UNIVERSITEIT•STELLENBOSCH•UNIVERSITY

jou kennisvennoot • your knowledge partner

Autonomous Aerobatic Flight of a Fixed Wing Unmanned Aerial Vehicle

by

Willem J. Hough

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

Master of Science in Engineering
(Electronic Engineering with Computer Science)

Department of Electric & Electronic Engineering
University of Stellenbosch
Private Bag X1, 7602 Matieland, South Africa

Supervisor: Prof Thomas Jones
Co-Supervisor: Mr Iain K. Peddle

March 2007

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: .....

Date: .....

Abstract

This thesis relates to the successful development of a flight control system to perform a range of aerobatic manoeuvres autonomously. The project is the first to try to extend the flight control capabilities of the Computer and Control group at the University of Stellenbosch.

A simplified mathematical aircraft model is developed which encapsulates the important dynamic characteristics of the airframe. It is demonstrated how computational fluid dynamics software can be used to calculate the stability and control derivatives of a conventional airframe.

A vehicle independent kinematic state estimator is presented and used to obtain the complete aircraft state vector. The estimator makes use of extended Kalman filter theory to combine a series of low quality sensor measurements in an optimal manner. A model predictive control strategy is then used to regulate the aircraft about arbitrary, time variant trajectories. The controller's architecture is not in any way specific to the aerobatic manoeuvres demonstrated in this project.

The avionics and ground station used for the implementation of the estimator and control algorithms are presented. The development of a hardware in the loop simulator is discussed and used to verify the correct implementation of the respective algorithms. Finally, practical results from two days of flight tests are presented.

Uittreksel

Hierdie tesis handel oor die suksesvolle ontwikkeling van 'n vlugbeheerstelsel om 'n verskeidenheid akrobatiese manoeuvres automaties uit te voer. Dit is die eerste projek wat die vlugbeheer tegnieke van die Rekenaar en Beheer groep by the Universiteit van Stellenbosch probeer uitbrei.

'n Vereenvoudigde wiskundige vliegtuig model is ontwikkel wat die belangrikste dinamiese karakteristieke van die lugraam omsluit. Die proses demonstreer hoe vloeidinamika sagteware gebruik kan word om die stabiliteits- en beheerafgeleides van 'n konvensionele lugraam te bereken.

'n Voertuigonafhanklike kinematiese toestandsafskatter word voorgedra en gebruik om die vliegtuig se hele toestandsvektor te bereken. Die afskatter maak gebruik van uitgebreide Kalman filter teorie om 'n reeks lae kwaliteit sensormetings bymekaar te voeg in 'n optimale manier. 'n Model voorspellende beheerstrategie word dan gebruik om die vliegtuig te reguleer om arbitrêre, tyd-variante bane. Die beheerderargitektuur is nie op enige manier spesifiek tot die akrobatiese manoeuvres gedemonstreer in hierdie projek.

Die avionika en grondstasie wat gebruik is vir die implementering van die afskatter- en beheeralgoritmes word voorgedra. Die ontwikkeling van 'n hardeware in die lus simulator word bespreek en gebruik om die korrekte implementering van die verskeie algoritmes te bevestig. Laastens word praktiese resultate van twee dae se vlugtoetse voorgedra.

Acknowledgements

I would like to express my sincere gratitude to the following people and organisations who have contributed to making a project of this magnitude and complexity possible:

- Prof. T. Jones for guiding me into the right direction and providing me with deeper insight on many topics.
- Mr. I.K. Peddle for helping me get to terms with numerous theoretical problems and his willingness to always lend a helping hand regardless of the problem at hand.
- Denel Aerospace Systems, ARMSKOR and ATE for funding UAV related work at the University of Stellenbosch.
- The National Research Foundation for helping fund the project during its second year.
- Jacob Venter for your valued input regarding RC problems and helping with the construction of the aircraft.
- Arno Barnard, Willie van Rooyen and Johan Arendse to name a few who provided technical advice and support during various stages of the project.
- Dr. Kas Hamman for flying the aircraft.
- All my friends in the ESL, especially Johan Bijker and Carlo van Schalkwyk, for helping me with practical tests and the many non-work related discussions.
- Laura Garbarino for her interest in my work and testing my ability to explain various engineering related topics.
- My mother and brother for your continuous belief in me and your support throughout the project.

Dedications

This thesis is dedicated to everyone who has contributed towards my education.

Contents

List of Figures

List of Tables

Nomenclature

Greek Letters:

α	Angle of attack
β	Angle of side-slip
δ_a	Aileron deflection
δ_e	Elevator deflection
δ_r	Rudder deflection
$\mathbf{\Gamma}$	Discrete input matrix
λ	Latitude
μ	Rotation about Euler axis vector
$\boldsymbol{\omega}$	Inertial angular velocity vector
ω_p	Pade approximation pole/zero frequency
ϕ	Roll angle
$\mathbf{\Phi}$	Discrete dynamics matrix
ψ	Yaw angle
ρ	Air density
σ	Standard deviation
τ_E	Engine time constant
θ	Pitch angle
φ	Longitude
ϑ	Integrated pitch rate through loop manoeuvre

Small Letters:

b	Wing span
$\boldsymbol{b_{16}}$	Servo mechanical offset vector
c	Mean aerodynamic chord

$\boldsymbol{c_8}$, $\boldsymbol{c_{16}}$	Servo timer offset matrices
e	Efficiency factor
$\boldsymbol{e}$	Euler axis vector
g	Gravitational acceleration
h	Height above round earth model radius
$\boldsymbol{i}$	Unit vector along X-axis
$\boldsymbol{j}$	Unit vector along Y-axis
$\boldsymbol{k}$	Unit vector along Z-axis
m	Mass
p	Roll rate
q	Pitch rate or Dynamic pressure
$\boldsymbol{q}$	Quaternion
r	Yaw rate
u	Axial velocity
v	Lateral velocity
$\boldsymbol{v}$	Measurement noise sequence
w	Normal velocity
$\boldsymbol{w}$	Process noise sequence or TRIAD vector

Capital Letters:

A	Aspect ratio
$\boldsymbol{A_8}$, $\boldsymbol{A_{16}}$	Servo timer gain matrices
$\boldsymbol{B}$	Constant earth magnetic field
C_D	Aerodynamic drag coefficient
C_l	Aerodynamic roll moment coefficient
C_L	Aerodynamic lift coefficient
C_m	Aerodynamic pitch moment coefficient
C_n	Aerodynamic yaw moment coefficient
C_x	Axial aerodynamic force coefficient
C_y	Lateral aerodynamic force coefficient
C_z	Normal aerodynamic force coefficient
E	East displacement
$\boldsymbol{F}$	Force vector
$\boldsymbol{F_k}$	Continuous linearised dynamics matrix
$\boldsymbol{G_k}$	Continuous linearised input matrix
$\boldsymbol{H}$	Output matrix

$\boldsymbol{I}$	Identity matrix
J	LQR Cost
J_x	Moment of inertia about roll axis
J_{xz}	Roll and yaw product of inertia
J_y	Moment of inertia about pitch axis
J_z	Moment of inertia about yaw axis
$\boldsymbol{K}$	LQR gain matrix
L	Roll moment or Lift vector
$\boldsymbol{L}$	Kalman filter gain matrix
M	Pitch moment
$\boldsymbol{M}$	Moment vector, Propagated states error covariance matrix or Discrete matrix Riccati variable
$\boldsymbol{M_{16}}$	Servo mixing matrix
N	Yaw moment or North displacement
$\boldsymbol{P}$	Best state estimate error covariance matrix
$\boldsymbol{Q}$	Process noise covariance matrix or LQR state and control weight matrices
$\boldsymbol{Q_c}$	Process noise PSD matrix
R	Round earth radius
R_L	Constant loop radius
$\boldsymbol{R}$	Measurement noise covariance matrix
$\boldsymbol{R_c}$	Measurement noise PSD matrix
S	Wing reference area
$\boldsymbol{S}$	Discrete matrix Riccati variable
$\boldsymbol{S_{16}}$	Servo mechanical gain matrix
T	Engine thrust magnitude
T_c	Commanded engine thrust
T_s	Discrete sampling time
$\boldsymbol{T}$	Transformation matrix
V	Airspeed or Ground speed
$\boldsymbol{V}$	Velocity vector
W	West displacement
X	Axial force
Y	Lateral force
Z	Normal force

Subscripts:

A	Aerodynamic force and moment
B	Body axis
d	Estimator delayed state
D	Down indicator
E	Engine force and moment or East indicator
G	Gravitational force and moment
I	Inertial axis
N	North indicator
n	Nominal state and control vector
0	Trim or zero/offset value
p	State and control perturbations
S	Stability body axis

Superscripts:

S	Stability body axis

Acronyms:

2D	Two Dimensional
3D	Three Dimensional
ADC	Analogue to Digital Converter
AMR	Anisotropic Magneto Resistive
AVL	Athena Vortex Lattice
CAN	Controller Area Network
CG	Centre of gravity
CS	Checksum
DAC	Digital to Analogue Converter
DCM	Direction Cosine Matrix
DOF	Degree of Freedom
ECEF	Earth Centred Earth Fixed
EKF	Extended Kalman Filter
ESL	Electronic Systems Laboratory

FTDI	Future Technology Devices International
GPS	Global Positioning System
GUI	Graphical User Interface
HIL	Hardware In the Loop
HILS	Hardware In the Loop Simulator/Simulation
IGRF	International Geomagnetic Reference Field
IMU	Inertial Measurement Unit
ISA	Industry Standard Architecture
LAN	Local Area Network
LQR	Linear Quadratic Regulator
LTV	Linear Time Variant
MIMO	Multi Input Multi Output
MIT	Massachusetts Institute of Technology
MPC	Model Predictive Controller
NASA	National Aeronautics and Space Administration
NED	North-East-Down
NMEA	National Marine Electronics Association
OBC	Onboard Computer
OEM	Original Equipment Manufacturer
PC	Personal Computer
PCB	Printed Circuit Board
PSD	Power Spectral Density
PWM	Pulse Width Modulation
RC	Remote Controlled
RF	Radio Frequency
RHC	Receding-Horizon Controller
RHCP	Right-Hand Circular Polarized
RMS	Root Mean Square
RPV	Remote Piloted Vehicle
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver and Transmitter
UAV	Unmanned Aerial Vehicle
UCA	User Control Application
UDP	User Datagram Protocol
USB	Universal Serial Bus
ZOH	Zero Order Hold

Chapter 1

Introduction and Overview

1.1 Background

Unmanned Aerial Vehicles (UAVs) are defined as vehicles that use aerodynamic and propulsion forces to sustain predefined flight paths without the help of a human pilot [33]. The Computer and Control Systems Group, a subdivision of the Electric and Electronic Engineering Department at the University of Stellenbosch, does active research in the field of UAVs. Since its establishment in January 2001, the group has experienced rapid growth in this field of study. Recently completed projects include the stabilisation and control of an electrically powered helicopter [37], complete autonomous waypoint navigation of a conventional fixed wing aircraft [35], as well as the development of rotary [38] and tilt-wing [39] demonstrator vehicles. Being motivated by previous successes, the desire arose to push the state of the art in aircraft flight control. In order to achieve this goal, a project was defined which required a fixed wing aircraft to perform a range of aerobatic manoeuvres autonomously.

Aerobatic flight can be defined as a series of spectacular manoeuvres which requires an aircraft to operate in abnormal, difficult, and most often dangerous, flight conditions. This type of flight also has military applications in that certain manoeuvres allow fighter pilots to obtain a position of advantage with respect to the enemy [50].

1.2 Task Description and Strategy

This project requires the design of a flight control system to automatically perform three inherently different aerobatic manoeuvres. The first manoeuvre is an aileron roll as demonstrated by Figure 1.1. The manoeuvre was chosen due to its relative simplicity and was considered an

adequate starting point before more complex flight paths are attempted.

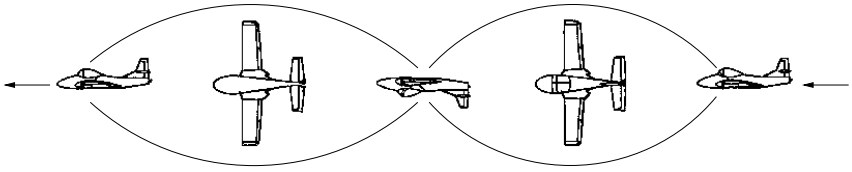


Figure 1.1: Picture of Aileron Roll Manoeuvre

The second manoeuvre is a constant radius loop as demonstrated by Figure 1.2. This well known manoeuvre was chosen due to the spectacular show it provides when performed with great accuracy.

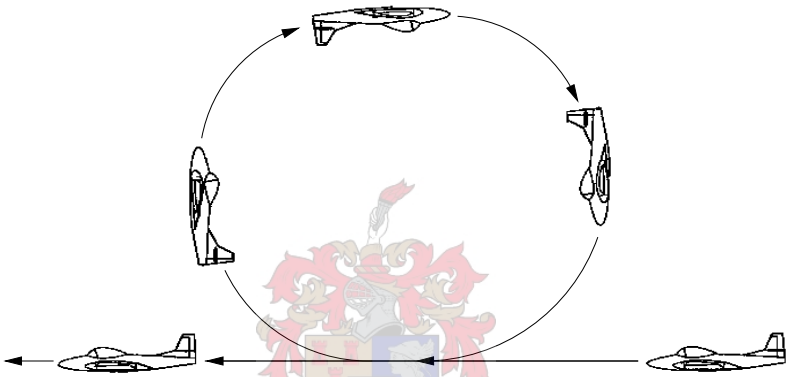


Figure 1.2: Picture of Loop Manoeuvre

The third and last manoeuvre is the Immelmann, shown in Figure 1.3, and is a combination of a loop and aileron roll. After having performed half a loop, a 180° roll is performed resulting in the aircraft flying in the opposite direction. The Immelmann is also known as a backwards Split-S, a manoeuvre used in warfare to disengage from a combat scenario.

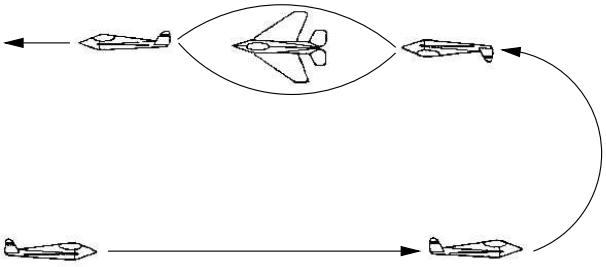


Figure 1.3: Picture of Immelmann Manoeuvre

Performing these manoeuvres autonomously presents diverse and difficult challenges. Previously completed projects only required the respective airframes to operate in conventional flight

conditions. This has the advantage that the non-linear aircraft dynamics can be simplified by linearisation about some equilibrium point. However, when no fixed equilibrium point exists, other methods have to be investigated to solve the problem at hand.

Various methods exist which are capable of dealing with highly non-linear systems [29]. The main disadvantage of using these algorithms is the fact that they often require large amounts of processing power. When resorting to such control methods, the practicality of the algorithms needs to be considered by evaluating their ability to operate in real-time environments. Disturbance rejection is another important factor that needs consideration when choosing a particular control strategy. Finally, the decision is complicated even more by having to deal with inaccurate vehicle models, and the use of low cost sensors.

Taking all these factors into account, a receding horizon, model predictive control strategy was adopted to perform the various manoeuvres. The process involves regulating the aircraft about a predefined, time-varying state trajectory. A non-linear, full state estimator was implemented since the control strategy requires all aircraft states for feedback. The estimation strategy assumes the aircraft is operating in little or no wind conditions. This assumption is justified by the project being the first step in the research towards complete autonomous aerobatic flight.

1.3 Thesis Outline

The thesis covers the autopilot design from first principles up to flight tests. Chapter 2 presents a non-linear aircraft model which lays the foundation for the simulation platform to be developed later. The method used to calculate the airframe's aerodynamic model extends the research group's modelling capabilities, and in doing so allows confidence to be instilled in the developed model. In Chapter 3, a vehicle independent kinematic state estimator is discussed. Chapter 4 introduces the adopted control strategy used to perform the specified aerobatic manoeuvres. After having discussed the theory and its application, the controller's performance is evaluated by means of non-linear simulation. The avionics and ground station are the topics of Chapter 5. Chapter 6 focusses on the development of a hardware in the loop simulator. This tool greatly extends the group's simulation capabilities in that the correct implementation of the control algorithms can be verified before practical flight tests. Lastly, flight test results are provided in Chapter 7.

Chapter 2

Aircraft Model

The aim of this chapter is to develop a mathematical aircraft model to serve as a simulation platform to be used throughout the project. The derivation is mostly based on the theory discussed in [25]. This approach was adopted since it delivers a simplified six degree of freedom (6-DOF) non-linear model, whilst retaining the important dynamic characteristics of the airframe.

2.1 Axis System Definitions

To aid in modelling the aircraft as a rigid body, two axis systems need to be defined: an earth-fixed reference frame and a body-fixed frame. With these axis systems defined, the motion of the aircraft is described by the translation and rotation of the two axis systems with respect to each other.

2.1.1 Inertial Reference Frame

Newton's laws only apply in an inertial reference frame. Before the inertial reference frame can be defined, the earth's surface is assumed flat and non-rotating. Considering the distance and duration of flight of the aircraft used in this project, and the fact that the angular velocities of the airframe will be much greater than the earth's angular rotation, these are reasonable assumptions to make. Figure 2.1 defines the inertial reference frame used during the development of the aircraft model.

Given some convenient point on the earth's surface, the inertial reference frame is defined as follows: The OX_I-axis points to the North pole and is tangential to the earth's surface. The OY_I-axis is also tangential to the earth's surface and points East. The OZ_I-axis is perpendicular

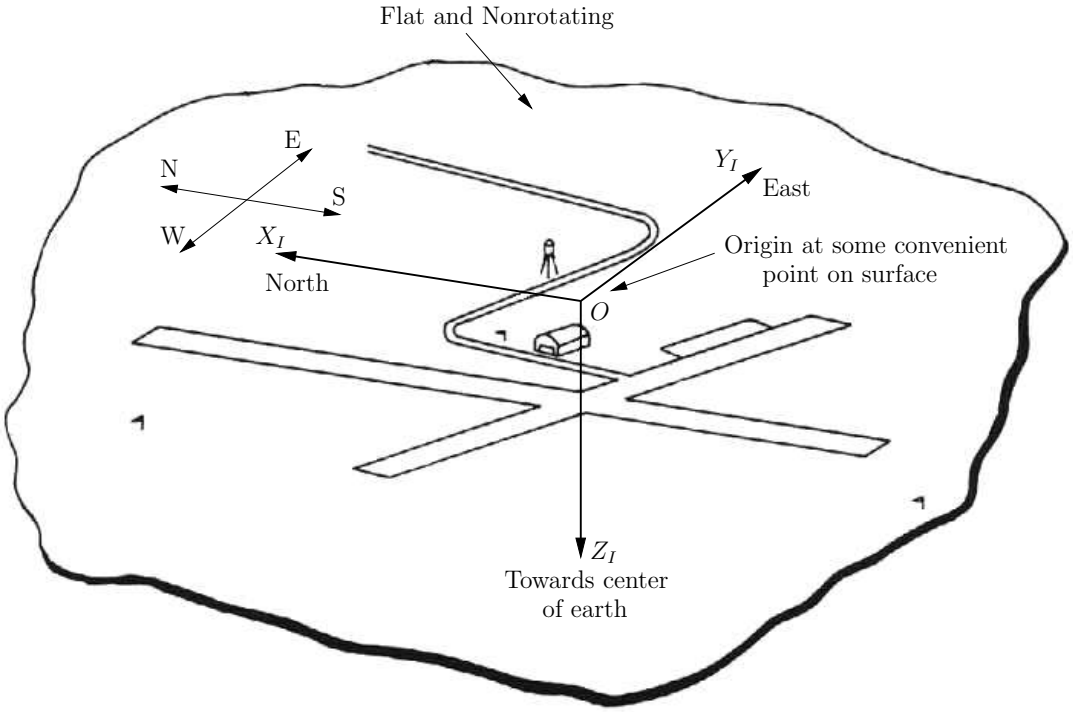


Figure 2.1: Inertial Axis System Definition [46]

to both the OX_I and OY_I axes and points towards the centre of the earth.

2.1.2 Body Reference Frame

The body axis system's origin is the aircraft's centre of mass and is fixed to the airframe and subsequently rotates and translates along with it. This project will make use of reference line body axes which are defined as follows: The OX_B-axis (longitudinal) is in the aircraft's plane of symmetry, parallel to some longitudinal reference line, and points forward in the direction of the nose. The OY_B-axis (lateral) is perpendicular to the OX_B-axis and points along the starboard wing. The OZ_B-axis (vertical) is perpendicular to the X_BY_B-plane and points downwards. Figure 2.2 shows the basic definition of the body axis system, as well as the positive directions of the roll, pitch and yaw motions of the aircraft.

Having defined the respective reference frames, the aircraft linear and angular velocity vectors, as well as the force and moments vectors, can be defined. These are listed below,

Linear Velocity The velocity of the aircraft relative to inertial space but coordinated in the body axis frame is given by,

$$\boldsymbol{V_B} = u\,\boldsymbol{i_B} + v\,\boldsymbol{j_B} + w\,\boldsymbol{k_B} \tag{2.1}$$

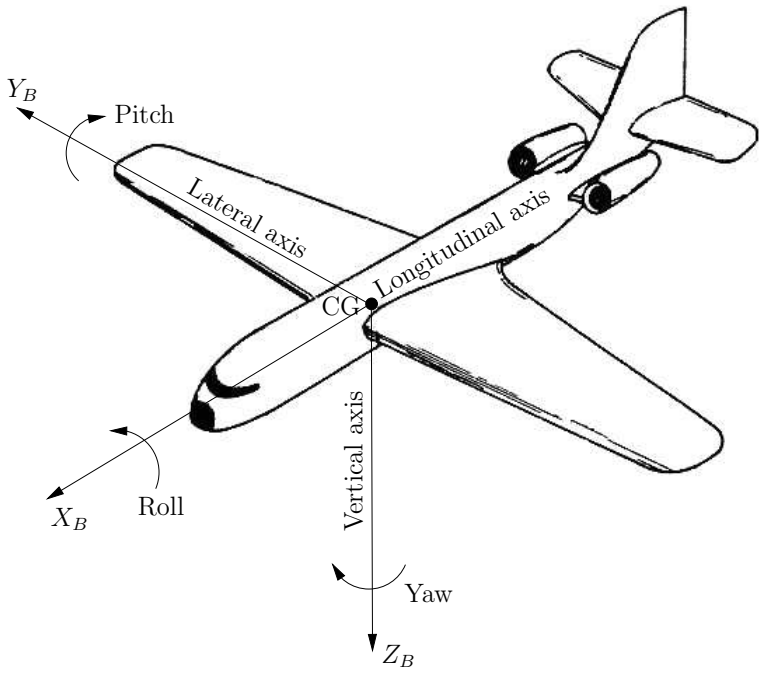


Figure 2.2: Aircraft Body Axis System Definition [46]

Angular Velocity The angular velocity of the body axis system with respect to the inertial axis system is given by,

$$\boldsymbol{\omega_B} = p\,\boldsymbol{i_B} + q\,\boldsymbol{j_B} + r\,\boldsymbol{k_B} \tag{2.2}$$

Force Vector The force vector acting on the aircraft is given by,

$$\boldsymbol{F_B} = X\,\boldsymbol{i_B} + Y\,\boldsymbol{j_B} + Z\,\boldsymbol{k_B} \tag{2.3}$$

Moment Vector The moment vector acting on the aircraft is given by,

$$\boldsymbol{M_B} = L\,\boldsymbol{i_B} + M\,\boldsymbol{j_B} + N\,\boldsymbol{k_B} \tag{2.4}$$

In equations (2.1) to (2.4), $\boldsymbol{i_B}$, $\boldsymbol{j_B}$ and $\boldsymbol{k_B}$ are unit vectors defined along the OX_B, OY_B and OZ_B axes respectively. It should also be noted that the linear and angular velocity components portray absolute values and not small perturbations as were used in [35]. The velocity vector, $\boldsymbol{V_B}$, can also be rewritten in polar form which makes more intuitive sense than the Cartesian representation used thus far, and is useful for quantising aerodynamic forces and moments. The polar form is explained visually in Figure 2.3 and is defined as,

$$V = \sqrt{u^2 + v^2 + w^2} \tag{2.5}$$

$$\alpha = \arctan\left(\frac{w}{u}\right) \tag{2.6}$$

$$\beta = \arcsin\left(\frac{v}{V}\right) \tag{2.7}$$

where V is the airspeed when it is assumed that the velocity of the wind with respect to the inertial reference frame is zero, and α and β are the angles of attack and side-slip respectively.

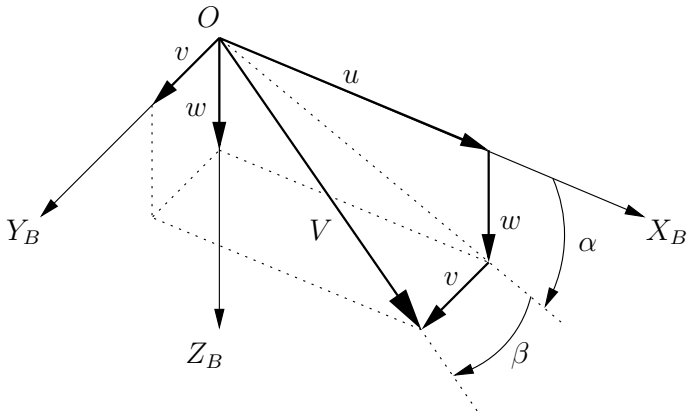


Figure 2.3: Linear Velocity Polar Form

Inspecting the geometry in Figure 2.3, the components of the linear velocity vector can be extracted from the polar form as follows,

$$u = V \cos\alpha \cos\beta \tag{2.8}$$

$$v = V \sin\beta \tag{2.9}$$

$$w = V \sin\alpha \cos\beta \tag{2.10}$$

Summarised below are the conventional aerodynamic control surface deflection variables as well as the sign conventions associated with each surface.

- δ_a - Aileron deflection where a positive deflection causes a negative roll moment about the OX_B-axis.
- δ_e - Elevator deflection where a positive deflection causes a negative pitch moment about the OY_B-axis.
- δ_r - Rudder deflection where a positive deflection causes a negative yaw moment about the OZ_B-axis.

2.2 Equations of Motion

Stated below are the non-linear, coupled differential equations describing the aircraft's motion in six degrees of freedom [35],

$$X = m\left(\dot{u} - vr + wq\right) \tag{2.11}$$

$$Y = m\left(\dot{v} - wp + ur\right) \tag{2.12}$$

$$Z = m\left(\dot{w} - uq + vp\right) \tag{2.13}$$

$$L = \dot{p}J_x + qr\left(J_z - J_y\right) \tag{2.14}$$

$$M = \dot{q}J_y + pr\left(J_x - J_z\right) \tag{2.15}$$

$$N = \dot{r}J_z + pq\left(J_y - J_x\right) \tag{2.16}$$

where m is the mass of the aircraft, and J_x, J_y and J_z are the moments of inertia about the OX_B, OY_B and OZ_B axes respectively. In equations (2.11) to (2.16) the assumptions are made that the aircraft is symmetrical about the X_BZ_B-plane and that the product of inertia between the OX_B and OZ_B axes, J_{xz}, is negligibly small. For a derivation of the equations of motion from first principles see [12].

2.3 Attitude Descriptions

Attitude determination involves methods to describe the orientation of the body axis frame with respect to the inertial axis frame. The two most common techniques are Euler angles and the quaternion which are discussed in the following sections.

2.3.1 Euler Angles

Euler angles arguably provide the simplest attitude definition and consist of a set of three angles and an order of rotation. The advantage of using Euler angles is the intuitive sense they possess, and subsequently are well suited for analytical work. The Euler 3-2-1 method [4] was adopted in this project and is discussed briefly.

Consider Figure 2.4 and assume the inertial and body axis systems are aligned. Rotate the body reference frame about the OZ_I-axis in a clockwise direction through the heading angle, ψ, to a displaced body axis system, $OX_1Y_1Z_1$. Next, rotate the body reference frame about the OY_1-axis in a clockwise direction through the pitch angle, θ, to a displaced body axis system, $OX_2Y_2Z_2$. Lastly, the body reference frame is rotated clockwise about the OX_2-axis through the roll angle, ϕ, to the final position, $OX_3Y_3Z_3$. This axis system now coincides with the aircraft's

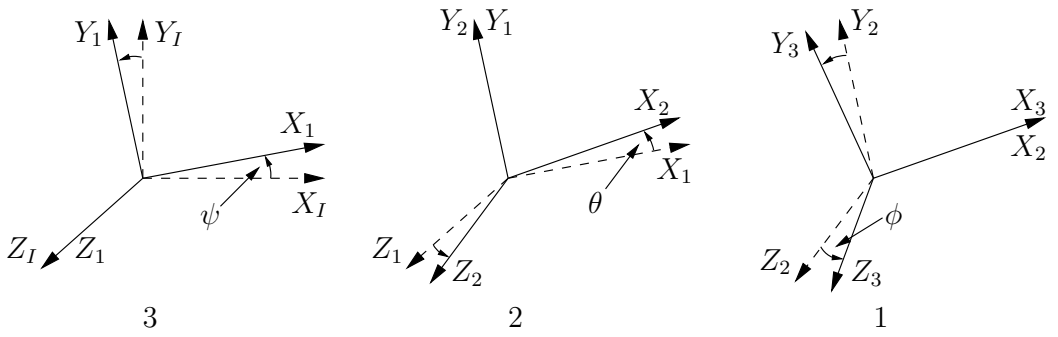


Figure 2.4: Euler 3-2-1 Rotations

body reference frame and the three Euler angles (ψ, θ and ϕ) and the order of rotation (3-2-1) completely define the attitude.

Transformation Matrix

Given a vector $\boldsymbol{V_I}$, coordinated in the inertial axis system, and the three Euler angles, the aim is to find a transformation matrix that will coordinate $\boldsymbol{V_I}$ in the body axis system to obtain $\boldsymbol{V_B}$. The Direction Cosine Matrix (DCM) provides such a relationship which is defined as,

$$\boldsymbol{V_B} = \boldsymbol{T}(\phi, \theta, \psi)\, \boldsymbol{V_I} \tag{2.17}$$

where $\boldsymbol{T}$ is the DCM in terms of Euler angles and is defined as [4],

$$\boldsymbol{T}(\phi, \theta, \psi) = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix} \tag{2.18}$$

To coordinate vectors in the inertial reference frame the inverse DCM needs to be calculated. It can be shown that $\boldsymbol{T}$ is orthogonal and thus its inverse is its transpose,

$$\boldsymbol{V_I} = \boldsymbol{T}^{-1}(\phi, \theta, \psi)\, \boldsymbol{V_B} = \boldsymbol{T}^{T}(\phi, \theta, \psi)\, \boldsymbol{V_B} \tag{2.19}$$

It is important to realise that the transformation matrix does not change the magnitude or direction of vector $\boldsymbol{V}$, it only provides a way to coordinate vectors in the different reference frames.

Euler Angle Differential Equations

Now that a method exists to coordinate vectors in different axis systems, it is required to be able to calculate the three Euler angles given the aircraft's body angular velocities. Such a

relationship is given by [4],

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.20}$$

Evaluation of equation (2.20) reveals that it suffers from singularities when the pitch angle is set to $\pm 90^\circ$. Since the focus of this project is to perform aerobatic manoeuvres, pitching through $\pm 90^\circ$ is a definite possibility. This provides the basis for using Euler symmetric parameters, more commonly known as quaternions, as a means to represent the aircraft's attitude.

2.3.2 Quaternion

The working behind the quaternion can be explained by Euler's theorem which states that two reference frames can be made to coincide by rotating one of them through a certain angle about a certain vector [5]. Now, define a unit vector $\boldsymbol{e}$ coordinated in the inertial reference frame as follows,

$$\boldsymbol{e} = \begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix} \tag{2.21}$$

where e_x, e_y and e_z are the components of the unit vector along the OX_I, OY_I and OZ_I axes respectively, and the rotation angle is given by μ. According to [4], the quaternion parameters are defined in vector form as follows,

$$\boldsymbol{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} e_x \sin\left(\frac{\mu}{2}\right) \\ e_y \sin\left(\frac{\mu}{2}\right) \\ e_z \sin\left(\frac{\mu}{2}\right) \\ \cos\left(\frac{\mu}{2}\right) \end{bmatrix} \tag{2.22}$$

In equation (2.22), the first three parameters primarily contain rotation vector information, whilst the fourth parameter contains only rotation angle information. Another representation commonly used introduces the rotation angle information first in the form of the parameter q_0. The vector $\boldsymbol{q}$ now completely defines the attitude of the aircraft. The norm of the quaternion is an important property and should hold at all times. Remembering that $\boldsymbol{e}$ is a unit vector, the quaternion norm is calculated as,

$$|\boldsymbol{q}| = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2} = 1 \tag{2.23}$$

Transformation Matrix

The transformation matrix in terms of the quaternion is given by equation (2.24). For a theoretical derivation of the matrix from first principles see [5].

$$\boldsymbol{T}(\boldsymbol{q}) = \begin{bmatrix} q_4^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_4 q_3 + q_1 q_2) & 2(q_1 q_3 - q_4 q_2) \\ 2(q_1 q_2 - q_4 q_3) & q_4^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_4 q_1 + q_2 q_3) \\ 2(q_4 q_2 + q_1 q_3) & 2(q_2 q_3 - q_4 q_1) & q_4^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{2.24}$$

Again $\boldsymbol{T}$ is orthogonal and its inverse is merely its transpose. Thus,

$$\boldsymbol{T}^{-1}(\boldsymbol{q}) = \boldsymbol{T}^T(\boldsymbol{q}) \tag{2.25}$$

Quaternion Differential Equations

To derive the quaternion differential equations, define a time independent vector in the body reference frame and call it $\boldsymbol{V_B}$. The vector is coordinated in the inertial reference frame to obtain $\boldsymbol{V_I}$, and then its time derivative is calculated. The result is given below,

$$\dot{\boldsymbol{V}}_{\boldsymbol{I}} = \dot{\boldsymbol{T}}^T(\boldsymbol{q})\,\boldsymbol{V_B} \tag{2.26}$$

The equation of Coriolis [12] provides a relationship between the time derivatives of a vector coordinated in inertial and body axes. Applying the equation to $\boldsymbol{V_I}$ the following result is obtained,

$$\dot{\boldsymbol{V}}_{\boldsymbol{I}} = \dot{\boldsymbol{V}}_{\boldsymbol{B}} + \boldsymbol{\omega_B} \times \boldsymbol{V_I} = \boldsymbol{\omega_B} \times \boldsymbol{V_I} = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \boldsymbol{T}^T(\boldsymbol{q})\,\boldsymbol{V_B} \tag{2.27}$$

where the vector cross product operation has been written in terms of a matrix multiplication. Now, equations (2.26) and (2.27) are set equal and after some manipulation the following result is obtained,

$$\begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} = \dot{\boldsymbol{T}}(\boldsymbol{q})\,\boldsymbol{T}^T(\boldsymbol{q}) \tag{2.28}$$

Next, the time derivative of the quaternion transformation matrix is calculated and substituted into equation (2.28). Performing the required matrix multiplication and comparing certain matrix elements, equations have been derived which relate the rate of change of the quaternion

parameters to the aircraft's angular velocity. The result is stated below in matrix form,

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = 2 \begin{bmatrix} q_4 & q_3 & -q_2 & -q_1 \\ -q_3 & q_4 & q_1 & -q_2 \\ q_2 & -q_1 & q_4 & -q_3 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} \tag{2.29}$$

In order to make the time derivatives of the quaternion parameters the subject of equation (2.29), the time derivative of the quaternion norm is calculated and augmented into the equation to obtain,

$$\begin{bmatrix} p \\ q \\ r \\ 0 \end{bmatrix} = 2 \begin{bmatrix} q_4 & q_3 & -q_2 & -q_1 \\ -q_3 & q_4 & q_1 & -q_2 \\ q_2 & -q_1 & q_4 & -q_3 \\ q_1 & q_2 & q_3 & q_4 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} \tag{2.30}$$

The derivation is completed by calculating the inverse of the 4×4 matrix in equation (2.30). The final result is stated below,

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.31}$$

Appendix A provides some valuable relationships between Euler angles and the quaternion.

2.4 Forces and Moments

The following sections discuss the various forces and moments which will drive the equations of motion as a function of the aircraft's current state, as well as the control inputs. The forces and moments acting on the airframe are divided into three main groups: gravitational, engine and aerodynamic which are denoted by the subscripts G, E and A respectively.

2.4.1 Gravitational

The force acting on the aircraft due to gravitational acceleration is obtained by coordinating the weight vector in the body axis frame,

$$\begin{bmatrix} X_G \\ Y_G \\ Z_G \end{bmatrix} = \boldsymbol{T}(\boldsymbol{q}) \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \tag{2.32}$$

where m and g are the airframe mass and gravitational acceleration respectively. Since the gravitational force acts through the centre of gravity, the moment contributions are set to zero.

2.4.2 Engine

The aircraft used in this project is equipped with a rigid, single propeller methanol engine where it is assumed that the engine's primary thrust vector acts parallel to the OX_B-axis and through the centre of gravity. The model can easily be extended to allow for a thrust vector acting at a set angle from the OX_B-axis as was used in [35]. The moment contributions are again zero whilst the forces are given by,

$$\begin{bmatrix} X_E \\ Y_E \\ Z_E \end{bmatrix} = \begin{bmatrix} T \\ 0 \\ 0 \end{bmatrix} \tag{2.33}$$

where T is the magnitude of the thrust vector. The change in thrust with change in airspeed has been neglected for now, and should be kept in mind when the controller design is considered. Most combustion engines display a significant delay between commanded and output thrust. The engine model is completed by introducing first order dynamics to model the engine's response, and is given by,

$$\dot{T} = -\frac{1}{\tau_E} T + \frac{1}{\tau_E} T_c \tag{2.34}$$

where T_c is the commanded thrust and τ_E is the engine's dynamic time constant which can be determined by experiment.

2.4.3 Aerodynamic

The origin of the aerodynamic forces and moments acting on the aircraft are explained by making use of two laws of physics namely the continuity principle and Bernoulli's equation. This section does not include any in depth details about aerodynamic modelling. The aim is to state well known equations on how to quantify aerodynamic influences on a conventional airframe. For more detail see [2].

Aerodynamic Coefficients

According to [35], the aerodynamic forces and moments acting along and about the aircraft's body axes are given by,

$$X_A = \frac{1}{2}\rho V^2 S C_x \tag{2.35}$$

$$Y_A = \frac{1}{2}\rho V^2 S C_y \tag{2.36}$$

$$Z_A = \frac{1}{2}\rho V^2 S C_z \tag{2.37}$$

$$L_A = \frac{1}{2}\rho V^2 S b C_l \tag{2.38}$$

$$M_A = \frac{1}{2}\rho V^2 S c C_m \tag{2.39}$$

$$N_A = \frac{1}{2}\rho V^2 S b C_n \tag{2.40}$$

where C_x, C_y, C_z, C_l, C_m and C_n are non-dimensional aerodynamic coefficients coordinated in the reference line body axis system and S, b and c are the wing area, wing span and wing mean aerodynamic cord respectively, and ρ is the air density. Equations (2.35) to (2.40) show that the solution to the aerodynamic forces and moments acting on the aircraft is now reduced to finding a set of aerodynamic coefficients. The first step in finding the coefficients is to establish a relationship between coefficients coordinated in reference line body axes and stability body axes. The two systems differ in that when using stability body axes, OX_B-axis is parallel to the aircraft's velocity vector during straight and level flight. Figure 2.5 shows this situation where the subscripts B and S denote reference line and stability body axes respectively and the superscript, S, indicates stability axes coefficients.

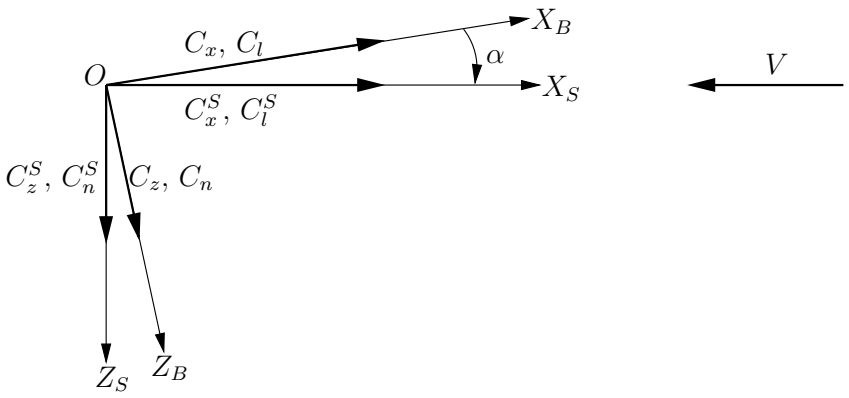


Figure 2.5: Body Reference Frames During Trim Flight

Given aerodynamic coefficients coordinated in stability axes, the associated body reference line coefficients can be calculated by coordinating the respective quantities into the instantaneous

body axis system using equations (2.41) to (2.46),

$$C_x = C_x^S - C_z^S \alpha \tag{2.41}$$
$$C_y = C_y^S \tag{2.42}$$
$$C_z = C_x^S \alpha + C_z^S \tag{2.43}$$
$$C_l = C_l^S - C_n^S \alpha \tag{2.44}$$
$$C_m = C_m^S \tag{2.45}$$
$$C_n = C_l^S \alpha + C_n^S \tag{2.46}$$

where C_x^S, C_y^S, C_z^S, C_l^S, C_m^S and C_n^S are the aerodynamic coefficients coordinated in stability axes. In these equations the assumptions have been made that the incidence angles are relatively small and should be kept in mind when wanting to perform aggressive manoeuvres at low airspeeds.

When the aircraft flies straight and level and at a constant velocity, the engine and gravitational force and moment contributions are cancelled by their aerodynamic counterparts. Inevitably the aircraft will experience a perturbation from this trim condition and extra aerodynamic forces and moments will be generated. Stability derivatives provide a means to describe these aerodynamic influences given a perturbation in one or more of the aircraft's motion variables (V, α, β, p, q and r) whereas control derivatives describe the influences generated by an aerodynamic control surface (δ_a, δ_e and δ_r) perturbation. According to [25], the aerodynamic coefficients discussed thus far can be expressed as a combination of non-dimensional stability and control derivatives when it is assumed that the incidence angles are small,

$$C_x^S = -C_D \tag{2.47}$$
$$C_y^S = C_{y_\beta}\beta + \frac{b}{2V}\left[\left(C_{y_p}p + C_{y_r}r\right) + \alpha\left(C_{y_p}r - C_{y_r}p\right)\right] + C_{y_{\delta_a}}\delta_a + C_{y_{\delta_r}}\delta_r \tag{2.48}$$
$$C_z^S = -C_L + \frac{c}{2V}C_{L_q}q \tag{2.49}$$
$$C_l^S = C_{l_\beta}\beta + \frac{b}{2V}\left[\left(C_{l_p}p + C_{l_r}r\right) + \alpha\left(C_{l_p}r - C_{l_r}p\right)\right] + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r \tag{2.50}$$
$$C_m^S = C_{m_0} + C_{m_\alpha}\alpha + \frac{c}{2V}C_{m_q}q + C_{m_{\delta_e}}\delta_e \tag{2.51}$$
$$C_n^S = C_{n_\beta}\beta + \frac{b}{2V}\left[\left(C_{n_p}p + C_{n_r}r\right) + \alpha\left(C_{n_p}r - C_{n_r}p\right)\right] + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r \tag{2.52}$$

where C_{m_0} is the zero angle of attack pitching moment coefficient and C_L and C_D are the lift and drag coefficients respectively and are given by,

$$C_L = C_{L_0} + C_{L_\alpha}\alpha \tag{2.53}$$
$$C_D = C_{D_0} + \frac{C_L^2}{\pi A e} \tag{2.54}$$

where C_{L_0} is the zero angle of attack lift coefficient, C_{D_0} is the parasitic drag and A and e are the wing's aspect ratio and efficiency factor respectively. The following example illustrates the notation used to represent the respective derivatives,

$$C_{y_\beta} \equiv \frac{\partial C_y}{\partial \beta} \tag{2.55}$$

In order to complete the aerodynamic forces and moments model, a set of stability and control derivatives needs to be calculated. [35] calculated the derivatives from first principles by making use of the aircraft's geometry and empirical formulas. This approach has the advantage that it provides physical insight into the calculated quantities since most of the formulas depend on the aircraft geometry only, whilst the drawback is that a model of limited accuracy is developed. Other methods include wind tunnel measurements as well as system identification techniques which can provide very accurate derivatives but either require substantial amounts of infrastructure or very good measurement equipment to capture the relevant data in order to develop a more accurate model.

A fourth method makes use of computational fluid dynamics methods which involves using a computer to solve a fluid dynamics problem. This solution is attractive since it only requires the the use of a desktop computer and arguably will give more accurate results than the calculation from first principles method. Athena Vortex Lattice (AVL) is a software package developed at the Massachusetts Institute of Technology as part of their Athena TODOR aero software collection. Given the aircraft geometry, AVL performs an aerodynamic analysis on the airframe at a specific trim condition and ultimately provides a set of stability and control derivatives. The trim condition used in this project is discussed in Section B.4.5, while Section B.4.6 describes how AVL was used to calculate the respective derivatives.

2.5 Non-linear State-space Model

The following sections state the aircraft's displacement and linear and angular velocity differential equations, followed by the non-linear aircraft model representation in state-space form.

2.5.1 Displacement Differential Equations

The displacement differential equations describe the aircraft's displacement relative to the inertial reference frame. The equations are derived by firstly applying small incidence angles assumptions to equations (2.8) to (2.10). The linear velocity vector is then coordinated in the inertial reference frame by means of the inverse quaternion transformation matrix. The result

is stated below,

$$\begin{bmatrix} \dot{N} \\ \dot{E} \\ -\dot{h} \end{bmatrix} = \boldsymbol{T}^{-1}(\boldsymbol{q}) \begin{bmatrix} V \\ V\beta \\ V\alpha \end{bmatrix} \tag{2.56}$$

where N, E and h are the North, East and altitude states respectively.

2.5.2 Linear and Angular Velocity Differential Equations

The linear and angular velocity differential equations are derived by firstly substituting the time derivatives of the linear velocity vector, as well as the respective force and moment contributions, into the equations of motion. This is then followed by applying mathematical manipulation to make the time derivative of the respective motion variables the subjects of the equations. In the derivation the following assumptions have been made,

$$1 \pm \alpha^2 \approx 1 \tag{2.57}$$

$$1 \pm \beta^2 \approx 1 \tag{2.58}$$

2.5.3 Non-linear State Equations

The respective differential equations are combined in state-space form to finally complete the aircraft model. The state vector contains the aircraft's motion, attitude and displacement states, together with the augmented engine thrust state, and is given by,

$$\boldsymbol{x} = [V \;\; \alpha \;\; \beta \;\; p \;\; q \;\; r \;\; q_1 \;\; q_2 \;\; q_3 \;\; q_4 \;\; N \;\; E \;\; h \;\; T]^T \tag{2.59}$$

whilst the control vector is given by,

$$\boldsymbol{u} = [T_c \;\; \delta_e \;\; \delta_a \;\; \delta_r]^T \tag{2.60}$$

The non-linear system, affine in the control vector $\boldsymbol{u}$, is now described by the differential equation,

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\,\boldsymbol{u} \tag{2.61}$$

where $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{g}(\boldsymbol{x})$ are defined as,

$$\boldsymbol{f}(\boldsymbol{x}) = [f_1 \;\; f_2 \;\; f_3 \;\; f_4 \;\; f_5 \;\; f_6 \;\; f_7 \;\; f_8 \;\; f_9 \;\; f_{10} \;\; f_{11} \;\; f_{12} \;\; f_{13} \;\; f_{14}]^T \tag{2.62}$$

$$\boldsymbol{g}(\boldsymbol{x}) =$$

$$\begin{bmatrix}
0 & 0 & \frac{\rho SC_{y_{\delta_a}}}{2m}V^2\beta & \frac{\rho SC_{y_{\delta_r}}}{2m}V^2\beta \\
0 & -\frac{\rho SC_{L_{\delta_e}}}{2m}V & 0 & 0 \\
0 & 0 & \frac{\rho SC_{y_{\delta_a}}}{2m}V & \frac{\rho SC_{y_{\delta_r}}}{2m}V \\
0 & 0 & \frac{\rho SbC_{l_{\delta_a}}}{2J_x}V^2 - \frac{\rho SbC_{n_{\delta_a}}}{2J_x}V^2\alpha & \frac{\rho SbC_{l_{\delta_r}}}{2J_x}V^2 - \frac{\rho SbC_{n_{\delta_r}}}{2J_x}V^2\alpha \\
0 & \frac{\rho ScC_{m_{\delta_e}}}{2J_y}V^2 & 0 & 0 \\
0 & 0 & \frac{\rho SbC_{n_{\delta_a}}}{2J_z}V^2 + \frac{\rho SbC_{l_{\delta_a}}}{2J_z}V^2\alpha & \frac{\rho SbC_{n_{\delta_r}}}{2J_z}V^2 + \frac{\rho SbC_{l_{\delta_r}}}{2J_z}V^2\alpha \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\frac{1}{\tau_E} & 0 & 0 & 0
\end{bmatrix} \tag{2.63}$$

f_1 to f_{14} are non-linear functions in the state variables and are given by,

$$\begin{aligned}
f_1 = {} & \frac{\rho SbC_{y_p}}{4m}(V\alpha\beta r + V\beta p) - V\alpha^2\beta r + \frac{\rho SC_{y_\beta}}{2m}V^2\beta^2 - \left(\frac{\rho SC_{D_0}}{2m} + \frac{\rho SC_{L_0}^2}{2m\pi Ae}\right)V^2 \\
& - \frac{\rho SC_{L_0}C_{L_\alpha}}{m\pi Ae}V^2\alpha - \frac{\rho SC_{L_\alpha}^2}{2m\pi Ae}V^2\alpha^2 + 2g(q_1q_3 - q_4q_2) \\
& + g\alpha\left(q_4^2 - q_1^2 - q_2^2 + q_3^2\right) + 2g\beta(q_4q_1 + q_2q_3) + \frac{1}{m}T
\end{aligned} \tag{2.64}$$

$$\begin{aligned}
f_2 = {} & q - \frac{\rho SC_{L_0}}{2m}V - \frac{\rho SC_{L_\alpha}}{2m}V\alpha - \alpha\beta r - \beta p + gV^{-1}\left(q_4^2 - q_1^2 - q_2^2 + q_3^2\right) \\
& - 2gV^{-1}\alpha(q_1q_3 - q_4q_2) - \frac{1}{m}V^{-1}\alpha T
\end{aligned} \tag{2.65}$$

$$f_3 = \frac{\rho SbC_{y_p}}{4m}(p + \alpha r) + \alpha p - r + \left(\frac{\rho SC_{y_\beta}}{2m} + \frac{\rho SC_{D_0}}{2m} + \frac{\rho SC_{L_0}^2}{2m\pi Ae}\right) V\beta$$
$$+ \frac{\rho SC_{L_0}C_{L_\alpha}}{m\pi Ae} V\alpha\beta + \frac{\rho SC_{L_\alpha}^2}{2m\pi Ae} V\alpha^2\beta + 2gV^{-1}(q_4q_1 + q_2q_3) \tag{2.66}$$
$$- gV^{-1}\alpha\beta\left(q_4^2 - q_1^2 - q_2^2 + q_3^2\right) - 2gV^{-1}\beta(q_1q_3 - q_4q_2) - \frac{1}{m}V^{-1}\beta T$$

$$f_4 = -\frac{J_z - J_y}{J_x}qr + \frac{\rho SbC_{l_\beta}}{2J_x}V^2\beta - \frac{\rho SbC_{n_\beta}}{2J_x}V^2\alpha\beta + \frac{\rho Sb^2C_{l_p}}{4J_x}(Vp + V\alpha r)$$
$$- \frac{\rho Sb^2C_{n_p}}{4J_x}\left(V\alpha p + V\alpha^2 r\right) + \frac{\rho Sb^2C_{l_r}}{4J_x}(Vr - V\alpha p) \tag{2.67}$$
$$+ \frac{\rho Sb^2C_{n_r}}{4J_x}\left(V\alpha^2 p - V\alpha r\right)$$

$$f_5 = \frac{\rho ScC_{m_0}}{2J_y}V^2 - \frac{J_x - J_z}{J_y}pr + \frac{\rho ScC_{m_\alpha}}{2J_y}V^2\alpha + \frac{\rho Sc^2C_{m_q}}{4J_y}Vq \tag{2.68}$$

$$f_6 = -\frac{J_y - J_x}{J_z}pq + \frac{\rho SbC_{n_\beta}}{2J_z}V^2\beta + \frac{\rho SbC_{l_\beta}}{2J_z}V^2\alpha\beta + \frac{\rho Sb^2C_{n_p}}{4J_z}(Vp + V\alpha r)$$
$$+ \frac{\rho Sb^2C_{l_p}}{4J_z}\left(V\alpha p + V\alpha^2 r\right) + \frac{\rho Sb^2C_{n_r}}{4J_z}(Vr - V\alpha p) \tag{2.69}$$
$$+ \frac{\rho Sb^2C_{l_r}}{4J_z}\left(V\alpha r - V\alpha^2 p\right)$$

$$f_7 = \frac{1}{2}(q_4p - q_3q + q_2r) \tag{2.70}$$

$$f_8 = \frac{1}{2}(q_3p + q_4q - q_1r) \tag{2.71}$$

$$f_9 = \frac{1}{2}(-q_2p + q_1q + q_4r) \tag{2.72}$$

$$f_{10} = \frac{1}{2}(-q_1p - q_2q - q_3r) \tag{2.73}$$

$$f_{11} = V\left(q_4^2 + q_1^2 - q_2^2 - q_3^2\right) + 2V\beta(q_1q_2 - q_4q_3) + 2V\alpha(q_4q_2 + q_1q_3) \tag{2.74}$$

$$f_{12} = 2V(q_4q_3 + q_1q_2) + V\beta\left(q_4^2 - q_1^2 + q_2^2 - q_3^2\right) + 2V\alpha(q_2q_3 - q_4q_1) \tag{2.75}$$

$$f_{13} = -2V(q_1q_3 - q_4q_2) - 2V\beta(q_4q_1 + q_2q_3) - V\alpha\left(q_4^2 - q_1^2 - q_2^2 + q_3^2\right) \tag{2.76}$$

$$f_{14} = -\frac{1}{\tau_E}T \tag{2.77}$$

Equations (2.61) to (2.77) remain valid in describing the aircraft's motion, attitude and displacement for as long as the incidence angle approximations hold and the flight envelope does not deviate too far from the trim condition at which the stability and control derivatives were calculated.

2.6 Summary

This chapter developed a mathematical aircraft model which will be used as a simulation platform in the chapters to follow. The process was started off by the definition of the respective axis systems used, and establishing some basic concepts which were key to the development of the model.

This was followed by a summary of the equations of motion for a 6-DOF rigid body after which attitude representations in terms of Euler angles and the quaternion were introduced. The Euler angle method displayed singularities when the pitch angle was set to $\pm 90^\circ$ and subsequently the quaternion was investigated as the preferred method of attitude parametrisation.

Lastly, the respective forces and moments that act on a conventional model aircraft were introduced and combined with the equations of motion to complete the non-linear aircraft model.

Chapter 3

Kinematic State Estimator

The control strategy adopted in this project requires full state feedback. This chapter is dedicated to the development of a kinematic state estimator to obtain aircraft states which are not measured directly. The applied theory and kinematic modelling is largely based on the discussion in [23]. The main focus of the chapter is on establishing the thought behind a recursive extended Kalman filter algorithm and the application of the algorithm to a set of non-linear kinematic equations to estimate the required states in an optimal manner. For a more detailed discussion on optimal estimation, see [15] and [16].

3.1 Overview and Strategy

Consider Figure 3.1 which shows a simplified representation of the kinematics used in the development of the state estimator. The figure shows an arbitrary platform, equipped with rate gyroscopes and accelerometers in a strapdown configuration. The strapdown term indicates that the various sensors are fixed to the platform and rotates and translates along with it. The measured angular velocity with respect to some inertial reference frame is used to calculate the quaternion differential equations, which are then integrated to obtain the platform's attitude. Next, the accelerometer measurements are coordinated in the inertial reference frame using the inverse quaternion transformation matrix. After compensating for gravity, the accelerations are integrated twice to obtain the platform's velocity and displacement with respect to the inertial reference frame.

The attitude, velocity and position states, all functions of integration processes, will inevitably deteriorate with time if not corrected by some means. The deterioration can be contributed to biases on the rate gyroscopes and accelerometers, as well as approximations made during the

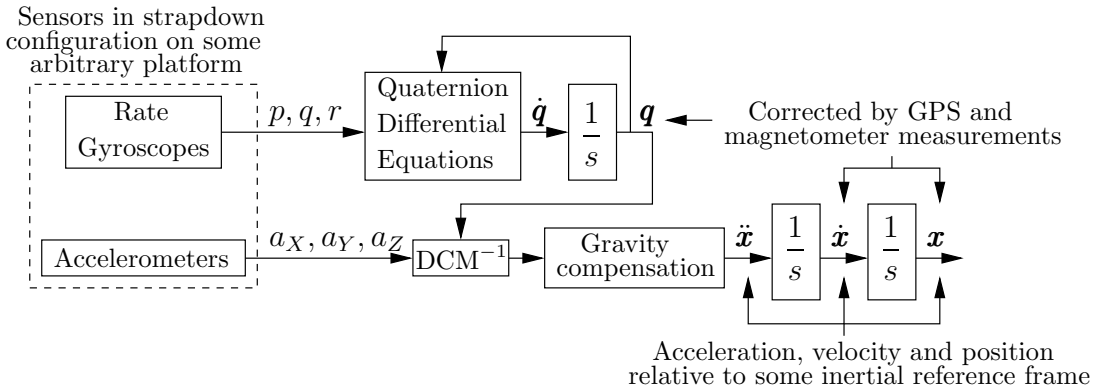


Figure 3.1: Simplified State Estimator Kinematics

integration processes. The indicated GPS and magnetometer measurements provide direct and indirect measurements of the respective states, and can be used to compensate for the drift. This forms the basis of the kinematic estimator to be implemented in this chapter.

The advantages of this approach are clear. The implementation is completely independent of the platform, flight envelope and the position. These make integration between different vehicles extremely easy. Also, the solution yields a full attitude estimate, a feat not easily achieved, as well as velocity and position states available at high frequencies. The disadvantage is that the process can be less accurate compared with techniques where very accurate vehicle dependant dynamic models are available.

An extended Kalman filter (EKF) approach will be adopted to combine the different measurements in an optimal manner. It will be shown that under no wind assumptions, and when using low cost sensors, the kinematic state estimator provides accurate enough results to completely calculate the aircraft's state vector.

3.2 Optimal State Estimation Theory

The aim of this section is to briefly summarise the concepts behind optimal state estimation and to define a recursive extended Kalman filter algorithm. Consider a discrete linear plant in the form of equations (3.1) and (3.2),

$$\boldsymbol{x}(k+1) = \boldsymbol{\Phi}\,\boldsymbol{x}(k) + \boldsymbol{\Gamma}\,\boldsymbol{u}(k) + \boldsymbol{\Gamma}_{\boldsymbol{w}}\boldsymbol{w}(k) \tag{3.1}$$

$$\boldsymbol{y}(k) = \boldsymbol{H}\,\boldsymbol{x}(k) + \boldsymbol{v}(k) \tag{3.2}$$

where $\boldsymbol{w}(k)$ and $\boldsymbol{v}(k)$ represent the plant's process and measurement noise respectively and are

random processes with the following properties,

$$\mathcal{E}\left[\boldsymbol{w}\left(k\right)\right] = \mathcal{E}\left[\boldsymbol{v}\left(k\right)\right] = \mathbf{0} \tag{3.3}$$

$$\mathcal{E}\left[\boldsymbol{w}\left(j\right)\boldsymbol{w}^{T}\left(k\right)\right] = \mathcal{E}\left[\boldsymbol{v}\left(j\right)\boldsymbol{v}^{T}\left(k\right)\right] = 0 \quad \text{if} j \neq k \tag{3.4}$$

$$\mathcal{E}\left[\boldsymbol{w}\left(j\right)\boldsymbol{w}^{T}\left(k\right)\right] = \boldsymbol{Q}\left(k\right)\delta_{jk} \tag{3.5}$$

$$\mathcal{E}\left[\boldsymbol{v}\left(j\right)\boldsymbol{v}^{T}\left(k\right)\right] = \boldsymbol{R}\left(k\right)\delta_{jk} \tag{3.6}$$

where $\mathcal{E}\left[\cdot\right]$ represents the expected value operator. The equations indicate that the process and measurement noise have zero mean and are uncorrelated. $\boldsymbol{Q}\left(k\right)$ and $\boldsymbol{R}\left(k\right)$ are defined as the discrete process and measurement noise covariance matrices. The state error vector, $\boldsymbol{e}\left(k\right)$, is the difference between the actual and the estimated state vectors and is defined below along with the error covariance matrix, $\boldsymbol{P}\left(k\right)$,

$$\boldsymbol{e}\left(k\right) = \boldsymbol{x}\left(k\right) - \widehat{\boldsymbol{x}}\left(k\right) \tag{3.7}$$

$$\boldsymbol{P}\left(k\right) = \mathcal{E}\left[\boldsymbol{e}\left(k\right)\boldsymbol{e}^{T}\left(k\right)\right] \tag{3.8}$$

Summarised below is a recursive algorithm implementing the Kalman Filter equations in order to perform real time optimal estimation for a discrete linear plant.

1. The previous best state estimate, $\widehat{\boldsymbol{x}}\left(k\right)$, is propagated forward in time by one sample instance using the plant dynamics and the deterministic input, $\boldsymbol{u}\left(k\right)$, to obtain $\overline{\boldsymbol{x}}\left(k+1\right)$. The process noise input is neglected here since only mean values are taken into account during state propagation. The error covariance of the propagated state is also calculated in the form of $\boldsymbol{M}\left(k+1\right)$,

$$\overline{\boldsymbol{x}}\left(k+1\right) = \boldsymbol{\Phi}\,\widehat{\boldsymbol{x}}\left(k\right) + \boldsymbol{\Gamma}\,\boldsymbol{u}\left(k\right) \tag{3.9}$$

$$\boldsymbol{M}\left(k+1\right) = \boldsymbol{\Phi}\boldsymbol{P}\left(k\right)\boldsymbol{\Phi}^{T} + \boldsymbol{Q}\left(k\right) \tag{3.10}$$

2. If a measurement update is available, calculate the filter gain,

$$\boldsymbol{L}\left(k+1\right) = \boldsymbol{M}\left(k+1\right)\boldsymbol{H}^{T}\left[\boldsymbol{H}\boldsymbol{M}\left(k+1\right)\boldsymbol{H}^{T} + \boldsymbol{R}\left(k+1\right)\right]^{-1} \tag{3.11}$$

   This equation encapsulates the working of the Kalman filter in that the feedback gain, $\boldsymbol{L}\left(k+1\right)$, is calculated in such a way so as to minimise the errors of the best state estimate by taking into account the certainties of the propagated and actual measurements. It is also important to note that the order of the matrix to be inverted in equation (3.11) is equal to the number of available measurements. Since matrix inversion is an expensive operation and can become inaccurate for large matrices [23], the number of measurements should be kept in mind when implementing Kalman filters for real time operation.

3. Perform the required innovation update using the calculated filter gain and the error between the actual and propagated measurements in a recursive weighted least squares manner,

$$\widehat{\boldsymbol{x}}(k+1) = \overline{\boldsymbol{x}}(k+1) + \boldsymbol{L}(k+1)\left[\boldsymbol{y}(k+1) - \boldsymbol{H}\,\overline{\boldsymbol{x}}(k+1)\right] \tag{3.12}$$

Also, calculate the error covariances on the current best state estimate,

$$\begin{aligned}\boldsymbol{P}(k+1) = &\left[\boldsymbol{I} - \boldsymbol{L}(k+1)\,\boldsymbol{H}\right]\boldsymbol{M}(k+1)\left[\boldsymbol{I} - \boldsymbol{L}(k+1)\,\boldsymbol{H}\right] \\ &+ \boldsymbol{L}(k+1)\,\boldsymbol{R}(k+1)\,\boldsymbol{L}(k+1)^T\end{aligned} \tag{3.13}$$

where $\boldsymbol{I}$ denotes the identity matrix. Equation (3.13) is one of many different forms to update the error covariance matrix. The Joseph's form, as used here, has the advantage that it improves the numerical stability of the filter in making sure the error covariance matrix remains positive-semi-definite [23].

In practice, very few systems can be completely described by linear dynamics, and thus a technique is required to accommodate non-linear systems. The EKF is an extension of the theory discussed thus far in that it approximates the non-linear system as a linear time variant system, and in doing so provides a solution to state estimation for non-linear systems. Given a continuous non-linear system of the form,

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{w} \tag{3.14}$$

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{v} \tag{3.15}$$

where $\boldsymbol{f}$ and $\boldsymbol{h}$ are non-linear functions in the state and/or control vectors, and $\boldsymbol{w}$ and $\boldsymbol{v}$ are continuous white noise signals with Gaussian distributions and have the following power spectral density (PSD) matrices,

$$\mathcal{E}\left[\boldsymbol{w}\boldsymbol{w}^T\right] = \boldsymbol{Q_c} \tag{3.16}$$

$$\mathcal{E}\left[\boldsymbol{v}\boldsymbol{v}^T\right] = \boldsymbol{R_c} \tag{3.17}$$

Summarised below are six steps which describe how an Extended Kalman filter can be implemented for a plant in the form of equations (3.14) and (3.15),

1. The continuous dynamics are linearised about the previous best state estimate by calculating the respective Jacobian matrices and evaluating them at $\widehat{\boldsymbol{x}}(k)$ and $\boldsymbol{u}(k)$,

$$\boldsymbol{F_k} = \left[\frac{\partial \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{x}}\right]_{\widehat{\boldsymbol{x}}(k), \boldsymbol{u}(k)} \tag{3.18}$$

$$\boldsymbol{G_k} = \left[\frac{\partial \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{u}}\right]_{\widehat{\boldsymbol{x}}(k), \boldsymbol{u}(k)} \tag{3.19}$$

2. The continuous linear system dynamics are converted to the discrete time domain using the discrete sampling time T_s,

$$\boldsymbol{\Phi}(k) = \boldsymbol{I} + \boldsymbol{F_k} T_s + \frac{\boldsymbol{F_k}^2 T_s^2}{2!} + \ldots \approx \boldsymbol{I} + \boldsymbol{F_k} T_s \tag{3.20}$$

$$\boldsymbol{\Gamma}(k) = \boldsymbol{G_k} T_s + \boldsymbol{G_k} \frac{\boldsymbol{F_k} T_s^2}{2!} + \boldsymbol{G_k} \frac{\boldsymbol{F_k}^2 T_s^3}{3!} + \ldots \approx \boldsymbol{G_k} T_s \tag{3.21}$$

where the approximations in equations (3.20) and (3.21) are based on the fact that the discrete sampling time is much shorter than the system time constants [7].

3. The next step is to calculate the equivalent discrete process and measurement noise covariances. Equation (3.14) indicates that the continuous process noise, $\boldsymbol{w}$, maps directly onto the state vector. This is a very simple scenario and only requires the process noise PSD matrix, $\boldsymbol{Q_c}$, to be converted to its discrete covariance equivalent. According to [7], such a relationship is given by,

$$\boldsymbol{Q}(k) = \frac{\boldsymbol{Q_c}}{T_s} \tag{3.22}$$

and remains valid for as long as the discrete sampling time is much shorter than the system time constants. Most often the input vector, $\boldsymbol{u}$, is not completely deterministic and corrupted by noise. The process noise for such a system then becomes a lumped parameter consisting of process and actuation noise contributions. The inputs' PSD matrix is converted to its discrete covariance counterpart to obtain $\boldsymbol{Q_u}(k)$. The noise properties are then mapped onto the states making use of the discrete input matrix. The result is stated below,

$$\boldsymbol{Q}(k) = \boldsymbol{\Gamma}(k)\, \boldsymbol{Q_u}(k)\, \boldsymbol{\Gamma}(k)^T \tag{3.23}$$

The final process noise covariance is then calculated by summing equations (3.22) and (3.23). Equation (3.15) indicates that measurement updates are available continuously. However, in practice EKFs are most often updated sporadically by different measurements. Also, the noise properties of these sensors are usually provided in terms of discrete noise covariances. Therefore, for the scope of this project it is assumed that the measurement noise properties are quantified in terms of a covariances matrix denoted by $\boldsymbol{R}(k)$.

4. Propagate the previous best state estimate forward in time by integrating the complete non-linear dynamics over one sample instance,

$$\overline{\boldsymbol{x}}(k+1) = \widehat{\boldsymbol{x}}(k) + \dot{\widehat{\boldsymbol{x}}}(k)\, T_s \tag{3.24}$$

where

$$\dot{\widehat{\boldsymbol{x}}}(k) = \boldsymbol{f}\left(\widehat{\boldsymbol{x}}(k), \boldsymbol{u}(k)\right) \tag{3.25}$$

The integration could also have been performed using any numerical integration technique such as Runge-Kutta. Euler integration was used throughout the project due to its simplicity and provides accurate enough results when the integration time steps are kept small. Making use of the linearised discrete system dynamics and noise properties, the error covariances on the propagated states are also calculated,

$$\boldsymbol{M}\left(k+1\right)=\boldsymbol{\Phi}\left(k\right)\boldsymbol{P}\left(k\right)\boldsymbol{\Phi}\left(k\right)^{T}+\boldsymbol{Q}\left(k\right) \tag{3.26}$$

5. When a measurement update becomes available, the non-linear measurements stated in equation (3.15) are linearised about the propagated state vector and the filter gain is calculated as stated below,

$$\boldsymbol{H}\left(k+1\right)=\left[\frac{\partial\boldsymbol{h}\left(\boldsymbol{x}\right)}{\partial\boldsymbol{x}}\right]_{\overline{\boldsymbol{x}}(k+1)} \tag{3.27}$$

$$\boldsymbol{L}\left(k+1\right)=\boldsymbol{M}\left(k+1\right)\boldsymbol{H}\left(k+1\right)^{T}\left[\boldsymbol{H}\left(k+1\right)\boldsymbol{M}\left(k+1\right)\boldsymbol{H}\left(k+1\right)^{T}+\boldsymbol{R}\left(k+1\right)\right]^{-1} \tag{3.28}$$

6. Perform the innovation update to obtain the current best state estimate and also update the accompanying error covariance matrix,

$$\widehat{\boldsymbol{x}}\left(k+1\right)=\overline{\boldsymbol{x}}\left(k+1\right)+\boldsymbol{L}\left(k+1\right)\left[\boldsymbol{y}\left(k+1\right)-\boldsymbol{h}\left(\overline{\boldsymbol{x}}\left(k+1\right)\right)\right] \tag{3.29}$$

$$\begin{aligned}\boldsymbol{P}\left(k+1\right)=&\left[\boldsymbol{I}-\boldsymbol{L}\left(k+1\right)\boldsymbol{H}\left(k+1\right)\right]\boldsymbol{M}\left(k+1\right)\left[\boldsymbol{I}-\boldsymbol{L}\left(k+1\right)\boldsymbol{H}\left(k+1\right)\right]\\&+\boldsymbol{L}\left(k+1\right)\boldsymbol{R}\left(k+1\right)\boldsymbol{L}\left(k+1\right)^{T}\end{aligned} \tag{3.30}$$

Important to notice is when no updates are available, the output matrix, $\boldsymbol{H}\left(k+1\right)$, and the filter gain go to zero. This in turn implies that the current best state estimate simply becomes the propagated state along with their respective error covariance matrices. Now, given a non-linear system in the form of equations (3.14) and (3.15), the process described above can be used to implement an Extended Kalman filter to perform real-time state estimation. To start the process, the following initial values are required,

- The process noise PSD matrix, $\boldsymbol{Q_c}$, and the measurement noise covariance matrix, $\boldsymbol{R}$.
- The best state estimate, $\widehat{\boldsymbol{x}}\left(0\right)$, as well as the associated error covariance matrix, $\boldsymbol{P}\left(0\right)$.

3.3 Kinematic Modelling

Chapter 2 introduced an inertial reference frame, fixed to the earth's surface at some convenient point, which was used in the development of the aircraft model. However, since the aim is to develop a vehicle and position independent estimator, a more general reference frame will be

defined. Consider Figure 3.2 which introduces the Earth Centred Earth Fixed (ECEF) geocentric system. As the name suggests, the origin of this system is the centre of the earth and introduces the concepts of latitude, longitude and height above the earth's surface.

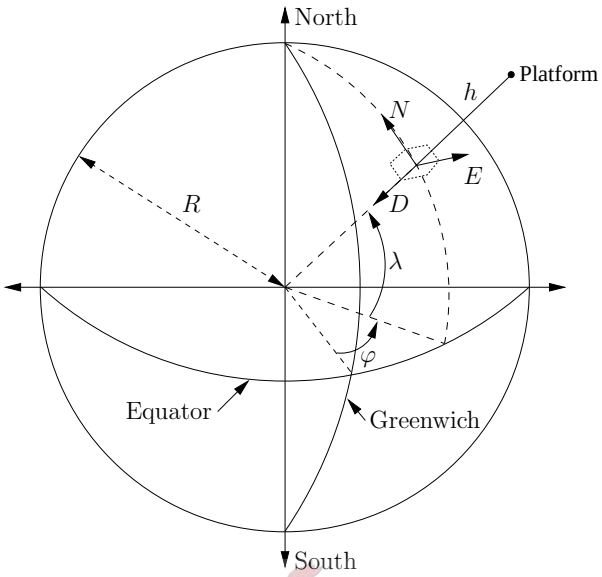


Figure 3.2: ECEF Geocentric Axis System Definition

The latitude, denoted by λ, is positive in the Northern hemisphere and takes on values between $-90°$ and $90°$. Longitude, denoted by φ, is positive to the East of Greenwich and takes on values between $-180°$ and $180°$. The height, denoted by h, is the distance above the earth's surface. The ECEF system used in this project assumes a round earth model where the constant radius is donated by R. This assumption avoids the extra complexity introduced when having to model the earth more accurately. The figure also shows the North-East-Down (NED) axis system. The NED system is not strictly speaking an inertial reference frame since it constantly changes with variations in latitude and longitude. Given a certain position, this axis system is defined as follows: The N-axis points North and is tangential to the earth's surface. The E-axis, also tangential, points East along a constant latitude curve, whilst the D-axis points towards the centre of the earth.

3.3.1 Non-linear State Equations

Inspecting the geometry in Figure 3.2, the latitude, longitude and altitude time derivatives can be written in terms of the NED velocity components as follows,

$$\dot{\lambda} = \frac{V_N}{R+h} \tag{3.31}$$

$$\dot{\varphi} = \frac{V_E}{(R+h)\cos\lambda} \tag{3.32}$$

$$\dot{h} = -V_D \tag{3.33}$$

The time derivatives of the respective velocity components are given by,

$$\dot{V}_N = \alpha_N \tag{3.34}$$

$$\dot{V}_E = \alpha_E \tag{3.35}$$

$$\dot{V}_D = \alpha_D + g'_D \tag{3.36}$$

where α_N, α_E and α_D are the accelerometer measurements coordinated in the inertial reference frame whilst the term denoted g'_D, is included to compensate for the unmeasured gravitational acceleration. Equations for α_N, α_E and α_D are obtained by taking the body axis accelerometer measurements and coordinating them in the inertial frame using the inverse quaternion transformation matrix,

$$\begin{bmatrix} \alpha_N \\ \alpha_E \\ \alpha_D \end{bmatrix} = \boldsymbol{T}^{-1}(\boldsymbol{q}) \begin{bmatrix} a_X \\ a_Y \\ a_Z \end{bmatrix} \tag{3.37}$$

where a_X, a_Y and a_Z are the respective accelerometer measurements. Combining equations (3.34) to (3.37), the time derivatives of the velocity components are found to be,

$$\dot{V}_N = \left(q_4^2 + q_1^2 - q_2^2 - q_3^2\right) a_X + 2\left(q_1 q_2 - q_4 q_3\right) a_Y + 2\left(q_4 q_2 + q_1 q_3\right) a_Z \tag{3.38}$$

$$\dot{V}_E = 2\left(q_4 q_3 + q_1 q_2\right) a_X + \left(q_4^2 - q_1^2 + q_2^2 - q_3^2\right) a_Y + 2\left(q_2 q_3 - q_4 q_1\right) a_Z \tag{3.39}$$

$$\dot{V}_D = 2\left(q_1 q_3 - q_4 q_2\right) a_X + 2\left(q_4 q_1 + q_2 q_3\right) a_Y + \left(q_4^2 - q_1^2 - q_2^2 + q_3^2\right) a_Z + g'_D \tag{3.40}$$

The angular velocity measurements' relation to the time derivatives of the quaternion components is given by equation (2.31) and is not restated here. The Global Positioning System (GPS) receiver used in this project displays a measurement time delay in the order of 310 ms [37]. Considering the platform being a high velocity application, substantial errors would occur in the estimated state vector should such a delay be left un-modelled. The continuous time delay, e^{-Ts} (where T is the delay in seconds and s the Laplace operator), is modelled using a

first order Padé approximation of the form,

$$\frac{y_d(s)}{y(s)} = \frac{-s+\omega_p}{s+\omega_p} \tag{3.41}$$

where ω_p is the approximation's pole/zero location and the subscript, d, indicates the signal's delayed version. The transfer function in equation (3.41) has unity gain at all frequencies but the delay approximation starts to degrade as the frequency increases. Figure 3.3 shows a frequency response of the actual and approximated delays.

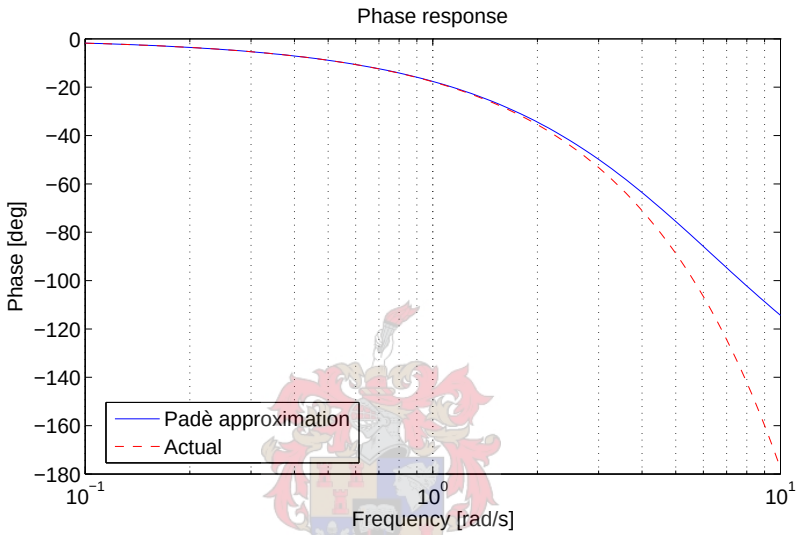


Figure 3.3: Phase Responses of Actual GPS Delay and Padé Approximation

The figure indicates that for frequencies up to 1 Hz, the first order approximation remains valid and considering the bandwidth of the platform's position and velocity motion variables, the approximation is assumed adequate in modelling the GPS delay. Writing equation (3.41) in differential equation form and applying to equations (3.31) to (3.33) and (3.38) to (3.40), the delayed position and velocity dynamics are obtained,

$$\dot{\lambda}_d = \omega_p\lambda - \frac{V_N}{R+h} - \omega_p\lambda_d \tag{3.42}$$

$$\dot{\varphi}_d = \omega_p\varphi - \frac{V_E}{(R+h)\cos\lambda} - \omega_p\varphi_d \tag{3.43}$$

$$\dot{h}_d = \omega_p h + V_D - \omega_p h_d \tag{3.44}$$

$$\dot{V}_{N_d} = \omega_p V_N - \left(q_4^2 + q_1^2 - q_2^2 - q_3^2\right) a_X - 2\left(q_1 q_2 - q_4 q_3\right) a_Y - 2\left(q_4 q_2 + q_1 q_3\right) a_Z - \omega_p V_{N_d} \tag{3.45}$$

$$\dot{V}_{E_d} = \omega_p V_E - 2\left(q_4 q_3 + q_1 q_2\right) a_X - \left(q_4^2 - q_1^2 + q_2^2 - q_3^2\right) a_Y - 2\left(q_2 q_3 - q_4 q_1\right) a_Z - \omega_p V_{E_d} \tag{3.46}$$

$$\dot{V}_{D_d} = \omega_p V_D - 2\left(q_1 q_3 - q_4 q_2\right) a_X - 2\left(q_4 q_1 + q_2 q_3\right) a_Y - \left(q_4^2 - q_1^2 - q_2^2 + q_3^2\right) a_Z - g'_D - \omega_p V_{D_d} \tag{3.47}$$

where the subscript, d, indicates a delayed state. Having modelled the kinematics, the state and input vectors can be defined,

$$\boldsymbol{x} = [\lambda \;\; \varphi \;\; h \;\; V_N \;\; V_E \;\; V_D \;\; q_1 \;\; q_2 \;\; q_3 \;\; q_4 \;\; \lambda_d \;\; \varphi_d \;\; h_d \;\; V_{N_d} \;\; V_{E_d} \;\; V_{D_d}]^T \tag{3.48}$$

$$\boldsymbol{u} = [a_X \;\; a_Y \;\; a_Z \;\; p \;\; q \;\; r]^T \tag{3.49}$$

The non-linear state equation for the platform is now given by,

$$\dot{\boldsymbol{x}} = \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{u}\right) = [f_1 \;\; f_2 \;\; f_3 \;\; f_4 \;\; f_5 \;\; f_6 \;\; f_7 \;\; f_8 \;\; f_9 \;\; f_{10} \;\; f_{11} \;\; f_{12} \;\; f_{13} \;\; f_{14} \;\; f_{15} \;\; f_{16}]^T \tag{3.50}$$

where f_1 to f_{16} are non-linear functions in the state and control vectors and are defined by the following equations,

$$\begin{aligned} f_1 \text{ to } f_3 &\Rightarrow \text{Equations (3.31) to (3.33)} \\ f_4 \text{ to } f_6 &\Rightarrow \text{Equations (3.38) to (3.40)} \\ f_7 \text{ to } f_{10} &\Rightarrow \text{Equation (2.31)} \\ f_{11} \text{ to } f_{16} &\Rightarrow \text{Equations (3.42) to (3.47)} \end{aligned}$$

Effects which remain un-modelled include bias instability of the accelerometer and rate gyroscope measurements, as well as Coriolis forces exerted on the platform due to the earth's rotation rate. When the simulation results are presented, it will be shown that rate gyroscope bias instability does influence the performance of the estimator, but acceptable accuracy can still be achieved. For a discussion on how to model these effects see [23].

3.3.2 Non-Linear Measurement Equations

The GPS module used in this project provides direct measurements of the delayed position and velocity states and subsequently can be treated as linear measurements. The GPS measurements are defined as follows,

$$h_1 = \lambda_d \qquad h_2 = \varphi_d \qquad h_3 = h_d \tag{3.51}$$

$$h_4 = V_{N_d} \qquad h_5 = V_{E_d} \qquad h_6 = V_{D_d} \tag{3.52}$$

The magnetometer provides an absolute measurement which is directly related to the attitude of the platform. For the magnetometer measurements to become valuable, a model is required to provide the earth's magnetic field as a function of time and the platform's position. However, for the scope of this project the magnetic field is assumed constant, given a certain region of operation, but can easily be extended to become state dependent. The magnetometer measurements are related to the constant magnetic field via the quaternion transformation matrix,

$$\begin{bmatrix} h_7 \\ h_8 \\ h_9 \end{bmatrix} = \boldsymbol{T}(\boldsymbol{q}) \begin{bmatrix} B_N \\ B_E \\ B_D \end{bmatrix} \tag{3.53}$$

where B_N, B_E and B_D are the earth's constant magnetic field components along the N, E and D axes respectively. The non-linear measurement vector now consists of GPS and magnetometer measurements and is given by,

$$\boldsymbol{h}(\boldsymbol{x}) = [h_1 \quad h_2 \quad h_3 \quad h_4 \quad h_5 \quad h_6 \quad h_7 \quad h_8 \quad h_9]^T \tag{3.54}$$

where h_1 to h_6 are given by equations (3.51) and (3.52), and h_7 to h_9 are obtained from equation (3.53) after having performed the required matrix multiplication,

$$h_7 = \left(q_4^2 + q_1^2 - q_2^2 - q_3^2\right) B_N + 2\left(q_4 q_3 + q_1 q_2\right) B_E + 2\left(q_1 q_3 - q_4 q_2\right) B_D \tag{3.55}$$

$$h_8 = 2\left(q_1 q_2 - q_4 q_3\right) B_N + \left(q_4^2 - q_1^2 + q_2^2 - q_3^2\right) B_E + 2\left(q_4 q_1 + q_2 q_3\right) B_D \tag{3.56}$$

$$h_9 = 2\left(q_4 q_2 + q_1 q_3\right) B_N + 2\left(q_2 q_3 - q_4 q_1\right) B_E + \left(q_4^2 - q_1^2 - q_2^2 + q_3^2\right) B_D \tag{3.57}$$

3.4 State Initialisation

The sections to follow discuss how the various filter states are initialised. Throughout the discussion it is assumed that the platform is stationary and that all sensors provide valid measurements.

Position and Velocity

The position states, along with their delayed counterparts, are initialised by averaging the GPS position measurements over a predefined number of samples. Due to the assumption made that the platform is stationary during initialisation, all velocity states are set to zero.

Attitude

The quaternion initialisation requires a more complex approach since no direct measurement of the platform's attitude is available. Several algorithms exist that take body axis measurement vectors and inertial axis reference vectors and then calculate the attitude. The TRIAD algorithm is one such method and returns the attitude indirectly in the form of a transformation matrix [28]. In order to use the TRIAD algorithm, the platform's sensors must provide measurements of two independent vectors. The magnetometer provides a vector measurement of the earth's magnetic field and, because it is assumed that the platform is stationary, the triad of accelerometers can be used to provide a measurement of the reaction force to the earth's gravitation vector.

Now, denote the measured vectors in body axes by $\boldsymbol{w_B^1}$ and $\boldsymbol{w_B^2}$ and their accompanying inertial modelled vectors by $\boldsymbol{w_I^1}$ and $\boldsymbol{w_I^2}$. The TRIAD algorithm will now try and determine a transformation matrix, $\boldsymbol{T}$, subject to the constraints imposed by the equations,

$$\boldsymbol{w_B^1} = \boldsymbol{T}\boldsymbol{w_I^1} \tag{3.58}$$

$$\boldsymbol{w_B^2} = \boldsymbol{T}\boldsymbol{w_I^2} \tag{3.59}$$

This is accomplished by three simple steps:

1. Construct a TRIAD of orthonormal (orthogonal and normalised) vectors, $\boldsymbol{t_1}$, $\boldsymbol{t_2}$ and $\boldsymbol{t_3}$, in the body reference frame from measurements $\boldsymbol{w_B^1}$ and $\boldsymbol{w_B^2}$. Consider Figure 3.4. $\boldsymbol{t_1}$ is obtained by normalising measurement $\boldsymbol{w_B^1}$,

$$\boldsymbol{t_1} = \frac{\boldsymbol{w_B^1}}{|\boldsymbol{w_B^1}|} \tag{3.60}$$

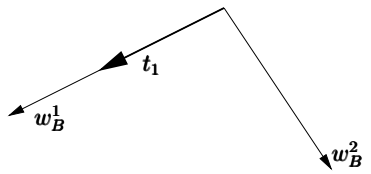


Figure 3.4: TRIAD Algorithm: Construction of $\boldsymbol{t_1}$

Consider Figure 3.5. $\boldsymbol{t_2}$ is the vector perpendicular to $\boldsymbol{t_1}$ and $\boldsymbol{w_B^2}$ and normalised,

$$\boldsymbol{t_2} = \frac{\boldsymbol{t_1} \times \boldsymbol{w_B^2}}{|\boldsymbol{t_1} \times \boldsymbol{w_B^2}|} \tag{3.61}$$

where $\times$ denotes the vector cross product operator.

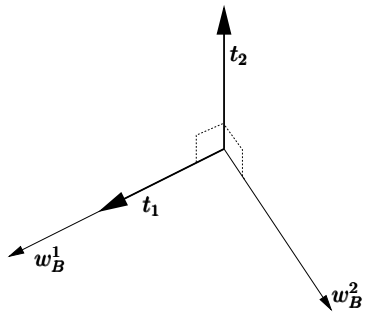


Figure 3.5: TRIAD Algorithm: Construction of $\boldsymbol{t_2}$

Consider Figure 3.6. $\boldsymbol{t_3}$ is the vector perpendicular to $\boldsymbol{t_1}$ and $\boldsymbol{t_2}$ and normalised,

$$\boldsymbol{t_3} = \frac{\boldsymbol{t_1} \times \boldsymbol{t_2}}{|\boldsymbol{t_1} \times \boldsymbol{t_2}|} = \boldsymbol{t_1} \times \boldsymbol{t_2} \tag{3.62}$$

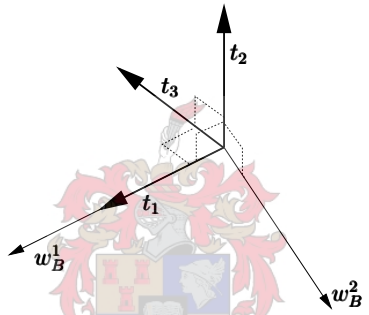


Figure 3.6: TRIAD Algorithm: Construction of $\boldsymbol{t_3}$

2. Construct a TRIAD of orthonormal vectors, $\boldsymbol{r_1}$, $\boldsymbol{r_2}$ and $\boldsymbol{r_3}$, in the reference frame from modelled vectors $\boldsymbol{w_I^1}$ and $\boldsymbol{w_I^2}$ in exactly the same manner as step 1.

3. The last step involves calculating the transformation matrix that transforms the reference TRIAD to the body TRIAD. This is accomplished by transforming the individual normalised vectors that make up the respective TRIADs and are stated below,

$$\boldsymbol{t_1} = \boldsymbol{T}\boldsymbol{r_1} \tag{3.63}$$

$$\boldsymbol{t_2} = \boldsymbol{T}\boldsymbol{r_2} \tag{3.64}$$

$$\boldsymbol{t_3} = \boldsymbol{T}\boldsymbol{r_3} \tag{3.65}$$

Equations (3.63) to (3.65) can also be written in matrix form,

$$[\boldsymbol{t_1} \;\; \boldsymbol{t_2} \;\; \boldsymbol{t_3}] = \boldsymbol{T}\,[\boldsymbol{r_1} \;\; \boldsymbol{r_2} \;\; \boldsymbol{r_3}] \tag{3.66}$$

After manipulating equation (3.66), the transformation matrix can be calculated as follows,

$$\boldsymbol{T} = [\boldsymbol{t_1} \;\; \boldsymbol{t_2} \;\; \boldsymbol{t_3}]\,[\boldsymbol{r_1} \;\; \boldsymbol{r_2} \;\; \boldsymbol{r_3}]^{-1} \tag{3.67}$$

Finally, the technique discussed in Section A.3 is used to extract the quaternion from $\boldsymbol{T}$.

During practical implementation, the body axis measurement vectors are obtained by averaging the magnetometer and accelerometer measurements over a predefined number of samples for increased accuracy. Other more complex algorithms exist which will yield more accurate results than the TRIAD algorithm discussed here. One such algorithm is the Quaternion Estimator, which caters for more than two measured/modelled vector pairs and also takes into account the accuracy of each measurement. However, the aim is to simply find an initial value for the attitude and not necessarily a best estimate and subsequently the TRIAD algorithm was chosen as the preferred method to find the initial quaternion.

3.5 Simulation

In order to evaluate the performance of the kinematic state estimator, the aircraft model developed in the previous chapter was used as a platform to generate a reference state vector, as well as the associated measurement inputs and updates. The data sets were recorded as noise free sequences while the aircraft was performing conventional flight, as well as various aerobatic manoeuvres. Next, the accelerometer and rate gyroscope measurements were corrupted with white noise signals to introduce process noise into the system. The input noise covariance matrix is a 6×6 matrix defined as follows,

$$\boldsymbol{Q_u}(k) = \text{diag}\left(\begin{bmatrix}\sigma_{a_X}^2 & \sigma_{a_Y}^2 & \sigma_{a_Z}^2 & \sigma_p^2 & \sigma_q^2 & \sigma_r^2\end{bmatrix}\right) \tag{3.68}$$

The respective standard deviations were calculated by firstly recording data sets from the accelerometers and rate gyroscopes discussed in Chapter 5. The sets were then analysed by calculating their respective standard deviations. It should be noted that these values do not include any vibration induced sensor noise which will potentially be evident during flight tests. The figures are given below,

$$\sigma_{a_X} = \sigma_{a_Y} = \sigma_{a_Z} = 0.1414\,\text{ms}^{-2} \tag{3.69}$$

$$\sigma_p = \sigma_q = \sigma_r = 0.8\,^\circ/\text{s} \tag{3.70}$$

The measurement noise covariance matrix is a 9×9 matrix with the GPS and magnetometer noise variances on its diagonal,

$$\boldsymbol{R}(k) = \text{diag}\left(\begin{bmatrix}\sigma_{\lambda_d}^2 & \sigma_{\varphi_d}^2 & \sigma_{h_d}^2 & \sigma_{V_{N_d}}^2 & \sigma_{V_{E_d}}^2 & \sigma_{V_{D_d}}^2 & \sigma_{m_X}^2 & \sigma_{m_Y}^2 & \sigma_{m_Z}^2\end{bmatrix}\right) \tag{3.71}$$

The GPS module used in this project has the functionality that it can provide the accuracy of its measurements in real time. The module was set up to provide this information which

was then recorded and analysed to obtain the GPS noise figures. During the practical implementation of the filter, the measurement noise covariance matrix was made time dependent by continuously updating its entries with the accuracy information obtained from the module. The magnetometer standard deviations were calculated in the same manner as the accelerometers and rate gyroscopes. The figures are given below,

$$\sigma_{\lambda_d} = \sigma_{\varphi_d} = 6.2832 \times 10^{-7}\,\text{rad} \tag{3.72}$$

$$\sigma_{h_d} = 4\,\text{m} \tag{3.73}$$

$$\sigma_{V_{N_d}} = \sigma_{V_{E_d}} = \sigma_{V_{D_d}} = 0.5\,\text{ms}^{-1} \tag{3.74}$$

$$\sigma_{m_X} = \sigma_{m_Y} = \sigma_{m_Z} = 2 \times 10^{-2}\,\text{Gauss} \tag{3.75}$$

where the latitude and longitude standard deviations correspond to North/South and East/West standard deviations in the order of 4 meters. Inspecting the process and measurement noise figures, it is clear that the project was restricted to the use of low cost sensors. This led to the decision to make the process noise figures a function of the input measurements alone, and not of un-modelled kinematics. The noise figures should be kept in mind when the simulation results are presented.

With all the sensor data at hand, the EKF recursive algorithm, as discussed earlier, was implemented to estimate the platform's state vector. The filter was set to execute at 50 Hz, and GPS updates were made available at 4 Hz. When implementing any form of state estimator, it is extremely important to make sure the system is observable before calculating the filter gain. Even though magnetometer updates are generally available at a much higher rate than GPS updates, it has been established that the filter is not observable with a magnetometer update only. This led to the decision to only use magnetometer measurements when a GPS update is also available. Appendix C contains more details specific to the implementation.

Figures 3.7 to 3.9 show the estimated states on the left hand side, and errors between the estimated and actual state vectors on the right hand side. The attitude graphs show the aircraft performing a 360° roll after 45 seconds, and two loop manoeuvres at 50 and 120 seconds respectively. The error graphs also show the 2σ error bounds by means of the dashed lines, and were obtained from the error covariance matrix, $\boldsymbol{P}(k)$. By definition, 95% of the state errors should fall within these bounds when dealing with a completely linear filter. However, in the EKF case this is not always true since the non-linear transformation of a Gaussian probability density function is no longer Gaussian. Inspecting the figures closely, it appears as if the property does indeed hold in this case.

To get a feel for the position accuracy, the latitude and longitude 2σ bounds can be converted to equivalents in meters by means of the constant earth radius. The North/South and East/West 2σ

bounds were calculated to be in the order of 1.4 meters at Stellenbosch's latitude and longitude. The Euler angles and their respective error covariances were calculated making use of the theory discussed in Appendix A. Another interesting aspect to note is that the filter converges after about 10 seconds and nearly settles into a steady state solution. When the filter operates in steady state, the increase in state error covariance is continuously nullified by the respective measurements. Figure 3.10 shows the error on the estimated North velocity component with the aircraft flying straight and level. Inspecting the figure, it is clearly visible how the error covariance grows during state propagation and is then decreased upon GPS and magnetometer measurements becoming available at 4Hz.

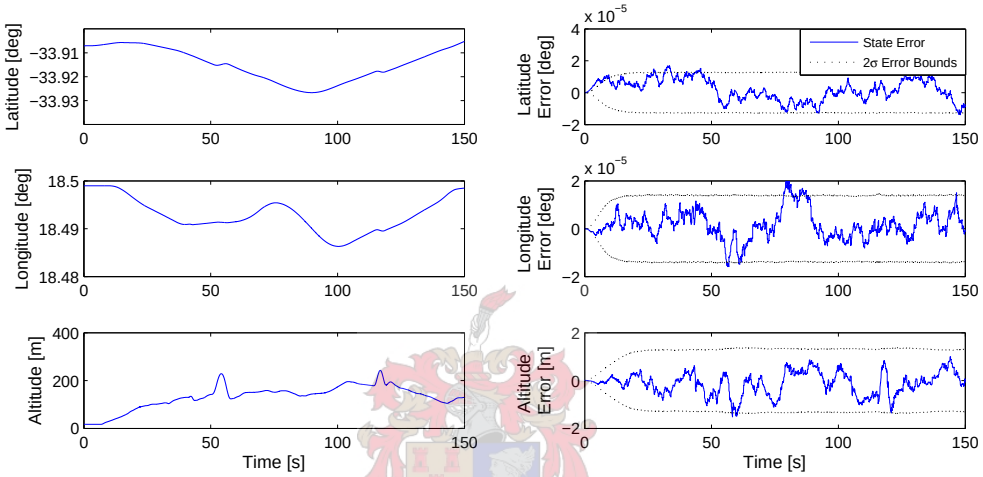


Figure 3.7: EKF Estimated Position States (on the left) and State Errors (on the right)

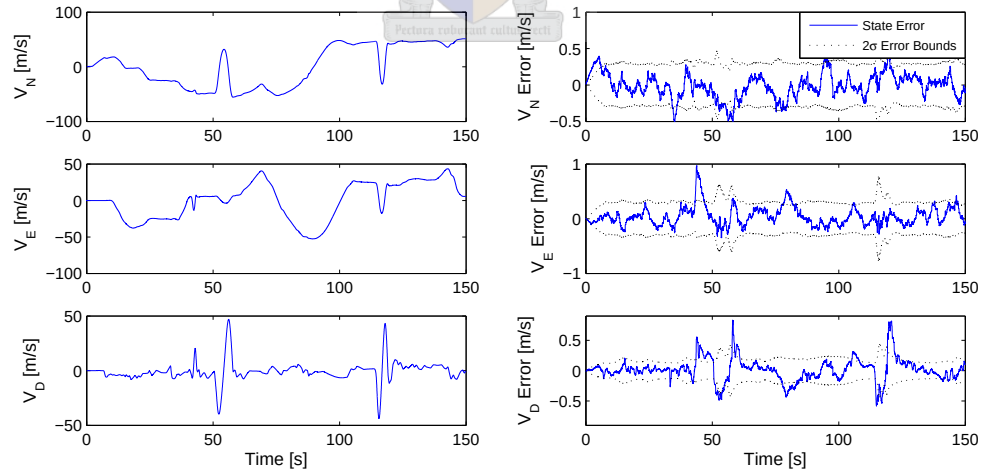


Figure 3.8: EKF Estimated Velocity States (on the left) and State Errors (on the right)

3.5.1 Calculation of Aircraft State Vector

As stated before, the kinematic state estimator can be used to completely calculate the aircraft's state vector under certain assumptions. Equation (3.37) shows that the filter estimates the

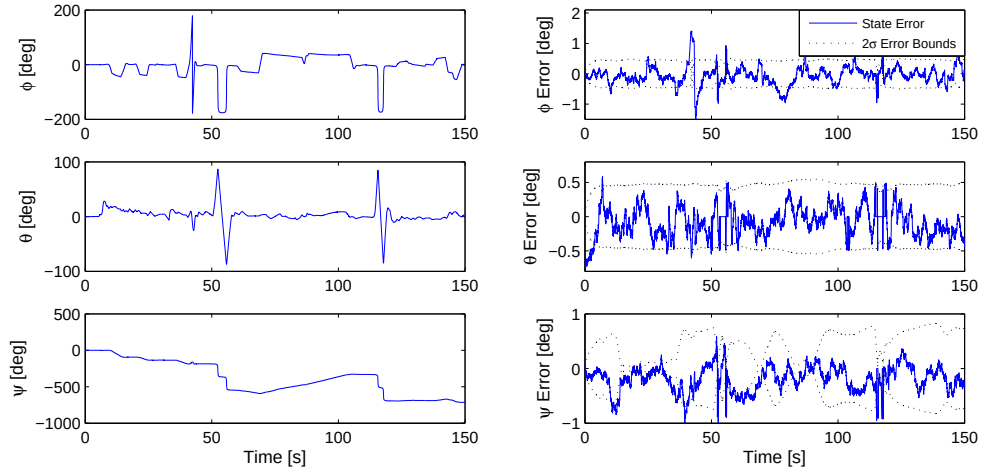


Figure 3.9: EKF Estimated Attitude States (on the left) and State Errors (on the right)

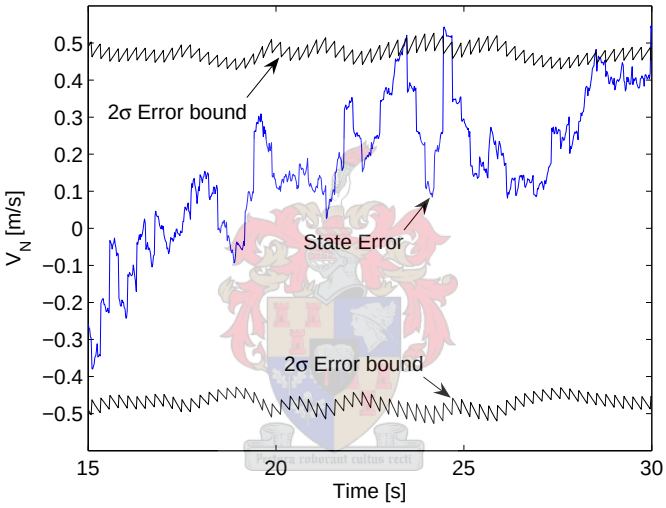


Figure 3.10: EKF Estimated North Velocity State Error

attitude between the instantaneous NED axis system and the platform axis system in which the respective sensors were strapped down. Now, when the estimator is used in the region where the aircraft model's inertial axis system was defined, the NED and $OX_IY_IZ_I$ axis systems would be aligned. Also, if the sensors are strapped down in the reference line body axis system, the kinematic state estimator must estimate the aircraft's attitude.

Under no wind assumptions, the estimated NED velocity components can be coordinated into the aircraft's body axis system and by making use of equations (2.5) to (2.7), the airspeed and incidence angle states can be calculated. It is extremely important to realise that this method would only yield accurate results when the wind speed is very low relative to the ground speed. If the approximation does not hold, other methods would have to be investigated to obtain the airspeed and incidence angle states. The North and East displacement states can be calculated

as follows,

$$N = R\,(\lambda - \lambda_0) \tag{3.76}$$

$$E = R\,(\varphi - \varphi_0)\cos\lambda_0 \tag{3.77}$$

where λ_0 and φ_0 are the latitude and longitude values where the $OX_IY_IZ_I$ axis system is defined.

The accelerometers used in this project display very good bias stability. However, the same cannot be said about the rate gyroscopes and implies that the rate gyroscope noise properties cannot be presented as a zero mean random process any more. The solution to the problem is usually to augment bias states into the filter. However, in order to avoid increasing the size of the filter, bias drifts will simply be superimposed on the measurements, and then the influence on the accuracy of the filter will be evaluated. The bias drifts were modelled as random walk sequences and generated by the integration of white noise sequences. The magnitude and rate of the drift is manipulated by changing the noise power to be integrated until the drift reflects practical values. The noise powers were set up to obtain drifts in the order of $0.25°/s$ in 3 minutes [40].

Consider Table 3.1 which shows the RMS errors on the estimated state vector under no-bias and bias conditions, denoted by scenarios 1 and 2 respectively. A third scenario has also been included to evaluate the accuracy when bias instabilities are superimposed on the rate gyroscope measurements and the filter is only updated with GPS measurements at 4 Hz. The error values were calculated using the same flight conditions as shown by Figures 3.7 to 3.9. The table indicates that there are negligible differences between most of the states for the three scenarios. However, it is clear that the incidence angles start to deteriorate rapidly in the case where bias drifts are present and the magnetometer is not used as a measurement update. This is due to the fact that the incidence angles have a large dependency on the attitude accuracy (via the quaternion transformation matrix). Since the magnetometer provides a measurement directly related to the attitude, this result was expected. It should be kept in mind that the bias drifts included in the simulations are an exaggeration of a real life scenario and magnetometer measurements should in general be available at all times. It is important to keep the uncertainty in the incidence angle estimates in mind when used for feedback in a controller design.

Table 3.1: EKF RMS State Errors

State / Scenario	1	2	3	Unit
Airspeed, V	0.22	0.22	0.28	ms^{-1}
Angle of attack, α	0.58	0.60	0.73	deg
Angle of side-slip, β	0.75	0.77	1.18	deg
Roll angle, ϕ	0.61	0.61	0.66	deg
Pitch angle, θ	0.54	0.54	0.71	deg
Yaw angle, ψ	0.69	0.73	1.00	deg
North displacement, N	0.74	0.74	0.75	m
East displacement, E	0.69	0.69	0.69	m
Altitude, h	0.56	0.56	0.56	m

[1] Filter updated with GPS and magnetometer measurements. Rate gyroscope bias instabilities not included in measurements.

[2] Filter updated with GPS and magnetometer measurements. Rate gyroscope bias instabilities included in measurements.

[3] Filter updated with GPS measurements only. Rate gyroscope bias instabilities included in measurements.

3.6 Summary

In this chapter, the development of a platform independent, kinematic state estimator was presented. The process was started off by establishing the concept behind the estimator, which was followed by a summary of the thought process behind optimal state estimation. The summary eventually led to the definition of a recursive EKF algorithm which can be used to perform optimal state estimation for non-linear systems.

Next, the initialisation of the estimator was discussed which introduced the TRIAD algorithm as a means to find the initial attitude of the platform. The EKF algorithm was then applied to the non-linear kinematic equations to estimate the respective states. Finally, it has been shown how the estimator can be used under certain assumptions to completely obtain the aircraft's state vector.

Chapter 4

Control and Simulation

This chapter focusses on the design of the controller implemented in this project. The chapter starts off by discussing some of the problems encountered when having to design controllers for non-linear systems. After having formulated a general idea of the solution, the discussion gets focused on the specific application to this project. Lastly, the adopted control strategy is evaluated with simulation results.

4.1 Overview and Strategy

The main task of the control system is to make the aircraft perform certain motions and manoeuvres autonomously. This is usually achieved using some form of feedback control. Feedback control also has the advantage that it desensitises the system to errors introduced during the modelling process, and reduces the effects of disturbances. Typical disturbances that need to be rejected include the centre of gravity shifting as a function of fuel level changes, non-zero static aerodynamic coefficients and wind effects.

As mentioned earlier, very few systems can be described by linear dynamics alone. Also, most of the techniques commonly used to design control systems, like root locus and pole placement, only cater for linear systems. This problem is usually accounted for by linearising the non-linear dynamics in the vicinity of some equilibrium point. Well known design techniques are then applied to the linearised system to design a linear control law. The control law remains valid as long as the the plant stays in the region where the linearisation was performed. This technique works well when designing autopilots for conventional flight as was illustrated by [35].

The specification for this project requires the autopilot to be able to perform a range of aerobatic manoeuvres. These maneouvres can require the aircraft to operate in regions which differ signif-

icantly from conventional flight conditions. Should the controller want to be designed using the entire aircraft state space, the technique mentioned above needs to be extended. Gain scheduling is a technique commonly adopted to deal with non-linear systems, and implies the design of a range of linear controllers at various equilibrium points. This method was not considered as an elegant solution to the problem at hand, and was not implemented in this project due to the following reasons,

- The process is time consuming. Some form of thought process needs to be applied to determine the respective equilibrium points. After this is accomplished, a range of linear controllers needs to be designed.
- The process yields a control law which is specific to the trajectory under study. When a new manoeuvre is introduced, the process needs to be repeated.
- Most importantly, since the feedback gains are only calculated at certain fixed equilibrium points, the process cannot be theoretically proven stable.

Receding-horizon control (RHC), more commonly known as model predictive control (MPC), are strategies capable of dealing with highly non-linear, multi-variable systems. The concept is not entirely new in that the process has been described in papers published as early as the 1980's. However, MPC strategies have not often been practically applied to fast systems [29] like for example an aerobatic aircraft.

The MPC strategy uses a mathematical plant model to predict the behaviour of certain states. The plant's behaviour is then used to calculate an optimal control input such that a cost function, either linear or non-linear, is minimised over a future horizon. Although computationally expensive, the process can also accommodate constraints imposed on the plant's state and control variables. This project makes use of a specific application of model predictive control to regulate the aircraft about a time varying trajectory. The task can be divided into two parts:

1. Off-line trajectory planning
2. On-line trajectory tracking

Off-line trajectory planning involves the specification/calculation of a feasible flight trajectory. The trajectory consists of the aircraft's state and control variables, related by the airframe's dynamic equations. The calculated control inputs can then be used to control the aircraft in an open loop fashion. However, should the nominal control inputs be erroneous, or the plant gets subjected to un-modelled disturbances, the aircraft will eventually diverge from the specified flight path.

On-line trajectory tracking is responsible for regulating the aircraft about the reference trajectory by continuously superimposing actuator perturbations on the nominal control inputs. The regulation is performed by firstly approximating the trajectory perturbation dynamics as a linear time variant (LTV) system. A cost function, quadratic in the state and control perturbations, is then minimised over a finite prediction horizon using a linear quadratic regulator (LQR) approach. This approach greatly reduces the complexity of the problem at hand since the solution is analytical.

4.2 Controller Setup

Figure 4.1 shows a block diagram representation of the non-linear aircraft model described by equations (2.59) to (2.77).

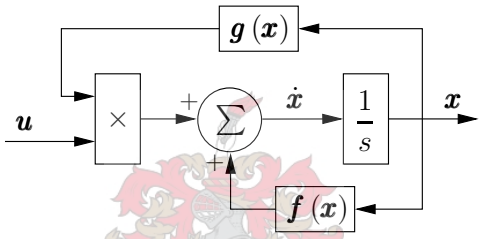


Figure 4.1: Aircraft Model

The state and control vectors are restated here for convenience, and also to define the units of the respective variables,

$$\boldsymbol{x} = [V \quad \alpha \quad \beta \quad p \quad q \quad r \quad q_1 \quad q_2 \quad q_3 \quad q_4 \quad N \quad E \quad h \quad T]^T \tag{4.1}$$

$$\boldsymbol{u} = [T_c \quad \delta_e \quad \delta_a \quad \delta_r]^T \tag{4.2}$$

The state variable units are defined as follows: The airspeed is given in meters per second, the incidence angles and angular rates have units of radians and radians per second respectively, the quaternion parameters are dimensionless, the position states are presented in meters, and the engine thrust is presented in Newtons. The control variable units are defined as follows: the commanded throttle is given in Newtons, while all control deflections have units of radians.

At this stage it will be assumed that for each of the aerobatic manoeuvres to be performed, nominal state and control trajectories, denoted by $\boldsymbol{x_n}$ and $\boldsymbol{u_n}$ respectively, are already in existence. The nominal trajectories are set up in such a way that they satisfy the aircraft dynamics encapsulated by equation (2.61). Perturbations from the nominal state and control trajectories

are denoted by $\boldsymbol{x_p}$ and $\boldsymbol{u_p}$ respectively, and are defined as follows,

$$\boldsymbol{x_p} = \boldsymbol{x} - \boldsymbol{x_n} \tag{4.3}$$

$$\boldsymbol{u_p} = \boldsymbol{u} - \boldsymbol{u_n} \tag{4.4}$$

The aim now is to obtain a set of dynamic equations describing perturbations from the nominal trajectories. The first step in achieving this goal is to calculate the time derivative of equation (4.3),

$$\dot{\boldsymbol{x}}_{\boldsymbol{p}} = \dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_{\boldsymbol{n}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\,\boldsymbol{u} - \dot{\boldsymbol{x}}_{\boldsymbol{n}} \tag{4.5}$$

Next, the non-linear functions in equation (4.5) are expressed as Taylor series expansions about the nominal trajectory,

$$\begin{aligned}\dot{\boldsymbol{x}}_{\boldsymbol{p}} = \boldsymbol{f}(\boldsymbol{x_n}) + \boldsymbol{g}(\boldsymbol{x_n})\,\boldsymbol{u_n} + \left[\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right]_{\boldsymbol{x_n}} (\boldsymbol{x} - \boldsymbol{x_n}) + \left[\frac{\partial \boldsymbol{g}(\boldsymbol{x})\,\boldsymbol{u}}{\partial \boldsymbol{x}}\right]_{\boldsymbol{x_n},\boldsymbol{u_n}} (\boldsymbol{x} - \boldsymbol{x_n}) \\ + \left[\frac{\partial \boldsymbol{g}(\boldsymbol{x})\,\boldsymbol{u}}{\partial \boldsymbol{u}}\right]_{\boldsymbol{x_n}} (\boldsymbol{u} - \boldsymbol{u_n}) - \dot{\boldsymbol{x}}_{\boldsymbol{n}} + \text{higher order terms}\end{aligned} \tag{4.6}$$

The perturbation dynamics are simplified by truncating the higher order terms of the Taylor series, and using the definitions stated in equations (4.3) and (4.4),

$$\dot{\boldsymbol{x}}_{\boldsymbol{p}} \approx \left(\left[\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right]_{\boldsymbol{x_n}} + \left[\frac{\partial \boldsymbol{g}(\boldsymbol{x})\,\boldsymbol{u}}{\partial \boldsymbol{x}}\right]_{\boldsymbol{x_n},\boldsymbol{u_n}}\right) \boldsymbol{x_p} + \boldsymbol{g}(\boldsymbol{x_n})\,\boldsymbol{u_p} + [\boldsymbol{f}(\boldsymbol{x_n}) + \boldsymbol{g}(\boldsymbol{x_n})\,\boldsymbol{u_n} - \dot{\boldsymbol{x}}_{\boldsymbol{n}}] \tag{4.7}$$

Inspecting equation (4.7) closely it is noted that the terms in square brackets reflect the aircraft dynamic equations evaluated at the nominal trajectory. According to the definition of a nominal trajectory, these terms should sum to zero. However, due to the nature in which nominal trajectories will be obtained in this project, the terms may not quite sum to zero. The residue can then simply be interpreted as a small disturbance that needs to be rejected by the control system. Discretising the nominal trajectories, equation (4.7) can be rewritten as follows,

$$\dot{\boldsymbol{x}}_{\boldsymbol{p}} \approx \boldsymbol{F_k}\boldsymbol{x_p} + \boldsymbol{G_k}\boldsymbol{u_p} + \boldsymbol{d} \tag{4.8}$$

where $\boldsymbol{d}$ reflects disturbances due to trajectories which do not completely satisfy the aircraft dynamic equations, and $\boldsymbol{F_k}$ and $\boldsymbol{G_k}$ are defined as follows,

$$\boldsymbol{F_k} = \left[\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right]_{\boldsymbol{x_n}(k)} + \left[\frac{\partial \boldsymbol{g}(\boldsymbol{x})\,\boldsymbol{u}}{\partial \boldsymbol{x}}\right]_{\boldsymbol{x_n}(k),\boldsymbol{u_n}(k)} \tag{4.9}$$

$$\boldsymbol{G_k} = \boldsymbol{g}(\boldsymbol{x_n}(k)) \tag{4.10}$$

Next, the continuous system is converted to the discrete time domain making use of equations (3.20) and (3.21) and applying the same assumptions,

$$\boldsymbol{\Phi_p}(k) \approx \boldsymbol{I} + \boldsymbol{F_k} T_s \tag{4.11}$$

$$\boldsymbol{\Gamma_p}(k) \approx \boldsymbol{G_k} T_s \tag{4.12}$$

Equations (2.35) to (2.40) show that the aerodynamic forces and moments are all related to the square of the airspeed. This provides an argument to want to be able to regulate the airspeed with zero steady state error. Also, when the engine model was presented, it was pointed out that the change in thrust with changes in airspeed had been neglected. Lastly, the parasitic drag coefficient, C_{D_0}, is calculated using empirical formulas only (see Section B.4.3), and subsequently its accuracy should be considered dubious. Considering all these facts, the decision was made to include integrator dynamics on the airspeed error. This will result in a zero steady error during trim flight and also push out disturbances originating from the engine approximation and parasitic drag coefficient calculation. When the simulation results are presented, one can expect to see steady state errors on other motion variables when the aircraft is made to fly ill-conditioned trajectories or subjected to disturbances. The output matrix, $\boldsymbol{H}$, was set up to extract the airspeed from the perturbation state vector and is given by,

$$\boldsymbol{H} = [1 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0] \tag{4.13}$$

Next, the integrator state is augmented into the discrete dynamic and input matrices as follows,

$$\boldsymbol{\Phi'_p}(k) = \begin{bmatrix} 1 & \boldsymbol{H} \\ \boldsymbol{0} & \boldsymbol{\Phi_p}(k) \end{bmatrix} \tag{4.14}$$

$$\boldsymbol{\Gamma'_p}(k) = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{\Gamma_p}(k) \end{bmatrix} \tag{4.15}$$

The discrete linearised difference equation describing the perturbation dynamics along a reference trajectory is given by,

$$\boldsymbol{x'_p}(k+1) = \boldsymbol{\Phi'_p}(k)\,\boldsymbol{x'_p}(k) + \boldsymbol{\Gamma'_p}(k)\,\boldsymbol{u'_p}(k) \tag{4.16}$$

Having arrived at this point, a mathematical model has been derived which describes the perturbation dynamics along any feasible trajectory. The focus is now shifted onto how an LQR approach can be used to regulate the aircraft about the respective trajectories. The biggest advantage of using the LQR is that it has an analytical solution. However, the process also has other beneficial properties which contributed towards its selection,

- It provides a way to handle LTV, Multi Input Multi Output (MIMO) systems.
- The process has the property that it aims to regulate state and control variables to zero. This is exactly what is required since the dynamics are set up in terms of perturbations.

The LQR uses a performance index which is quadratic in the state and control perturbations and is defined as follows,

$$J = \frac{1}{2}\sum_{k=0}^{N}\left[\boldsymbol{x_p'}\left(k\right)^T \boldsymbol{Q_1}\left(k\right)\boldsymbol{x_p'}\left(k\right) + \boldsymbol{u_p'}\left(k\right)^T \boldsymbol{Q_2}\left(k\right)\boldsymbol{u_p'}\left(k\right)\right] \tag{4.17}$$

where $\boldsymbol{Q_1}\left(k\right)$ and $\boldsymbol{Q_2}\left(k\right)$ are the state and control perturbation weight matrices respectively. Given an initial condition, $\boldsymbol{x_p'}\left(k=0\right)$, an optimal control law, $\boldsymbol{u_p'}\left(k\right)$, is defined as the input which minimises the quadratic performance index over the period $k = 0...N$. Equations (4.16) and (4.17) form a constrained minima problem which can be solved by making use of Lagrange multipliers [7]. The solution is presented in the form of a backward difference equation known as the discrete matrix Riccati equation. The perturbation control input is defined as,

$$\boldsymbol{u_p'}\left(k\right) = -\boldsymbol{K}\left(k\right)\boldsymbol{x_p'}\left(k\right) \tag{4.18}$$

where

$$\boldsymbol{M}\left(k\right) = \boldsymbol{S}\left(k\right) - \boldsymbol{S}\left(k\right)\left[\boldsymbol{Q_2}\left(k\right) + \boldsymbol{\Gamma_p'}\left(k\right)^T \boldsymbol{S}\left(k\right)\boldsymbol{\Gamma_p'}\left(k\right)\right]^{-1}\boldsymbol{\Gamma_p'}\left(k\right)^T \boldsymbol{S}\left(k\right) \tag{4.19}$$

$$\boldsymbol{S}\left(k-1\right) = \boldsymbol{\Phi_p'}\left(k\right)^T \boldsymbol{M}\left(k\right)\boldsymbol{\Phi_p'}\left(k\right) + \boldsymbol{Q_1}\left(k\right) \tag{4.20}$$

$$\boldsymbol{K}\left(k-1\right) = \left[\boldsymbol{Q_2}\left(k\right) + \boldsymbol{\Gamma_p'}\left(k\right)^T \boldsymbol{S}\left(k\right)\boldsymbol{\Gamma_p'}\left(k\right)\right]^{-1}\boldsymbol{\Gamma_p'}\left(k\right)^T \boldsymbol{S}\left(k\right)\boldsymbol{\Phi_p'}\left(k\right) \tag{4.21}$$

subject to the initial conditions,

$$\boldsymbol{S}\left(N\right) = \boldsymbol{Q_1}\left(N\right) \tag{4.22}$$

$$\boldsymbol{K}\left(N\right) = \boldsymbol{0} \tag{4.23}$$

Inspecting the equations closely, some interesting properties are noticed about the solution. Although it was never assumed, the control law implies full state feedback. Also, the order of the matrix inversion to be performed is equal to the number of control inputs. This must be kept in mind should the equations want to be solved in real time. The initial condition stated in equation (4.23) makes intuitive sense considering the performance index to be minimised. Setting $\boldsymbol{u_p'}\left(N\right) = \boldsymbol{0}$ does not increase the performance index and only influences $\boldsymbol{x_p'}\left(N+1\right)$, which falls outside the optimization window. Finally, the solution only yields a perturbation control input. Using equation (4.4), the total control input to be applied to the aircraft can be obtained,

$$\boldsymbol{u}\left(k\right) = \boldsymbol{u_n}\left(k\right) + \boldsymbol{u_p}\left(k\right) \tag{4.24}$$

Equation (4.24) shows that the control input consists of two terms. The first term provides a strong feed-forward component, obtained from the nominal trajectory. The second term provides feedback of state perturbations in order to keep the aircraft in the vicinity of the state trajectory. To help establish the working behind the adopted MPC strategy, consider Figure 4.2 which shows the position information of a loop manoeuvre.

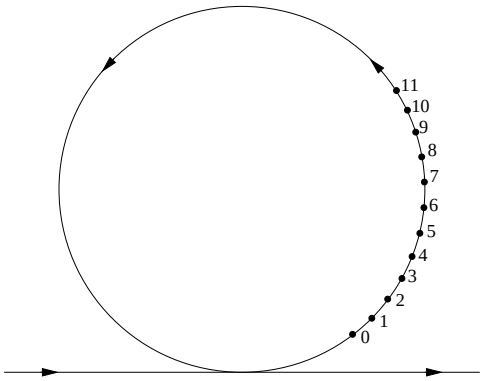


Figure 4.2: Nominal Loop Trajectory to Demonstrate MPC Concept

Assume the aircraft is currently on the trajectory at point 0 and ready to apply some form of control signal. The task of the MPC is to construct a control signal, given future knowledge of the trajectory between points 0 and 9, to best keep the aircraft in the vicinity of the nominal trajectory.

Firstly, denote the trajectory between points 0 and 9 as the prediction horizon. The non-linear aircraft model is linearised about the discrete trajectory to obtain the perturbation dynamics. The linear models are then used to solve the Riccati equation backwards from point 9 to 0, resulting in a whole range of feedback gains. Only the first set of feedback gains, $\boldsymbol{K}(0)$, is used to construct the feedback part of the control law to be applied at point 0. The rest of the solution is discarded. The final control input is then calculated by summing the feedback and feed-forward control signals. This introduces the receding horizon concept in that every time a control law is calculated, a fixed amount of the future is taken into account. Figure 4.3 shows a block diagram representation of the final controller configuration.

For clarity, the first set of feedback gains in the solution to the Riccati equations have been defined as follows,

$$\boldsymbol{K}(0) = \begin{bmatrix} \boldsymbol{K_I}'(0) & \boldsymbol{K}'(0) \end{bmatrix} \tag{4.25}$$

where $\boldsymbol{K_I}'(0)$ is a 4×1 matrix containing the gains used to feed the integrated airspeed error back to the four actuators. $\boldsymbol{K}'(0)$ is a 4×14 matrix containing the gains used to feed state perturbations back to the respective actuators. Having arrived at this point, the mathematical setup of the controller is complete. Appendix C contains additional information specific to

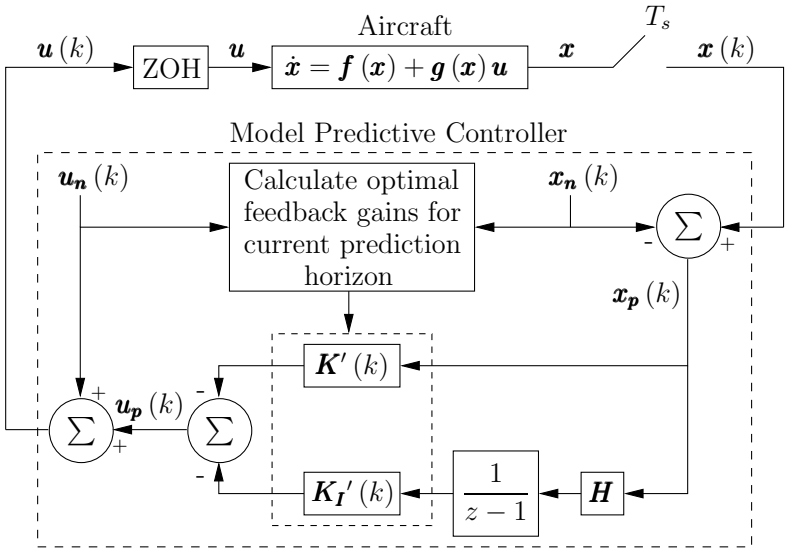


Figure 4.3: Model Predictive Controller Configuration

the implementation. To finalise the controller setup, the following sections address issues like trajectory specification, prediction horizon length, state and control weight matrices, and the thought behind the position governor implemented.

4.2.1 Trajectory Specification

Listed below are four possible methods that could be used to obtain nominal trajectories for the respective aerobatic manoeuvres,

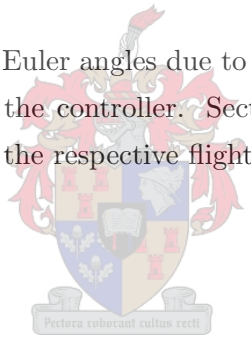
1. Mathematically using optimal control techniques.
2. Recording aircraft states and actuator commands while a safety pilot is flying the manoeuvres.
3. Recording aircraft states and actuator commands while flying the manoeuvres using the mathematical aircraft model developed in Chapter 2.
4. Applying intuitive sense.

Option 1 is attractive since it is capable of handling state and actuator constraints, and the solution by definition satisfies the aircraft dynamics. However, the process is complex and falls beyond the scope of this project. Options 2 and 3 yield trajectories which approximately and completely satisfy the aircraft dynamics respectively. The drawback of these two methods is the fact that the results are dependant on how well a human pilot can fly the respective manoeuvres.

Since the aerobatic manoeuvres attempted in this project are relatively simply, option 4 also becomes a feasible solution. It was established that it is possible to specify all the trajectories by simply applying some intuitive sense. The drawback of this option is that the trajectories will be somewhat ill-conditioned. However, it has previously been argued that ill-conditioned trajectories can be seen as modelling disturbances to be rejected by the control system. Subsequently, option 4 was adopted as the preferred method to specify trajectories in this project. A trajectory consists of the following nominal state and actuator information:

- Airspeed and incidence angles
- Angular velocities
- Attitude in terms of Euler angles
- North, East and altitude displacements
- Thrust, commanded thrust, and elevator, aileron and rudder deflections

The attitude is specified in terms of Euler angles due to the ease of visualisation, but converted to quaternions just before entering the controller. Section A.2 describes how this conversion was performed. The trajectories for the respective flight envelopes will be introduced when the simulation results are presented.



4.2.2 Prediction Horizon

As already stated, the definition of the prediction horizon is the amount of the foreseeable future taken into account to calculate the current perturbation control input. Keeping this in mind, it comes naturally to want to make the horizon as long as possible. Inspecting the Riccati equations, it is clear that the length of the horizon is directly related to the amount of computational power required to obtain the solution. This is a hard constraint when having to solve the equations in real time.

The horizon length used in this project was determined doing benchmark tests on the hardware platform described in Chapter 5. Considering the controller is running at 50 Hz, a maximum of 20 ms real time processing power is available to find the required solution. The horizon length was set at 0.2 seconds, which implies 10 discrete sample instances. Although quite short, it is the same horizon length used by [31] to successfully demonstrate the control of a highly non-linear manoeuvre in the form of a high-amplitude velocity vector roll. This is the major drawback of using a complete non-linear MPC strategy in that the solution is expensive to solve and most often results in short prediction horizons.

4.2.3 State and Control Weights

The state and control perturbation weight matrices were not implemented as time varying in this project. The matrix entries were predetermined and remained fixed for the different manoeuvres. The state perturbation weight matrix, $\boldsymbol{Q_1}$, is a 15×15 matrix with the individual state weights on its diagonal and is defined as follows,

$$\boldsymbol{Q_1} = \text{diag}\left(\begin{bmatrix} Q_{\int V} & Q_V & Q_\alpha & Q_\beta & Q_p & Q_q & Q_r & Q_{q_1} & Q_{q_2} & Q_{q_3} & Q_{q_4} & Q_N & Q_E & Q_h & Q_T \end{bmatrix}\right) \quad (4.26)$$

The control perturbation weight matrix, $\boldsymbol{Q_2}$, is a 4×4 matrix with the individual actuator weights on its diagonal and is defined as follows,

$$\boldsymbol{Q_2} = \text{diag}\left(\begin{bmatrix} Q_{T_c} & Q_{\delta_e} & Q_{\delta_a} & Q_{\delta_r} \end{bmatrix}\right) \quad (4.27)$$

The system designer is left with the task of choosing the respective weights. This usually involves some kind of iterative process until the desired system response is obtained. To help the iterative process converge, it is important to understand the system dynamics involved and how the various weights influence the cost function. A simple way to interpret the cost function is to see it as a representation of the system's energy. Increasing $\boldsymbol{Q_1}$ will penalise the state perturbations more heavily and will result in greater amounts of control energy being used to drive the respective deviations to zero. The designer expects to see an increase in system bandwidth. Increasing $\boldsymbol{Q_2}$ will restrict the use of control energy and decrease the bandwidth. These are the basic concepts that need to be considered when setting up the respective weights. Another important aspect to consider is the quality of the estimated state variables used to calculate the state perturbations.

A maximum state and control deviation approach, as introduced by [7] and [12], was adopted to find initial values for the weights. The process involves setting the weights equal to the square of the inverse of the maximum allowed deviations. Another degree of freedom is obtained by introducing a scaling factor between the two matrices. It was established that it is possible for the MPC to regulate the aircraft about the various state trajectories with acceptable accuracy, while using only one set of state and control weights. The state perturbation weights, together with their respective units in square brackets, are given below,

$$Q_{\int V} = \frac{1}{4^2} \quad \left[\frac{1}{\text{m}^2}\right], \quad Q_V = \frac{1}{0.3^2} \quad \left[\frac{1}{\text{m}^2\text{s}^{-2}}\right], \quad Q_\alpha = 0 \quad \left[\frac{1}{\text{rad}^2}\right],$$

$$Q_\beta = 0 \quad \left[\frac{1}{\text{rad}^2}\right], \quad Q_p = \frac{1}{\left(5\frac{\pi}{180}\right)^2} \quad \left[\frac{1}{\text{rad}^2\text{s}^{-2}}\right], \quad Q_q = \frac{1}{\left(3\frac{\pi}{180}\right)^2} \quad \left[\frac{1}{\text{rad}^2\text{s}^{-2}}\right],$$

$$Q_r = \frac{1}{\left(10\frac{\pi}{180}\right)^2} \quad \left[\frac{1}{\text{rad}^2\text{s}^{-2}}\right], \quad Q_{q_1} = \frac{1}{0.007^2} \quad [\text{ND}], \quad Q_{q_2} = \frac{1}{0.007^2} \quad [\text{ND}],$$

$$Q_{q_3} = \frac{1}{0.007^2} \quad [\text{ND}], \quad Q_{q_4} = \frac{1}{0.007^2} \quad [\text{ND}], \quad Q_N = \frac{1}{0.07^2} \quad \left[\frac{1}{\text{m}^2}\right],$$
$$Q_E = \frac{1}{0.07^2} \quad \left[\frac{1}{\text{m}^2}\right], \quad Q_h = \frac{1}{0.07^2} \quad \left[\frac{1}{\text{m}^2}\right], \quad Q_T = 0 \quad \left[\frac{1}{\text{N}^2}\right],$$

where ND stands for non-dimensional. The control perturbation weights are given by,

$$Q_{T_c} = \frac{2}{5^2} \quad \left[\frac{1}{\text{N}^2}\right], \quad Q_{\delta_e} = \frac{2}{\left(0.8\frac{\pi}{180}\right)^2} \quad \left[\frac{1}{\text{rad}^2}\right], \quad Q_{\delta_a} = \frac{2}{\left(3\frac{\pi}{180}\right)^2} \quad \left[\frac{1}{\text{rad}^2}\right],$$
$$Q_{\delta_r} = \frac{2}{\left(0.5\frac{\pi}{180}\right)^2} \quad \left[\frac{1}{\text{rad}^2}\right],$$

Most of the values were found by continuously changing certain weight parameters, and evaluating the controller's performance while regulating about the different trajectories. Even so, the final results still make some intuitive sense. An initial weight for the airspeed perturbation state was obtained by taking the estimated state's accuracy into account. The integrated airspeed error state weight was determined iteratively. The decision to not weigh the incidence angles were based on their relatively inaccurate estimates, and also the fact that the states are estimated under no wind assumptions. Also, the nominal incidence angles are difficult to specify using only intuitive sense. Weighing the angular rate states provides some stability augmentation by increasing the natural damping of the aircraft. The weights also show a large emphasis placed on the aircraft's attitude. This is firstly due to the good attitude estimates obtained from the state estimator. Secondly, being able to control the aircraft's attitude results in direct control of its acceleration. This is beneficial since the acceleration ultimately filters down to the velocity and position states.

In order to understand the large position state weights, imagine the aircraft flying a straight and level trajectory, but with a constant altitude error. To be able to remain in this state, the airspeed, angle of attack and pitch angle states must be regulated with relatively small perturbations. This again implies the LQR cost function is at a minimum when only considering these states. Correcting the error in altitude would imply having to disturb the airspeed, angle of attack and pitch angle states by pitching the aircraft's nose 'upwards'. For the LQR to be willing to do this, perturbations in the altitude state would have to be penalised heavily.

4.2.4 Position Governor

Up till now very little has been said about the influences of physical constraints. The aircraft is limited in the amounts of available actuation, and various state constraints also need to be considered. Since the specified trajectories will be largely based on intuitive sense, a scenario

will inevitably arise where some actuator saturation occurs. Accompanying the MPC with a position governor helps to alleviate some of the problems associated with actuator saturation. To help explain the concept, consider Figure 4.4.

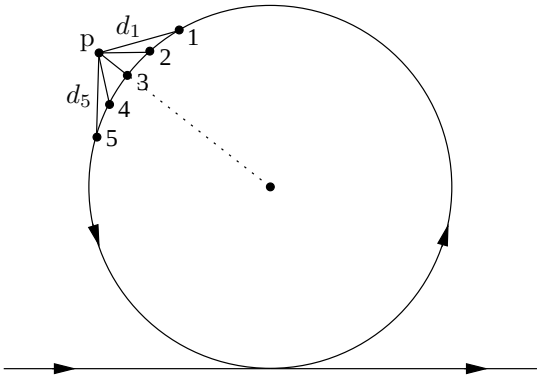


Figure 4.4: Nominal Loop Trajectory to Demonstrate Position Governor Concept

The controller applied some control at point 1. Due to possible actuator saturation and disturbances, the aircraft currently finds itself at point p. Without a position governor, the controller will simply move on to point 2, define the prediction horizon from point 2 onwards, and calculate the respective gains. It is obvious to see that should such a situation repeat itself, the actual aircraft position, and where the controller expects the aircraft to be, will diverge. The position governor uses the estimated aircraft position to index the nominal trajectory using a simple algorithm. Starting off at the previous index (1 in this case), a predefined number of distances, denoted by d_1 to d_5, are calculated. The current index to be used is then defined as the distance associated with the shortest distance. In short, the position governor has the sole task of ensuring the correct part of the trajectory is used as prediction horizon.

Having arrived at this stage, all the different aspects of the model predictive controller have been addressed. The following section demonstrates the working of the concept by means of simulation results.

4.3 Simulation Results

This section introduces the various manoeuvre trajectories and the accompanying simulation results. Unless otherwise stated, all sensor data used for feedback and in the EKF process were corrupted with noise sources as previously discussed. Wind gusts were also injected into the system to evaluate the controller's ability to reject disturbances. When the various trajectories are specified, all states coordinated in the inertial reference frame are defined in the North-Down plane to help emphasise the working of the controller. However, the trajectories can easily be

rotated and displaced to obtain reference trajectories defined in any heading direction, and at any point in inertial space. The following limitations, obtained from practical measurements and considering the validity of the aerodynamics model, were imposed on the various actuator variables,

- Maximum commanded thrust of 60 Newtons
- Maximum elevator deflection of ±15 degrees
- Maximum aileron deflection of ±15 degrees
- Maximum rudder deflection of ±20 degrees

4.3.1 Conventional Flight

A straight and level trajectory was defined to help obtain the initial values for the state and control weight matrices, as well as ensuring the correct working of the controller. The nominal airspeed, angle of attack, commanded thrust and elevator deflection for the trajectory are the trim values stated in equations (B.20) to (B.23). For straight and level flight, the nominal angle of side-slip, body angular velocities, roll and yaw angles, and the aileron and rudder deflections are all set to zero. Because reference line body axes are used, the angle of attack and pitch angle are equal during trim flight. Having set the the yaw angle to zero and taking the no wind assumption into account, the North displacement can be obtained by integrating the airspeed state. The zero yaw angle also implies that the East displacement goes to zero for the entire trajectory. The trajectory is completed by setting the altitude state to 150 meters.

Figure 4.5 shows the simulated airspeed, angle of attack, altitude, commanded thrust and elevator deflection for the trajectory. The graphs on the left hand side show the controller's performance without any sensor noise and wind disturbances, whilst the right hand side graphs demonstrate the performance with sensor noise and wind gusts present. The gusts had magnitudes in the order of 1 m/s and lasted for approximately 2 seconds. The nominal state and control values are shown by the dashed lines.

The figure firstly shows a zero steady state error on the airspeed. This result was expected due to the augmented integral control on the state. The un-augmented altitude state suffers from a steady state error in the order of 0.5 meters, and can be contributed to a slight error in the trim condition calculation. However, considering the altitude of operation of the aircraft, this is perfectly acceptable. The sub-figures in the last two rows show the nominal and actual commanded thrust and elevator deflection. The difference between the nominal and actual values are the control perturbations commanded by the MPC. Lastly, the figure indicates the

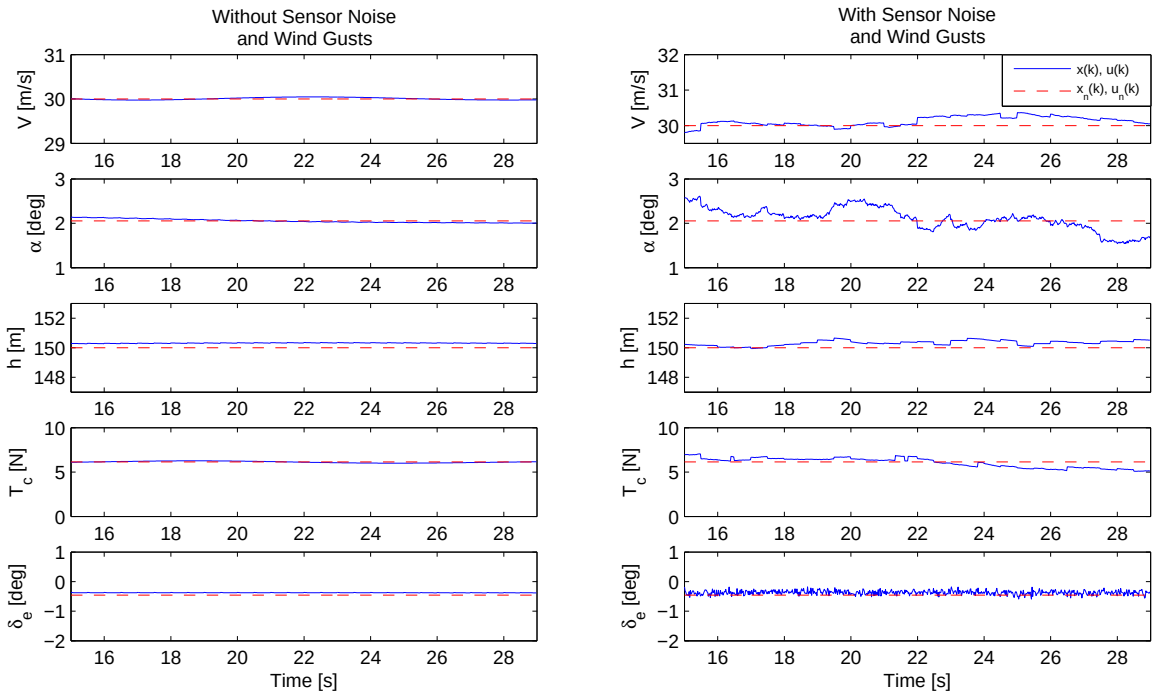


Figure 4.5: Simulation Results for Level Flight Trajectory

controller's ability to regulate the angle of attack even under constant wind gust disturbances. This instils some confidence in the regulation process for when more complex trajectories are attempted.

4.3.2 Aileron Roll

The aileron roll is the first aerobatic manoeuvre presented, and is relatively simple to perform from a controller's point of view. The trajectory is divided into three sections. The first and last sections correspond to straight and level flight, and subsequently were defined in exactly the same manner as discussed in the previous section. The roll was specified to be performed in 2 seconds, corresponding to a roll rate of 180°/s. Since the aircraft enters and leaves the roll flying straight and level, the angle of attack before and after the manoeuvre should correspond to the trim angle of attack. When inverted, the angle of attack was set to the negative of the trim angle in order to provide the required lift force. The normal and negated angles were simply interpolated to obtain the values used during the roll. Although based on very rough assumptions, this approach was adopted since it at least yields zero angle of attack values when the aircraft's roll angle is ±90°. This is beneficial since it results in zero lift forces at these points which is required should the roll want to be performed in a straight line.

The angle of side-slip was set to zero since it is difficult to get an intuitive feel for the state. The roll angle was obtained by integrating the roll rate, while the pitch and yaw angles were set to

zero. The nominal aileron and elevator deflections for the roll, both directly related to the roll rate and angle of attack respectively, were determined by experiment. Lastly, the altitude was once again set to 150 meters. Figure 4.6 shows the simulated roll rate, roll angle, altitude and angle of attack states for the trajectory. The commanded aileron and elevator deflections are shown in the bottom two sub-figures.

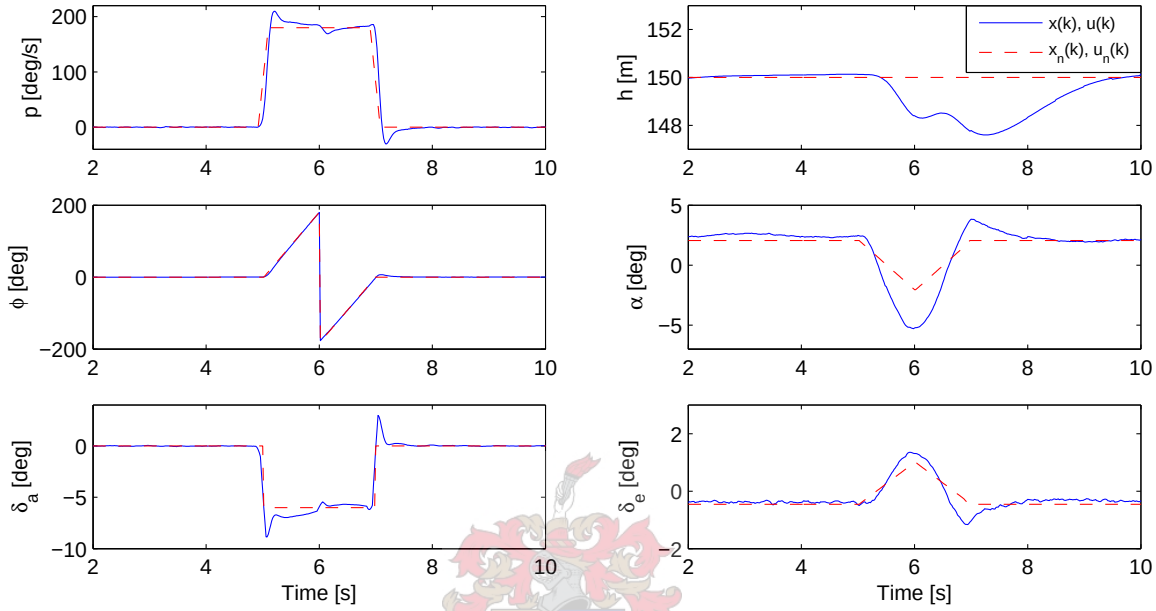


Figure 4.6: Simulation Results for Aileron Roll Trajectory

The figure firstly shows that for the paths leading in and out of the roll, the controller regulates the respective states extremely well about the nominal trajectory. At 5 seconds the roll is performed, evident in the roll rate and roll angle states. The 180°/s roll rate step commands are completed in roughly 0.5 seconds. Next, the figure shows a 2 meter drop in altitude during the roll. This is a common scenario when roll manoeuvres are performed, and originates from the loss in lift during certain stages of the manoeuvre. The loss in altitude was deemed acceptable since compensation for the altitude loss would complicate the trajectory specification. The assumptions made to specify the nominal angle of attack and elevator deflection states are highlighted by the deviations between the nominal and actual states. However, the approximations seem to be adequate to perform the manoeuvre with good accuracy. Lastly, the figure shows the nominal and commanded aileron deflection which is well within the physical limits of the aircraft.

4.3.3 Loop

The construction of the aileron roll trajectory required little intuitive sense as it was possible to specify a set of well related states describing the aircraft's motion throughout the manoeuvre. With the loop trajectory this is not so easily achieved, and is mostly due to the fact that saturation of the throttle actuator needs to be considered. In order to simplify the setup procedure, various assumptions were made about the trajectory.

The trajectory is once again divided into three parts with the first and last parts corresponding to straight and level flight. In order to keep the aerodynamic model valid, it is paramount for the angle of attack to not deviate too far from the value used to calculate the stability and control derivatives. Keeping this in mind, the loop radius, denoted by R_L, was set at 35 meters. For the first half of the loop, the airspeed magnitude was set at 30 m/s. The increase in airspeed during the second half was determined by experiment. Having defined the airspeed and radius, the pitch rate can be calculated using the equation,

$$q = \frac{V}{R_L} \tag{4.28}$$

The pitch angle can then be obtained by integrating the pitch rate, and applying the required wrapping around of the integration result. In order to calculate the angle of attack, a slightly more sophisticated approach is required. Consider Figure 4.7 which approximates the aircraft as a point mass moving around the loop trajectory.

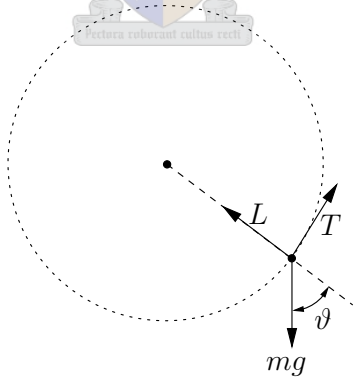


Figure 4.7: Aircraft as a Point Mass Moving Around a Loop Trajectory

Only the thrust, lift and weight vectors act on the point mass, and the lift and thrust vectors act perpendicular to one another. The figure also assumes that the aircraft's lift vector, denoted by L, always points towards the centre of the loop. The variable, ϑ, was introduced to make the derivations to follow easier, and is defined as the integral of the pitch rate while in the loop. The lift vector is responsible for providing the required centrifugal acceleration to keep the point mass on the circle, as well as cancelling out the component of the weight vector acting parallel

to the lift vector. Expressed in mathematical terms this becomes,

$$L = \frac{1}{2}\rho V^2 C_{L_\alpha}\alpha = m\frac{V^2}{R_L} + mg\cos\vartheta \tag{4.29}$$

Making the angle of attack the subject of equation (4.29), a formula is derived which approximates the angle of attack as a function of the airspeed and integrated pitch rate,

$$\alpha\left(V,\vartheta\right) = \frac{2m\left(\dfrac{V^2}{R_L} + g\cos\vartheta\right)}{\rho V^2 C_{L_\alpha}} \tag{4.30}$$

The angle of side-slip together with the roll and yaw rates were set to zero for the entire trajectory, while the roll and yaw angles were defined by applying some intuitive sense. Having obtained the pitch angle earlier, simple equations were set up to calculate the nominal North and altitude states. A bang-bang approach was used to approximate the commanded thrust after which the engine thrust state was calculated using the engine dynamics described by equation (2.34). Lastly, the nominal elevator deflection, related to the nominal angle of attack, was determined by experiment whilst the nominal aileron and rudder deflections were set to zero. Figure 4.8 shows the simulated airspeed, angle of attack and pitch angle states for the trajectory. Also included are the commanded thrust and elevator deflections and the trajectory index.

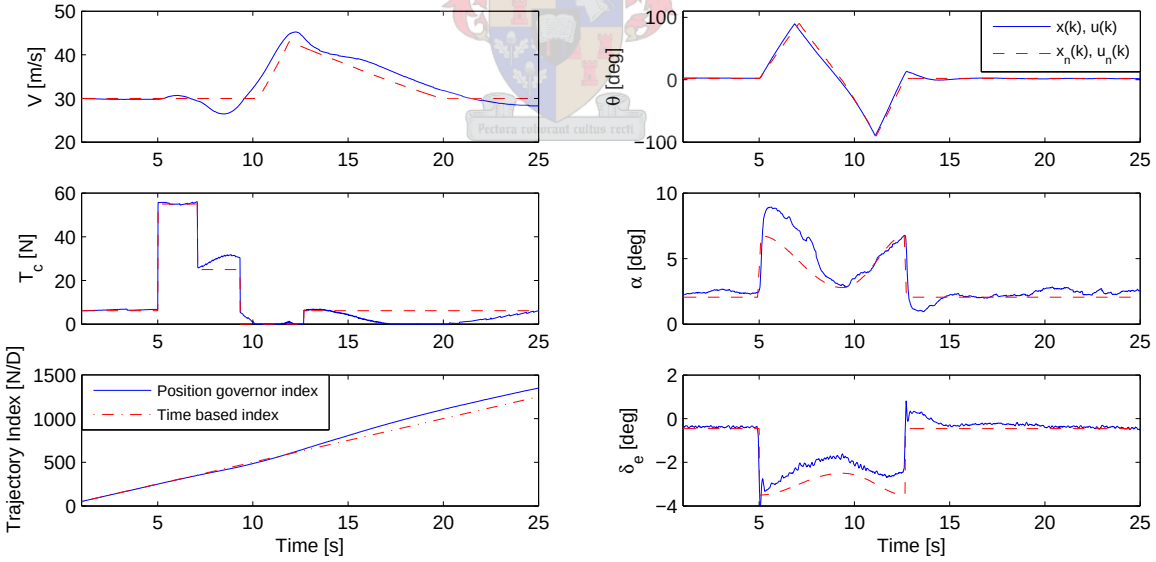


Figure 4.8: Simulation Results for Loop Trajectory

The sub-figure showing the simulated and nominal pitch angle once again emphasises the controller's ability to regulate the aircraft's attitude with good accuracy. The figure also indicates that the aircraft entered the loop after 5 seconds and completed the manoeuvre 8 seconds later. Between 5 and 9 seconds, the aircraft is in the first half of the loop. During this period the

nominal commanded thrust is stepped up twice to best keep the airspeed constant at 30 m/s. Next, the thrust gets cut completely and the aircraft is left to glide down the second half of the loop.

The controller's inability to regulate the airspeed with zero steady state error between 9 and 20 seconds can be contributed to the throttle saturation experienced during the period. This is a typical scenario where it would be required to index the reference trajectory using a position governor. The sub-figure in the bottom left corner demonstrates the working of the position governor by comparing the governor's index with its time based counter part. For as long as the throttle saturates at its minimum value, the position governor index increasingly leads the time based version. Much more time can be invested to set up a better conditioned trajectory which might result in no throttle saturations. However, if no proper analytical tools are available, such an approach might turn out to be a time consuming task. The simulation results show that even though its not possible to regulate the airspeed perfectly throughout the entire manoeuvre, the controller still ensures the stability of the aircraft. Again, the assumptions made to specify the nominal angle of attack and elevator deflection states are exposed by the respective graphs. To obtain a better visual representation of the controller's performance, consider Figure 4.9 which plots the aircraft's altitude versus its North displacement.

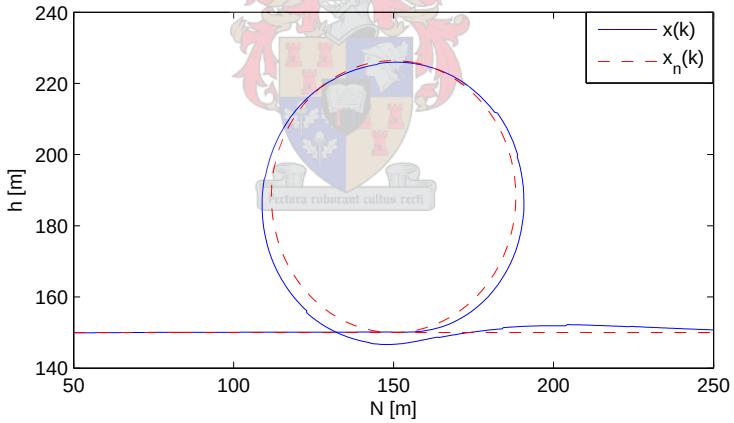


Figure 4.9: Altitude versus North Displacement for Loop Trajectory

The figure indicates a steady state error on the order of 3 meters around the loop. This can be contributed to a number of factors. The absence of integral control on the position states, state weights, and the intuitively specified nominal trajectory being some of them. However, considering the relatively good shape of the loop, and the altitude at which the manoeuvre would typically be performed, the response was deemed acceptable.

4.3.4 Immelmann

The Immelmann, more commonly known as a reverse Split-S, is a combination of the loop and aileron roll manoeuvres demonstrated thus far. Flying straight and level, the aircraft enters a constant radius loop. Half way through the loop, while flying upside down, a 180° roll is performed resulting in the aircraft flying in the opposite direction at an increased altitude. Due to its similarities with the loop and aileron roll manoeuvres, the trajectory setup is not discussed here. Figure 4.10 shows the simulated pitch and roll angles, airspeed, commanded thrust, and elevator and aileron deflections.

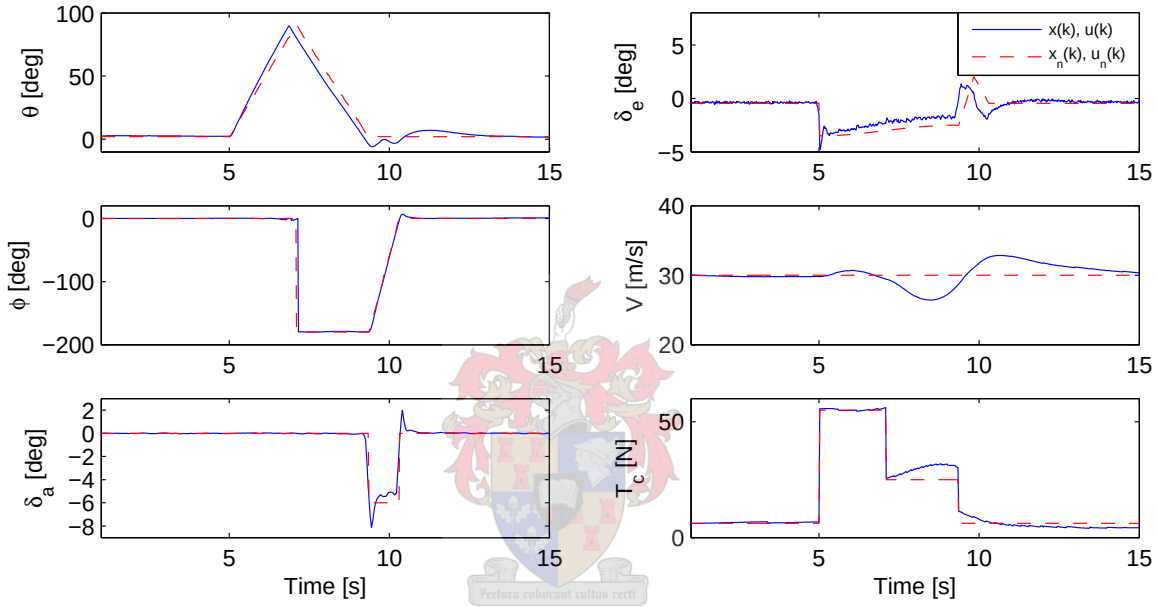


Figure 4.10: Simulation Results for Immelmann Trajectory

The figure indicates that the half loop is completed after 9 seconds. This is followed by the aircraft performing the required roll in 1 second to end up in straight and level flight. Figure 4.11 shows the aircraft's altitude versus its North displacement. The figure again shows the drop in altitude as the 180° roll is performed, and can be contributed to the reasons mentioned when the aileron roll trajectory was presented. The Immelmann concludes the aerobatic manoeuvres presented in this project. However, due to the adopted control strategy, the array of manoeuvres can be extended by simply specifying more nominal trajectories and would in most likelihood require little or no modifications to the controller configuration.

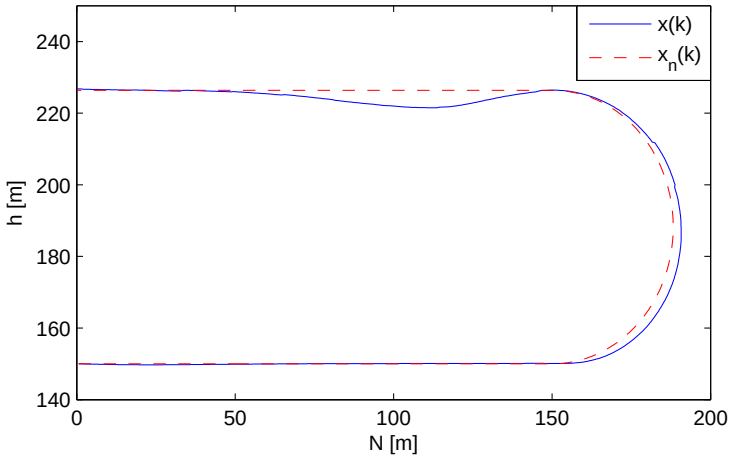


Figure 4.11: Altitude versus North Displacement for Immelmann Trajectory

4.4 Conclusion

This chapter presented the controller design and simulation results for the project. A model predictive control strategy was adopted due to its ability to deal with highly non-linear systems. The controller's versatility has also been demonstrated by performing a range of aerobatic manoeuvres.

The concept behind the adopted MPC strategy was formulated, followed by its specific application to the aircraft model developed in Chapter 2. Lastly, simulation results were presented to demonstrate the controller's ability to perform conventional flight, as well as various aerobatic manoeuvres in the form of an aileron roll, a loop and a reverse Split-S.

This chapter concludes the theoretical design of the autopilot. The focus is now shifted to the hardware used to implement and test the solutions presented thus far. These topics are the subject of the next chapter.

Chapter 5

Avionics and Ground Station

The avionics and ground station design are the topics of this chapter. An overview of the system is shown in Figure 5.1.

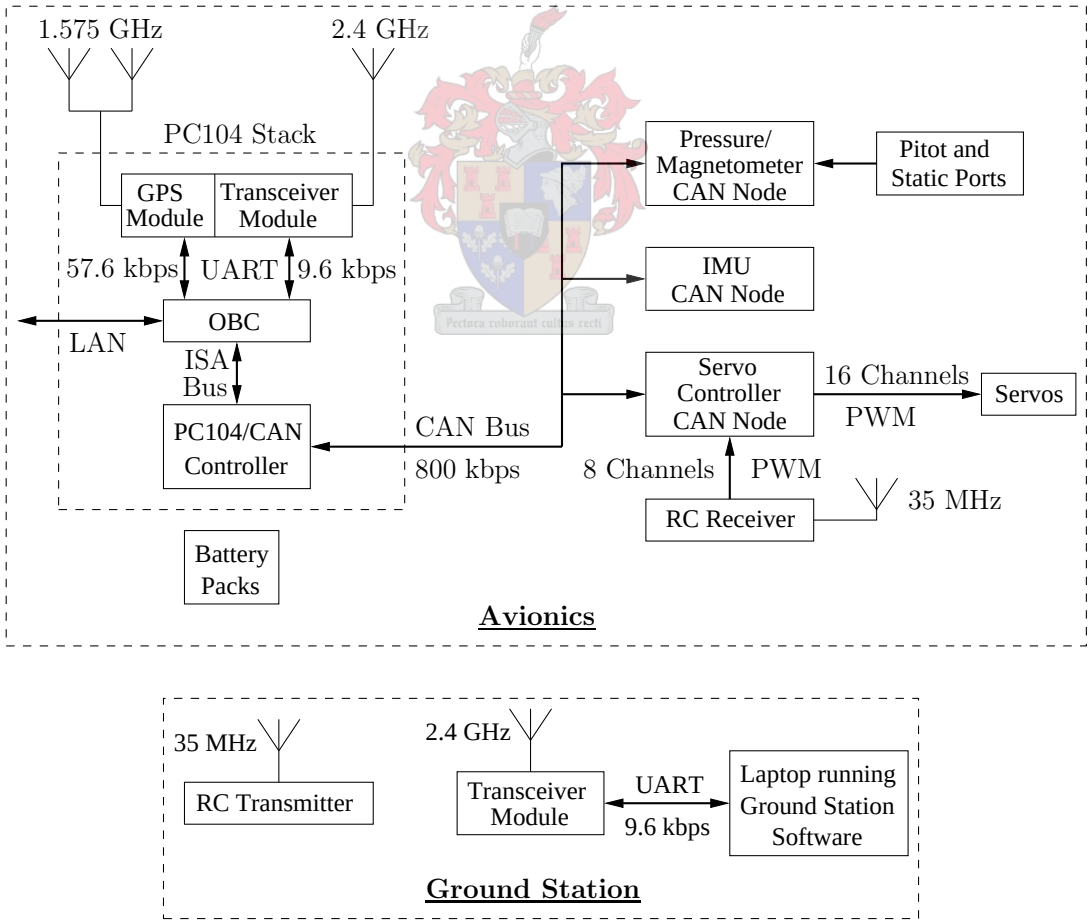


Figure 5.1: Avionics and Ground Station Overview

5.1 Avionics

The avionics used in this project is based on the architecture developed by [38] and [39]. The system was chosen as the platform for implementation since it provides large amounts of procession power, and high sensor and actuator resolution. Another advantage of using this system is its extendibility due to its modular design. This property was exploited to extend the sensing capabilities of the system. The following sections briefly discuss the various components of the avionics and its specific application to this project.

5.1.1 PC104 Stack

A picture of the PC104 stack is shown in Figure 5.2. The stack represents the bulk of the hardware associated with the avionics, and consists of the following components,

- Onboard Computer (OBC)
- PC104/CAN Controller
- GPS and Radio Frequency (RF) link daughter board

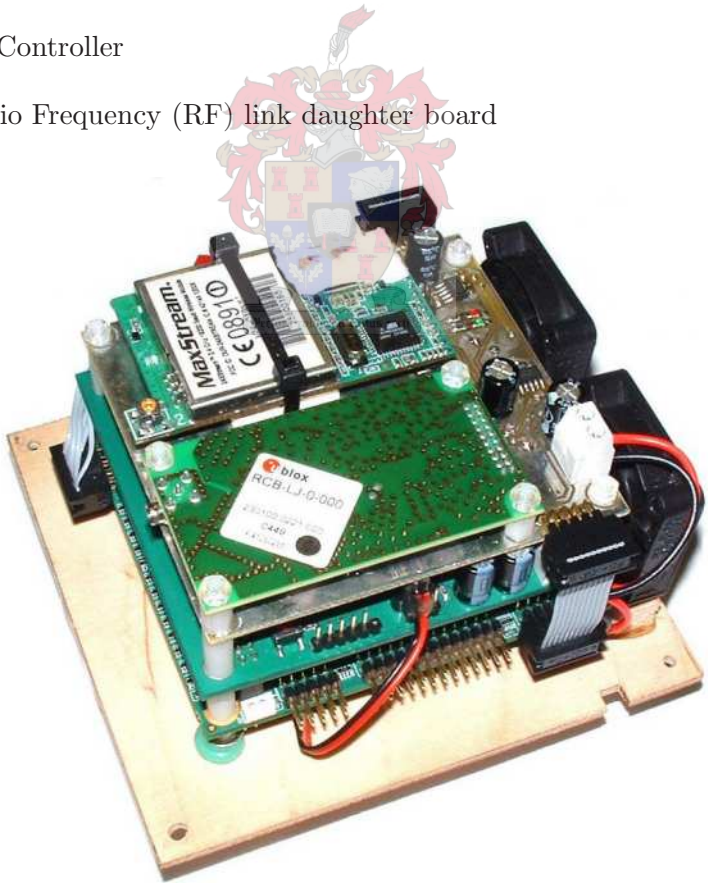


Figure 5.2: Picture of PC104/Stack

At the bottom of the stack sits the OBC, followed by the PC104/CAN controller. The third layer shows the GPS and RF link daughter board, housing the respective modules. The stack

is mounted on soft rubber grommets to protect the electronics from mechanical vibration. The various layers are discussed in more detail in the following sections.

OBC

The OBC is a 300 MHz Intel Pentium III processor, presented in PC104 form factor, and provides all the processing power required by the control system. The major advantage of using the processor is its capability to perform floating point calculations with great efficiency. The OBC runs a reduced Linux kernel, optimised for the hardware in use. All estimator and controller data gets logged on a 32 MB solid state hard disk, and downloaded via a Local Area Network (LAN) connection after an arbitrary number of flight tests.

PC104/CAN Controller

The PC104/CAN controller can be seen as the master of the avionics package since it dictates the timing of the entire system. The board also provides a data path between the OBC and the Controller Area Network (CAN) bus. Consider Figure 5.3 which shows the interaction between the OBC and PC104/CAN controller.

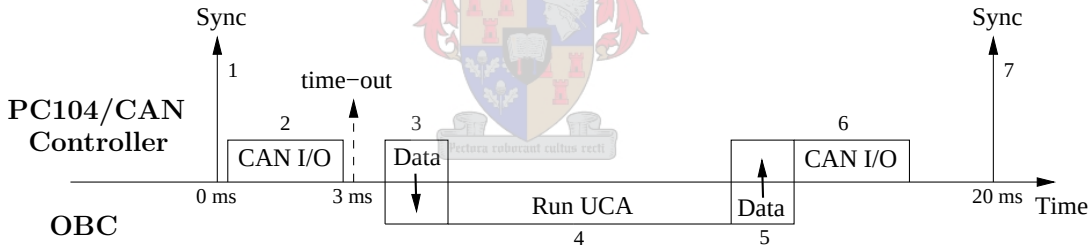


Figure 5.3: OBC and PC104/CAN Controller Timing Scheme [38]

The figure encapsulates the working of the entire avionics package, and is explained by the following seven steps,

1. The PC104/CAN controller starts the process off by issuing a global time synchronisation packet on the CAN bus. In turn, the sensor nodes respond by placing their most recent measurements on the CAN bus. The servo controller updates all servo channels to their most recently received positions, and also places the current safety pilot servo positions on the bus.

2. A predefined amount of time is allocated for the PC104/CAN controller to receive the sensor data and safety pilot servo positions. The data gets packed into an internal buffer until the 3 ms time-out occurs. Any CAN bus activity beyond this point gets discarded.

3. A communication link is established between the PC104/CAN controller and the OBC. The sensor data and safety pilot commands are then transferred to the OBC via the Industry Standard Architecture (ISA) bus.

4. With all the sensor and safety pilot data at hand, the User Control Application (UCA) gets evoked. The process includes the execution of the estimation and control algorithms, which ultimately generates a set of autopilot servo commands. Section 5.2 is dedicated to a more detailed discussion of the UCA.

5. The generated servo commands, along with any additional CAN packets enqueued by the UCA, get transferred to the PC104/CAN controller.

6. The servo commands and CAN packets get placed on the CAN bus, correctly addressed to the respective nodes.

7. The global synchronisation packet is issued once more, causing the entire process to repeat itself.

Inspecting the process closely, the reader will notice that a 20 ms actuator delay gets introduced. However, considering the length of delay and the plant's bandwidth, its effects on the control system's performance are negligible. For an in-depth discussion on the design of the PC104/CAN controller, see [39].

GPS and RF Link Daughter Board

Figure 5.4 shows a close-up picture of the GPS and RF link daughter board. Apart from housing the RF transducer and GPS receiver, the board also provides the required electronic interfaces for the modules to interact directly with the OBC.

The RF module and antenna used are listed below,

- MaxStream XStream 2.4 GHz OEM Transceiver
- Active dipole antenna with 2.1 dB gain

The decision was made to migrate to RF modules from MaxStream due to the unreliability of the AeroComm modules previously used. The modules have excellent track records, and have proven their applicability to UAV projects [36]. Furthermore, the modules are extremely easy to integrate and provide long range communication capabilities. The transceivers provide addressed packets with full duplex communication at 9600 bps, and were found to work excellently for

Figure 5.4: Picture of GPS and RF Link Daughter Board

command and telemetry purposes in this project. The GPS module and antennas used are listed below,

- u-Blox RCB-LJ OEM Receiver
- 2 × Active antennas with 25 dB gain

The GPS module has been used extensively in past UAV projects, and provides a good combination of measurement accuracy and low cost. The plane was equipped with two right-hand circular polarized (RHCP) antennas to try to obtain a GPS fix at all times. This has the advantage that the EKF state propagation periods are kept as short as possible, ensuring more accurate state estimates. Although the process involves placing two active antennas in parallel, the setup proved to work well in that a adequate number of satellites could always be obtained, irrespective of the aircraft's orientation. The module was set up to deliver information about the aircraft's latitude, longitude, altitude and NED velocities at 4 Hz.

5.1.2 IMU CAN Node

The Inertial Measurement Unit (IMU) CAN node consists of two Printed Circuit Boards (PCBs) in a stack configuration. The base board, designed by [38], provides a generic sensor platform. The PCB implements eight, second order analogue Butterworth filters used to suppress high frequency noise. The filtered analogue signals are sampled by 16-bit Analogue to Digital Converters (ADCs) at 1 kHz, and finally digitally low pass filtered down to a bandwidth of 8 Hz. The over sampling approach has the advantage that it averages out any inherent noise in the

ADC process. The 8 Hz cut-off frequency provides high enough bandwidth for the aerobatic manoeuvres attempted in this project. The second PCB, designed by [40], is node specific and contains the following inertial sensors,

- 3 × ADXRS300 Single-Axis Rate Gyroscopes from Analog Devices
- 2 × ADXL210E Dual-Axis Accelerometers from Analog Devices

Figure 5.5 shows a picture of the IMU CAN Node, mounted on rubber grommets to reduce the effects of vibration noise.

Figure 5.5: Picture of IMU CAN Node

Rate Gyroscopes

The rate gyroscopes are capable of measuring angular rates up to ±300°/s. It is a necessity to be able to measure high angular rates since the EKF performance will be greatly effected should sensor saturation occur. The ±300°/s angular rate is mapped over a voltage span of 3 V. Using a 16-bit ADC, angular rates as little as 0.015°/s can be sensed. This is below the sensor's noise figures and subsequently no amplification was performed on the sensor's output. Appendix D contains the signal conditioning circuit used for the rate gyroscopes.

Accelerometers

The accelerometers are capable of measuring accelerations up to ±10 g, mapped over a voltage span of 2 V. Using a 16-bit ADC, accelerations as little as 0.008 m/s^2 can be sensed. This

is again below the sensor's noise figures and led to the decision to neglect amplification of the sensor's output. Appendix D contains the signal conditioning circuit used for the accelerometers.

5.1.3 Magnetometer/Pressure CAN Node

The Magnetometer/Pressure CAN node is the second sensor node on the CAN bus. The node again consists of two stacked PCBs, one of which is generic as discussed in the previous section. The node specific PCB was developed during this project, and contains the following sensors,

- 1 × MPXV5004G Differential Pressure Sensor from Freescale Semiconductor
- 1 × MPXA4115A Absolute Pressure Sensor from Freescale Semiconductor
- 1 × HMC2003 Three-Axis Magnetometer from Honeywell

Although the pressure sensors are not required for feedback purposes in the controller design, the sensors were included for possible use in future projects. Figure 5.6 shows a picture of the Magnetometer/Pressure CAN node, mounted on rubber grommets to reduce the effects of vibration noise.

Figure 5.6: Picture of Magnetometer/Pressure CAN Node

Differential Pressure Sensor

Bernoulli's equation relates airspeed to dynamic pressure and is mathematically expressed as,

$$q = \frac{1}{2}\rho V^2 \tag{5.1}$$

where q reflects differential pressure. A pitot-static tube system, designed in exactly the same manner as described by [35], was used to measure the difference in pressure. The sensor is capable of measuring pressure differences ranging from 0 to 3.92 kPa, mapped over a 4 V voltage span. Using equation (5.1), the measurable airspeed is calculated to be between 0 and 80 m/s at sea level, which provides adequate leeway given the 30 m/s trim condition. In order to evaluate the sensors resolution, equation (5.1) is differentiated with respect to the airspeed, and evaluated at the trim airspeed,

$$\left[\frac{\partial q}{\partial V}\right]_{V_0} = \rho V_0 = 36.75\,\text{Pa/ms}^{-1} \tag{5.2}$$

The airspeed digital resolution is obtained by converting the result to an ADC value as follows,

$$\text{Airspeed Digital Resolution} = \left(\frac{36.75 \times 4 \times 2^{16}}{3.92 \times 10^3 \times 5}\right)^{-1} = 0.002\,\text{ms}^{-1} \tag{5.3}$$

The calculation shows the airspeed can be measured with high enough resolution, and subsequently no additional signal conditioning was performed. Appendix D contains the signal conditioning circuit used for the differential pressure sensor.

Absolute Pressure Sensor

According to [35], in a standard atmosphere, static pressure is related to the altitude above sea level by the following equation,

$$p\,(h) = p_0 e^{-\frac{Mgh}{RT}} \tag{5.4}$$

where,

$$\begin{aligned} p_0 &= 101.325\,\text{kPa} \equiv \text{Average static pressure at sea level} \\ M &= 0.02897\,\text{kg/mole} \equiv \text{Average molecular mass of atmosphere} \\ g &= 9.81\,\text{m/s}^2 \equiv \text{Average gravitational acceleration} \\ R &= 8.31451\,\text{J/K/mole} \equiv \text{Gas law constant} \\ T &= 288\,\text{K} \equiv \text{Average temperature at sea level} \end{aligned}$$

The sensor is capable of measuring absolute pressure ranging from 15 to 115 kPa, mapped over a voltage range from 0.2 to 4.8 V. Feeding the sensor's output directly into a 16-bit ADC results in an altitude accuracy of 0.14 meters at sea level. For extra resolution, the sensor output was amplified to map 70 to 105 kPa over 0 to 5 V, resulting in altitudes being measured between 0 and 3000 meters. The range provides adequate leeway for the aircraft to operate in, and corresponds to a 0.04 meter resolution at sea level. Appendix D contains the signal conditioning circuit used for the absolute pressure sensor.

Magnetometer

The HMC2003 is an anisotropic magneto resistive (AMR) sensor which provides a vector measurement of the earth's magnetic field. The sensor is capable of measuring magnetic fields up to ± 2 Gauss, mapped over a voltage span of 4 V. However, the magnetic field in the vicinity of Stellenbosch has a magnitude in the order of 0.26 Gauss (see Section C.4), and subsequently the sensor's output was magnified by a factor four. This restricts the magnetometer to measure fields with magnitudes up to ± 0.625 Gauss, with 20μ Gauss resolution.

AMR magnetometers make use of a resistive perm alloy strip, polarised in a certain direction, to sense external magnetic fields. When exposed to a strong magnetic field, the polarisation of the strip can be altered, resulting in a change in sensor characteristics. A set/reset circuit is used to apply a strong magnetic field to the sensor, restoring the polarisation of the perm alloy strips. A set/reset cycle also provides the opportunity to calculate any inherent offsets associated with the sensor. This is based on the fact that the sensor's sensitivity can be reversed during the respective cycles, but the associated offsets remain the same. In mathematical terms, the magnetometer output during a reset cycle is given by,

$$\text{Output}_{\text{reset}} = -\text{Sensor sensitivy} \times \text{External field} + \text{Offset} \tag{5.5}$$

And during a set cycle,

$$\text{Output}_{\text{set}} = \text{Sensor sensitivy} \times \text{External field} + \text{Offset} \tag{5.6}$$

By combining equations (5.5) and (5.6), the inherent sensor offsets can be calculated. Hard and soft iron effects are typical disturbances corrupting magnetometer measurements. When it is not possible to mount the sensor in a place where these effects have little or no influence, the disturbances need to be calibrated out as best possible. This project makes use of a simplified relative calibration technique as introduced by [32]. The method relies on the fact that, in a perfect world, all magnetometer measurements will lie on a 3D sphere with radius equalling the magnitude of the earth's magnetic field.

The technique involves the magnetometer being mounted in its final operating environment, with all disturbances present. The aircraft is then rotated randomly through a series of angles, and the sensor outputs recorded. The calibrated measurements are then expressed as a function of the recorded data, as well as gain and offset matrices,

$$\begin{bmatrix} x_{cal} \\ y_{cal} \\ z_{cal} \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{bmatrix} \begin{bmatrix} x_{rec} \\ y_{rec} \\ z_{rec} \end{bmatrix} + \begin{bmatrix} o_1 \\ o_2 \\ o_3 \end{bmatrix} \tag{5.7}$$

where the subscripts *cal* and *rec* indicate calibrated and recorded measurements respectively. Next, the square radius of the calibrated measurements is calculated as follows,

$$r^2 = (g_{11}x_{rec} + g_{12}y_{rec} + g_{13}z_{rec} + o_1)^2 + (g_{21}x_{rec} + g_{22}y_{rec} + g_{23}z_{rec} + o_2)^2 + (g_{31}x_{rec} + g_{32}y_{rec} + g_{33}z_{rec} + o_3)^2 \tag{5.8}$$

A disturbance error is then defined as the difference between the radius calculated in equation (5.8), and the known magnitude of the earth's magnetic field. Finally, using a minimum seeking algorithm, the gain and offset matrices are calculated in such a manner as to minimise the sum of the disturbance errors. Figure 5.7 shows the magnitude of the magnetometer measurements before and after calibration. The dashed lines indicate the magnitude of the known earth magnetic field used during the calibration process.

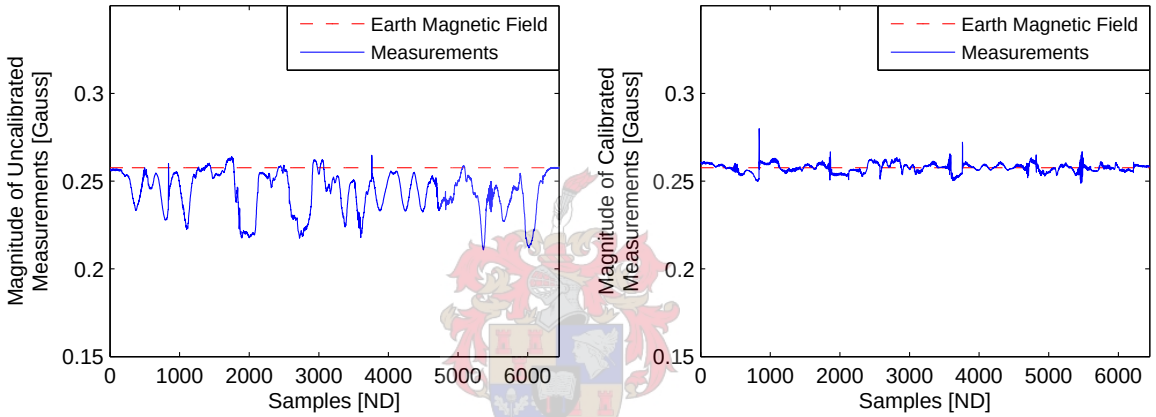


Figure 5.7: Magnetometer Measurements Before and After Calibration

The figures clearly demonstrate the working of the calibration technique. The figure on the left hand side shows the raw magnetometer output without any calibration. The significant fluctuations in the magnitude can be contributed to hard and soft iron disturbances. The figure on the right hand side shows the magnitude of the calibrated measurements. The magnitude closely approximates the constant earth magnetic field for all measurements used during the calibration process.

5.1.4 Servo Controller CAN Node

The hardware design of the servo controller CAN node was done by [39]. The node's main purpose is to allow the autopilot to command the aircraft's actuators, but also provides excellent safety features in the form of smart backup battery logic, provision for CAN bus failure, and smooth transition algorithms. Figure 5.8 shows a picture of the node.

According to Figure 5.1, the servo controller accommodates 8 input channels and provides

Figure 5.8: Picture of Servo Controller CAN Node

outputs to 16 servos. A structure was developed to accommodate the mapping of channels, as well as to simplify the mechanical calibration of the servos. Consider Figure 5.9 which shows the servo controller's interaction with the OBC.

After receiving the time synchronisation packet, the servo controller places its most recently received safety pilot commands on the CAN bus in the form of 8 times values, denoted by $\boldsymbol{x_{T8}}$. Timer values of 2400 and 4800 correspond to pulse widths of 1 and 2 ms respectively. Also, the 16 output servo channels are driven either with timer values received from the OBC, or values calculated using the on-board offset vector and gain matrix. Input channel 8 is dedicated to determine if the autopilot is armed or not, and which timer values are used for the output. When switching between autopilot and safety pilot control, the servo controller implements a 1.28 seconds linear transition. The smooth transition helps protect the aircraft by avoiding sudden changes of the servo positions. Having received the 8 timer values, the OBC converts the values to non-dimensional quantities known as aero units,

$$\boldsymbol{x_{A8}} = \boldsymbol{A_8}\left(\boldsymbol{x_{T8}} - \boldsymbol{c_8}\right) \tag{5.9}$$

where -1 and $+1$ aero units correspond to 1 and 2 ms pulse widths respectively, and $\boldsymbol{A_8}$ and $\boldsymbol{c_8}$ are defined as follows,

$$\boldsymbol{A_8} = \frac{2}{4800 - 2400} \times \text{eye}\,(8,8)$$
$$\boldsymbol{c_8} = 3600 \times \text{ones}\,(8,1)$$

Next, the 8 safety pilot commands are put through a 16×8 mixing matrix to obtain 16 safety pilot commands in aero units,

$$\boldsymbol{x_{A16}} = \boldsymbol{M_{16}}\boldsymbol{x_{A8}} \tag{5.10}$$

The setup of $\boldsymbol{M_{16}}$ is purely theoretical and relates safety pilot static inputs to servo outputs. The safety pilot commands are also put through a user defined transformation to obtain a set

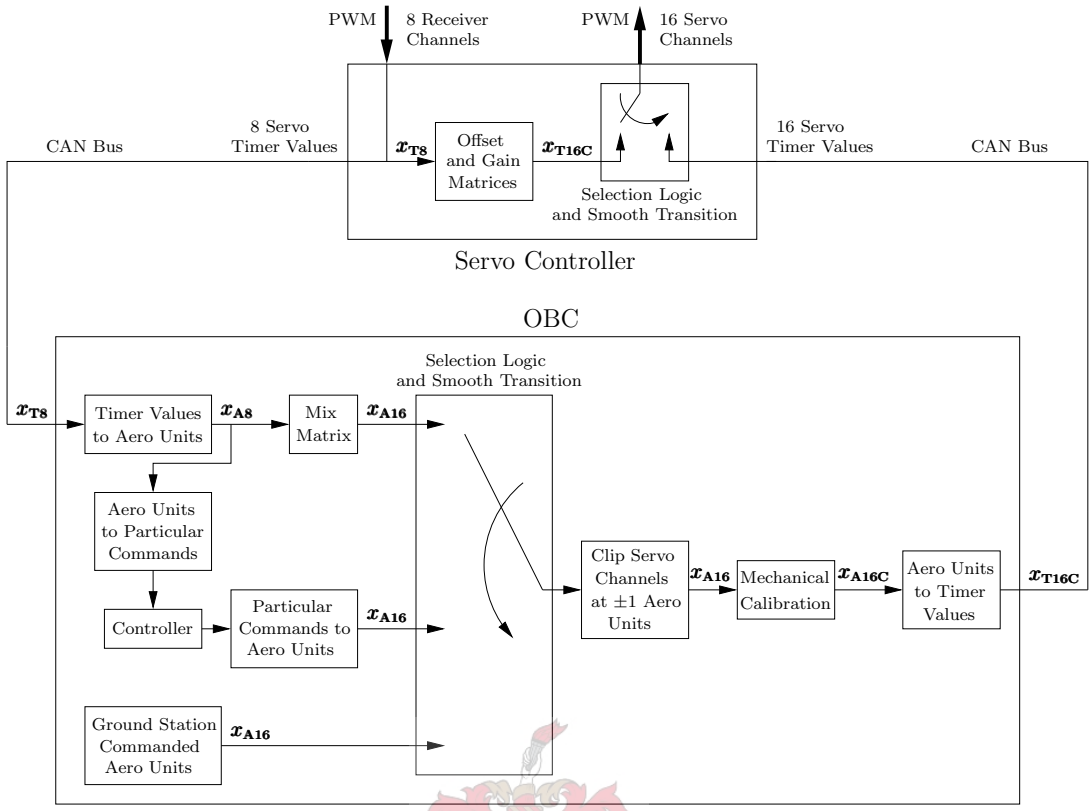


Figure 5.9: Interaction between Servo Controller and OBC

of 8 particular commands. This allows the user to test controllers in stages by letting the safety pilot control certain actuators even when the autopilot is armed. In order to calibrate the servos, the ground station can also be used to command the servos. The ground station commanded aero units enter the process as shown by Figure 5.9.

Next, selection logic is applied to determine which values are to be used to drive the servos. Smooth transition, identical to the implementation on the servo controller, is used to switch between safety pilot and autopilot control, and vice versa. The selected commands are clipped and put through a mechanical offset and gain stage to obtain 16 calibrated aero units,

$$\boldsymbol{x_{A16C}} = \boldsymbol{S_{16}x_{A16}} + \boldsymbol{b_{16}} \tag{5.11}$$

where $\boldsymbol{S_{16}}$ is the mechanical gain matrix and $\boldsymbol{b_{16}}$ is the mechanical offset vector. The matrix and vector entries are determined by using the ground station to command aero units for each of the 16 channels, and then changing the associated offset and gain values until the desired particular command is reflected by the actuator. The process is completed by converting the calibrated aero units back to timer values,

$$\boldsymbol{x_{T16C}} = \boldsymbol{A_{16}x_{A16C}} + \boldsymbol{c_{16}} \tag{5.12}$$

where $\boldsymbol{A_{16}}$ and $\boldsymbol{c_{16}}$ are defined as follows,

$$\boldsymbol{A_{16}} = \frac{4800 - 2400}{2} \times \text{eye}\,(16, 16)$$

$$\boldsymbol{c_{16}} = 3600 \times \text{ones}\,(16, 1)$$

The offset vector and gain matrix residing on the servo controller relate safety pilot timer values to output timer values. Having used the OBC to calibrate the actuators, formulas need to be derived to calculate the offset vector and gain matrix. Combining equations (5.9) to (5.12), and applying some mathematical manipulation, the following result is obtained,

$$\boldsymbol{x_{T16C}} = [\boldsymbol{A_{16}S_{16}M_{16}A_8}]\,\boldsymbol{x_{T8}} + [\boldsymbol{c_{16}} + \boldsymbol{A_{16}b_{16}} - \boldsymbol{A_{16}S_{16}M_{16}A_8c_8}] \tag{5.13}$$

Inspecting the result, it is obvious to see that the terms encapsulated by the first set of brackets provide a formula to calculate the gain matrix, whilst the remaining terms can be used to calculate the offset vector. The technique discussed here greatly simplifies the servo calibration process, and has been implemented with great success in this project.

5.1.5 Power Distribution

The avionics is powered by two independent sources,

- 11.1 V, 2000 mAh Lithium Polymer Battery Pack
- 4.8 V, 1500 mAh Nichol Cadmium Battery Pack

The Lithium Polymer battery powers the PC104 stack and all the CAN bus nodes. When operational, the avionics consumes 20 watts of power on average, effectively limiting the operational time to one hour. However, for safety reasons, a limit of 50 minutes was imposed on the time of operation. Considering that a fuel tank only lasts in the order of 15 minutes, this is perfectly acceptable. The Nichol Cadmium battery is connected to the servo board, providing power to the servos. The battery also serves as a backup power source should the servo controller node get disconnected from the CAN bus. For a more detailed discussion on the avionics' power distribution, see [39].

5.1.6 Cost and Mass Summary

Table 5.1 presents a mass and cost summary of the avionics package. The table shows that the avionics adds 1.35 kg to the aircraft's weight and costs a total of R 15200. The added weight

increases the wing loading from $7\,\mathrm{kg/m^2}$ to $10\,\mathrm{kg/m^2}$, and decreases the static thrust to weight ratio from 1.5 to approximately unity. The decreased thrust to weight ratio might prevent the autopilot from keeping the aircraft's airspeed constant during the initial stages of the loop and Immelmann manoeuvres. Should this be the case, the respective maneouvre trajectories will have to be altered to compensate for the inadequate thrust. Considering the airframe strength of aerobatic aircraft, the increase in wing loading was deemed acceptable.

Table 5.1: Avionics Mass and Cost Summary

Item	**Mass [g]**	**Cost [R]**
PC104 Stack	600	7800
IMU CAN Node	120	1400
Magnetometer/Pressure CAN Node	110	4100
Servo Controller CAN Node	100	500
GPS and RF Antennas	120	580
Battery Packs	300	820
Total	1350	15200

Figure 5.10: Picture of Avionics Mounted inside Fuselage

5.2 User Control Application

The UCA was written in C-code and compiled using a standard Linux platform. The executable is compiled static since the OBC does not support the dynamic linking of libraries. A modular design approach was adopted to make the code adaptable to future projects. Figure 5.11 presents a flow chart of the UCA.

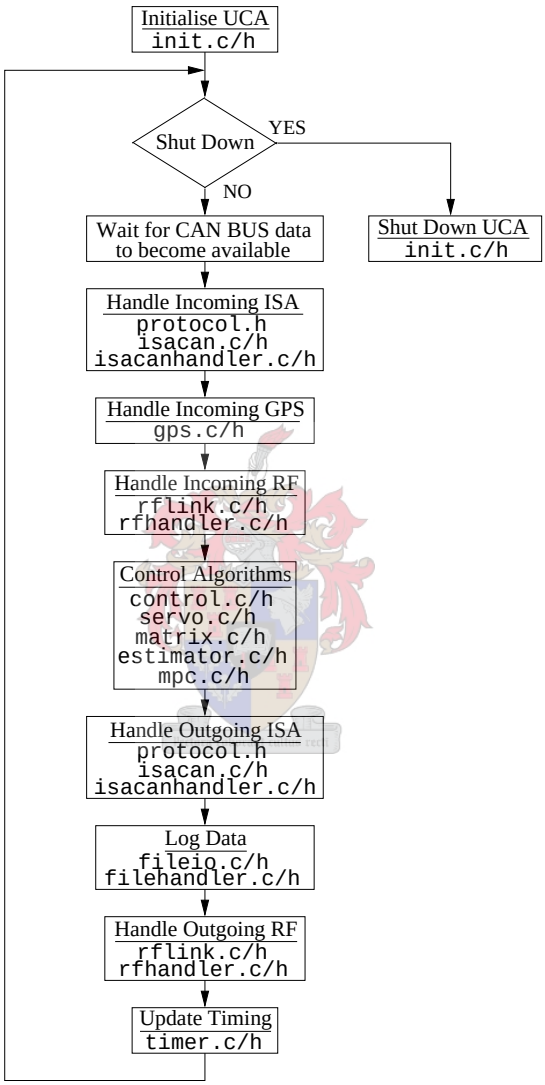
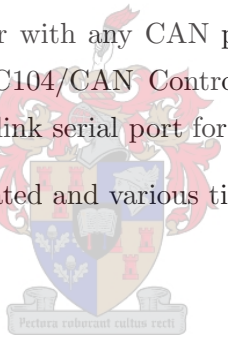


Figure 5.11: UCA Flow Chart

The figure shows the main files responsible for the various stages and in doing so emphasises the source code's modular design. For example, `rflink.c/h` is responsible for implementing basic functionality associated with the RF link. The file handles the low level communication with the serial port and, amongst other things, provides procedures to parse the communication protocol. `rfhandler.c/h` on the other hand is application specific and makes use of the functionality provided by `rflink.c/h`. The same file name methodology was adopted throughout the design.

The UCA flow chart is explained in greater detail by the following 6 steps,

1. The PC104/CAN controller announces the arrival of new CAN bus data. The ADC and timer values get stored on the OBC after which they are put through various offset and gain stages to obtain sensor measurements and safety pilot commands.
2. The GPS serial port buffer is read after which any received data gets parsed. Having successfully parsed a complete string, a flag is set to indicate the availability of GPS updates for the current sample instance.
3. The RF link serial port buffer is read, and the received data parsed. The messages then get interpreted, and the autopilot is updated as required.
4. Having arrived at this point, all sensor measurements and safety pilot commands are available in user friendly units. If enabled, the EKF and MPC algorithms are executed. Selection logic, as explained by Figure 5.9, is then applied to determine which commands are to be used to drive the servos.
5. The servo timer values, together with any CAN packets enqueued during the preceding steps, are transferred to the PC104/CAN Controller. Telemetry packets are also constructed and written to the RF link serial port for transmission to the ground station.
6. Finally, the OBC's clock is updated and various timing calculations are performed.



5.3 Ground Station

The ground station consists of a laptop running the ground station software, an RF transducer module providing a link to the aircraft, and an RC transmitter. The ground station software allows the user to manage the autopilot, and evaluate the estimator and controller's performance when the aircraft is in flight. The program also provides valuable information concerning the avionics' status. The RC transmitter is used by the safety pilot to control the aircraft whenever the autopilot is not armed. One of the transmitter's channels, in conjunction with a switch controller via the ground station software, is used to turn the autopilot on and off. This allows the safety pilot to disarm the autopilot at any given point, regaining complete control over the aircraft. The next section discusses the ground station software in greater detail.

5.3.1 Ground Station Software

In order to ensure source code consistency, the ground station was developed using Borland C++ builder. The software provides a Graphical User Interface (GUI) to the avionics, allowing the

autopilot to be managed from within a user friendly environment. To keep the OBC and ground station synchronised, an aircraft data file gets read in by both entities upon start-up. From then on, the ground station is able to upload new data and commands which are acknowledged by the avionics once received. The following sections briefly discuss the various interface pages of the ground station.

Ground Station Page

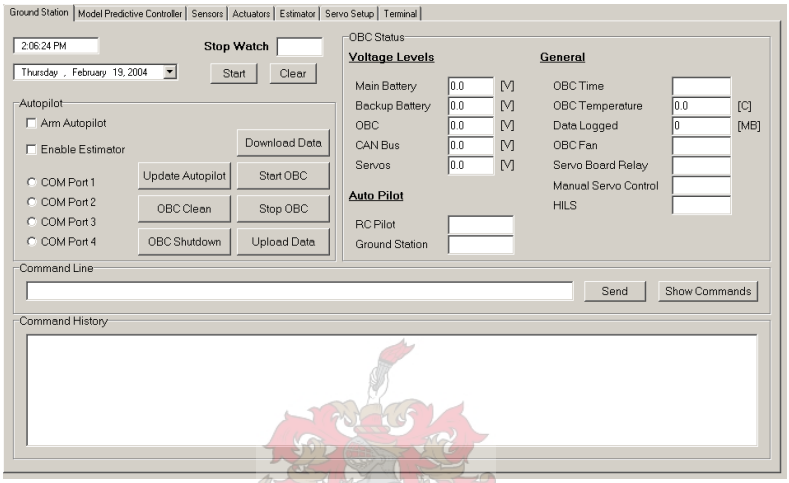


Figure 5.12: Screenshot of Ground Station Page

The ground station page is shown by Figure 5.12. The *Autopilot* group box allows the estimator and autopilot to be armed, and also provides basic functionality to control the OBC. Starting and stopping the UCA, and uploading logged data via the LAN connection are some of the more important features. The *Command Line* allows predefined commands to be transmitted to the autopilot, while the *Command History* lists the various commands issued by the ground station together with the avionics' response to each command. Any errors that occur are also displayed here. The *OBC Status* group box displays critical system information like voltage levels, whether the autopilot is armed by the safety pilot or ground station, and the OBC temperature. A stop watch is provided to time flights, ensuring the aircraft does not run out of fuel.

Sensors Page

The sensors page is shown by Figure 5.13. The *Inertial Measurement Unit* group box displays the rate gyroscope and accelerometer measurements. The buttons allow the IMU to be self tested and the rate gyroscopes to be zeroed before a flight. The bias values used for zeroing are calculated by averaging the sensor outputs for 5 seconds. The *Magnetometer* group box displays the magnetometer measurements, whilst the button can be used to initialise a reset/set cycle.

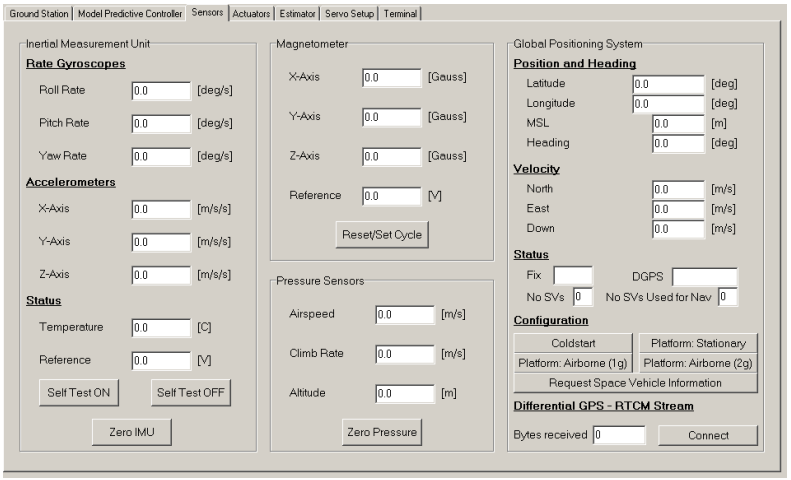


Figure 5.13: Screenshot of Sensors Page

Although not used in this project, the *Pressure Sensors* group box was created and displays the measured airspeed, climb rate and altitude. To compensate for drift, the altitude needs to be zeroed before the start of a flight. The *Global Positioning System* group box displays position and velocity measurements, the fix indicator and the number of satellites used. The various buttons allow the GPS to be configured and monitored in flight. Having clicked the *Request Space Vehicle Information* button, information regarding the current satellites used in the navigation solution is displayed in the *Command History* text box in the main window. Lastly, the page provides the functionality to upload GPS pseudo range corrections. The corrections are streamed over the internet and received by the ground station. The messages are then packaged, transmitted via the RF link and injected into the GPS module causing it to perform like a differential receiver.

Estimator Page

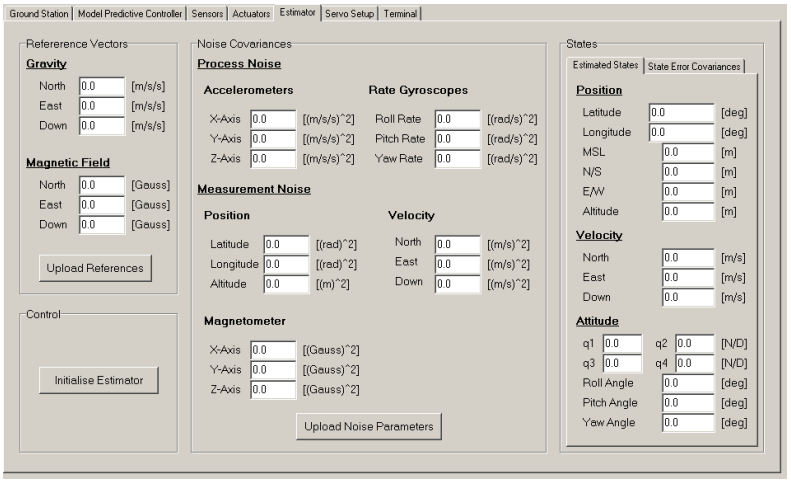


Figure 5.14: Screenshot of Estimator Page

The estimator page is shown by Figure 5.14. The *Reference Vectors* group box allows the gravity and magnetic field reference vectors to be updated, whilst the *Control* group box is used to initialise the EKF using the algorithms discussed in Section 3.4. If required, the process and measurement noise covariance matrices can be altered using the *Noise Covariance* group box. The estimated position, velocity and attitude states are shown in the *Estimated States* tab, while the error covariances of the estimated states can be inspected using the *State Error Covariances* tab.

Model Predictive Controller Page

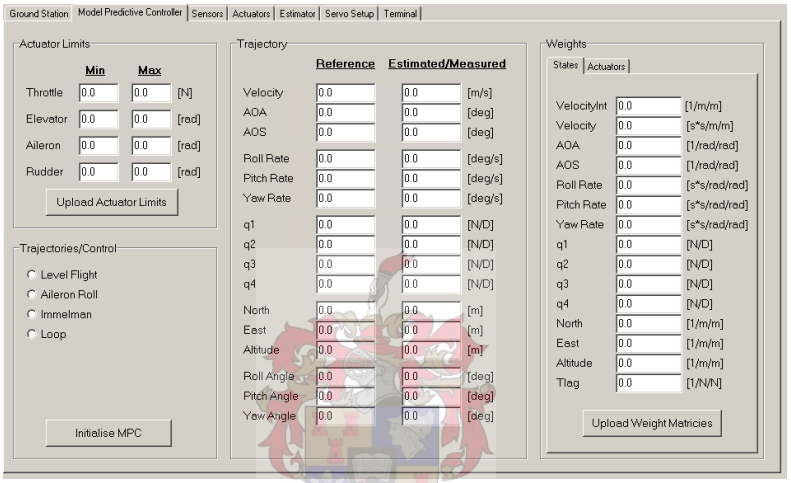


Figure 5.15: Screenshot of Model Predictive Controller Page

The model predictive controller page is shown by Figure 5.15. The *Trajectories/Control* group box is used to select the manoeuvre to be performed, while the *Actuator Limits* group box allows the respective actuator limits to be set up. In order to evaluate the controller's performance, the *Trajectory* group box can be consulted. The nominal/reference state values are shown on the left hand side, whilst the right hand side values reflect the estimated/measured states. Lastly, the page allows the user to change the state and actuator perturbation weights by means of the various tabs in the *Weights* group box.

Actuators, Servo Setup and Terminal Pages

The actuators page allows the user to inspect the safety pilot commands, and also shows the actuator commands transmitted to the servo controller in aero and particular units. Although not used in this project, the page also allows the autopilot to save the safety pilot trim condition while in flight. These values can then be used to serve as bias points if required by the controller.

The servo setup page provides a means for calibrating the servos by allowing each of the 16

servos to be commanded via the ground station. The page allows the user to update the mixing matrix, the mechanical gain matrix, and the mechanical offset vector. Finally, the terminal page can be used to monitor the OBC as it boots up and shuts down. The page can also be used to verify the correct execution of the various OBC tasks as commanded by the ground station page.

5.4 Summary

This chapter presented the avionics and ground station used to implement the estimator and control algorithms of Chapters 3 and 4. Figure 5.16 shows a picture of the complete system. Before any flight tests are attempted, the correct implementation of the various algorithms need to verified. The development of a tool to perform such a task is the topic of the following chapter.

Figure 5.16: Picture of Complete Aircraft and Ground Station

Chapter 6

Hardware in the Loop Simulator

This chapter introduces the development of a Hardware in the Loop Simulator (HILS). Firstly, an overview of the system is provided followed by a discussion of the simulation environment developed during this project. The hardware associated with the system and its interaction with the simulation environment are then presented. Finally, the autopilot's implementation is verified using the developed system.

6.1 HILS Overview

Hardware in the loop simulation provides the functionality to verify the autopilot's implementation before risking the aircraft in a flight test. The process involves replacing parts of a pure simulation environment by hardware components, and in doing so dramatically reduces the chances of failure during a practical flight test. Consider Figure 6.1 which shows a functional block diagram of the HIL system.

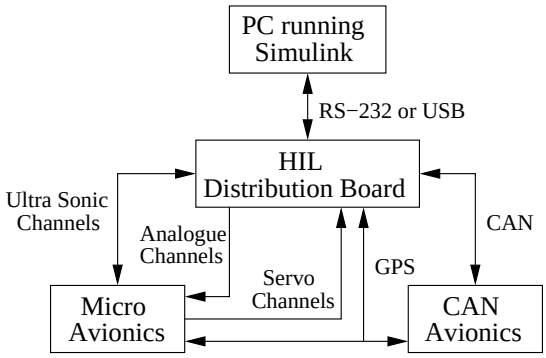


Figure 6.1: Functional Block Diagram of HIL System

The two avionics packages developed by the UAV research group are shown in the bottom half of the figure. The platform discussed in Chapter 5 will be referred to as *CAN avionics* throughout the chapter. The *Micro avionics* package is the system prototyped by [35], and refined by [41]. Next, the figure shows the HIL distribution board which provides electronic interfaces to the two avionics packages. When connected to the CAN avionics package, the distribution board simply acts as another node on the CAN bus. The node receives any actuator commands destined for the servo controller, and provides the functionality to place HIL sensor data on the bus. The OBC then uses this sensor data to override the actual measurements received from the IMU and Pressure/Magnetometer nodes. When connected to the Micro avionics package, the board captures the servo pulse width commands and provides sensor data in the form of 16 analogue channels. The avionics selects between the actual and HIL sensor data by means of a multiplexing scheme. The board is also capable of simulating measurements for two independent ultra sonic sensors and provides GPS updates to the avionics packages.

The received/captured actuator commands are transmitted to a PC via an RS-232 or USB link, and then used to propagate the vehicle dynamics forward in time. The PC closes the loop by returning sensor data to the respective avionics packages by means of the distribution board. The following sections discuss the developed simulation environment, the design of the distribution board, and its interaction with the simulation environment.

6.2 Simulation Environment

To help achieve success during practical flight tests, the entire autopilot needs to be thoroughly simulated. A good simulation environment is paramount to the successful implementation and evaluation of any control system. [35] developed two non-linear simulation environments, each with their own benefits. In this project, the functionality of the two simulators was combined.

A Simulink block diagram simulator was built based on the aircraft model discussed in Chapter 2. The block diagram structure is advantageous since it provides visual insight into how the autopilot and aircraft model interact with each other. A disadvantage of the block diagram simulator developed by [35] was that it was unable to display the aircraft visually. This problem was overcome by the addition of an OpenGL graphics display engine developed by [43]. The graphics engine greatly improves the functionality of the simulator in that it allows the autopilot to be visually scrutinised.

Consider Figure 6.2 which shows the highest level of the block diagram simulator. The figure shows the four main blocks which make up the simulator. The *Aircraft Model* block outputs various information related to the aircraft's current state in the form of a *Data Vector*. The

Data Vector enters the *Autopilot Model* block which in turn generates the *Control Vector* being fed back to the *Aircraft Model*.

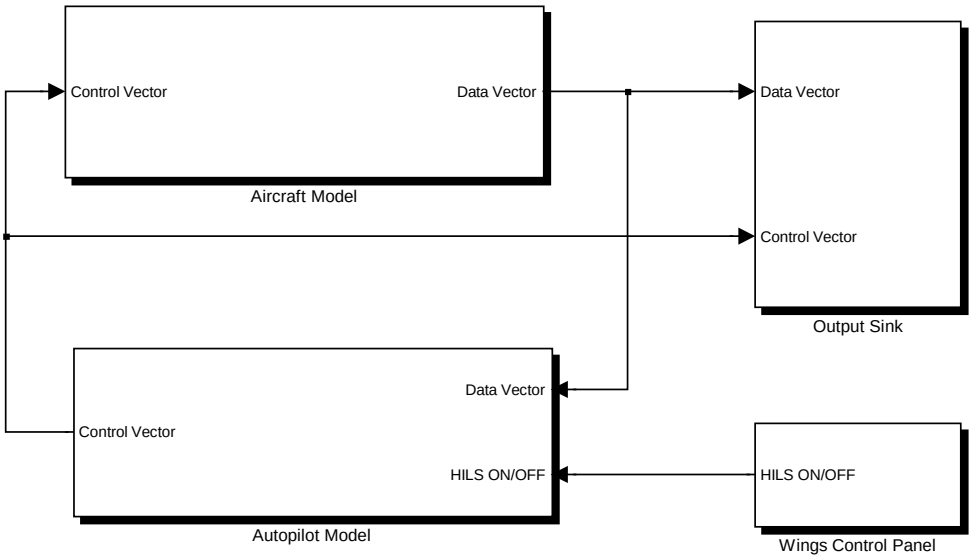


Figure 6.2: Highest Level of Block Diagram Simulator

The *Output Sink* block is responsible for generating different simulator outputs. Its first task is the logging of the data and control vectors to a data file. This functionality is beneficial when wanting to plot certain data sets after a simulation has been completed. Secondly, the block handles the output to the OpenGL graphics engine. A User Datagram Protocol (UDP) connection is used to transmit the aircraft's position and attitude states to the graphics engine. The engine then interprets this information and displays the aircraft in a 3D environment. To further extend the functionality, the engine is capable of displaying key aircraft states as graphs or text boxes. Various objects can also be added into the 3D environment which then serve as references or waypoints which in turn assist in evaluating the autopilot's performance. Figure 6.3 shows a screenshot of the graphics engine with the aircraft in flight. Some objects were added to display an arbitrary trajectory. For an in depth discussion of the graphic engine, see [43].

The *Wings Control Panel* block is responsible for controlling the simulation environment. This block sets up controller reference values, as well as selection logic to decide which control loops are armed. Figure 6.4 shows an expansion of the *Aircraft Model* block. The block diagram shows a good overview of the aircraft model developed in Chapter 2.

The inputs to the diagram are the unpacked *Control Vector*, whilst the outputs are used to make up the *Data Vector*. At the heart of the diagram is the *6-DoF (Quaternion)* block. The *Gravity Model*, *Engine Model* and *Aerodynamic Model* blocks are responsible for generating the respective forces and moments acting on the airframe. The *Runway Model* block, developed by

Figure 6.3: Screenshot of OpenGL Graphics Engine

[41], contributes no forces and moments when the aircraft is in flight.

Figure 6.5 shows an expansion of the *Autopilot Model* block. The *Sensor Model* and *Servo Model* blocks contain dynamic models of the sensors and actuators respectively. The *Sensor Model* block performs two tasks. Firstly, it is responsible for converting *Data Vector* information to sensor values. And secondly, the block generates sensor noise using predetermined noise models. Sitting between the input and output blocks are the autopilot control algorithms. The figure lastly shows a first glimpse of the HILS interface. The interface is discussed in greater detail in Section 6.4.

Quantisation Effects

The effects of quantisation should be taken into account when the control algorithms are implemented on digital processors only capable of fixed point representations. The avionics platform used in this project is suited to perform floating point calculations efficiently. Also, the analogue to digital conversion processes, and servo output routines, provide very high resolution. These arguments provided the basis to neglect the effects of quantisation in the simulation environment.

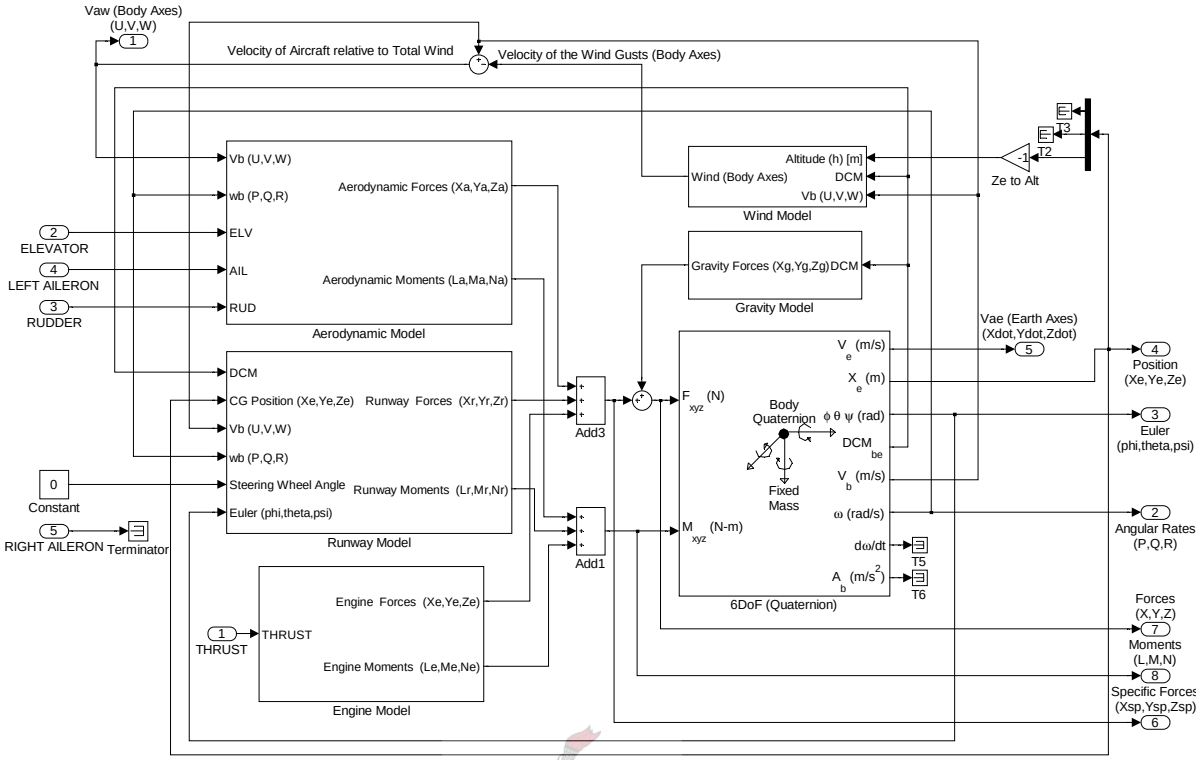


Figure 6.4: Aircraft Model Block of Block Diagram Simulator

Controller Code Conversion

The *Autopilot Model* block in Figure 6.5 contains the EKF and MPC algorithms. Both these algorithms were implemented in C-code in the form of Simulink S-functions. Although the initial stages of implementation are relatively slow with this method, the advantages are overwhelming. By implementing the control law in C-code, invaluable insight is obtained into the working behind the controller. Also, if the code environment is set up correctly, the controller implementation on the avionics is reduced to simply modifying certain variable names. This approach was adopted with great success in this project and is highly recommended when dealing with complex

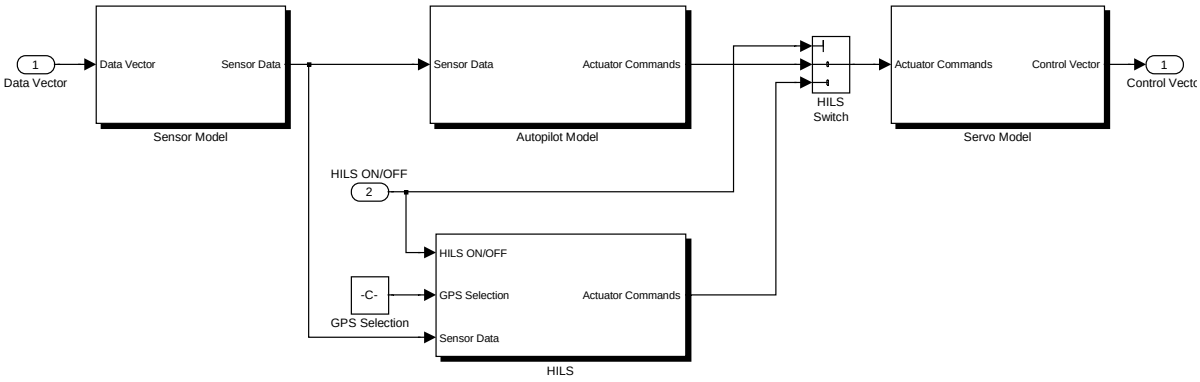


Figure 6.5: Autopilot Model Block of Block Diagram Simulator

algorithms like the model predictive controller. Lastly, the EKF and MPC algorithms require substantial amounts of matrix algebra. A C-code library was developed, capable of performing the basic matrix algebra required by the two algorithms. The most complex routine is a matrix inversion algorithm. The routine firstly performs a lower-upper decomposition on the matrix under study, which is followed by an exploitation of a property of lower-upper decomposition to obtain the inverse. The algorithm was obtained from [20].

Wind Model

The effects of wind gusts can easily be analysed using the block diagram simulator. The gusts enter the system as shown by Figure 6.4. The *Wind Model* block generates the gusts by passing a white noise signal through a noise shaping filter. The noise power and filter characteristics were manipulated to produce wind gusts with magnitudes of 1 m/s, and lasting in the order of 2 seconds. These figures were assumed to represent a real life scenario. For more detail on the wind model, see [41].

6.3 HIL Distribution Board

The HIL distribution board provides the required electronic interfaces between Simulink and the two avionics packages. A block diagram overview of the board is shown in Figure 6.6

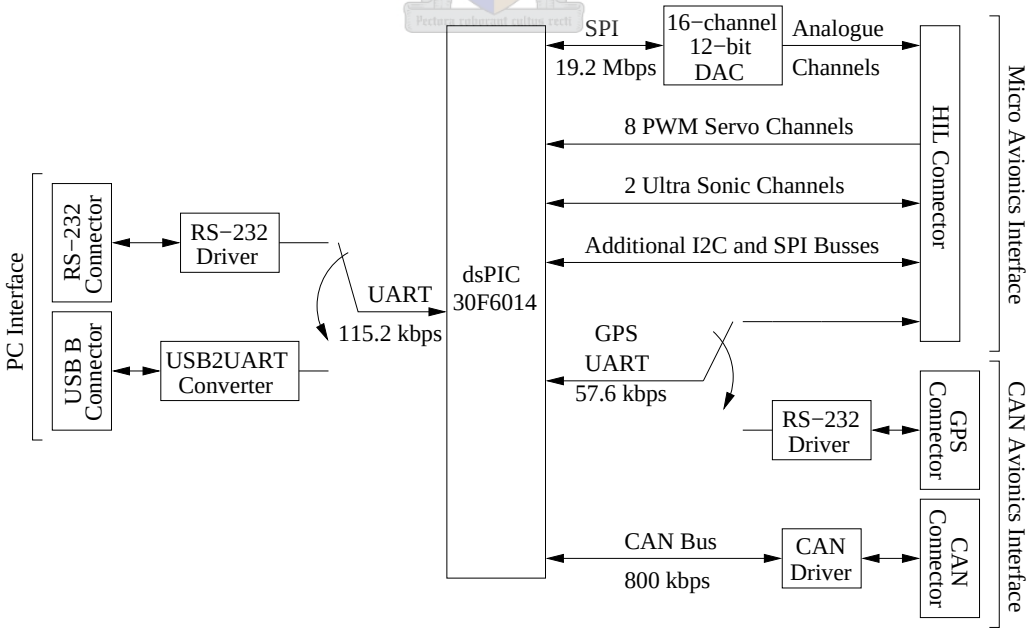


Figure 6.6: Block Diagram of HIL Distribution Board

At the heart of this board are,

- 1 × dsPIC30F6014 Microcontroller from Microchip
- 1 × AD5391 16 Channel, 12-bit DAC from Analog Devices
- 1 × FT232BM USB to UART Converter from FTDI

The microcontroller was chosen because it supports all the communication interfaces and processing requirements of the board. The elegant manner in which the processor deals with interrupts is also advantageous due to the many hardware peripherals used in the design. The Digital to Analogue Converter (DAC) was chosen because of the fast output settling times, simultaneous updating of all the channels, and its very fast serial peripheral interface (SPI). The 12-bit resolution is mapped over a 5 V span, providing the signal accuracy required by the Micro avionics package. The current hardware design also caters for a 14-bit version of the DAC (the AD5390) should higher accuracy be required. The USB (Universal Serial Bus) to UART converter was chosen because of its ease of implementation and relatively low cost.

A UART is included to provide the avionics packages with GPS measurement at 57.6 kbps. The 8 servo channels are timed in independently making use of the microcontroller's input capture modules. Hardware peripherals are also used for the ultra sonic channels' operation which provides for very accurate measurements. A jumper is used to select between the RS-232 and USB interfaces to the PC. The additional I2C and SPI buses were routed out for potential extension of the board. Section D.3 provides detailed descriptions of the PC, Micro avionics and CAN avionics interfaces. A picture of the top and bottom layers of the distribution board is shown in Figure 6.7.

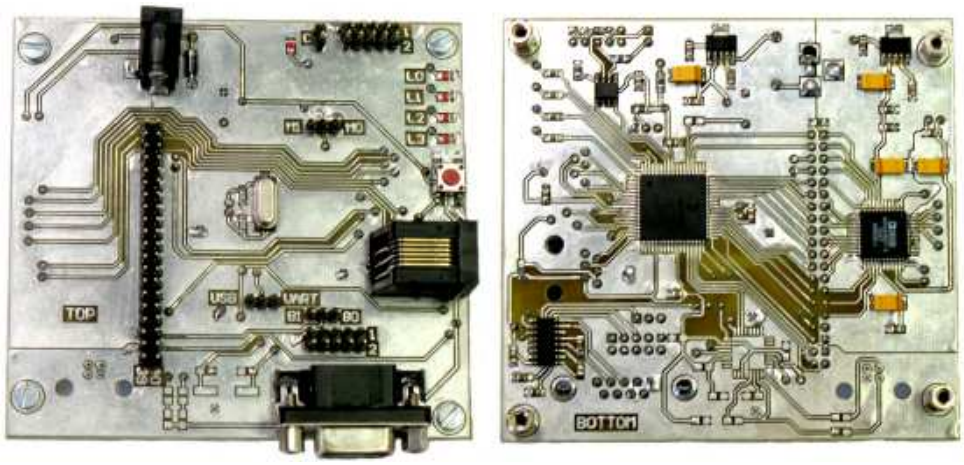


Figure 6.7: Picture of Top and Bottom Layers of HIL Distribution Board

The board's operation is best explained by presenting timing diagrams for when the system

is connected to the various avionics packages. Figure 6.8 shows a timing diagram of the HIL system when connected to the CAN avionics package. The figure shows the interaction between the CAN avionics, the HIL distribution board and Simulink.

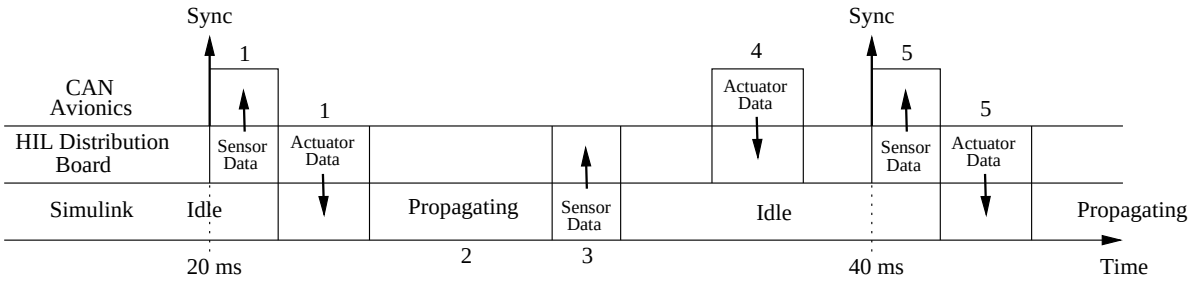


Figure 6.8: HIL Timing Diagram when Connected to CAN Avionics

The explanation to follow assumes that the reader has a thorough understanding of how the CAN avionics operates. The system's operation, when connected to the CAN avionics, is explained by the following 5 steps,

1. The PC104/CAN controller issues the global time synchronisation packet on the CAN bus. After receiving the packet, the distribution board transmits its most recent sensor data to the avionics package. The board also transmits the last received actuator data to Simulink via the RS-232/USB link. The origin of the sensor and actuator data will be explained by steps 3 and 4 respectively.

2. Simulink is allowed to propagate the vehicle dynamics forward in time by 20 ms using the received actuator data.

3. The propagated vehicle state vector is manipulated to obtain measurements for the various sensors. The measurements are then transmitted to the distribution board via the RS-232/USB link.

4. The avionics package transmits its actuator commands via the CAN bus and is received by the distribution board.

5. The time synchronisation packet is issued once again and the entire process is repeated.

Inspecting the time line closely, it is noticed that a 20 ms sensor delay gets introduced by the process. However, considering the length of delay and typical vehicle bandwidths, the delay's effects are negligible. Although not shown in the figure, Simulink generates new GPS measurements every 250 ms. Since the RS-232/USB link has limited bandwidth, the GPS messages are split into smaller packets and transmitted to the distribution board over a 250 ms period. Having received a complete GPS update, the distribution board transmits the string to the

avionics package. This approach introduces a 250 ms delay on all GPS measurements. However, as previously mentioned, the uBlox receivers currently in use display a measurement delay in the order of 310 ms. Thus the 250 ms delay introduced by the HIL system perfectly mimics the actual receivers. Figure 6.9 shows a timing diagram of the HIL system when connected to the Micro avionics package. The figure shows the interaction between the Micro avionics, the HIL distribution board and Simulink.

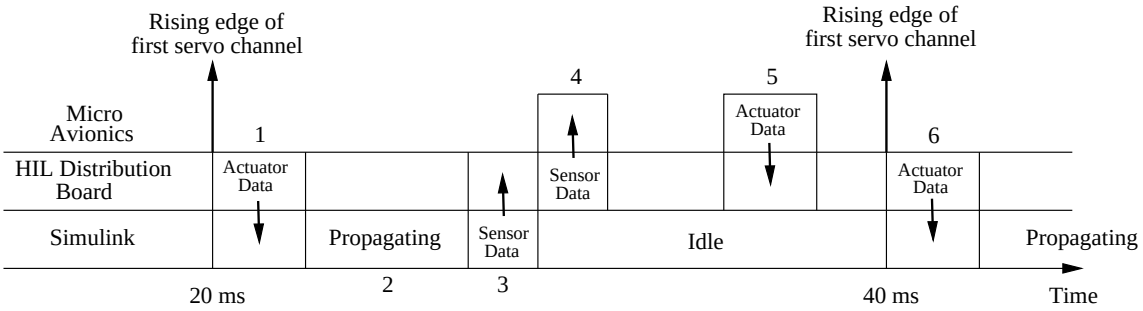


Figure 6.9: HIL Timing Diagram when Connected to Micro Avionics

The time line is closely related to the previous discussion but introduces subtle differences that need to be mentioned. The explanation to follow assumes the reader has a thorough understanding of how the Micro avionics operates. For a detailed discussion of the system, see [35] and [41]. The system's operation, when connected to the Micro avionics, is explained by the following 6 steps,

1. The distribution board gets triggered by the rising edge of the first servo channel, SERVO0 (see Figure D.6). The board starts transmitting the last received actuator data to Simulink via the RS-232/USB link. The origin of the actuator data will be explained in step 5.

2. Simulink is allowed to propagate the vehicle dynamics forward in time by 20 ms using the received actuator data.

3. The propagated vehicle state vector is manipulated to obtain measurements for the various sensors. The measurements are then transmitted to the distribution board via the RS-232/USB link.

4. The distribution board updates the DAC channels, driving the latest sensor data directly into the avionics where the analogue signals then get digitised.

5. Since the rising edge of the first servo channel was received, the avionics has been driving pulse widths for all 8 servo channels directly into the distribution board. Step 5 earmarks the last channel being captured and all pulse widths, in the form of timer values, are now ready to be transmitted to Simulink.

6. The distribution board once again gets triggered by the rising edge of the first servo channel and the entire process is repeated.

The process once again introduces a 20 ms sensor delay and GPS updates are handled in the same manner as previously discussed. The Micro avionics can ping the distribution board at any time for ultra sonic measurements. After receiving a request for either of the channels, the board uses the last received ultra sonic measurement to generate an echo reflecting the respective distance measurement.

6.4 Simulink Interface

The Simulink interface provides a gateway to the HIL distribution board. Figure 6.5 already showed where the interface fits into the block diagram simulator. A more detailed description of the block is shown in Figure 6.10.

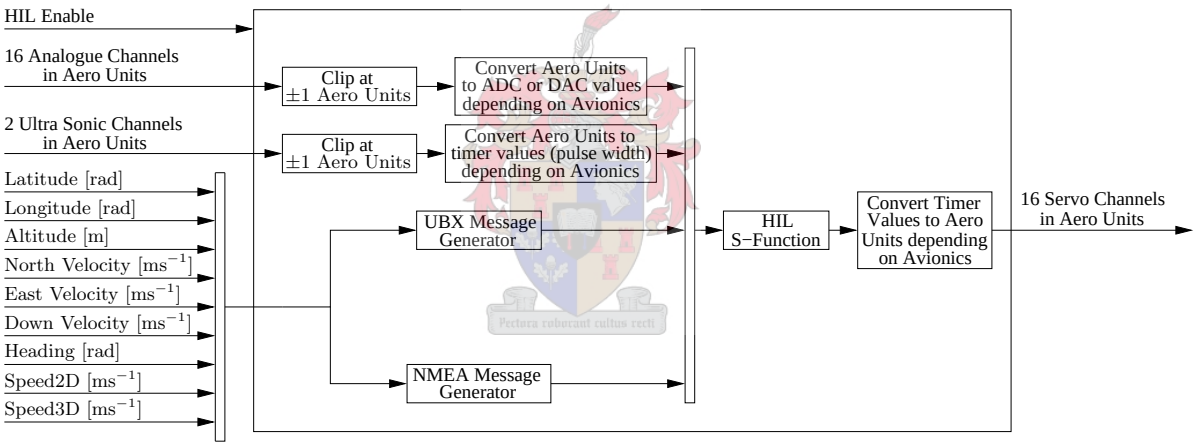


Figure 6.10: HIL Simulink Interface

At the heart of the block is the HIL S-function which performs the low level communication with the serial port and ensures Simulink's state propagation is stopped and started as shown by Figures 6.8 and 6.9. Entering the block are analogue, ultra sonic and GPS sensor data to be transferred to the avionics packages, and leaving are the received actuator commands. The block inputs, together with their respective units, are defined as follows,

- HIL Enable - Control line to enable/disable the block. When set to 1, the system is enabled and Simulink will run in real time for as long as the connected avionics package provides synchronisation pulses to the HIL distribution board. When set to 0, the system is disabled and Simulink will run normally.

- Analogue Channels - 16×1 vector containing information for 16 analogue channels. The values are non-dimensional and should be scaled between -1 and $+1$. When the CAN avionics is in use, -1 and $+1$ correspond to ADC values of 0 and 65535 respectively. The values are transferred from the distribution board to the avionics via the CAN bus. When the Micro avionics is in use, the values correspond to DAC values of 0 (0 V) and 4095 (5 V) respectively. The analogue voltage levels are driven out on the header on the distribution board.

- Ultra Sonic Channels - 2×1 vector containing information for 2 ultra sonic channels. The values are non-dimensional and should be scaled between -1 and $+1$, corresponding to distance measurements of 0 and 9 meters respectively. Timer values representative of the distance measurements are transferred via the CAN bus when the CAN avionics is used. As already mentioned, the Micro avionics pulses the distribution board after which an echo is returned reflecting the correct distance measurement. The speed of sound was set at 338.6 m/s for the calculation of the timer values. For the CAN avionics, the timer values are in accordance with a timer running at 1.2 MHz. The distribution board makes use of a timer running at 300 kHz to generate the respective echoes for the Micro avionics.

- GPS Vector - 9×1 vector containing GPS related fields in the following order,
  - Latitude measurement in radians, scaled between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$.
  - Longitude measurement in radians, scaled between $-\pi$ and π.
  - Altitude above mean sea level in meters.
  - NED velocity components in meters per second.
  - True heading measurement in radians, scaled between 0 and 2π.
  - 2D speed in meters per second.
  - 3D speed in meters per second.

The block outputs, together with their respective units, are defined as follows,

- Servo Channels - 16×1 vector containing information for 16 servo channels. The values are non-dimensional and reflect servo pulse widths. Values of -1 and +1 correspond to pulse widths of 1 and 2 ms respectively. When the CAN avionics is in use, the values are received directly via the CAN bus, whereas the micro avionics drives the pulse width signals directly into the distribution board.

The GPS heading, ground speed and 3D speed measurements can all be expressed in terms of the NED velocity components. However, the fields enter the block as separate entities to allow

the user to explicitly specify noise properties for these fields. The HIL system is capable of providing GPS measurements using two different protocols. When the UBX propriety protocol is selected, the following messages are generated by the *UBX Message Generator* block,

- UBX NAV-POSLLH: Geodetic Position Solution. This message provides position information in terms of latitude, longitude and altitude above mean sea-level.
- UBX NAV-STATUS: Receiver Navigation Status. This message provides important status information like the GPS fix indicator.
- UBX NAV-VELNED: Velocity Solution in NED. This message provides velocity information in terms of NED velocity components, ground and 3D speed, and heading.

When the National Marine Electronics Association (NMEA) protocol is selected, the following messages are generated by the *NMEA Message Generator* block,

- NMEA GGA: GPS Fix Data. This message provides position information in terms of latitude, longitude and altitude, as well as GPS fix information.
- NMEA VTG: Course over Ground and Ground Speed. This message provides ground speed and heading information.

In order to speed up the state propagation process in Simulink, it is highly recommended to run the simulation in Accelerator mode. Accelerator mode converts the entire Simulink model to C-code and then compiles the source code to a dynamically linked library. The model executes significantly faster when in binary form and helps ensure real time operation of the system. The HIL distribution board is responsible for monitoring the real time operation of the system. A light emitting diode, denoted L1 on the distribution (see Figure 6.7), is flashed for 100 ms to indicate a timing constraint has occurred. A timing constraint occurs when the distribution board is triggered twice in succession without receiving sensor updates from Simulink.

6.5 Autopilot Verification

Figure 6.11 shows a typical HIL setup used to verify the correct implementation of the autopilot algorithms.

The figure shows the PC104 stack connected to the servo controller and the HIL distribution board via the CAN bus. The RC transmitter and receiver are used to control the aircraft whenever the controller is not engaged, and also to arm the autopilot in conjunction with the

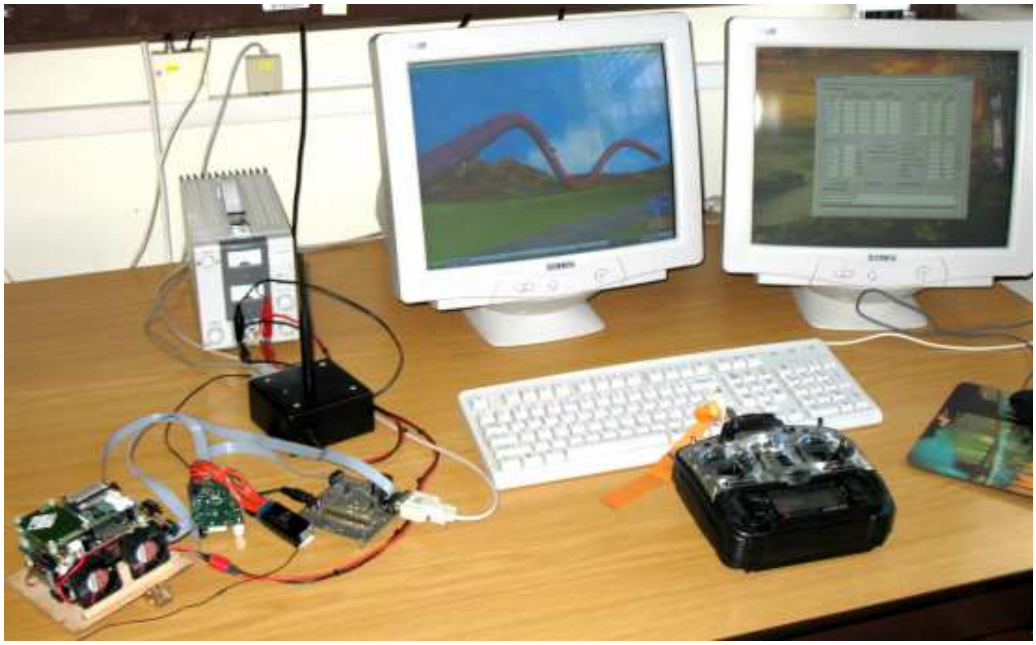


Figure 6.11: Picture of CAN Avionics Connected to the HIL System

ground station as previously discussed. The screen on the left hand side is used to display the aircraft's motion by means of the OpenGL graphics engine. The right hand side screen shows the ground station software used to upload commands to the autopilot via the RF link. Having started the simulation, the aircraft is on the runway after which the state estimator can be enabled. Using the RC console, a take-off is performed and the aircraft can be flown around in the 3D environment provided by the graphics engine. The maneouvre to be performed is then selected and the autopilot is armed, firstly from the ground station, and then by the safety pilot. The autopilot is then in completely control of the aircraft as the select aerobatic maneouvre is performed. Figure 6.12 shows the HIL simulated results for the loop trajectory. The figure shows the airspeed, angle of attack and pitch angle states. Also included are the commanded thrust and elevator deflections, and a graph to indicate when the autopilot was armed and disarmed.

The figure indicates the autopilot was armed just after 110 seconds, and disarmed just before 140 seconds. The pitch angle shows the loop was completed after 125 seconds, after which the aircraft was regulated about a straight and level trajectory for an additional 13 seconds. The graphs compare exceptionally well with the previous results shown in Figure 4.8, and in doing so verifies the autopilot's correct implementation. Although not shown in Figure 6.12, the autopilot was armed when the aircraft was flying East at a yaw angle of 95°. This implies that a meaningful East versus altitude graph can be plotted to obtain a better visual evaluation of the autopilot's performance. The graph is shown in Figure 6.13 and compares very well with the North versus altitude plot shown by Figure 4.9.

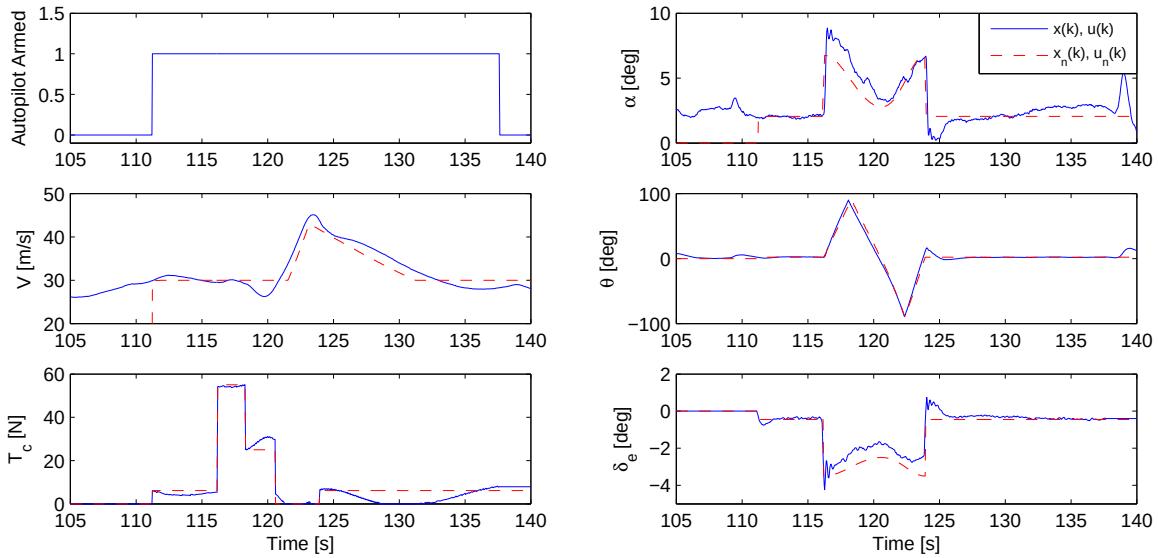


Figure 6.12: HIL Simulation Results for Loop Trajectory

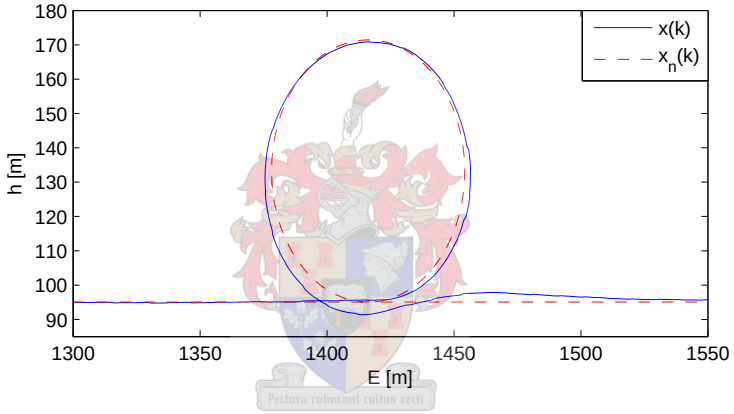


Figure 6.13: Altitude versus East Displacement for Loop Trajectory

6.6 Summary

This chapter presented the development of a hardware in the loop simulator. The process was started off by discussing the benefits of hardware in the loop simulation, after which an overview of the system was presented. Next, the simulation environment developed in Simulink was discussed. The hardware associated with the system and its interface with the simulation environment was then presented. Finally, the system was used to verify the correct implementation of the EKF and MPC algorithms.

The developed system is compatible with both avionics packages currently used by the UAV group. Performing successful hardware in the loop simulations instils confidence in the controller's ability to control the aircraft. The system extends the research group's simulation capabilities, and contributed greatly towards the successful flight tests demonstrated in this project and by [41].

Chapter 7

Flight Tests

This chapter presents practical results from two days of flight tests. Firstly, an overview of the various flight test days is provided and the goals for each day are stated. Telemetry data is then presented and analysed to evaluate the performance of the kinematic state estimator and model predictive controller.

7.1 Flight Test Overview

Flight tests were conducted at the Helderberg Radio Flyers Club with the Chief Flying Instructor, Dr. Kas Hamman, as safety pilot. A checklist, created by [35], was modified and used to conduct a range of pre-flight checks to make sure nothing was overlooked before leaving for the airfield. The checklist was also used to verify the integrity of the aircraft and avionics package between flight days.

7.1.1 Wednesday the 15th of November 2006

The first flight test was scheduled for the 15th of November 2006. However, the aircraft did not get to fly due to unfavourable wind conditions. Instead, time was spent setting up the various actuator limits on the transmitter, and programming the RC receiver fail-safe positions. The safety pilot also got acquainted with the modified airframe and was briefed on the various manoeuvres that need to be performed.

In order to obtain some results from the outing to the airfield, the estimator was armed after having established a position fix with the dual GPS antenna configuration. The magnetic reference field used by the estimator was then verified by rotating the aircraft through a series

Figure 7.1: Pictures of Aircraft After Take-off and Before Landing

of angles and inspecting the filter's attitude estimates. The day was concluded by performing some RC range tests to ensure the safety pilot will be able to control the aircraft at all times.

7.1.2 Friday the 17th of November 2006

Friday the 17th of November 2006 was the first time the aircraft flew with its complete avionics payload. The day involved the verification of the structural integrity of the airframe and the testing of the state estimator. A single flight test was performed during which all relevant tests were performed. The estimator was armed just before take-off after which its performance was monitored by means of the telemetry displayed on the ground station laptop.

Figure 7.2: Picture of Aircraft During Runway Flyby

After having familiarised himself with the aircraft, the safety pilot was instructed to perform various aerobatic manoeuvres. This allowed for nominal trajectories to be recorded for the

manoeuvres attempted in this project. Valuable information regarding the aircraft's thrust to weight ratio was also obtained. Having established confidence in the various state estimates, the decision was made to arm the straight and level flight controller. The transition between safety pilot and autopilot control appeared smooth and, apart from loosing some altitude, the controller regulated the aircraft about the straight and level trajectory. After the safety pilot landed the aircraft, wind conditions started to deteriorate rapidly and no further flights were conducted.

7.1.3 Friday the 24th of November 2006

Figure 7.3: Picture of Telemetry Downloading After Flight

The aim of this day was to arm the controller with the various aerobatic trajectories as flight paths. Only two flights were conducted during which completely autonomous aerobatic flight was demonstrated. The first flight involved obtaining confidence in the estimator's performance and the arming of the autopilot for the conventional flight and aileron roll trajectories. After some minor complications, the controller regulated the aircraft about the respective trajectories with acceptable precision.

The second flight involved the testing of the two remaining aerobatic maneouvres. Great excitement was experienced by the ground station operator and flight spectators after both maneouvres were performed completely autonomously. The rest of the flight was spent arming the controller with different headings to try and minimise the effects of the windy conditions experienced during the flight day. Many more days could have been spent evaluating the autopilot's performance but due to the arrival of the project deadline, it was decided that no further flight tests would be conducted.

7.2 Autopilot Results

Telemetry data representative of the estimator and controller's performance is presented and discussed in the following sections. When inspecting the aircraft states it should be noted that all motion variables are estimated by the EKF except for the angular velocity states which are measured directly by the rate gyroscopes.

7.2.1 Kinematic State Estimator

The kinematic state estimator forms an integral part of the controller's success. Using erroneous state estimates for feedback purposes will decrease the controller's performance or even cause the closed loop system to go unstable. The following sections briefly present state estimates, recorded sensor measurements and safety pilot commands for the aileron roll and loop maneouvres.

Aileron Roll

Consider Figure 7.4 which shows the measured roll rate, estimated roll angle, airspeed and altitude, and recorded safety pilot aileron and elevator commands for the aileron roll maneouvre.

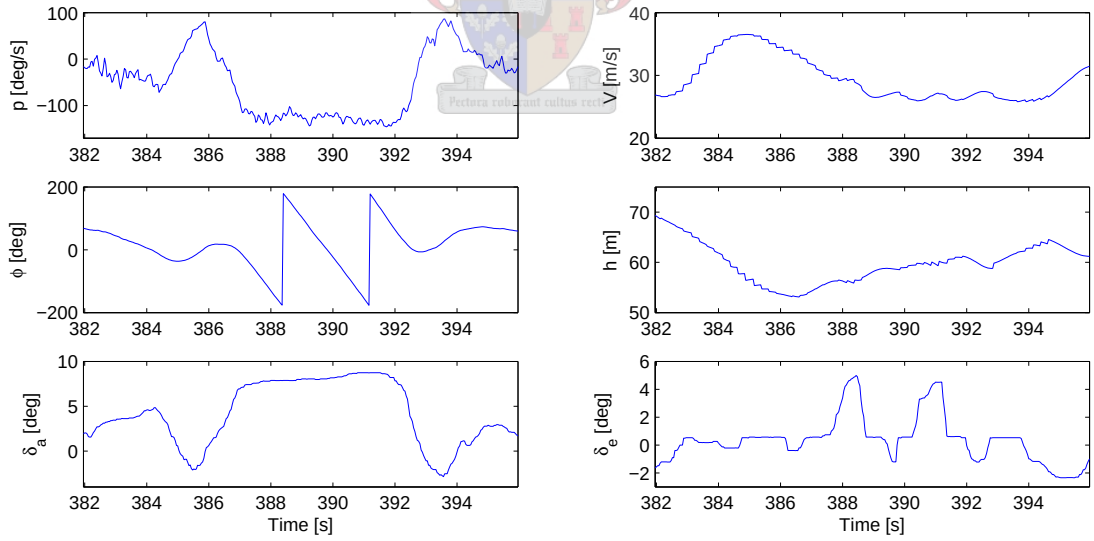


Figure 7.4: Recorded and Estimated Aircraft States for Aileron Roll Maneouvre

At 385 seconds the pilot starts to apply positive aileron causing the aircraft to roll at a constant rate of 130°/s. The aircraft is rolled through 720° as illustrated by the estimated roll angle graph. The initial increase in airspeed and decrease in altitude show how the pilot builds up the aircraft's momentum before entering the manoeuvre. Also interesting to note is the large

amount of commanded elevator when the aircraft is flying upside down. The visible jumps in the airspeed and altitude states are due to GPS updates becoming available at 4 Hz and used to correct the propagated state variables. Figure 7.4 also shows that the pilot flies the maneouvre at 27 m/s airspeed which is closely related to the 30 m/s airspeed used by the controller. Lastly, the commanded aileron and roll rate relationship compares reasonably well with the intuitive nominal aileron roll trajectory defined in Section 4.3.2.

Loop

Consider Figure 7.5 which shows the measured pitch rate, estimated pitch angle and airspeed, and recorded safety pilot throttle command for the loop maneouvre. The aircraft enters the loop at 457 seconds and completes the maneouvre at 472 seconds as shown by the estimated pitch angle.

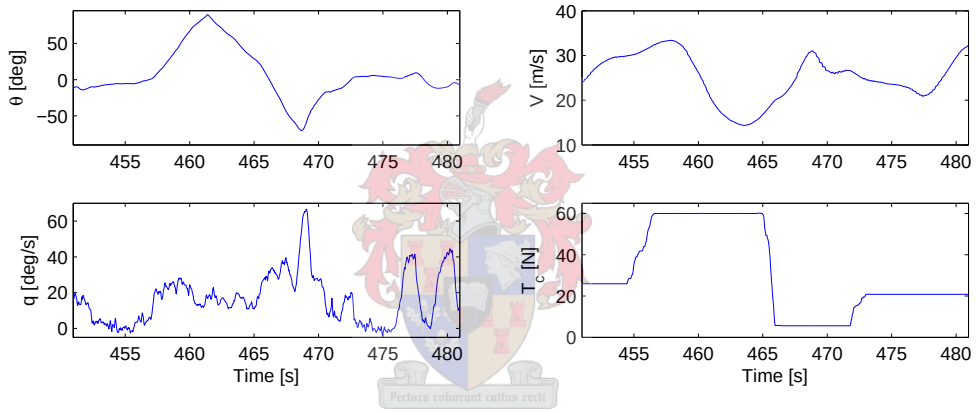


Figure 7.5: Recorded and Estimated Aircraft States for Loop Maneouvre

The airspeed and commanded thrust graphs indicate the airframe does not have enough thrust to maintain a constant airspeed during the first half of the loop. This can be largely contributed to the very large radius loop performed by the safety pilot. However, the inadequate engine thrust needs to be taken into account when the autopilot results for the loop trajectory are presented. Consider Figure 7.6 which shows the estimated North, West and altitude displacement of the aircraft for the loop maneouvre.

The aircraft approaches the loop from the right flying in a Southern direction. The constant drop in altitude during this stage corresponds well with the negative pitch angle between 451 and 457 seconds as shown in Figure 7.5. The position corrections due to GPS updates are once again clearly visible. Interesting to note are the significant corrections made as the aircraft passes the three quarter mark of the loop. Inspecting the estimated pitch angle at this point it is noticed that the angle is estimated as -75° instead of -90°. The state estimator uses the estimated attitude to coordinate the accelerometer measurements in the inertial reference frame.

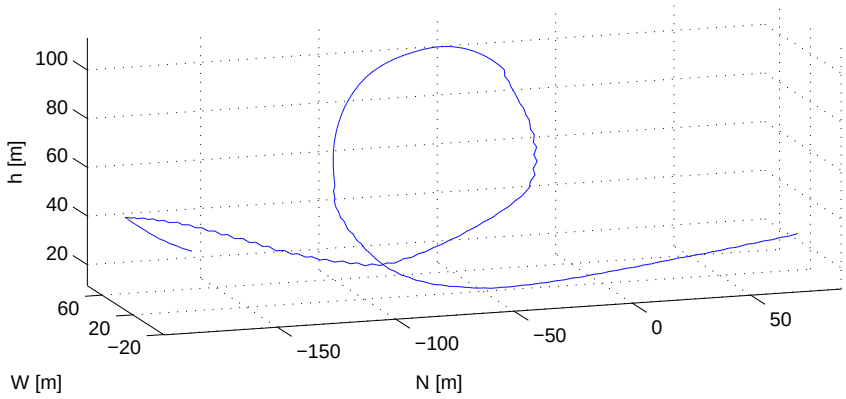


Figure 7.6: Estimated North, West and Altitude Displacement States for Loop Maneouvre

Should the attitude be erroneous, the time derivative of the NED velocity components will be incorrect which ultimately filters down to the position states. Due to time limitations, a detailed investigation into the erroneous pitch angle estimate could not be conducted. Further tests are required to track down possible causes for the incorrect state estimate. Lastly, the increase in altitude after completing the loop corresponds with the positive pitch angle after 473 seconds as shown in Figure 7.5.

7.2.2 Model Predictive Controller

The following sections are dedicated to analysing the model predictive controller's performance. In the graphs to follow, the nominal state and control values are indicated by dashed lines, whilst the estimated/measured states and final control values are portrayed by solid lines.

Conventional Flight

Figure 7.7 shows the estimated airspeed, angle of attack and altitude together with the commanded thrust and elevator deflection for the conventional flight trajectory. The autopilot is armed and disarmed at 416 and 425 seconds respectively. The figure firstly shows an initial loss of airspeed just after the controller is armed. This can be contributed to an incorrect nominal thrust setting as seen in the commanded thrust graph. The effect of the integral control on the airspeed error is clearly visible as the thrust is gradually increased from its nominal value in order to regulate the state with zero error. This response can be greatly improved by modelling the aircraft's engine more accurately. The 3 meter steady state error on the altitude is perfectly acceptable considering the aircraft's altitude of operation. From the angle of attack estimate it appears as if the controller is able to regulate the state with reasonable accuracy. This result instils confidence in the estimator's ability to estimate the incidence angles.

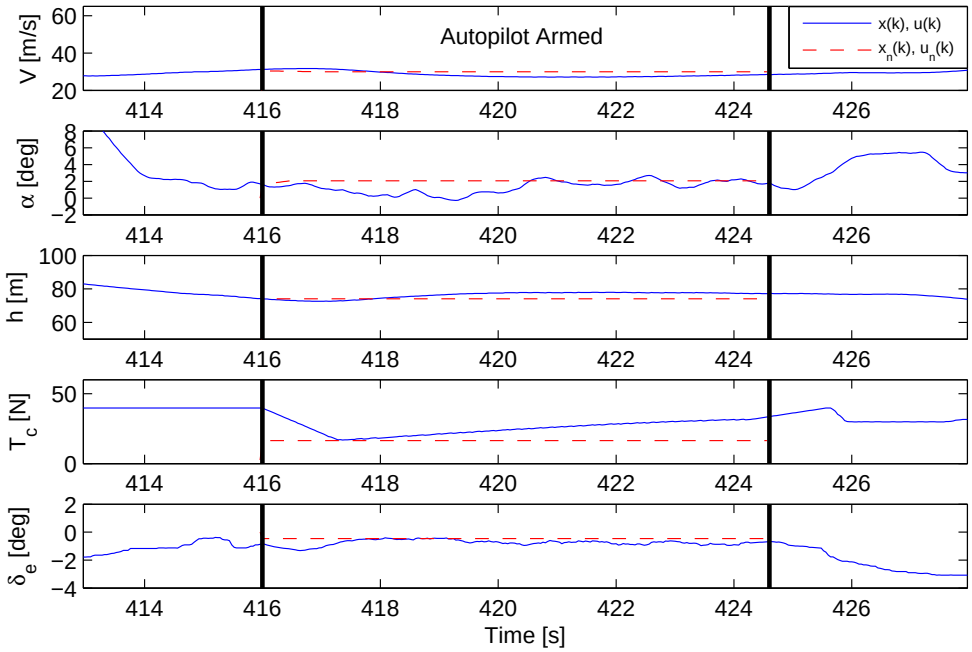


Figure 7.7: Flight Test Results for Level Flight Trajectory

Aileron Roll

Figure 7.8 shows the measured roll rate, estimated roll angle, altitude and angle of attack, as well as the aileron and elevator deflections for the aileron roll trajectory. The autopilot is armed and disarmed at 447 and 456 seconds respectively. For the paths leading in and out of the roll it is clear how the controller regulates the roll rate and roll angle to zero. At 451 seconds the 360° roll is performed in 2 seconds as can be seen in the roll rate, roll angle and commanded aileron graphs. As expected, the estimated altitude decreases throughout the roll and compares adequately with the simulation results presented in Section 4.3.2. Although specified intuitively, there is reasonable correlation between the nominal and actual angle of attack and commanded elevator deflection. This verifies the approximations used to obtain the nominal values for these states.

Loop

Figure 7.9 shows the estimated airspeed, pitch angle and angle of attack together with the commanded thrust and elevator deflection for the loop trajectory. The autopilot is armed and disarmed at 663 and 676 seconds respectively, whilst the loop is performed between 666 and 675 seconds as shown by the pitch angle graph. Inspecting the airspeed, it is noticed that the aircraft does not have enough thrust to maintain the nominal airspeed throughout the maneouvre. In spite of commanding 100% thrust during the first half of the loop, the estimated airspeed diviates

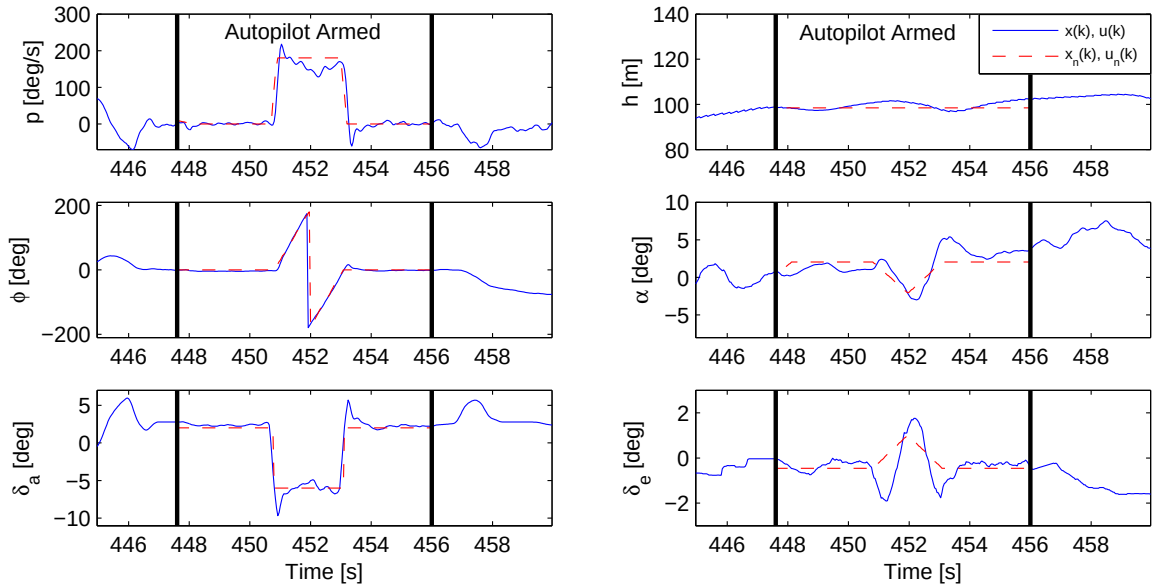


Figure 7.8: Flight Test Results for Aileron Roll Trajectory

significantly from the nominal value. Such a scenario can be avoided by defining a more realistic nominal trajectory which caters for the inadequate thrust and loss of airspeed during this part of the trajectory.

Having reached the top of the loop, the throttle is reduced significantly and the aircraft glides down the remaining part of the trajectory, regaining most of the lost airspeed. The approximate constant offset between the nominal and commanded elevator deflection indicates the elevator is more effective than what was calculated with AVL. This postulation can be verified by inspecting Figure 7.10 which shows the aircraft's estimated displacement states for the loop.

The figure indicates an average position error on the order of 8 meters around the loop. This is significantly different from the 3 meter error obtained from the simulation results presented in Section 4.3.3. The difference can be contributed to the erroneous nominal elevator deflection for the maneouvre. The elevator perturbations commanded by the controller can not completely compensate for the incorrect nominal, resulting in the aircraft flying a reduced radius loop. The origin of the problem goes all the way back to the calculation of the elevator's control derivative. This calculation needs to be revisited before the current aircraft model is used for further control system design. In spite of the reduced radius, the controller regulated the aircraft about the trajectory with acceptable accuracy.

Immelmann

Figure 7.11 shows the estimated pitch and roll angles together with the commanded elevator and aileron deflections for the Immelmann trajectory. The autopilot is armed and disarmed at

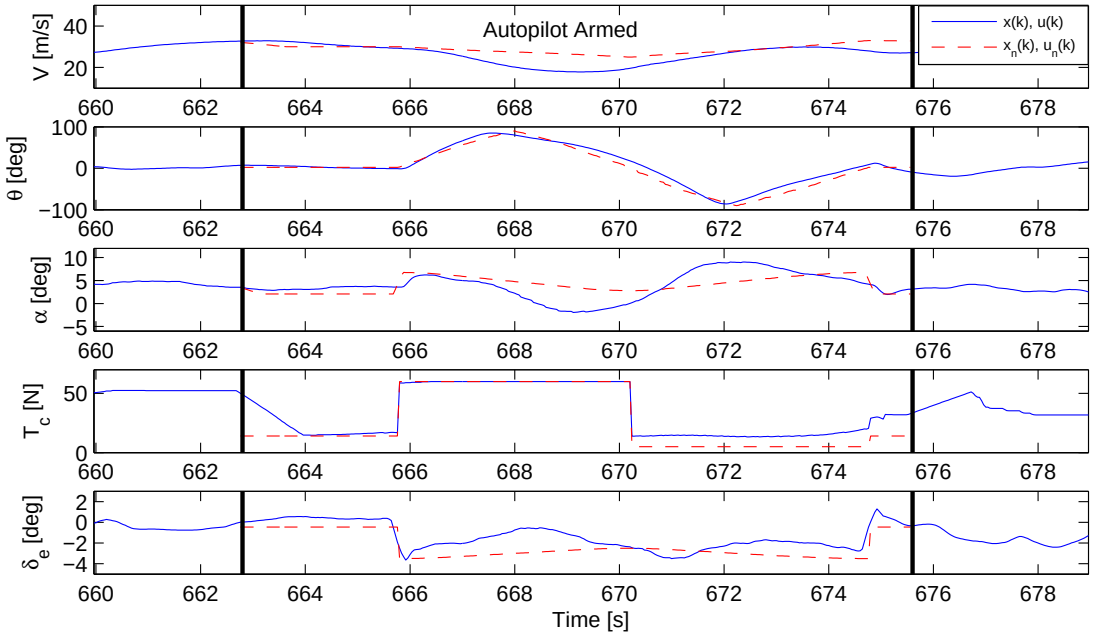


Figure 7.9: Flight Test Results for Loop Trajectory

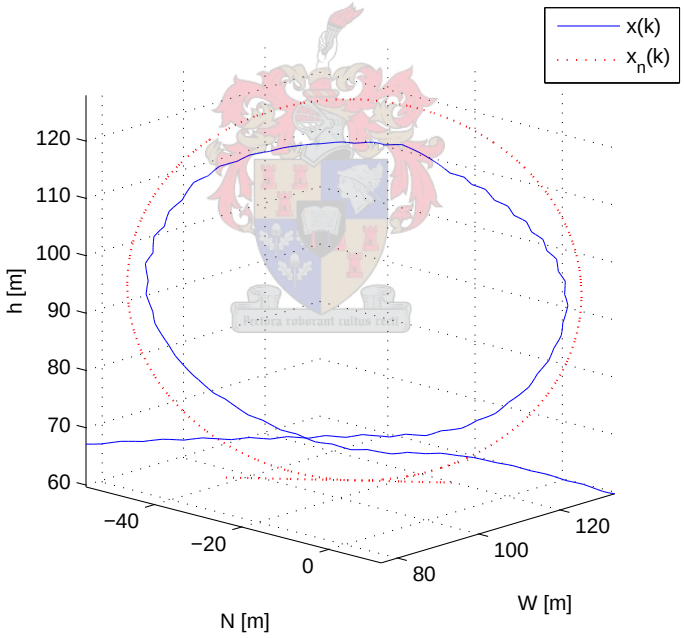


Figure 7.10: Flight Test Displacement States for Loop Maneouvre

561 and 573 seconds respectively.

The half loop is performed between 565 and 569 seconds after which the 180° roll is performed in 1 second. In order to obtain a better visual representation of the autopilot's performance for the Immelmann trajectory, consider Figure 7.12 which shows shows the aircraft's estimated displacement states.

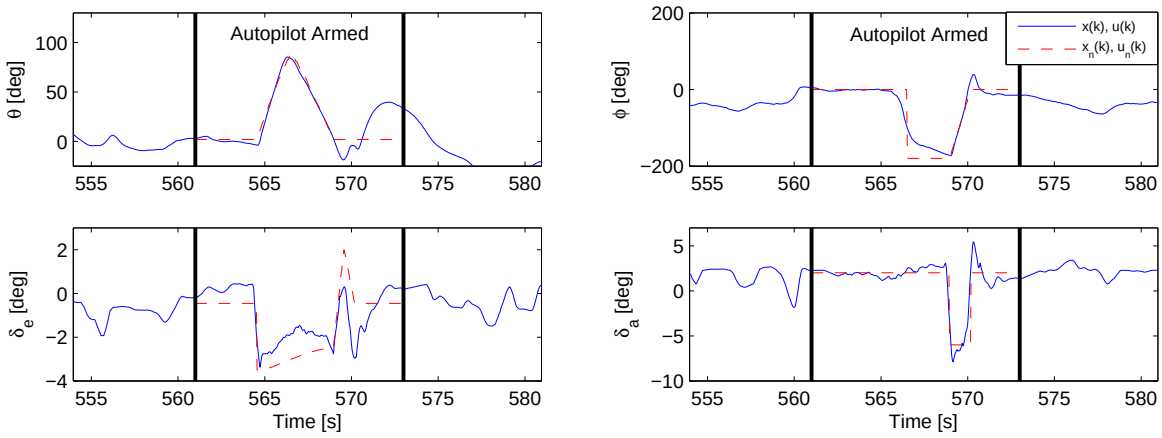


Figure 7.11: Flight Test Results for Immelmann Trajectory

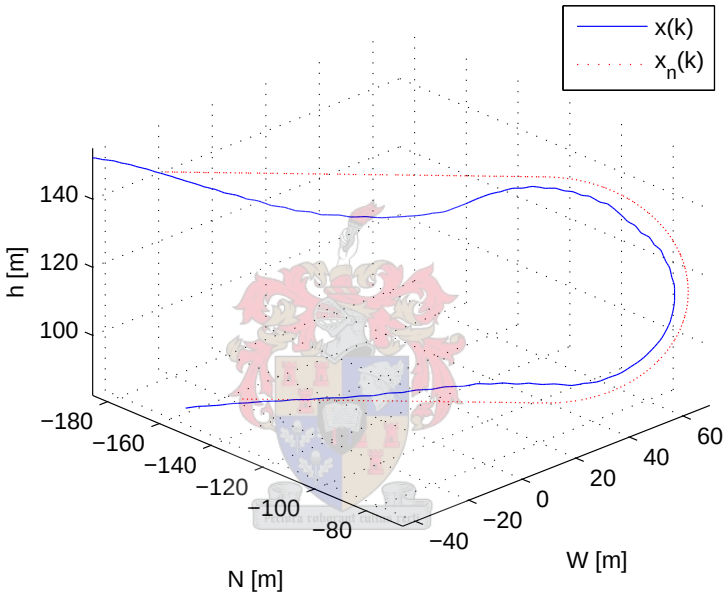


Figure 7.12: Flight Test Displacement States for Immelmann Maneouvre

The steady state position error during the half loop can be contributed to the same reasons discussed in the previous section. As the 180° roll is performed, the aircraft experiences significant loss of altitude. This drop is altitude corresponds well with the negative estimated pitch angle during this period as shown in Figure 7.11. After having completed the roll, the controller starts to correct the error in altitude as portrayed by the 40° deviation in pitch angle.

7.3 Summary

This chapter described and analysed two days of flight tests during which the estimator and controller algorithms were shown to work in practice. The autopilot results compared well with the non-linear simulation results presented in Chapter 4. Whenever significant differences were

noted, possible causes were discussed together with solutions to the anomalies. The rapid success of the flight tests can be mainly contributed to the numerous non-linear hardware in the loop simulations performed beforehand.

Chapter 8

Summary and Recommendations

8.1 Summary

This thesis reported two years of work to try and push the state of the art in aircraft flight control at the University of Stellenbosch. The focus was on the design and implementation of an autopilot to perform a range of aerobatic manoeuvres. A simplified 6-DOF non-linear aircraft model was developed which encapsulated the important dynamic characteristics of the airframe. The project was restricted to the use of low cost, readily available sensors providing a limited subset of the aircraft's state vector. The adopted control strategy required full state feedback and led to the design of a full state estimator. An extended Kalman filter was implemented to estimate the complete aircraft state vector by combining the low quality sensor measurements in an optimal manner.

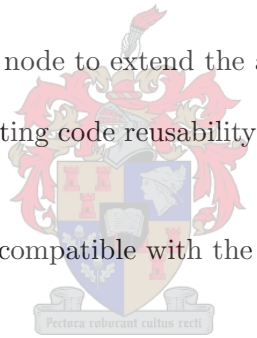
A model predictive control strategy was then used to regulate the aircraft about arbitrary, time varying trajectories in six degrees of freedom. This globally applicable control law was used to demonstrate how three well known aerobatic manoeuvres can be performed completely autonomously and with great precision. The controller configuration is not in any way specific to the manoeuvres presented in this thesis. The strategy can be used to perform any aerobatic manoeuvre as long as feasible state and control trajectories can be specified. The estimator and controller algorithms were then evaluated by numerous non-linear simulations.

The avionics and ground station used for the practical implementation of these algorithms were then presented. A hardware in the loop simulator was developed and used to verify the correct implementation of the autopilot. Finally, practical results from two days of flight tests were presented. The success achieved during the limited number of flight tests can be contributed firstly to the numerous non-linear simulations performed before hand. Also, the autopilot's im-

plementation was validated extensively by means of numerous hardware in the loop simulations.

The following tools developed during this project will contribute towards future UAV work,

- Construction and modification of an aerobatic aircraft to serve as a demonstrator vehicle for more advanced control algorithms.
- The calculation of an aerodynamic model using computational fluid dynamics software. The adopted method can be used in conjunction with the modelling software developed by [35] to verify the validity of the model.
- Vehicle independent kinematic state estimator able to estimate an arbitrary platform's position, velocity and attitude.
- A comprehensive non-linear simulation environment.
- A hardware in the loop simulator compatible with both avionics packages developed by the research group.
- Pressure/Magnetometer CAN node to extend the avionics' sensing capabilities.
- Modular OBC software promoting code reusability and simplifying integration into future projects.
- Ground station GUI software compatible with the developed OBC software.



8.2 Recommendations

Recommendations on how the current autopilot can be improved and extended are discussed in point form below,

- When the adopted MPC strategy was presented, it was argued why it is beneficial to have a long prediction horizon. The current, relatively short horizon can be extended by firstly decreasing the controller's sample rate to 25 Hz which is still much faster than the system's bandwidth. Also, by using the first couple of feedback gains of the solution to the Riccati equations, significantly more processing time becomes available to obtain a solution over an extended horizon. Combining these two methods, the author is confident that a horizon length in the order of 2 seconds is obtainable. Finally, [29] proposes alternatives to the complete non-linear MPC strategy as was used in this project. These methods can further reduce the required processing power but remain to be investigated in future projects.

- Only three aerobatic manoeuvres have been performed up to date. The controller's versatility needs to be evaluated further by performing other, more complex manoeuvres. Also, Chapter 4 proposed four methods to obtain the nominal state and control variables. This project has demonstrated that adequate trajectories can be constructed using intuitive sense only. Future projects are left with the task of exploring the remaining three options. The recording of the aircraft states and safety pilot commands requires no additional infrastructure and subsequently should be explored first.

- Should the autopilot want to be used in windy conditions, alternative methods will have to be investigated to obtain wind dependant states. The current hardware configuration has the capability to measure the airframe's airspeed directly. [36] proposes a method to obtain the angle of attack by using a linearised aircraft model to relate the z-axis accelerometer measurement to the required incidence angle. It might be possible to extend such a method to obtain the incidence angles under windy conditions.

- The OBC can be used to perform periodic set/reset cycles on the magnetometer. This will ensure that the resistive perm alloy strips stay polarised correctly throughout an entire flight. However, it would not be possible to recalculate the inherent sensor offsets since the measured magnetic field is not constant during flight.

- The engine model needs to be improved by incorporating the loss of thrust with an increase in airspeed as used by [36]. This will further improve the MPC's performance since the controller will be assured of the requested amount of thrust at all times. A higher fidelity aerodynamic model can also be implemented by using a lookup table to provide stability and control derivatives for extreme flight conditions like high angles of attack and fast angular velocities.

- During flight tests the dual GPS antenna configuration proved to be somewhat problematic in that the receiver could not establish a position fix at all times. This is most probably due to the fact that the current antenna setup is not a completely omnidirectional configuration. In order to keep the EKF state propagation periods as short as possible, it is highly recommended that a proper omnidirectional GPS antenna be used.

- The current OBC and ground station software provides the functionality for the GPS module to operate as a differential receiver. Extensive testing is required to establish if measurement accuracies increase significantly during differential operation. Should this be the case, the more accurate GPS updates can be used to further increase the state estimator's accuracy.

- Chapter 3 showed that bias drifts on the rate gyroscopes decreased the accuracy of the estimated attitude. Should more accuracy be required for these states, the filter can be

augmented to include bias states for the three rate gyroscopes. If the augmented filter is not observable with the current GPS and magnetometer measurements, temperature calibrations need to be performed on the rate gyroscopes to establish a relationship between the bias drift and temperature output of each sensor. The temperature measurements can then be used as direct measurements of the bias states in the Kalman filter configuration. The estimated biases can then also be subtracted from the rate gyroscope measurements to obtain more accurate angular rate measurements.

Bibliography

[1] M.V. Cook. *Flight Dynamics Principles*. Elsevier Butterworth-Heinemann, 1997.

[2] R.S. Shevell. *Fundamentals of Flight*. Prentice-Hall, 1983.

[3] B. Etkin, L.D. Reid. *Dynamics of Flight, Stability and Control, 3rd ed*. John Wiley & Sons, 1996.

[4] P.H. Zipfel. *Modelling and Simulation of Aerospace Vehicle Dynamics*. AIAA Education Series, 2000.

[5] J. Wittenburg. *Dynamics of Systems of Rigid Bodies*. BG Teubner, 1977.

[6] M.J. Sidi. *Spacecraft Dynamics and Control*. Cambridge Aerospace Series, 1997.

[7] G.F. Franklin, J.D. Powell, M. Workman. *Digital Control of Dynamic Systems*. Addison Wesely Longman, 1998.

[8] G.F. Franklin, J.D. Powell, A. Emami-Naeini. *Feedback Control of Dynamic Systems*. Addison-Wesely Publishing Company, 1994.

[9] C.L. Philips, H.T. Nagle. *Digital Control System Analysis and Design*. Prentice-Hall, 1995.

[10] K. Ogata. *Modern Control Engineering*. Prentice-Hall, 1970.

[11] A.E. Bryson. *Control of Spacecraft and Aircraft*. Princeton University Press, 1994.

[12] J.H. Blakelock. *Automatic Control of Aircraft and Missiles, 2nd ed*. Wiley-Interscience, 1991.

[13] D.E. Kirk. *Optimal Control Theory - An Introduction*. Prentice-Hall, 1970.

[14] A.E. Bryson, Jr., Y. Ho. *Applied Optimal Control. Optimization, Estimation, and Control*. Ginn and Company, 1969.

[15] P. Zarchan, H. Musoff. *Fundamentals of Kalman Filtering - A Practical Approach*. Volume 190 of Progress in Astronautics and Aeronautics. AIAA, 2000.

[16] A. Gelb. *Applied Optimal Estimation*. MIT Press. 1974.

[17] B. Vaglienti, M. Niculescu. *Hardware in the loop simulator for the Piccolo avionics*. Cloud Cap Technology, 2004.

[18] M. Drela, H. Youngren. *AVL 3.14 User Primer*. MIT Aero & Astro, 2004.

[19] P.Z. Peebles, Jr. *Probability, Random Variables and Random Signal Principles*. McGraw-Hill Companies, 2001.

[20] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery. *Numerical Recipes in C - The Art of Scientific Computing*. Cambridge University Press, 1988.

[21] J.W. Nilsson, S.A. Riedel. *Electric Circuits*. Prentice-Hall, 1996.

[22] D.A. Neamen. *Electronic Circuit Analysis and Design*. McGraw-Hill Companies, 1996.

[23] T. Jones. *Advanced Estimation 813 - Course Notes*. University of Stellenbosch, 2005.

[24] M.S. Triantafyllou, F.S. Hover. *Maneuvering and Control of Marine Vehicles - Course Notes*. Massachusetts Institute of Technology, 2003.

[25] M.C. Koen. *Modelling and Simulation of an RPV for Flight Control System Design Purposes*. Department of Electronic and Computer Engineering, University of Pretoria, 2006.

[26] G.W. Milne. *Simplifying Vector Rotations and Coordination*. AIAA 2001-4310, 2005.

[27] E.J. Lefferts, F.L. Markley, M.D. Shuster. *Kalman Filtering for Spacecraft Attitude Estimation*. Journal of Guidance, Control and Dynamics, Sept/Oct 1982.

[28] M.D. Shuster, S.D. Oh. *Three-Axis Attitude Determination from Vector Observations*. Journal of Guidance, Control and Dynamics, Jan/Feb 1981.

[29] T. Keviczky, G.J. Balas. *Receding Horizon Control Of An F-16 Aircraft: A Comparative Study*. Department of Aerospace Engineering and Mechanics, University of Minnesota.

[30] P. Lu. *Regulation About Time-Varying Trajectories: Precision Entry Guidance Illustrated*. Journal of Guidance, Control and Dynamics, Nov/Dec 1999.

[31] M. Pachter, P.R. Chandler, L. Smith. *Maneuvering Flight Control*. Journal of Guidance, Control and Dynamics, May/June 1998.

[32] D. Gebre-Egziabher, D. Elkaim, J.D. Powell, B.W. Parkinson. *A non-linear, two-step estimation algorithm for calibrating solid-state strapdown magnetometers.* Proceedings of the 8th international conference on integrated navigation systems, May 2001.

[33] A. Budiyono. *Design and Development of Autonomous Uninhabited Air Vehicles at ITB: Challenges and Progress Status.* Paper presented at the Aerospace Indonesia Meeting, July 2005.

[34] K.Y. Lou, A.E. Bryson. *Inverse and Optimal Control for Precision Aerobatic Maneuvers.* Journal of Guidance, Control and Dynamics, March/April 1996.

[35] I.K. Peddle. *Autonomous Flight of a Model Aircraft.* Masters dissertation, University of Stellenbosch, 2005.

[36] S. Park. *Avionics and Control System Development for Mid-Air Rendezvous of Two Unmanned Aerial Vehicles.* Ph.D dissertation, Massachusetts Institute of Technology, 2004.

[37] N. Carstens. *Development of a Low-Cost and Low-Weight Flight Control System for an Electrically Powered Free-Flying Model Helicopter.* Masters dissertation, University of Stellenbosch, 2004.

[38] S. Groenewald. *Development of a Rotary-Wing Test Bed for Autonomous Flight.* Masters dissertation, University of Stellenbosch, 2006.

[39] J. Venter. *Development of an Experimental Tilt-Wing VTOL Unmanned Aerial Vehicle.* Masters dissertation, University of Stellenbosch, 2006.

[40] J. Bijker. *Development of an Attitude Heading Reference System for an Air Ship.* Masters dissertation, University of Stellenbosch, 2006.

[41] J.C. Roos. *Autonomous Take-Off and Landing of an Unmanned Aerial Vehicle.* Masters dissertation, University of Stellenbosch, 2006.

[42] S. Park. *Examples of Estimation Filters from Recent Projects at MIT.* Advanced Estimation for GPS and Inertial Navigation - 16.324, 2004.

[43] M.C. Silberbauer. *Simulation Visualisation System.* Undergraduate Final Year Project, University of Stellenbosch, 2005.

[44] C. Nel. *Real-time Communications Link.* Undergraduate Final Year Project, University of Stellenbosch, 2004.

[45] D. Palmer. Vacation work using AVL. 2004.

[46] NASA History web page. `http://history.nasa.gov/SP-367/appendc.htm`. 2006.

[47] National Geophysical Data Centre web page. `http://www.ngdc.noaa.gov/seg/geomag/models.shtml`. 2006.

[48] Cloud Cap Technology web page. `http://www.cloudcaptech.com/`. 2006.

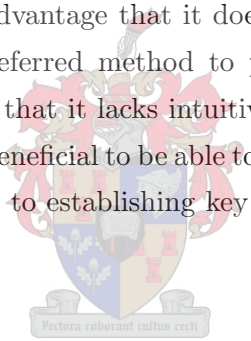
[49] ETH Zurich web page. http://www.imrt.ethz.ch/research/uav. 2006.

[50] Acrobatic Flight and the Frecce Tricolor web page.
http://www.malignani.ud.it/WebEnis/aer/history.html. 2006.

Appendix A

Euler Angles and Quaternion Relationships

The quaternion has the dominant advantage that it does not suffer from any singularities and subsequently was chosen as the preferred method to present attitude in this project. The drawback of using the quaternion is that it lacks intuitive sense. Due to the ease of visualizing the Euler angle representation, it is beneficial to be able to convert between the two presentations. The following sections are dedicated to establishing key relationships between quaternions and Euler angles.



A.1 Converting a Quaternion to Euler Angles

Given the attitude of the aircraft in terms of a quaternion, find the corresponding Euler angles which will represent the same attitude. According to [23], transformation matrices are unique and imply that if a transformation matrix in terms of a quaternion is equal to a transformation matrix in terms of Euler angles, then the corresponding quaternion and Euler angles must

represent the same attitude. This result is stated below,

$$\begin{aligned}
\boldsymbol{T} &= \boldsymbol{T}(\boldsymbol{q}) = \boldsymbol{T}(\phi, \theta, \psi) \\
&= \begin{bmatrix} q_4^2 + q_1^2 - q_2^2 - q_3^2 & 2\left(q_4 q_3 + q_1 q_2\right) & 2\left(q_1 q_3 - q_4 q_2\right) \\ 2\left(q_1 q_2 - q_4 q_3\right) & q_4^2 - q_1^2 + q_2^2 - q_3^2 & 2\left(q_4 q_1 + q_2 q_3\right) \\ 2\left(q_4 q_2 + q_1 q_3\right) & 2\left(q_2 q_3 - q_4 q_1\right) & q_4^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \\
&= \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix} \\
&= \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}
\end{aligned} \tag{A.1}$$

By manipulating the matrix elements in equation (A.1), the following results are obtained,

$$\begin{aligned}
\frac{A_{23}}{A_{33}} &= \frac{\sin\phi\cos\theta}{\cos\phi\cos\theta} = \frac{2\left(q_4 q_1 + q_2 q_3\right)}{q_4^2 - q_1^2 - q_2^2 + q_3^2} \\
\Rightarrow \phi &= \arctan\left(\frac{2\left(q_4 q_1 + q_2 q_3\right)}{q_4^2 - q_1^2 - q_2^2 + q_3^2}\right)
\end{aligned} \tag{A.2}$$

$$\begin{aligned}
A_{13} &= -\sin\theta = 2\left(q_1 q_3 - q_4 q_2\right) \\
\Rightarrow \theta &= \arcsin\left(-2\left(q_1 q_3 - q_4 q_2\right)\right)
\end{aligned} \tag{A.3}$$

$$\begin{aligned}
\frac{A_{12}}{A_{11}} &= \frac{\cos\theta\sin\psi}{\cos\theta\cos\psi} = \frac{2\left(q_4 q_3 + q_1 q_2\right)}{q_4^2 + q_1^2 - q_2^2 - q_3^2} \\
\Rightarrow \psi &= \arctan\left(\frac{2\left(q_4 q_3 + q_1 q_2\right)}{q_4^2 + q_1^2 - q_2^2 - q_3^2}\right)
\end{aligned} \tag{A.4}$$

A.2 Converting Euler Angles to a Quaternion

The next important relationship that needs to be established is, given the attitude in terms of Euler angles, how one should determine the rotation quaternion that yields the same attitude representation. Recalling the three Euler rotations and the order in which they were performed, three separate quaternions can be set up for each of the Euler rotations. Finally, the quaternions are combined to obtain the equivalent attitude representation.

Yaw Quaternion

Starting off with the $OX_IY_IZ_I$ axis system, define the axis to be rotated about as the OZ_I-axis and the rotation angle as the Euler yaw angle, ψ, as demonstrated in Figure A.1.

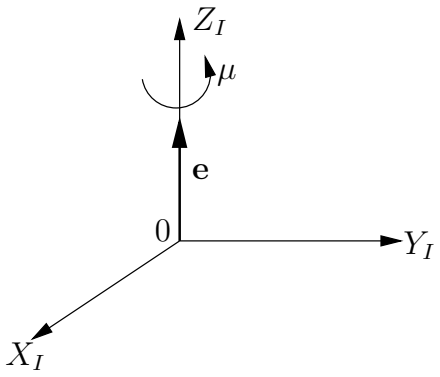


Figure A.1: Quaternion Yaw Rotation

According to the quaternion definition in equation (2.22),

$$\boldsymbol{e} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$$
$$\mu = \psi$$

and the yaw rotation quaternion is now given by,

$$\boldsymbol{q}_{\boldsymbol{\psi}} = \begin{bmatrix} 0 \\ 0 \\ \sin\left(\frac{\psi}{2}\right) \\ \cos\left(\frac{\psi}{2}\right) \end{bmatrix} \tag{A.5}$$

Pitch Quaternion

After the yaw rotation, define the axis to be rotated about as the OY_1-axis and the rotation angle as the Euler pitch angle, θ, as demonstrated in Figure A.2.

Using the definition,

$$\boldsymbol{e} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T$$
$$\mu = \theta$$

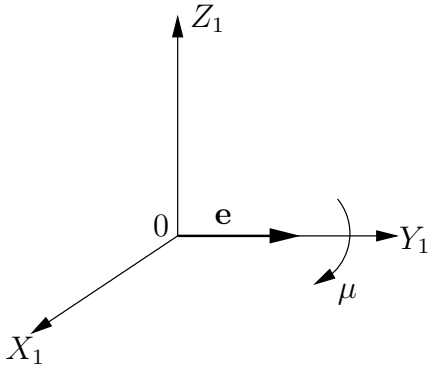


Figure A.2: Quaternion Pitch Rotation

and the pitch rotation quaternion is given by,

$$\boldsymbol{q_\theta} = \begin{bmatrix} 0 \\ \sin\left(\frac{\theta}{2}\right) \\ 0 \\ \cos\left(\frac{\theta}{2}\right) \end{bmatrix} \tag{A.6}$$

Roll Quaternion

Lastly define the axis to be rotated about as the OX_2-axis and the rotation angle as the Euler roll angle, ϕ, as demonstrated in Figure A.3.

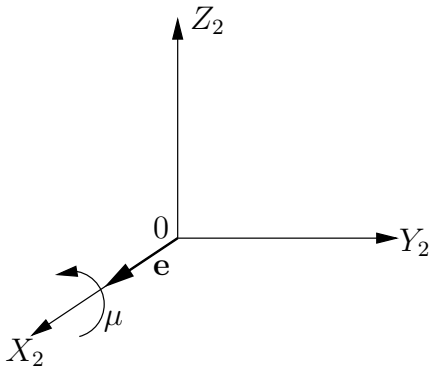


Figure A.3: Quaternion Roll Rotation

Again using the definition,

$$\boldsymbol{e} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$$

$$\mu = \psi$$

and the roll rotation quaternion is given by,

$$\boldsymbol{q}_{\boldsymbol{\phi}} = \begin{bmatrix} \sin\left(\frac{\phi}{2}\right) \\ 0 \\ 0 \\ \cos\left(\frac{\phi}{2}\right) \end{bmatrix} \tag{A.7}$$

Combining Quaternion Rotations

In order to combine the three quaternions, a definition for quaternion multiplication is required. [27] states that given two quaternions,

$$\boldsymbol{q} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^T$$

$$\boldsymbol{q'} = \begin{bmatrix} q'_1 & q'_2 & q'_3 & q'_4 \end{bmatrix}^T$$

the product of $\boldsymbol{q}$ and $\boldsymbol{q'}$ is given by,

$$\boldsymbol{q} \otimes \boldsymbol{q'} = \begin{bmatrix} q_4 & q_3 & -q_2 & q_1 \\ -q_3 & q_4 & q_1 & q_2 \\ q_2 & -q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix} \begin{bmatrix} q'_1 \\ q'_2 \\ q'_3 \\ q'_4 \end{bmatrix} \tag{A.8}$$

where $\otimes$ denotes the quaternion multiplication operator.

Using equation (A.8), the yaw, pitch and roll quaternions found in equations (A.5), (A.6) and (A.7) respectively, are combined as follows,

$$\begin{aligned} \boldsymbol{q}(\phi, \theta, \psi) &= \boldsymbol{q}_{\boldsymbol{\phi}} \otimes \boldsymbol{q}_{\boldsymbol{\theta}} \otimes \boldsymbol{q}_{\boldsymbol{\psi}} \\ &= \begin{bmatrix} \cos\left(\frac{\psi}{2}\right)\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\phi}{2}\right) - \sin\left(\frac{\psi}{2}\right)\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\phi}{2}\right) \\ \cos\left(\frac{\psi}{2}\right)\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\phi}{2}\right) + \sin\left(\frac{\psi}{2}\right)\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\phi}{2}\right) \\ \sin\left(\frac{\psi}{2}\right)\cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\phi}{2}\right) - \cos\left(\frac{\psi}{2}\right)\sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\phi}{2}\right) \\ \cos\left(\frac{\psi}{2}\right)\cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\phi}{2}\right) + \sin\left(\frac{\psi}{2}\right)\sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\phi}{2}\right) \end{bmatrix} \end{aligned} \tag{A.9}$$

A.3 Extracting the Quaternion from a Transformation Matrix

By manipulation of the elements on the diagonal of the quaternion transformation matrix in equation (A.1), and using the definition of the norm of a quaternion stated in equation (2.23),

equations for the absolute values of the four quaternion components can be found given any transformation matrix. These are stated below,

$$\begin{aligned} q_1 &= \frac{1}{2}\sqrt{1 + A_{11} - A_{22} - A_{33}} \\ q_2 &= \frac{1}{2}\sqrt{1 - A_{11} + A_{22} - A_{33}} \\ q_3 &= \frac{1}{2}\sqrt{1 - A_{11} - A_{22} + A_{33}} \\ q_4 &= \frac{1}{2}\sqrt{1 + A_{11} + A_{22} + A_{33}} \end{aligned} \tag{A.10}$$

To get the correct signs of the quaternion elements, one of the elements needs to be selected as a pivot element and is then used to calculate the other elements. The largest element is usually chosen because it yields the best accuracy, as is shown in the calculations to follow.

Using q_1 as pivot element,

$$q_1 = q_1 \tag{A.11}$$

$$q_2 = \frac{A_{12} + A_{21}}{4q_1} \tag{A.12}$$

$$q_3 = \frac{A_{13} + A_{31}}{4q_1} \tag{A.13}$$

$$q_4 = \frac{A_{23} - A_{32}}{4q_1} \tag{A.14}$$

Using q_2 as pivot element,

$$q_1 = \frac{A_{12} + A_{21}}{4q_2} \tag{A.15}$$

$$q_2 = q_2 \tag{A.16}$$

$$q_3 = \frac{A_{23} + A_{32}}{4q_2} \tag{A.17}$$

$$q_4 = \frac{A_{31} - A_{13}}{4q_2} \tag{A.18}$$

Using q_3 as pivot element,

$$q_1 = \frac{A_{13} + A_{31}}{4q_3} \tag{A.19}$$

$$q_2 = \frac{A_{23} + A_{32}}{4q_3} \tag{A.20}$$

$$q_3 = q_3 \tag{A.21}$$

$$q_4 = \frac{A_{12} - A_{21}}{4q_3} \tag{A.22}$$

Using q_4 as pivot element,

$$q_1 = \frac{A_{23} - A_{32}}{4q_4} \tag{A.23}$$

$$q_2 = \frac{A_{31} - A_{13}}{4q_4} \tag{A.24}$$

$$q_3 = \frac{A_{12} - A_{21}}{4q_4} \tag{A.25}$$

$$q_4 = q_4 \tag{A.26}$$

A.4 Tracing Variances between the Quaternion and Euler Angles

Given a rotation quaternion, $\boldsymbol{q}$, with an error covariance matrix, $P_{\boldsymbol{q}}$, the aim is to find the error covariance matrix of the corresponding Euler angles. The first step is to linearise equations (A.2), (A.3) and (A.4) in the vicinity of $\boldsymbol{q}$. This is done by setting up the Jacobian matrix and evaluating it at $\boldsymbol{q}$,

$$A = \begin{bmatrix} \frac{\partial \phi}{\partial q_1} & \frac{\partial \phi}{\partial q_2} & \frac{\partial \phi}{\partial q_3} & \frac{\partial \phi}{\partial q_4} \\ \frac{\partial \theta}{\partial q_1} & \frac{\partial \theta}{\partial q_2} & \frac{\partial \theta}{\partial q_3} & \frac{\partial \theta}{\partial q_4} \\ \frac{\partial \psi}{\partial q_1} & \frac{\partial \psi}{\partial q_2} & \frac{\partial \psi}{\partial q_3} & \frac{\partial \psi}{\partial q_4} \end{bmatrix}_{\boldsymbol{q}} \tag{A.27}$$

Now that a linear relationship exists between the quaternion and the corresponding Euler angles, [19] provides a method to perform linear transformations of Gaussian random variables which is stated as,

$$P_{\phi,\theta,\psi} = A P_{\boldsymbol{q}} A^T \tag{A.28}$$

where $P_{\phi,\theta,\psi}$ is the error covariance matrix of the Euler angles. The Jacobian matrix entries are defined as follows,

$$\frac{\partial \phi}{\partial q_1} = \frac{2q_4\left(q_4^2 - q_1^2 - q_2^2 + q_3^2\right) + 4q_1\left(q_4 q_1 + q_2 q_3\right)}{\left(q_4^2 - q_1^2 - q_2^2 + q_3^2\right)^2 + 4\left(q_4 q_1 + q_2 q_3\right)^2},$$

$$\frac{\partial \phi}{\partial q_2} = \frac{2q_3\left(q_4^2 - q_1^2 - q_2^2 + q_3^2\right) + 4q_2\left(q_4 q_1 + q_2 q_3\right)}{\left(q_4^2 - q_1^2 - q_2^2 + q_3^2\right)^2 + 4\left(q_4 q_1 + q_2 q_3\right)^2},$$

$$\frac{\partial \phi}{\partial q_3} = \frac{2q_2\left(q_4^2 - q_1^2 - q_2^2 + q_3^2\right) - 4q_3\left(q_4 q_1 + q_2 q_3\right)}{\left(q_4^2 - q_1^2 - q_2^2 + q_3^2\right)^2 + 4\left(q_4 q_1 + q_2 q_3\right)^2},$$

$$\frac{\partial \phi}{\partial q_4} = \frac{2q_1\left(q_4^2 - q_1^2 - q_2^2 + q_3^2\right) - 4q_4\left(q_4 q_1 + q_2 q_3\right)}{\left(q_4^2 - q_1^2 - q_2^2 + q_3^2\right)^2 + 4\left(q_4 q_1 + q_2 q_3\right)^2},$$

$$\frac{\partial\theta}{\partial q_1} = \frac{-2q_3}{\sqrt{1-4\left(q_1q_3-q_4q_2\right)^2}}, \quad \frac{\partial\theta}{\partial q_2} = \frac{2q_4}{\sqrt{1-4\left(q_1q_3-q_4q_2\right)^2}}$$

$$\frac{\partial\theta}{\partial q_3} = \frac{-2q_1}{\sqrt{1-4\left(q_1q_3-q_4q_2\right)^2}}, \quad \frac{\partial\theta}{\partial q_4} = \frac{2q_2}{\sqrt{1-4\left(q_1q_3-q_4q_2\right)^2}}$$

$$\frac{\partial\psi}{\partial q_1} = \frac{2q_2\left(q_4^2+q_1^2-q_2^2-q_3^2\right)-4q_1\left(q_4q_3+q_1q_2\right)}{\left(q_4^2+q_1^2-q_2^2-q_3^2\right)^2+4\left(q_4q_3+q_1q_2\right)^2}$$

$$\frac{\partial\psi}{\partial q_2} = \frac{2q_1\left(q_4^2+q_1^2-q_2^2-q_3^2\right)+4q_2\left(q_4q_3+q_1q_2\right)}{\left(q_4^2+q_1^2-q_2^2-q_3^2\right)^2+4\left(q_4q_3+q_1q_2\right)^2}$$

$$\frac{\partial\psi}{\partial q_3} = \frac{2q_4\left(q_4^2+q_1^2-q_2^2-q_3^2\right)+4q_3\left(q_4q_3+q_1q_2\right)}{\left(q_4^2+q_1^2-q_2^2-q_3^2\right)^2+4\left(q_4q_3+q_1q_2\right)^2}$$

$$\frac{\partial\psi}{\partial q_4} = \frac{2q_3\left(q_4^2+q_1^2-q_2^2-q_3^2\right)-4q_4\left(q_4q_3+q_1q_2\right)}{\left(q_4^2+q_1^2-q_2^2-q_3^2\right)^2+4\left(q_4q_3+q_1q_2\right)^2}$$

Appendix B

Aircraft Details

This appendix describes the aircraft data used in the non-linear model developed in Chapter 2 and is based on a CAP232 0.90 size aerobatic aircraft fitted with a GMS 1.20 cubic inch methanol engine as was used in the project.

B.1 Physical

B.1.1 Mass

The aircraft's mass with avionics included was determined as,

$$m = 5.0\,\text{kg} \tag{B.1}$$

B.1.2 Moments of Inertia

The inertial properties were determined making use of a double pendulum setup as was utilised by [35] and [36]. The technique involves suspending the aircraft from two thin strings where the strings are parallel to the axis in which the inertia wants to be measured. Figure B.1 demonstrates the setup used to measure the roll moment of inertia.

The aircraft is perturbed by applying a moment about the axis of concern and measuring the period, T, of the oscillation. The inertia about the axis under study is then given by,

$$J = \frac{mgd^2T^2}{4\pi^2 l} \tag{B.2}$$

where m is the aircraft mass, g the gravitational acceleration, d the half distance between the suspension strings and l is the length of each of the strings. It is important to realise that

Figure B.1: Picture of Moment of Inertia Measurement

this technique only yields accurate results when the oscillation perturbations are kept small and when $d \ll l$. Table B.1 summarises the moment of inertia measurements.

Table B.1: Moment of Inertia Measurements

Inertia	**Value**	**Unit**
J_x	0.200	kgm^2
J_y	0.360	kgm^2
J_z	0.525	kgm^2

B.2 Gravitational

The gravitational acceleration per unit mass is given by,

$$g = 9.81\,\text{ms}^{-2} \tag{B.3}$$

B.3 Engine

The engine's dynamic time constant and maximum static thrust was found by experiment to be,

$$\tau_E = 0.5\,\mathrm{s} \tag{B.4}$$

$$T_{Max} = 60\,\mathrm{N} \tag{B.5}$$

B.4 Aerodynamic

B.4.1 Air Density

According to [25], the air density as a function of the altitude above mean sea level is given by,

$$\rho\,(h) = 1.225\left(1 - 2.2558 \times 10^{-5} h\right)^{4.2559} \tag{B.6}$$

where h represents the altitude above mean sea level and remains valid for altitudes up to 11000 meters.

Differentiating equation (B.6) with respect to the altitude and evaluating the differential equation at altitudes close to sea level, it can be shown that the change in air density is negligible and the air density can be approximated as a constant. Equation (B.6) is evaluated at sea level to obtain the air density used throughout this project,

$$\rho = 1.225\,\mathrm{kgm}^{-3} \tag{B.7}$$

B.4.2 Airfoil Data

In order to calculate the wing reference values used in the aerodynamic force and moment equations, the main wing is considered a flat surface with straight leading and trailing edges as well as straight root and tip chords. Keeping this definition in mind, the chord length variation over the wing semi-span is defined as,

$$c(y) = c_r + \frac{2y}{b}\left(c_t - c_r\right) \tag{B.8}$$

where c_r and c_t are the root and tip chord lengths respectively and y is the distance from the root chord. The wing reference area, S, can now be calculated by integrating the chord length variation over the wing span,

$$S = \int_{-s}^{s} c(y)dy \tag{B.9}$$

where s is the wing semi-span and is given by,

$$s = \frac{b}{2} \tag{B.10}$$

The wing mean aerodynamic chord and aspect ratio are calculated using equations (B.11) and (B.12) respectively,

$$c = \frac{1}{S}\int_{-s}^{s} c^2(y)dy \tag{B.11}$$

$$A = \frac{b}{c} \tag{B.12}$$

For more information on these equations see [1]. The wing's aerodynamic data specific to the airframe used in this project is summarised in Table B.2 where the abbreviation ND stands for Non-Dimensional.

Table B.2: Wing Aerodynamic Data

Parameter	Symbol	Value	Unit
Root chord	c_r	0.38	m
Tip chord	c_t	0.2	m
Span	b	1.73	m
Semi-span	s	0.865	m
Area	S	0.5017	m^2
Mean aerodynamic chord	c	0.2993	m
Aspect ratio	A	5.9655	ND
Efficiency factor	e	0.85	ND

B.4.3 Parasitic Drag Coefficient

The aircraft's parasitic drag coefficient is made up from wing and fuselage contributions. The tailplane and fin contributions are neglected due to their relatively small reference areas compared to the wing and fuselage. Subsequently the parasitic drag coefficient, C_{D_0}, can now be written as,

$$C_{D_0} = C_{D_{0_w}} + C_{D_{0_f}} \tag{B.13}$$

where the subscripts w and f denote the wing and fuselage contributions respectively. [2] provides a formula for calculating an airfoil's parasitic drag coefficient,

$$C_{D_{0_i}} = \frac{KC_f S_{wet}}{S} \tag{B.14}$$

where K and C_f are the correction factor for pressure drag and skin friction coefficient respectively and both are calculated using empirical formulas. S_{wet} is the airfoil wetted area which is the entire area in contact with the airstream. S is the wing reference area as previously defined.

Table B.3 summarises the data used in equation (B.14) along with the calculated wing and fuselage parasitic drag coefficients.

Table B.3: Parasitic Drag Summary

Airfoil	K [ND]	C_f [ND]	S_{wet} [m^2]	S [m^2]	**Coefficient**
Wing	1.253	4.90×10^{-3}	1.024	0.5017	12.5×10^{-3}
Fuselage	1.200	3.77×10^{-3}	0.676	0.5017	6.1×10^{-3}

B.4.4 Zero Angle of Attack Lift and Pitching Moment Coefficients

Due to the wing having symmetrical camber, the zero angle of attack lift and pitching moment coefficients were set to zero,

$$C_{L_0} = 0 \tag{B.15}$$

$$C_{m_0} = 0 \tag{B.16}$$

B.4.5 Trim Data

During a trim flight condition, the force and moment contributions that drive the equations of motion sum to zero. This scenario can be exploited to calculate the non-zero aircraft states which are required for trim level flight and is stated below,

$$X = X_A + X_E = 0 \tag{B.17}$$

$$Z = Z_A + Z_G = 0 \tag{B.18}$$

$$M = M_A = 0 \tag{B.19}$$

Setting the trim velocity at,

$$V_0 = 30\,\mathrm{ms}^{-1} \tag{B.20}$$

and substituting the various components into equations (B.17) to (B.19), the trim states were

calculated as,

$$T_0 = 6.152\,\mathrm{N} \tag{B.21}$$

$$\alpha_0 = 2.05^\circ \tag{B.22}$$

$$\delta_{e_0} = -1.5^\circ \tag{B.23}$$

B.4.6 Stability and Control Derivatives Calculation using AVL

The following sections describe how AVL can be used to calculate the stability and control derivatives of a conventional airframe.

Limitations

AVL makes use of computational fluid dynamics methods to characterise a given airframe's aerodynamic properties. When dealing with fluid dynamics methods it is important to know the limitations of the process and to adhere to these in order to obtain accurate results. The following aspects need to be considered when performing a flow analysis with AVL.

Airframe Configuration Aerodynamic configurations with small incidence angles and which consist of thin lifting surfaces and slender bodies are best suited for analysis in AVL. The modelling of slender bodies has only recently been added to the solution and subsequently should be used with caution. If it is estimated that the fuselage has limited effect on the aerodynamic model, it is best left un-modelled.

Compressibility To quantify the effects of compressibility in AVL, the Prandtl and Glauert factor needs to be calculated and is defined as [2],

$$B = \frac{1}{\sqrt{1 - M_0^2}} \tag{B.24}$$

where M_0 is the freestream Mach number. According to [18], the flow analysis should be considered suspicious when the Prandtl and Glauert factor starts to exceed 1.4, which implies a freestream Mach number of 0.7. However, for the scope of this project where freestream velocities in the vicinity of 0.1 Mach are considered, the effects of compressibility can be neglected.

Unsteady Flow Unsteady flow is avoided by analysing the flow angles, also known as helix angles. The helix angles can be seen as non-dimensional angular rates and are calculated by scaling the angular rates with factors consisting of the airspeed, wingspan and mean aerodynamic chord. These relationships are stated in Table B.4 where p, q and r are the

aircraft's angular rates and are measured in radians per second and b and c are the wing's span and mean aerodynamic chord respectively.

Table B.4: Helix Angles

Angular rate	**Helix angle**
Roll rate	$\frac{pb}{2V}$
Pitch rate	$\frac{qc}{2V}$
Yaw rate	$\frac{rb}{2V}$

According to [18], the helix angles should be kept within the following bounds to ensure a steady flow solution,

$$-0.10 < \frac{pb}{2V} < 0.10 \tag{B.25}$$

$$-0.03 < \frac{qc}{2V} < 0.03 \tag{B.26}$$

$$-0.25 < \frac{rb}{2V} < 0.25 \tag{B.27}$$

The limits stated in equations (B.25) to (B.27) are easily achievable except when extremely aggressive aerobatic manoeuvres are performed at relatively low airspeeds.

Geometry Axis System

The axis system used to define the aircraft's geometric properties is defined as follows: the X-axis points downstream. The Y-axis is perpendicular to the X-axis and points out the starboard wing. The Z-axis is perpendicular to the XY-plane to form a right handed Cartesian system.

Input Files

An AVL model consists of three plain text files which can be created using any standard text editor. The files have the same prefix, but are identified by different extensions. A brief discussion of the individual files are given below.

Geometry configuration has a `.avl` extension and is mandatory since it describes the aerodynamic configuration to be modelled. The file starts off with header data which consists of:

- The model title.
- Freestream Mach number used to calculate the Prandtl and Glauert correction factor.
- Symmetry options used to simplify the flow analysis. This option is only used when limited processing power is available.
- Airfoil reference values used to calculate the non-dimensional derivatives which include the wing's span, mean aerodynamic chord and reference area.
- Centre of gravity location for trim calculations.

The rest of the geometry file consists of a series of keywords describing the airfoil and body components which make up the airframe.

Mass and inertia configuration has a `.mass` extension and is optional since the information contained in this file is not required to complete only a flow analysis. The file becomes necessary when AVL is used to generate linear state-space models for the airframe where the mass and inertial properties of the configuration are required.

Run case descriptions has a `.run` extension. This file is usually created from within AVL itself since it is much more convenient to set up the various run case parameters in AVL and then simply using an option to save the run cases to a file.

For an in depth discussion on how each of the input files are created, see [18].

CAP232 Geometry Configuration

Included below is the geometry configuration file specific to the CAP232 0.90 size aerobatic aircraft as was used in the project. All dimensions are in meters.

```
Plane Laura
#Modelling of plane Laura
#Willem Hough
#Stellenbosch University
#2005/09/21
#Mach
 0.1
#IYsym  IZsym  Zsym                No symmetry assumed
 0      0      0.0
#Sref    Cref              Bref Sref is the area of the main wing used to define all
#aerodynamic coefficients. Cref is the MAC of the main wing. Bref is the span of the main wing.
 0.5017  0.29931034482759  1.73
#Xref  Yref  Zref              This is the current location of the CG.
 0.02  0.0   0.05
#==============================================================================
#This defines the entire wing surface. Note that the wing is mirrored about the X-Z plane
```

```
#and that the area is divided into two sections. One section does not include aileron or
#flapperon and the other section does.
#==============================================================================================
SURFACE
wing
#----------------------------------------------------------------------------------------------
#This defines 20 chord wise and and 20 span wise horseshoe vortices both with a cosine
#distribution. Note that the spacing of the vortices are visible when a geometric plot is
#done when running AVL.
#----------------------------------------------------------------------------------------------
#Nchordwise  Cspace   Nspanwise   Sspace
 20          1.0      20          1.0
#Duplicate the wing around the X-Z plane (Y = 0.0)
YDUPLICATE
 0.0
# Zero incidence angle on the wing.
ANGLE
 0.0
#----------------------------------------------------------------------------------------------
#This defines the root chord of the section without aileron and flapperon.
#----------------------------------------------------------------------------------------------
SECTION
#Xle     Yle  Zle  Chord  Ainc  Nspanwise  Sspace
 -0.117  0.0  0.0  0.38   0.0   0          0
#----------------------------------------------------------------------------------------------
#This defines the tip chord of the section without aileron and flapperon but is also the
#root chord of the section with aileron and flapperon.
#----------------------------------------------------------------------------------------------
SECTION
#Xle          Yle       Zle            Chord        Ainc  Nspanwise  Sspace
 -0.10545718  0.240308  0.00755198314  0.329993711  0.0   0          0
#Cname   Cgain                Xhinge        HingeVec     SgnDup
CONTROL
 flap    57.29577951308232    0.8181783531  0.0 0.0 0.0  1.0
#Cname   Cgain                Xhinge        HingeVec     SgnDup
CONTROL
 aileron 57.29577951308232    0.8181783531  0.0 0.0 0.0  -1.0
#----------------------------------------------------------------------------------------------
#This defines the tip chord of the section with aileron and flapperon.
#----------------------------------------------------------------------------------------------
SECTION
#Xle            Yle    Zle           Chord  Ainc  Nspanwise  Sspace
 -0.075451071   0.865  0.02718372013 0.2    0.0   0          0
#Cname   Cgain                Xhinge    HingeVec      SgnDup
CONTROL
 flap    57.29577951308232    0.7       0.0 0.0 0.0   1.0
#Cname   Cgain                Xhinge    HingeVec      SgnDup
CONTROL
 aileron 57.29577951308232    0.7       0.0 0.0 0.0   -1.0
#==============================================================================================
#This defines the entire tailplane surface. Note that the tailplane is mirrored about the
#X-Z plane.
#==============================================================================================
SURFACE
```

```
tailplane
#--------------------------------------------------------------------------------------------
#This defines 20 chordwise and and 20 spanwise horseshoe vortices both with a cosine
#distribution. Note that the spacing of the vortices is visible when a geometric plot is
#done when running AVL.
#--------------------------------------------------------------------------------------------
#Nchordwise  Cspace   Nspanwise   Sspace
 20          1.0      20          1.0
#Duplicate the tailplane around the X-Z plane around Y = 0
YDUPLICATE
0.0
# Zero incidence angle on the tailplane.
ANGLE
 0.0
#--------------------------------------------------------------------------------------------
#This defines the root chord of the tailplane.
#--------------------------------------------------------------------------------------------
SECTION
#Xle    Yle   Zle    Chord  Ainc  Nspanwise  Sspace
 0.66   0.0   0.09   0.193  0.0   0               0
#Cname     Cgain                 Xhinge        HingeVec     SgnDup
CONTROL
 elevator  57.29577951308232     0.5699481865  0.0 0.0 0.0  1.0
#--------------------------------------------------------------------------------------------
#This defines the tip chord of the tailplane.
#--------------------------------------------------------------------------------------------
SECTION
#Xle           Yle    Zle   Chord  Ainc  Nspanwise  Sspace
 0.6948749029  0.335  0.09  0.158  0.0   0          0
#Cname     Cgain                 Xhinge        HingeVec     SgnDup
CONTROL
 elevator  57.29577951308232     0.4746835443  0.0 0.0 0.0  1.0
#
#==============================================================
#This defines the entire fin surface.
#==============================================================
SURFACE
fin
#--------------------------------------------------------------------------------------------
#This defines 20 chordwise and and 20 spanwise horseshoe vortices both with a cosine
#distribution. Note that the spacing of the vortices is visible when a geometric plot is
#done when running AVL.
#--------------------------------------------------------------------------------------------
#Nchordwise  Cspace  Nspanwise  Sspace
 20          1.0     20         1.0
#--------------------------------------------------------------------------------------------
#This defines the root chord of the fin.
#--------------------------------------------------------------------------------------------
SECTION
#Xle    Yle   Zle  Chord   Ainc  Nspanwise  Sspace
 0.713  0.0   0.0  0.2825  0.0   0          0
#Cname    Cgain                 Xhinge        HingeVec     SgnDup
CONTROL
 rudder   -57.29577951308232    0.5663716814  0.0 0.0 0.0  1.0
```

```
#------------------------------------------------------------------------------------------
#This defines the tip chord of the fin.
#------------------------------------------------------------------------------------------
SECTION
#Xle    Yle   Zle    Chord   Ainc  Nspanwise  Sspace
 0.828  0.0   0.31   0.1675  0.0   0          0
#Cname    Cgain                 Xhinge        HingeVec    SgnDup
CONTROL
 rudder   -57.29577951308232    0.2686567164  0.0 0.0 0.0  1.0
#=============================================================
#This defines the entire fuselage.
#=============================================================
BODY
fuselage
#------------------------------------------------------------------------------------------
#This defines 20 source-line nodes with a cosine distribution. Note that the spacing of
#the nodes are visible when a geometric plot is done when running AVL.
#------------------------------------------------------------------------------------------
#Nbody  Bspace
 20     1.0
#Translates the fuselage to the correct coordinates.
TRANSLATE
-0.337  0.0  0.08
#Loads the airfoil information from a file
BFILE
fusePL.dat
```

Geometry Plot

Because the geometry is specified via a text file it is paramount to be able to view a geometric plot of the specified airframe. Also, before any flow analysis is attempted, it is beneficial to inspect the geometric plot and to verify that the model represents the actual vehicle. Figure B.2 shows an AVL geometry plot of the airframe as was specified in the previous section.

Derivatives

The aircraft's non-dimensional stability and control derivatives, valid for the trim condition stated in equations (B.20) to (B.23), are stated in Tables B.5 and B.6.

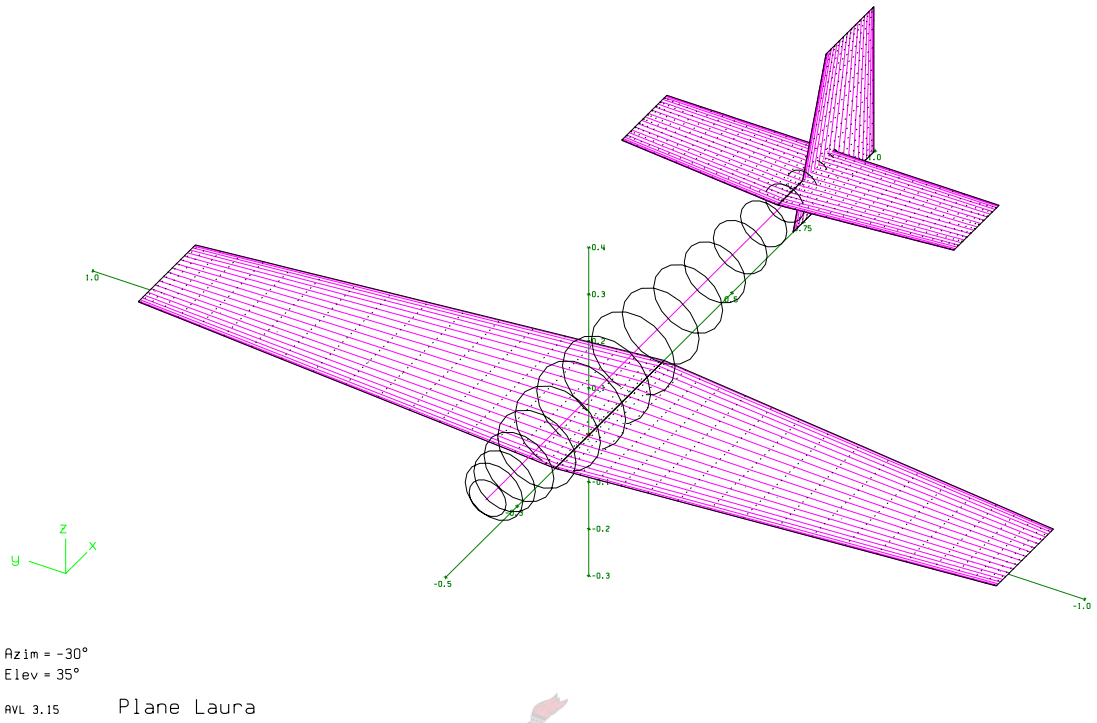


Figure B.2: AVL Airframe Geometry Plot

Table B.5: AVL Stability Derivatives

Derivative	Value	Derivative	Value	Derivative	Value
C_{L_α}	5.1309	C_{y_p}	0.0102	C_{y_r}	0.212231
C_{m_α}	−0.2954	C_{l_p}	−0.4248	C_{l_r}	0.045011
C_{y_β}	−0.2777	C_{n_p}	−0.0251	C_{n_r}	−0.124994
C_{l_β}	−0.0331	C_{L_q}	7.7330		
C_{n_β}	0.0860	C_{m_q}	−10.2807		

Table B.6: AVL Control Derivatives

Derivative	Value	Derivative	Value
$C_{y_{\delta_a}}$	−0.0077	$C_{y_{\delta_r}}$	0.2303
$C_{l_{\delta_a}}$	−0.3731	$C_{l_{\delta_r}}$	0.0080
$C_{n_{\delta_a}}$	−0.0065	$C_{n_{\delta_r}}$	−0.1129
$C_{m_{\delta_e}}$	−1.5852		

Appendix C

Estimator and Controller Details

This appendix contains various details specific to the kinematic state estimator and model predictive controller discussed in Chapters 3 and 4 respectively.

C.1 Sampling Time

The sampling time used to discretise the various dynamic equations, as well as to propagate the non-linear kinematic equations forward in time is given by,

$$T_s = 0.02\,\mathrm{s} \tag{C.1}$$

C.2 Gravitational Compensation

The gravitational acceleration not measured by the accelerometers is compensated for by the term,

$$g'_D = g = 9.81\,\mathrm{ms}^{-2} \tag{C.2}$$

C.3 GPS Delay

The pole/zero location of the first order Padé approximation used to model the 310 ms GPS delay is given by,

$$\omega_p = 6.45 \tag{C.3}$$

C.4 Earth Radius, Magnetic Field Components and TRIAD Reference Vectors

The constant radius used in the round earth approximation is given by [23],

$$R = 6\,378\,137\,\text{m} \tag{C.4}$$

The earth's magnetic field is modelled by the International Geomagnetic Reference Field (IGRF) model. [47] provides a software tool which, given the date and location of interest, estimates the magnetic field according to the IGRF model. The constant magnetic field for Stellenbosch was calculated as,

$$B_N = 0.09656\,\text{Gauss} \tag{C.5}$$

$$B_E = -0.043841\,\text{Gauss} \tag{C.6}$$

$$B_D = -0.237397\,\text{Gauss} \tag{C.7}$$

The reference vectors used in the implementation of the TRIAD algorithm is given below,

$$\boldsymbol{w_I^1} = \begin{bmatrix} 0 & 0 & -g \end{bmatrix}^T \text{ms}^{-2} \tag{C.8}$$

$$\boldsymbol{w_I^2} = \begin{bmatrix} B_N & B_E & B_D \end{bmatrix}^T \text{Gauss} \tag{C.9}$$

where $\boldsymbol{w_I^1}$ and $\boldsymbol{w_I^2}$ are the reference vectors associated with the triad of accelerometers and magnetometer respectively.

C.5 EKF Linearised Kinematics and Measurements

Stated below are the respective Jacobian matrices used to linearise the kinematic and measurement equations. Only the non-zero matrix entries are given.

C.5.1 Linear State Equations

$$\frac{\partial \boldsymbol{f}\left(\boldsymbol{x},\boldsymbol{u}\right)}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial \lambda} & \frac{\partial f_1}{\partial \varphi} & \cdots & \frac{\partial f_1}{\partial V_{D_d}} \\ \frac{\partial f_2}{\partial \lambda} & \frac{\partial f_2}{\partial \varphi} & \cdots & \frac{\partial f_2}{\partial V_{D_d}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{16}}{\partial \lambda} & \frac{\partial f_{16}}{\partial \varphi} & \cdots & \frac{\partial f_{16}}{\partial V_{D_d}} \end{bmatrix} \tag{C.10}$$

where

$$\frac{\partial f_1}{\partial h} = -\frac{V_N}{(R+h)^2}, \quad \frac{\partial f_1}{\partial V_N} = \frac{1}{R+h}, \quad \frac{\partial f_2}{\partial \lambda} = \frac{V_E \tan\lambda}{(R+h)\cos\lambda}, \quad \frac{\partial f_2}{\partial h} = -\frac{V_E}{(R+h)^2\cos\lambda},$$

$$\frac{\partial f_2}{\partial V_E} = \frac{1}{(R+h)\cos\lambda}, \quad \frac{\partial f_3}{\partial V_D} = -1, \quad \frac{\partial f_4}{\partial q_1} = 2q_1a_X + 2q_2a_Y + 2q_3a_Z,$$

$$\frac{\partial f_4}{\partial q_2} = -2q_2a_X + 2q_1a_Y + 2q_4a_Z, \quad \frac{\partial f_4}{\partial q_3} = -2q_3a_X - 2q_4a_Y + 2q_1a_Z,$$

$$\frac{\partial f_4}{\partial q_4} = 2q_4a_X - 2q_3a_Y + 2q_2a_Z, \quad \frac{\partial f_5}{\partial q_1} = 2q_2a_X - 2q_1a_Y - 2q_4a_Z,$$

$$\frac{\partial f_5}{\partial q_2} = 2q_1a_X + 2q_2a_Y + 2q_3a_Z, \quad \frac{\partial f_5}{\partial q_3} = 2q_4a_X - 2q_3a_Y + 2q_2a_Z,$$

$$\frac{\partial f_5}{\partial q_4} = 2q_3a_X + 2q_4a_Y - 2q_1a_Z, \quad \frac{\partial f_6}{\partial q_1} = 2q_3a_X + 2q_4a_Y - 2q_1a_Z,$$

$$\frac{\partial f_6}{\partial q_2} = -2q_4a_X + 2q_3a_Y - 2q_2a_Z, \quad \frac{\partial f_6}{\partial q_3} = 2q_1a_X + 2q_2a_Y + 2q_3a_Z,$$

$$\frac{\partial f_6}{\partial q_4} = -2q_2a_X + 2q_1a_Y + 2q_4a_Z, \quad \frac{\partial f_7}{\partial q_2} = \frac{1}{2}r, \quad \frac{\partial f_7}{\partial q_3} = -\frac{1}{2}q, \quad \frac{\partial f_7}{\partial q_4} = \frac{1}{2}p,$$

$$\frac{\partial f_8}{\partial q_1} = -\frac{1}{2}r, \quad \frac{\partial f_8}{\partial q_3} = \frac{1}{2}p, \quad \frac{\partial f_8}{\partial q_4} = \frac{1}{2}q, \quad \frac{\partial f_9}{\partial q_1} = \frac{1}{2}q, \quad \frac{\partial f_9}{\partial q_2} = -\frac{1}{2}p, \quad \frac{\partial f_9}{\partial q_4} = \frac{1}{2}r,$$

$$\frac{\partial f_{10}}{\partial q_1} = -\frac{1}{2}p, \quad \frac{\partial f_{10}}{\partial q_2} = -\frac{1}{2}q, \quad \frac{\partial f_{10}}{\partial q_3} = -\frac{1}{2}r, \quad \frac{\partial f_{11}}{\partial \lambda} = \omega_p, \quad \frac{\partial f_{11}}{\partial h} = \frac{V_N}{(R+h)^2},$$

$$\frac{\partial f_{11}}{\partial V_N} = -\frac{1}{R+h}, \quad \frac{\partial f_{11}}{\partial \lambda_d} = -\omega_p, \quad \frac{\partial f_{12}}{\partial \lambda} = -\frac{V_E \tan\lambda}{(R+h)\cos\lambda}, \quad \frac{\partial f_{12}}{\partial \varphi} = \omega_p,$$

$$\frac{\partial f_{12}}{\partial h} = \frac{V_E}{(R+h)^2\cos\lambda}, \quad \frac{\partial f_{12}}{\partial V_E} = -\frac{1}{(R+h)\cos\lambda}, \quad \frac{\partial f_{12}}{\partial \varphi_d} = -\omega_p, \quad \frac{\partial f_{13}}{\partial h} = \omega_p,$$

$$\frac{\partial f_{13}}{\partial V_D} = 1, \quad \frac{\partial f_{13}}{\partial h_d} = -\omega_p, \quad \frac{\partial f_{14}}{\partial V_N} = \omega_p, \quad \frac{\partial f_{14}}{\partial q_1} = -2q_1a_X - 2q_2a_Y - 2q_3a_Z,$$

$$\frac{\partial f_{14}}{\partial q_2} = 2q_2a_X - 2q_1a_Y - 2q_4a_Z, \quad \frac{\partial f_{14}}{\partial q_3} = 2q_3a_X + 2q_4a_Y - 2q_1a_Z,$$

$$\frac{\partial f_{14}}{\partial q_4} = -2q_4a_X + 2q_3a_Y - 2q_2a_Z, \quad \frac{\partial f_{14}}{\partial V_{N_d}} = -\omega_p, \quad \frac{\partial f_{15}}{\partial V_E} = \omega_p,$$

$$\frac{\partial f_{15}}{\partial q_1} = -2q_2a_X + 2q_1a_Y + 2q_4a_Z, \quad \frac{\partial f_{15}}{\partial q_2} = -2q_1a_X - 2q_2a_Y - 2q_3a_Z,$$

$$\frac{\partial f_{15}}{\partial q_3} = -2q_4a_X + 2q_3a_Y - 2q_2a_Z, \quad \frac{\partial f_{15}}{\partial q_4} = -2q_3a_X - 2q_4a_Y + 2q_1a_Z,$$

$$\frac{\partial f_{15}}{\partial V_{E_d}} = -\omega_p, \quad \frac{\partial f_{16}}{\partial V_D} = \omega_p, \quad \frac{\partial f_{16}}{\partial q_1} = -2q_3a_X - 2q_4a_Y + 2q_1a_Z,$$

$$\frac{\partial f_{16}}{\partial q_2} = 2q_4a_X - 2q_3a_Y + 2q_2a_Z, \quad \frac{\partial f_{16}}{\partial q_3} = -2q_1a_X - 2q_2a_Y - 2q_3a_Z,$$

$$\frac{\partial f_{16}}{\partial q_4} = 2q_2a_X - 2q_1a_Y - 2q_4a_Z, \quad \frac{\partial f_{16}}{\partial V_{D_d}} = -\omega_p$$

$$\frac{\partial \boldsymbol{f}\left(\boldsymbol{x},\boldsymbol{u}\right)}{\partial \boldsymbol{u}} = \begin{bmatrix} \frac{\partial f_1}{\partial a_X} & \frac{\partial f_1}{\partial a_Y} & \cdots & \frac{\partial f_1}{\partial r} \\ \frac{\partial f_2}{\partial a_X} & \frac{\partial f_2}{\partial a_Y} & \cdots & \frac{\partial f_2}{\partial r} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{16}}{\partial a_X} & \frac{\partial f_{16}}{\partial a_Y} & \cdots & \frac{\partial f_{16}}{\partial r} \end{bmatrix} \tag{C.11}$$

where

$$\frac{\partial f_4}{\partial a_X} = q_4^2 + q_1^2 - q_2^2 - q_3^2, \quad \frac{\partial f_4}{\partial a_Y} = 2\left(q_1 q_2 - q_4 q_3\right), \quad \frac{\partial f_4}{\partial a_Z} = 2\left(q_4 q_2 + q_1 q_3\right),$$

$$\frac{\partial f_5}{\partial a_X} = 2\left(q_4 q_3 + q_1 q_2\right), \quad \frac{\partial f_5}{\partial a_Y} = q_4^2 - q_1^2 + q_2^2 - q_3^2, \quad \frac{\partial f_5}{\partial a_Z} = 2\left(q_2 q_3 - q_4 q_1\right),$$

$$\frac{\partial f_6}{\partial a_X} = 2\left(q_1 q_3 - q_4 q_2\right), \quad \frac{\partial f_6}{\partial a_Y} = 2\left(q_4 q_1 + q_2 q_3\right), \quad \frac{\partial f_6}{\partial a_Z} = q_4^2 - q_1^2 - q_2^2 + q_3^2,$$

$$\frac{\partial f_7}{\partial p} = \frac{1}{2}q_4, \quad \frac{\partial f_7}{\partial q} = -\frac{1}{2}q_3, \quad \frac{\partial f_7}{\partial r} = \frac{1}{2}q_2, \quad \frac{\partial f_8}{\partial p} = \frac{1}{2}q_3, \quad \frac{\partial f_8}{\partial q} = \frac{1}{2}q_4,$$

$$\frac{\partial f_8}{\partial r} = -\frac{1}{2}q_1, \quad \frac{\partial f_9}{\partial p} = -\frac{1}{2}q_2, \quad \frac{\partial f_9}{\partial q} = \frac{1}{2}q_1, \quad \frac{\partial f_9}{\partial r} = \frac{1}{2}q_4, \quad \frac{\partial f_{10}}{\partial p} = -\frac{1}{2}q_1,$$

$$\frac{\partial f_{10}}{\partial q} = -\frac{1}{2}q_2, \quad \frac{\partial f_{10}}{\partial r} = -\frac{1}{2}q_3, \quad \frac{\partial f_{14}}{\partial a_X} = -q_4^2 - q_1^2 + q_2^2 + q_3^2,$$

$$\frac{\partial f_{14}}{\partial a_Y} = -2\left(q_1 q_2 - q_4 q_3\right), \quad \frac{\partial f_{14}}{\partial a_Z} = 2\left(q_4 q_2 + q_1 q_3\right), \quad \frac{\partial f_{15}}{\partial a_X} = -2\left(q_4 q_3 + q_1 q_2\right),$$

$$\frac{\partial f_{15}}{\partial a_Y} = -q_4^2 + q_1^2 - q_2^2 + q_3^2, \quad \frac{\partial f_{15}}{\partial a_Z} = 2\left(q_2 q_3 - q_4 q_1\right), \quad \frac{\partial f_{16}}{\partial a_X} = -2\left(q_1 q_3 - q_4 q_2\right),$$

$$\frac{\partial f_{16}}{\partial a_Y} = 2\left(q_4 q_1 + q_2 q_3\right), \quad \frac{\partial f_{16}}{\partial a_Z} = q_4^2 - q_1^2 - q_2^2 + q_3^2$$

C.5.2 Linear Measurement Equations

$$\frac{\partial \boldsymbol{h}\left(\boldsymbol{x}\right)}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial h_1}{\partial \lambda} & \frac{\partial h_1}{\partial \varphi} & \cdots & \frac{\partial h_1}{\partial V_{D_d}} \\ \frac{\partial h_2}{\partial \lambda} & \frac{\partial h_2}{\partial \varphi} & \cdots & \frac{\partial h_2}{\partial V_{D_d}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_9}{\partial \lambda} & \frac{\partial h_9}{\partial \varphi} & \cdots & \frac{\partial h_9}{\partial V_{D_d}} \end{bmatrix} \tag{C.12}$$

where

$$\frac{\partial h_1}{\partial \lambda_d} = 1, \quad \frac{\partial h_2}{\partial \varphi_d} = 1, \quad \frac{\partial h_3}{\partial h_d} = 1, \quad \frac{\partial h_4}{\partial V_{N_d}} = 1, \quad \frac{\partial h_5}{\partial V_{E_d}} = 1, \quad \frac{\partial h_6}{\partial V_{D_d}} = 1,$$

$$\frac{\partial h_7}{\partial q_1} = 2q_1B_N + 2q_2B_E + 2q_3B_D, \quad \frac{\partial h_7}{\partial q_2} = -2q_2B_N + 2q_1B_E - 2q_4B_D,$$
$$\frac{\partial h_7}{\partial q_3} = -2q_3B_N + 2q_4B_E + 2q_1B_D, \quad \frac{\partial h_7}{\partial q_4} = 2q_4B_N + 2q_3B_E - 2q_2B_D,$$
$$\frac{\partial h_8}{\partial q_1} = 2q_2B_N - 2q_1B_E + 2q_4B_D, \quad \frac{\partial h_8}{\partial q_2} = 2q_1B_N + 2q_2B_E + 2q_3B_D,$$
$$\frac{\partial h_8}{\partial q_3} = -2q_4B_N - 2q_3B_E + 2q_2B_D, \quad \frac{\partial h_8}{\partial q_4} = -2q_3B_N + 2q_4B_E + 2q_1B_D,$$
$$\frac{\partial h_9}{\partial q_1} = 2q_3B_N - 2q_4B_E - 2q_1B_D, \quad \frac{\partial h_9}{\partial q_2} = 2q_4B_N + 2q_3B_E - 2q_2B_D,$$
$$\frac{\partial h_9}{\partial q_3} = 2q_1B_N + 2q_2B_E + 2q_3B_D, \quad \frac{\partial h_9}{\partial q_4} = 2q_2B_N - 2q_1B_E + 2q_4B_D$$

C.6 MPC Linearised Dynamics

Stated below are the Jacobian matrices used to linearise the aircraft perturbation dynamics. Only the non-zero matrix entries are given.

$$\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial V} & \frac{\partial f_1}{\partial \alpha} & \cdots & \frac{\partial f_1}{\partial T} \\ \frac{\partial f_2}{\partial V} & \frac{\partial f_2}{\partial \alpha} & \cdots & \frac{\partial f_2}{\partial T} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{14}}{\partial V} & \frac{\partial f_{14}}{\partial \alpha} & \cdots & \frac{\partial f_{14}}{\partial T} \end{bmatrix} \tag{C.13}$$

where

$$\frac{\partial f_1}{\partial V} = \frac{\rho SbC_{y_p}}{4m}(\alpha\beta r + \beta p) - \alpha^2\beta r + \frac{\rho SC_{y_\beta}}{m}V\beta^2 - \left(\frac{\rho SC_{D_0}}{m} + \frac{\rho SC_{L_0}^2}{m\pi Ae}\right)V$$
$$-\frac{2\rho SC_{L_0}C_{L_\alpha}}{m\pi Ae}V\alpha - \frac{\rho SC_{L_\alpha}^2}{m\pi Ae}V\alpha^2, \quad \frac{\partial f_1}{\partial \alpha} = \frac{\rho SbC_{y_p}}{4m}V\beta r - 2V\alpha\beta r - \frac{\rho SC_{L_0}C_{L_\alpha}}{m\pi Ae}V^2$$
$$-\frac{\rho SC_{L_\alpha}^2}{m\pi Ae}V^2\alpha + g\left(q_4^2 - q_1^2 - q_2^2 + q_3^2\right), \quad \frac{\partial f_1}{\partial \beta} = \frac{\rho SbC_{y_p}}{4m}(V\alpha r + Vp) - V\alpha^2 r$$
$$+\frac{\rho SC_{y_\beta}}{m}V^2\beta + 2g\left(q_4q_1 + q_2q_3\right), \quad \frac{\partial f_1}{\partial p} = \frac{\rho SbC_{y_p}}{4m}V\beta, \quad \frac{\partial f_1}{\partial r} = \frac{\rho SbC_{y_p}}{4m}V\alpha\beta$$
$$-V\alpha^2\beta, \quad \frac{\partial f_1}{\partial q_1} = 2gq_3 - 2g\alpha q_1 + 2g\beta q_4, \quad \frac{\partial f_1}{\partial q_2} = -2gq_4 - 2g\alpha q_2 + 2g\beta q_3,$$
$$\frac{\partial f_1}{\partial q_3} = 2gq_1 + 2g\alpha q_3 + 2g\beta q_2, \quad \frac{\partial f_1}{\partial q_4} = -2gq_2 + 2g\alpha q_4 + 2g\beta q_1, \quad \frac{\partial f_1}{\partial T} = \frac{1}{m},$$
$$\frac{\partial f_2}{\partial V} = -\frac{\rho SC_{L_0}}{2m} - \frac{\rho SC_{L_\alpha}}{2m}\alpha - gV^{-2}\left(q_4^2 - q_1^2 - q_2^2 + q_3^2\right) + 2gV^{-2}\alpha\left(q_1q_3 - q_4q_2\right) + \frac{1}{m}V^{-2}\alpha T,$$
$$\frac{\partial f_2}{\partial \alpha} = -\frac{\rho SC_{L_\alpha}}{2m}V - \beta r - 2gV^{-1}\left(q_1q_3 - q_4q_2\right) - \frac{1}{m}V^{-1}T, \quad \frac{\partial f_2}{\partial \beta} = -\alpha r - p, \quad \frac{\partial f_2}{\partial p} = -\beta,$$

$$\frac{\partial f_2}{\partial q} = 1, \quad \frac{\partial f_2}{\partial r} = -\alpha\beta, \quad \frac{\partial f_2}{\partial q_1} = -2gV^{-1}q_1 - 2gV^{-1}\alpha q_3,$$

$$\frac{\partial f_2}{\partial q_2} = -2gV^{-1}q_2 + 2gV^{-1}\alpha q_4, \quad \frac{\partial f_2}{\partial q_3} = 2gV^{-1}q_3 - 2gV^{-1}\alpha q_1, \quad \frac{\partial f_2}{\partial q_4} = 2gV^{-1}q_4$$

$$+2gV^{-1}\alpha q_2, \quad \frac{\partial f_2}{\partial T} = -\frac{1}{m}V^{-1}\alpha, \quad \frac{\partial f_3}{\partial V} = \left(\frac{\rho SC_{y_\beta}}{2m} + \frac{\rho SC_{D_0}}{2m} + \frac{\rho SC_{L_0}^2}{2m\pi Ae}\right)\beta$$

$$+\frac{\rho SC_{L_0}C_{L_\alpha}}{m\pi Ae}\alpha\beta + \frac{\rho SC_{L_\alpha}^2}{2m\pi Ae}\alpha^2\beta - 2gV^{-2}\left(q_4q_1 + q_2q_3\right) + gV^{-2}\alpha\beta\left(q_4^2 - q_1^2 - q_2^2 + q_3^2\right)$$

$$+2gV^{-2}\beta\left(q_1q_3 - q_4q_2\right) + \frac{1}{m}V^{-2}\beta T, \quad \frac{\partial f_3}{\partial \alpha} = \frac{\rho SbC_{y_p}}{4m}r + p + \frac{\rho SC_{L_0}C_{L_\alpha}}{m\pi Ae}V\beta + \frac{\rho SC_{L_\alpha}^2}{m\pi Ae}V\alpha\beta$$

$$-gV^{-1}\beta\left(q_4^2 - q_1^2 - q_2^2 + q_3^2\right), \quad \frac{\partial f_3}{\partial \beta} = \left(\frac{\rho SC_{y_\beta}}{2m} + \frac{\rho SC_{D_0}}{2m} + \frac{\rho SC_{L_0}^2}{2m\pi Ae}\right)V$$

$$+\frac{\rho SC_{L_0}C_{L_\alpha}}{m\pi Ae}V\alpha + \frac{\rho SC_{L_\alpha}^2}{2m\pi Ae}V\alpha^2 - gV^{-1}\alpha\left(q_4^2 - q_1^2 - q_2^2 + q_3^2\right) - 2gV^{-1}\left(q_1q_3 - q_4q_2\right) - \frac{1}{m}V^{-1}T,$$

$$\frac{\partial f_3}{\partial p} = \frac{\rho SbC_{y_p}}{4m} + \alpha, \quad \frac{\partial f_3}{\partial r} = \frac{\rho SbC_{y_p}}{4m}\alpha - 1, \quad \frac{\partial f_3}{\partial q_1} = 2gV^{-1}q_4 + 2gV^{-1}\alpha\beta q_1 - 2gV^{-1}\beta q_3,$$

$$\frac{\partial f_3}{\partial q_2} = 2gV^{-1}q_3 + 2gV^{-1}\alpha\beta q_2 + 2gV^{-1}\beta q_4, \quad \frac{\partial f_3}{\partial q_3} = 2gV^{-1}q_2 - 2gV^{-1}\alpha\beta q_3 - 2gV^{-1}\beta q_1,$$

$$\frac{\partial f_3}{\partial q_4} = 2gV^{-1}q_1 - 2gV^{-1}\alpha\beta q_4 + 2gV^{-1}\beta q_2, \quad \frac{\partial f_3}{\partial T} = -\frac{1}{m}V^{-1}\beta, \quad \frac{\partial f_4}{\partial V} = \frac{\rho SbC_{l_\beta}}{J_x}V\beta$$

$$-\frac{\rho SbC_{n_\beta}}{J_x}V\alpha\beta + \frac{\rho Sb^2C_{l_p}}{4J_x}\left(p + \alpha r\right) - \frac{\rho Sb^2C_{n_p}}{4J_x}\left(\alpha p + \alpha^2 r\right) + \frac{\rho Sb^2C_{l_r}}{4J_x}\left(r - \alpha p\right)$$

$$+\frac{\rho Sb^2C_{n_r}}{4J_x}\left(\alpha^2 p - \alpha r\right), \quad \frac{\partial f_4}{\partial \alpha} = -\frac{\rho SbC_{n_\beta}}{2J_x}V^2\beta + \frac{\rho Sb^2C_{l_p}}{4J_x}Vr - \frac{\rho Sb^2C_{n_p}}{4J_x}\left(Vp + 2V\alpha r\right)$$

$$-\frac{\rho Sb^2C_{l_r}}{4J_x}Vp + \frac{\rho Sb^2C_{n_r}}{4J_x}\left(2V\alpha p - Vr\right), \quad \frac{\partial f_4}{\partial \beta} = \frac{\rho SbC_{l_\beta}}{2J_x}V^2 - \frac{\rho SbC_{n_\beta}}{2J_x}V^2\alpha,$$

$$\frac{\partial f_4}{\partial p} = \frac{\rho Sb^2C_{l_p}}{4J_x}V - \left(\frac{\rho Sb^2C_{n_p}}{4J_x} + \frac{\rho Sb^2C_{l_r}}{4J_x}\right)V\alpha + \frac{\rho Sb^2C_{n_r}}{4J_x}V\alpha^2, \quad \frac{\partial f_4}{\partial q} = -\frac{J_z - J_y}{J_x}r,$$

$$\frac{\partial f_4}{\partial r} = -\frac{J_z - J_y}{J_x}q + \left(\frac{\rho Sb^2C_{l_p}}{4J_x} - \frac{\rho Sb^2C_{n_r}}{4J_x}\right)V\alpha - \frac{\rho Sb^2C_{n_p}}{4J_x}V\alpha^2 + \frac{\rho Sb^2C_{l_r}}{4J_x}V,$$

$$\frac{\partial f_5}{\partial V} = \frac{\rho ScC_{m_0}}{J_y}V + \frac{\rho ScC_{m_\alpha}}{J_y}V\alpha + \frac{\rho Sc^2C_{m_q}}{4J_y}q, \quad \frac{\partial f_5}{\partial \alpha} = \frac{\rho ScC_{m_\alpha}}{2J_y}V^2,$$

$$\frac{\partial f_5}{\partial p} = -\frac{J_x - J_z}{J_y}r, \quad \frac{\partial f_5}{\partial q} = \frac{\rho Sc^2C_{m_q}}{4J_y}V, \quad \frac{\partial f_5}{\partial r} = -\frac{J_x - J_z}{J_y}p, \quad \frac{\partial f_6}{\partial V} = \frac{\rho SbC_{n_\beta}}{J_z}V\beta$$

$$+\frac{\rho SbC_{l_\beta}}{J_z}V\alpha\beta + \frac{\rho Sb^2C_{n_p}}{4J_z}\left(p + \alpha r\right) + \frac{\rho Sb^2C_{l_p}}{4J_z}\left(\alpha p + \alpha^2 r\right) + \frac{\rho Sb^2C_{n_r}}{4J_z}\left(r - \alpha p\right)$$

$$+\frac{\rho Sb^2C_{l_r}}{4J_z}\left(\alpha r - \alpha^2 p\right), \quad \frac{\partial f_6}{\partial \alpha} = \frac{\rho SbC_{l_\beta}}{2J_z}V^2\beta + \frac{\rho Sb^2C_{n_p}}{4J_z}Vr + \frac{\rho Sb^2C_{l_p}}{4J_z}\left(Vp + 2V\alpha r\right)$$

$$-\frac{\rho Sb^2C_{n_r}}{4J_z}Vp + \frac{\rho Sb^2C_{l_r}}{4J_z}\left(Vr - 2V\alpha p\right), \quad \frac{\partial f_6}{\partial \beta} = \frac{\rho SbC_{n_\beta}}{2J_z}V^2 + \frac{\rho SbC_{l_\beta}}{2J_z}V^2\alpha,$$

$$\frac{\partial f_6}{\partial p} = -\frac{J_y - J_x}{J_z}q + \frac{\rho S b^2 C_{n_p}}{4J_z}V + \left(\frac{\rho S b^2 C_{l_p}}{4J_z} - \frac{\rho S b^2 C_{n_r}}{4J_z}\right)V\alpha - \frac{\rho S b^2 C_{l_r}}{4J_z}V\alpha^2,$$

$$\frac{\partial f_6}{\partial q} = -\frac{J_y - J_x}{J_z}p, \quad \frac{\partial f_6}{\partial r} = \left(\frac{\rho S b^2 C_{n_p}}{4J_z} + \frac{\rho S b^2 C_{l_r}}{4J_z}\right)V\alpha + \frac{\rho S b^2 C_{l_p}}{4J_z}V\alpha^2 + \frac{\rho S b^2 C_{n_r}}{4J_z}V,$$

$$\frac{\partial f_7}{\partial p} = \frac{1}{2}q_4, \quad \frac{\partial f_7}{\partial q} = -\frac{1}{2}q_3, \quad \frac{\partial f_7}{\partial r} = \frac{1}{2}q_2, \quad \frac{\partial f_7}{\partial q_2} = \frac{1}{2}r, \quad \frac{\partial f_7}{\partial q_3} = -\frac{1}{2}q, \quad \frac{\partial f_7}{\partial q_4} = \frac{1}{2}p,$$

$$\frac{\partial f_8}{\partial p} = \frac{1}{2}q_3, \quad \frac{\partial f_8}{\partial q} = \frac{1}{2}q_4, \quad \frac{\partial f_8}{\partial r} = -\frac{1}{2}q_1, \quad \frac{\partial f_8}{\partial q_1} = -\frac{1}{2}r, \quad \frac{\partial f_8}{\partial q_3} = \frac{1}{2}p, \quad \frac{\partial f_8}{\partial q_4} = \frac{1}{2}q,$$

$$\frac{\partial f_9}{\partial p} = -\frac{1}{2}q_2, \quad \frac{\partial f_9}{\partial q} = \frac{1}{2}q_1, \quad \frac{\partial f_9}{\partial r} = \frac{1}{2}q_4, \quad \frac{\partial f_9}{\partial q_1} = \frac{1}{2}q, \quad \frac{\partial f_9}{\partial q_2} = -\frac{1}{2}p, \quad \frac{\partial f_9}{\partial q_4} = \frac{1}{2}r,$$

$$\frac{\partial f_{10}}{\partial p} = -\frac{1}{2}q_1, \quad \frac{\partial f_{10}}{\partial q} = -\frac{1}{2}q_2, \quad \frac{\partial f_{10}}{\partial r} = -\frac{1}{2}q_3, \quad \frac{\partial f_{10}}{\partial q_1} = -\frac{1}{2}p, \quad \frac{\partial f_{10}}{\partial q_2} = -\frac{1}{2}q,$$

$$\frac{\partial f_{10}}{\partial q_3} = -\frac{1}{2}r, \quad \frac{\partial f_{11}}{\partial V} = q_4^2 + q_1^2 - q_2^2 - q_3^2 + 2\beta\left(q_1q_2 - q_4q_3\right) + 2\alpha\left(q_4q_2 + q_1q_3\right),$$

$$\frac{\partial f_{11}}{\partial \alpha} = 2V\left(q_4q_2 + q_1q_3\right), \quad \frac{\partial f_{11}}{\partial \beta} = 2V\left(q_1q_2 - q_4q_3\right), \quad \frac{\partial f_{11}}{\partial q_1} = 2Vq_1 + 2V\beta q_2 + 2V\alpha q_3,$$

$$\frac{\partial f_{11}}{\partial q_2} = -2Vq_2 + 2V\beta q_1 + 2V\alpha q_4, \quad \frac{\partial f_{11}}{\partial q_3} = -2Vq_3 - 2V\beta q_4 + 2V\alpha q_1,$$

$$\frac{\partial f_{11}}{\partial q_4} = 2Vq_4 - 2V\beta q_3 + 2V\alpha q_2, \quad \frac{\partial f_{12}}{\partial V} = 2\left(q_4q_3 + q_1q_2\right) + \beta\left(q_4^2 - q_1^2 + q_2^2 - q_3^2\right)$$

$$+2\alpha\left(q_2q_3 - q_4q_1\right), \quad \frac{\partial f_{12}}{\partial \alpha} = 2V\left(q_2q_3 - q_4q_1\right), \quad \frac{\partial f_{12}}{\partial \beta} = V\left(q_4^2 - q_1^2 + q_2^2 - q_3^2\right),$$

$$\frac{\partial f_{12}}{\partial q_1} = 2Vq_2 - 2V\beta q_1 - 2V\alpha q_4, \quad \frac{\partial f_{12}}{\partial q_2} = 2Vq_1 + 2V\beta q_2 + 2V\alpha q_3,$$

$$\frac{\partial f_{12}}{\partial q_3} = 2Vq_4 - 2V\beta q_3 + 2V\alpha q_2, \quad \frac{\partial f_{12}}{\partial q_4} = 2Vq_3 + 2V\beta q_4 - 2V\alpha q_1,$$

$$\frac{\partial f_{13}}{\partial V} = -2\left(q_1q_3 - q_4q_2\right) - 2\beta\left(q_4q_1 + q_2q_3\right) - \alpha\left(q_4^2 - q_1^2 - q_2^2 + q_3^2\right),$$

$$\frac{\partial f_{13}}{\partial \alpha} = -V\left(q_4^2 - q_1^2 - q_2^2 + q_3^2\right), \quad \frac{\partial f_{13}}{\partial \beta} = -2V\left(q_4q_1 + q_2q_3\right),$$

$$\frac{\partial f_{13}}{\partial q_1} = -2Vq_3 - 2V\beta q_4 + 2V\alpha q_1, \quad \frac{\partial f_{13}}{\partial q_2} = 2Vq_4 - 2V\beta q_3 + 2V\alpha q_2,$$

$$\frac{\partial f_{13}}{\partial q_3} = -2Vq_1 - 2V\beta q_2 - 2V\alpha q_3, \quad \frac{\partial f_{13}}{\partial q_4} = 2Vq_2 - 2V\beta q_1 - 2V\alpha q_4,$$

$$\frac{\partial f_{14}}{\partial T} = -\frac{1}{\tau_E}$$

$$\frac{\partial \boldsymbol{g}\left(\boldsymbol{x}\right)\boldsymbol{u}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial g_1}{\partial V} & \frac{\partial g_1}{\partial \alpha} & \cdots & \frac{\partial g_1}{\partial T} \\ \frac{\partial g_2}{\partial V} & \frac{\partial g_2}{\partial \alpha} & \cdots & \frac{\partial g_2}{\partial T} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_{14}}{\partial V} & \frac{\partial g_{14}}{\partial \alpha} & \cdots & \frac{\partial g_{14}}{\partial T} \end{bmatrix} \tag{C.14}$$

where

$$\frac{\partial g_1}{\partial V} = \frac{\rho S C_{y_{\delta_a}}}{m} V\beta\delta_a + \frac{\rho S C_{y_{\delta_r}}}{m} V\beta\delta_r, \quad \frac{\partial g_1}{\partial \beta} = \frac{\rho S C_{y_{\delta_a}}}{2m} V^2\delta_a + \frac{\rho S C_{y_{\delta_r}}}{2m} V^2\delta_r,$$

$$\frac{\partial g_2}{\partial V} = -\frac{\rho S C_{L_{\delta_e}}}{2m}\delta_e, \quad \frac{\partial g_3}{\partial V} = \frac{\rho S C_{y_{\delta_a}}}{2m}\delta_a + \frac{\rho S C_{y_{\delta_r}}}{2m}\delta_r,$$

$$\frac{\partial g_4}{\partial V} = \left(\frac{\rho S b C_{l_{\delta_a}}}{J_x} V - \frac{\rho S b C_{n_{\delta_a}}}{J_x} V\alpha\right)\delta_a + \left(\frac{\rho S b C_{l_{\delta_r}}}{J_x} V - \frac{\rho S b C_{n_{\delta_r}}}{J_x} V\alpha\right)\delta_r,$$

$$\frac{\partial g_4}{\partial \alpha} = -\frac{\rho S b C_{n_{\delta_a}}}{2J_x} V^2\delta_a - \frac{\rho S b C_{n_{\delta_r}}}{2J_x} V^2\delta_r, \quad \frac{\partial g_5}{\partial V} = \frac{\rho S c C_{m_{\delta_e}}}{J_y} V\delta_e,$$

$$\frac{\partial g_6}{\partial V} = \left(\frac{\rho S b C_{n_{\delta_a}}}{J_z} V + \frac{\rho S b C_{l_{\delta_a}}}{J_z} V\alpha\right)\delta_a + \left(\frac{\rho S b C_{n_{\delta_r}}}{J_z} V + \frac{\rho S b C_{l_{\delta_r}}}{J_z} V\alpha\right)\delta_r,$$

$$\frac{\partial g_6}{\partial \alpha} = \frac{\rho S b C_{l_{\delta_a}}}{2J_z} V^2\delta_a + \frac{\rho S b C_{l_{\delta_r}}}{2J_z} V^2\delta_r$$

Appendix D

Hardware and Software Details

This appendix contains various details specific to the hardware and software discussed in Chapters 5 and 6.

D.1 Communication

D.1.1 Serial Communication Protocol

To help ensure data integrity, all serial communications were performed according to the protocol introduced by [35]. The string layout protocol is restated here for convenience and clarity, and is defined as follows,

$$\$AA \text{ data } * \text{ CS}$$

Table D.1 gives an explanation of the various identifiers making up the string. When constructing a message, the following rules must be adhered to,

- If any byte in the binary data string is a start or end of string delimiter, the respective data byte should be enqueued twice as to avoid ambiguity.
- The checksum is calculated by taking the exclusive OR (XOR) of all the bytes between the start and end of string delimiters, before any characters were doubled up as described by the previous rule.
- If the checksum is calculated to be an end of string delimiter character, the checksum is to be replaced with a '+' character as to avoid ambiguity.

Table D.1: Serial Communication Protocol Identifiers

Identifier	Description
$	Start of string delimiter
AA	Two matching letters uniquely identifying the message
data	Binary data string of arbitrary length
*	End of string delimiter
CS	1 byte checksum

D.1.2 Telemetry

The logging of sensor, estimator and controller data is primarily the OBC's task. However, it is paramount to be able to monitor the autopilot's state while in flight. Various telemetry packets are transmitted serially to perform this task. The RF link is capable of transmitting approximately 480 bytes of data at 2 Hz. When the various packets are discussed, the packet size will be expressed as a percentage of the maximum available bandwidth since it gives a good indication of the bandwidth usage of the RF module.

Primary

The primary packet is sent at 2 Hz and contains the following information,

- 4 bytes: Time stamp
- 12 bytes: Gyroscope and accelerometer measurements
- 6 bytes: Magnetometer measurements
- 6 bytes: Altitude, airspeed and climb rate from pressure measurements
- 10 bytes: 5 Autopilot servo commands
- 10 bytes: 5 Safety pilot servo commands
- Total: 48 bytes $\Rightarrow$ 10% of 2 Hz cycle

Secondary

The secondary packet is sent at 0.5 Hz and contains the following information,

- 3 bytes: Gyroscope and magnetometer temperature and voltage measurements
- 1 byte: Main battery voltage measurement
- 1 byte: CAN bus voltage measurement
- 1 byte: Backup battery voltage measurement
- 1 byte: Servo voltage measurement
- 1 byte: OBC voltage measurement
- 1 byte: OBC temperature measurement
- 1 byte: OBC logged data
- 2 bytes: Autopilot status bits
- Total: 12 bytes ⇒ 3% of 2 Hz cycle

GPS

The GPS packet is sent at 2 Hz and contains the following information,

- 4 bytes: Latitude
- 4 bytes: Longitude
- 2 bytes: Altitude
- 2 bytes: Heading
- 6 bytes: NED velocities
- 1 byte: Fix and differential GPS indicator
- Total: 19 bytes ⇒ 4% of 2 Hz cycle

EKF

The EKF packet is sent at 2 Hz and contains the following information,

- 4 bytes: Estimated latitude
- 4 bytes: Estimated longitude

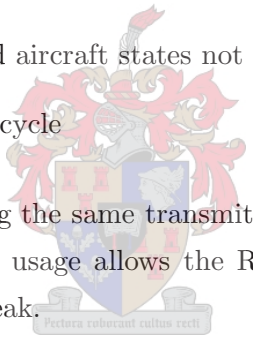
- 2 bytes: Estimated altitude above mean sea level
- 6 bytes: Estimated North/South and East/West displacements and altitude above ground level
- 6 bytes: Estimated NED velocities
- 8 bytes: Estimated attitude
- 24 bytes: State error covariances
- Total: 54 bytes ⇒ 12% of 2 Hz cycle

MPC

The MPC packet is sent at 2 Hz and contains the following information,

- 26 bytes: Trajectory state references
- 14 bytes: Additional estimated aircraft states not sent down with the EKF packet
- Total: 40 bytes ⇒ 9% of 2 Hz cycle

Assuming all packets are sent during the same transmit cycle, only 38% of the available bandwidth is used. The low bandwidth usage allows the RF module leaway to re-send lost data especially when the link becomes weak.



D.1.3 CAN Messages

All messages on the CAN bus are uniquely identified by a 29-bit CAN 2.0B identifier, and can contain up to 8 bytes of data. A CAN protocol is embedded in the identifier by allocating certain bits to represent information about the respective messages. For an in-depth discussion of the protocol, see [38]. Discussed below are the respective CAN messages used in this project and are grouped according to the node which generates them. The message identifiers are stated first in hexadecimal form, followed by a brief explanation of the data contents and destination.

PC104/CAN Controller Node

- `0x010101FF`: Synchronisation packet issued at 50 Hz, received by all nodes.
- `0x02800108`: Autopilot/safety pilot servo timer values for channels 0 to 3, received by servo controller node and HIL distribution board node.

- `0x02810108`: Autopilot/safety pilot servo timer values for channels 4 to 7, received by servo controller node and HIL distribution board node.
- `0x02820108`: Autopilot/safety pilot servo timer values for channels 8 to 11, received by servo controller node and HIL distribution board node.
- `0x02830108`: Autopilot/safety pilot servo timer values for channels 12 to 15, received by servo controller node and HIL distribution board node.
- `0x0B8F0102`: IMU self-test on/off function, received by IMU sensor node.
- `0x0B8F0104`: Magnetometer set/reset function, received by pressure/magnetometer sensor node.

IMU Sensor Node

- `0x0A800201`: Filtered ADC values, received by PC104/CAN controller node.
- `0x0A810201`: Filtered ADC values, received by PC104/CAN controller node.

Pressure/Magnetometer Sensor Node

- `0x0A800401`: Filtered ADC values, received by PC104/CAN controller node.
- `0x0A810401`: Filtered ADC values, received by PC104/CAN controller node.

Servo Controller Node

- `0x0B800801`: Safety pilot receiver timer values for channels 0 to 3, received by PC104/CAN controller node.
- `0x0B810801`: Safety pilot receiver timer values for channels 4 to 7, received by PC104/CAN controller node.
- `0x0B820801`: Servo controller status information, received by PC104/CAN controller node.

HIL Distribution Board Node

- `0x0B800601`: Simulink analogue sensor data for channels 0 to 3, received by PC104/CAN controller node.
- `0x0B810601`: Simulink analogue sensor data for channels 4 to 7, received by PC104/CAN controller node.

- `0x0B820601`: Simulink analogue sensor data for channels 8 to 11, received by PC104/CAN controller node.
- `0x0B830601`: Simulink analogue sensor data for channels 12 to 15, received by PC104/CAN controller node.
- `0x0B840601`: Simulink ultra sonic sensor data for channels 0 and 1, received by PC104/CAN controller node.

D.2 Sensor Signal Conditioning

D.2.1 Rate Gyroscopes and Accelerometers

The rate gyroscope and accelerometer signal conditioning circuits are shown in Figures D.1 and D.2 respectively. Table D.2 lists the component values for the two circuits.

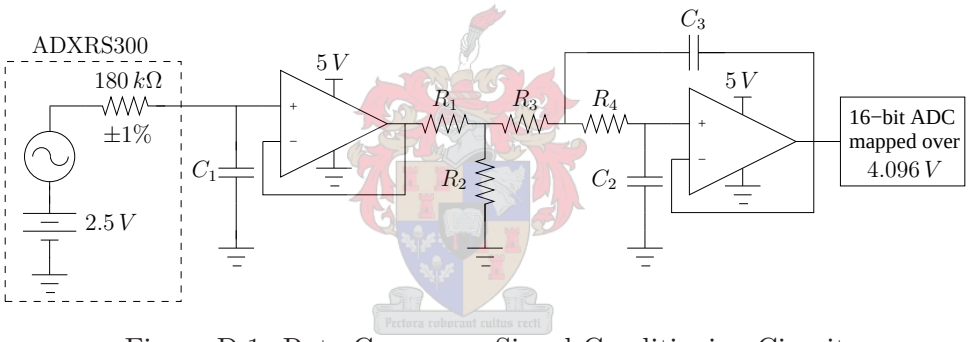


Figure D.1: Rate Gyroscope Signal Conditioning Circuit

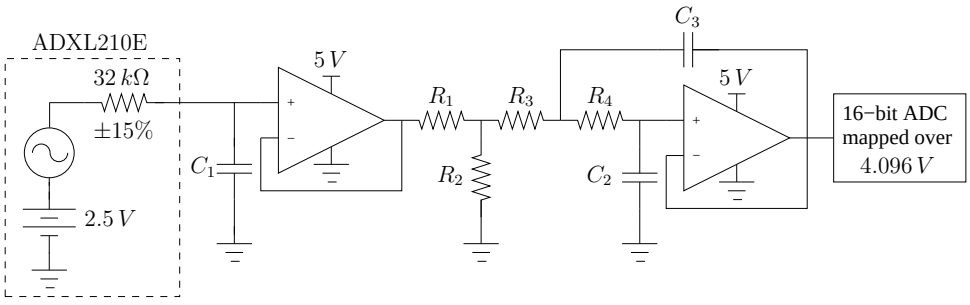


Figure D.2: Accelerometer Signal Conditioning Circuit

D.2.2 Magnetometer

The magnetometer signal conditioning circuit is shown in Figure D.3.

Table D.2: Figures D.1 and D.2 Component Values

Component	Rate Gyroscope	Accelerometer
R_1	$15\,k\Omega$	$15\,k\Omega$
R_2	$68\,k\Omega$	$68\,k\Omega$
R_3	$22\,k\Omega$	$22\,k\Omega$
R_4	$33\,k\Omega$	$33\,k\Omega$
C_1	$6.8\,nF$	$47\,nF$
C_2	$47\,nF$	$47\,nF$
C_3	$47\,nF$	$47\,nF$

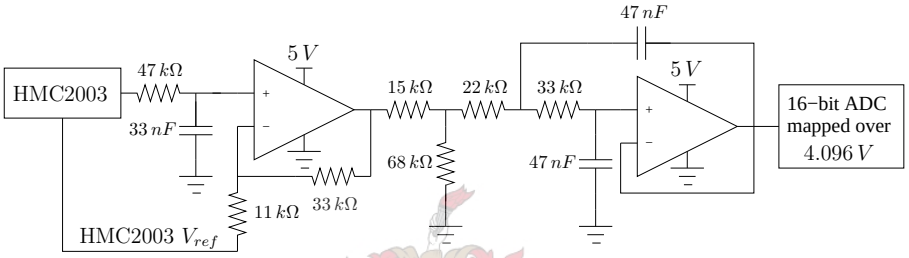


Figure D.3: Magnetometer Signal Conditioning Circuit

D.2.3 Differential Pressure

The differential pressure signal conditioning circuit is shown in Figure D.4.

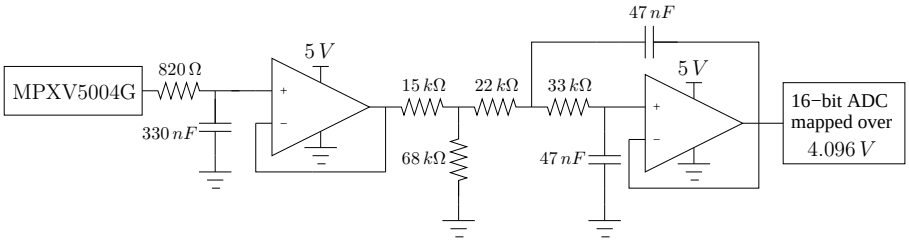


Figure D.4: Differential Pressure Signal Conditioning Circuit

D.2.4 Absolute Pressure

The absolute pressure signal conditioning circuit is shown in Figure D.5.

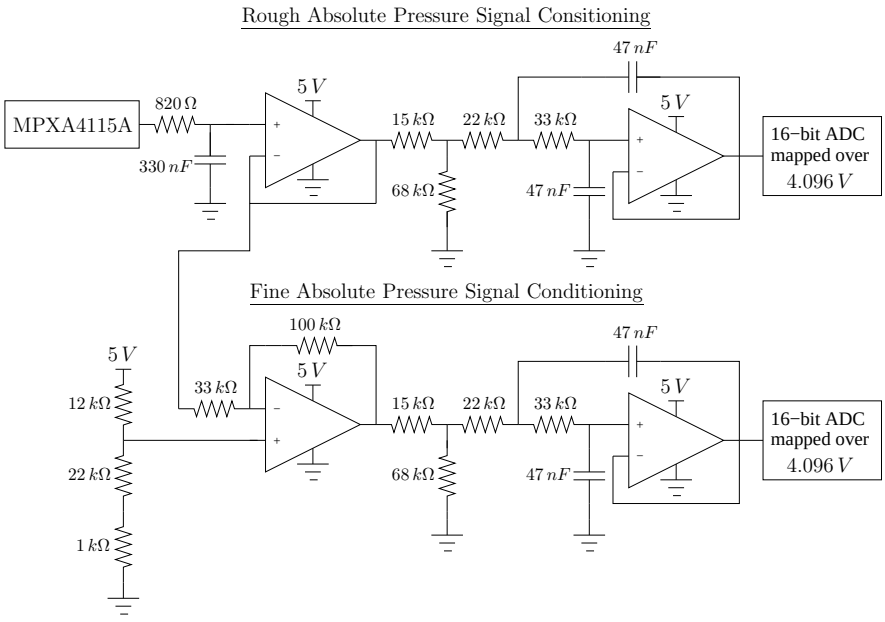


Figure D.5: Absolute Pressure Signal Conditioning Circuit

D.3 Hardware in the Loop Connector Interfaces

The hardware in the loop connector interfaces are shown in Figure D.6. The figure also shows whether a pin is an input or output by means of the I and O mnemonics respectively.

Analogue 0 (O)	1	2	Servo 0 (I)
Analogue 1 (O)	3	4	Servo 1 (I)
Analogue 2 (O)	5	6	Ultra Sonic 0 Ping (I)
Analogue 3 (O)	7	8	Ultra Sonic 1 Ping (I)
Analogue 4 (O)	9	10	Ultra Sonic 0 Echo (O)
Analogue 5 (O)	11	12	Ultra Sonic 1 Echo (O)
Analogue 6 (O)	13	14	HIL Select (I)
Analogue 7 (O)	15	16	Servo 2 (I)
Analogue 8 (O)	17	18	Servo 3 (I)
Analogue 9 (O)	19	20	Servo 4 (I)
Analogue 10 (O)	21	22	Servo 5 (I)
Analogue 11 (O)	23	24	I2C Clock (I/O)
Analogue 12 (O)	25	26	I2C Data (I/O)
Analogue 13 (O)	27	28	SPI Clock (I/O)
Analogue 14 (O)	29	30	SPI Data In (I)
Analogue 15 (O)	31	32	SPI Data Out (O)
No Connection	33	34	SPI Slave Select (I/O)
Analogue GND	35	36	GPS RX (I)
Digital GND	37	38	GPS TX (O)
Servo 7 (I)	39	40	Servo 6 (I)

HIL Connector

Digital GND	1	2	No Connection
Digital GND	3	4	No Connection
Digital GND	5	6	CAN Low
Digital GND	7	8	CAN High
No Connection	9	10	No Connection

CAN Connector

No Connection	1	2	No Connection
GPS TX (O)	3	4	No Connection
GPS RX (I)	5	6	No Connection
No Connection	7	8	No Connection
Digital GND	9	10	No Connection

GPS Connector

Figure D.6: Hardware in the Loop Connector Interfaces