small changes is not necessary since these changes are both small and short lived. The system only needs to be relinearised when a major change occurs, as during trajectory flight. For this implementation the model is linearised and gains calculated every 20 time steps, but the prediction horizon is increased to 200 time steps.

Model Predictive Control In the preceding project [5] Model Predictive Control was used. When using MPC the dynamics used to calculate LQR gains change over time as predicted by the trajectory. As shown in figure 3.1 the model is linearised for at each step in the prediction horizon using the *expected future models*. In contrast *static* LQR uses the current linearised model for all steps in the horizon. The reasoning is that MPC will result in better performance as changes in the model will be pre-empted. This method of performing LQR was abandoned in this project for two reasons. Firstly it has not been shown to deliver better, or provably stable results. The second problem arises when the aircraft deviates from the specified trajectory, which occurs frequently. Since MPC uses future information of the trajectory it *has* to use trajectory states to linearised the model. Considering that the model could deviate from the trajectory quite substantially a large modelling error is introduced. *Static* LQR used in this project linearised the model around

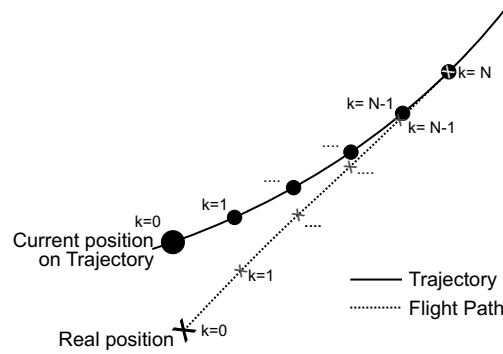


Figure 3.1: Model Predictive Control uses Expected Future Models

its current estimated states thereby ensuring optimal control *at that time*. The only way the advantage of *future* knowledge gained from MPC can be used in conjunction with the good *present* knowledge offered by static LQR is to continually update the trajectory. A real time trajectory will have to be calculated to return the aircraft to the originally defined path. The computational complexity this introduces is huge and beyond the scope of this project, especially considering the unproven results of MPC.

3.2 Controller Structure

3.2.1 Linearisation

In order to apply LQR to a system a linear model must be found. Small disturbance theory will be used to find an approximate linear model of the non-linear system, as in [5]. Perturbations are defined as the difference between the true state and the specified trajectory. The input perturbation is the input applied to correct this error.

$$\mathbf{x} = \mathbf{x}_p + \mathbf{x}_t \quad (3.2.1)$$

$$\mathbf{u} = \mathbf{u}_p + \mathbf{u}_t \quad (3.2.2)$$

A set of dynamic equations is derived to describe the behaviour of the perturbations.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u} \quad (3.2.3)$$

$$\dot{\mathbf{x}}_p + \dot{\mathbf{x}}_t = \mathbf{f}(\mathbf{x}_p + \mathbf{x}_t) + \mathbf{g}(\mathbf{x}_p + \mathbf{x}_t)\mathbf{u} \quad (3.2.4)$$

$$\dot{\mathbf{x}}_p = \mathbf{f}(\mathbf{x}_p + \mathbf{x}_t) + \mathbf{g}(\mathbf{x}_p + \mathbf{x}_t)\mathbf{u} - \dot{\mathbf{x}}_t \quad (3.2.5)$$

A Taylor series expansion performed about the trajectory yields

$$\begin{aligned} \dot{\mathbf{x}}_p &= \mathbf{f}(\mathbf{x}_t) + \mathbf{g}(\mathbf{x}_t)\mathbf{u}_t - \dot{\mathbf{x}}_t & (3.2.6) \\ &+ \left[\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right]_{\mathbf{x}_t} (\mathbf{x} - \mathbf{x}_t) + \left[\frac{\partial \mathbf{g}(\mathbf{x})\mathbf{u}}{\partial \mathbf{x}} \right]_{\mathbf{x}_t, \mathbf{u}_t} (\mathbf{x} - \mathbf{x}_t) \\ &+ \left[\frac{\partial \mathbf{g}(\mathbf{x})\mathbf{u}}{\partial \mathbf{u}} \right]_{\mathbf{x}_t, \mathbf{u}_t} (\mathbf{u} - \mathbf{u}_t) \\ &+ [\text{Higher order terms in } (\mathbf{x} - \mathbf{x}_t) \text{ and } (\mathbf{u} - \mathbf{u}_t)] \end{aligned}$$

The approximate dynamic function is found by truncating the higher order terms of the Taylor series yielding

$$\begin{aligned} \dot{\mathbf{x}}_p &\approx \left(\left[\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right]_{\mathbf{x}_t} + \left[\frac{\partial \mathbf{g}(\mathbf{x})\mathbf{u}}{\partial \mathbf{x}} \right]_{\mathbf{x}_t, \mathbf{u}_t} \right) \mathbf{x}_p + \mathbf{g}(\mathbf{x}_t)\mathbf{u}_p & (3.2.7) \\ &+ [\mathbf{f}(\mathbf{x}_t) + \mathbf{g}(\mathbf{x}_t)\mathbf{u}_t - \dot{\mathbf{x}}_t] \\ &= \mathbf{F}\mathbf{x}_p + \mathbf{G}\mathbf{u}_p + d \end{aligned}$$

The trajectory dynamics are given by $\dot{\mathbf{x}}_t = \mathbf{f}(\mathbf{x}_t) + \mathbf{g}(\mathbf{x}_t)\mathbf{u}_t$ which renders the last term of equation (3.2.7) approximately zero; depending on how well the trajectory has been calculated. Small mistakes in the trajectory are compensated for by the closed loop dynamics and are represented as noise d in the system. It is clear that the performance of the complete system relies as much on control as it does on the creation of accurate trajectories.

The perturbation dynamics are described by the Jacobian matrices \mathbf{F} and \mathbf{G} . The matrices are calculated by deriving the state equation defined in section 2.4.1 as shown below.

$$\mathbf{F} = \begin{bmatrix} \frac{\partial f_V}{\partial V} & \frac{\partial f_V}{\partial \alpha} & \cdots & \frac{\partial f_V}{\partial T} \\ \frac{\partial f_\alpha}{\partial V} & \frac{\partial f_\alpha}{\partial \alpha} & \cdots & \frac{\partial f_\alpha}{\partial T} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_T}{\partial V} & \frac{\partial f_T}{\partial \alpha} & \cdots & \frac{\partial f_T}{\partial T} \end{bmatrix}_{\mathbf{x}_t} + \begin{bmatrix} \frac{\partial g_V}{\partial V} & \frac{\partial g_V}{\partial \alpha} & \cdots & \frac{\partial g_V}{\partial T} \\ \frac{\partial g_\alpha}{\partial V} & \frac{\partial g_\alpha}{\partial \alpha} & \cdots & \frac{\partial g_\alpha}{\partial T} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_T}{\partial V} & \frac{\partial g_T}{\partial \alpha} & \cdots & \frac{\partial g_T}{\partial T} \end{bmatrix}_{\mathbf{x}_t, \mathbf{u}_t} \quad (3.2.8)$$

$$\mathbf{G} = \mathbf{g}(\mathbf{x}_t) \quad (3.2.9)$$

Calculating the Jacobian matrices is trivial and the derivatives will not be listed in the thesis as it would be extremely cumbersome.

3.2.2 Discretization and Integrators

The system is still in continuous form and must be converted to the discrete time domain. The Euler approximation is a computationally simple method to convert a state-space model to the discrete time domain. It remains valid

as long as the dynamics of the system does not approach the sampling frequency. The approximation states that

$$\Phi(k) \approx I + FT_s \quad (3.2.10)$$

$$\Gamma(k) \approx GT_s \quad (3.2.11)$$

Velocity enters into most terms in the linearised matrix since all aerodynamic and translation terms have a velocity component. The ability to accurately control velocity in the steady state is negatively affected by the imprecise value of C_{D_0} that is very difficult to measure or calculate. To this end an integrator is placed on velocity to ensure a zero steady state error. The discrete matrices are appended to include the integrated velocity error resulting in

$$H_I = [1 \ 0 \ 0 \ \dots \ . \ . \ \dots \ 0 \ 0] \quad (3.2.12)$$

$$\Phi_I(k) = \begin{bmatrix} T_s & H_I \\ \mathbf{0} & \Phi(k) \end{bmatrix} \quad (3.2.13)$$

$$\Gamma_I(k) = \begin{bmatrix} \mathbf{0} \\ \Gamma(k) \end{bmatrix} \quad (3.2.14)$$

3.2.3 Design Choices

Several design choices were presented pertaining to attitude description, coordinating position errors and decoupling. Each of these options has associated strengths and weaknesses. There are many ways to combine these choices to form a complete viable system. The three candidates listed here were tested extensively through the course of the project, but only one will be used throughout the remainder of this text.

3.2.3.1 Candidate 1

Area	Choice
Attitude Description	Quaternions
Position Errors	Earth Axes
Decoupled	No

Quaternions have the advantage of not exhibiting the discontinuities associated with Euler angles and do not require switching between attitude descriptions. However problems can occur at specific orientations where one quaternion becomes uncontrollable. Simulations show that the control system is still capable of controlling the aircraft, but the system is not provably stable. Robustness is one of the primary considerations of this project and the possibility of instability is not acceptable. Furthermore the non-linear nature of quaternions makes them unsuitable for use with LQR.

3.2.3.2 Candidate 2

Area	Choice
Attitude Description	Euler
Position Errors	Earth Axes
Decoupled	No

Despite the difficulties associated with Euler angles they still perform better than quaternions. To enable the aircraft to fly over the entire intended flight envelope more than one Euler sequence is used. Euler 231 is used during vertical flight while Euler 321 is used during conventional horizontal flight. If all three Euler angles are weighted equally the switch does not induce any transient response. It should be possible to weight Euler angles independently, but weightings would need to change during Euler switching to minimize transients.

3.2.3.3 Candidate 3

Area	Choice
Attitude Description	Euler
Position Errors	Body Axes
Decoupled	Yes

For the system to be decoupled the position states must be coordinated in the body axis, as explained in section 2.2.2. Unfortunately the position tracking response of the system worsened slightly and due to time constraints the exact cause could not be established. Decoupling also creates problems when Euler switching is performed. Since the definition of each Euler angle changes, the way the matrix is decoupled must also change. When Euler 321 is used the θ angle is included in the longitudinal model but in the case of Euler 231 θ describes lateral orientation. While this is easily implemented as part of Euler switching, the greater the number of these changes, the more likely unfavourable responses becomes. In cases where computational power is a restriction the vast speed improvement due to decoupling will overshadow the slight drop in trajectory tracking performance. There is definite merit in decoupling and further investigation can lead significant improvements.

3.2.3.4 Final Choice

Considering the performance of each candidate the full state Euler model (Candidate 2) is used throughout the rest of this project.

3.2.4 Overview

Figure 3.2 shows the general layout of the LQR controller. Note the points where the trajectory variables are included, creating a controller that only operates on deviations from the current trajectory. The Euler selection switch selects one of two Euler sequences. The controller runs at the sampling time $T_s = 20ms$, however calculating new controller gains at this rate is not practical. The model of the system does not change quickly and so the system is relinearized and new gains are calculated once every $T_l = 200ms$.

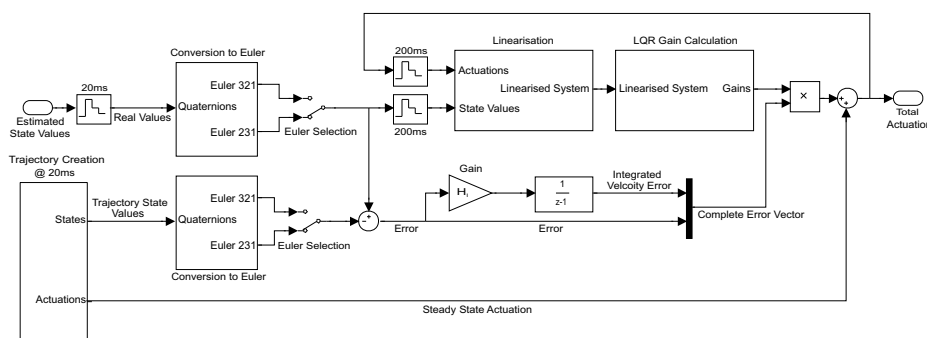


Figure 3.2: LQR Controller Block Diagram

3.3 Trajectory Creation

3.3.1 Methodology

As shown before, the accuracy of the trajectory greatly influences the performance of the complete system. Any errors introduced in the trajectory add noise to the system. Creating a trajectory therefore becomes a trade-off between performance and simplicity. A good control system should be able to correct acceptable trajectory errors. The following methods of trajectory creation were considered.

1. Mathematically calculate a complete trajectory including all states and actuations.
2. Record aircraft states as a safety pilot flies the desired trajectory.
3. Record aircraft states as the trajectory is flown in simulation using the non-linear model.
4. Create the trajectories intuitively.

5. Combine a number of these methods.

The first option will provide a very good trajectory for the mathematical model, but is very complicated and can only be used reliably on some states. The second and third rely on a pilot's ability to fly the trajectories optimally with the former carrying the added risk of damaging the aircraft. Trajectories would also have to be flown again whenever the aircraft model changes to a substantial degree. Option four is viable since the trajectories are relatively simple and forces and moments can easily be calculated. Other states can be approximated using general aircraft knowledge.

A combination of said techniques is used in this project. Some states are calculated intuitively (such as pitch angle), while others are calculated empirically (like elevator deflection). An important part of creating trajectories was iterative refinement. Once a trajectory was created the performance was evaluated and trajectory was changed to improve the system's response.

Early observations of the closed loop response showed that any discontinuity in the trajectory has serious negative effects. To prevent immediate changes in a state sinusoidal functions are used, ensuring that no steps occur in a state trajectory or its first derivative. The time required to perform a transition is determined by the accuracy of the trajectory and physical limitations of the airframe. If a trajectory is very accurate it introduces fewer errors into the system and the aircraft can complete the transition faster. Transition from vertical flight to forward flight takes five seconds, while transition back takes ten seconds. It is possible to complete both trajectories in five seconds under simulation conditions. When a craft pulls up at high speed the angle of attack increases quickly. In a five second transition from level flight the angle of attack can easily rise to over twenty degrees. A very high angle of attack can induce a non-linear response. The aircraft's large moments of inertia, its mass and relatively small actuation surfaces make substantially less manoeuvrable than the aircraft used in [1] and [5]. In light of the results obtained in [5] an equally manoeuvrable aircraft should be able to complete both transitions in three seconds or less. The sequence in which trajectories are calculated is chosen to ensure that empirical calculations rely only on previously calculated trajectories. Figure 3.3.1 shows the notation used when calculating the trajectories. The subscript V and L denote *constant* quantities during vertical and level flight respectively. Subscripts 1 and 2 are added to *functions* that describe the change of a state over the course of a transition. T_1 and T_2 refer to the duration of each transition. The trajectories are expressed in the continuous time domain. The control update frequency will determine how these functions are discretized.

Velocity During a transition the velocity of the aircraft will change from, V_L to V_V . Initially a linear function was used to specify the trajectory, but this was adapted to improve the response of thrust. A sudden change in

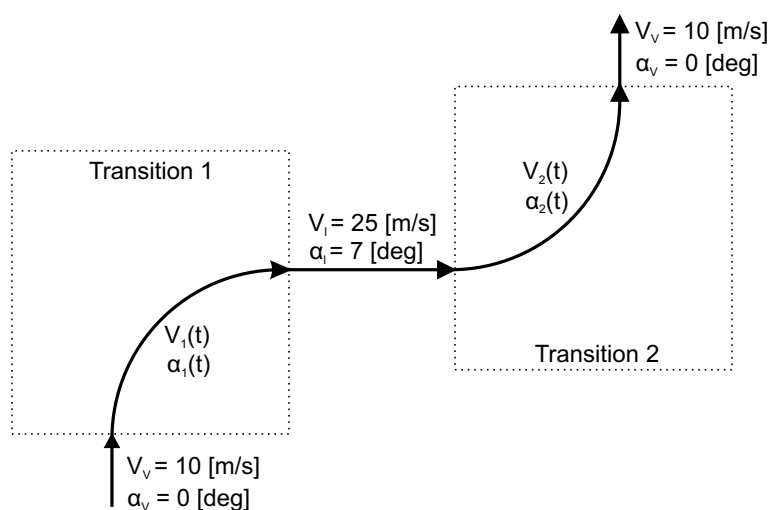


Figure 3.3: Diagram showing the notations used for trajectory state functions

acceleration command causes overshoot in the thrust actuation command. If the overshoot is too large the actuator will saturate and result in non-linear behaviour.

$$V_1(t) = V_v + \frac{V_l - V_v}{2} \left[1 - \cos\left(\frac{\pi}{T_1}\right) \right] \quad (3.3.1)$$

$$V_2(t) = V_l + \frac{V_v - V_l}{2} \left[1 - \cos\left(\frac{\pi}{T_2}\right) \right] \quad (3.3.2)$$

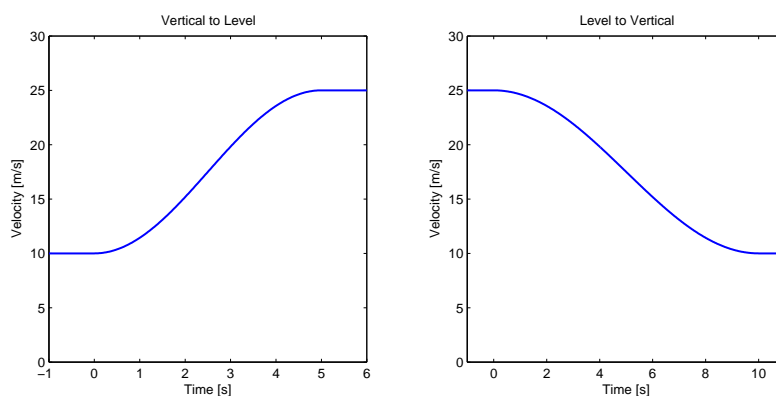


Figure 3.4: Velocity Trajectory

Angle of Attack Angle of attack is very difficult to calculate or predict. When the aircraft pulls up the angle of attack increases first, and then the direction of the velocity vector changes. The increase in angle of attack cannot be calculated easily and has to be approximated. The trajectory is found by weighting angle of attack very lightly and allowing the control system to fly the trajectory. The control system will generate the angle of attack require to keep the aircraft on its intended course. The value is recorded and approximated with cosine functions. This method is not accurate and as a result angle of attack was never heavily weighted in the LQR algorithm during transition. Despite this, the trajectory must be created as it forms part of subsequent state functions. During level or vertical flight the required angle of attack is well known and a slightly heavier weighting can be applied. The angle of attack is never given a very large weighting as it will impair the control system's ability to correct position errors. α_{max} is used to denote the highest value α reaches during the transition.

$$\alpha_1(t) = \alpha_l \sin\left(\frac{t}{T_1} \frac{\pi}{2}\right) \quad (3.3.3)$$

$$\alpha_2(t) = \begin{cases} \alpha_l + \frac{\alpha_{max} - \alpha_l}{2} \left[1 - \cos\left(\frac{t}{T_2/3} \pi\right)\right] & t \in [0, \frac{T_2}{3}) \\ \frac{\alpha_{max}}{2} \left[1 + \cos\left(\frac{t}{2T_2/3} \pi - \frac{\pi}{2}\right)\right] & t \in [\frac{T_2}{3}, T_2] \end{cases} \quad (3.3.4)$$

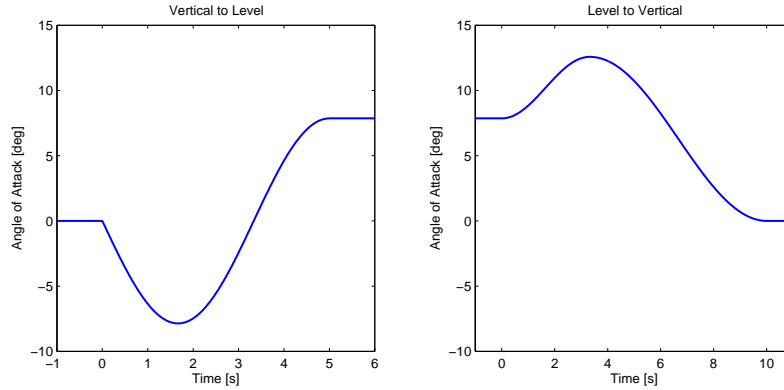


Figure 3.5: Angle of Attack Trajectory

Pitch Angle Even though the trajectories are stored using quaternions they are designed using Euler angles, as it is easier to visualize. The pitch angle of the wind axes is added to the angle of attack to find the body pitch angle

trajectory.

$$\theta_1(t) = \frac{\theta_v^w - \theta_s^w}{2} \left[1 + \cos\left(\frac{t}{T_1}\pi\right) \right] + \alpha_1(t) \quad (3.3.5)$$

$$\theta_2(t) = \frac{\theta_v^w - \theta_s^w}{2} \left[1 - \cos\left(\frac{t}{T_2}\pi\right) \right] + \alpha_2(t) \quad (3.3.6)$$

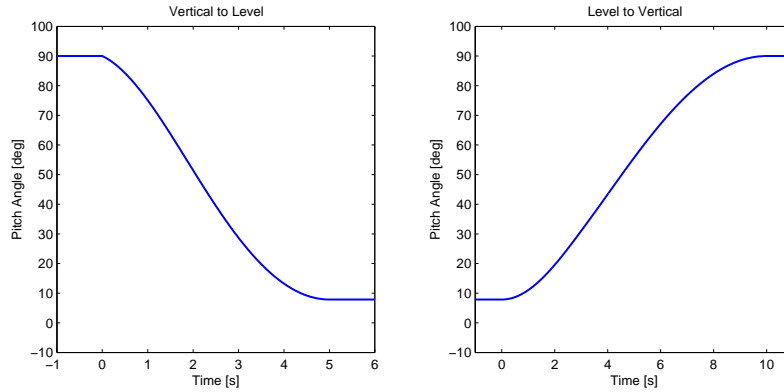


Figure 3.6: Pitch Angle Trajectory

It should be noted that the angle of attack forms part of the pitch angle trajectory. By applying a weight to the pitch angle the angle of attack is also inadvertently weighted. This is not optimal since it perpetuates the uncertainty introduced by the angle of attack trajectory to other states. An alternative is to describe the orientation of the wind axes and not the aircraft's body, thereby removing the uncertainty from the trajectory. This will negatively impact on the estimated value of orientation as the estimator has a direct measurement of body orientation, but not of wind axis orientation. Despite this it is believed that the estimator will provide better compensation for the uncertainty.

Pitch Rate To find the pitch rate the pitch angle is differentiated.

$$q_1(t) = \frac{d}{dt}\theta_1(t) \quad (3.3.7)$$

$$q_2(t) = \frac{d}{dt}\theta_2(t) \quad (3.3.8)$$

The definition of pitch angle creates a step reference in the pitch rate trajectory, since the angle of attack trajectory does not start with a zero derivative. This immediate change in pitch rate commanded by the trajectory is not

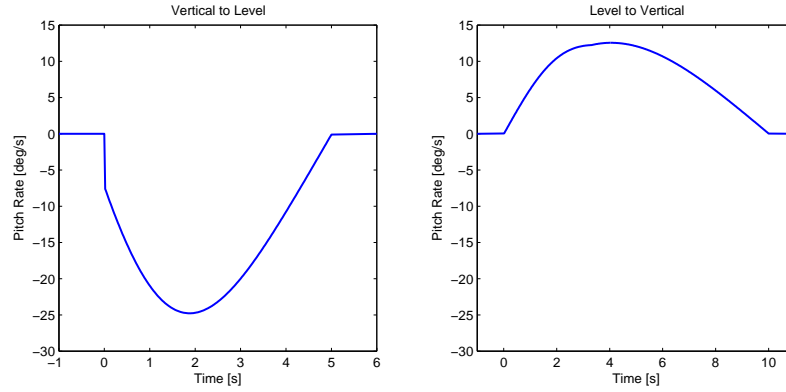


Figure 3.7: Pitch Rate Trajectory

physically possible, as it requires infinite angular acceleration. This error is small and causes very little overshoot in the elevator command. If a smooth transition is not used for pitch angle the overshoot will be substantial.

Position The position trajectory is calculated by combining the velocity, pitch angle and angle of attack information. It is important that the wind axes pitch angle be used to calculate displacement.

$$N_1(t) = N_0 + \int_0^t V_1(x) \cos[\theta_1(x) - \alpha_1(x)] dx \quad (3.3.9)$$

$$N_2(t) = N_0 + \int_0^t V_2(x) \cos[\theta_2(x) - \alpha_2(x)] dx \quad (3.3.10)$$

$$h_1(t) = h_0 + \int_0^t V_1(x) \sin[\theta_1(x) - \alpha_1(x)] dx \quad (3.3.11)$$

$$h_2(t) = h_0 + \int_0^t V_2(x) \sin[\theta_2(x) - \alpha_2(x)] dx \quad (3.3.12)$$

Thrust When calculating the actuation trajectory for thrust as many factors as possible must be considered, including inertia, gravity and drag. If an aircraft is decelerating it is important to ensure that the thrust command is always positive. To this end the velocity trajectory is chosen so that significant deceleration only occurs when the aircraft is already pitched thereby enabling it to be slowed by gravity. The downside of this method is that the angle of attack becomes very large when the aircraft pulls up with a high velocity.

$$T_1(t) = m \frac{d}{dt} [V_1(t)] + mg \sin(\theta_1(t) - \alpha_1(t)) + \frac{1}{2} \rho V_1(t)^2 S C_{D_0} \quad (3.3.13)$$

$$T_2(t) = m \frac{d}{dt} [V_2(t)] + mg \sin(\theta_2(t) - \alpha_2(t)) + \frac{1}{2} \rho V_2(t)^2 S C_{D_0} \quad (3.3.14)$$

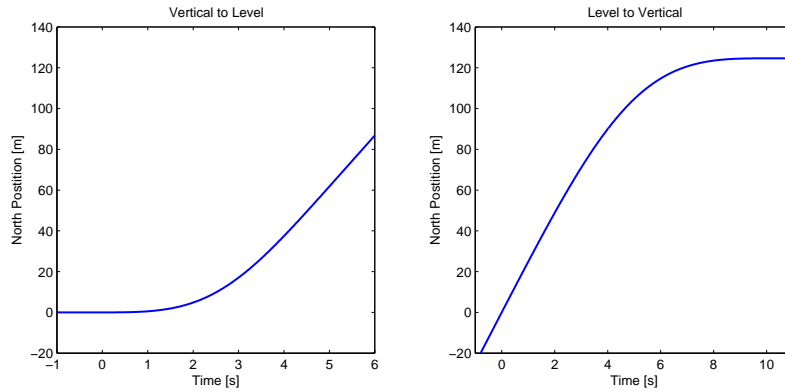


Figure 3.8: North Position Trajectory

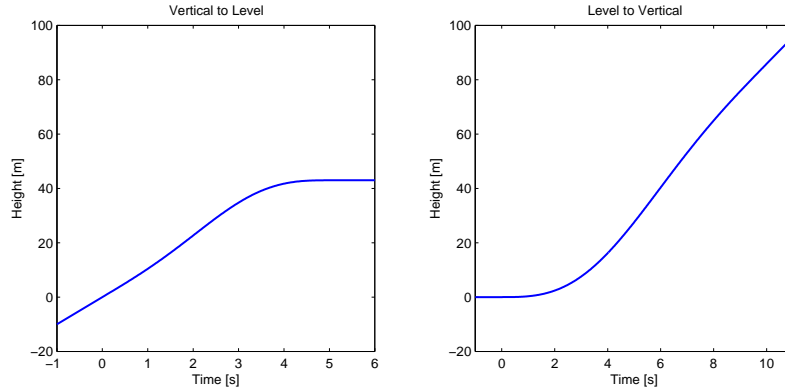


Figure 3.9: Height Trajectory

A possible remedy to implement a 2-part transition: A slow-down stage where the aircraft decelerates during level flight followed by a transition at low speed.

Elevator The elevator actuation trajectory is calculated empirically by investigating all the pitching moments acting on the craft together with the desired angular acceleration.

$$\mathbf{M} = \mathbf{I}\dot{\boldsymbol{\omega}} \tag{3.3.15}$$

$$I_{yy}\dot{q} = \frac{1}{2}\rho V^2 S c \left[C_{m_\alpha} \alpha + C_{m_{\delta_e}} \delta_e + \left(\frac{c}{2V} \right) C_{m_q} q \right] \tag{3.3.16}$$

It is however important to note that this calculation does not take into account the effect of induced velocity. The elevator actuation must be scaled to include the increased effectiveness of the actuation surfaces. $\omega(t)$ is cal-

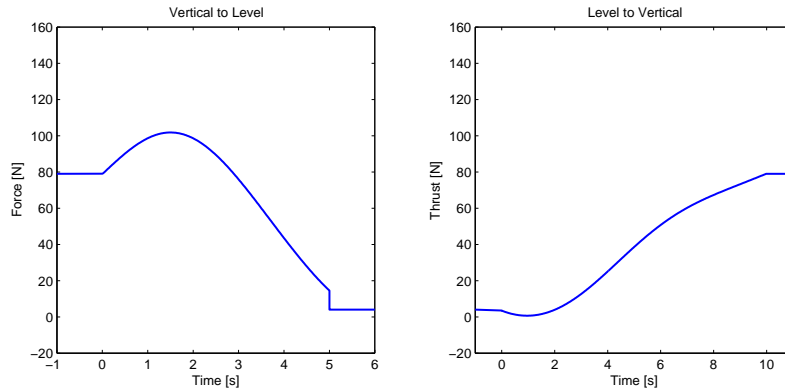


Figure 3.10: Thrust Trajectory

culated as described in section 2.3.4. Making δ_e the subject of the formula yields

$$\delta_e = \frac{1}{C_{m_{\delta_e}}} \left[\frac{2I_{yy}}{\rho V^2 S c} \dot{q} - C_{m_\alpha} \alpha - \left(\frac{c}{2V} \right) C_{m_q} q \right] \left(\frac{V(t)}{V(t) + 2w(t)} \right)^2 \quad (3.3.17)$$

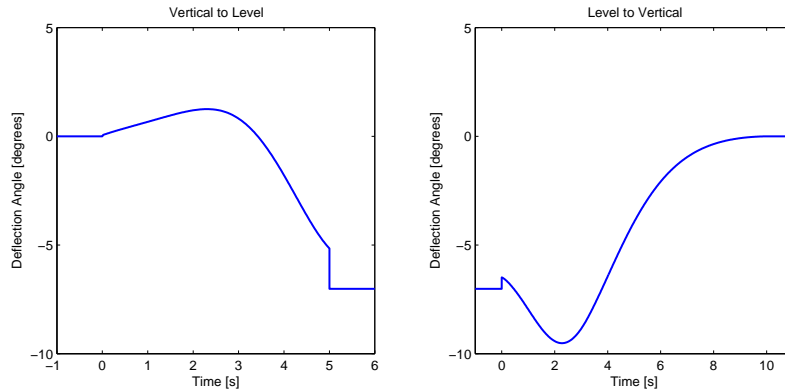


Figure 3.11: Elevator Deflection Trajectory

Banked Turn In addition to performing transitions between vertical and level flight the aircraft should also be able to navigate as a conventional fixed wing aircraft. The system is robust enough to ensure that only position and orientation references are needed to turn the aircraft. It is however much more efficient to calculate the bank angle, yaw rate and pitch rate required to perform a banked turn. This allows the aircraft to complete tighter turns

with smaller errors. The trajectory is created by first choosing a turn radius ρ . A negative turn radius implies a left turn, while a positive turn radius would result in a right turn. From here all other states can be calculated with w being the turn rate.

$$\phi_{turn} = \arctan\left(\frac{V^2}{\rho g}\right) \quad (3.3.18)$$

$$w_{turn} = \frac{V}{\rho} \quad (3.3.19)$$

$$q_{turn} = w_{turn} \sin \phi_T \quad (3.3.20)$$

$$r_{turn} = w_{turn} \cos \phi_T \quad (3.3.21)$$

Other states such as position and orientation are trivial to calculate. For a slow turn it is not necessary to command actuator inputs since the control system will respond quickly enough to track the commanded rates.

3.4 Weighting Selection

As shown before LQR controls a system by minimising a cost function. The cost function is calculated by applying weights to each state and actuation, usually in the form of a diagonal matrix. If a state's weighting is increased it is considered to be more important and the feedbacks will be calculated to control it more aggressively. It must be noted that complex coupling between states makes the actual outcome less predictable.

The weightings on control allow the designer to specify the amount of actuation used. For example the weighting of thrust will be less than that of elevator deflection since one unit of thrust (one newton) is significantly less than one unit of deflection (one radian or 57 degrees). An additional scaling factor R is added to weight all control signals. The effect of this scaling is to change bandwidth of the system as a whole.

It is important that no state should be given a zero weighting [13]. It may be necessary to weight a state very lightly, but zero weighting could result in an unstable system, or at best a system that is not provably stable.

$$Q_1 = \text{diag}([Q_{V_{int}} \quad Q_V \quad \dots \quad \dots \quad Q_h \quad Q_T]) \quad (3.4.1)$$

$$Q_2 = R \times \text{diag}\left(\left[Q_{T_c} \quad Q_{\delta_e} \quad Q_{\delta_a} \quad Q_{\delta_r} \quad Q_{\delta_{fc}} \quad Q_{\delta_{fd}} \quad Q_{\delta_v}\right]\right) \quad (3.4.2)$$

Many sources [14] suggest weighting states as the square of the inverse of the maximum desired deviation from the steady state value.

$$Q_N = \frac{1}{(\text{Max deviation in } N)^2} \quad (3.4.3)$$

State	Max Deviation	Control	Max Deviation
V_{int}	1 [m]	T_c	1 [N]
V	0.5 [m/s]	δ_e	1 [deg]
α, β	3 [deg]	δ_a	1 [deg]
p, q, r	0.2 [deg/s]	δ_r	1 [deg]
q_{1-4}	0.05 [ND]	δ_{fc}	1 [deg]
N, E, h	1 [m]	δ_{fd}	1 [deg]
T	1 [N]	δ_v	1 [deg]

Table 3.1: First Iteration Maximum Deviations

Note that this definition makes sense considering the units of Q_N since the cost $J = \mathbf{x}^T Q_1 \mathbf{x}$ must be dimensionless. Initial weightings were found using this method and then refined through iteration.

The results of simulations using these weightings made it clear that position and orientation were poorly regulated. This can be explained by the way a position or angle error would be corrected. To correct such an error other states would need to be perturbed. Consider the response to a height error.

- The *elevator* is deflected to provide a *pitch rate*.
- The pitch rate increases the *angle of attack* thereby increasing the lift, but also the drag.
- Increased drag will perturb the *velocity* and subsequently require additional *thrust*.

It is clear that the weightings of other states also have an effect on regulating position. To correct this, position and orientation need to be weighted heavier than most other states.

Caution must be taken when refining the state weightings by inspection alone. Apparent performance, especially during manoeuvres, can be greatly improved by *tweaking* the state weightings. This leads to a loss of insight and is strongly discouraged. A better approach is to find a valid reason for changing weightings whenever possible. There are several aspects to consider when evaluating new weightings and some of these will be discussed in the following sections.

3.4.1 Step Response

Step responses are a very good way to judge the robustness of a system. During trajectory flight the aircraft never deviates a great distance from the intended path and large errors never exist. There is a danger that in an attempt to improve trajectory tracking a system is made too aggressive. If a large

disturbance occurs, such as a wind gust, the system could become unstable before returning the correct path. To test the stability of the system when subjected to such disturbances position steps were used. In order to correct a position error nearly all other states will have to be perturbed significantly.

There are two methods of correcting a large position error. The heading can be altered, or the actuation surfaces can be used to move the aircraft directly in the desired direction, albeit slowly. The first method is usually used for navigation over large distances, while the second is suitable for small corrections. LQR employs a combination of these techniques depending on the exact state weighting.

The following sections will briefly explain how a system controlled by LQR responds to position step disturbances. It will be shown that the control system commands are similar to those of a pilot or conventional control system.

3.4.1.1 Level Flight

Normal Step In response to the step command the system pitches up, creating an angle of attack to supply more lift. The thrust is increased to overcome drag and further accelerate the aircraft upwards. Note that the angle of attack does not approach the stall point. This is an example of where too aggressive control can drive the system unstable.

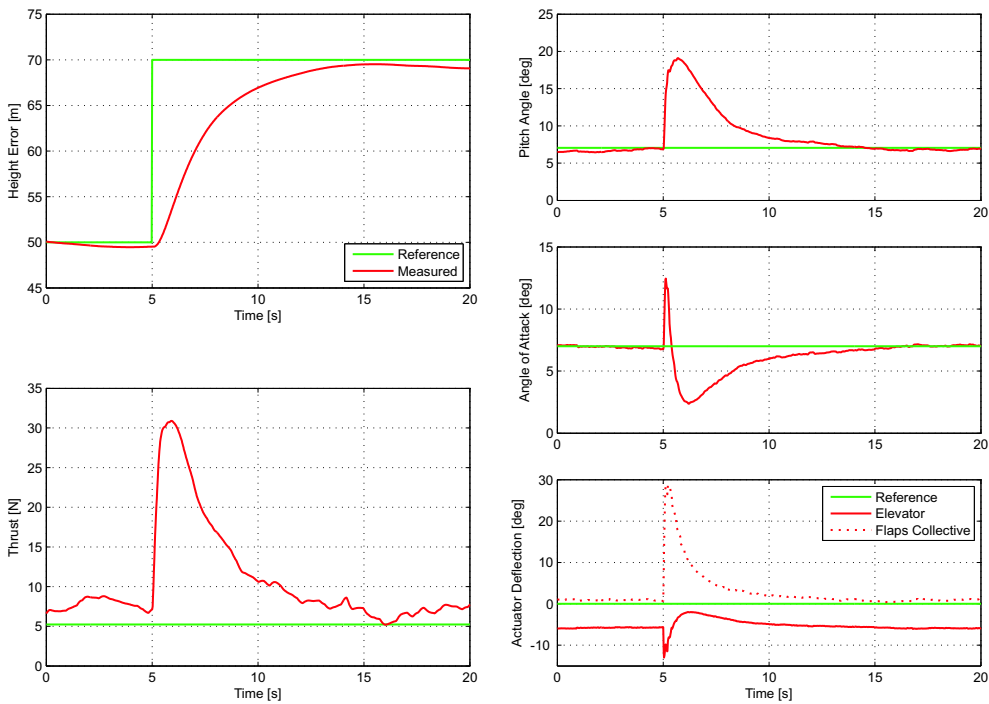


Figure 3.12: Normal step during level flight

Lateral Step Two actuations are used to correct a lateral step. The vertical flaps are actuated to create a side force. In addition the ailerons are used to roll the aircraft and redirect the lift vector to create a force in the desired direction. This is similar to how an actual pilot, or conventional control system, would perform a coordinated turn.

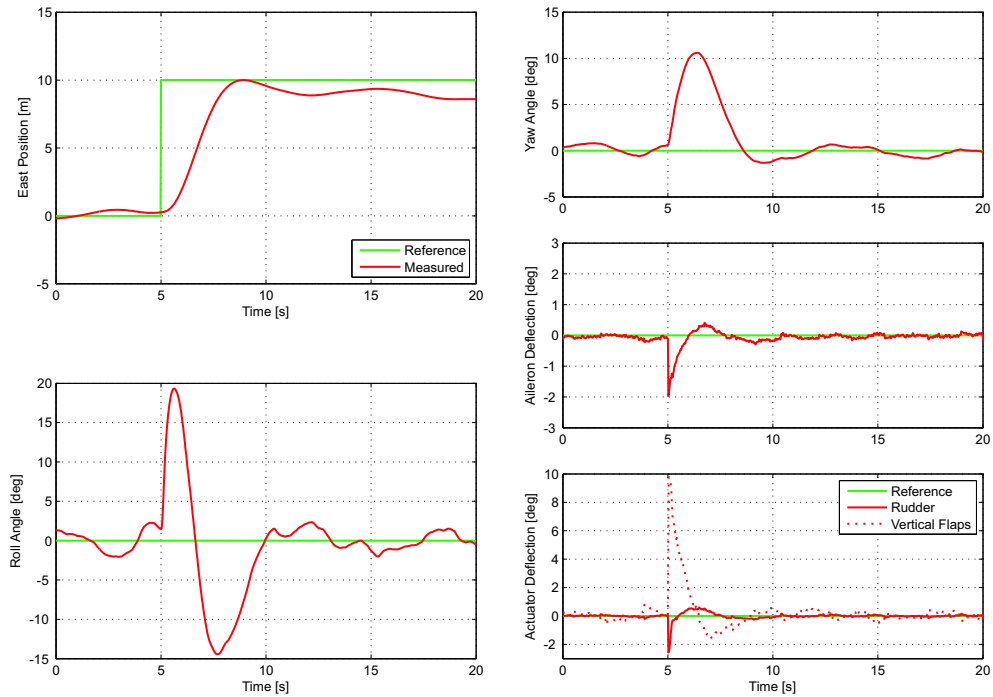


Figure 3.13: Lateral step during level flight

3.4.1.2 Vertical Flight

Normal Step To follow a normal step during vertical flight LQR deflects the elevator to pitch the aircraft down and uses the redirected thrust vector and negative lift to move in the desired direction. Usually this causes non-minimal phase, since the force created by the elevator is in the opposite direction. The NMP is corrected by applying the paddles collectively in the opposite direction. The force created by these two actuations is comparable, but the moment arms differ in length, resulting in a pitch rate in the desired direction without a force in the opposite direction. This is equivalent to moving the zero in the left half plane further from the origin.

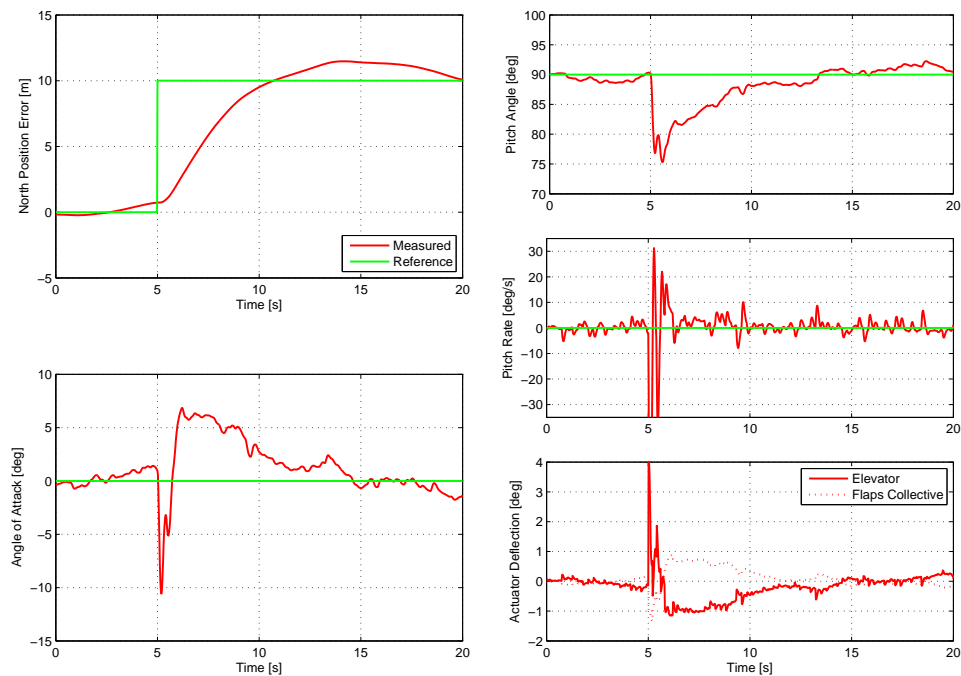


Figure 3.14: Normal step during vertical flight

Lateral Step It is clear that the lateral system is more difficult to control than the longitudinal. The rudder and vertical flaps are used in a similar way as the elevator and paddles during the normal step to combat non-minimum phase. The aileron and paddles are also used to correct roll errors caused by these actuation. The lateral step takes longer to settle since there is no significant lateral lift force: $|C_{Y\beta}| \ll |C_{L\alpha}|$.

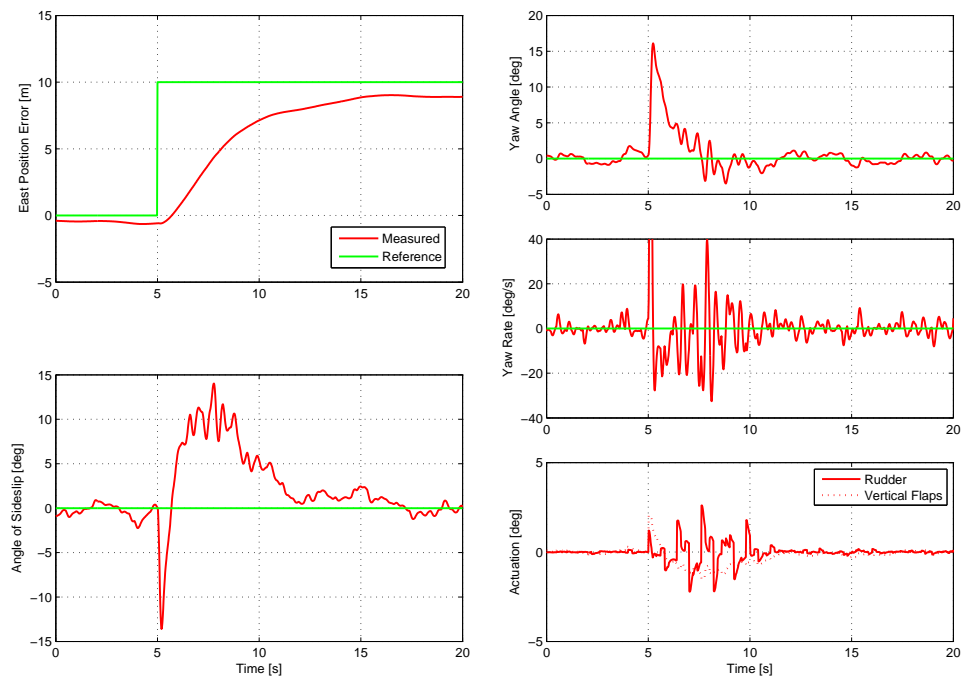


Figure 3.15: Lateral step during vertical flight

3.4.2 Pole Locations and Modal Analysis

Another way to gain insight into the response of an LQR system is to investigate where the poles have been placed. The natural modes of motion can also be identified. By weighting all the states contributing to a certain mode of motion the response of that specific mode can be improved. This is important when one state contributes significantly to more than one mode. For example: Trying to control the phugoid mode by increasing the weighting of the pitch angle will also affect the short period mode. In the extreme case the system will respond so aggressively to a pitch error that non-linear behaviour will be induced in the short period mode before the phugoid mode is significantly improved. The solution is to increase the weighting pitch angle, velocity and height to effectively change the response of the phugoid mode.

In summary the pole locations give the designer insight into *why* the system responds in a certain way. By examining the modal form of the system the designer can easily establish which states must be weighed to change a certain pole's location and what actuators affect what modes.

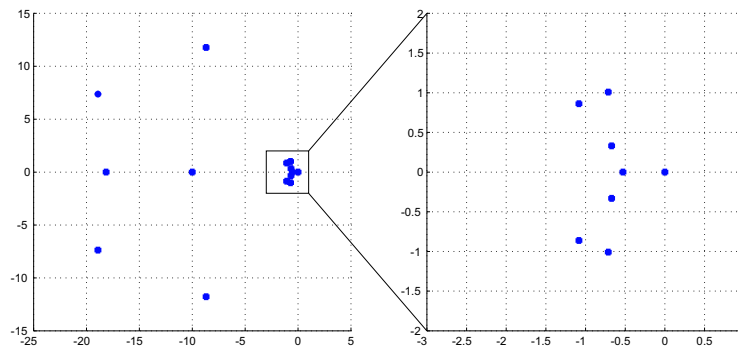


Figure 3.16: Level Flight Pole Locations

Pole Locations A few noteworthy observations can be made from the final pole locations:

- Most importantly all poles are placed in the left-half plane which results in a stable system. Of course, this does not guarantee that the system will be stable once non-linear effects have been included. It is the task of the designer to ensure that the poles are slow enough to ensure that non-linear effects (such as servo-slew rate) do not cause instability, but fast enough to effectively reject noise. Additionally some effects like uncontrollable states and gain settling can cause unstable poles. A pole plot is the fastest method of identifying such problems.

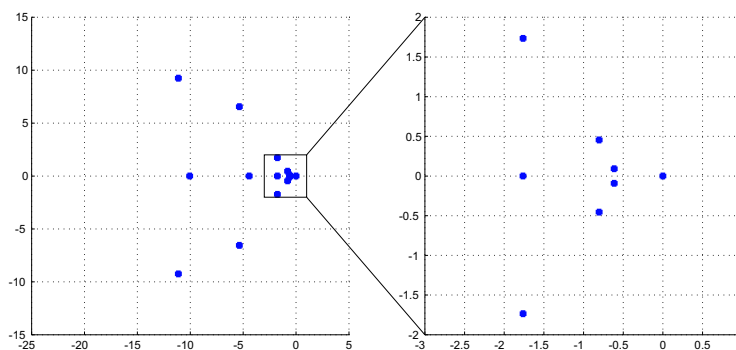


Figure 3.17: Vertical Flight Pole Locations

- There is a clear separation between slow and fast dynamics. Typically fast poles will represent modes with large rate components, while position modes tend to be slower. This fact is exploited in the TSD control system.
- Most complex poles have close to critical damping. If an angular rate has a particularly high weighting its fast modes will be very well damped.
- The lower speeds of vertical flight cause the aircraft to be less responsive to actuation. The weightings were changed to lower the bandwidth of the control system at low airspeeds.

Modal Analysis All conventional aircraft exhibit certain modes of motion. These modes have been well documented (see [16]) and describe the behaviour of the aircraft in the absence of control inputs. Traditionally an aircraft is controlled by moving the poles of these modes to more desirable locations. This section will attempt to show that LQR moves the poles in a similar way and to similar locations. The exact implications of each mode will not be discussed as the attraction of LQR lies in the fact that the system does not necessarily have to be understood in such depth. Note that the names given to these modes only apply to the uncontrolled poles, before the controller moves the poles from their natural position.

The phugoid mode is an under-damped exchange of energy from potential to kinematic form. It is clear that the mode is initially very poorly damped. Through control the poles are made faster and damping is improved, causing the aircraft to stabilize at a steady state height much faster.

The short period mode is a very fast oscillation around the Y-axis. The mode is damped significantly by the LQR system obtaining a similar result as the stability augmentation discussed in [1]. The damping prevents the large angular rate associated with this mode of motion.

The Dutch Roll Mode is essentially the lateral equivalent of the Short Period Mod: A fast oscillation about the Z-axis, damped by the weathercock

	Phugoid		Short Period	
	Uncontrolled	Controlled	Uncontrolled	Controlled
	$-0.006 \pm 0.462i$	$-0.714 \pm 1.008i$	$-10.44 \pm 24.204i$	-18.917 ± 7.374
V	$1.000 \angle 0^\circ$	$1.000 \angle 0^\circ$	$0.136 \angle -54^\circ$	$0.019 \angle -139^\circ$
α	$0.001 \angle 175^\circ$	$0.027 \angle 56^\circ$	$0.045 \angle -105^\circ$	$0.057 \angle -154^\circ$
q	$0.022 \angle -7^\circ$	$0.112 \angle 62^\circ$	$1.000 \angle 0^\circ$	$1.000 \angle 0^\circ$
ψ	$0.0477 \angle -97^\circ$	$0.091 \angle -64^\circ$	$0.042 \angle -116^\circ$	$0.049 \angle -159^\circ$

Table 3.2: Longitudinal Modes during Level Flight

	Dutch Roll		Roll	
	Uncontrolled	Controlled	Uncontrolled	Controlled
	$-3.334 \pm 14.06i$	$-8.695 \pm 11.776i$	-10.340	-18.143
β	$0.071 \angle 84^\circ$	$0.068 \angle 61^\circ$	$0.030 \angle 0^\circ$	$0.017 \angle 0^\circ$
p	$0.124 \angle -111^\circ$	$0.096 \angle -96^\circ$	$1.000 \angle 0^\circ$	$1.000 \angle 0^\circ$
r	$1.000 \angle 0^\circ$	$1.000 \angle 0^\circ$	$0.088 \angle 0^\circ$	$0.106 \angle 0^\circ$
ϕ	$0.090 \angle 146^\circ$	$0.007 \angle 136^\circ$	$0.097 \angle 180^\circ$	$0.055 \angle 180^\circ$
θ	$0.692 \angle -103^\circ$	$0.068 \angle -126^\circ$	$0.009 \angle 180^\circ$	$0.006 \angle 180^\circ$

	Spiral	
	Uncontrolled	Controlled
	$+0.225$	-1.083 ± 0.862
β	$0.007 \angle 0^\circ$	$0.029 \angle 119^\circ$
p	$0.024 \angle 0^\circ$	$1.000 \angle 0^\circ$
r	$0.280 \angle 0^\circ$	$0.169 \angle -124^\circ$
ϕ	$0.556 \angle 0^\circ$	$0.721 \angle -141^\circ$
θ	$1.000 \angle 0^\circ$	$0.074 \angle 106^\circ$

Table 3.3: Lateral Modes

effect of the tail section. As before the fast pole is damped to prevent large angular rates.

The Roll Mode describes an aircraft’s reaction to a roll moment disturbance. The difference in lift force acting on the wings cause the aircraft to enter a constant roll. The pole is already in a desirable location, but is made slightly faster.

The Spiral Mode is a slow, unstable mode that describes the tendency of the aircraft to enter a turn if the wings are perturbed from the normal level state. This effect is usually counteracted by dihedral and swept back wing, both of which are absent in this aircraft. Through feedback the pole is combined with a controller pole and moved to the left half plane with close to critical damping.

3.4.3 Gain Settling

The stability of LQR is only guaranteed when the feedback gains have sufficiently stabilized [13]. Since a limit is placed on the possible iterations it is very important to ensure that enough steps are used. Several problems were encountered in this aspect. Some feedback gains take thousands of time steps to reach steady state which is an order of magnitude more than can be computed on the OBC.

To determine the number of iterations needed the feedback gains are plotted at every time step of the iterative Riccati solution. A few calculations were performed on the gain matrices to aid in the interpretation of the graphs. The feedback matrix was multiplied with scaling matrices to determine the relative sizes of feedback elements e.g. A feedback of 1 from an angle error to thrust is very small, while a feedback of 1 from position error to a flap deflection is huge. All feedback elements smaller than 1% of the average elements size of this weighted feedback matrix are considered unimportant and omitted. LQR uses all the couplings between states and sometimes these couplings are so weak that it cannot realistically be used in a controller e.g. A phugoid mode can be controlled by thrust, but an elevator is much more effective. For the purpose of investigating the dominant feedback these lesser feedbacks are omitted. Only the feedback gains to conventional actuators (Thrust, elevator, aileron and rudder) are shown as the additional feedback gains settle in a similar fashion.

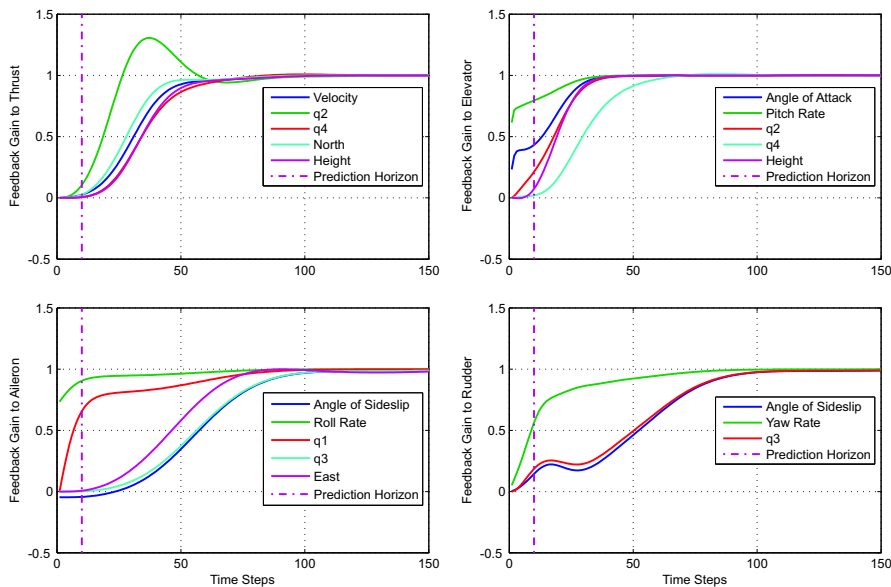


Figure 3.18: Gain Settling - Previous Project [5]

Figure 3.18 shows the gain settling from [5]. The simulations were done with the latest aircraft parameters and weightings available. It is seen that the prediction horizon of ten time steps is too short for most significant gains to stabilize. This is a possible explanation for why the final weightings from [5] are substantially larger than the final weightings used in this project.

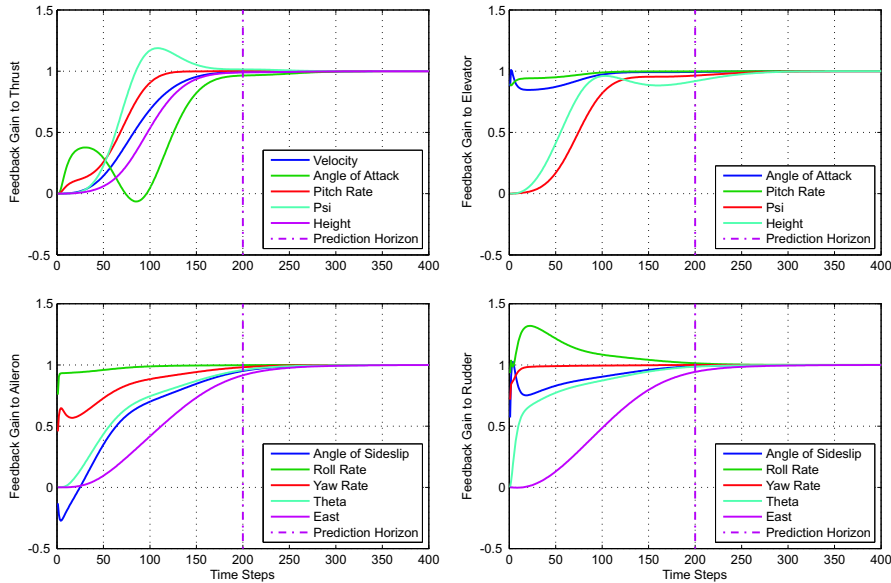


Figure 3.19: Gain Settling - Level Flight

From figure 3.19 it is clear that all significant gains have stabilized after 200 time steps. If fewer time steps are used stability cannot be guaranteed. The question now arises as to why the gain settling in this project and [5] take considerably longer than in other cases [17]. Some observations have been made in this regard.

- At certain orientations the system matrix is *badly scaled* (or close to singular) as there are several states that are difficult or impossible to control (as described in section 2.2.1). When this occurs the gains associated with the uncontrollable states take very long to stabilize.
- It has been seen that the speed at which gains settle are comparable to the time response of the individual states of the system. When investigating gain settling it was found that the feedback gains from rates stabilize significantly faster than angle feedback. Angle feedback gains in turn stabilize faster than position feedback gains.
- Changing weightings can significantly impact the stabilization of gains. This presents two possible dangers:

- A weighting is increased to make a gain stabilize faster. In doing so the system is made too aggressive and non-linear behaviour is induced.
- A weighting is decreased to make the system less aggressive, but this causes a feedback gain to not stabilize in the given horizon. An incorrect gain will be used resulting in an unexpected response.

It is therefore advisable that both the gain settling and step response be investigated when a weighting is changed.

- The time complexity of matrix inversions are $O(n^3)$ where n is the number of states. When the system matrix is decoupled the matrix inversions will be performed faster and more time steps can be calculated.

3.5 Results

Even with the insight gained through these investigations the ultimate selection of weightings is still an iterative, sometimes hit-and-miss, process. To ensure a specific choice of weightings deliver a viable system a list of requirements is created. When changing a weighting for any reason all the requirements *must* be checked.

1. All significant gains should stabilize in 200 time steps to guarantee stability. This implies that all poles will be placed in the left half plane.
2. The system must be stable when subjected to a significant step error. This ensures stability when experiencing wind gust disturbances.
3. The aircraft must be able to fly the trajectories consecutively.
4. Noise rejection must be as good as possible, while satisfying all above criteria. This is equivalent to requiring a high bandwidth.

If a system fails to comply with one of these requirements it cannot be used, even if performance in a different area is excellent. The final weightings given in table 3.4.

Some reasoning is given here to explain the weighting selections, noteworthy weightings have been highlighted.

- The angle of attack is weighed heavily during straight flight, but very lightly during a transition. This is due to the inaccurate angle of attack trajectory.
- The thrust state is weighed lightly. When the control system commands thrust the thrust actuation changes first and then the thrust state changes. Effectively the thrust state and the thrust actuation represent

Vertical				Level			
V_{int}	1.00	ψ	0.50	V_{int}	1.00	ψ	1.00
V	0.50	N	0.05	V	0.50	N	0.10
α	0.05	E	0.05	α	0.50	E	0.05
β	0.05	h	0.10	β	0.50	h	0.05
p	1.00	T_s	10.0	p	0.50	T_s	10.0
q	1.00	T_c	10.0	q	0.50	T_c	1.0
r	2.00	δ_e, δ_a	0.50	r	0.50	$\delta_e, \delta_a, \delta_r$	0.30
ϕ	0.50	δ_r	1.0	ϕ	1.00	$\delta_{fc}, \delta fd, \delta v$	1.00
θ	0.50	$\delta_{fc}, \delta fd, \delta v$	0.50	θ	0.05	R	75.0
		R	100				
Tran1				Tran2			
V_{int}	1.00	ψ	2.00	V_{int}	1.00	ψ	1.00
V	0.50	N	0.10	V	0.50	N	0.05
α	5.00	E	0.10	α	5.00	E	0.05
β	0.50	h	0.10	β	0.50	h	0.05
p	1.00	T_s	10.0	p	1.00	T_s	10.0
q	1.00	T_c	1.00	q	1.00	T_c	1.00
r	1.00	$\delta_e, \delta_a, \delta_r$	0.30	r	1.00	$\delta_e, \delta_a, \delta_r$	0.30
ϕ	2.00	δ_{fc}	0.1	ϕ	2.00	$\delta_{fc}, \delta fd, \delta v$	0.30
θ	2.00	$\delta fd, \delta v$	0.30	θ	2.00	R	75.0
		R	75.0				

Table 3.4: Final Max Deviations

the same quantity. Weighing both could result in the system not commanding enough thrust to effectively correct velocity and position errors.

- The direction in which the aircraft is travelling is weighted lightly. To control forward displacement is very difficult as the velocity needs to be perturbed for a substantial length of time. With an integrator on forward velocity this is not practically possible. During transitions all position states are weighed heavily.
- The weightings during transitions are not substantially more aggressive than during normal flight. Very aggressive weightings will result in better trajectory tracking, but can cause problems when switching from one trajectory to the next. Suddenly switching to high gains will create large actuations and excite high frequency modes.
- During vertical flight the actuation weighting is increased to lower the bandwidth of the system.

3.5.1 Trajectory Tracking

The ultimate performance of the system is judged on whether it is able to regulate the craft along the proposed trajectory. Figures 3.20 to 3.22 show the trajectory tracking of the LQR system. The aircraft starts in vertical flight and transitions to level flight. After ten seconds of level flight the aircraft transitions back to vertical flight.

- The velocity response is very good. The integrator placed on velocity ensures a zero steady state error.
- The angle of attack trajectory is very inaccurate, as expected.
- Larger errors are present during vertical flight. This is caused by the decreased effectiveness of the actuators and the reduced natural stability caused by less airflow over the aircraft. In addition, angle of attack and angle of side slip are more difficult to estimate at low forward speed.
- The position errors are quite large. When a trajectory is calculated the current aircraft position is used as a starting point. This *resets* the position errors to zero at the start of each new trajectory.
- Euler switching works well as there is no perceptible impact on the system
- The calculation of accurate trajectories are clearly the limiting factor in this system's performance.

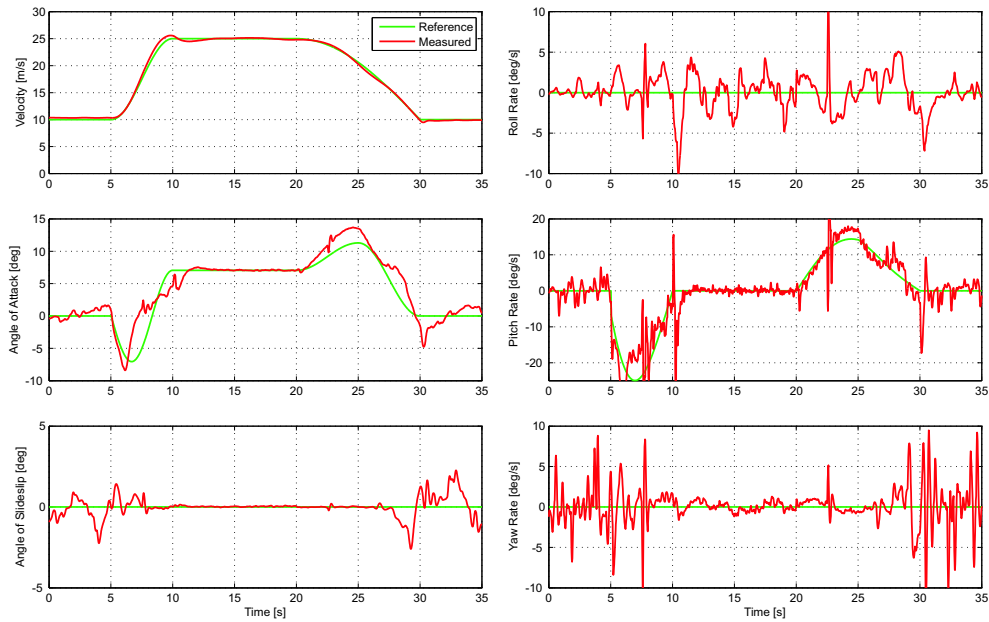


Figure 3.20: Trajectory Tracking of the LQR system

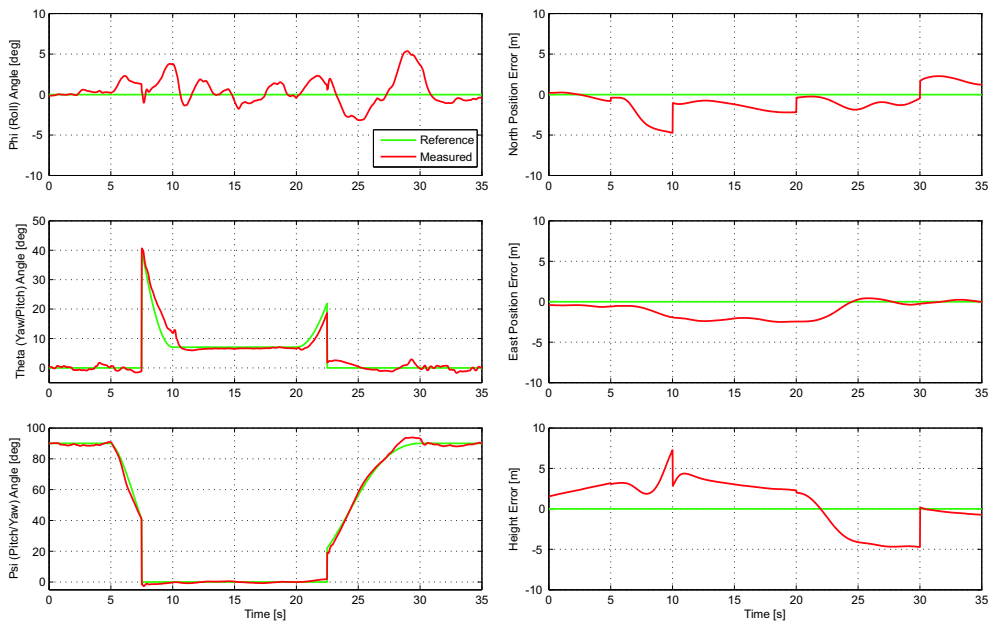


Figure 3.21: Trajectory Tracking of the LQR system

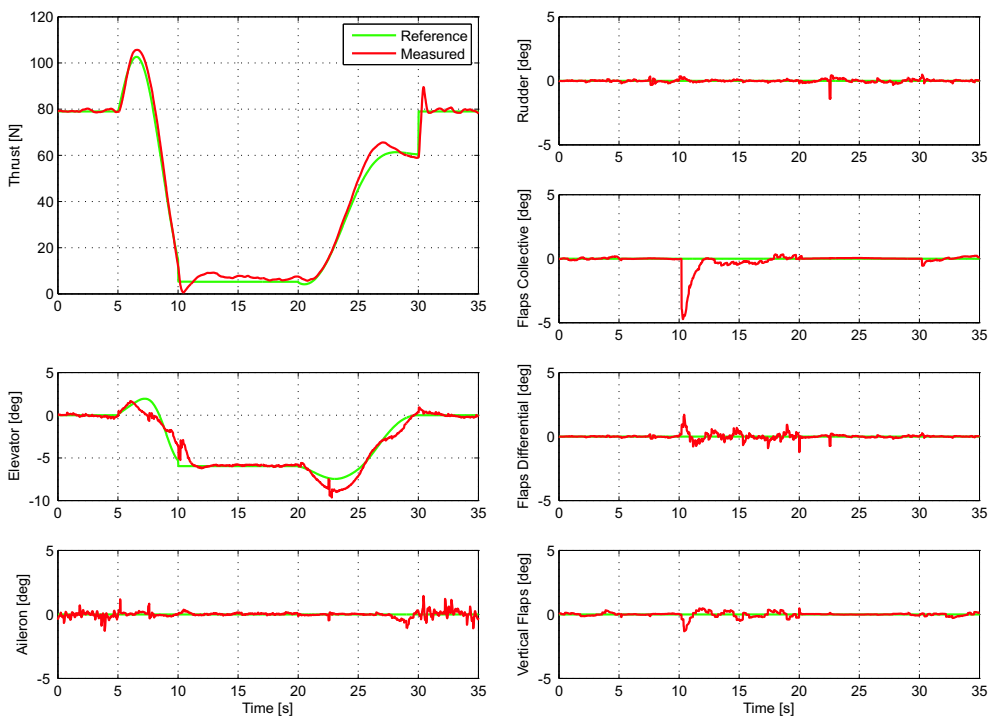


Figure 3.22: Trajectory Tracking of the LQR system

Chapter 4

Time-Scale Decoupled

4.1 Introduction

As stated before, this controller relies on the natural timescale separation between the rigid body and point mass dynamics of an aircraft as set out in [2]. The two models have already been derived in Chapter 2. The time scale separation enables any point mass states present in the rigid body equations to be treated as constant parameters resulting in an inner loop controller that is totally attitude independent. Additionally, all aircraft parameters are contained in the inner loop, so the guidance dynamics are aircraft independent.

When designing these controllers the quantities that are available for feedback must be considered. Low cost sensors are available for acceleration, angular rates and airspeed. These quantities can be fed back without the need for a full state estimator.

4.1.1 Inner Loop - Specific Acceleration

The inner loop must control the rigid body dynamics and track commanded wind axes accelerations. Forces acting on the aircraft are divided into *specific* forces and gravity. Specific forces include all forces originating from actuation and the form and motion of the aircraft, primarily aerodynamic forces and thrust. Specific forces are orientation independent when coordinated in wind axis system. This aids in decoupling the point mass dynamics. Furthermore, wind axes accelerations are easily measured using the on-board accelerometers and gyrometers. To make the dynamics truly attitude independent the effect of gravity needs to be removed. A well defined model exists for how gravity couples into the rigid body dynamics, enabling cancellation using feedback linearisation. Feedback linearisation is used extensively throughout this control scheme and is discussed in more depth in appendix B.

The rigid body dynamics are decoupled into several controllers: axial specific acceleration, normal specific acceleration, lateral specific acceleration, roll rate and directional controller [2].

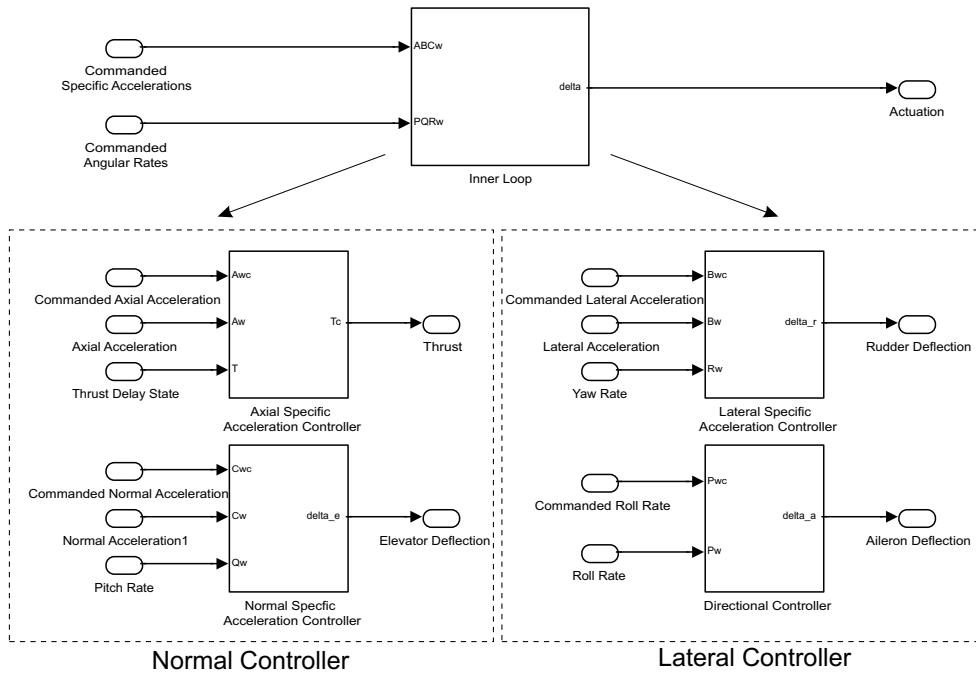


Figure 4.1: Inner Loop Block Diagram

4.1.2 Outer Loop - Guidance Controller

The outer loop controls the point mass dynamics. All the aircraft specific variables are encapsulated in the high bandwidth inner loop. The outer loop controller is thus aircraft independent, greatly simplifying the creation of a trajectory since any kinematically viable trajectory can be used. The guidance controller commands accelerations to the inner loop to remain on the chosen path.

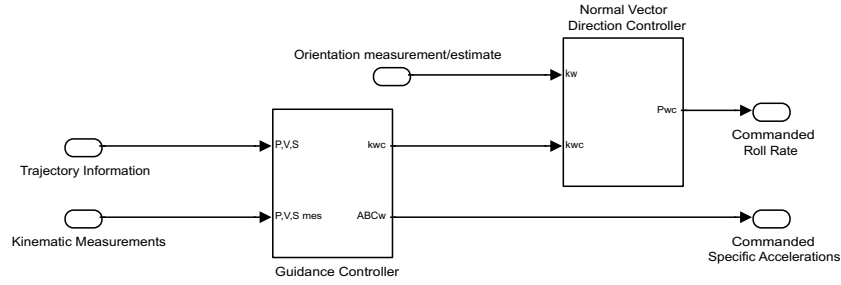


Figure 4.2: Outer Loop Block Diagram

4.2 Inner Loop Design

4.2.1 Axial Specific Acceleration Controller

From section 2.5.4, the model for the axial acceleration can be written as.

$$\dot{T} = \left[-\frac{1}{\tau_T} \right] T + \left[\frac{1}{\tau_T} \right] T_c \quad (4.2.1)$$

$$A_w = \left[\frac{1}{m} \right] T + \left[-\frac{\rho \bar{V}^2 S}{m} C_D \right] \quad (4.2.2)$$

The assumption has been made that the angle of attack is small enough that the thrust can be approximated as a wind axes acceleration. The drag model is not very accurate, so instead of compensating for it directly, integral control will be used. A high bandwidth will ensure that the integral control quickly compensates for any drag error.

$$\begin{bmatrix} \dot{T} \\ \dot{E}_A \end{bmatrix} = \begin{bmatrix} -\frac{1}{\tau_T} & 0 \\ \frac{1}{m} & 0 \end{bmatrix} \begin{bmatrix} T \\ E_A \end{bmatrix} + \begin{bmatrix} \frac{1}{\tau_T} \\ 0 \end{bmatrix} T_c \quad (4.2.3)$$

$$\begin{bmatrix} A_w \\ E_A \end{bmatrix} = \begin{bmatrix} \frac{1}{m} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} T \\ E_A \end{bmatrix} + \begin{bmatrix} -\frac{\rho \bar{V}^2 S}{m} C_D \\ 0 \end{bmatrix} \quad (4.2.4)$$

The following control law is selected:

$$T_c = - \begin{bmatrix} K_A & K_E \end{bmatrix} \begin{bmatrix} A_w \\ E_A \end{bmatrix} = - \begin{bmatrix} K_A & K_E \end{bmatrix} \begin{bmatrix} \frac{1}{m} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} T \\ E_A \end{bmatrix} \quad (4.2.5)$$

The closed loop dynamics can now be calculated: $A_{cl} = A - BKC$.

$$\begin{aligned} A_{cl} &= \begin{bmatrix} -\frac{1}{\tau_T} & 0 \\ \frac{1}{m} & 0 \end{bmatrix} - \begin{bmatrix} \frac{1}{\tau_T} \\ 0 \end{bmatrix} \begin{bmatrix} K_A & K_E \end{bmatrix} \begin{bmatrix} \frac{1}{m} & 0 \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} -\frac{1}{\tau_T}(1 + K_A) & -\frac{1}{\tau_T}K_E \\ \frac{1}{m} & 0 \end{bmatrix} \end{aligned} \quad (4.2.6)$$

The characteristic equation of a system is given by: $\det(sI - A)$.

$$\begin{vmatrix} s + \frac{1}{\tau_T}(1 + \frac{K_A}{m}) & \frac{1}{\tau}K_E \\ -\frac{1}{m} & s \end{vmatrix} = s^2 + \frac{1}{\tau_T}(1 + \frac{K_A}{m})s + \frac{K_E}{\tau_T m} \quad (4.2.7)$$

Given the desired characteristic equation, $\alpha_c(S) = s^2 + \alpha_1 s + \alpha_0$, the gains are solved.

$$K_A = m(\tau_t \alpha_1 - 1) \quad \text{and} \quad K_E = m \tau_t \alpha_0 \quad (4.2.8)$$

All other controllers follow a similar method of derivation and will not be discussed in detail again. For more detail on how each individual controller is derived refer to [2].

Integrator Anti-Windup Over the course of the trajectories flown in this project there are some occasions where the commanded specific acceleration will change significantly over a short time. Coupled with the delay in the thrust, this can cause the integrators to *wind up* and the actuator to saturate by either commanding a too high or negative thrust. The resultant non-linear response quickly causes instability in the system.

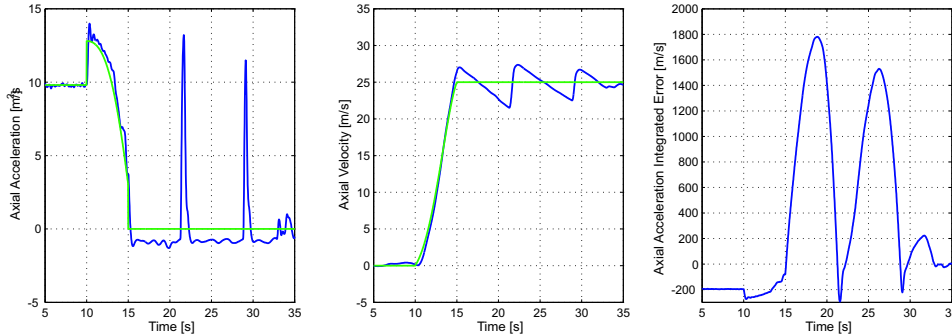


Figure 4.3: Integrator Windup can cause instability

This can be prevented by adding integrator anti-windup to the system. When the thrust command approaches saturation the error is decreased to prevent the integrator from winding up.

The tracking time constant, K_A , is used to specify the amount of anti-windup used. A suggested rule-of-thumb value for K_A in a PID controller is given in [11]: $K_D \leq K_A \leq K_I$. Throughout the project the size of K_I has been found to be in the order of 100 and K_D is zero since no differential control is used. In accordance to the rule-of-thumb K_A was chosen as 50.

From simulation results it is clear that the actuator still saturates, but prevents the system from going unstable.

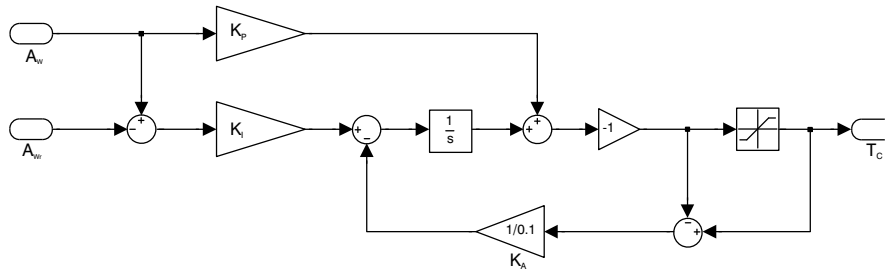


Figure 4.4: Axial specific acceleration controller with Anti-Windup

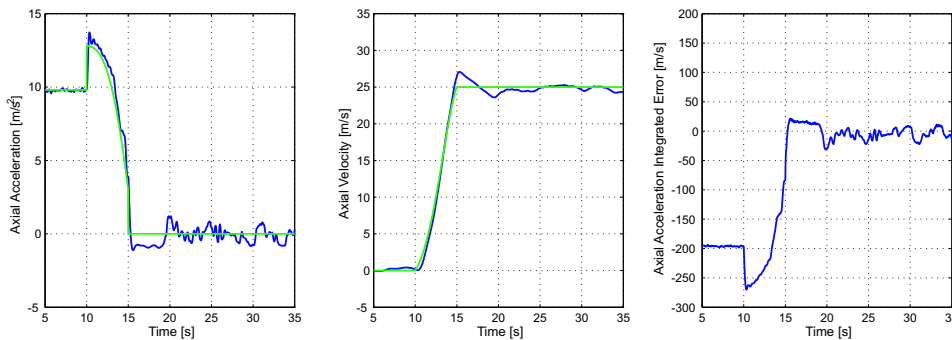


Figure 4.5: Integrator Anti-Windup prevents system from going unstable

4.2.2 Normal Specific Acceleration Controller

From section 2.5.4 the model for normal acceleration can be written as:

$$\begin{bmatrix} \dot{\alpha} \\ \dot{Q} \end{bmatrix} = \begin{bmatrix} a_{11} & 1 \\ a_{21} & a_{22} \end{bmatrix} + \begin{bmatrix} 0 \\ b_2 \end{bmatrix} \delta_E + \begin{bmatrix} d_1 + \frac{g_1}{V} \\ d_2 \end{bmatrix} \quad (4.2.9)$$

$$C_W = \begin{bmatrix} \bar{V}a_{11} & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ Q \end{bmatrix} + [0] \delta_e + [\bar{V}d] \quad (4.2.10)$$

Several factors can induce an error in the normal acceleration, most notable an inaccurate $C_{L\alpha}$ and steady state velocity error. To correct this, integral control is placed on normal acceleration. To make the inner controller truly orientation independent the gravity term needs to be compensated for or cancelled. This can be achieved in a variety of ways. The bandwidth of the controller can be increased to compensate quickly enough, but this is not viable as it will require a too high bandwidth. The gravity term will be cancelled using feedback linearisation as shown in appendix B.

In order the use feedback linearisation the normal force due to pitch rate and elevator deflection will have to be ignored, removing non-minimum phase from the system. In a normal aircraft this is a valid assumption, as the elevator actuation force is quite small, but acting over a long moment

arm. The shorter the moment arm is the closer the non-minimum phase zero moves to the origin. The location of this zero places a limit on the bandwidth of the system, a problem considering the pole location must be chosen to enforce time scale separation. A rule of thumb suggest that no pole can be faster than one third of the NMP-zero's frequency [2].

In order to reduce the effect of non-minimum phase, the zero is moved to a higher frequency by using a combination of actuators. The collective paddles have a very short moment arm and are primarily used to provide forces, not moments. The paddles are thus used to cancel the force created by the elevator, while the difference in moment arms ensures that a sizeable moment is still created. The transfer function for elevator deflection to normal acceleration is calculated.

$$\frac{C_W}{\delta_e} = -5.13 \frac{(s - 11.5)(s + 18.0)}{s^2 + 8.19s + 85.9} \quad (4.2.11)$$

The ratio between the lift coefficients will determine the mixing of the two actuators. The new actuation's aerodynamic coefficients can be found by adding the coefficients of elevator and flaps in the same ratio.

$$r = -1.1 \frac{C_{L\delta_m}}{C_{L\delta_{fc}}} = -1.833 \quad (4.2.12)$$

$$C_{L\delta_m} = C_{L\delta_e} + rC_{L\delta_{fc}} \quad (4.2.13)$$

The ratio is chosen 10% larger than is strictly necessary. This is done to ensure that modelling errors do not cause unexpected non-minimum phase.

Coefficient	Elevator	Flaps Collective	Mixed
Lift	1.246	0.748	-0.1246
Pitch Moment	-1.809	-0.083	-1.657

Table 4.1: Mixed Normal Actuator Aerodynamic Coefficients

$$\frac{C_W}{\delta_m} = 0.513 \frac{s^2 + 6.47s + 2654}{s^2 + 8.19s + 85.9} \quad (4.2.14)$$

By calculating the new transfer function, from mixed actuation to normal acceleration, it is clear that the non-minimum phase zeros have been moved into the left half plane.

From [2] the control law is chosen as:

$$\delta_m = -K_Q Q - K_C C_W - K_E E_C + \delta_{mg} \quad (4.2.15)$$

for the desired characteristic equation $\alpha_c(s) = s^3 + \alpha_2 s^2 + \alpha_1 s + \alpha_0$ the gains can be calculated as follows.

$$\begin{aligned} \dot{E}_C &= C_W - C_{W_r} & K_Q &= \frac{1}{b_2}(\alpha_2 + a_{11} + a_{22}) \\ K_C &= \frac{1}{\bar{V}a_{11}b_2}(\alpha_1 + a_{21} + a_{11}\alpha_2 + a_{11}^2) & K_E &= \frac{1}{\bar{V}a_{11}b_2}\alpha_0 \\ \delta_{mg} &= -\frac{1}{\bar{V}b_2}[(\alpha_2 + a_{11})g_1 + \dot{g}_1] \end{aligned} \quad (4.2.16)$$

with

$$\begin{aligned} a_{11} &= -\frac{qS}{m\bar{V}}C_{L_\alpha} & a_{21} &= \frac{qS\bar{c}}{I_{yy}}C_{m_\alpha} & a_{22} &= \frac{qS\bar{c}}{I_{yy}}\frac{\bar{c}}{2\bar{V}}C_{m_Q} \\ b_2 &= \frac{qS\bar{c}}{I_{yy}}C_{m_{\delta_E}} & g_1 &= e_{33}^{WI}g & \dot{g}_1 &= -g \left[\left(\frac{C_W + e_{33}^{WI}g}{\bar{V}} \right) e_{13}^{WI} + P_W e_{23}^{WI} \right] \end{aligned} \quad (4.2.17)$$

4.2.3 Lateral Specific Acceleration Controller

The lateral controller must control the roll and directional dynamics of the system. A choice has to be made over whether, and how, the dynamics will be decoupled. By ignoring the coupling between the two systems modelling errors will be introduced. Luckily these errors will be minimized by feedback. The effect can be compensated for by using feedback linearisation whenever possible. Feedback linearisation should be used with caution as it creates sensitivity to aircraft parameters. This sensitivity causes poles to move from their selected positions when aircraft parameters are changed slightly. This effect will be examined in Chapter 5.

The lateral dynamics are given by:

$$\begin{aligned} \begin{bmatrix} \dot{\beta} \\ \dot{P} \\ \dot{R} \end{bmatrix} &= \begin{bmatrix} a_{11} & 0 & -1 \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} \beta \\ P \\ R \end{bmatrix} \\ &+ \begin{bmatrix} 0 & 0 \\ b_{21} & 0 \\ b_{31} & b_{32} \end{bmatrix} \begin{bmatrix} \delta_A \\ \delta_R \end{bmatrix} + \begin{bmatrix} \frac{c_l}{\bar{V}} \\ 0 \\ 0 \end{bmatrix} \end{aligned} \quad (4.2.18)$$

$$B_W = [\bar{V}a_{11}]\beta \quad (4.2.19)$$

$$P_W = P \quad (4.2.20)$$

It has been assumed that side force due to roll rate, yaw rate and aileron deflection as well as the roll moment due to rudder deflection is negligible. Ignoring the side forces due to roll rate and aileron deflection is common practice. The roll rate due to rudder deflection is very small on this aircraft, because of the symmetrical tail section. The non-minimum phase created by the side force due to rudder deflection will be compensated for in a similar way as the lift generated by the elevators.

Coefficient	Rudder	Vertical Paddles	Mixed
Side Force	1.008	0.312	-0.101
Yaw Moment	-0.389	-0.017	-0.327

Table 4.2: Mixed Lateral Actuator Aerodynamic Coefficients

Roll Rate Controller The roll rate controller follows a commanded roll rate with the ultimate goal of positioning the lift vector. From equation (4.2.18) the roll dynamics are

$$\dot{P} = a_{21}\beta + a_{22}P + a_{23}R + b_{21}\delta_A \quad (4.2.21)$$

The control law is created with integral control on the roll rate to counter any asymmetry in the aircraft's construction.

$$\delta_A = -K_P P - K_E E_P + \delta_{A_f} \quad \text{with} \quad \dot{E}_P = P - P_R \quad (4.2.22)$$

The resulting dynamics are:

$$\begin{aligned} \dot{P} &= a_{21}\beta + a_{22}P + a_{23}R + b_{21}(-K_P P - K_E E_P + \delta_{A_f}) \\ &= [a_{22} - K_P b_{21}]P + [-K_E b_{21}]E_P + [a_{21}]\beta + [a_{23}R + [b_{21}]\delta_{A_f}] \\ &= [a_{22} - K_P b_{21}]P + [-K_E b_{21}]E_P \end{aligned} \quad (4.2.23)$$

$$\text{if} \quad [a_{21}]\beta + [a_{23}]R + [b_{21}]\delta_{A_f} = 0 \quad (4.2.24)$$

The feedback linearisation term, δ_{A_f} is calculated so that equation (4.2.24) holds.

$$\begin{aligned} \delta_{A_f} &= -\frac{1}{b_{21}} ([a_{21}\beta + [a_{23}]R]) \\ &= -\frac{1}{b_{21}} \left(\left[\frac{a_{21}}{\sqrt{V}a_{11}} \right] B_W + [a_{23}]R \right) \end{aligned} \quad (4.2.25)$$

For the desired characteristic equation $\alpha_C(s) = s^2\alpha_1s + \alpha_0$ the gains can be calculated as.

$$\begin{aligned} K_P &= \frac{\alpha_1 + a_{22}}{b_{21}} & K_E &= \frac{\alpha_0}{b_{21}} \\ a_{11} &= \frac{qS}{mV} C_{y\beta} & a_{21} &= \frac{qSb}{\Delta} (I_{zz}C_{l\beta} + I_{xz}C_{n\beta}) \\ a_{22} &= \frac{qSb}{\Delta} \frac{b}{2V} (I_{zz}C_{l_p} + I_{xz}C_{n_p}) & a_{23} &= \frac{qSb}{\Delta} \frac{b}{2V} (I_{zz}C_{l_R} + I_{xz}C_{n_R}) \\ b_{21} &= \frac{qSb}{\Delta} (I_{zz}C_{l_{\delta_A}} + I_{xz}C_{n_{\delta_A}}) & \Delta &= I_{xx}I_{zz} - I_{xz}^2 \end{aligned} \quad (4.2.26)$$

Directional Controller The directional controller regulates lateral acceleration by changing the angle of side slip. During normal forward flight the lateral acceleration will be regulated to zero to ensure that the aircraft always

flies in a coordinated turn. The decoupled directional dynamics are described by

$$B_W = [\bar{V}a_{11}]\beta \quad \text{with} \quad \dot{\beta} = \frac{1}{\bar{V}}B_W - R + \frac{1}{\bar{V}}c_1 \quad (4.2.27)$$

For the control law $\delta_R = -K_R R - K_B B_W - K_E E_B + \delta_{R_f}$ the system is expanded by differentiation so that feedback linearisation can be applied. Integral control is used to correct steady state errors.

$$\begin{aligned} \ddot{B}_W = & [a_{11} + a_{33} - K_R b_{32}]\dot{B}_W + [-a_{31} - a_{33}a_{11} + K_B \bar{V}a_{11}b_{32} + K_R b_{32}a_{11}]B_W \\ & + [K_E \bar{V}a_{11}b_{32}]E_B + [-\bar{V}a_{11}a_{32}]P + [-\bar{V}a_{11}a_{32}]\delta_A \\ & + [K_R b_{32}a_{11} - a_{33}a_{11}]c_1 + [a_{11}]\dot{c}_1 + [-\bar{V}a_{11}b_{32}]\delta_{R_f} \end{aligned} \quad (4.2.28)$$

Resulting in the feedback linearisation control:

$$\delta_{R_f} = \frac{1}{\bar{V}b_{32}} [(\alpha_2 + a_{11}c_1 + \dot{c}_1) - \frac{1}{b_{32}} ([a_{32}]P + [b_{31}]\delta_A)] \quad (4.2.29)$$

and system dynamics:

$$\begin{aligned} \begin{bmatrix} \ddot{B}_W \\ \dot{B}_W \\ \dot{E}_B \end{bmatrix} = & \begin{bmatrix} (a_{11} + a_{33} - K_R b_{32}) & (-a_{31} + K_B \bar{V}a_{11}b_{32} + K_R b_{32}a_{11} - a_{33}a_{11}) & (K_E \bar{V}a_{11}b_{32}) \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{B}_W \\ B_W \\ E_B \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} B_{WR} \end{aligned} \quad (4.2.30)$$

For the desired characteristic equation $\alpha_B(s) = s^3 + \alpha_2 s^2 + \alpha_1 s + \alpha_0$ the required feedback gains can be calculated as:

$$\begin{aligned} K_R &= \frac{1}{b_{32}}(\alpha_2 + a_{11} + a_{33}) & K_B &= -\frac{1}{\bar{V}a_{11}b_{32}}(\alpha_1 + a_{11}\alpha_2 + a_{11}^2 - a_{31}) \\ K_E &= -\frac{1}{\bar{V}a_{11}b_{32}}\alpha_0 & \dot{c}_1 &= -g \left[\left(\frac{B_W + \dot{c}_{23}^{WI}}{\bar{V}} \right) \dot{e}_{13}^{WI} - P_W \dot{e}_{33}^{WI} \right] \\ a_{31} &= \frac{qSb}{\Delta} (I_{xx}C_{n_\beta} + I_{xz}C_{l_\beta}) & a_{32} &= \frac{qSb}{\Delta} \frac{b}{2\bar{V}} (I_{xx}C_{n_p} + I_{xz}C_{l_p}) \\ a_{33} &= \frac{qSb}{\Delta} \frac{b}{2\bar{V}} (I_{xx}C_{n_r} + I_{xz}C_{l_r}) & b_{31} &= \frac{qSb}{\Delta} (I_{xx}C_{n_{\delta_A}} + I_{xz}C_{l_{\delta_A}}) \\ b_{32} &= \frac{qSb}{\Delta} (I_{xx}C_{n_{\delta_R}} + I_{xz}C_{l_{\delta_R}}) \end{aligned} \quad (4.2.31)$$

4.3 Outer Loop Design

With the inner loop closed, the aircraft can now be viewed as a point mass reacting to acceleration commands. The outer loop controller commands these accelerations to follow a prescribed trajectory. The outer loop is decoupled into a Normal Vector Direction and Guidance controller. Note that throughout the outer loop design commanded rates and accelerations are assumed

to occur *immediately*. In other words the commanded rate (actuation) and the real rate measured by the sensors are treated as the same quantity. This is a valid assumption as long as the time scale separation is maintained.

4.3.1 Normal Specific Acceleration Vector Direction Controller

Controlling an aircraft's lateral movement using a rudder is not effective and places strain in the structure. It is more effective to roll the aircraft and use the sideways component of the lift vector to provide an *initially orientated lateral* force to fly in a coordinated turn. The commanded specific acceleration thus always lies in the XZ-plane. This controller commands roll rates to the inner loop to control the direction of the normal specific acceleration vector.

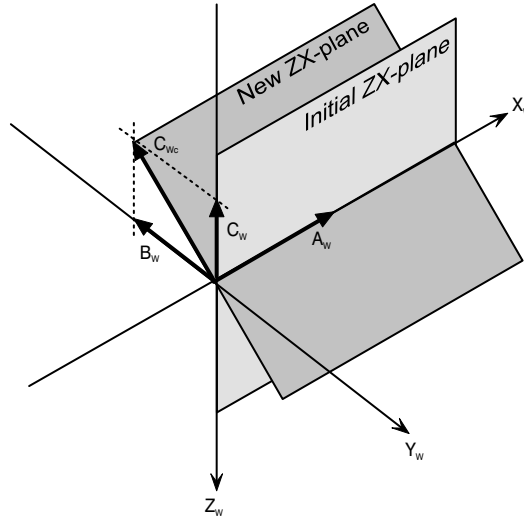


Figure 4.6: Aircraft rolls so that commanded acceleration remains in the ZX-plane

Given the wind axes roll rate P_W and the commanded roll rate P_{W_C} (pre-defined roll rate from the trajectory) the dynamics of the error angle, ϕ , can be described as:

$$\dot{\phi} = P_{W_C} - P_W \quad (4.3.1)$$

Ignoring the inner loop dynamics (because of time scale separation) the control law $P_W = P_{W_C} + K_\phi \phi$ is used to regulate the error angle to zero over time. The feed forward term, P_{W_C} follows the trajectory while the feedback corrects error arising from disturbances. The closed loop dynamics can now be calculated,

$$\dot{\phi} = P_{W_C} - (P_{W_C} - K_\phi \phi) \Rightarrow \dot{\phi} + K_\phi \phi = 0 \quad (4.3.2)$$

with a closed loop pole at $s = -K_\phi$.

To calculate the necessary control, the error angle must be found. To keep the controller as general as possible the angle will be expressed in terms of unit vectors that can be found from the DCM, allowing any attitude specification to be used.

$$\phi = \arccos(\mathbf{k}^{W_C} \cdot \mathbf{k}^W) \quad (4.3.3)$$

The range of arccos is defined as $[0, \pi)$, resulting in the loss of sign information. The sign is regained by investigating:

$$\mathbf{k}^{W_C} \cdot \mathbf{j}^W = \cos(\pi/2 + \phi) = -\sin(\phi) \quad (4.3.4)$$

Thus for the range $(-\pi, \pi)$ the sign can be regained and the control law written in terms of unit vectors.

$$\phi = -\text{sgn}(\mathbf{k}^{W_C} \cdot \mathbf{j}^W) \cdot \arccos(\mathbf{k}^{W_C} \cdot \mathbf{k}^W) \quad (4.3.5)$$

$$P_W = P_{W_C} - K_\phi [\text{sgn}(\mathbf{k}^{W_C} \cdot \mathbf{j}^W) \cdot \arccos(\mathbf{k}^{W_C} \cdot \mathbf{k}^W)] \quad (4.3.6)$$

Combined Roll and Normal Vector Direction Controller When the dynamics of the complete system are calculated the poles do not lie exactly where they were placed. This is caused by the various assumptions made. Luckily, in most cases the errors are small as the conditions for assumptions (such as time scale separation) are easily met. In the case of the NVD and Roll controllers, it is not possible to place the poles far enough apart. The NVD controller must be placed faster than the guidance dynamics, but slower than the rotational dynamics.

To improve the result of pole placement the Roll controller is designed in conjunction with the NVD controller. The system is still relatively easy to understand and will result in more accurate pole placement.

$$\begin{bmatrix} \dot{P} \\ \dot{E}_P \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} a_{22} & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} P \\ E_P \\ \phi \end{bmatrix} + \begin{bmatrix} b_{21} & 0 \\ 0 & -1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_A \\ P_R \end{bmatrix} \quad (4.3.7)$$

When developing the decoupled NVD, it is assumed that the commanded roll rate occurs immediately, thereby ignoring the dynamics of the inner loop roll controller. In order to combine the dynamics of the two systems the control law will include feedback from the error angle, ϕ , to the roll rate reference, P_R , that drives the roll controller.

$$K = \begin{bmatrix} -K_P & -K_E & 0 \\ 0 & 0 & -K_\phi \end{bmatrix} \quad (4.3.8)$$

Closing the loop results in the following closed loop dynamics,

$$A_{CL} = A - BK = \begin{bmatrix} a_{22} - b_{21}K_P & -b_{21}K_E & 0 \\ 1 & 0 & K_\phi \\ -1 & 0 & 0 \end{bmatrix} \quad (4.3.9)$$

with characteristic equation

$$|sI - A_{CL}| = s^3 - (K_P b_{21} - a_{22})s^2 + K_E b_{21}s - K_E b_{21}K_\phi \quad (4.3.10)$$

For the desired CE $\alpha_P(s) = s^3 + \alpha_2 s^2 + \alpha_1 s + \alpha_0$ the gains can be calculated as:

$$K_P = \frac{\alpha_2 + a_{22}}{b_{21}} \quad K_E = \frac{\alpha_1}{b_{21}} \quad K_\phi = -\frac{\alpha_0}{K_E b_{21}} \quad (4.3.11)$$

4.3.2 Guidance Controller

With the normal vector direction controller closed it is now possible to command any specific acceleration in three dimensional space. The guidance controller receives kinematic vectors coordinated in the earth axes from the created trajectories and commands wind axes specific accelerations to the inner loop controllers to remain on said trajectory.

In order to apply feedback the position, velocity and acceleration is written as a the sum of a *reference* and *error* vector.

$$\mathbf{P}^{WE} = \mathbf{P}^{WR} + \mathbf{P}^{RE} \quad (4.3.12)$$

$$\mathbf{V}^{WE} = \mathbf{V}^{WR} + \mathbf{V}^{RE} \quad (4.3.13)$$

$$\mathbf{A}^{WE} = \mathbf{A}^{WR} + \mathbf{A}^{RE} \quad (4.3.14)$$

These quantities can be substituted into the equations of motion

$$\left. \frac{d}{dt} \mathbf{P}^{WR} + \mathbf{P}^{RE} \right|_E = \mathbf{V}^{WR} + \mathbf{V}^{RE} \quad (4.3.15)$$

$$\left. \frac{d}{dt} \mathbf{V}^{WR} + \mathbf{V}^{RE} \right|_E = \mathbf{A}^{WR} + \mathbf{A}^{RE} \quad (4.3.16)$$

Considering that the trajectory is kinematically feasible (introduces no errors) the reference and error dynamics can be separated.

$$\left. \frac{d}{dt} \mathbf{P}^{WR} \right|_E = \mathbf{V}^{WR} \quad (4.3.17)$$

$$\left. \frac{d}{dt} \mathbf{V}^{WR} \right|_E = \mathbf{A}^{WR} \quad (4.3.18)$$

The acceleration error is expanded into specific and gravitational acceleration. Note that the gravitational acceleration vector is time invariant and inertially fixed.

$$\mathbf{A}^{WR} = \mathbf{A}^{WE} - \mathbf{A}^{RE} \quad (4.3.19)$$

$$= \boldsymbol{\Sigma}^{WE} + \mathbf{G}^{WE} - \boldsymbol{\Sigma}^{RE} - \mathbf{G}^{RE} \quad (4.3.20)$$

$$= \boldsymbol{\Sigma}^{WE} - \boldsymbol{\Sigma}^{RE} \quad (4.3.21)$$

The error dynamics are coordinated in the earth axis system.

$$\dot{\mathbf{P}}_E^{WR} = \mathbf{V}_E^{WR} \quad (4.3.22)$$

$$\dot{\mathbf{V}}_E^{WR} = \boldsymbol{\Sigma}_E^{WE} - \boldsymbol{\Sigma}_E^{RE} \quad (4.3.23)$$

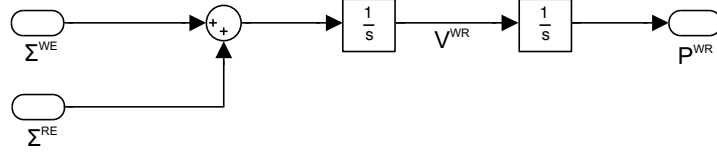


Figure 4.7: Open Loop Guidance Dynamics

The guidance controller will use only proportional feedback, since the inner loop integrators will compensate for any steady state errors. The control law, with feed forward term $\boldsymbol{\Sigma}_E^{RE}$ is:

$$\boldsymbol{\Sigma}_E^{WE} = -K_P \mathbf{P}_E^{WR} - K_V \mathbf{V}_E^{WR} + \boldsymbol{\Sigma}_E^{RE} \quad (4.3.24)$$

This results in the following closed loop dynamics:

$$\dot{\mathbf{P}}_E^{WR} = -K_P \mathbf{P}_E^{WR} - K_V \mathbf{V}_E^{WR} + \boldsymbol{\Sigma}_E^{RE} - \boldsymbol{\Sigma}_E^{RE} \quad (4.3.25)$$

$$\dot{\mathbf{P}}_E^{WR} + K_V \dot{\mathbf{P}}_E^{WR} + K_P \mathbf{P}_E^{WR} = \mathbf{0} \quad (4.3.26)$$

The gains K_V and K_P can be chosen to move the closed loop poles to the desired locations.

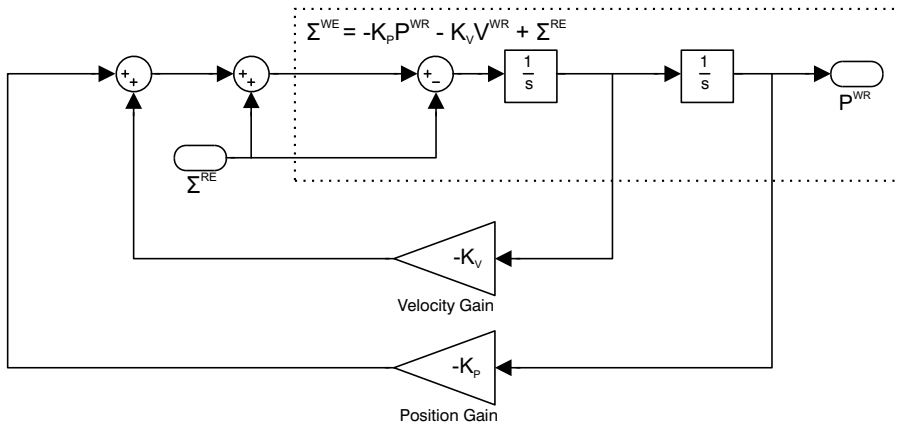


Figure 4.8: Closed Loop Guidance Controller

The calculated acceleration needs to be transformed to a form that can be commanded to the inner loop controllers, namely: An axial acceleration, normal acceleration and normal vector direction. Since the commanded acceleration has been defined as acting in the XZ-plane it only has components along the \mathbf{i}^W and \mathbf{k}^{Wc} unit vectors. \mathbf{k}^{Wc} represents the commanded normal vector. The components in each direction can be found as follows.

$$\Sigma^{Wc} = A_{Wc} \mathbf{i}^W + C_{Wc} \mathbf{k}^{Wc} \quad (4.3.27)$$

$$A_{Wc} = \Sigma^{Wc} \cdot \mathbf{i}^W \quad (4.3.28)$$

$$C_{Wc} = \pm |\mathbf{N}^{Wc}| \text{ with } \mathbf{N}^{Wc} = \Sigma^{Wc} - A_{Wc} \mathbf{i}^W \quad (4.3.29)$$

$$\mathbf{k}^{Wc} = \frac{1}{C_{Wc}} \mathbf{N}^{Wc} \quad (4.3.30)$$

The sign of the normal acceleration is found by investigating the orientation of the aircraft. During normal flight the normal acceleration acts in the negative Z-direction to counteract gravity, while inverted flight will result in a positive normal specific acceleration.

The guidance controller differs from previous controllers in that it operates on an error *vector*. The same feedback is applied to all three components of the error vector. This is not optimal as the aircraft does not behave the same in each plane. The slow lateral system limits the bandwidth of the guidance controller. This effect is illustrated in the step responses. Normal and lateral steps have comparable settling times. In contrast the normal step of the LQR controlled system has a faster settling time than its lateral step.

4.3.2.1 Zero Angle of Attack Problem

This method of determining the commanded accelerations is very effective over much of the flight envelope. The mathematics break down when the normal acceleration approaches zero. Unfortunately this is encountered in this project, mainly during vertical flight.

Intuitively the problem can be described as follows: To provide lateral acceleration the aircraft rolls slightly and uses a component of the lift vector to supply a *initially orientated lateral* force. The term *initially orientated lateral* refers to the lateral direction in the wind axes *before* the aircraft rolls. The initially orientated axis system is denoted with the subscript Wi . During vertical flight there is no steady state lift force. If this controller attempts to translate laterally the aircraft will roll to align the body normal vector with the desired acceleration. In the absence of a large lift force this angle can be anywhere in the $[-\pi, \pi]$ range as illustrated in figure 4.9.

It can be seen then that, in certain situations, a resultant lateral acceleration (specifically avoided by this controller) is actually required. In these cases there is no way to extract the desired normal vector from the calculated accelerations as in equations (4.3.27) to (4.3.30). No other options exist than to prescribe a normal vector manually as part of the trajectory.

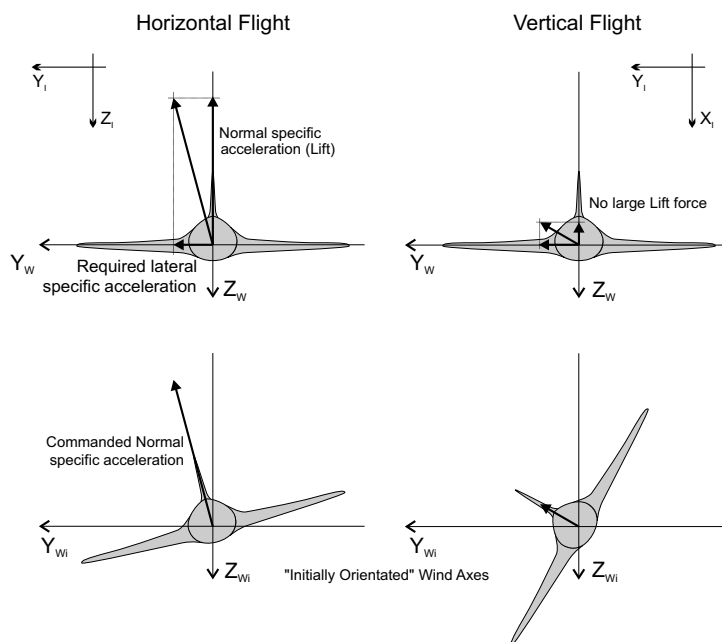


Figure 4.9: Small Normal Specific Acceleration problem

The next design choice is deciding where in the flight envelope to use what method of finding the normal vector. Several methods were experimented with, including a smooth changeover over the course of the transition. The final choice was forced by an anomaly in the normal acceleration trajectory as can be seen in figure 4.19. Throughout the transition to forward flight the normal acceleration remains small due to gravitational and centripetal acceleration canceling during the transition. The result is that no large normal specific acceleration is present during the transition and the method described in equations (4.3.27) to (4.3.30) is only used during forward flight. This is acceptable considering that large turns will only be executed during forward flight.

4.3.3 Trajectory Creation

Creating trajectories for the time scale decoupled system is substantially easier than for LQR. Since all aircraft specific parameters are encapsulated in the fast inner loop, the trajectory consists of only kinematic states. This is perfectly illustrated in forward flight where the system adopts the required angle of attack and steady state thrust to remain on the prescribed trajectory.

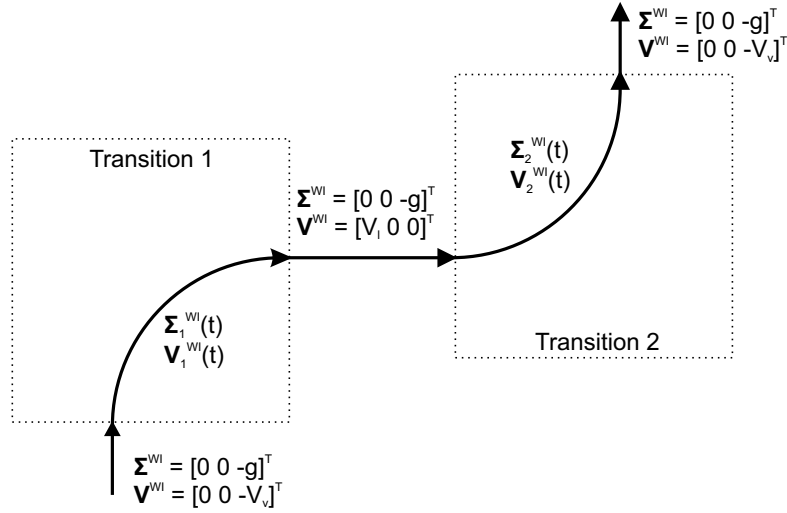


Figure 4.10: Diagram showing the notations used for kinematic trajectory

Airspeed One of the TSD system's strong points is the ease with which trajectories can be created. To test this statement no trajectory smoothing was performed at the start and end of the transition and large jumps in commanded acceleration will occur. Refer to section 3.3 where significant effort went into preventing such jumps in LQR's trajectories. If the system performs as intended, these jumps will not cause significant transients in the response. The airspeed is chosen as linearly increasing or decreasing from the initial to end value.

$$\bar{V}_1(t) = V_v + (V_l - V_v) \frac{t}{T_1} \quad (4.3.31)$$

$$\bar{V}_2(t) = V_l + (V_v - V_l) \frac{t}{T_2} \quad (4.3.32)$$

The reference velocity must be coordinated in the earth axis system, and thus the orientation of the wind axis must be found.

Pitch Angle If the transition is assumed to be a quarter loop the velocity vector will remain tangential to the circle. The pitch angle can be found by calculating the arc-length, s , flown.

$$s_1(t) = V_v t + \frac{1}{2} \frac{V_l - V_v}{T_1} t^2 \quad \text{and} \quad \theta_1(t) = \frac{s_1(t)}{r_1} \quad (4.3.33)$$

$$s_2(t) = V_l t + \frac{1}{2} \frac{V_v - V_l}{T_2} t^2 \quad \text{and} \quad \theta_2(t) = \frac{s_2(t)}{r_2} \quad (4.3.34)$$

With r_1 and r_2 being the radius of the quarter loop flown and T_1 and T_2 being the duration of the individual transitions.

Acceleration To find the specific acceleration of the aircraft at a given position in the trajectory a vector diagram is set up, as seen in figure 4.11. Note that the earth axis system is used. The *specific* acceleration required to over-

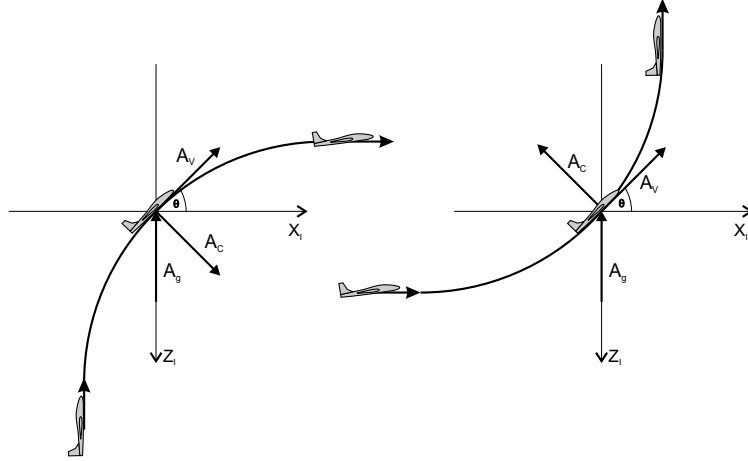


Figure 4.11: Specific Acceleration Vector Diagram

come the *gravitational* acceleration is directed upwards and denoted as A_g , the specific acceleration required to maintain a circular path (centripetal acceleration), A_C , is directed towards the center of the circle. A_V represents the acceleration, or deceleration, needed to change the airspeed.

$$\Sigma_1^{WE} = \begin{bmatrix} A_V \cos \theta_1(t) + A_C \sin \theta_1(t) \\ 0 \\ -A_G - A_V \sin \theta_1(t) + A_C \cos \theta_1(t) \end{bmatrix} \quad (4.3.35)$$

$$\Sigma_2^{WE} = \begin{bmatrix} -A_V \cos \theta_2(t) - A_C \sin \theta_2(t) \\ 0 \\ -A_G - A_V \sin \theta_2(t) - A_C \cos \theta_2(t) \end{bmatrix} \quad (4.3.36)$$

with $A_{C_1} = \frac{\bar{V}_1(t)^2}{r_1}$, $A_{C_2} = \frac{\bar{V}_2(t)^2}{r_2}$
 and $A_{V_1} = \frac{V_L - V_V}{T_1}$, $A_{V_2} = \frac{V_V - V_L}{T_2}$

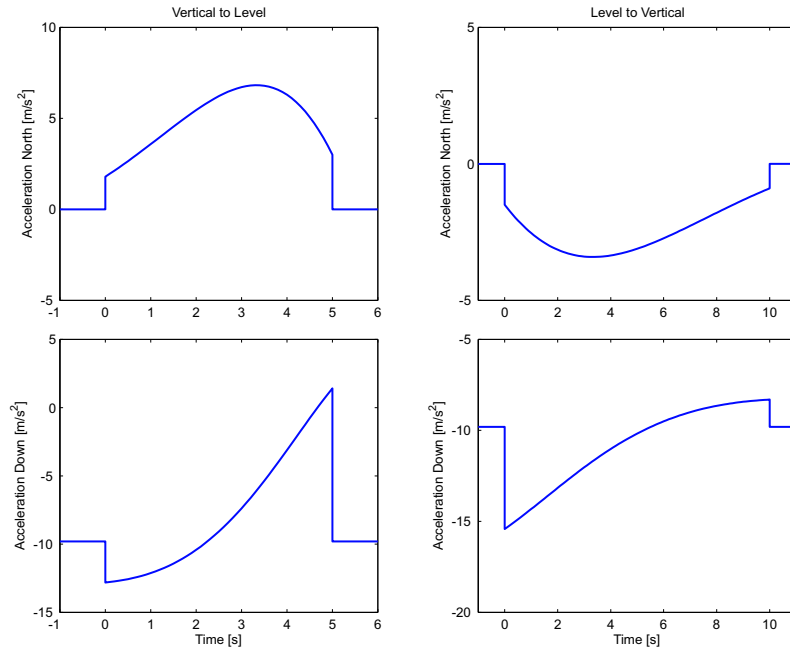


Figure 4.12: Specific Acceleration Trajectory

Velocity The airspeed is known, so all that remains is to coordinate velocity in the earth axes.

$$\mathbf{V}_1^{WE} = \begin{bmatrix} \bar{V} \cos \theta_1(t) \\ 0 \\ -\bar{V} \sin \theta_1(t) \end{bmatrix} \quad (4.3.37)$$

$$\mathbf{V}_2^{WE} = \begin{bmatrix} \bar{V} \cos \theta_2(t) \\ 0 \\ -\bar{V} \sin \theta_2(t) \end{bmatrix} \quad (4.3.38)$$

Position The aircraft's position is easily calculated.

$$\mathbf{P}_1^{WE} = \begin{bmatrix} N_0 + r(1 - \sin \theta_1(t)) \\ E_0 \\ D_0 - r \cos \theta_1(t) \end{bmatrix} \quad (4.3.39)$$

$$\mathbf{P}_2^{WE} = \begin{bmatrix} N_0 + r(1 - \sin \theta_2(t)) \\ E_0 \\ D_0 - r \cos \theta_2(t) \end{bmatrix} \quad (4.3.40)$$

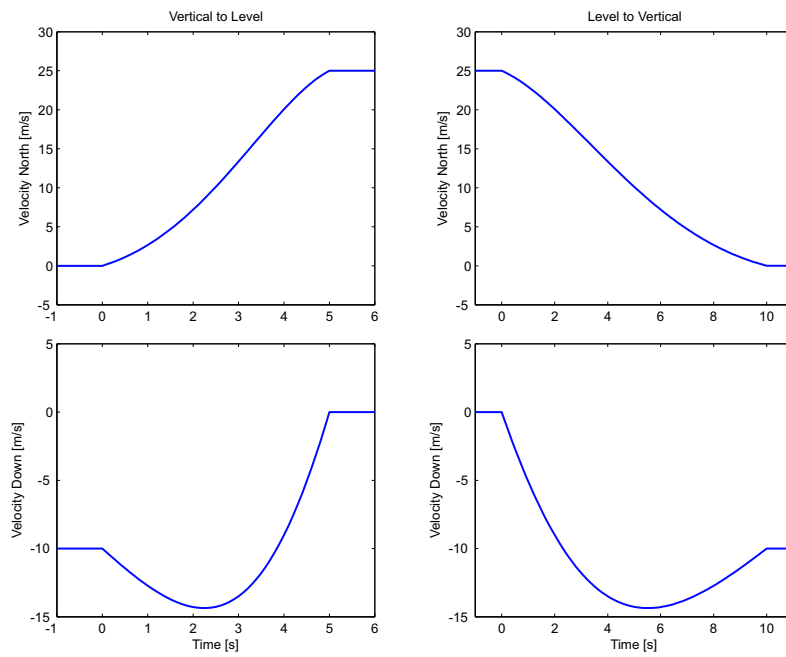


Figure 4.13: Velocity Trajectory

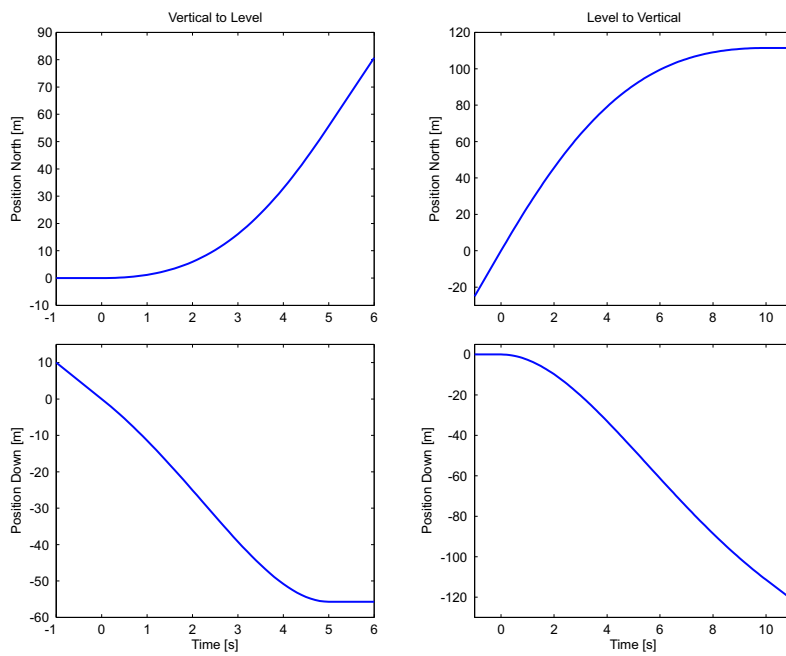


Figure 4.14: Position Trajectory

4.4 Results

4.4.1 Pole Location

The individual pole locations chosen for each controller are listed in table 4.3. It is important to remember that the locations must be chosen to enforce the time scale separation that was assumed when designing the controllers. Primarily this involves placing the NVD and guidance poles far away from all other poles, ideally by a factor of ten. Unfortunately this is not possible in this system as the large inertia of the aircraft limits the system's bandwidth. This necessitated the use of the combined roll and NVD controller, relaxing the constraint on the NVD poles. If non-minimum phase was not compensated for it would introduce an upper limit on pole location for the normal and directional controllers, nevertheless care is still taken to place the affected poles conservatively as large modelling errors can still cause non-minimum phase to occur. An effort was made to place the roll and directional poles as far apart as possible to minimize the effect of the coupling between the two systems.

Controller	Vertical Flight	Level Flight
Axial	$-10, -15$	$-10, -15$
Roll rate	$-5.5 \pm 3.3i$	$-9 \pm 4.5i$
Normal	$-8.5, -7.2 \pm 3i$	$-14, -15 \pm 5i$
Directional	$-5, -3.5 \pm 1.75i$	$-10, -7 \pm 3.5i$
Normal Vector	-0.5	-3.5
Guidance	$-0.5 \pm 0.2i$	$-0.75 \pm 0.45i$

Table 4.3: Chosen Pole Locations

Considering the amount of assumptions made it is expected that the pole locations of the complete system will not lie exactly where they were placed. The calculated feedback is applied to the complete system, with no assumptions made, to ensure that the pole does not move to undesired locations. Considering the way the inner and outer loop combine to form the complete system, calculating the resulting guidance pole locations are difficult. Only the inner loop and NVD pole locations are shown here.

4.4.2 Step Response

As stated before the robustness of a system can be tested by introducing a step disturbance. Once again positions have been chosen as inputs as it will perturb nearly all the aircraft's states. The performance of both the inner loop and guidance controller is tested. The commanded acceleration (from

Controller	Vertical Flight	Level Flight
Axial	$-10.0, -15.0$	$-10.0, -15.0$
Roll rate	$-6.4 \pm 1.5i$	$-10.6 \pm 4.5i$
Normal	$-12.2, -6.0 \pm 3.8i$	$-9.6, -17.6 \pm 9.1i$
Directional	$-2.9, -4.0 \pm 2.3i$	$-5.4, -7.8 \pm 5.1i$
Normal Vector	-0.5	-3.6

Table 4.4: Actual Pole Locations

the guidance controller) is compared to the actual acceleration, illustrating the time scale separation.

4.4.2.1 Level Flight

Normal Step To correct a height error the guidance controller commands a negative normal acceleration. The inner loop controller follows this command by increasing the angle of attack and thus increasing lift. It is clear that the acceleration lags behind the command, but is fast enough to ensure the time scale separation principle is applicable. The control actuation illustrates how the two controls (elevator and collective flaps) work together to remove non-minimum phase from the system.

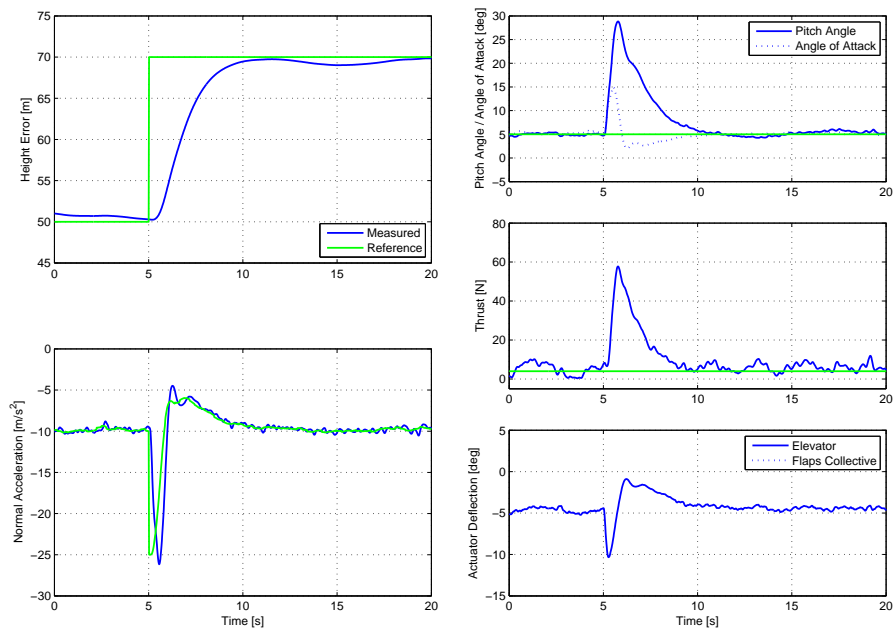


Figure 4.15: Normal step during level flight

Lateral Step During forward flight a significant lift force is present, so lateral translation is achieved by rolling in order to redirect the lift vector. Figure 4.16 shows how the roll angle changes over the course of the response. Note that the normal acceleration increases momentarily to ensure that the vertical component is large enough to maintain altitude. The purpose of such coordinated turns is to regulate lateral acceleration to zero, the only lateral acceleration present is a small error caused when the ailerons are deflected.

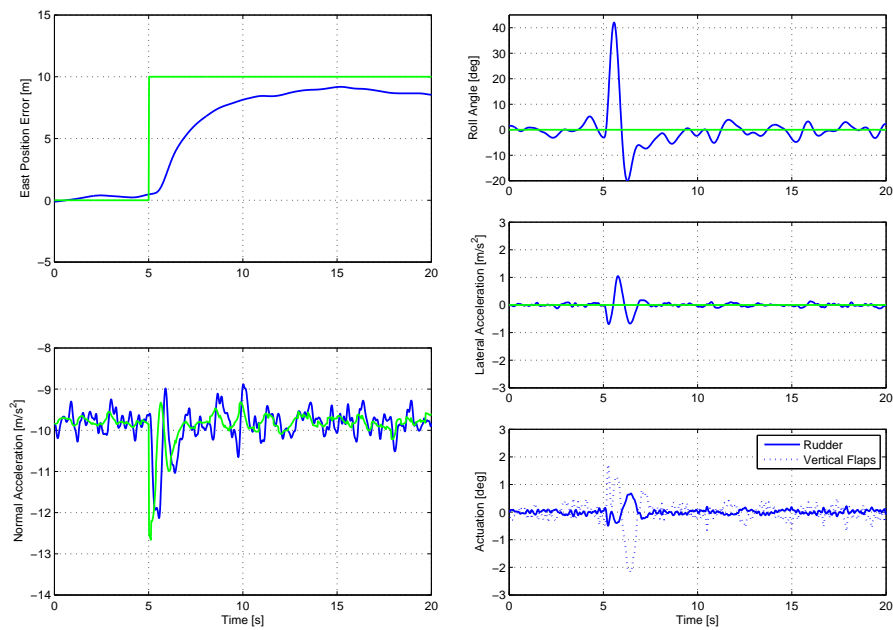


Figure 4.16: Lateral step during level flight

4.4.3 Vertical flight

During vertical flight the lower airspeed necessitates the poles to be closer to the origin. It is clear that the system's response is significantly slower than during forward flight.

Normal Step When the guidance controller receives the reference step it commands a normal acceleration. The inner loop controller increases the angle of attack to create a lift force in the desired direction. It is clear that the combination of actuators successfully combats non-minimum phase.

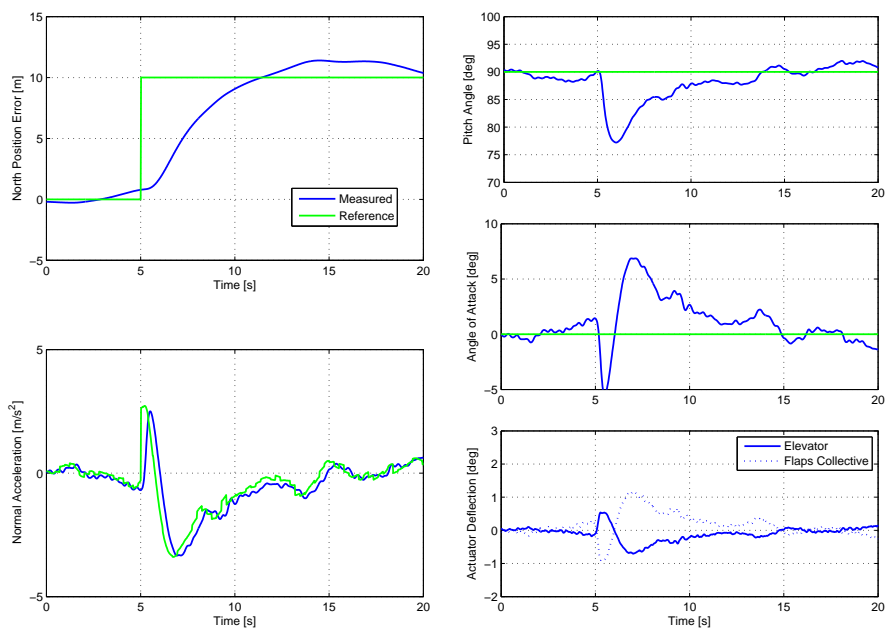


Figure 4.17: Normal step during vertical flight

Lateral Step During vertical flight there is no steady state lift vector and no angle of attack. The method of lateral translation used in forward flight is thus not viable. To create a lateral acceleration the rudder is deflected to create an angle of sideslip resulting in a side-ways force.

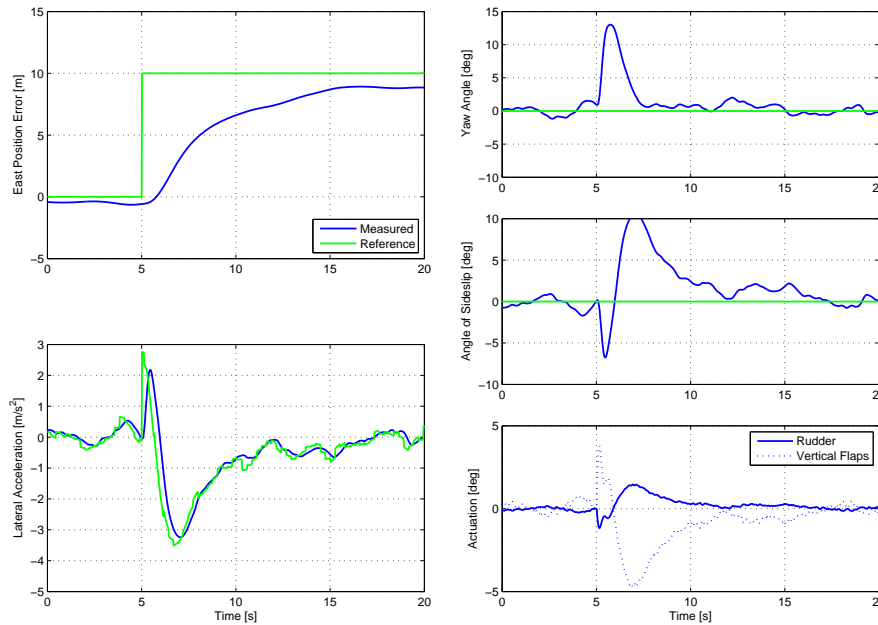


Figure 4.18: Lateral step during vertical flight

4.4.4 Trajectory Tracking

The system's performance was very good, even surpassing the response of the LQR system. Some noteworthy observations are listed here. To aid the comparison of the two control systems the response of five and ten second transitions are given here.

- Positions errors are slightly smaller than those of the LQR system.
- The transitions can be performed very quickly, as fast as 3 and 4 seconds, as shown in appendix E.1. This is due in large part to the very accurate trajectories.
- The angle of attack remains in the linear region.
- The guidance controller works as expected. The lateral acceleration is regulated to zero during level flight, but not during vertical flight or

transitions. Consequently the ailerons are used mainly during level flight, while the rudder and vertical flaps are used elsewhere.

- The response is generally more favourable than that of LQR.

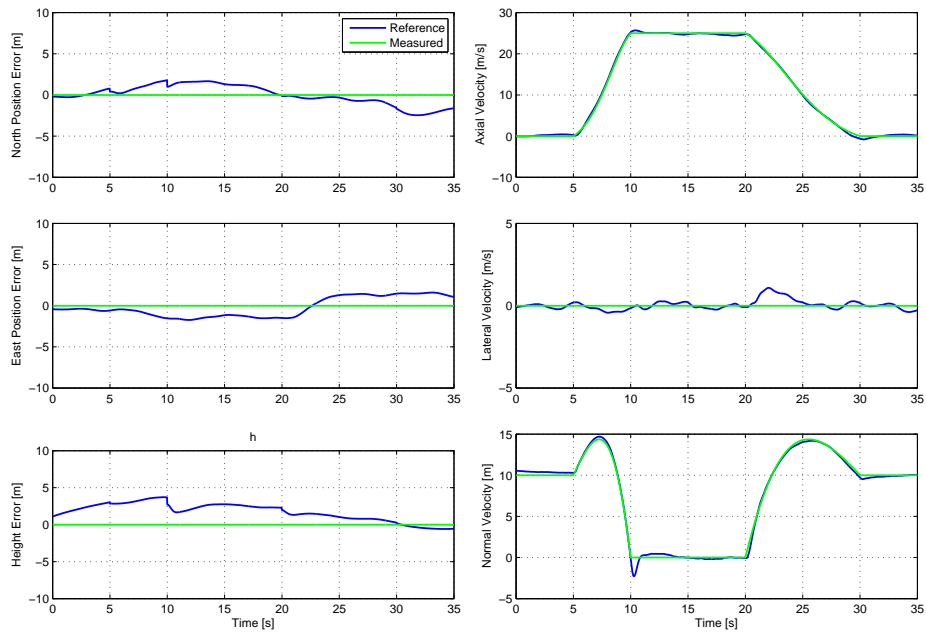


Figure 4.19: Decoupled System Response

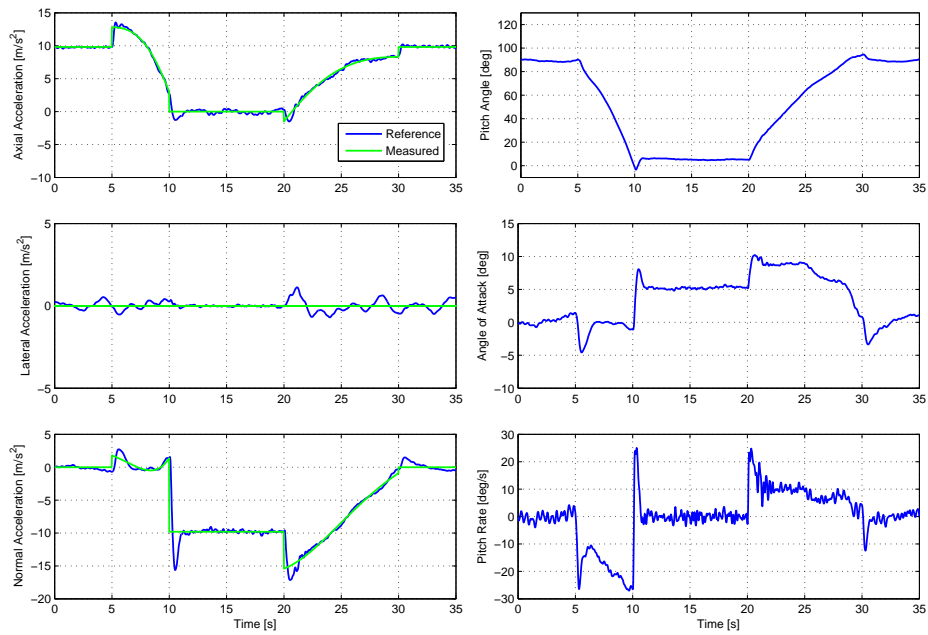


Figure 4.20: Decoupled System Response

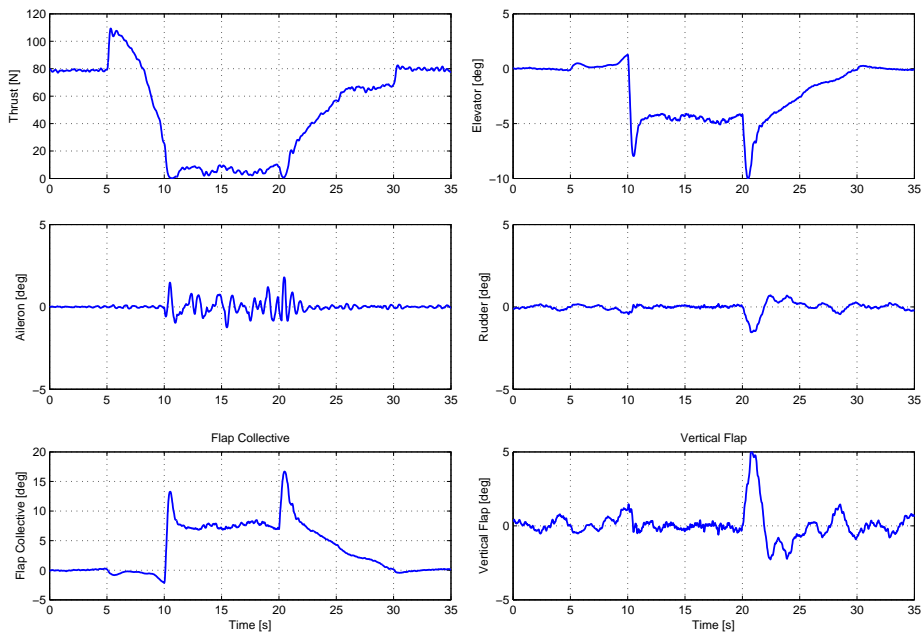


Figure 4.21: Decoupled System Response

Chapter 5

Sensitivity Analysis

5.1 Methodology

Aircraft parameters obtained through empirical calculations and simulations always contain some uncertainty. The uncertainty can be attributed to various unmodelled effects. A good control system must be able to maintain stability even when such modelling errors occur. Some techniques, such as the place algorithm implemented in Matlab, are specifically designed to deliver a robust system. Robustness refers to a systems ability to perform consistently in spite of inaccurate modelling.

To test the robustness of LQR and TSD the gains calculated using the *perfect* model are applied to the a linear system using an altered state matrix. Only one parameter is changed at a time to illustrate the system's sensitivity to each individual parameter. Identifying important parameters enables a designer to spend time and money effectively.

$$C(k) = C_0^{\tau(-1+2\frac{k}{N})} \text{ with } N = 50 \text{ and } k = 0..N - 1 \quad (5.1.1)$$

Parameters have been scaled by a factor of τ over 50 intervals using a geometric spacing. Aerodynamic constants are difficult to determine and a large scale is used $\tau = 1.5$. Physical constants, on the other hand, are much easier to determine accurately and a smaller scaling, $\tau = 1.1$ is used to make results comparable.

A new linear model is calculated using the altered parameter values and feedback is applied using the original gains. The new pole locations are investigated to determine how far the system deviated from its intended response.

5.2 Results

Figures 5.1 and 5.2 shows the movement of poles in the LQR and TSD systems for a change in pitch moment coefficients. The effects of other parame-

ters are listed in appendix E.2.

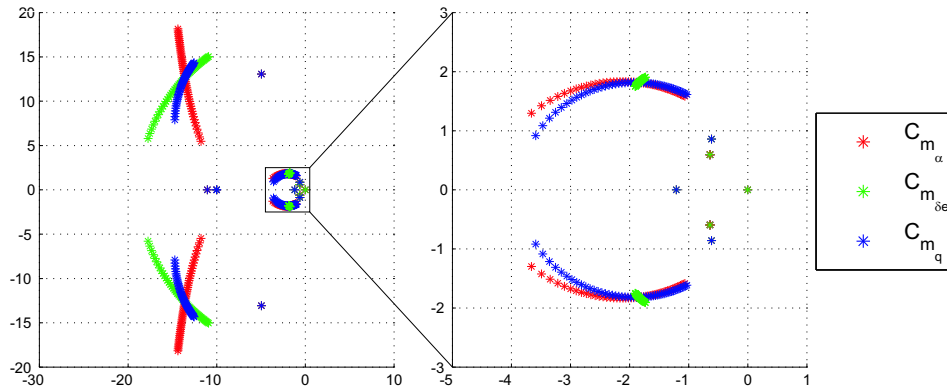


Figure 5.1: LQR Sensitivity to Pitch Moment parameters

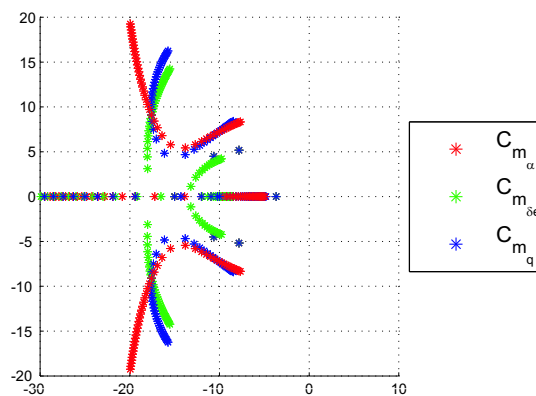


Figure 5.2: TSD Sensitivity to Pitch Moment parameters

5.2.1 Comparison

From the loci it is clear that the TSD system is more susceptible to parameter changes. It would be helpful to attribute a number to each parameter to represent its influence on the system. Finding a meaningful definition for this *influence* is not easy and a very straightforward method is used here. The distance each pole moves is calculated. The distance is written as a percentage of the poles natural frequency. A pole at 0.5 rad/s moving 1 rad/s can cause instability, while a 20 rad/s pole moving the same distance will have a very small effect. The RMS of these *distances* are taken to describe the change

to the system as a whole. The RMS was used since one pole moving a significant distance can affect the system more than all pole locations changing slightly.

LQR				TSD			
Name	Change	Name	Change	Name	Change	Name	Change
$C_{L\alpha}$	39.7%	$C_{l_{\delta a}}$	31.2%	$C_{L\alpha}$	33.6%	$C_{l_{\delta a}}$	40.3%
C_{Lq}	6.4%	$C_{l_{\delta r}}$	1.2%	C_{Lq}	6.3%	$C_{l_{\delta r}}$	39.1%
$C_{L\delta e}$	9.5%	$C_{m\alpha}$	47.1%	$C_{L\delta e}$	8.8%	$C_{m\alpha}$	89.4%
$C_{Y\beta}$	3.2%	C_{m_q}	24.0%	$C_{Y\beta}$	60.4%	C_{m_q}	56.0%
C_{Y_p}	1.2%	$C_{m_{\delta e}}$	59.0%	C_{Y_p}	1.7%	$C_{m_{\delta e}}$	105.0%
C_{Y_r}	0.7%	$C_{n_{\beta}}$	19.1%	C_{Y_r}	11.1%	$C_{n_{\beta}}$	95.8%
$C_{Y_{\delta a}}$	0.0%	C_{n_p}	1.1%	$C_{Y_{\delta a}}$	0.0%	C_{n_p}	3.4%
$C_{Y_{\delta r}}$	2.6%	C_{n_r}	5.9%	$C_{Y_{\delta r}}$	30.2%	C_{n_r}	24.7%
$C_{l_{\beta}}$	0.1%	$C_{n_{\delta a}}$	0.6%	$C_{l_{\beta}}$	1.2%	$C_{n_{\delta a}}$	1.4%
C_{l_p}	51.4%	$C_{n_{\delta r}}$	5.5%	C_{l_p}	61.4%	$C_{n_{\delta r}}$	124.4%
C_{l_r}	1.9%			C_{l_r}	7.0%		
m	7.4%	I_y	5.1%	m	14.0%	I_y	12.0%
I_x	5.5%	I_z	3.4%	I_x	10.0%	I_z	9.4%

Table 5.1: Sensitivity to Various Parameters

Table 5.1 lists the change caused by each parameter. The parameters causing the largest change for both LQR and TSD are listed in bold. A few conclusions can be drawn from the listed values:

- Across the board LQR is less sensitive to parameter changes.
- LQR and TSD are largely sensitive to the same parameters: $C_{m_{\delta e}}$, $C_{L\alpha}$ and $C_{m\alpha}$ significantly effects both systems, suggesting that the sensitivity is introduced by the natural dynamics of the aircraft, not the control system.
- In some cases TSD is considerably affected by a certain parameter, such as $C_{n_{\delta r}}$, while LQR is not. This sensitivity is introduced by control techniques such a feedback linearisation.
- The amount of uncertainty that can be expected in physical parameters does not effect the system as significantly as the aerodynamic coefficients.

The exact cause of TSD's increased sensitivity must be investigated. Some possible explanations are:

- Some of the assumptions made when simplifying the model do not remain valid when certain coefficients become large.
- Feedback linearisation can introduce significant sensitivity to the system
- The chosen pole locations can affect sensitivity. Unexpected coupling between poles have a larger effect when the poles are close together. An effort should be made to find optimal locations, especially for the roll and lateral poles.

Chapter 6

Conclusions and Recommendations

6.1 Overview

This thesis detailed the design and simulation of control systems for a VTOL UAV. A proven non-linear model was adapted to suite the aircraft's particular geometry and included in an existing simulation environment. Several design choices that affect modelling and control were presented, especially in the areas of axis systems, attitude description and decoupling. Arguments were presented to support the ultimate decisions in each area. Two control systems were created:

The derived non-linear model was linearised in order to apply feedback control. Optimal feedback gains were found and a Linear Quadratic Regulator was implemented. Various techniques, such as maximum deviations and modal analysis were investigated to aid in the design of weighting matrices. The system's step responses and gain settings were examined to ensure that the conditions for guaranteed stability are met. A step-by-step method for creating trajectories was presented.

A novel Time-Scale Decoupled model was applied. Control systems were created for each of the individual models and arguments were raised for optimal pole locations. The validity of the assumptions required for decoupling was tested by calculating the pole locations of the complete model. The trajectory creation procedure for TSD was detailed and contrasted with LQR.

Sensitivity analysis was performed on both systems to test the effect of uncertain aircraft parameters. The effect of individual parameters was also investigated to establish their relative importance.

6.2 Comparison

Judging a control system solely on the final performance is sensible, but in ignoring the method of reaching the final iteration one can lose insight into the design process. The purpose of this section is to investigate the differences between LQR and the TSD controller throughout the design process.

6.2.1 Understanding

Having a thorough understanding of a control system is essential. The easier a scheme is to grasp the faster it will develop into a first simulation. The basis for LQR is very mathematical in nature; luckily the understanding of the derivation has no impact on its use. The concept of weighting matrices offers a physical handle on the mathematics. In contrast, a TSD system requires a much deeper understanding of the dynamics of the aircraft and specifically how the individual systems interconnect. To understand the *big picture* of the entire system can take some time.

6.2.2 Design

A troublesome step in designing an LQR system is finding initial weightings that successfully control the system. In an unstable system the states diverge too quickly and the designer cannot establish what state weightings to change. Once suitable weightings are found, performance is increased through an iterative process. Care must be taken to investigate all aspects of the system (gain settling, step response and trajectory tracking) when changing the weightings. This tends to be a very time consuming and often frustrating procedure that easily leads to a loss of insight.

Choosing the pole locations is a more direct method of changing the response of the system. The choice of pole locations is constrained by time scale separation making a viable first iteration easier to find. Some design freedom remains to choose exact speed and damping of each pole.

6.2.3 Refinement

Inevitable the first iteration of any system will not be perfect, it will need to be refined. Having insight into *how* a controller accomplishes its goal enables the designer to effectively refine the design. In this aspect LQR is a closed book. As shown, LQR operates in the same way as a conventional control system, but this is only apparent once *optimal* weightings have been found. Throughout the design process several strange and non-intuitive behaviour has been observed. Small changes in weightings have been noted to dramatically change the response. For a large section of the design process LQR is a mystery resulting in very long debug times and lost hours spent investigating the effect of various weightings.

The time invested in understanding the TSD system becomes advantageous once a system has been implemented. When investigating a specific problem the controller where the problem originates can easily be found. By knowing exactly how the system should respond, any strange response can be investigated. In an LQR system many unexpected occurrences may be attributed to the particular set of weightings, while it could conceivably be an error in the model or the simulation.

6.2.4 Implementation

The implementation of a Linear Quadratic Regulator can be broken down into two important steps: Calculating gains and applying those gains. The calculation of LQR gains, while the most computationally demanding process, is easy to implement in code. In a high level simulations environment such as Matlab, solving the Riccati equation is very easy. Implementing the same algorithm on an avionics system in a language such as C++ can be more troublesome as matrix computations become cumbersome. Care should be taken with the implementation to optimize speed, memory usage and stability. Applying the calculated gains are trivial.

The TSD system contains significantly more *structure* than LQR and will require more time to implement in either high or low level environments. While this structure does result in a system that is easier to debug and analyse, it can cause several problems if not implemented correctly.

6.2.5 Trajectory Creation

As shown, the creation of LQR trajectories can be a laborious and inaccurate procedure. Calculating a trajectory for every state is often not viable. The errors introduced in this area of the controller remains the limiting factor when performing transitions.

The trajectories created for a TSD control system are not only 100% accurate, but can be used for any aircraft. This is incredibly useful if a variety of aircraft must perform the same maneuver or if a single aircraft's parameters are changed. An LQR trajectory would need to be recalculated if modifications were made to an aircraft, while the TSD trajectory would be unaffected.

Both TSD and LQR have the benefit of the trajectory being separate from the actual control system. It would be possible for a designer with no control system knowledge to specify a trajectory without the need to change the system itself.

6.2.6 Performance

Once optimal weightings are found LQR performs well during level and vertical flight. Performance during transitions is hampered by inaccurate trajectories. LQR delivers a robust system and performs well when modelling

errors are introduced. Through the design experience it has become obvious that LQR is not suited for the large system used in this project.

The TSD system performs significantly better than LQR as trajectories can be flown faster and with smaller errors. The improvement over LQR can be attributed to the accurate trajectory calculations. Unfortunately, TSD was significantly more sensitive to uncertainties. Hopefully this will lead to further investigation into how TSD can be made more robust.

6.3 Recommendations

Of the two systems tested the TSD controller is the clear choice to use in this application.

- There are several other implementations of quaternion that do not exhibit the problems listed in this thesis. It revolves around describing the aircraft's orientation relative to the reference trajectory and using only three of the four quaternions. The fourth quaternion can be calculated if the other three are known. The resulting system will behave consistently and remain controllable over the entire flight envelope.
- The performance of the TSD system is limited by the amount of assumptions made. Some of these, especially the decoupling of the roll and directional dynamics, must be further investigated.
- LQR is not suited to operate on such a large system containing large range of time scales. Gains take several hundreds of time steps to stabilize placing restrictions on the possible weighting selections. LQR would be ideally suited to control the individual decoupled systems obtained when a model is time scale decoupled. If LQR is used the over-simplification that hampers TSD can be reduced. LQR can control the directional and roll dynamics as a single system.
- The TSD system is very sensitive to modelling uncertainty; this could be attributed to feedback linearisation or poorly placed poles and should be investigated further.
- The `place` algorithm can be used to better utilize the additional control surfaces in the TSD system. Currently the actuators operate in a fixed mixing. By adding an additional degree of freedom the system can be made more robust.

Appendices

Appendix A

Axis System Transformations

A.1 Vector Notation

Throughout this project several different axis systems are used to express vectors. Understanding the notation of vectors and vector transformations are integral to grasping concepts quickly and easily. What follow is a quick overview of vector notation and the origin of the direction cosine matrices used in axis transformations as explained in [2].

Consider a vector in three dimensional space, $\mathbf{R} \in \mathfrak{R}^3$. This vector is not bound to any specific axis system and can be expressed as a linear sum of three vectors that span three dimensional space. For axis system A basis vectors are chosen three orthogonal unit vectors i^A, j^A and k^A . Consequently \mathbf{R} can be written as:

$$\mathbf{R} = X^A i^A + Y^A j^A + Z^A k^A = X^B i^B + Y^B j^B + Z^B k^B \quad (\text{A.1.1})$$

Note that the vector coordinated in two different systems is equal.

Writing the basis vectors of the axis systems becomes cumbersome. To simplify matters a *coordinate vector* is defined as a matrix of coordinate values:

$$\mathbf{R}_A = \begin{bmatrix} X^A \\ Y^A \\ Z^A \end{bmatrix} \quad (\text{A.1.2})$$

It is very important to note that, since the basis vectors have been removed, two coordinate vectors of the same vector is not equal.

$$\mathbf{R}_A = \begin{bmatrix} X^A \\ Y^A \\ Z^A \end{bmatrix} \neq \begin{bmatrix} X^B \\ Y^B \\ Z^B \end{bmatrix} = \mathbf{R}_B \quad (\text{A.1.3})$$

A.2 Coordinate Transformations

There are several situations where a vector coordinated in one axis system needs to be expressed in another. A common example is coordinating velocity in the earth axis system to calculate displacement in N,E,D. Note that when a vector is transformed the physical vector does not change, only its representation.

To transform a vector coordinated in axis system A to axis system B each unit vector of A is coordinated in B as in [2].

$$\begin{aligned}
 \mathbf{R} &= X^A \mathbf{i}^A + Y^A \mathbf{j}^A + Z^A \mathbf{k}^A & (A.2.1) \\
 &= X^A [(\mathbf{i}^A \cdot \mathbf{i}^B) \mathbf{i}^B + (\mathbf{i}^A \cdot \mathbf{j}^B) \mathbf{j}^B + (\mathbf{i}^A \cdot \mathbf{k}^B) \mathbf{k}^B] + \\
 &\quad Y^A [(\mathbf{j}^A \cdot \mathbf{i}^B) \mathbf{i}^B + (\mathbf{j}^A \cdot \mathbf{j}^B) \mathbf{j}^B + (\mathbf{j}^A \cdot \mathbf{k}^B) \mathbf{k}^B] + \\
 &\quad Z^A [(\mathbf{k}^A \cdot \mathbf{i}^B) \mathbf{i}^B + (\mathbf{k}^A \cdot \mathbf{j}^B) \mathbf{j}^B + (\mathbf{k}^A \cdot \mathbf{k}^B) \mathbf{k}^B] \\
 &= X^B \mathbf{i}^B + Y^B \mathbf{j}^B + Z^B \mathbf{k}^B
 \end{aligned}$$

This can easily be represented as a matrix multiplication.

$$\begin{aligned}
 \mathbf{R} &= \begin{bmatrix} \mathbf{i}^B & \mathbf{j}^B & \mathbf{k}^B \end{bmatrix} \begin{bmatrix} X^B \\ Y^B \\ Z^B \end{bmatrix} & (A.2.2) \\
 &= \begin{bmatrix} \mathbf{i}^B & \mathbf{j}^B & \mathbf{k}^B \end{bmatrix} \begin{bmatrix} \mathbf{i}^A \cdot \mathbf{i}^B & \mathbf{j}^A \cdot \mathbf{i}^B & \mathbf{k}^A \cdot \mathbf{i}^B \\ \mathbf{i}^A \cdot \mathbf{j}^B & \mathbf{j}^A \cdot \mathbf{j}^B & \mathbf{k}^A \cdot \mathbf{j}^B \\ \mathbf{i}^A \cdot \mathbf{k}^B & \mathbf{j}^A \cdot \mathbf{k}^B & \mathbf{k}^A \cdot \mathbf{k}^B \end{bmatrix} \begin{bmatrix} X^A \\ Y^A \\ Z^A \end{bmatrix} \\
 \Rightarrow \mathbf{R}_B &= [DCM^{BA}] \mathbf{R}_A & (A.2.3)
 \end{aligned}$$

The transformation matrix is known as a Direction Cosine Matrix (DCM) as each element represents the angle between two basis vectors. Further investigation of the DCM reveals that:

$$\mathbf{R}_A = [DCM^{AB}] \mathbf{R}_B = [DCM^{BA}]^T \mathbf{R}_B \quad (A.2.4)$$

This shows that the transpose of a DCM matrix is equal to its inverse, sufficient condition for orthonormality.

To describe the attitude of an axis system it is not necessary to store all 9 DCM parameters. Orthonormality ensures that only three of the parameters are independent, subject to six constraints. Several systems exist through which the amount of parameters and constraints are reduced, each with positive and negative aspects.

Euler Angles Euler Angles use three angles to describe the difference between two axis systems [7]. Each of these axis systems represents a rotation around a basis vector. To transform from axis system A to B the following transformations are performed, where \mathbf{u}_i^A represents the i^{th} unit vector of the A axes system.

1. Rotate A through angle ψ about unit vector \mathbf{u}_u^A . Call this new axis system B_1 .
2. Rotate B_1 through angle θ about unit vector $\mathbf{u}_v^{B_1}$. Call this new axis system B_2 .
3. Rotate B_2 through angle ϕ about unit vector $\mathbf{u}_w^{B_2}$. Call this new axis system B .

By choosing values for u, v and w different Euler sequences can be created. The most commonly used sequence is 3-2-1, translating to Yaw-Pitch-Roll or Heading-Attitude-Bank. To be a valid Euler sequence the same basis vector may not be used consecutively making only 12 of the possible 27 sequences useful.

Unfortunately, each Euler sequence introduces a singularity at a certain attitude. This occurs when the second rotation causes the basis vectors of the first and third rotation to be equal. This results in an infinite amount of possible Euler representations for the same orientation. For Euler 3-2-1 this occurs at a pitch angle of $\pi/2$. This poses a serious problem since vertical flight occurs at the singularity. For vertical flight the Euler 2-3-1 sequence was used.

To derive the DCMs for each Euler sequence the transformation matrices for rotation around a each basis vector is investigated.

$$\mathbf{T}_\theta^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \quad \mathbf{T}_\theta^2 = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix}$$

$$\mathbf{T}_\theta^3 = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

\mathbf{T}_θ^i is the transformation matrix that describes a rotation of θ about the i^{th} unit vector. From here the DCM matrices can be found by combining the single rotation transformation matrices. Note that in this project the attitude parameters store the orientation of the body axis with respect to the inertial axis system.

$$[\mathbf{DCM}_{321}^{BE}] = \mathbf{T}_\phi^1 \mathbf{T}_\theta^2 \mathbf{T}_\psi^3 \quad (\text{A.2.5})$$

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

$$[\mathbf{DCM}_{231}^{BE}] = \mathbf{T}_\phi^1 \mathbf{T}_\theta^3 \mathbf{T}_\psi^2 \quad (\text{A.2.6})$$

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

These two matrices represent the same transformation and are therefore equal, different notations are only used to show what attitude parametrization was used to obtain it.

Wind Axis Similarly a transformation matrix can be set up to transform from the wind to body axis system:

$$\begin{aligned} [DCM^{BW}] &= T_{\alpha}^2 T_{-\beta}^3 & (A.2.7) \\ &= \begin{bmatrix} C_{\alpha}C_{\beta} & -C_{\alpha}S_{\beta} & -S_{\alpha} \\ S_{\beta} & C_{\beta} & 0 \\ S_{\alpha}C_{\beta} & -S_{\alpha}S_{\beta} & C_{\alpha} \end{bmatrix} \end{aligned}$$

Quaternions Quaternions operate on Euler's theorem that the orientation of axis system B relative to A can be uniquely described as a rotation about vector λ of angle μ . The Quaternion vector used in this project is defined in [6] as

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} \lambda_x \sin \mu/2 \\ \lambda_y \sin \mu/2 \\ \lambda_z \sin \mu/2 \\ \cos \mu/2 \end{bmatrix} \quad \text{with } q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1 \quad (A.2.8)$$

Reference [6] shows that the DCM can be written as:

$$[DCM_{q_1-4}^{BI}] = \begin{bmatrix} q_4^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_4q_3 + q_1q_3) & 2(q_1q_3 - q_4q_2) \\ 2(q_1q_2 - q_4q_3) & q_4^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_4q_1 + q_2q_3) \\ 2(q_4q_2 + q_1q_3) & 2(q_2q_3 - q_4q_1) & q_4^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (A.2.9)$$

Appendix B

Feedback Linearisation

B.1 Overview

This appendix will briefly discuss the method of feedback linearisation as explain in [15]. This technique is used extensively in the controllers of the time scale decoupled system [2]. The central principle is to transform non-linear systems into a linear system so that established linear control techniques can be used.

Central to this method is the concept of *relative degree*. Observe the SISO system:

$$\begin{aligned}\dot{x} &= f(x) + g(x)u \\ y &= h(x)\end{aligned}\tag{B.1.1}$$

With f , g and h begin differentiable functions. The output, y , does not depend directly on the input, u . The output is derived to give:

$$\dot{y} = \frac{\partial h}{\partial x} \dot{x} = \frac{\partial h}{\partial x} (f(x) + g(x)u)\tag{B.1.2}$$

$$= \frac{\partial h}{\partial x} f(x) + \frac{\partial h}{\partial x} g(x)u\tag{B.1.3}$$

If $\frac{\partial h}{\partial x} g(x) \neq 0$ then the *derivative* of the output depends directly on the input and the system is said to have a *relative degree* of one. Otherwise the output is differentiated again:

$$\ddot{y} = \frac{\partial}{\partial x} \left[\frac{\partial h}{\partial x} f(x) \right] \dot{x}\tag{B.1.4}$$

$$= \frac{\partial}{\partial x} \left[\frac{\partial h}{\partial x} f(x) \right] f(x) + \frac{\partial}{\partial x} \left[\frac{\partial h}{\partial x} f(x) \right] g(x)u\tag{B.1.5}$$

If $\frac{\partial}{\partial x} \left[\frac{\partial h}{\partial x} f(x) \right] \neq 0$ then the system has a relative order of 2, if not the system is differentiated again until a non-zero coefficient for $g(x)u$ is found.

To simplify the notation the Lie Derivative is defined as:

$$\begin{aligned} L_f^k h(x) &= \frac{\partial (L_f^{k-1})}{\partial x} f(x) \\ L_f^0 h(x) &= h(x) \end{aligned}$$

Enabling the following definition of a system with relative degree r .

$$\begin{aligned} L_g^0 L_f^{k-1} h(x) &= 0 \quad \text{with } k \in [1, r-1] \\ L_g^0 L_f^{r-1} h(x) &\neq 0 \end{aligned} \quad (\text{B.1.6})$$

To illustrate how feedback linearisation works, the r^{th} derivative of the output is investigated:

$$y^{(r)} = L_f^r h(x) + L_g^0 L_f^{r-1} h(x) u \quad \text{with } L_g^0 L_f^{r-1} h(x) \neq 0 \quad (\text{B.1.7})$$

By setting the feedback law equal to,

$$u = \frac{1}{L_g^0 L_f^{r-1} h(x)} (v - L_f^r h) \quad (\text{B.1.8})$$

where v is a reference signal. The resulting relationship between v and $y^{(r)}$ is,

$$y^{(r)} = v \quad \text{or} \quad y(s) = \frac{1}{s^r} v(s) \quad (\text{B.1.9})$$

The system is now linear form the reference signal to the output.

Appendix C

Aircraft Parameters

Various aircraft parameters are required to complete the non-linear model. Some of these can be easily measured; others can be calculated with empirical methods while some need to be approximated with numerical simulations.

C.1 Physical Properties

Mass The energy density of the lithium polymer batteries used to drive the motors are substantially less than the methanol used to fuel traditional RC two-stroke engines. Subsequently the aircraft is significantly heavier than a engine powered aircraft of the same size. The weight with avionics included has been determined as,

$$m = 9.0 \text{ kg} \quad (\text{C.1.1})$$

Moment of Inertia and Center of Gravity The moment of inertia in each axis is measured using a double pendulum method described in [1]. The aircraft is suspended from two strings that are parallel to the body axis in which the inertia is to be measured. The aircraft is perturbed around the axis and the period of the oscillation is measured. Figure C.1 describes the setup. For the method to yield accurate results the strings must be light, the oscillations must be kept small and $d \ll l$. Table C.1 lists the moments of inertia calculated for the aircraft. The position of the CG is measured from the tip of the wings.

The tests were done without the avionics and batteries. The avionics and batteries can be approximated as objects with uniform density making it possible to track the movement of the CG for different locations of the batteries and avionics. The optimal location for the CG is slightly ahead of the center of lift.

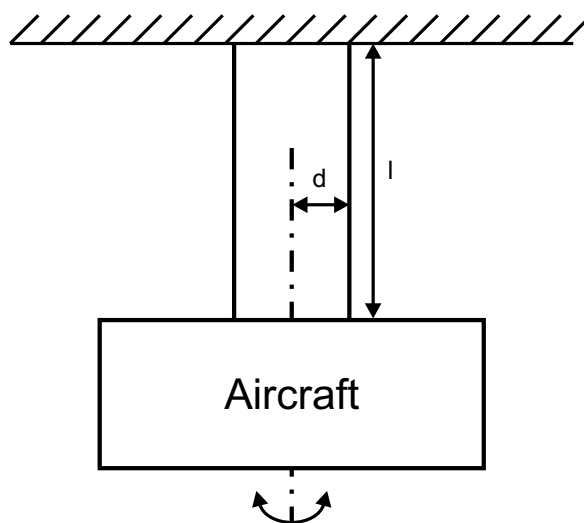


Figure C.1: Moment of Inertia Measurement Setup

Axis (Body)	Moment of Inertia	Center of Gravity
X	0.8877 [kg.m ²]	-0.1371 [m]
Y	0.4208 [kg.m ²]	0 [m]
Z	1.2434 [kg.m ²]	0 [m]

Table C.1: Moments of Inertia

C.2 Aerodynamic

C.2.1 Static Coefficients

Static coefficients are used to calculate the drag, lift and pitching moment at zero angle of attack. AVL cannot calculate static values, so these need to be found empirically. The wing of this particular aircraft is symmetrical resulting in no lift force or pitching moment generated at zero angle of attack.

$$C_{L_0} = 0 \quad \text{and} \quad C_{M_0} = 0 \quad (\text{C.2.1})$$

Empirical drag calculations are not accurate and large errors can occur. Controllers must be made robust enough to compensate for an incorrect drag

model. The static drag can be approximated as outlined in [20].

$$R = \frac{\rho V c}{\mu} \quad (\text{C.2.2})$$

$$C_f = \frac{0.455}{(\log_{10} R)^{2.58}} \quad (\text{C.2.3})$$

$$K = 1 + 2\left(\frac{t}{c}\right) + 100\left(\frac{t}{c}\right)^4 \quad (\text{C.2.4})$$

$$C_{D_0} = 1.1 K C_f \frac{S_{wet}}{S_{ref}} \quad (\text{C.2.5})$$

When calculating drag it has been assumed that only the wing and tail section contributes to drag. The parameters for each of these surfaces are given in Table C.2.

Name	Symbol	Wing	Tail
Velocity	V	25	25
Mean Area Chord	c	0.3656	0.1400
Natural Viscosity	μ	1.7894×10^{-5}	1.7894×10^{-5}
Reynolds Number	R	6.256×10^5	2.3961×10^5
Skin Friction Coefficient	C_f	0.0049	0.0059
Pressure Drag Correction Factor	K	1.2332	1.1006
Thickness to MAC ration	$\frac{t}{c}$	0.1094	0.0500
Wetted Area	S_{wet}	1.0961	0.7498
Reference Surface Area	S	0.5737	0.5737
Parasitic Drag Coefficient	C_{D_0}	0.0172	0.0100

Table C.2: Parameters for Calculating Drag

C.2.2 Stability and Control Derivatives

Most aerodynamic coefficients were found through simulation in AVL (Athena Vortex Lattice). AVL makes use of computational fluid dynamics to calculate aerodynamics properties. Being a numeric simulation, AVL's results are only accurate under certain conditions. This aircraft has a conventional configuration and no aggressive manoeuvres are performed, ensuring that the restrictions placed in [12] are never broken. A geometry plot of the aircraft is shown in figure C.2.

The following stability and control derivatives were found at a trim airspeed of 25m/s. All derivatives are dimensionless and coordinated in the wind axes and control derivatives have been converted to *per radians* from the *per degrees* values provided by AVL.

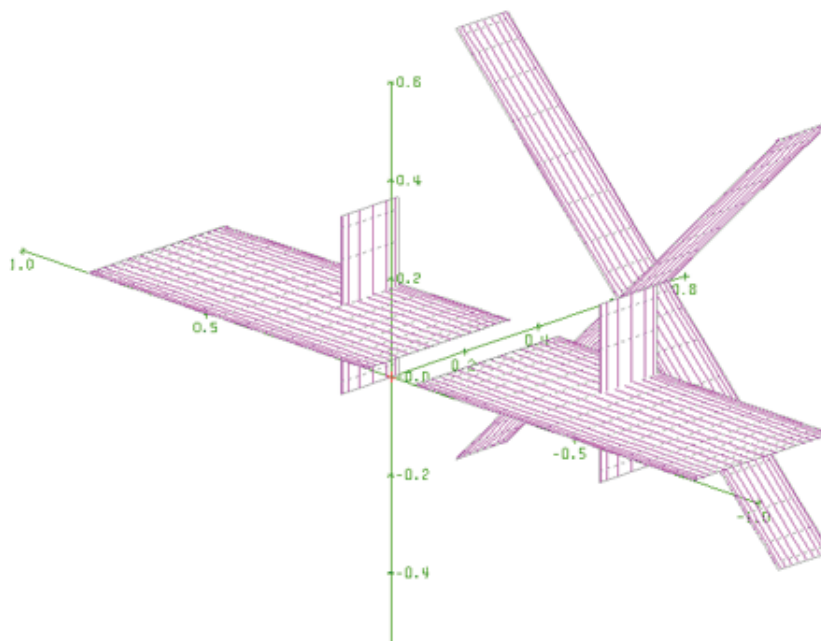


Figure C.2: Aircraft Geometry in AVL

Name	$\alpha = 0^\circ$	$\alpha = 5.9^\circ$	Name	$\alpha = 0^\circ$	$\alpha = 5.9^\circ$
$C_{L\alpha}$	4.2536	4.1877	C_{Y_p}	0	0.25571
$C_{m\alpha}$	-1.6497	-1.6147	C_{l_p}	-0.6615	-0.6566
$C_{Y\beta}$	-1.7373	-1.7190	C_{n_p}	0	-0.0689
$C_{l\beta}$	0.0000	-0.0043	C_{L_q}	8.17509	8.1626
$C_{n\beta}$	0.4766	0.4745	C_{m_q}	-7.636722	-7.63672
			C_{Y_r}	1.21431	1.1875
			C_{l_r}	0	0.10391
			C_{n_r}	-0.3960	-0.3359

Table C.3: Stability Derivatives

Name	$\alpha = 0^\circ$	$\alpha = 5.9^\circ$	Name	$\alpha = 0^\circ$	$\alpha = 5.9^\circ$
$C_{L_{\delta_e}}$	1.2618	1.2434	$C_{L_{\delta_{fc}}}$	0.7479	0.7479
$C_{m_{\delta_e}}$	-1.8281	-1.8088	$C_{m_{\delta_{fc}}}$	-0.0827	-0.0827
$C_{Y_{\delta_a}}$	0	0.0009	$C_{l_{\delta_{fd}}}$	-0.1149	-0.1149
$C_{l_{\delta_a}}$	-0.2225	-0.2193	$C_{n_{\delta_{fd}}}$	0	0.0092
$C_{n_{\delta_a}}$	0	0.0186	$C_{Y_{\delta_v}}$	0.3113	0.3113
$C_{Y_{\delta_r}}$	1.0186	1.0078	$C_{l_{\delta_v}}$	0	-0.0033
$C_{l_{\delta_r}}$	0	-0.0410	$C_{n_{\delta_v}}$	-0.0173	-0.0173
$C_{n_{\delta_r}}$	-0.3948	-0.3885			

Table C.4: Control Derivatives

Derivatives have been calculated at two different angle of attack values. During vertical flight the angle of attack is zero, or close to zero, while the forward flight trim angle of attack is 5.9° . A linear interpolation was used to find stability derivatives for other angles of attack. The system is robust enough to not require the derivatives to be recalculated for changing angles of attack, but performance is increased slightly.

Appendix D

Simulation Environment

As stated before a modular design approach is essential for an effective simulation environment. In an Electronic Systems Laboratory a library of objects has been built to ensure consistent and reproducible results for all projects. The simulation environment consists of several such objects and some project specific blocks interfacing a predefined manner. All objects have been designed and tested as part of previous projects conducted at the ESL and have been well documented. Subsequently the exact operation of each object will not be discussed in detail and it is assumed that all objects' performance are up to standard.

D.1 Non-Linear Aircraft Simulation

The non-linear aircraft model given in [8] has been implemented as a MATLAB S-function as part of [5]. The aircraft parameters calculated in Appendix C entered as parameters. The S-function receives the perfect state vector (from the six degree of freedom block) and the actuator positions (from the servo-model) and outputs the forces and moments acting on the body. The model is fully non-linear and operates at a very high sampling rate offering the best possible simulation of real-world dynamics.

For this project the object has been expanded slightly to include factors unique to this aircraft:

- The effect of induced airspeed over actuation surfaces has been added. To make the simulation as accurate as possible the airspeed is calculated empirically using MATLAB's `solve` function. This is in contrast to a numerical method that is used in the controller.
- The new actuation surfaces have been included.
- Additional aerodynamic coefficients have been included to accurately simulate flight at zero angle of attack as explained in appendix C.

D.2 Six Degree of Freedom

The purpose of the 6DOF-block is to track the position and orientation of the aircraft through three dimensional space. The mass and moments of inertia of the aircraft are entered as parameters. The aircraft's orientation is stored as quaternions to ensure that no discontinuities occur.

D.3 Sensor Model

To minimise time spent on test flights the inputs received by the control system in simulation must match real-world inputs as closely as possible. The sensor model uses the perfect state vector from the 6DOF-block to simulate values measured by real world sensors. Position is converted to longitude, latitude and height to mimic the output of a GPS. The outputs of accelerometers, rate gyrometers and magnetometers are replicated in a similar way. All outside factors have to be considered when generating the sensor measurements:

- Different sensors can be sampled at different rates, some as high as 1 KHz (Accelerometers), other as low as 4Hz (GPS).
- Several filters are implemented on the avionics to condition the measured signals. All analogue signals received on the CAN bus are filtered low pass filtered with a bandwidth of 8Hz by a second order Butterworth filter is implanted in software. The behaviour of this filter is mimicked in the simulation.
- Each sensor has unique noise characteristics. Manufacturer's datasheets and actual measured values are used to estimate the magnitude and bandwidth of the noise on each sensor.

D.4 Extended Kalman Filter

As explained before the optimal control calculated by LQR takes the form of full state feedback. Unfortunately, not all states can be measured directly. An extended Kalman filter is an optimal estimator that uses measured quantities and known inputs to approximate the states of the system. The design of the EKF is detailed in [5]. In order for the estimation to be optimal the variances of the noise present on the measurements is required. The variances are analogous to the weighting matrices LQR and are given in Table D.1

D.5 Servo Model

When a command is given to a servo-motor controlling an actuation surface it does not change position immediately. Even a small delay in actuator de-

Symbol	Standard Deviation	Units
$\sigma_{a_x}, \sigma_{a_y}, \sigma_{a_z}$	0.1414	m.s^{-2}
$\sigma_p, \sigma_q, \sigma_r$	27.925×10^{-3}	rad.s^{-1}
$\sigma_\lambda, \sigma_\phi$	6.2832×10^{-7}	rad
σ_h	4	m
$\sigma_{V_N}, \sigma_{V_E}, \sigma_{V_D}$	0.5	m.s^{-1}
$\sigma_{m_x}, \sigma_{m_y}, \sigma_{m_z}$	2×10^{-2}	Gauss

Table D.1: Standard Deviations of Noise on Sensors

flection can result in unexpected behaviour, especially in a high bandwidth system. In [5] the servo model was created as first order transfer functions. To better simulate the response of a real servo-model a slew-rate limiter is implemented by examining the manufacturer's specification.

Appendix E

Additional Figures

E.1 TSD three and four second transitions

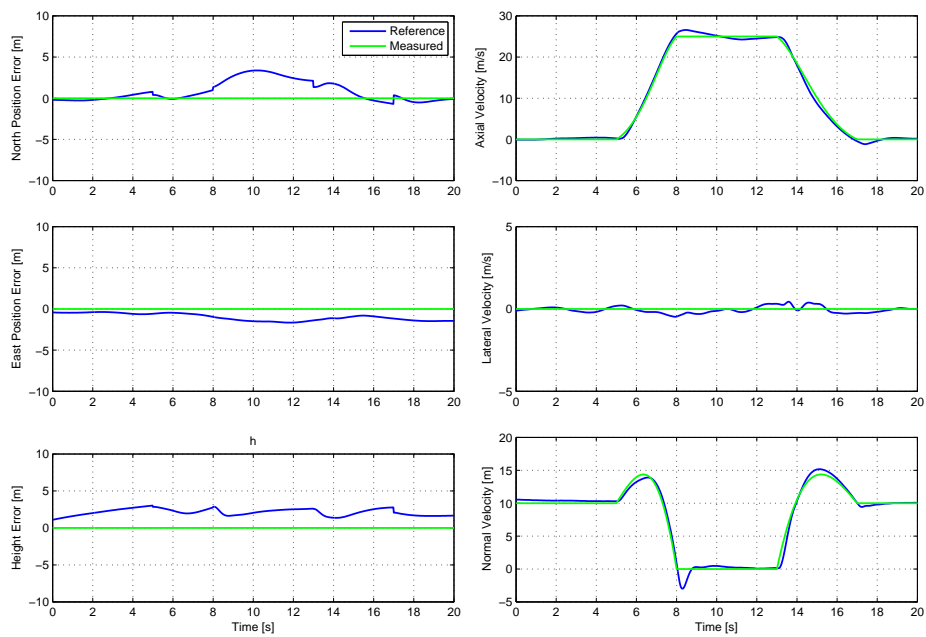


Figure E.1: Decoupled System Response

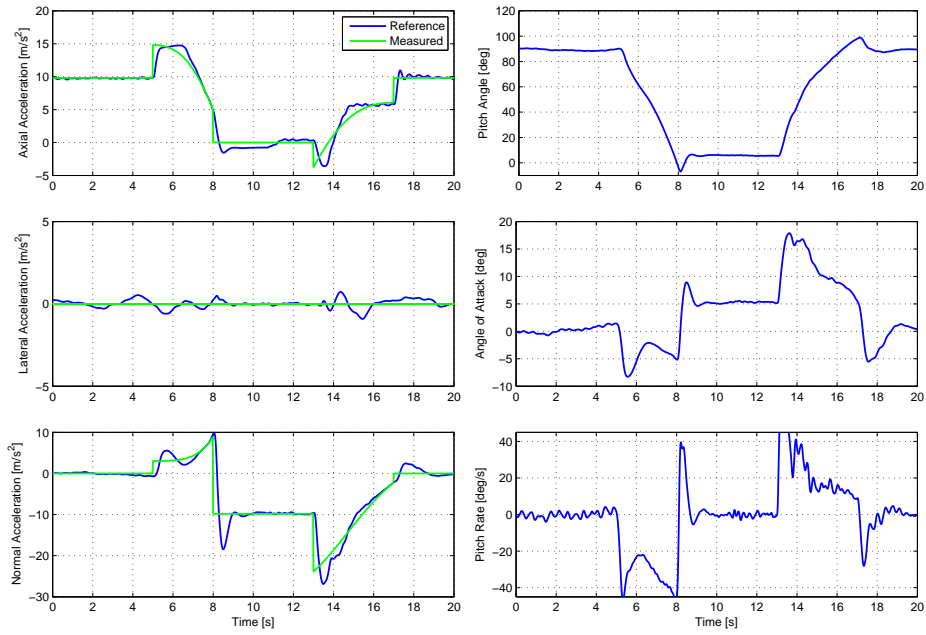


Figure E.2: Decoupled System Response

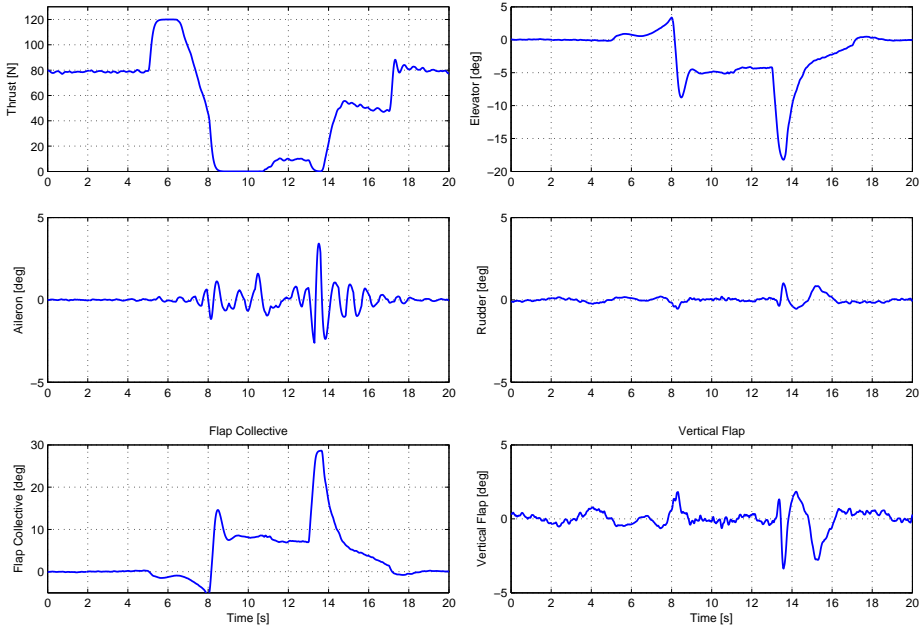


Figure E.3: Decoupled System Response

E.2 Sensitivity Analysis Loci

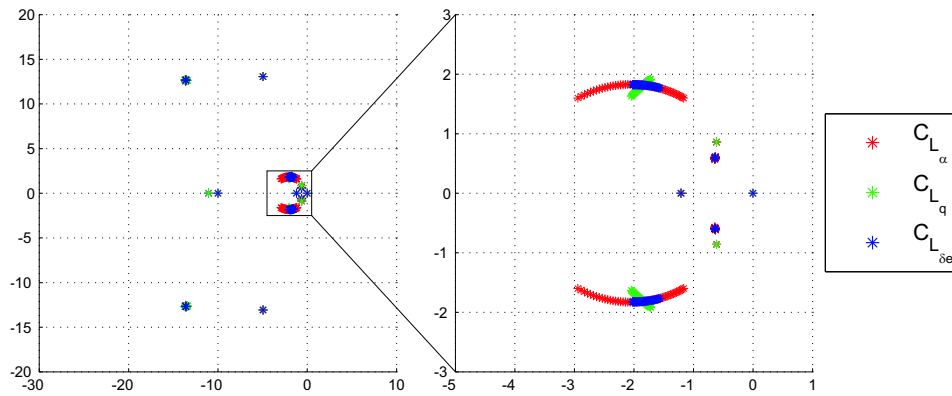


Figure E.4: LQR Sensitivity to Lift parameters

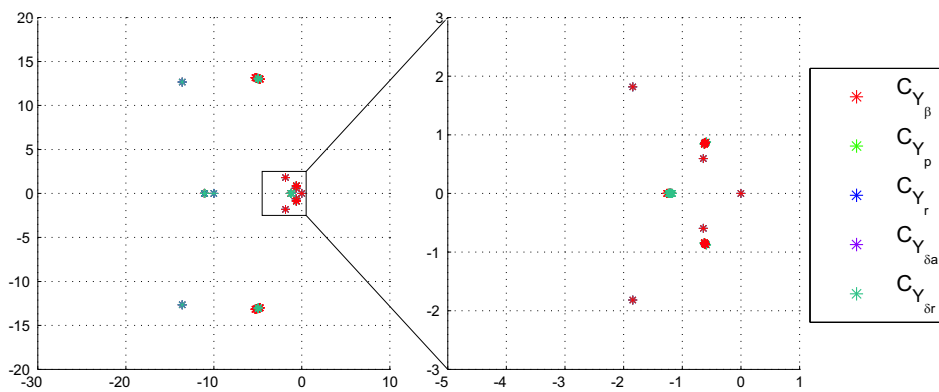


Figure E.5: LQR Sensitivity to Side Force parameters

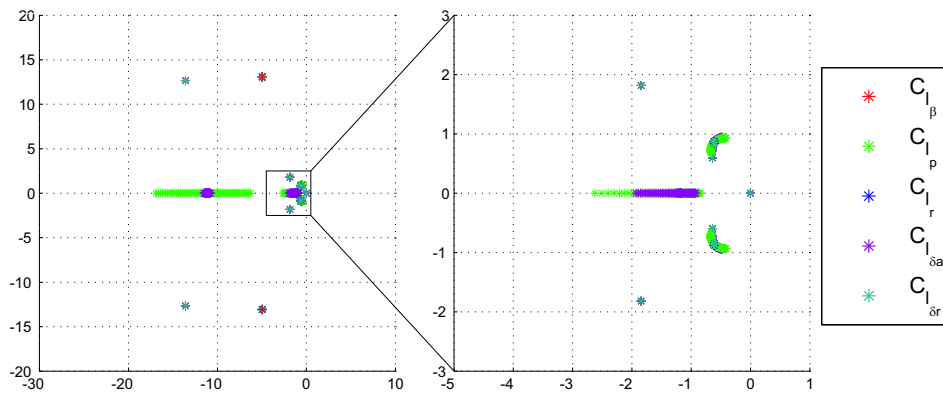


Figure E.6: LQR Sensitivity to Roll Moment parameters

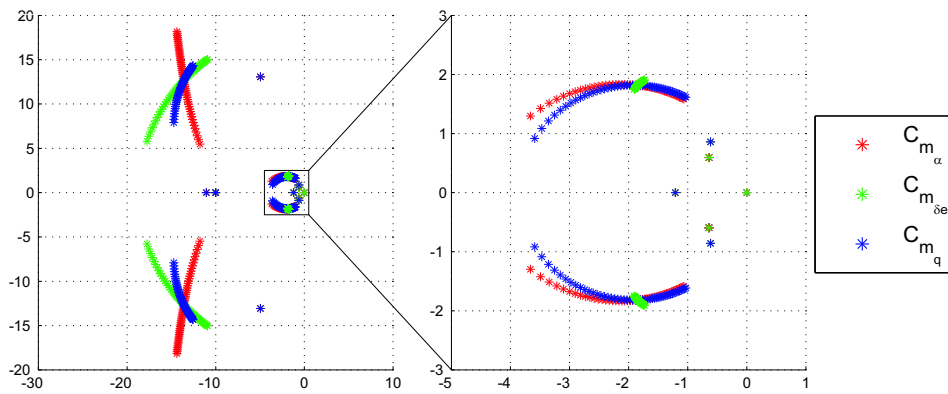


Figure E.7: LQR Sensitivity to Pitch Moment parameters

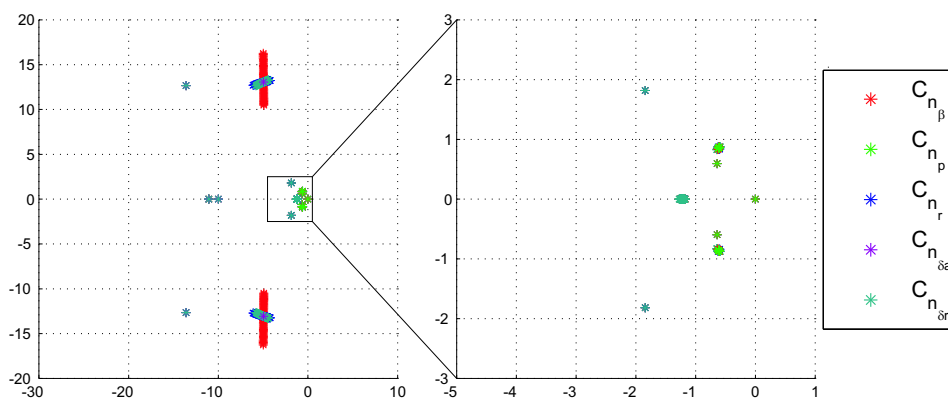


Figure E.8: LQR Sensitivity to Yaw Moment parameters

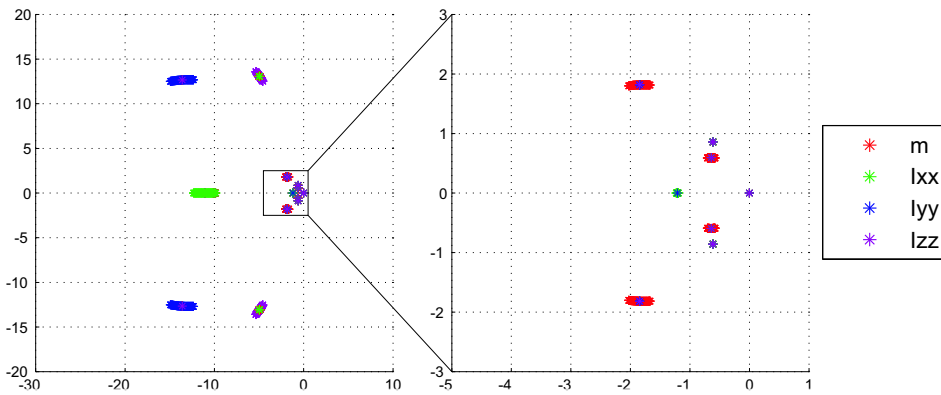


Figure E.9: LQR Sensitivity to Physical parameters

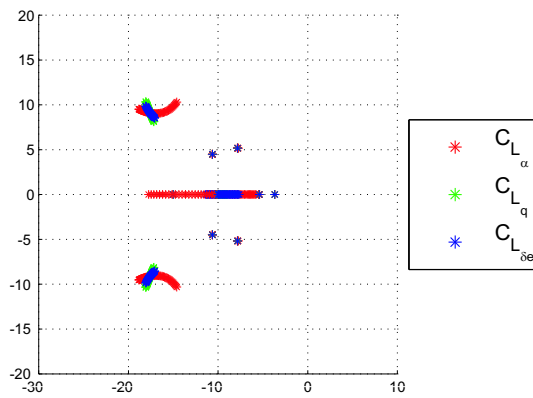


Figure E.10: TSD Sensitivity to Lift parameters

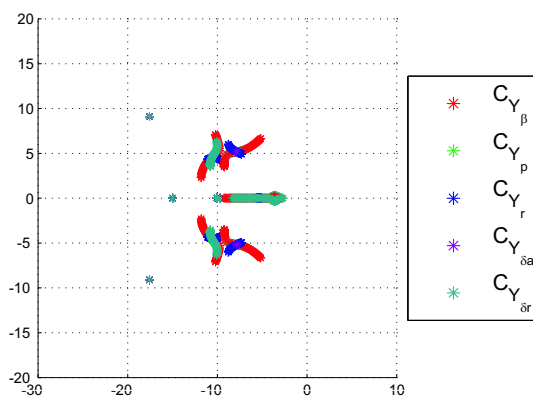


Figure E.11: TSD Sensitivity to Side Force parameters

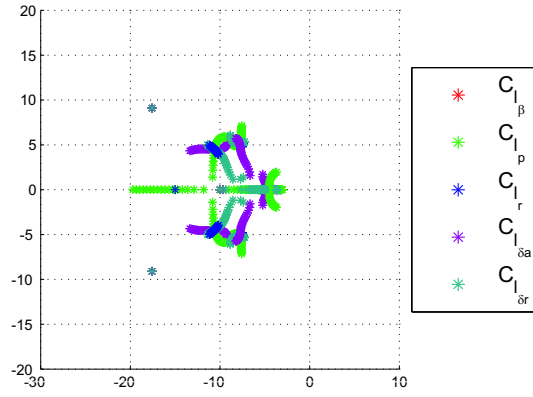


Figure E.12: TSD Sensitivity to Roll Moment parameters

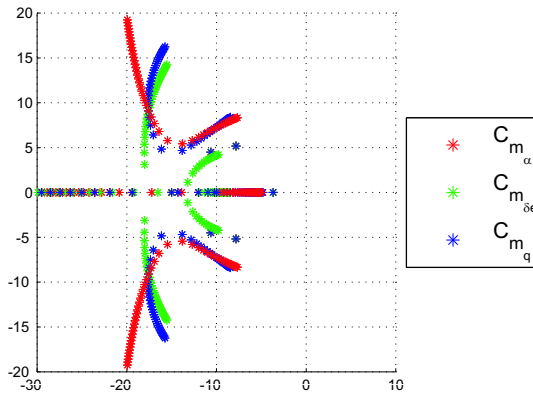


Figure E.13: TSD Sensitivity to Pitch Moment parameters

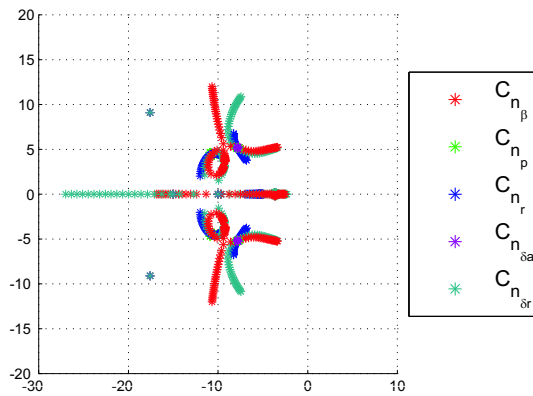


Figure E.14: TSD Sensitivity to Yaw Moment parameters

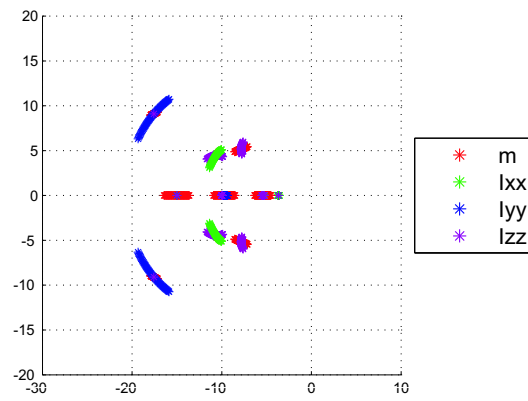


Figure E.15: TSD Sensitivity to Physical parameters

Bibliography

- [1] I.K. Peddle, *Autonomous Flight of a Model Aircraft*, University of Stellenbosch, 2004.
- [2] I.K. Peddle, *Acceleration Based Manoeuvre Flight Control System for Unmanned Aerial Vehicles*, University of Stellenbosch, 2007.
- [3] I.K. Peddle, T. Jones, *Acceleration Based 3D Maneuver Flight Control System for UAVs: Strategy and Longitudinal Design*, Submitted to Automatica, January 2008
- [4] J. Venter, *Development of an experimental Tilt-Wing VTOL Unmanned Aerial Vehicle*, University of Stellenbosch, 2005.
- [5] W. J. Hough, *Autonomous Aerobatic Flight of a Fixed Wing Unmanned Aerial Vehicle*, University of Stellenbosch 2007.
- [6] J. Treurnicht, *Axis Transformations*, University of Stellenbosch, 2003.
- [7] J. Diebel, *Representing Attitude: Euler Angles, Unit Quaternions and Rotation Vector*, Stanford University, 2006.
- [8] M.C. Koen *Modelling and Simulation of an RPV for Flight Control System Design Purposes*, University of Pretoria, 2006.
- [9] B.W. McCormick, *Aerodynamics of V/STOL Flight*, The Pennsylvania State University, 1967.
- [10] James A. Franklin, *Dynamics, Control and Flying Qualities of V/STOL Aircraft*. AIAA Educational Series, 2002.
- [11] K.J. Astrom, T. Haggund, *PID Controllers*, Instrument Society of America
- [12] M. Drela, *AVL 3.14 User Primer*, MIT Aero & Astro, 2004.
- [13] G.F. Franklin, J.D. Powell, A. Emama-Naeini, *Feedback Control of Dynamic Systems*. Addison-Wesely Publishing Company, 1994.

- [14] J.H. Blakelock, *Automatic Control of Aircraft and Missiles*, Wiley-Interscience, 1991.
- [15] Victor M. Becerra, *Advanced Nonlinear Control - Course Notes*. The University of Reading 2005.
- [16] M.V. Cook, *Flight Dynamics Principles*, Elsevier Butterworth-Heinemann, 1997.
- [17] M. Pachter, P.R. Chandler, L. Smith, *Maneuvering Flight Control*, Journal of Guidance, Control and Dynamics, 1998
- [18] J.D. Faires, R. Burden, *Numerical Analysis*
- [19] NASA History web page,
<http://history.nasa.gov/SP-367/appendc.htm>. 2006.
- [20] R.S. Shevell, *Fundamentals of Flight*. Prentice-Hall, 1983.