Autonomous Flight of a Model Aircraft

Iain K. Peddle

Thesis presented in partial fulfilment of the requirements for the degree of Master of Science in Electrical & Electronic Engineering at the University of Stellenbosch

Supervisor: Prof G.W. Milne

April 2005

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.

I.K. Peddle: .... Date: ....

Abstract

The successful development of a conventional flight autopilot for a model aircraft is presented. All aspects of the autopilot design are considered, from modeling to flight tests.

A mathematical aircraft model, as a function of the aircraft's physical parameters alone, is presented. A controller architecture capable of regulating the motion variables required for conventional flight using only low cost, off-the-shelf sensors is developed. The controller design complements the aircraft model development technique used, by reducing the sensitivity of the controller performance to the model accuracy.

The avionics and ground station design is presented. The avionics includes a generic Inertial Measurement Unit (IMU). The total avionics cost is only R5000.

Results from three days of flight tests demonstrate the autopilot's success. Its rapid success can largely be attributed to the extensive simulations of the entire autopilot in the two non-linear simulators developed.

Uittreksel

Die suksesvolle ontwerp van 'n konvensionele vliegstuuroutomaat vir 'n modelvliegtuig is voorgedra. Alle aspekte omtrent die stuuroutomaatontwerp, vanaf modellering tot vlugtoetse, is in ag geneem.

'n Wiskundige vliegtuigmodel is voorgedra as 'n funksie van slegs die vliegtuig se fisiese parameters. 'n Beheerder-argitektuur is ontwikkel met goedkoop, maklik bekombare sensore wat geskik is om die bewegingsveranderlikes wat nodig is vir konvensionele vlug te reguleer. Die ontwerp van die beheerder komplimenteer die ontwikkelingstegniek van die vliegtuigmodel, deur die sensitiwiteit van die beheerder se werkverrigting te minimeer relatief tot die akkuraatheid van die model.

Die avionika en grondstasie-ontwerp is voorgedra. Die avionika sluit 'n generiese Inersile Metings Eenheid (IME) in, en die totale koste is slegs R5000.

Resultate van drie dae se vlugtoetse demonstreer die sukses van die stuuroutomaat. Die spoedige sukses kan toegeskryf word aan die omvattende simulasies van die volledige stuuroutomaat wat in die twee nie-linere simulators ontwikkel is.

Acknowledgements

A project of this magnitude cannot be completed by a single person, and I would like to extend my gratitude to everybody who has assisted me. In particular, I would like to thank the following people for their contributions.

- Prof. G.W. Milne for your guidance and for providing me with deeper understanding on so many topics.
- Dr. Thomas Jones for always taking the time to help me. Your advice, support and friendship are much appreciated.
- Denel Aerospace Systems for funding the project.
- Nicol Carstens for your extremely valued input throughout the project and for our friendship.
- Dr. Kas Hamman for doing an excellent job at flying the aircraft.
- Ockert Van Schalkwyk for all your assistance with the flight tests and in repairing the aircraft.
- Reimer Berner for your assistance during the write-up of this thesis.
- All of my friends in the ESL for their assistance during practical tests and for the many laughs we shared during this project.
- My parents and sister for their continued love and support.
- And most importantly, my wonderful fiancée, Laura Taylor. Thank you for your love, patience, understanding and support throughout this project.

In memory of my grandfather Ken Walsh

Contents

List of Figures

List of Tables

Nomenclature

Greek Letters

α	Angle of attack
β	Angle of sideslip
δ_a	Aileron perturbation from trim condition
δ_A	Aileron total deflection
δ_e	Elevator perturbation from trim condition
δ_E	Elevator total deflection
δ_r	Rudder perturbation from trim condition
δ_R	Rudder total deflection
δ_t	Throttle perturbation from trim condition
δ_T	Throttle total deflection
ϵ	Downwash angle
ϵ_T	Thrust vector setting angle
ϕ	Roll angle perturbation
λ	Eigenvalue
θ	Pitch angle perturbation
ρ	Air density
τ_T	Engine time constant
ω_n	Natural frequency
ψ	Yaw angle perturbation
ζ	Damping ratio
Φ	Roll angle
Γ	Dihedral
$\Lambda_{\frac{1}{4}}$	Quarter chord sweep angle
Λ_{le}	Leading edge sweep angle
Θ	Pitch angle
Ψ	Yaw angle

Small Letters

b	Wing span
c	Standard mean chord
$\boldsymbol{c}$	Control array
$\boldsymbol{c_A}$	Aerodynamic control array
$\bar{c}$	Mean aerodynamic chord
e	Airfoil efficiency factor
$\boldsymbol{e}$	Euler angle array
g	Gravitational acceleration
h	Height perturbation
h_0	Quarter of wing $\bar{c}$, from leading edge of wing $\bar{c}$
h_F	Height of fin aerodynamic centre above roll axis
k	Induced drag coefficient
l_F	Fin moment arm
l_T	Tailplane moment arm
m	mass
m_T	Thrust vector moment arm
p	Roll rate perturbation or static pressure
p_N	Longitudinal position error
q	Pitch rate perturbation
r	Yaw rate perturbation
s	Semi-span
u	Axial velocity perturbation
v	Lateral velocity perturbation
w	Normal velocity perturbation
y	Cross track error

Capital Letters

A	Aspect ratio
C_D	Aerodynamic drag coefficient
C_f	Skin friction coefficient
C_l	Aerodynamic roll moment coefficient
C_L	Aerodynamic lift coefficient
C_m	Aerodynamic pitching 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
I_x	Moment of inertia about the roll axis
I_y	Moment of inertia about the pitch axis
I_z	Moment of inertia about the yaw axis
I_{xy}	Roll and pitch product of inertia
I_{xz}	Roll and yaw product of inertia
I_{yz}	Pitch and yaw product of inertia
K	Correction factor for pressure drag
L	Roll moment
M	Pitch moment
N	Yaw moment
P	Roll rate
P_D	Position in the down direction
P_E	Position in the east direction
P_N	Position in the north direction
Q	Pitch rate
R	Yaw rate
S	Wing area
T	Engine thrust magnitude
U	Axial velocity
V	Lateral velocity
$\bar{V}$	Speed
W	Normal velocity
X	Axial force
Y	Lateral force
Z	Normal force

Subscripts

0	Trim and zero angle of attack
A	Aerodynamic force and moment
B	Body axis
E	Inertial axis
F	Fin
G	Gravitational force and moment
T	Engine force and moment as well as tailplane
W	Wing

Acronyms

ADC	Analog to Digital Converter
CS	Checksum
EIRP	Effective Isotropic Radiated Power
GPS	Global Positioning System
GUI	Graphical User Interface
IMU	Inertial Measurement Unit
ISR	Intelligence Surveillance and Reconnaissance
MIMO	Multi Input Multi Output
OEM	Original Equipment Manufacturer
PC	Personal Computer
PCB	Printed Circuit Board
PI	Proportional Integral
PWM	Pulse Width Modulation
RC	Radio Controlled
RF	Radio Frequency
RSSI	Received Signal Strength Indicator
SISO	Single Input Single Output
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver and Transmitter
UAV	Unmanned Air Vehicle
ZOH	Zero Order Hold

Chapter 1

Introduction and Overview

1.1 Background

Unmanned air vehicles (UAV) have been actively used throughout the world for intelligence, surveillance and reconnaissance (ISR) missions since the 1960s. A whole host of vehicles have been produced by over 150 countries, ranging from the 'Pioneer': an inexpensive, over-the-horizon targeting and reconnaissance vehicle, to the 'Global Hawk': a multi-million dollar, high altitude and endurance unit.

Some of South Africa's contributions to UAVs are the 'Seeker' family, designed by the Kentron division of Denel and the ATE (Advanced Technologies and Engineering) designed 'Vulture'. The Seeker II system was designed for real-time reconnaissance, target location and fire-control and has been operational with the South African Air Force since 1986. It has also shown great success in assisting the Police Force in crime-prevention operations. The 'Vulture' is a pneumatically launched vehicle used for artillery targeting in the South African National Defense Force.

Motivated by the desire to contribute to this fast growing field of research, the Department of Electronic Engineering, at the University of Stellebosch, formed a UAV research group in January 2001. This project is the first involving a fixed wing aircraft, with the goal of achieving autonomous flight. This thesis presents the developments of two years of work on the subject, which includes aircraft modeling, control, simulation, implementation and successful flight tests.

1.2 Autopilot Overview

The scope of the project's autopilot design is limited to the autonomous navigation of a conventional model aircraft, within a non-aerobatic flight envelope. Autonomous navigation implies that the aircraft must be capable of automatically guiding itself from one location to another. Further design goals are presented in point form below,

Further Design Goals

1. To use only low cost, *off-the-shelf sensors*
2. To carry out all autopilot functions *on-board* the aircraft
3. To create a *simple* autopilot that can *easily* be *reconfigured* for another aircraft
4. To demonstrate the autopilot's success in a *minimum* number of *flight tests*
5. To contribute to the UAV group's collection of *hardware* and *software tools*
6. To achieve all of the goals *within* a *2 year* Masters program time period

The autopilot design is complicated by the poor measurement characteristics of low cost sensors. However, the controller architecture presented is designed to reduce the sensitivity of the system to the adverse characteristics of these sensors, while still maintaining the functionality required for autonomous navigation.

Performing all autopilot functions on-board the aircraft complements the goal of autonomous navigation well. The on-board calculations imply that the aircraft is not restricted to flying within communications range of a ground station, and that the navigation algorithm can be used to guide the aircraft anywhere within flying distance. The ability to 'fly over the horizon' greatly increases the number of possible applications for the autopilot.

Comprehensive stability and control derivative software was developed to generate the aircraft's mathematical model. The model is a function of the aircraft's physical parameters alone. Together with the simple controller architecture developed, it is possible to easily reconfigure the autopilot for other conventional model aircraft. The autopilot is also designed to reduce its performance sensitivity to the inaccuracies evident in this modeling process.

To demonstrate the autopilot's success in a minimum number of flight tests meant that the entire autopilot needed to be thoroughly simulated before any flights tests were conducted. Two non-linear simulators were designed for this purpose. These simulators were used to debug and fully prepare the controller and navigation algorithms for practical implementation and are invaluable tools for the successful achievement of this goal. These tools can also readily be applied to future projects.

1.3 Thesis outline

The thesis structure is illustrated by the following flow diagram,

Modeling → Analysis → Control → Simulation → Implementation → Testing

The thesis covers all aspects of the autopilot design from first principles to the final product. In Chapter 2, a non-linear aircraft model is developed. In Chapter 3, this model is linearised using small disturbance theory, about a level flight trim condition. The linear aircraft model, specific to the aircraft used in this project, is analysed in Chapter 4 using modal decomposition techniques. The analysis sets the scene for the controller design of Chapter 5, where the controller architecture is developed and the design details are presented. Chapter 6 presents the final algorithms required for autonomous navigation before moving on to a thorough non-linear simulation of the controller and navigation algorithms. This simulation leads the way to the implementation stage by preparing the controller and navigation code for the on-board control microprocessors. The avionics and ground station hardware and software design is then considered in Chapter 7 before the flight test results are provided in Chapter 8.

Chapter 2

Aircraft Dynamic Equations

This chapter begins by defining the axis systems used in the project. Thereafter, the equations of motion for a rigid aircraft are developed from first principles using Newton's laws. An attitude description system is then discussed and the kinematic equations are developed and added to the equations of motion to form a complete set of non-linear aircraft dynamic equations. Finally, the forces and moments that drive these equations are written in terms of the aircraft's states and physical properties.

The dynamic equations are presented in detail in this chapter since they form the basis of the simulation model used in the two non-linear simulators described in Chapter 6.

2.1 Axis System Definition

2.1.1 Inertial Axes

Before Newton's laws can be used to develop a dynamic model for the aircraft, an inertial axis system must be defined. In this axis system, all of Newton's laws can be applied.

With reference to Figure 2.1, the inertial axis system, $F_E(O_E X_E Y_E Z_E)$, is chosen as an earth fixed axis system with its origin at some convenient point on the earth's surface. Most often this will coincide with the runway location below the aircraft's centre of gravity at the start of the motion studied. The axis system is right handed with the positive Z_E-axis pointing towards the centre of the earth. The X_E-axis is perpendicular to the Z_E-axis, positive to the North. The Y_E-axis is perpendicular to the $X_E Z_E$-plane, positive to the East.

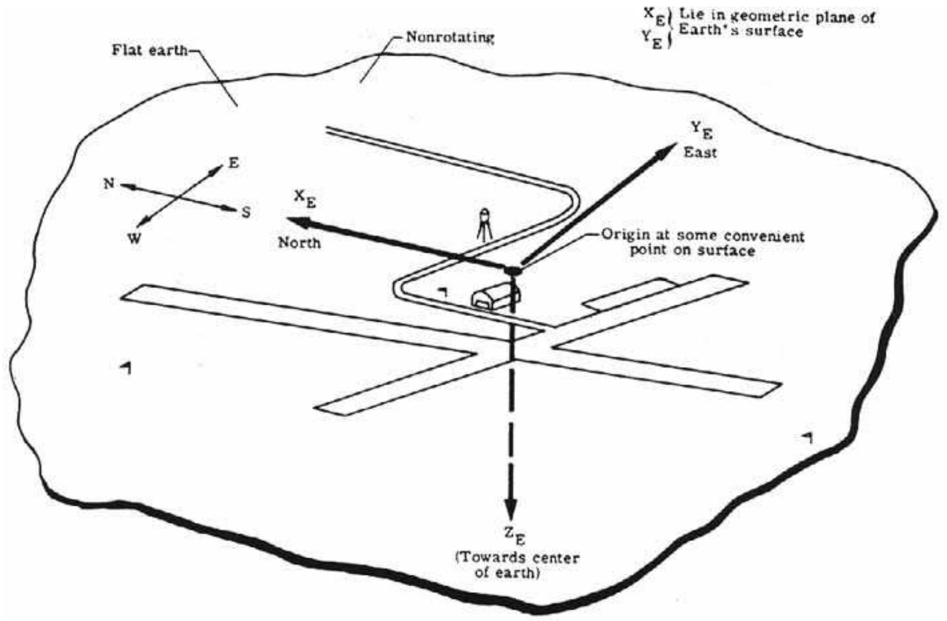

X_E } Lie in geometric plane of
Y_E } Earth's surface
Flat earth
Nonrotating
Y_E
East
N
E
S
W
X_E
North
Origin at some convenient point on surface
Z_E
(Towards center of earth)

Figure 2.1: *Inertial Axes Definition [48]*

The earth is considered flat and non-rotating: essential qualities of an inertial axis system. These assumptions are reasonable since the ranges of flight are small and the typical angular rotations of the aircraft are much greater than the angular rotation of the earth.

The inertial axis system is said to span inertial space.

2.1.2 Body Axes

The body axis system is essential to a concise set of aircraft dynamic equations and this will become clearer as the aircraft model is developed. Consider Figure 2.2.

The body axis system, $F_B(O_BX_BY_BZ_B)$, is a right handed reference frame, attached to the aircraft, with its centre coinciding with the aircraft's centre of gravity. The X_B-axis is in the aircraft's plane of symmetry and is positive in the 'forward' direction. The exact 'forward' direction is determined by the specific body axis system used. The Z_B-axis is also in the plane of symmetry, perpendicular to the X_B-axis and positive pointing 'downward'. The Y_B-axis is perpendicular to the X_BZ_B-plane, positive to the right (starboard wing). The aircraft is said to *roll* about the X_B-axis, *pitch* about the Y_B-axis and *yaw* about the Z_B-axis with the positive directions indicated in Figure 2.2.

Two specific body axis systems used in this project are discussed below.

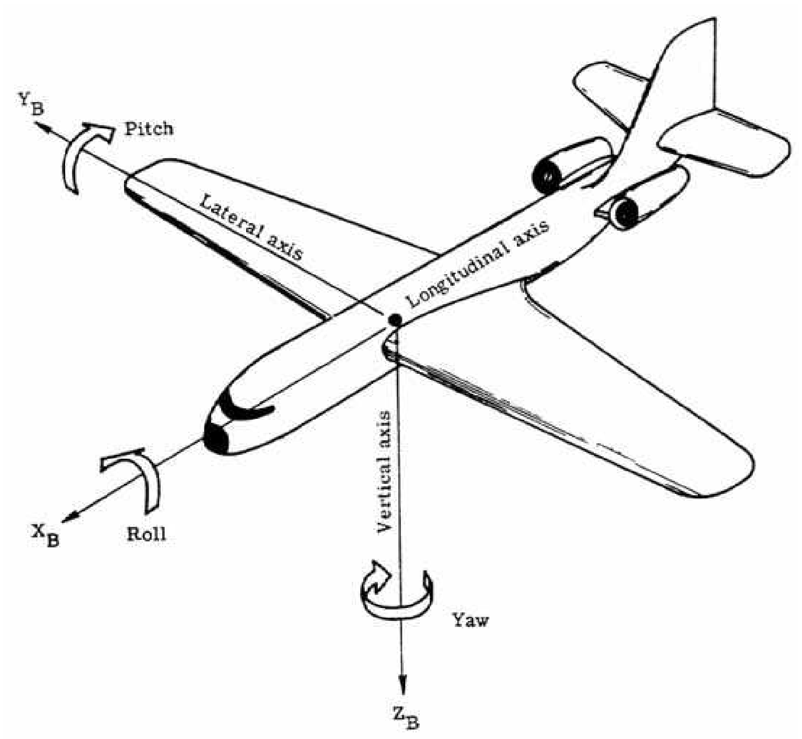


Figure 2.2: *Body Axes Definition [48]*

Reference Line Body Axes

When the 'forward' direction of the OX_B-axis is chosen to lie parallel to some longitudinal reference line of the aircraft, then the body axis system is most often referred to simply as *Body Axes*, but is referred to as *Reference Line Body Axes* here, to avoid confusion with body axes in general. In this project, the longitudinal reference line was chosen as the wing chord line because this is convenient from an aerodynamic analysis point of view. The advantage of reference line body axes is that all of the aircraft's geometric and inertial properties are easily quantified and are not a function of the aircraft's state.

Stability Axes

When the 'forward' direction of the OX_B-axis coincides with the aircraft's velocity vector during a trim flight condition, then the body axis system is referred to as *Stability Axes*. Stability axes have the advantage that they reduce the aerodynamic model to its simplest possible form and thus maximise the visibility of the physical phenomena involved.

The aircraft linear and angular velocity vectors as well as the external force and moment vectors are defined below. It is important to realise that they are all measured relative to inertial space, but are coordinated in the instantaneous body axis frame, as indicated by the subscript B.

$$\boldsymbol{V_B} = U\boldsymbol{i_B} + V\boldsymbol{j_B} + W\boldsymbol{k_B} \tag{2.1}$$

$$\boldsymbol{\omega_B} = P\boldsymbol{i_B} + Q\boldsymbol{j_B} + R\boldsymbol{k_B} \tag{2.2}$$

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

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

where, $\boldsymbol{i_B}$, $\boldsymbol{j_B}$ and $\boldsymbol{k_B}$ are unit vectors in the OX_B, OY_B and OZ_B-axis directions respectively. The linear velocity vector $\boldsymbol{V_B}$ can be written in polar form,

$$\bar{V} = \sqrt{U^2 + V^2 + W^2} \tag{2.5}$$

$$\alpha = \tan^{-1}(W/U) \tag{2.6}$$

$$\beta = \sin^{-1}(V/\bar{V}) \tag{2.7}$$

where, $\bar{V}$, α and β are the aircraft speed, angle of attack and angle of sideslip respectively. The speed is denoted with a bar so that it is not confused with the lateral velocity variable V. The conventional aerodynamic control surface deflection variables are defined below,

- δ_A - Aileron control surface deflection
  A positive deflection causes a negative rolling moment.
- δ_E - Elevator control surface deflection
  A positive deflection causes a negative pitching moment.
- δ_R - Rudder control surface deflection
  A positive deflection causes a negative yawing moment.

and summarised for notation convenience by the following aerodynamic control array.

$$\boldsymbol{c_A} = (\delta_A, \delta_E, \delta_R) \tag{2.8}$$

It is also prudent to include a throttle actuation variable, denoted by δ_T, that is related to the output magnitude of the aircraft thrust vector. The aerodynamic control array of equation (2.8) is extended to the aircraft control array of the form,

$$\boldsymbol{c} = (\boldsymbol{c_A}, \delta_T) \tag{2.9}$$

2.2 Development of Equations of Motion

Now that the axis systems and notation have been defined, the equations of motion for a rigid aircraft can be derived from Newton's second law of motion. This states that the summation of all external forces acting on a body must be equal to the time rate of change of its momentum, and the summation of all external moments must be equal to the time rate of change of its angular momentum [4]. The time rates of change are with respect to inertial space and the laws can be expressed by the following vector equations,

$$\boldsymbol{F_B} = \frac{d}{dt}(m\boldsymbol{V_B})\Big]_E \tag{2.10}$$

$$\boldsymbol{M_B} = \frac{d}{dt}(\boldsymbol{H_B})\Big]_E \tag{2.11}$$

where m is the aircraft's mass and $\boldsymbol{H_B}$ is the aircraft's angular momentum defined as,

$$\boldsymbol{H_B} = \boldsymbol{I_B}\ \boldsymbol{\omega_B} \tag{2.12}$$

where,

$$\boldsymbol{I_B} = \begin{bmatrix} I_x^2 & I_{xy} & I_{xz} \\ I_{xy} & I_y^2 & I_{yz} \\ I_{xz} & I_{yz} & I_z^2 \end{bmatrix} \tag{2.13}$$

is the moment of inertia dyadic of the aircraft in body axes. Because the aircraft is assumed symmetrical about the $X_B Z_B$-plane, the products of inertia, I_{xy} and I_{yz} are zero.

It would be convenient if the derivatives in (2.10) and (2.11) were with respect to body axes. Since the body axis system rotates with angular velocity $\boldsymbol{\omega_B}$ with respect to inertial space, the equation of Coriolis [4] is used to relate the inertial time derivative to the body time derivative. This is given as,

$$\frac{d}{dt}(\boldsymbol{R})\Big]_E = \frac{d}{dt}(\boldsymbol{R})\Big]_B + \boldsymbol{\omega_B} \times \boldsymbol{R} \tag{2.14}$$

where $\boldsymbol{R}$ is an arbitrary vector.

Applying equation (2.14) to equations (2.10) and (2.11) gives,

$$\boldsymbol{F_B} = \frac{d}{dt}(m\boldsymbol{V_B})\Big]_B + \boldsymbol{\omega_B} \times (m\boldsymbol{V_B}) \tag{2.15}$$

$$\boldsymbol{M_B} = \frac{d}{dt}(\boldsymbol{I_B\omega_B})\Big]_B + \boldsymbol{\omega_B} \times (\boldsymbol{I_B\omega_B}) \tag{2.16}$$

Equations (2.15) and (2.16) are the vector equations for general six degree of freedom motion. They are more commonly displayed in scalar form as a set of six, non-linear, coupled differential equations. These are shown below assuming only that the aircraft is symmetrical about the X_BZ_B-plane and that the inertial properties do not change over time.

$$X = m(\dot{U} + WQ - VR) \tag{2.17}$$

$$Y = m(\dot{V} + UR - WP) \tag{2.18}$$

$$Z = m(\dot{W} + VP - UQ) \tag{2.19}$$

$$L = \dot{P}I_x - \dot{R}I_{xz} + QR(I_z - I_y) - PQI_{xz} \tag{2.20}$$

$$M = \dot{Q}I_y + PR(I_x - I_z) + (P^2 - R^2)I_{xz} \tag{2.21}$$

$$N = \dot{R}I_z - \dot{P}I_{xz} + PQ(I_y - I_x) + QRI_{xz} \tag{2.22}$$

2.3 Attitude Definition

Given the inertial properties and the current state of the aircraft, the dynamic equations developed in Section 2.2 describe how the forces and moments coordinated in the body axis frame cause rates of change in the body axis linear and angular velocities. Thus, in order to describe the motion of the aircraft in inertial space, it is necessary to specify the orientation of the body axis system with respect to the inertial axis system. This is commonly done using a set of three Euler angles, which define an ordered set of rotations from one axis system to another.

2.3.1 Euler Angles and the Transformation Matrix

Define an axis system $F_0(O_0X_0Y_0Z_0)$ that coincides with the inertial axis system F_E. Consider a vector $\boldsymbol{V}$ coordinated in F_0 and described as $\boldsymbol{V_0} = [x_0, y_0, z_0]$. F_0 is now moved through three consecutive rotations to align itself with the body axis system with the goal of attaining a relationship between $\boldsymbol{V}$ coordinated in the original and final axis systems.

1. Rotate F_0 about the OZ_0-axis by the *yaw angle* Ψ, to create a new coordinate system $F_1(O_1X_1Y_1Z_1)$. The vector $\boldsymbol{V}$, coordinated in F_1, is now described as $\boldsymbol{V_1} = [x_1, y_1, z_1]$. Note that the rotation leaves $z_0 = z_1$.

2. Rotate F_1 about the OY_1-axis by the *pitch angle* Θ, to create a new coordinate system $F_2(O_2X_2Y_2Z_2)$. The vector $\boldsymbol{V}$, coordinated in F_2, is now described as $\boldsymbol{V_2} = [x_2, y_2, z_2]$. Note that the rotation leaves $y_1 = y_2$.

3. Rotate F_2 about the OX_2-axis by the *roll angle* Φ, to create a new coordinate system $F_3(O_3X_3Y_3Z_3)$. The vector $\boldsymbol{V}$, coordinated in F_3, is now described as $\boldsymbol{V_3} = [x_3, y_3, z_3]$. Note that the rotation leaves $x_2 = x_3$.

The coordinate system F_3 coincides with the body axis system F_B. With the rotation sequence defined, the transformation matrix that converts V_0 to V_3 or equivalently V_E to V_B can easily be found. Consider rotation 1 as shown in Figure 2.3.

The transformation of $V_0 \Rightarrow V_1$ can be described by,

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} \cos\Psi & \sin\Psi & 0 \\ -\sin\Psi & \cos\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \tag{2.23}$$

$$\boldsymbol{V_1} = \boldsymbol{T_\Psi V_0}$$

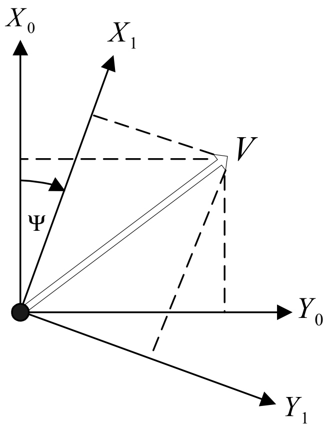


Figure 2.3: *Yaw Rotation*

Similarly, the transformations $V_1 \Rightarrow V_2$ and $V_2 \Rightarrow V_3$ can be described by,

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} \cos\Theta & 0 & -\sin\Theta \\ 0 & 1 & 0 \\ \sin\Theta & 0 & \cos\Theta \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \tag{2.24}$$

$$\boldsymbol{V_2} = \boldsymbol{T_\Theta V_1}$$

$$\begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Phi & \sin\Phi \\ 0 & -\sin\Phi & \cos\Phi \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} \tag{2.25}$$

$$\boldsymbol{V_3} = \boldsymbol{T_\Phi V_2}$$

Combining equations (2.23),(2.24) and (2.25) gives,

$$\boldsymbol{V_3} = [\boldsymbol{T_\Phi T_\Theta T_\Psi}]\ \boldsymbol{V_0} \tag{2.26}$$

or,

$$\boldsymbol{V_B} = [\boldsymbol{T_\Phi T_\Theta T_\Psi}]\ \boldsymbol{V_E} \tag{2.27}$$

Thus, given the three Euler angles (Φ, Θ, Ψ), it is possible to convert any vector coordinated in the inertial axis system to the equivalent vector coordinated in the body axis system by means of the transformation matrix $[\boldsymbol{T_\Phi T_\Theta T_\Psi}]$. Of course, the inverse transformation could be carried out by using the inverse of the transformation matrix $[\boldsymbol{T_\Phi T_\Theta T_\Psi}]^{-1}$. However, simple algebraic manipulation shows that the transformation matrix is orthogonal and thus its inverse exists and is merely its transpose,

$$\boldsymbol{V_E} = [\boldsymbol{T_\Psi^T T_\Theta^T T_\Phi^T}]\ \boldsymbol{V_B} \tag{2.28}$$

The transformation matrix is commonly referred to as the Direction Cosine Matrix (DCM) since each of its elements is the cosine of the angle between a body unit vector and the respective inertial axis [49]. For notation purposes, the Euler angle set is summarised in the following array form,

$$\boldsymbol{e} = (\Phi, \Theta, \Psi) \tag{2.29}$$

2.3.2 Kinematic Equations

It is now necessary to derive a set of differential equations that allow the dynamic calculation of the Euler angles. The angular velocity of the aircraft can also be expressed as a vector sum of the Euler angular velocities [2],

$$\boldsymbol{\omega} = \dot{\Phi}\boldsymbol{i_\Phi} + \dot{\Theta}\boldsymbol{j_\Theta} + \dot{\Psi}\boldsymbol{k_\Psi} \tag{2.30}$$

where,

- $\boldsymbol{i_\Phi}$ is the unit vector along the OX_2-axis
- $\boldsymbol{j_\Theta}$ is the unit vector along the OY_1-axis
- $\boldsymbol{j_\Psi}$ is the unit vector along the OZ_0-axis

To equate the body axis angular velocity components P, Q and R to the Euler components, the unit vectors $\boldsymbol{i_\Phi}$, $\boldsymbol{j_\Theta}$, $\boldsymbol{k_\Psi}$ must be resolved into the body axis frame F_B as follows,

$$\boldsymbol{i_{\Phi_B}} = \boldsymbol{T_\Phi i_\Phi} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{2.31}$$

$$\boldsymbol{j_{\Theta_B}} = \boldsymbol{T_\Phi T_\Theta j_\Theta} = \begin{bmatrix} 0 \\ \cos\Phi \\ -\sin\Phi \end{bmatrix} \tag{2.32}$$

$$\boldsymbol{k}_{\boldsymbol{\Psi}_B} = \boldsymbol{T}_{\boldsymbol{\Phi}}\boldsymbol{T}_{\boldsymbol{\Theta}}\boldsymbol{T}_{\boldsymbol{\Psi}}\boldsymbol{k}_{\boldsymbol{\Psi}} = \begin{bmatrix} -\sin\Theta \\ \cos\Theta\sin\Phi \\ \cos\Theta\cos\Phi \end{bmatrix} \tag{2.33}$$

Equating the body axis angular rate vector with the vector sum of equations (2.31), (2.32) and (2.33), scaled by their respective Euler angular rates, the desired transformation is given as,

$$\begin{bmatrix} P \\ Q \\ R \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\Theta \\ 0 & \cos\Phi & \cos\Theta\sin\Phi \\ 0 & -\sin\Phi & \cos\Theta\cos\Phi \end{bmatrix} \begin{bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} \tag{2.34}$$

This matrix is not orthogonal but its inverse does exist,

$$\begin{bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\Phi\tan\Theta & \cos\Phi\tan\Theta \\ 0 & \cos\Phi & -\sin\Phi \\ 0 & \sin\Phi\sec\Theta & \cos\Phi\sec\Theta \end{bmatrix} \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \tag{2.35}$$

subject to the constraint that $|\Theta| \neq \pi/2$. This discontinuity is one of the major drawbacks of using Euler angles and often motivates the use of the Quaternion: an alternative angular rotation measure that has no singularities [49]. However, Euler angles are well suited for non-aerobatic flight where $|\Theta| \ll \pi/2$ and the added complexity of using quaternions is not warranted. For this reason, Euler angles are used throughout the text.

2.4 Forces and Moments

Before the equations of motion, (2.17) to (2.22), can be solved, the forces and moments acting upon the aircraft must be determined. The equations of motion developed up to this point are valid for any body axis system subject to the constraint that the OX_BZ_B-plane coincides with the aircraft's plane of symmetry. However, to model the forces and moments that act upon the aircraft, a particular set of body axes must be chosen.

Two specific body axis systems were discussed in Section 2.1.2 and their particular advantages were stated. In the developments to follow, the stability axis system is used throughout. Although this choice simplifies the aerodynamic model developed, it complicates the quantification of the aircraft's geometric and inertial properties because the body axis system changes for each choice of trim condition. However, once the trim angle of attack (α_0) has been set, the aircraft's geometric and inertial properties can simply be measured in the reference line body axis system and coordinated into the stability axis system using the transformation equations summarised in Appendix A. Thus, all of the aircraft's geometric and inertial properties used in the developments to follow are assumed to already be coordinated in the stability axes system.

With the axis system defined, the deterministic forces and moments that act on conventional model aircraft are considered. These forces and moments can be divided into aerodynamic, gravitational and engine groups.

2.4.1 Aerodynamic

The aerodynamic analysis of the aircraft is restricted to two dimensional, subsonic, incompressible flow. Two dimensional flow means that the airflow velocity at every point on an airfoil has a direction that is parallel to some reference plane, such as the OX_BZ_B-axis during symmetrical flow. As a result, the aerodynamic forces can merely be resolved into lift and drag components. Restricting the analysis to subsonic, incompressible flow greatly simplifies the aerodynamic analysis and is a reasonable assumption for model aircraft flight, where speeds are well below Mach 0.4 [5].

Bernoulli's equation and the continuity principle for incompressible fluids are fundamental to the understanding of the generation of aerodynamic forces. The former is listed in equation (2.36) and states that the total pressure of a fluid (p_T) is the sum of the static (p) and dynamic ($\frac{1}{2}\rho V^2$) pressures and is a constant (B) as long as the system is not acted upon by an external force. In words, the continuity principle states that if the path area normal to the fluid direction of flow is changed, then the flow velocity must change such that the same fluid volume continues to pass per second.

$$p_1 + \frac{1}{2}\rho\bar{V}_1^2 = p_2 + \frac{1}{2}\rho\bar{V}_2^2 = p_T = B \tag{2.36}$$

Armed with the above information, it is clear that if the airflow is forced by an airfoil to travel a longer path than the undisturbed flow would, then the airflow velocity must speed up (Continuity principle), which then means that the dynamic pressure increases and thus the static pressure must decrease (Bernoulli). By varying the airflow path lengths over an airfoil, the static pressure on the surface can be manipulated to create the desired net force.

Now, if the freestream airflow velocity and static pressure are denoted $\bar{V}$ and p respectively and the local airflow velocity and static pressure over a part of the airfoil denoted $\bar{V}_A$ and p_A respectively, then after substitution into and manipulation of equation (2.36), an equation for the difference between the local and freeflow static pressure is,

$$p_A - p = \frac{1}{2}\rho\bar{V}^2\left[1 - \left(\frac{\bar{V}_A}{\bar{V}}\right)^2\right] = \frac{1}{2}\rho\bar{V}^2 C_{A_L} \tag{2.37}$$

where C_{A_L} is a dimensionless proportionality coefficient. The result shows that during subsonic, incompressible flow, the difference between the local and freeflow static pressure is proportional to the dynamic pressure of the freestream flow. By integrating the static pressure difference over the entire airfoil area, the resultant aerodynamic force can be shown to be proportional to the product of the dynamic pressure and the airfoil area,

$$F_A = \frac{1}{2}\rho\bar{V}^2 S_A C_A \tag{2.38}$$

where S_A is the airfoil area and C_A is a non-dimensional proportionality constant that quantifies the airfoil aerodynamic force produced as a function of its shape and orientation with respect to the freeflow velocity vector.

The aerodynamic force and moment analysis thus becomes a matter of calculating the dimensionless force and moment coefficients of the aircraft coordinated in body axes.

$$X_A = \frac{1}{2}\rho\bar{V}^2 S\ C_x \tag{2.39}$$

$$Y_A = \frac{1}{2}\rho\bar{V}^2 S\ C_y \tag{2.40}$$

$$Z_A = \frac{1}{2}\rho\bar{V}^2 S\ C_z \tag{2.41}$$

$$L_A = \frac{1}{2}\rho\bar{V}^2 Sb\ C_l \tag{2.42}$$

$$M_A = \frac{1}{2}\rho\bar{V}^2 S\bar{c}\ C_m \tag{2.43}$$

$$N_A = \frac{1}{2}\rho\bar{V}^2 Sb\ C_n \tag{2.44}$$

where S, b and $\bar{c}$ are the aircraft's wing area, wing span and wing mean aerodynamic chord respectively and are used as the reference areas and lengths for the aircraft non-dimensional aerodynamic analysis.

It is now necessary to develop equations for the dimensionless force and moment coefficients of equations (2.39) to (2.44). These coefficients are functions of the force and moment coefficients of the various aircraft airfoils and the aircraft geometry. To simplify the aerodynamic model, the aircraft is divided into four major aerodynamic surfaces: the wing, tailplane, fin and fuselage.

The aerodynamic analysis is now divided into a symmetric flow (longitudinal motion) and asymmetric flow (lateral motion) analysis. During symmetric flow, the lateral aerodynamic coefficients (C_y,C_l,C_n) are exactly zero due to the symmetry of the aircraft. During asymmetric flow however, there is aerodynamic interference between the different aircraft surfaces, which means that all of the aerodynamic coefficients are affected. However, the effect of asymmetric flow on the longitudinal force and moment coefficients is typically small compared to that of the symmetric flow and is assumed zero. This assumption decouples the longitudinal and lateral aerodynamic model.

Longitudinal Model

The following assumptions are made about the aircraft airfoils during symmetric flow,

- Only the wing and the tailplane contribute to the aircraft lift
- Only the wing and the fuselage contribute to the aircraft drag

The fuselage lift is ignored because it is typically a lot smaller than that of the wing. The tailplane lift is also often negligible compared to that of the wing, but it is included in the model because of its significant contribution to the pitching moment coefficient. The tailplane drag however, is ignored because it is negligible compared to that of the wing and fuselage. The symmetrical airflow together with the symmetry of the aircraft, implies that the fin produces no lift force and its drag force is ignored for the same reason as that of the tailplane.

With the above assumptions and with reference to [1], the longitudinal aerodynamic coefficients can be written as,

$$C_x = -C_D \cos\alpha + C_L \sin\alpha \tag{2.45}$$

$$C_z = -C_D \sin\alpha - C_L \cos\alpha \tag{2.46}$$

$$C_m = C_{m_0} + (h - h_0)C_{L_W} - \left(\frac{S_T l_T}{S\bar{c}}\right)C_{L_T} \tag{2.47}$$

where,

$$C_L = C_{L_W} + \left(\frac{S_T}{S}\right)C_{L_T} \tag{2.48}$$

$$C_D = C_{D_W} + \left(\frac{S_B}{S}\right)C_{D_B} \tag{2.49}$$

and the subscripts W, T and B indicate wing, tailplane and fuselage (body) respectively. C_L and C_D are the lift and drag coefficients of the respective airfoils and C_{m_0} is the zero lift pitching moment coefficient of the aircraft as a result of the camber of the wing, tailplane and fuselage. S_T is the tailplane area and l_T is the length along the OX_B-axis from the aircraft's centre of gravity to the quarter chord location of the tailplane's mean aerodynamic chord. h is the distance, as a fraction of $\bar{c}$, between the aircraft's centre of gravity and the leading edge of the wing's mean aerodynamic chord. h_0 is the quarter chord location on the wing's mean aerodynamic chord. It is standard practice to use the quarter chord location of an airfoil's mean aerodynamic chord as the reference point in the moment arm lengths because the lift and drag forces of a subsonic airfoil act very close to this point [1].

To complete the longitudinal aerodynamic model, expressions for the wing and tailplane lift coefficients (C_{L_W} and C_{L_T}) and the wing and fuselage drag coefficients (C_{D_W} and C_{D_B}) must be provided. These expressions are typically functions of the airfoil properties, airfoil location and the aircraft motion variables. To gain insight into these coefficients, they are considered here under static, symmetric flow conditions where the effects of induced angles of incidence due to the aircraft motion are neglected,

$$C_{L_W} = f_W(\alpha_0 + \alpha) \tag{2.50}$$

$$C_{D_W} = C_{D_{W_0}} + k_W C_{L_W}^2 \tag{2.51}$$

$$C_{L_T} = \frac{\partial C_{L_T}}{\partial \alpha}\left[\alpha_T + \left(1 - \frac{d\epsilon}{d\alpha}\right)(\alpha_0 + \alpha)\right] + \frac{\partial C_{L_T}}{\partial \delta_E}\delta_E \tag{2.52}$$

$$C_{D_B} = C_{D_{B_0}} \tag{2.53}$$

where, f_W is the non-linear function that describes the wing's relationship to the angle of incidence, $C_{D_{W_0}}$ is the wing parasitic drag coefficient, k_W is the wing induced drag coefficient, α_T is the tailplane setting angle, $d\epsilon/d\alpha$ is the downwash angle variation with angle of attack of the wing and $C_{D_{B_0}}$ is the fuselage parasitic drag coefficient.

The function f_W describes the complete lift curve slope of the wing with stall effects included. This extends the applicability of the model to include high angle of attack flight conditions such as landings and aerobatic flight. The tailplane lift coefficient is modeled as linear in its local angle of attack and elevator deflection. The linearity in the local angle of attack is assumed because the tailplane of the aircraft used in this project, lies in the downwash flow off the wing. The downwash flow reduces the local angle of incidence and thus results in the wing stalling before the tailplane does. The linearity in elevator deflection is assumed because the deflection angles are typically small. The wing drag coefficient is modeled in the standard way for subsonic aircraft: as the sum of a parasitic and induced drag coefficient [5]. Since the fuselage is modeled as not providing any lift, its drag coefficient simply reduces to the fuselage parasitic drag coefficient.

In Chapter 3, the coefficient models of equations (2.50) to (2.53) are extended by including linear terms in the aircraft motion variables to form the final dynamic longitudinal airfoil coefficient models used in this project.

Lateral Model

The following assumptions are made about the aircraft airfoils during asymmetric flow,

- Only the lift force of the fin contributes to the aircraft side force
- Only the wing and fin contribute to the aircraft roll and yaw moments

During asymmetric flow, there is an induced angle of incidence on the fin, which produces a side force and corresponding roll and yaw moments as a function of the fin's aerodynamic centre location. Asymmetric flow also induces differential lift across the wings which predominantly gives rise to roll and yaw moments with negligible side force. Tailplane effects are ignored during asymmetric flow because they are negligible compared to those of the wing. The fuselage effects can be significant but are very difficult to quantify without using complex modeling techniques. Neglecting these effects does introduce errors into the lateral aerodynamic model, but because they are not dominant, a model of satisfactory accuracy can still be developed.

With the above assumptions and with reference to [1], the lateral aerodynamic coefficients can be written as,

$$C_y = -\left(\frac{S_F}{S}\right) C_{L_F} \cos\beta \tag{2.54}$$

$$C_l = C_{l_W} - \left(\frac{S_F h_F}{Sb}\right) C_{L_F} \cos\beta \tag{2.55}$$

$$C_n = C_{n_W} + \left(\frac{S_F l_F}{Sb}\right) C_{L_F} \cos\beta \tag{2.56}$$

where, C_{L_F} is the lift coefficient of the fin and C_{l_W} and C_{n_W} are the roll and yaw moment coefficients of the wing respectively. The wing's moment coefficients quantify the differential lift and drag forces experienced under asymmetric flow. h_F is the height of the fin's aerodynamic centre above the OX_B-axis and l_F is the distance along the OX_B-axis from the aircraft's centre of gravity to the quarter chord location of the fin's mean aerodynamic chord.

All that is required to complete the lateral aerodynamic model is to obtain expressions for the fin lift coefficients and the wing roll and yaw moment coefficients. These expressions are typically functions of the airfoil properties, airfoil location and the aircraft motion variables. Continuing in a similar fashion to the analysis of the longitudinal airfoil coefficients, the lateral coefficients are considered here under static, symmetric flow conditions, where the effects of induced angles of incidence due to the aircraft motion are neglected. Under these conditions the coefficients are of course simply zero,

$$C_{L_F} = 0 \tag{2.57}$$

$$C_{l_W} = 0 \tag{2.58}$$

$$C_{n_W} = 0 \tag{2.59}$$

In Chapter 3 however, the coefficient models of equations (2.57) to (2.59) are extended by including linear terms in the aircraft motion variables to form the final dynamic lateral airfoil coefficient models used in this project.

2.4.2 Gravitational

The gravitational force acting on a conventional aircraft is easily coordinated into body axes using the transformation matrix of equation (2.27),

$$\begin{bmatrix} X_G \\ Y_G \\ Z_G \end{bmatrix}_B = \begin{bmatrix} \boldsymbol{T_\Phi T_\Theta T_\Psi} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}_E \tag{2.60}$$

where, m is the aircraft mass and g is the gravitational force per unit mass. Equation (2.60) can be simplified to,

$$\begin{bmatrix} X_G \\ Y_G \\ Z_G \end{bmatrix} = \begin{bmatrix} -mg\sin\Theta \\ mg\cos\Theta\sin\Phi \\ mg\cos\Theta\cos\Phi \end{bmatrix} \tag{2.61}$$

All gravitational moment contributions are zero since, by definition, the gravitational forces act through the aircraft's centre of gravity. Thus,

$$L_G = M_G = N_G = 0 \tag{2.62}$$

2.4.3 Engine

Model aircraft display a wide variety of engine types and locations. However, for this project it is assumed that the aircraft has a single, centrally positioned propeller engine. The engine is modeled as a rigid unit, where all inertial and gyroscopic forces, as a result of moving parts, are considered negligible because of their inertia relative to the aircraft's. With reference to Figure 2.4, the engine model can be written as follows,

$$X_T = T\cos\epsilon_T \tag{2.63}$$

$$Z_T = -T\sin\epsilon_T \tag{2.64}$$

$$M_T = Tm_T \tag{2.65}$$

$$Y_T = L_T = N_T = 0 \tag{2.66}$$

where, T is the magnitude of the engine thrust vector in the plane of symmetry, ϵ_T is the thrust vector setting angle and m_T is the thrust vector moment arm. Thus, the task is reduced to finding an equation for the engine thrust magnitude (T), which is typically a function of the aircraft states and the throttle actuation variable. An expression for the thrust magnitude is developed in the linear analysis of Chapter 3.

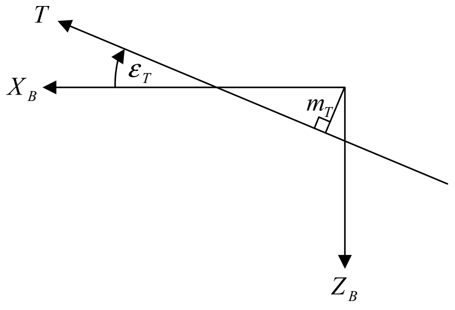


Figure 2.4: *Engine Thrust Geometry*

2.5 Summary

In this chapter, the dynamic equations for aircraft motion have been developed and are restated here for convenience.

$$X_A + X_G + X_T = m(\dot{U} + WQ - VR) \tag{2.67}$$

$$Y_A + Y_G + Y_T = m(\dot{V} + UR - WP) \tag{2.68}$$

$$Z_A + Z_G + Z_T = m(\dot{W} + VP - UQ) \tag{2.69}$$

$$L_A + L_G + L_T = \dot{P}I_x - \dot{R}I_{xz} + QR(I_z - I_y) - PQI_{xz} \tag{2.70}$$

$$M_A + M_G + M_T = \dot{Q}I_y + PR(I_x - I_z) + (P^2 - R^2)I_{xz} \tag{2.71}$$

$$N_A + N_G + N_T = \dot{R}I_z - \dot{P}I_{xz} + PQ(I_y - I_x) + QRI_{xz} \tag{2.72}$$

$$\begin{bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\Phi\tan\Theta & \cos\Phi\tan\Theta \\ 0 & \cos\Phi & -\sin\Phi \\ 0 & \sin\Phi\sec\Theta & \cos\Phi\sec\Theta \end{bmatrix} \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \tag{2.73}$$

$$\begin{bmatrix} \dot{P}_N \\ \dot{P}_E \\ \dot{P}_D \end{bmatrix} = \begin{bmatrix} C_\Psi C_\Theta & (C_\Psi S_\Theta S_\Phi - S_\Psi C_\Phi) & (C_\Psi S_\Theta C_\Phi + S_\Psi S_\Phi) \\ S_\Psi C_\Theta & (S_\Psi S_\Theta S_\Phi + C_\Psi C_\Phi) & (S_\Psi S_\Theta C_\Phi - C_\Psi S_\Phi) \\ -S_\Theta & C_\Theta S_\Phi & C_\Theta C_\Phi \end{bmatrix} \begin{bmatrix} U \\ V \\ W \end{bmatrix} \tag{2.74}$$

where, equations (2.67) to (2.72) are taken directly from equations (2.17) to (2.22) with the respective force and moment variables substituted, equation (2.73) is equation (2.35) and equation (2.74) is calculated using equation (2.28) and specific to the aircraft linear velocity transformation. P_N, P_E and P_D are the aircraft north, east and down positions respectively. $S_{()}$ and $C_{()}$ indicate the sine and cosine of the subscripted angles respectively.

Chapter 3

Development of a Linear Aircraft Model

In this chapter, a linear aircraft model is developed by linearising the dynamic model of the previous chapter about a steady, level flight trim condition. The linearisation is carried out using small disturbance theory, the details and applicability of which are discussed in Section 3.1. The aircraft dynamic equations are then linearised about the trim condition in Section 3.2 and linear aerodynamic, gravitational and engine force and moment models are developed in Sections 3.3 to 3.5. Finally, the complete linear aircraft model is assembled in Section 3.6.

A linear aircraft model is desired because it allows full use to be made of the well developed theory and software for linear systems analysis. Linear analysis is capable of providing good insight into the aircraft's natural motions. This insight is used to assist in the design of an effective controller.

The linear aerodynamic model is presented in detail in this chapter since it forms the basis of the stability and control derivative software written during this project.

3.1 Small Disturbance Theory

Small disturbance theory is based on the absolute aircraft states being replaced by the sum of a constant trim value and a small deviation from the trim state, as illustrated by the axial velocity example below,

$$U = U_0 + u \tag{3.1}$$

With the absolute states replaced, the non-linear aircraft model of Chapter 2 can be linearised about the trim condition by assuming that,

1. Products of small perturbations are insignificantly small and can be ignored
2. The cosine of a small angle is unity
3. The sine of a small angle is merely the angle itself (in radians)

The use of small disturbance theory has been found to give good practical results. It predicts with satisfactory precision, the stability of unaccelerated flight as well as response calculations where the disturbances are not infinitesimal [6]. There are two reasons for the success of this method,

1. Often the major aerodynamic effects are nearly linear functions of the disturbances
2. Disturbances of considerable violence can occur with quite small values of linear and angular velocity

The trim states and small perturbation variables now need to be defined. Because stability axes are used, the OX_B-axis is attached to the aircraft and initially aligned with the trim velocity vector. Thus, if the aircraft is in steady level flight with a trim velocity of U_0, trim angle of attack of α_0 and trim elevator and throttle actuator settings of δ_{E_0} and δ_{T_0} respectively, then the trim condition can be written as,

$$\boldsymbol{V_B} = U_0\boldsymbol{i_B} \tag{3.2}$$

$$\boldsymbol{\omega_B} = \boldsymbol{0} \tag{3.3}$$

$$\boldsymbol{e} = \boldsymbol{0} \tag{3.4}$$

$$\boldsymbol{c} = (0, \delta_{E_0}, 0, \delta_{T_0}) \tag{3.5}$$

If the aircraft now experiences small disturbances in linear velocity, angular velocity, attitude and control deflection, denoted by (u, v, w), (p, q, r), (ψ, θ, ϕ) and $(\delta_a, \delta_e, \delta_r, \delta_t)$ respectively, then the aircraft states become,

$$\boldsymbol{V_B} = (U_0 + u)\boldsymbol{i_B} + v\boldsymbol{j_B} + w\boldsymbol{k_B} \tag{3.6}$$

$$\boldsymbol{\omega_B} = p\boldsymbol{i_B} + q\boldsymbol{j_B} + r\boldsymbol{k_B} \tag{3.7}$$

$$\boldsymbol{e} = (\phi, \theta, \psi) \tag{3.8}$$

$$\boldsymbol{c} = (\delta_a, \delta_{E_0} + \delta_e, \delta_r, \delta_{T_0} + \delta_t) \tag{3.9}$$

Also note, that under small perturbation assumptions, the angle of attack and angle of sideslip, defined in equations (2.6) and (2.7), become linearly related to the normal and lateral velocity perturations as shown below,

$$\alpha = \tan^{-1}\left(\frac{w}{U_0 + u}\right) \approx \frac{w}{U_0} \tag{3.10}$$

$$\beta = \sin^{-1}\left(\frac{v}{\sqrt{(U_0 + u)^2 + v^2 + w^2}}\right) \approx \frac{v}{U_0} \tag{3.11}$$

The aircraft dynamic equations as well as the force and moment equations can now be linearised about the trim condition using the small disturbance technique and the variables defined in this section.

3.2 Linearising the Equations of Motion

Substituting the components of $\boldsymbol{V_B}$, $\boldsymbol{\omega_B}$ and $\boldsymbol{e}$ from equations (3.6), (3.7) and (3.8) into equations (2.67) to (2.74) and applying the small perturbation linearisation assumptions, gives the linearised dynamic equations for the aircraft.

$$X_A + X_G + X_T = m(\dot{u}) \tag{3.12}$$

$$Y_A + Y_G + Y_T = m(\dot{v} + U_0 r) \tag{3.13}$$

$$Z_A + Z_G + Z_T = m(\dot{w} - U_0 q) \tag{3.14}$$

$$L_A + L_G + L_T = \dot{p}I_x - \dot{r}I_{xz} \tag{3.15}$$

$$M_A + M_G + M_T = \dot{q}I_y \tag{3.16}$$

$$N_A + N_G + N_T = \dot{r}I_z - \dot{p}I_{xz} \tag{3.17}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3.18}$$

$$\begin{bmatrix} \dot{P}_N \\ \dot{P}_E \\ \dot{P}_D \end{bmatrix} = \begin{bmatrix} U_0 + u \\ U_0\psi + v \\ -U_0\theta + w \end{bmatrix} \tag{3.19}$$

where, the aerodynamic, gravitational and engine forces and moments still need to be linearised about the trim condition.

3.3 Linear Aerodynamic Model

In this section, a linear aerodynamic model is built up by carrying out a Taylor series expansion of the aerodynamic forces of Section 2.4.1 about the trim condition. The coefficients of the linear terms in the Taylor series expansion are then examined in more detail and quantified in terms of the aircraft geometry and airfoil properties. The linear dynamic model developed is then used to extend the static airfoil coefficient models of Chapter 2.

3.3.1 Linearising the Aerodynamic Forces and Moments

While the aircraft flies in its trim condition, aerodynamic forces counter the gravitational and engine forces and the aircraft maintains equilibrium. When the aircraft is disturbed, the changes in aerodynamic forces can be very difficult to quantify. According to [1], these forces can be described by the sum of an infinite number of Taylor series expansions about the trim condition, where each series expansion involves the n^{th} time derivative of one motion or control variable where n moves from zero to infinity. However, because only small disturbances are being considered the Taylor series expansions can be limited to first order terms in each motion or control variable and its time derivative as shown in the aircraft axial force example below,

$$X_A = X_{A_0} + \left(\frac{\partial X_A}{\partial u}u + \frac{\partial X_A}{\partial \dot{u}}\dot{u}\right) + \left(\frac{\partial X_A}{\partial v}v + \frac{\partial X_A}{\partial \dot{v}}\dot{v}\right) + \left(\frac{\partial X_A}{\partial w}w + \frac{\partial X_A}{\partial \dot{w}}\dot{w}\right)$$

$$+\left(\frac{\partial X_A}{\partial p}p+\frac{\partial X_A}{\partial \dot{p}}\dot{p}\right)+\left(\frac{\partial X_A}{\partial q}q+\frac{\partial X_A}{\partial \dot{q}}\dot{q}\right)+\left(\frac{\partial X_A}{\partial r}r+\frac{\partial X_A}{\partial \dot{r}}\dot{r}\right)$$

$$+\left(\frac{\partial X_A}{\partial \delta_a}\delta_a+\frac{\partial X_A}{\partial \dot{\delta}_a}\dot{\delta}_a\right)+\left(\frac{\partial X_A}{\partial \delta_e}\delta_e+\frac{\partial X_A}{\partial \dot{\delta}_e}\dot{\delta}_e\right)+\left(\frac{\partial X_A}{\partial \delta_r}\delta_r+\frac{\partial X_A}{\partial \dot{\delta}_r}\dot{\delta}_r\right)(3.20)$$

where, X_{A_0} is the axial aerodynamic trim force that is a function of the trim states $(U_0,\alpha_0,\delta_{E_0},\delta_{T_0})$. For conventional aircraft, the only state derivative term of any significance is the $\dot{w}$ term [1] and thus equation (3.20) can be further simplified to,

$$X_A = X_{A_0}+\frac{\partial X_A}{\partial u}u+\frac{\partial X_A}{\partial v}v+\frac{\partial X_A}{\partial w}w+\frac{\partial X_A}{\partial p}p+\frac{\partial X_A}{\partial q}q+\frac{\partial X_A}{\partial r}r+\frac{\partial X_A}{\partial \dot{w}}\dot{w}$$

$$+\frac{\partial X_A}{\partial \delta_e}\,\delta_e+\frac{\partial X_A}{\partial \delta_a}\,\delta_a+\frac{\partial X_A}{\partial \delta_r}\,\delta_r \tag{3.21}$$

The partial derivatives on the top line of equation (3.21) are know as the *aerodynamic stability derivatives* while the partial derivatives on the bottom line are known as the *aerodynamic control derivatives.* They are called derivatives because they relate a change in motion variable to a change in force or moment. Since there are actually five more equations similar to (3.21), which can be found by replacing X with Y, Z, L, M and N, there are essentially 42 (6 × 7) aerodynamic stability derivatives and 18 (6 × 3) aerodynamic control derivatives that need to be calculated. However, the decoupling assumption in the aerodynamic model of Chapter 2, halves the number of required stability and control derivatives, since lateral derivatives with respect to longitudinal motion or control variables and longitudinal derivatives with respect to lateral motion or control variables are zero.

With the decoupling simplifications, the aerodynamic forces and moments as a result of motion and control variables reduce to,

$$X_A = X_{A_0}+X_{A_u}u+X_{A_w}w+X_{A_q}q+X_{A_{\dot{w}}}\dot{w}+X_{A_{\delta_e}}\delta_e \tag{3.22}$$

$$Y_A = Y_{A_0}+Y_{A_v}v+Y_{A_p}p+Y_{A_r}r+Y_{A_{\delta_a}}\delta_a+Y_{A_{\delta_r}}\delta_r \tag{3.23}$$

$$Z_A = Z_{A_0}+Z_{A_u}u+Z_{A_w}w+Z_{A_q}q+Z_{A_{\dot{w}}}\dot{w}+Z_{A_{\delta_e}}\delta_e \tag{3.24}$$

$$L_A = L_{A_0}+L_{A_v}v+L_{A_p}p+L_{A_r}r+L_{A_{\delta_a}}\delta_a+L_{A_{\delta_r}}\delta_r \tag{3.25}$$

$$M_A = M_{A_0}+M_{A_u}u+M_{A_w}w+M_{A_q}q+M_{A_{\dot{w}}}\dot{w}+M_{A_{\delta_e}}\delta_e \tag{3.26}$$

$$N_A = N_{A_0} + N_{A_v} v + N_{A_p} p + N_{A_r} r + N_{A_{\delta_a}} \delta_a + N_{A_{\delta_r}} \delta_r \tag{3.27}$$

where the following example indicates the partial derivative notation used,

$$\frac{\partial X_A}{\partial u} \equiv X_{A_u}$$

The aerodynamic model of the aircraft has now been reduced to a set of stability and control derivatives. These derivatives are commonly determined in four ways [1].

1. Calculation from first principles
2. Calculation using computational fluid dynamics methods
3. Wind tunnel measurement
4. Flight test measurement using system identification techniques

The calculation from first principles method was used in this project. This method involves using the aircraft's geometry and approximate airfoil aerodynamic models to calculate the stability and control derivatives. It provides the simplest solution to obtaining the derivatives and thus allows a model for an arbitrary conventional aircraft to be readily calculated. The method also provides good physical insight into the derivatives. The disadvantage of the method is that a model of limited accuracy is developed. However, this deficiency is accounted for in the controller design of Chapter 5.

Formulas for the aerodynamic stability and control derivatives of equations (3.22) to (3.27) are presented in the next two sections, where the subscript A has been dropped for clarity.

3.3.2 Longitudinal Stability and Control Derivatives

The calculation of each longitudinal stability and control derivative starts with the partial differentiation of equations (2.39), (2.41) and (2.43) with respect to the longitudinal motion or control variable of concern. The differentiation is taken about the aircraft trim condition where small perturbation assumptions are then applied.

A summary of the longitudinal stability and control derivatives is now provided, where they are grouped by perturbation of a motion or control variable. A brief discussion of the effects of each perturbation and the origin of the derivative is provided and any assumptions are stated. They are also displayed such that their dimensionless form is shown in square brackets as this provides maximum insight into each derivative. For a complete discussion of the longitudinal derivatives see [1].

Axial velocity derivatives

A change in axial velocity changes both lift and drag simply because of the airflow speed change over the aerodynamic surfaces. The angle of attack remains almost constant during this perturbation. Mach effects are ignored which means that $\frac{\partial C_L}{\partial u} = \frac{\partial C_D}{\partial u} = \frac{\partial C_m}{\partial u} = 0$.

$$X_u = \frac{1}{2}\rho U_0^2 S \left[-2C_D\right] \frac{1}{U_0} \tag{3.28}$$

$$Z_u = \frac{1}{2}\rho U_0^2 S \left[-2C_L\right] \frac{1}{U_0} \tag{3.29}$$

$$M_u = 0 \tag{3.30}$$

Normal velocity derivatives

As shown by equation (3.10), a small perturbation in normal velocity is essentially a small perturbation in angle of attack. This changes the lift, drag and pitching moment coefficients of the aircraft and produces the following derivatives.

$$X_w = \frac{1}{2}\rho U_0^2 S \left[C_L - \frac{\partial C_D}{\partial \alpha}\right] \frac{1}{U_0} \tag{3.31}$$

$$Z_w = \frac{1}{2}\rho U_0^2 S \left[-\Big(\frac{\partial C_L}{\partial \alpha} + C_D\Big)\right] \frac{1}{U_0} \tag{3.32}$$

$$M_w = \frac{1}{2}\rho U_0^2 S\bar{c} \left[\frac{\partial C_m}{\partial \alpha}\right] \frac{1}{U_0} \tag{3.33}$$

Pitch rate derivatives

Pitch rate derivatives are largely caused by the induced angle of incidence on the tailplane during a pitch rate perturbation. Under small perturbation assumptions, the induced tailplane incidence is given by $\Delta\alpha_T \approx \frac{ql_T}{U_0}$. Since the tailplane is modeled as dragless, only a lift force is generated which produces an aircraft normal force and a pitching moment that provides the natural *pitch rate damping* of the aircraft.

$$X_q = 0 \tag{3.34}$$

$$Z_q = \frac{1}{2}\rho U_0^2 S \left[-\frac{S_T l_T}{S\bar{c}} \cdot \frac{\partial C_{L_T}}{\partial \alpha}\right] \frac{\bar{c}}{U_0} \tag{3.35}$$

$$M_q = \frac{1}{2}\rho U_0^2 S\bar{c} \left[-\frac{S_T l_T^2}{S\bar{c}^2} \cdot \frac{\partial C_{L_T}}{\partial \alpha}\right] \frac{\bar{c}}{U_0} \tag{3.36}$$

Normal acceleration derivatives

Derivatives due to acceleration perturbations quantify the extra mass effect that is experienced when objects move through the air: dragging an air mass along with them. Usually, this extra mass is negligible compared to the mass of the aircraft [1]. However, the derivative due to the rate of change of normal velocity perturbation can sometimes be quite significant, not because of the mass effect, but because of a phenomenon called *downwash lag.* When the wing experiences a normal acceleration perturbation, a transient disturbance is experienced in the downwash field that passes over the tailplane a while later. This change in downwash field results in a transient change in tailplane incidence, which under small perturbation assumptions, can be quantified by $\Delta\alpha_T \approx \frac{d\epsilon}{d\alpha}\frac{\dot{w} l_T}{U_0^2}$. Again, because the tailplane is dragless, only a normal force and pitching moment are produced.

$$X_{\dot{w}} = 0 \tag{3.37}$$

$$Z_{\dot{w}} = \frac{1}{2}\rho U_0^2 S \left[-\frac{S_T l_T}{S\bar{c}} \cdot \frac{\partial C_{L_T}}{\partial \alpha} \cdot \frac{d\epsilon}{d\alpha}\right] \frac{\bar{c}}{U_0^2} \tag{3.38}$$

$$M_{\dot{w}} = \frac{1}{2}\rho U_0^2 S\bar{c} \left[-\frac{S_T l_T^2}{S\bar{c}^2} \cdot \frac{\partial C_{L_T}}{\partial \alpha} \cdot \frac{d\epsilon}{d\alpha}\right] \frac{\bar{c}}{U_0^2} \tag{3.39}$$

Elevator derivatives

An elevator disturbance essentially changes the camber of the tailplane which also changes the tailplane angle of incidence. The net result is a change in the tailplane lift, which produces an aircraft normal force and a corresponding pitching moment.

$$X_{\delta_e} = 0 \tag{3.40}$$

$$Z_{\delta_e} = \frac{1}{2}\rho U_0^2 S \left[-\frac{S_T}{S} \cdot \frac{\partial C_{L_T}}{\partial \delta_e}\right] \tag{3.41}$$

[1] except in airships and balloons

$$M_{\delta_e} = \frac{1}{2}\rho U_0^2 S\bar{c}\left[-\frac{S_T l_T}{S\bar{c}} \cdot \frac{\partial C_{L_T}}{\partial \delta_e}\right] \tag{3.42}$$

3.3.3 Lateral Stability and Control Derivatives

The calculation of each lateral stability and control derivative starts with the partial differentiation of equations (2.40), (2.42) and (2.44) with respect to the lateral motion or control variable of concern. The differentiation is taken about the aircraft trim condition where small perturbation assumptions are then applied.

A summary of the lateral stability and control derivatives is now provided in a similar fashion to the longitudinal derivatives. For a complete discussion of the lateral derivatives see [1].

Sideslip derivatives

During a sideslip, the airflow induces an angle of incidence on the fin which creates a lateral force and a yawing moment known as the *weather cock effect.* The fin can also cause a significant roll moment because of the offset in its aerodynamic centre from the OX_B-axis. However, the roll moment is predominantly caused by the dihedral and quarter chord sweep back effects of the wing. If the wing has dihedral, then lateral velocity induces a differential angle of incidence across the wing which produces a roll moment that tends to restore a wings level flight condition. If the quarter chord line of the wing is swept back, then a sideslip velocity also produces a restoring roll moment, because the lift of a subsonic wing is a function of the velocity perpendicular to this line. The differential lift that causes these roll moments, also comes accompanied with differential drag, which will induce a yaw moment. However, the yaw moment is usually a lot smaller than that caused by the weather cock effect and is thus ignored.

$$Y_v = \frac{1}{2}\rho U_0^2 S\left[-\frac{S_F}{S} \cdot \frac{\partial C_{L_F}}{\partial \beta}\right]\frac{1}{U_0} \tag{3.43}$$

$$L_v = \frac{1}{2}\rho U_0^2 Sb\left[-\frac{2\int_0^s c_y y dy}{Sb} \cdot \left(2C_{L_W}\tan\Lambda_{\frac{1}{4}} + \Gamma\frac{\partial C_{L_W}}{\partial \alpha}\right) - \frac{S_F h_F}{Sb} \cdot \frac{\partial C_{L_F}}{\partial \beta}\right]\frac{1}{U_0} \tag{3.44}$$

$$N_v = \frac{1}{2}\rho U_0^2 Sb\left[\frac{S_F l_F}{Sb} \cdot \frac{\partial C_{L_F}}{\partial \beta}\right]\frac{1}{U_0} \tag{3.45}$$

Roll rate derivatives

A roll rate disturbance induces an increase in angle of incidence on the down going wing and a decrease in angle of incidence on the up going wing. The differential incidence across the wing is by far the major contributor to the roll and yaw moments during a roll rate perturbation, and provides the natural *roll damping* of the aircraft. Sideforce due to roll rate is similarly caused by the induced angle of incidence as the fin rolls through the airflow. However, this effect is usually negligible and is thus not modeled.

$$Y_p = 0 \tag{3.46}$$

$$L_p = \frac{1}{2}\rho U_0^2 Sb\left[-\frac{2\int_0^s c_y y^2 dy}{Sb^2}\cdot\left(\frac{\partial C_{L_W}}{\partial\alpha}+C_{D_W}\right)\right]\frac{b}{U_0} \tag{3.47}$$

$$N_p = \frac{1}{2}\rho U_0^2 Sb\left[-\frac{2\int_0^s c_y y^2 dy}{Sb^2}\cdot\left(C_{L_W}-\frac{\partial C_{D_W}}{\partial\alpha}\right)\right]\frac{b}{U_0} \tag{3.48}$$

Yaw rate derivatives

A yaw rate disturbance increases the airflow velocity over the forward moving wing and decreases the velocity over the aft moving wing. This causes differential lift and drag across the wings which causes roll and yaw moments respectively. The disturbance also gives rise to an induced angle of incidence ($\Delta\alpha_F \approx \frac{rl_F}{U_0}$) on the fin which provides the aircraft's natural *yaw damping* as well as a side force and a roll moment.

$$Y_r = \frac{1}{2}\rho U_0^2 S\left[\frac{S_F l_F}{Sb}\cdot\frac{\partial C_{L_F}}{\partial\beta}\right]\frac{b}{U_0} \tag{3.49}$$

$$L_r = \frac{1}{2}\rho U_0^2 Sb\left[\frac{4\int_0^s c_y y^2 dy}{Sb^2}\cdot C_{L_W}+\frac{S_F l_F h_F}{Sb^2}\cdot\frac{\partial C_{L_F}}{\partial\beta}\right]\frac{b}{U_0} \tag{3.50}$$

$$N_r = \frac{1}{2}\rho U_0^2 Sb\left[-\left(\frac{4\int_0^s c_y y^2 dy}{Sb^2}\cdot C_{D_W}+\frac{S_F l_F^2}{Sb^2}\cdot\frac{\partial C_{L_F}}{\partial\beta}\right)\right]\frac{b}{U_0} \tag{3.51}$$

Aileron derivatives

Ailerons work by differentially changing the camber and angle of incidence across the wing. This causes differential lift and drag over the wing span and as a result, roll and yaw moments. The sideforce is considered negligible and thus not modeled. The yaw moment due to aileron deflection is usually positive which means that the aircraft will tend to yaw out of a turn at its onset. This phenomenon is know as *adverse yaw.* For more detail on the lateral control derivatives see [12].

$$Y_{\delta_a} = 0 \tag{3.52}$$

$$L_{\delta_a} = \frac{1}{2}\rho U_0^2 S b \left[\frac{\partial C_{l_W}}{\partial \delta_a}\right] \tag{3.53}$$

$$N_{\delta_a} = \frac{1}{2}\rho U_0^2 S b \left[\frac{\partial C_{n_W}}{\partial \delta_a}\right] \tag{3.54}$$

Rudder derivatives

Rudder deflections change the camber of the fin, which also changes the fin angle of incidence. The net result is a change in the fin lift force. This produces an aircraft sideforce and a corresponding yaw moment, typically accompanied by a small roll moment.

$$Y_{\delta_r} = \frac{1}{2}\rho U_0^2 S \left[\frac{S_F}{S} \cdot \frac{\partial C_{L_F}}{\partial \delta_r}\right] \tag{3.55}$$

$$L_{\delta_r} = \frac{1}{2}\rho U_0^2 S b \left[\frac{S_F h_F}{Sb} \cdot \frac{\partial C_{L_F}}{\partial \delta_r}\right] \tag{3.56}$$

$$N_{\delta_r} = \frac{1}{2}\rho U_0^2 S b \left[-\frac{S_F l_F}{Sb} \cdot \frac{\partial C_{L_F}}{\partial \delta_r}\right] \tag{3.57}$$

3.3.4 Airfoil Aerodynamic Coefficients

The stability derivative analysis has provided insight into the aerodynamic forces and moments produced as a result of the aircraft motion variables. These force and moment terms are now added to the airfoil coefficient models of Chapter 2, to extend their applicability from static, symmetric flow conditions to full dynamic conditions. The extended longitudinal and lateral airfoil coefficient models are listed below.

Longitudinal Airfoil Coefficients

$$C_{L_W} = f_W(\alpha_0 + \alpha) \tag{3.58}$$

$$C_{D_W} = C_{D_{W_0}} + k_W C_{L_W}^2 \tag{3.59}$$

$$C_{L_T} = \frac{\partial C_{L_T}}{\partial \alpha}\left[\alpha_T + \left(1 - \frac{d\epsilon}{d\alpha}\right)(\alpha_0 + \alpha) + \frac{Ql_T}{\bar{V}} + \frac{\dot{\alpha} l_T}{\bar{V}}\frac{d\epsilon}{d\alpha}\right] + \frac{\partial C_{L_T}}{\partial \delta_E}\delta_E \tag{3.60}$$

$$C_{D_B} = C_{D_{B_0}} \tag{3.61}$$

where, linear terms that describe the induced angle of attack due to pitch rate and downwash lag have been added to the tailplane coefficient model.

Lateral Airfoil Coefficients

$$C_{L_F} = \frac{\partial C_{L_F}}{\partial \beta}\left(\beta + \frac{Rl_T}{\bar{V}}\right) + \frac{\partial C_{L_F}}{\partial \delta_R}\delta_R \tag{3.62}$$

$$\begin{aligned} C_{l_W} = -2\int_0^s \Bigg[\left(2C_{L_W}\tan\Lambda_{\frac{1}{4}} + \Gamma\frac{\partial C_{L_W}}{\partial \alpha}\right)\beta + \\ \left(\frac{\partial C_{L_W}}{\partial \alpha} + C_{D_W}\right)\frac{Py}{\bar{V}} - 2C_{L_W}\frac{Ry}{\bar{V}}\Bigg]\frac{c_y y dy}{Sb} + \frac{\partial C_{l_W}}{\partial \delta_A}\delta_A \end{aligned} \tag{3.63}$$

$$C_{n_W} = -2\int_0^s \left[\left(C_{L_W} - \frac{\partial C_{D_W}}{\partial \alpha}\right)\frac{Py}{\bar{V}} + 2C_{D_W}\frac{Ry}{\bar{V}}\right]\frac{c_y y dy}{Sb} + \frac{\partial C_{n_W}}{\partial \delta_A}\delta_A \tag{3.64}$$

where, linear terms that describe the induced angle of incidence on the fin during sideslip and yaw rate motions have been added as well as the change in fin lift coefficient due to the rudder deflection. Linear terms that describe the wing's roll and yaw moments due to sideslip, roll rate and yaw rate motions as well as the aileron deflection, have also been included.

Summary

The longitudinal and lateral airfoil coefficient models of equations (3.58) to (3.64), complete the aerodynamic model of Chapter 2. The model is consistent with the stability and control derivatives displayed in Sections 3.3.2 and 3.3.3, in that the derivative equations can be calculated by the partial differentiation of the respective aerodynamic model equations. The completed coefficient models now allow the simulation of the aircraft's entire airborne flight envelope, including stall effects. The two non-linear simulators described in Chapter 6 are based on this aircraft model.

The final problem that remains, is to quantify the following,

- The wing, tailplane and fin lift curve slopes as a function of their angles of incidence
- The tailplane and fin lift curve slopes due to their respective actuator deflections
- The wing roll and yaw moment change with aileron deflection
- The wing and fuselage parasitic drag coefficients
- The wing induced drag coefficient and downwash variation with angle of attack

These aerodynamic quantities were calculated using the empirical methods summarised in Appendix B.

3.4 Linear Gravitational Model

The linearisation of the gravitational model is very straight forward. Substituting the trim condition into equation (2.61) and applying small perturbation assumptions, the linearised gravitational model becomes,

$$\begin{bmatrix} X_G \\ Y_G \\ Z_G \end{bmatrix} = \begin{bmatrix} -mg\theta \\ mg\phi \\ mg \end{bmatrix} \tag{3.65}$$

$$L_G = M_G = N_G = 0 \tag{3.66}$$

3.5 Linear Engine Model

Proceeding in a similar fashion to the formulation of the linear aerodynamic model, the engine thrust formula can be written as the sum of a truncated Taylor series, in the aircraft motion and control variables, about the trim condition. However, the only Taylor series expansion of any significance, other than the throttle actuation variable expansion, is that in axial velocity perturbation. This is because the engine propeller is essentially a rotating airfoil and the engine thrust is produced by the lift force of this airfoil. As the aircraft changes speed through the air, so the effective angle of attack over the propeller blades changes, and as a result so does the engine thrust. Thus, the linear engine thrust model reduces to,

$$T = T_0 + \frac{\partial T}{\partial u}u + \frac{\partial T}{\partial \delta_t}\delta_t \tag{3.67}$$

where, T_0 is the engine trim thrust that is a function of the trim states $(U_0,\alpha_0,\delta_{E_0},\delta_{T_0})$, $\frac{\partial T}{\partial u}$ is the *engine velocity derivative* and $\frac{\partial T}{\partial \delta_t}$ is the *engine control derivative.* Note however, that if the units of δ_t and T are set the same, then the engine control derivative reduces to unity. Substition of equation (3.67) into equations (2.63) to (2.65) gives,

$$X_T = (T_0 + T_u u + \delta_t)\cos\epsilon_T \tag{3.68}$$

$$Z_T = -(T_0 + T_u u + \delta_t)\sin\epsilon_T \tag{3.69}$$

$$M_T = (T_0 + T_u u + \delta_t)m_T \tag{3.70}$$

$$Y_T = L_T = N_T = 0 \tag{3.71}$$

where the partial derivative notation of Section 3.3.1 is used. To write equations (3.68) to (3.71) in a more concise form, the $\cos\epsilon_T$, $\sin\epsilon_T$ and m_T terms are combined with the thrust variables as follows,

$$X_T = X_{T_0} + X_{T_u}u + X_{T_{\delta_t}}\delta_t \tag{3.72}$$

$$Z_T = Z_{T_0} + Z_{T_u}u + Z_{T_{\delta_t}}\delta_t \tag{3.73}$$

$$M_T = M_{T_0} + M_{T_u}u + M_{T_{\delta_t}}\delta_t \tag{3.74}$$

$$Y_T = L_T = N_T = 0 \tag{3.75}$$

where, as an example,

$$X_{T_0} = T_0\cos\epsilon_T \tag{3.76}$$

The engine velocity derivative can be calculated experimentally. However, a good approximation can be obtained by considering the engine to obey a constant power formula,

$$T = \frac{P_T}{V} \tag{3.77}$$

where, P_T is the engine power. Note that if the engine power setting remains constant, then an increase in velocity results in a decrease in thrust and vice versa. The engine velocity derivative can be found by differentiating equation (3.77) with respect to the aircraft axial velocity at the trim condition to give,

$$\frac{\partial T}{\partial u} = -\frac{P_{T_0}}{U_0^2} \tag{3.78}$$

where $P_{T_0} = T_0 U_0$ is the corresponding trim power setting. Thus, the engine velocity derivative reduces to,

$$\frac{\partial T}{\partial u} = -\frac{T_0}{U_0} \tag{3.79}$$

and can be calculated as a function of the trim thrust and trim velocity.

To complete the engine model, the thrust actuation lag must be taken into account. Unlike the aerodynamic actuators, the thrust actuation can display a significant delay from commanded thrust to output thrust. If the dynamics of the delay are not taken into consideration, they could cause serious degradation in the autopilot performance or even instability. The delay is well modeled by a first order differential equation of the form,

$$\dot{\delta}_t = -\frac{1}{\tau_T}\delta_t + \frac{1}{\tau_T}\delta_{t_c} \tag{3.80}$$

where, δ_{t_c} is the commanded throttle perturbation and τ_T is the *engine time constant* which can be determined by experiment.

3.6 Linear Aircraft Model

The linearised aircraft dynamic equations of Section 3.2 are now combined with the linear aerodynamic, gravitational and engine models of Sections 3.3, 3.4 and 3.5 to form a complete linear aircraft model valid about a steady, level flight trim condition. Substituting the respective force and moment terms into equations (3.12) to (3.17) gives,

$$X_{A_u}u + X_{A_w}w + X_{A_q}q + X_{A_{\dot{w}}}\dot{w} + X_{A_{\delta_e}}\delta_e - mg\theta + X_{T_u}u + X_{T_{\delta_t}}\delta_t = m\dot{u} \tag{3.81}$$

$$Y_{A_v}v + Y_{A_p}p + Y_{A_r}r + Y_{A_{\delta_a}}\delta_a + Y_{A_{\delta_r}}\delta_r + mg\phi = m(\dot{v} + U_0 r) \tag{3.82}$$

$$Z_{A_u}u + Z_{A_w}w + Z_{A_q}q + Z_{A_{\dot{w}}}\dot{w} + Z_{A_{\delta_e}}\delta_e + Z_{T_u}u + Z_{T_{\delta_t}}\delta_t = m(\dot{w} - U_0 q) \tag{3.83}$$

$$L_{A_v}v + L_{A_p}p + L_{A_r}r + L_{A_{\delta_a}}\delta_a + L_{A_{\delta_r}}\delta_r = \dot{p}I_x - \dot{r}I_{xz} \tag{3.84}$$

$$M_{A_u}u + M_{A_w}w + M_{A_q}q + M_{A_{\dot{w}}}\dot{w} + M_{A_{\delta_e}}\delta_e + M_{T_u}u + M_{T_{\delta_t}}\delta_t = \dot{q}I_y \tag{3.85}$$

$$N_{A_v}v + N_{A_p}p + N_{A_r}r + N_{A_{\delta_a}}\delta_a + N_{A_{\delta_r}}\delta_r = \dot{r}I_z - \dot{p}I_{xz} \tag{3.86}$$

where, the trim aerodynamic, gravitational and engine forces and moments cancel in the steady state to give the trim condition equations,

$$X_{A_0} = -X_{T_0} \tag{3.87}$$

$$Z_{A_0} = -(Z_{T_0} + mg) \tag{3.88}$$

$$M_{A_0} = -M_{T_0} \tag{3.89}$$

$$Y_{A_0} = L_{A_0} = N_{A_0} = 0 \tag{3.90}$$

Equations (3.87), (3.88) and (3.89) form a set of three equations in the four trim variables (U_0,α_0,δ_{E_0},δ_{T_0}) which, for a completely linear model, means that only a single trim variable need be defined to fully define a level flight trim state.

Equations (3.81) to (3.86) are now grouped with equations (3.18), (3.19) and (3.80) and rearranged into a longitudinal and lateral set. The states appearing in equations (3.81) to (3.86) are presented one step before pure state space form, with a *mass matrix* ($\boldsymbol{M}$) on the left hand side. This form provides a clear, compact viewing structure for the equations and is easily converted to state space form with a single matrix inversion. The state space form is desired because it is convenient for computer aided linear analysis.

Longitudinal Dynamic Equations

$$\boldsymbol{M_l}\dot{\mathbf{x}}_l = \boldsymbol{A'_l}\mathbf{x}_l + \boldsymbol{B'_l}\mathbf{u}_l \tag{3.91}$$

$$\dot{\delta}_t = -\frac{1}{\tau_T}\delta_t + \frac{1}{\tau_T}\delta_{t_c} \tag{3.92}$$

$$\dot{h} = -\dot{P}_D = U_0\theta - w \tag{3.93}$$

$$\dot{p}_N = \dot{P}_N - U_0 = u \tag{3.94}$$

where,

$$\mathbf{x_l} = \begin{bmatrix} u & w & q & \theta \end{bmatrix}^T \tag{3.95}$$

$$\mathbf{u_l} = \begin{bmatrix} \delta_e & \delta_t \end{bmatrix}^T \tag{3.96}$$

$$\boldsymbol{M_l} = \begin{bmatrix} m & -X_{\dot{w}} & 0 & 0 \\ 0 & (m - Z_{\dot{w}}) & 0 & 0 \\ 0 & -M_{\dot{w}} & I_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.97}$$

$$\boldsymbol{A_l'} = \begin{bmatrix} (X_{A_u} + X_{T_u}) & X_{A_w} & X_{A_q} & -mg \\ (Z_{A_u} + Z_{T_u}) & Z_{A_w} & (Z_{A_q} + mU_0) & 0 \\ (M_{A_u} + M_{T_u}) & M_{A_w} & M_{A_q} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.98}$$

$$\boldsymbol{B_l'} = \begin{bmatrix} X_{A_{\delta_e}} & X_{T_{\delta_t}} \\ Z_{A_{\delta_e}} & Z_{T_{\delta_t}} \\ M_{A_{\delta_e}} & M_{T_{\delta_t}} \\ 0 & 0 \end{bmatrix} \tag{3.99}$$

and h is the height state and p_N is the longitudinal position error state. The longitudinal state space vector ($\mathbf{x_l}$) is only made up of the core longitudinal states. The throttle lag, height and longitudinal position error dynamic equations are written separately so as not to cloud the fundamental longitudinal aircraft model.

Lateral Dynamic Equations

$$\boldsymbol{M_t}\dot{\mathbf{x}}_{\boldsymbol{t}} = \boldsymbol{A_t'}\mathbf{x_t} + \boldsymbol{B_t'}\mathbf{u_t} \tag{3.100}$$

$$\dot{\psi} = r \tag{3.101}$$

$$y = \dot{P}_E = U_0\psi + v \tag{3.102}$$

where,

$$\mathbf{x}_{\boldsymbol{t}} = \begin{bmatrix} v & p & r & \phi \end{bmatrix}^T \tag{3.103}$$

$$\mathbf{u}_{\boldsymbol{t}} = \begin{bmatrix} \delta_a & \delta_r \end{bmatrix}^T \tag{3.104}$$

$$\boldsymbol{M_t} = \begin{bmatrix} m & 0 & 0 & 0 \\ 0 & I_x & -I_{xz} & 0 \\ 0 & -I_{xz} & I_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.105}$$

$$\boldsymbol{A'_t} = \begin{bmatrix} Y_{A_v} & Y_{A_p} & (Y_{A_r} - mU_0) & mg \\ L_{A_v} & L_{A_p} & L_{A_r} & 0 \\ N_{A_v} & N_{A_p} & N_{A_r} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.106}$$

$$\boldsymbol{B'_t} = \begin{bmatrix} Y_{A_{\delta_a}} & Y_{A_{\delta_r}} \\ L_{A_{\delta_a}} & L_{A_{\delta_r}} \\ N_{A_{\delta_a}} & N_{A_{\delta_r}} \\ 0 & 0 \end{bmatrix} \tag{3.107}$$

and y is the cross track error state. The lateral state space vector ($\mathbf{x}_{\boldsymbol{t}}$) is only made up of the core lateral states. The heading and cross track error dynamic equations are written separately so as not to cloud the fundamental lateral aircraft model.

3.7 Summary

In this chapter a linear aircraft model about a level flight trim condition was developed. Small disturbance theory was used to linearise the aircraft dynamic equations as well as the force and moment equations. Finally, the model was decoupled into longitudinal and lateral systems and presented in a form one matrix inversion away from state space form.

The model developed up to this point is quite general and makes only a few assumptions as to what the aircraft should look like.

Chapter 4

Aircraft Natural Motions

The focus is now shifted to the analysis of the linear model specific to the aircraft used in this project - the Reliance 0.46 size trainer shown in Figure 4.1. To automate the process of calculating the linear model for an arbitrary aircraft, a MATLAB program was written that accepts an aircraft's physical parameters and trim condition from an Excel file and then calculates the desired longitudinal and lateral models. All of the Reliance data delivered by this program for a trim condition of 20m/s is summarised in Appendix C.

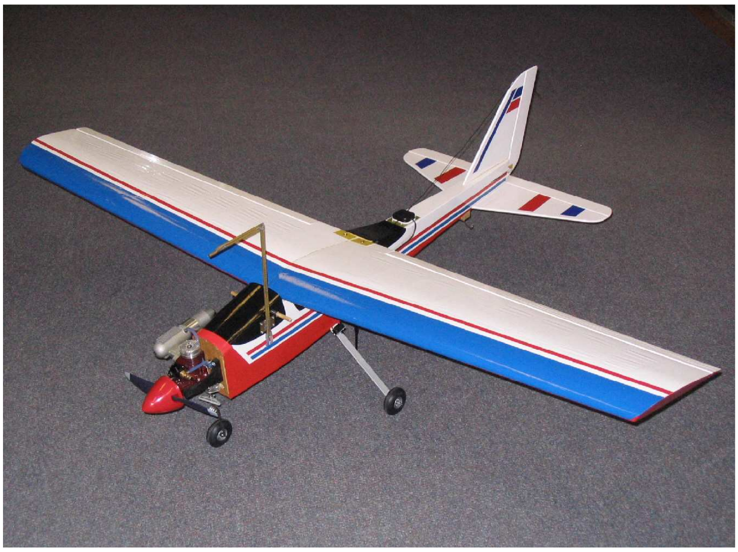


Figure 4.1: *Reliance 0.46 Size Trainer Model Aircraft*

In this chapter, both the longitudinal and lateral aircraft models, specific to the Reliance aircraft, are presented and analysed in detail using modal decomposition techniques. Modal decomposition involves using an eigenvector transformation matrix to convert the current system model to modal form. This representation provides maximum insight into the aircraft's natural modes of motion. A brief summary of the theory of modal decomposition is presented in Appendix D. The information gained in this chapter is used to verify the aircraft model developed as well as to assist in the design of an effective controller.

4.1 Longitudinal Aircraft Model

Figure 4.2 shows a block diagram representation of the longitudinal aircraft model described by equations (3.91) to (3.94).

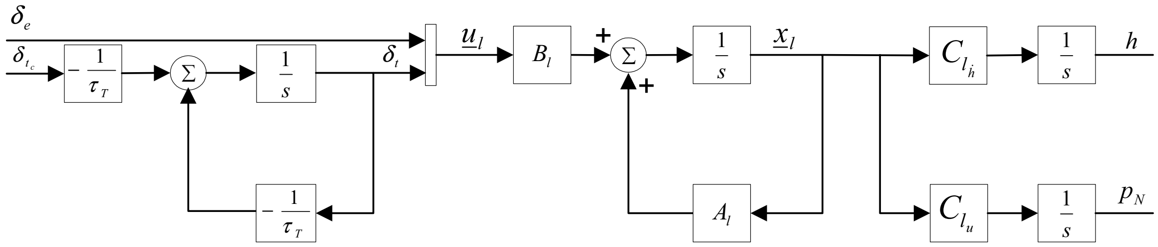


Figure 4.2: *Longitudinal Aircraft Model*

The longitudinal model can be written as,

$$\dot{\mathbf{x}}_{\boldsymbol{l}} = \boldsymbol{A_l}\mathbf{x}_{\boldsymbol{l}} + \boldsymbol{B_l}\mathbf{u}_{\boldsymbol{l}} \tag{4.1}$$

$$\dot{h} = \boldsymbol{C}_{\boldsymbol{l}_{\dot{h}}}\mathbf{x}_{\boldsymbol{l}} \tag{4.2}$$

$$\dot{p}_N = \boldsymbol{C}_{\boldsymbol{l_u}}\mathbf{x}_{\boldsymbol{l}} \tag{4.3}$$

$$\dot{\delta}_t = -\frac{1}{\tau_T}\delta_t + \frac{1}{\tau_T}\delta_{t_c} \tag{4.4}$$

where,

$$\boldsymbol{A_l} = \boldsymbol{M_l}^{-1}\boldsymbol{A'_l} = \begin{bmatrix} -0.15 & 0.23 & 0 & -0.17 \\ -0.97 & -12.13 & 3.49 & 0 \\ 0 & -45.56 & -11.18 & 0 \\ 0 & 0 & 10 & 0 \end{bmatrix} \tag{4.5}$$

$$\boldsymbol{B_l} = \boldsymbol{M_l}^{-1}\boldsymbol{B'_l} = \begin{bmatrix} 0 & 1 \\ -0.43 & 0 \\ -24.09 & 0 \\ 0 & 0 \end{bmatrix} \tag{4.6}$$

$$\boldsymbol{C}_{\boldsymbol{l}_{\dot{h}}} = \begin{bmatrix} 0 & -1 & 0 & \frac{20\pi}{180} \end{bmatrix} \tag{4.7}$$

$$\boldsymbol{C}_{\boldsymbol{l}_{\boldsymbol{u}}} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \tag{4.8}$$

$$\tau_T = 0.5 \tag{4.9}$$

The state vector ($\mathbf{x}_{\boldsymbol{l}}$) and control input vector ($\mathbf{u}_{\boldsymbol{l}}$) are defined in equations (3.95) and (3.96) respectively. The height, longitudinal position error and throttle states have not been augmented to the longitudinal state space system of equation (4.1), because their dynamics are trivial and only cloud the analysis of the core longitudinal dynamics.

The units of u and w are in metres per second (m/s), h and p_N in metres (m), q in tens of degrees per second (10deg/s), θ and δ_e in degrees (deg) and δ_t and δ_{t_c} in specific force (m/s^2). The units were chosen such that one unit of each variable has roughly the same physical significance to the aircraft motion [3]. There are a few approaches to choosing units that produce well scaled variables. A *maximum deviation* approach was adopted in this project. This approach involves comparing the maximum deviations of each motion and control variable from the trim condition, and then setting the units of each variable such that their maximum deviations are of a similar magnitude.

The maximum deviations of the variables in the longitudinal state vector are discussed in Table 4.1 and the maximum allowed control deflections are listed in Table 4.2. The tables make it clear that, with the pitch rate variable unit set to tens of degrees per second, each variable has roughly the same physical significance to the aircraft motion.

Var	**Description**	**Max**	**Unit**
u	With a trim velocity of 20m/s, it is estimated that the maximum deviation from this trim condition will be approximately 10m/s.	10	m/s
w	At a trim velocity of 20m/s, a normal velocity disturbance of 5m/s will produce an angle of incidence of approximately 15deg and stall the aircraft.	5	m/s
q	The maximum pitch rate disturbance during conventional flight is estimated as 60deg/s	6	10deg/s
θ	The maximum pitch angle deviation during conventional flight is estimated to be 15deg	15	deg

Table 4.1: *Longitudinal State Vector Maximum Deviations*

Var	Description	Max	Unit
δ_e	The maximum elevator deflection is limited to 10deg	10	deg
δ_t	From engine test data, the maximum specific force deviation is set at 5m/s^2	5	m/s^2

Table 4.2: *Longitudinal Control Vector Maximum Deflections*

With the units of the longitudinal state space variables handled, the focus turns to gaining insight into the natural motions of the longitudinal state space system. This is done using modal decomposition techniques. The modal transformation matrix that converts the system of equation (4.1) to modal form is shown below,

$$\begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} = \begin{bmatrix} -0.0023 & -0.0034 & -0.0574 & 0.3302 \\ 0.0106 & -0.2767 & 0.0022 & -0.0122 \\ 1.0000 & 0.0000 & -0.0067 & 0.0503 \\ -0.3954 & -0.4275 & 1.0000 & 0.0000 \end{bmatrix} \begin{bmatrix} z_{l_1} \\ z_{l_2} \\ z_{l_3} \\ z_{l_4} \end{bmatrix} \tag{4.10}$$

where the original state vector is shown on the left and the corresponding modal state vector on the right. The modal transformation matrix has the real and imaginary parts of each eigenvector as its columns (see Appendix D). Transforming the longitudinal state space model of equation (4.1) reveals,

$$\dot{\mathbf{z}}_l = \boldsymbol{F}_l \mathbf{z}_l + \boldsymbol{G}\mathbf{u}_l \tag{4.11}$$

where,

$$\mathbf{z}_l = \begin{bmatrix} z_{l_1} & z_{l_2} & z_{l_3} & z_{l_4} \end{bmatrix}^T \tag{4.12}$$

$$\boldsymbol{F}_l = \begin{bmatrix} -11.66 & 12.61 & 0 & 0 \\ -12.61 & -11.66 & 0 & 0 \\ 0 & 0 & -0.067 & 0.503 \\ 0 & 0 & -0.503 & -0.067 \end{bmatrix} \tag{4.13}$$

$$\boldsymbol{G}_l = \begin{bmatrix} -24.06 & -0.15 \\ 0.65 & -0.14 \\ -9.23 & -0.12 \\ -1.77 & 3.01 \end{bmatrix} \tag{4.14}$$

The longitudinal system matrix in modal form ($\boldsymbol{F_l}$), clearly shows the two sets of complex poles that define the natural modes of the primary longitudinal system. The higher frequency pole set is known as the *short period* mode while the lower frequency set is know as the *phugoid* mode. Each of these modes is now discussed in more detail with specific focus on the mode 'shapes' and the relative control of the longitudinal actuators over each mode.

4.1.1 Phugoid Mode

The phugoid mode poles are summarised below,

$$\lambda = -0.067 \pm 0.503i \tag{4.15}$$

$$\zeta = 0.13 \tag{4.16}$$

$$\omega_n = 0.51 \tag{4.17}$$

These poles form a stable, low frequency, lightly damped oscillatory mode. The period of oscillation is about 12.5s with an exponential decay time constant of around 15s.

The eigenvector corresponding to the phugoid mode is displayed in phasor form below,

$$\begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} = \begin{bmatrix} 0.3351 \angle 99.85 \\ 0.0124 \angle -79.83 \\ 0.0508 \angle 97.57 \\ 1.0000 \angle 0.00 \end{bmatrix} \tag{4.18}$$

where the phasor angles are shown in degrees. The eigenvector indicates the magnitude and phase contributions of each element of the original state vector ($\mathbf{x_l}$) to the specific mode of motion. It is clear that the mode is dominated by pitch angle and axial velocity perturbations with very small changes in pitch rate and normal velocity. Under small disturbance assumptions, the low magnitude of the normal velocity is equivalent to observing very little angle of attack change. This small angle of attack change is a key feature of the phugoid mode [1].

The phugoid mode can be visualised as a sinusoidal exchange of the aircraft's potential and kinetic energy, damped by the aerodynamic drag. Consider a disturbance that leaves the aircraft with a negative pitch angle, flying slightly downwards. As the aircraft moves downwards it builds speed, which at a constant angle of attack, translates directly into an increase in lift. The eigenvector shows that the change in airspeed leads the change in pitch angle by approximately 90° which means that the speed is a maximum around the lowest point of the motion. This kinetic energy then carries the aircraft up to the next peak and the motion continues in a sinusoidal fashion.

Interestingly, if the phugoid motion is assumed to be one of constant total energy (only gravitational forces acting on the aircraft), then the phugoid period can be calculated, to within a reasonable degree of accuracy, as a function of the aircraft velocity alone, and thus independently of the aircraft itself [6]! Using this approximation equation at a trim condition of 20m/s reveals a phugoid period of 9.1s (vs the 12.5s calculated using the linear model).

The axial velocity stability derivatives have the largest effect on the phugoid mode characteristics, with X_u primarily determining the damping and Z_u primarily determining the natural frequency [4].

A time history response of pure phugoid mode motion, obtained by initialising the system state vector ($\mathbf{x}_{\boldsymbol{l}}$) with the real parts of the phugoid mode eigenvector [3], is plotted in Figure 4.3. It is clear from the plot that the phugoid mode motion is well observed in axial velocity and pitch angle.

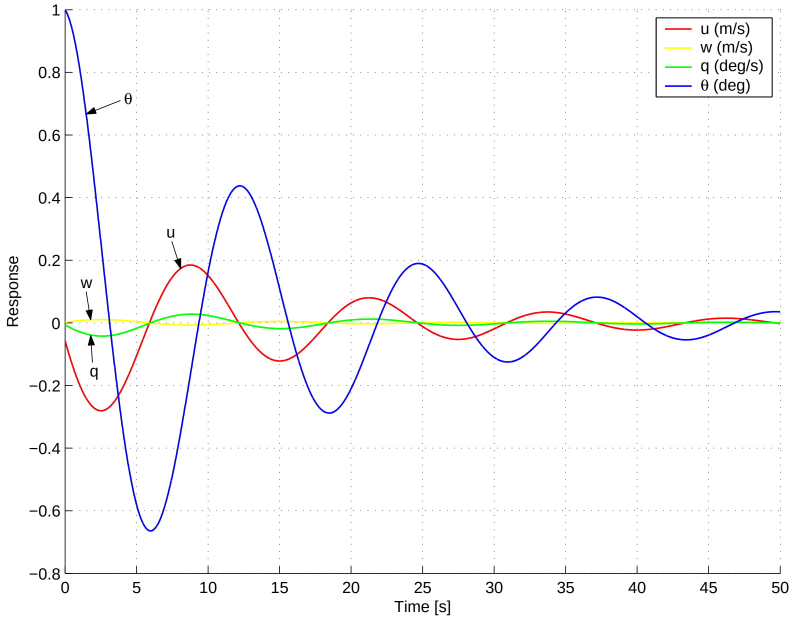


Figure 4.3: *Pure Phugoid Mode Response*

An extract of the longitudinal modal input matrix ($\boldsymbol{G_l}$) of equation (4.14), pertinent to the phugoid mode, is shown below,

$$\boldsymbol{G_{l_{ph}}} = \begin{bmatrix} -9.23 & -0.12 \\ -1.77 & 3.01 \end{bmatrix} \tag{4.19}$$

The magnitude of the first column vector of $\boldsymbol{G_{l_{ph}}}$ indicates the control gain from the elevator to the phugoid mode states. Similarly, the magnitude of the second column vector indicates the control gain from the throttle to the phugoid mode states [3]. Before the magnitudes of the columns can be directly compared, the values in $\boldsymbol{G_{l_{ph}}}$ must be scaled such that the elevator and throttle states have units of full scale deflection. The elevator and throttle full scale deflection quantities are listed in Table 4.2. Although the current elevator and throttle units are reasonably well scaled for a relative analysis, the scaling is still carried out because, unlike the state vector maximum deviations, the actuator maximum deflections are user defined values set within the capabilities of the actuator. Thus, the accuracy of the analysis is improved by carrying out the scaling.

Calculating the relative magnitudes of the scaled columns of $\boldsymbol{G_{l_{ph}}}$ gives,

$$\frac{\text{Elv}}{\text{Thr}} = \frac{\sqrt{(-9.23 \times 10)^2 + (-1.77 \times 10)^2}}{\sqrt{(-0.12 \times 5)^2 + (3.01 \times 5)^2}} = 6.25 \tag{4.20}$$

This result indicates that the elevator is approximately 6 times more effective at controlling the phugoid mode than the throttle is.

4.1.2 Short Period Mode

The short period mode poles are summarised below,

$$\lambda = -11.66 \pm 12.61i \tag{4.21}$$

$$\zeta = 0.68 \tag{4.22}$$

$$\omega_n = 17.17 \tag{4.23}$$

These poles form a stable, high frequency, well damped oscillatory mode. The period of oscillation is about 0.37s with an exponential decay time constant of just under 0.1s. The short period mode eigenvector is displayed in phasor form below,

$$\begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} = \begin{bmatrix} 0.0042 \angle -123.85 \\ 0.2769 \angle -87.81 \\ 1.0000 \angle 0.00 \\ 0.5823 \angle -132.77 \end{bmatrix} \tag{4.24}$$

The eigenvector indicates that the mode is observed mainly in pitch rate with smaller corresponding changes in pitch angle and angle of attack. A classic feature of this mode is that there is almost no change in axial velocity.

The short period mode can be compared to a torsional mass-spring-damper system, where the aircraft is restrained about the OY_B-axis. During a disturbance the tailplane plays the major role in producing the spring and damper forces. The spring effect arises from the natural *weathercock* tendency of the tailplane to align itself with the incident flow. This effect is quantified by the pitching moment due to normal velocity disturbance derivative (M_w). The damping is due to the induced angle of incidence on the tailplane as it oscillates through the oncoming airflow. This effect is quantified by the pitch rate damping stability derivative (M_q). The large tailplane of the aircraft in this project explains why the natural damping of the short period mode is so good.

A time history plot of pure short period mode motion is shown in Figure 4.4, where the longitudinal system (4.1) has been initialised with the real part of the short period mode eigenvector. Note that the axial velocity change is not even visible on the plot.

An extract of the longitudinal modal input matrix ($\boldsymbol{G_l}$) of equation (4.14), pertinent to the short period mode, is shown below,

$$\boldsymbol{G}_{\boldsymbol{l_{sp}}} = \begin{bmatrix} -24.06 & -0.15 \\ 0.65 & -0.14 \end{bmatrix} \tag{4.25}$$

Continuing in a similar fashion to the analysis of the phugoid mode input matrix, the calculation of the relative magnitudes of the scaled columns of $\boldsymbol{G}_{\boldsymbol{l_{sp}}}$ reveals,

$$\frac{\text{Elv}}{\text{Thr}} = \frac{\sqrt{(-24.06 \times 10)^2 + (0.65 \times 10)^2}}{\sqrt{(-0.15 \times 5)^2 + (-0.14 \times 5)^2}} = 233.48 \tag{4.26}$$

The above result makes it clear that the short period mode is orders of magnitude better controlled by the elevator than by the throttle.

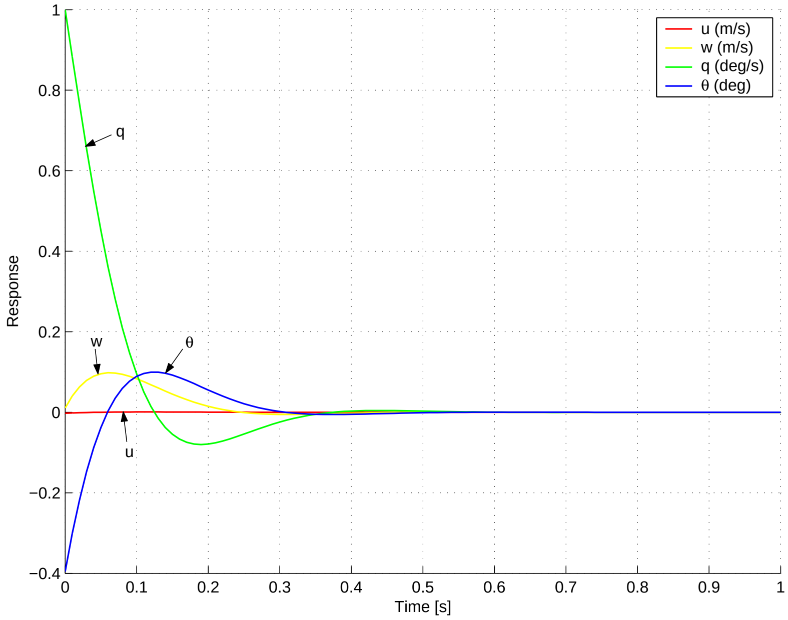


Figure 4.4: *Pure Short Period Mode Response*

4.2 Lateral Aircraft Model

Figure 4.5 shows a block diagram representation of the lateral aircraft model described by equations (3.100) to (3.102).

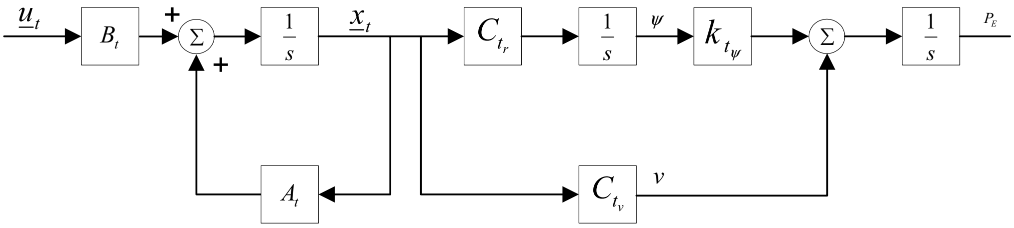


Figure 4.5: *Lateral Aircraft Model*

The lateral model can be written as,

$$\dot{\mathbf{x}}_{\boldsymbol{t}} = \boldsymbol{A_t}\mathbf{x}_{\boldsymbol{t}} + \boldsymbol{B_t}\mathbf{u}_{\boldsymbol{t}} \tag{4.27}$$

$$\dot{\psi} = \boldsymbol{C_{t_r}}\mathbf{x}_{\boldsymbol{t}} \tag{4.28}$$

$$\dot{y} = k_t\psi + \boldsymbol{C_{t_v}}\mathbf{x}_{\boldsymbol{t}} \tag{4.29}$$

where,

$$\boldsymbol{A_t} = \boldsymbol{M_t}^{-1}\boldsymbol{A'_t} = \begin{bmatrix} -0.57 & 0 & -3.49 & 0.17 \\ -37.22 & -45.15 & 4.72 & 0 \\ 18.81 & -0.36 & -2.52 & 0 \\ 0 & 10 & 0 & 0 \end{bmatrix} \tag{4.30}$$

$$\boldsymbol{B_t} = \boldsymbol{M_t}^{-1}\boldsymbol{B'_t} = \begin{bmatrix} 0 & 0.16 \\ -52.58 & 1.50 \\ 0.53 & -5.23 \\ 0 & 0 \end{bmatrix} \tag{4.31}$$

$$\boldsymbol{C_{t_r}} = \begin{bmatrix} 0 & 0 & 10 & 0 \end{bmatrix} \tag{4.32}$$

$$\boldsymbol{C_{t_v}} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \tag{4.33}$$

$$k_{t_\psi} = \frac{20\pi}{180} \tag{4.34}$$

The state vector ($\mathbf{x}_{\boldsymbol{t}}$) and control input vector ($\mathbf{u}_{\boldsymbol{t}}$) are defined in equations (3.103) and (3.104) respectively. The heading and cross track error states have not been augmented to the lateral state space system of equation (4.27), because their dynamics are trivial and only cloud the analysis of the core lateral dynamics.

The units of v are in metres per second (m/s), y in metres (m), p and r in tens of degrees per second (10deg/s) and ϕ, ψ, δ_a and δ_r in degrees (deg). The lateral model units were chosen in the same way as the longitudinal model units were. The maximum deviations of the variables in the lateral state vector are discussed in Table 4.3 and the maximum allowed control deflections are listed in Table 4.4. The tables make it clear that, with the roll and yaw rate units set to tens of degrees per second, a unit deviation in each variable has roughly the same physical significance to the lateral aircraft motion.

Proceeding in a similar fashion to the analysis of the longitudinal system, modal decomposition is used to gain insight into the primary lateral aircraft model. The modal transformation matrix that converts the system of equation (4.27) to modal form is shown below,

$$\begin{bmatrix} v \\ p \\ r \\ \phi \end{bmatrix} = \begin{bmatrix} 0.0015 & 0.0521 & 0.4287 & 0.0064 \\ 1.0000 & -0.0049 & -0.3650 & -0.0003 \\ 0.0077 & 1.0000 & 0.0000 & 0.0480 \\ -0.2214 & -0.4291 & 0.0864 & 1.0000 \end{bmatrix} \begin{bmatrix} z_{t_1} \\ z_{t_2} \\ z_{t_3} \\ z_{t_4} \end{bmatrix} \tag{4.35}$$

Var	Description	Max	Unit
v	At trim velocity of 20m/s, a lateral velocity of 5m/s causes approximately 15deg of sideslip. This is considered the maximum angle of sideslip	5	m/s
p	The maximum roll rate disturbance during conventional flight is estimated as 90deg/s	9	10deg/s
r	The maximum yaw rate disturbance during conventional flight is estimated as 60deg/s	6	10deg/s
ϕ	The maximum roll angle deviation during conventional flight is estimated as 15deg	15	deg

Table 4.3: *Lateral State Vector Maximum Deviations*

Var	Description	Max	Unit
δ_a	The maximum aileron deflection is limited to 10deg	10	deg
δ_r	The maximum rudder deflection is limited to 10deg	10	deg

Table 4.4: *Lateral Control Vector Maximum Deflections*

where the original state vector is shown on the left and the corresponding modal state vector on the right. Transforming the lateral state space model of equation (4.27) reveals,

$$\dot{\mathbf{z}}_{\boldsymbol{t}} = \boldsymbol{F_t}\mathbf{z}_{\boldsymbol{t}} + \boldsymbol{G_t}\mathbf{u}_{\boldsymbol{t}} \tag{4.36}$$

where,

$$\mathbf{z}_{\boldsymbol{t}} = \begin{bmatrix} z_{t_1} & z_{t_2} & z_{t_3} & z_{t_4} \end{bmatrix}^T \tag{4.37}$$

$$\boldsymbol{F_t} = \begin{bmatrix} -45.17 & 0 & 0 & 0 \\ 0 & -1.54 & 8.20 & 0 \\ 0 & -8.20 & -1.54 & 0 \\ 0 & 0 & 0 & -0.0027 \end{bmatrix} \tag{4.38}$$

$$\boldsymbol{G_t} = \begin{bmatrix} -52.51 & 1.84 \\ 1.46 & -5.15 \\ 0.16 & 1.02 \\ -11.01 & -1.89 \end{bmatrix} \tag{4.39}$$

The lateral system matrix in modal form ($\boldsymbol{F_t}$), clearly shows the two real poles and the set of complex poles that define the natural modes of motion of the primary lateral system. The set of complex poles is known as the *dutch roll* mode while the low frequency real pole is known as the *spiral* mode and the high frequency pole as the *roll* mode. Each of these modes is now discussed in more detail with specific focus on the mode 'shapes' and the relative control of the lateral actuators over each mode.

4.2.1 Roll Mode

The roll mode pole is at,

$$\lambda = -45.17 \tag{4.40}$$

This corresponds to a very fast, first order exponential response with a time constant of 0.022s. The roll mode eigenvector is displayed in phasor form below,

$$\begin{bmatrix} v \\ p \\ r \\ \phi \end{bmatrix} = \begin{bmatrix} 0.0015 \angle 0 \\ 1.0000 \angle 0 \\ 0.0077 \angle 0 \\ 0.2214 \angle 180 \end{bmatrix} \tag{4.41}$$

The eigenvector indicates that the roll mode is strongly observed in roll rate with a smaller corresponding amount of roll angle. The lateral velocity and yaw rate perturbations are negligible during this motion.

To visualise the mode, assume that the aircraft is constrained to one degree of freedom motion in roll. If a roll moment disturbance occurs, the aircraft will begin to roll in accordance with Newton's second law. However, as the aircraft rolls, the angle of attack on the down going wing is increased and the angle of attack on the up going wing is decreased. The differential angle of attack across the wing produces differential lift and thus an exponentially increasing roll moment in opposition to the disturbing moment. This interaction manifests itself in a first order roll rate response with a time constant dependent on the roll damping stability derivative (L_p) and the roll axis moment of inertia (I_y) [1]. The fast roll mode time constant is due to the low roll inertia of the model aircraft in this project.

A time history plot of pure roll mode motion is shown in Figure 4.6, where the lateral state space system (4.27) has been initialised with the roll mode eigenvector. The lateral velocity and yaw rate motion are not even visible on the plot.

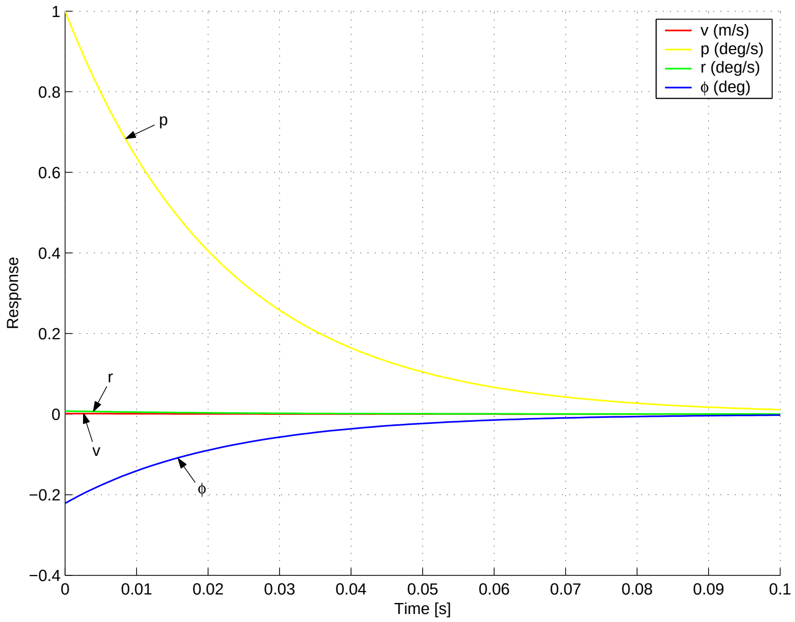


Figure 4.6: *Pure Roll Mode Response*

An extract of the lateral modal input matrix ($\boldsymbol{G_t}$) of equation (4.39), pertinent to the roll mode, is shown below,

$$\boldsymbol{G_{t_r}} = \begin{bmatrix} -52.51 & 1.84 \end{bmatrix} \tag{4.42}$$

With reference to Table 4.4, the maximum deflections of both the aileron and rudder are set at 10°. This means that the relative magnitudes of the elements of $\boldsymbol{G_{t_r}}$ can be directly compared without the need for scaling. Comparing the column magnitudes of $\boldsymbol{G_{t_r}}$ reveals,

$$\frac{\text{Ail}}{\text{Rud}} = \frac{|-52.51|}{|1.84|} = 28.54 \tag{4.43}$$

which implies that the roll mode is almost 30 times better controlled by the ailerons than by the rudder.

4.2.2 Spiral Mode

The spiral mode pole is at,

$$\lambda = -0.0027 \tag{4.44}$$

This corresponds to an extremely slow, first order exponential response with a time constant of 370s. The mode is only just stable and could well turn out to be unstable in reality given errors in the aerodynamic model. The spiral mode eigenvector is displayed in phasor form below,

$$\begin{bmatrix} v \\ p \\ r \\ \phi \end{bmatrix} = \begin{bmatrix} 0.0064 \angle 0 \\ 0.0003 \angle 180 \\ 0.0480 \angle 0 \\ 1.0000 \angle 0 \end{bmatrix} \tag{4.45}$$

The eigenvector indicates that the spiral mode is primarily observed in bank angle. With such a long time constant, the bank angle will translate into a slow turn which will make the spiral mode very well observed in heading angle.

The spiral mode describes the tendency of the aircraft to resettle to or diverge from a wings level position after a disturbance. The forces involved arise from a complex interaction of roll, yaw and sideslip motions. Assume the aircraft has been disturbed from its wings level position by a small positive bank angle ϕ. Left unchecked, this disturbance will result in positive sideslip which will cause the aircraft to yaw to the right as a result of the weather cock effect (N_v). The yaw rate motion causes an increase in airflow velocity over the left wing and a decrease in airflow velocity over the right wing. The differential airflow velocity results in a positive roll moment that tends to further increase the bank angle disturbance and cause a diverging motion. However, dihedral and quarter chord sweep back effects of the wing (L_v), produce a negative roll moment in response to the sideslip that tends to restore the wings to a level condition. These opposing forces are usually very similar in magnitude which gives rise to the long time constant of the spiral mode [1].

A time history plot of pure spiral mode motion is shown in Figure 4.7, where the lateral system (4.27) has been initialised with the spiral mode eigenvector.

An extract of the lateral modal input matrix ($\boldsymbol{G_t}$) of equation (4.39), pertinent to the spiral mode, is shown below,

$$\boldsymbol{G_{t_s}} = \begin{bmatrix} -11.01 & -1.89 \end{bmatrix} \tag{4.46}$$

Comparing the relative magnitudes of $\boldsymbol{G_{t_s}}$ reveals,

$$\frac{\text{Ail}}{\text{Rud}} = \frac{|-11.01|}{|-1.89|} = 5.83 \tag{4.47}$$

which implies that the spiral mode is almost 6 times better controlled by the ailerons than by the rudder.

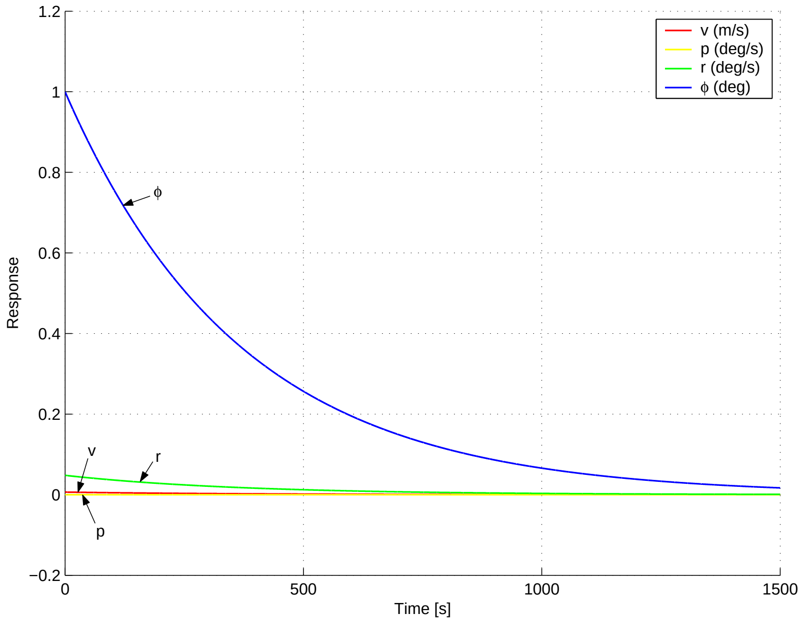


Figure 4.7: *Pure Spiral Mode Response*

4.2.3 Dutch Roll Mode

The dutch roll mode poles are summarised below,

$$\lambda = -1.54 \pm 8.20i \tag{4.48}$$

$$\zeta = 0.18 \tag{4.49}$$

$$\omega_n = 8.34 \tag{4.50}$$

These poles form a stable, lightly damped oscillatory mode. The period of oscillation is 0.75s with an exponential decay time constant of 0.65s. The dutch roll mode eigenvector is displayed in phasor form below,

$$\begin{bmatrix} v \\ p \\ r \\ \phi \end{bmatrix} = \begin{bmatrix} 0.4319 \angle 83.07 \\ 0.3650 \angle -90.77 \\ 1.0000 \angle 0.00 \\ 0.4377 \angle 168.62 \end{bmatrix} \tag{4.51}$$

The eigenvector indicates that the dutch roll mode is observed well in all of the lateral states but predominantly in yaw rate. Note that the yaw rate motion leads the roll rate motion by approximately 90° which is a classic feature of this mode [1].

Fundamentally, the dutch roll mode is the lateral equivalent of the short period mode with the fin causing the spring and damper forces. However, since the fin is usually less aerodynamically effective than the tailplane, the mode tends to be lightly damped. The mode can be visualised by initially considering the aircraft to be restrained about the OZ_B-axis. This is then the lateral equivalent of the short period mode setup, with the weather cock stability derivative (N_v) primarily determining the frequency of oscillation after a disturbance, and the yaw damping stability derivative (N_r) primarily determining the damping [4]. However, the yaw rate oscillation causes differential airflow across the wings and thus induces a rolling motion that lags the yaw rate motion by approximately 90°. This phase difference between the yawing and rolling motion means that the forward going wing is low and the aft moving wing is high. The dutch roll mode motion can be well visualised as a rhythmic lateral motion of the aircraft, where the wing tips trace ellipses in the OX_BZ_B-plane [1].

A time history of pure dutch roll mode motion is shown in Figure 4.8, where the lateral state space system (4.27) has been initialised with the real part of the dutch roll mode eigenvector. Note the 90° lag of roll rate from yaw rate.

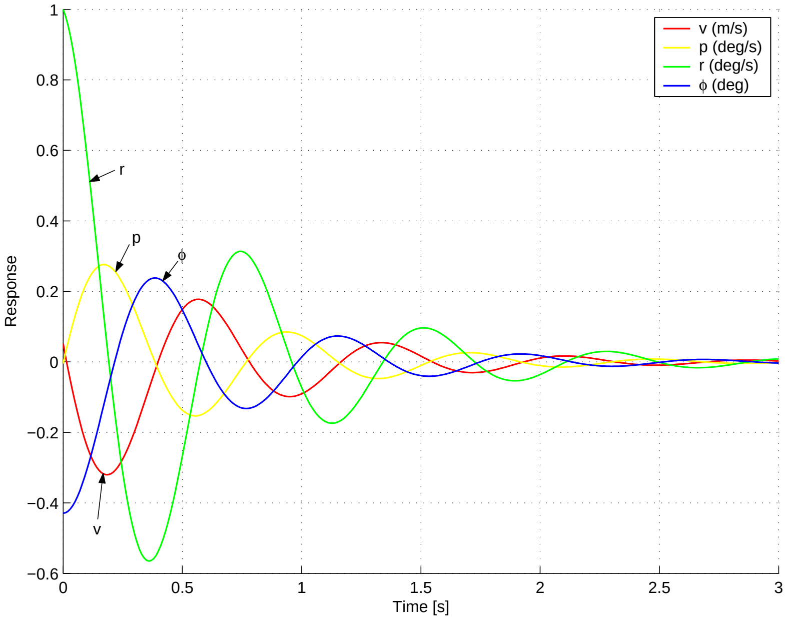


Figure 4.8: *Pure Dutch Roll Mode Response*

An extract of the lateral modal input matrix ($\boldsymbol{G_t}$) of equation (4.39), pertinent to the dutch roll mode, is shown below,

$$\boldsymbol{G}_{l_{dr}} = \begin{bmatrix} 1.46 & -5.15 \\ 0.16 & 1.02 \end{bmatrix} \tag{4.52}$$

The relative magnitudes of the columns of $\boldsymbol{G}_{\boldsymbol{l_{dr}}}$ reveal,

$$\frac{\text{Ail}}{\text{Rud}} = \frac{\sqrt{(1.46)^2 + (0.16)^2}}{\sqrt{(-5.15)^2 + (1.02)^2}} = 0.28 \tag{4.53}$$

which implies that the dutch roll mode is approximately 4 times better controlled by the rudder than by the ailerons.

4.3 Summary

In this chapter, the longitudinal and lateral aircraft models, specific to the aircraft used in this project, were presented. The modes of motion were analysed, showing which states are dominant in each mode, and what actuators are most effective at controlling each mode. The information gained on the mode shapes and actuator effectiveness is used in the next chapter to aid the design of the controller. Finally, the dynamics of motion analysed in this chapter are consistent with those of conventional aircraft, which instills confidence that there are no dramatic errors in the aircraft models.

Chapter 5

Control and Simulation

In this chapter, the autopilot design is considered. The design approach and controller architecture are discussed in Section 5.1. Then, Sections 5.2 and 5.3 provide the detailed designs of the longitudinal and lateral controllers.

5.1 Controller Overview

The role of the controller is to make the aircraft follow commanded motions. A simple approach to this end is to use open loop control, where the aircraft poles are merely cancelled by compensator zeros and the compensator poles used to define the desired aircraft dynamics. The problem with this solution is that if the aircraft's dynamic model changes or the aircraft experiences disturbances, then the control system performance deteriorates rapidly.

Feedback control is the solution to this problem as it dramatically reduces the sensitivity of the system to modeling uncertainties and provides disturbance rejection to those quantities that detract from the desired aircraft motion. Typical unwanted effects that must be rejected by the control system are disturbances due to wind, fuel level changes, backlash in the servos, the bending of servo rods and changes in the aircraft model due to the flight condition and aeroelastic effects. To reduce the sensitivity of the system to modeling errors, the controllers can be designed with high gain and phase margins, as this provides an indication of controller robustness.

The fundamental task in controlling an aircraft is to control its attitude. The attitude must be controlled because it essentially provides control over the aircraft's acceleration e.g. the bank angle of an aircraft determines its centripetal acceleration. Direct control over the aircraft's acceleration is desired because the stabilisation effects integrate down into the velocity and position states over time and thus leave only the reference velocity and position commands to be regulated by outer control loops.

In light of this, the controller design is broken into three sections.

1. Stability augmentation controllers
2. Attitude regulation controllers
3. Trajectory controllers

The stability augmentation controllers are responsible for augmenting the natural stability of the higher frequency aircraft modes, by providing artificial damping. The attitude controllers then form the lowest level of regulation, and are responsible for regulating the aircraft's pitch and roll angles, effectively controlling its acceleration. Finally, the trajectory controllers form the outer loops that regulate trajectory motion variables such as heading and altitude. These control loops provide commands to the attitude controllers and thus deal with the kinematics of the aircraft more than its natural dynamics. The advantage of this successive loop closure design methodology is that once the attitude controllers have been practically implemented, then the trajectory controllers can be implemented with confidence because the control algorithms are essentially independent of the aircraft.

Each of the three controller types is now discussed in more detail.

5.1.1 Stability Augmentation

Stability augmentation involves suppressing any high frequency aircraft dynamics by providing artificial damping. The reason that damping is used to reduce the high frequency dynamics, can be demonstrated by considering a simple one dimensional problem, where the power of a system (P) can be written as,

$$P = FV \tag{5.1}$$

where F is the force acting on a system and V is its velocity. If the control law is set to provide damping i.e. set F proportional to $(-V)$ then the system power becomes,

$$P = (-bV) \cdot V = -bV^2 \tag{5.2}$$

where b is the damper feedback gain and is positive for stability. Equation (5.2) makes it clear that with the damping control law in place, the system power is always negative. This implies that a damper always removes energy from a system, which implies stability in the sense of Lyapanov.

Modal analysis showed that the natural high frequency aircraft modes are the short period mode, the dutch roll mode and the roll mode. The natural damping of these modes is caused by induced angles of incidence on the aircraft's airfoils during angular rate perturbations. This means that the damping is dependant on the aircraft velocity and centre of gravity location, both of which change during flight. The damping is also always relative to the *airflow*, and not to *inertial space*. Thus, it is desired to provide artificial damping, to desensitise the natural damping to changes in flight parameters, and to provide damping with respect to inertial space. Damping with respect to inertial space is desired because it is effective at rejecting wind gust disturbances.

To provide damping, the aircraft's angular rates need to be sensed. Low cost rate gyroscopes can be used to adequately sense these rates. Although these sensors display significant biases, the outer attitude and trajectory loops will compensate for any steady state errors. The signals from these gyroscopes can then be fed to the respective actuators to augment the aircraft's natural damping.

In this project, only a short period damper (pitch rate damper) and a dutch roll damper are implemented. No artificial roll damping is included because the roll mode is heavily damped and of a high frequency. The outer attitude control loops will provide the desired wind gust disturbance rejection.

5.1.2 Attitude Control

If good sensors and a good aircraft model were available, then all of the aircraft states could either be measured or estimated and used in a Linear Quadratic Regulator (LQR) to solve the entire attitude control problem. However, the reality of the situation is that the aircraft model is of a limited accuracy and only low cost sensors are affordable. These two facts make it desirable to avoid using an estimator based on a low accuracy model, and updated with measurements from low cost sensors, but to rather use directly sensed information to control the aircraft's attitude.

Typical affordable sensors used to directly measure attitude are rate gyroscopes (where the signal is integrated over time), infrared sensors [21] and vision based systems [22]. Rate gyroscopes are inertial sensors and thus have the advantage that they work equally well in all surrounding conditions. This quality is desired for the purposes of autonomous navigation and thus discounts the use of the infrared and vision based systems. The difficultly with using low cost rate gyroscopes, is that they typically display biases on the order of a few tenths of a degree per second. Thus they cannot be integrated over time to provide the aircraft's attitude in a stand alone fashion. However, there are a number of techniques that can be used to relieve this problem.

One approach is to complement the rate gyroscopes with a set of accelerometers. Low cost accelerometers are capable of providing acceptable attitude information on stationary platforms, where the force they measure directly opposes that of gravity. Rate gyroscopes on the other hand, provide good angular rate measurements at high frequencies but display unacceptable biases at very low frequencies. Using these two sensors in a complementary filter, where the rate gyroscopes are made to provide the high frequency attitude information and the accelerometers the low frequency information, a good attitude measurement over a large bandwidth can be attained [40]. This approach however, has adverse properties during steady manoeuvres such as constant turns or loops, where there is a *false erection of the vertical* [3].

Another solution is to build a large estimator based on the rigid body aircraft dynamic equations and the sensor models. The estimator accepts measurements from low cost rate gyroscopes and accelerometers as well as a variety of other sensors such as a GPS receiver and a magnetometer. This approach provides estimates of all aircraft's states but tends to be computationally demanding [16], and is corrupted by failure of any one of the multiple sensors.

It is clear that determining the aircraft's attitude using only low cost sensors is a difficult task. However, if states that are directly related to the aircraft's attitude under common flight conditions can be found, and these states are easily measured using low cost sensors, then they can be used to indirectly control the aircraft's attitude. This approach was adopted in this project and the pitch and roll angle controllers are discussed below. The controllers have the advantage that they are simple to implement, only require low cost sensors and require very little processing power.

Pitch Angle Control

Consider the primary longitudinal state space model of equation (4.1) together with the climb rate equation (4.3), restated in vector form below,

$$\dot{\mathbf{x}}_{\boldsymbol{l}} = \boldsymbol{A_l}\mathbf{x}_{\boldsymbol{l}} + \boldsymbol{B_l}\mathbf{u}_{\boldsymbol{l}} \tag{5.3}$$

$$\dot{h} = \boldsymbol{C}_{\boldsymbol{l}_{\dot{h}}} \mathbf{x}_{\boldsymbol{l}} \tag{5.4}$$

These two vector equations define a set of five first order, linear differential equations in seven variables: u, w, q, θ, $\dot{h}$, δ_e and δ_t. Thus, all that is required for the unique solution of the system are two more, linearly independent equations that ensure that the system remains in the linear region about the model trim condition. These equations are of course the control law equations and define how the control variables are calculated from the various states. The airspeed change (u) and climb rate ($\dot{h}$) states can easily be measured using low cost differential and absolute pressure sensor respectively (see Chapter 7.1.3). An absolute pressure sensor is primarily an altitude sensor (which is required in Section 5.1.3), but is also capable of providing an adequate climb rate signal. Thus, if these two states are used to generate the control signals, then the pitch angle can be stabilised without the need for a direct measurement. This approach also automatically regulates two motion variables of interest for conventional flight.

Roll Angle Control

Consider the diagram of Figure 5.1 where the aircraft is shown in a steady, constant altitude, banked turn.

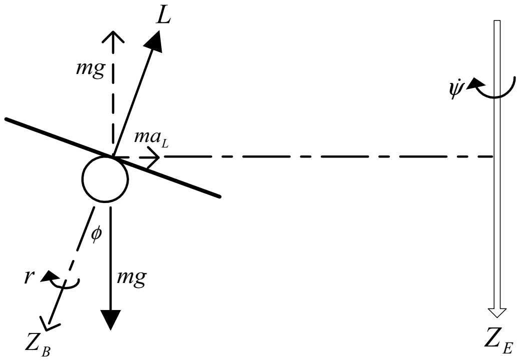


Figure 5.1: *Aircraft in a Steady Banked Turn*

During the turn, the lift vector is responsible for countering the weight of the aircraft and for providing the necessary centripetal acceleration for the turn. Thus, the lateral acceleration can be written as,

$$a_L = U_0 \dot{\psi} = g \tan \phi \tag{5.5}$$

If the roll and pitch angles are limited to small angles, then substitution of equation

(3.101) into equation (5.5) reveals,

$$\phi \approx \frac{U_0}{g} r \tag{5.6}$$

This means that for small pitch and roll angles, the aircraft's yaw rate is proportional to the roll angle. Thus, the roll angle can be controlled by merely controlling the aircraft's yaw rate. Controlling the yaw rate has the major advantage that the signal from a low cost rate gyroscope, never has to be integrated. The lack of integration avoids the problem of roll angle drift and leaves only the gyroscope bias to contend with. However, with a yaw rate controller in place, the bias will merely cause the aircraft to enter into a slow turn that opposes its magnitude. This turn can easily be corrected by an outer trajectory control loop.

5.1.3 Trajectory Control

The trajectory controllers form the outer-most loops of the control system and are responsible of regulating the following states,

- Altitude (h)
- Heading (ψ)
- Cross Track Error (y)

These controllers are chosen because, together with the airspeed, climb rate and yaw rate regulation capabilities of the attitude controllers, they make it possible for the autopilot to perform all conventional flight tasks.

The altitude is easily controlled by taking a height measurement and feeding it to the climb rate controller. The feedback is used to stabilise the extra integrator pole that the height state adds to the system. The required altitude measurement can be attained from an absolute pressure sensor, and in light of the design goal of keeping the avionics cost as low as possible, explains why an absolute pressure sensor is used to measure climb rate.

The heading and cross track error (guidance) controllers can be effectively implemented by successive loop closure. A heading measurement can be fed to the yaw rate controller to stabilise the extra integrator added to the system by the heading state. The guidance controller can then be implemented by feeding a cross track error measurement to the heading controller, to stabilise the single integrator added to the system by the cross track error state. This successive loop closure technique minimises the amount of dynamic compensation required in implementing the two lateral trajectory controllers. The heading and cross track error states can both be measured using a low cost GPS receiver. The performance of the low cost GPS receiver used in this project (see Chapter 7.1.5) is well documented in [23] and is capable of providing heading measurements accurate to within 2 degrees and position measurements accurate to within a 4m radius, 95% of the time. These measurements are acceptable for the purposes of this project where high accuracy lateral control is not required.

5.1.4 Overview of the Controllers

The stability augmentation, attitude and trajectory controllers discussed in the previous three sections, are divided into a longitudinal and lateral set and summarised below.

- Longitudinal:
    - Stability Augmentation: Pitch Rate Damper
    - Attitude Regulation: Airspeed/Climb Rate Controller
    - Trajectory Control: Altitude Controller
- Lateral:
    - Stability Augmentation: Dutch Roll Damper
    - Attitude Regulation: Turn Rate Controller
    - Trajectory Control: Heading Controller
    - Trajectory Control: Guidance Controller

The entire control system can be implemented using the following low cost sensors,

- 3 Rate Gyroscopes (R450 per rate gyroscope)
- Absolute and Differential Pressure Sensors (R120 and R950 respectively)
- GPS Receiver (R550)

The controller architecture keeps the autopilot design simple, requires very little processing power for implementation and makes it possible for the aircraft to perform all conventional flight manoeuvres.

The following two sections provide the detailed design of the longitudinal and lateral controllers as applied to the aircraft used in this project. All of the controllers, except the heading and guidance controllers are designed in the continuous time domain. The heading and guidance controllers are designed in the discrete domain because the GPS receiver used in this project is only capable of providing measurement updates at 4Hz. The rest of the controllers are converted to the discrete domain using Euler's digitisation method at a sample rate of 25Hz. This rate was chosen as a convenient rate for implementation, well above the frequencies of the aircraft's natural modes of motion. It also makes other discretisation approximations, such as Tustin's method, unnecessary. The discretisation of these controllers is not shown in this chapter.

All controller designs are carried out using root locus methods in MATLAB while simulations are performed in Simulink. During the following descriptions of each controller design, all subscripts have been dropped for clarity. The open loop system is merely referenced at the onset of the design and the closed loop system defined at the end for future reference.

5.2 Longitudinal Controllers

The design and simulation of the aircraft's longitudinal controllers is considered in this section. Each controller design begins with a discussion of the design approach before the details of the controller are considered and the results verified by linear simulation.

5.2.1 Pitch Rate Damper

The natural damping of the short period mode predominantly arises from pitch rate induced angles of incidence on the tailplane during pitch rate perturbations. Thus, the damping is a function of the airspeed and the tailplane moment arm length (l_T). The aircraft's airspeed changes regularly during flight and the tailplane moment arm length is a function of the centre of gravity location, which varies with fuel consumption. Artificial pitch rate damping is thus desired to desensitise the overall aircraft pitch rate damping to variations in these flight parameters, as well as to provide damping with respect to inertial space for wind gust disturbance rejection.

Modal analysis showed that the short period mode is orders of magnitude better controlled by the elevator than by the throttle. This suggests that simple feedback of the aircraft's pitch rate to the elevator will create an effective pitch rate damper. This feedback will essentially provide the aircraft with a larger tailplane during pitch rate perturbations.

Controller Design

The control system is summarised in the block diagram of Figure 5.2 where the plant dynamic equations are obtained from equation (4.1). The output matrix (***C***) is set to extract the pitch rate state in degrees per second (deg/s).

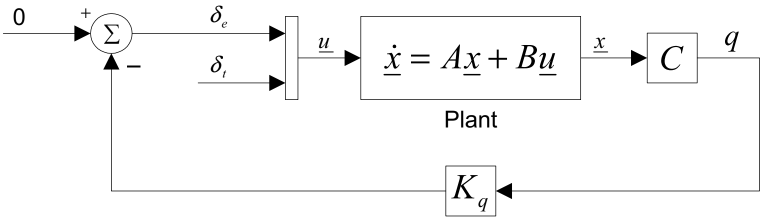


Figure 5.2: *Pitch Rate Damper Block Diagram*

The root locus plot for variation of the feedback gain from pitch rate to elevator is shown in Figure 5.3. The two poles with large movement are the short period mode poles while the phugoid mode poles are barely visible near the origin.

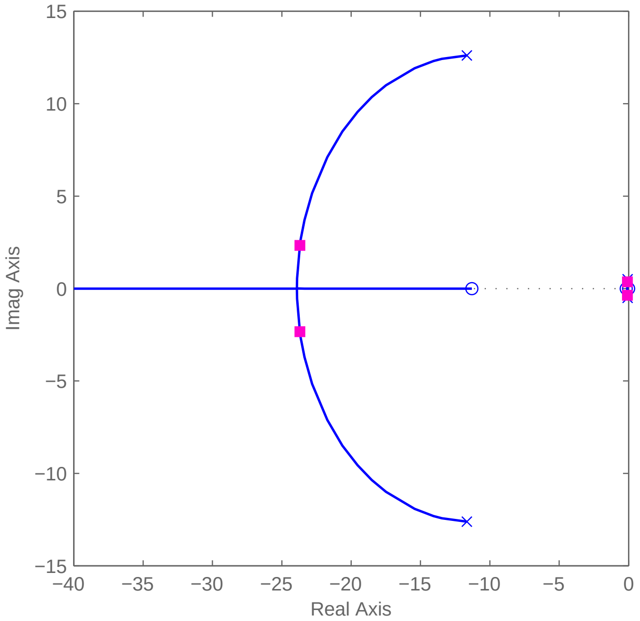


Figure 5.3: *Pitch Rate Damper Root Locus*

It is desired to choose the pitch rate damper feedback gain as high as possible, so as to provide maximum artificial damping. However, if the feedback gain is made too high then the bandwidth of the system begins to increase and unmodeled dynamics can be excited. These can include dynamics due to structural resonant modes as well as non-linear actuator effects such as slew rate and saturation. A feedback gain of,

$$K_q = -0.1 \tag{5.7}$$

increases the short period mode damping to 0.9, which is deemed sufficient to adequately desensitise the pitch damping loop without exciting unmodeled dynamics. Of course, this feedback gain can easily be fine tuned during practical flight tests as desired. The closed loop system is described by,

$$\dot{\mathbf{x}} = \Big[\boldsymbol{A} - \boldsymbol{B}[K_q \quad 0]^T \boldsymbol{C}\Big]\mathbf{x} + \boldsymbol{B}\mathbf{u} \tag{5.8}$$

5.2.2 Airspeed and Climb Rate Controller

The aircraft's pitch attitude is regulated by the control of airspeed and climb rate using the elevator and throttle actuators. The simplest approach to the controller design is to divide it into an airspeed control loop and a climb rate control loop. These two Single-Input-Single-Output (SISO) loops then work independently of each other and are only coupled through the aircraft's dynamics. This approach however, is not expected to perform well because the controllers essentially ignore the significant cross coupling effects evident between the two control loops. Thus, a Multi-Input-Multi-Output (MIMO) controller design is adopted.

One approach to the MIMO controller design is to build an estimator based on the aircraft's dynamic model that accepts airspeed and climb rate measurements and provides an estimate of the longitudinal state vector ($\mathbf{x}_{\boldsymbol{l}}$). A LQR can then be designed for full state feedback. This approach however, is unattractive because it involves building an estimator that is based on the low accuracy aircraft dynamic model, which places high demands on the airspeed and climb rate sensors to correct the state vector estimate. Thus, to avoid the use of an estimator, full state feedback is discarded, and only the states directly measured are used in the MIMO feedback controller.

The major limitations on the performance of the airspeed and climb rate controller are the elevator and throttle actuation limits. In terms of the actuator dynamics, only the throttle lag dynamics are of any significance to the controller design, with the servo dynamics being outside the bandwidth of interest. Thus, with the linear throttle lag dynamics of equation (4.4) included in the plant model, it is only necessary to impose limits on the maximum actuation authority such that the linear actuator models remain valid. These limits have already been listed in Table 4.2 and are restated here for convenience.

- Maximum elevator deflection of 10° from the trim condition
- Maximum throttle authority of 5m/s^2 from the trim condition

The specifications for the airspeed and climb rate controller are listed below, where the maximum step response capabilities are chosen as reasonable values for conventional flight.

- To cater for airspeed steps of up to 5m/s and climb rate steps of up to 2m/s
- Rise time of under 3s
- Overshoot of less than 20%
- Zero steady state errors

The speed of response performance of the airspeed and climb rate controller in this project is not critical. It is desired to make the controller respond as quickly as possible, within the physical limitations of the aircraft, so as to provide maximum disturbance rejection without saturating the elevator or throttle, or stimulating unmodeled dynamics. Thus, the rise time specification was chosen as a reasonable value, well within the capabilities of the aircraft. In a classic second order system, a 20% overshoot implies a damping of approximately 0.6 and a phase margin of roughly 60°. Although for a MIMO system, the phase and gain margins do not provide a complete indication of the controller's robustness [15], the specification was still imposed to provide a satisfactory bound on the airspeed and climb rate responses. The zero steady state error specification forces the use of a compensator integrator term to provide the infinite steady state gain required. These integrators are desired because they are capable of negating the effects of modeling errors in steady state terms such as C_{m_0} and $C_{L_{W_0}}$, as well as forcing out steady state errors due to unmodeled effects.

Controller Design

A block diagram of the controller is shown in Figure 5.4, where the plant dynamics include the throttle lag dynamics of equation (4.4), augmented to the dynamic model of equation (5.8), as follows,

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\delta}_t \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}[0 \;\; 1] \\ \mathbf{0} & -\frac{1}{\tau_T} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \delta_t \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}[1 \;\; 0] & \mathbf{0} \\ 0 & \frac{1}{\tau_T} \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_{t_c} \end{bmatrix} \tag{5.9}$$

The output matrix $\boldsymbol{C}$ is set to extract the airspeed change and climb rate states, both in metres per second (m/s).

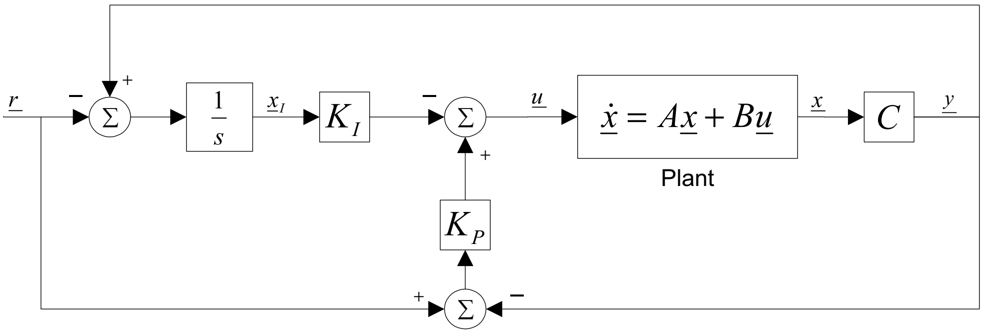


Figure 5.4: *Airspeed and Climb Rate Controller Block Diagram*

The integrator dynamics are,

$$\dot{\mathbf{x}}_{\boldsymbol{I}} = \mathbf{y} - \mathbf{r} = \boldsymbol{C}\mathbf{x} - \mathbf{r} \tag{5.10}$$

where $\mathbf{x_I}$ is the integration error state vector containing the airspeed and climb rate error integrals and $\mathbf{r}$ is the reference input vector for the airspeed and climb rate commands. The PI control law can be written as,

$$\mathbf{u} = -\boldsymbol{K_P}[\mathbf{y} - \mathbf{r}] - \boldsymbol{K_I}[\mathbf{x_I}] \tag{5.11}$$

which simplifies to,

$$\mathbf{u} = -\begin{bmatrix} \boldsymbol{K_P C} & \boldsymbol{K_I} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x_I} \end{bmatrix} + \boldsymbol{K_P}\mathbf{r} \tag{5.12}$$

where,

$$\boldsymbol{K_P} = \begin{bmatrix} K_{(\delta_e \leftarrow u)} & K_{(\delta_e \leftarrow \dot{h})} \\ K_{(\delta_t \leftarrow u)} & K_{(\delta_t \leftarrow \dot{h})} \end{bmatrix} \tag{5.13}$$

$$\boldsymbol{K_I} = \begin{bmatrix} K_{(\delta_e \leftarrow \int e_u)} & K_{(\delta_e \leftarrow \int e_{\dot{h}})} \\ K_{(\delta_t \leftarrow \int e_u)} & K_{(\delta_t \leftarrow \int e_{\dot{h}})} \end{bmatrix} \tag{5.14}$$

and the elements of $\boldsymbol{K_P}$ and $\boldsymbol{K_I}$ are the proportional and error integral feedback gains.

The closed loop system model is formed by first augmenting the integral states to the plant model (5.9) and then substituting equation (5.12) for the input vector $\mathbf{u}$.

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{x}}_{\boldsymbol{I}} \end{bmatrix} = \begin{bmatrix} (\boldsymbol{A} - \boldsymbol{BK_P C}) & -\boldsymbol{BK_I} \\ \boldsymbol{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_{\boldsymbol{I}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{BK_P} \\ -\boldsymbol{I} \end{bmatrix} \mathbf{r} \tag{5.15}$$

The eigenvalues of the closed loop system matrix of equation (5.15) are the closed loop pole locations, and the elements of the feedback gain matrices, $\boldsymbol{K_P}$ and $\boldsymbol{K_I}$, allow the manipulation of these eigenvalues. The design is thus reduced to finding appropriate values for $\boldsymbol{K_P}$ and $\boldsymbol{K_I}$ such that the closed loop pole locations produce responses that meet the design specifications.

A purely mathematical approach where the feedback gains are calculated subject to a certain set of constraints (such as control effort optimality) could be used but a more intuitive, graphical approach that provides good insight, was adopted. This involves an iterative method, where the current closed loop poles along with a set of root loci, corresponding to the change in each element of the feedback gain matrices ($\boldsymbol{K_P}$ and $\boldsymbol{K_I}$), are plotted on the complex frequency plane. The iterative process is summarised in point form below.

1. Plot the poles of the current closed loop system.
2. Vary the feedback gains of $\boldsymbol{K_P}$ and $\boldsymbol{K_I}$ one at a time and plot the corresponding root loci.
3. Analyse the multiple root locus plot and decide on a suitable gain variation to bring the system poles closer to within the specification boundaries.
4. Change the feedback gains accordingly to form a new closed loop system and repeat the process until satisfactory pole locations are reached.

Thus the design begins by plotting the open loop poles and thereafter iteratively changing the feedback gains by small amounts until the system poles have moved to within the specification boundaries. It is important however, to realise that even with infinite actuation power, a specific set of desired closed loop pole locations may not be able to be reached because the full system state vector is not being used for feedback.

The final feedback gains are presented in equations (5.16) and (5.17) and the multiple root locus plot of the system with these gains is shown in Figure 5.5. Root loci that involve feedback to the elevator are plotted as dots, while root loci that involve feedback to the throttle are plotted as plus signs. They are also colour coded depending on the feedback source and only one half of each root locus is shown for clarity. The five poles shown in Figure 5.5 represent the two phugoid mode poles, the two integrator states and the throttle lag dynamics. The short period mode poles are not shown because they are at a much higher frequency and are affected very little by the feedback.

$$\boldsymbol{K_P} = \begin{bmatrix} 0.08 & -0.6 \\ 0.80 & 0.10 \end{bmatrix} \tag{5.16}$$

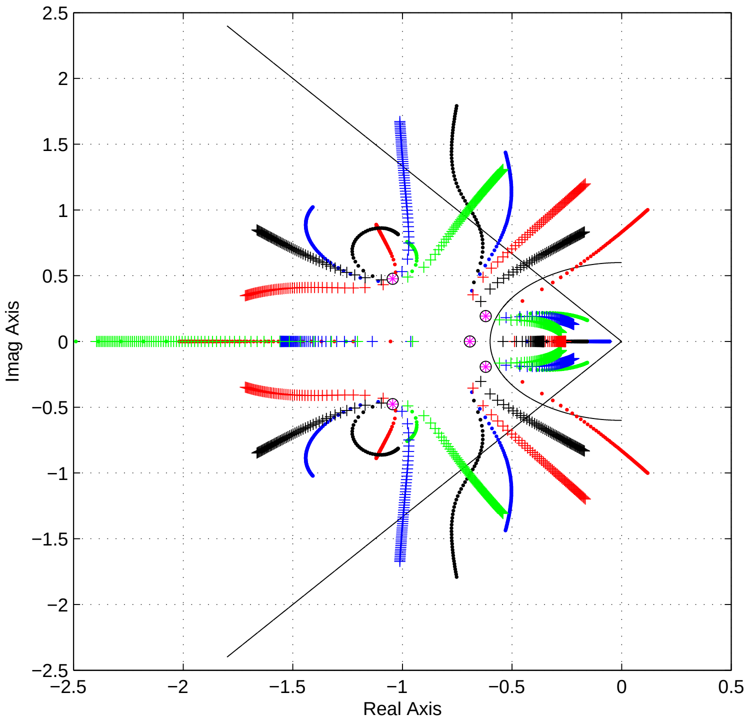


Figure 5.5: *Airspeed and Climb Rate Controller Root Locus*

$$\boldsymbol{K_I} = \begin{bmatrix} 0.08 & -0.12 \\ 0.10 & 0.60 \end{bmatrix} \tag{5.17}$$

All of the gains in $\boldsymbol{K_P}$ and $\boldsymbol{K_I}$ were checked to see if they made intuitive sense. Taking the climb rate feedback to elevator gain for example ($K_{(\delta_e \leftarrow \dot{h})} = -1.5$): if a positive climb rate is measured, the negative gain provides positive feedback and thus a positive elevator command is issued. This by definition causes a negative pitching moment which tends to nullify the climb rate motion. Although in a MIMO system, it is not necessary for all the gains to make intuitive sense for stability, it is still encouraging if they do.

Linear Simulation

The simulation results of a 5m/s step command in airspeed and a 2m/s step command in climb rate are shown in Figures 5.6 and 5.7 respectively. Both responses have maximum overshoots of less than 20% and rise times of less than 3s. The integrators also ensure that the command is followed with zero steady state error. The elevator and throttle are not saturated during either step response. However, in practice, the elevator and throttle may well saturate. To protect the airspeed and climb rate controller against further performance degradation due to saturation, integrator antiwindup was implemented on all of the compensator integrator terms. Without antiwindup, the integrator states build over time during actuator saturation, and this built up 'charge' can cause considerable overshoot when it is released after the actuator returns from saturation. Antiwindup basically stops the compensator integration effect when an actuator saturates so as not to build up large actuation commands.

Finally, note that because the dynamics of the elevator have not been modeled, it is expected to perform the step commands visible in Figures 5.6 and 5.7. These elevator step commands can be relieved with negligible controller performance degradation by low pass filtering the commands, where the filter time constant is on the order of a tenth of a second.

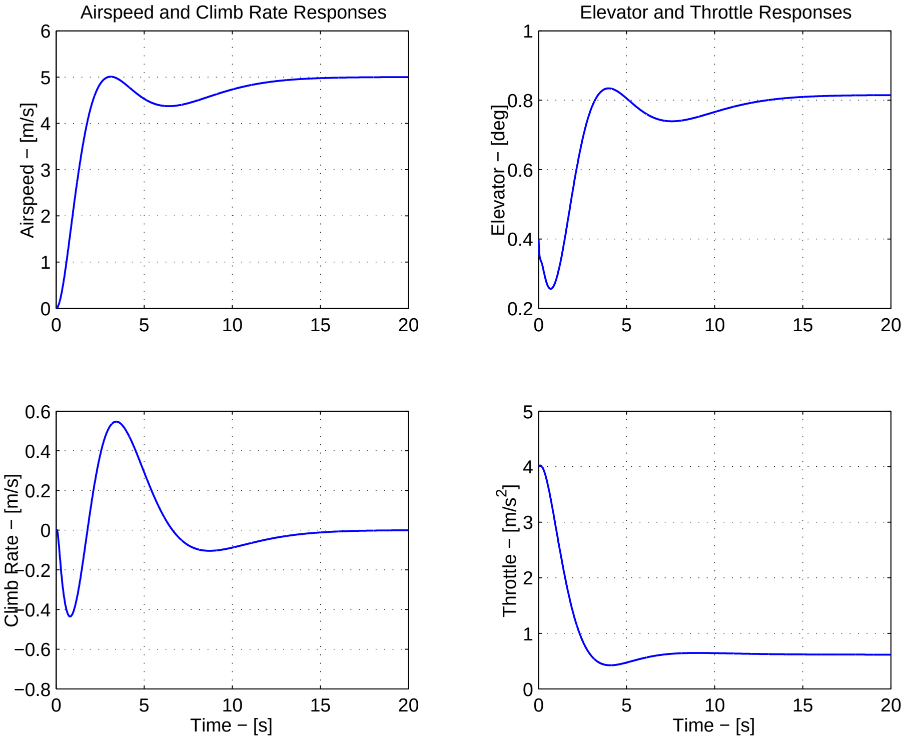


Figure 5.6: *Responses to 5m/s Step Command in Airspeed*

5.2.3 Altitude Controller

The altitude controller is the only longitudinal trajectory controller. The design approach is to generate a climb rate command from the altitude error signal, which means that only the extra integrator due to the height state need be stabilised by the feedback. The approach eases the design of the altitude controller because all of the dynamics specific to the aircraft, are already encapsulated by the airspeed and climb rate controller.

Before the height error signal is fed to the climb rate controller, it must be sent through

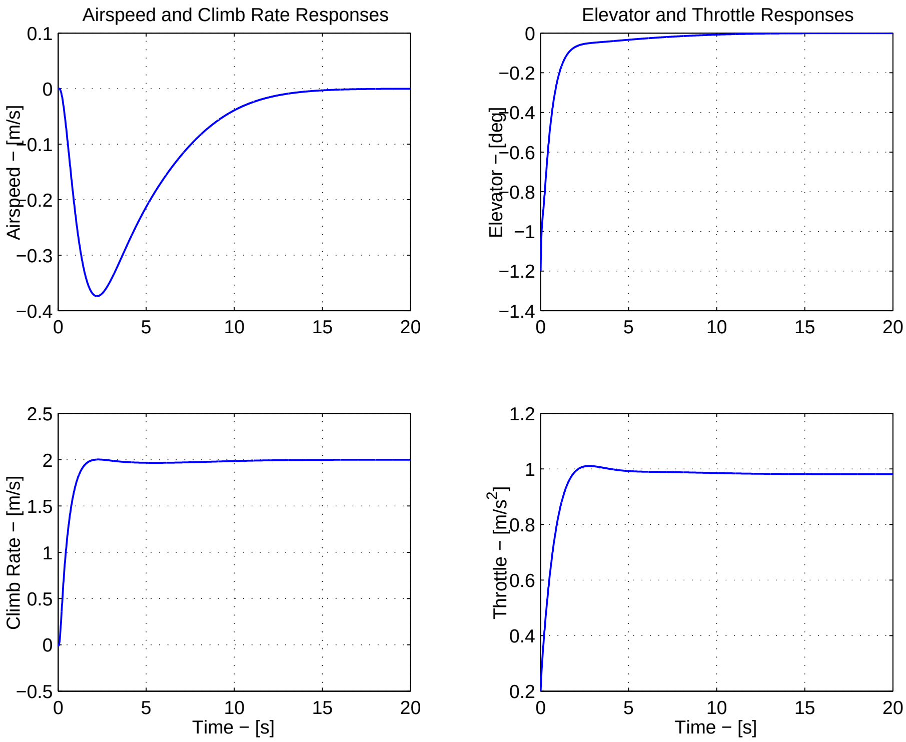


Figure 5.7: *Responses to 2m/s Step Command in Climb Rate*

a saturation block to ensure that it does not command climb rate signals in excess of ±2m/s. This block automatically ensures that the aircraft enters a constant climb rate mode when large altitude step commands are issued.

The specifications for an unsaturated altitude step response are,

- Rise time of under 6s
- Overshoot of less than 20%
- Zero steady state error

The rise time constraint is again not critical and is merely chosen as a reasonable value within the physical limitations of the aircraft. The low overshoot constraint essentially ensures a high phase margin. However, this is not critical in the design of the altitude controller, because once the airspeed and climb rate controller is implemented, a well known dynamic interface to the longitudinal aircraft system is presented. Finally, the extra integrator added as a result of the height state indicates that no added control effort is required to maintain a specific altitude and thus no compensator integrator term is required to maintain zero steady state error.

Controller Design

Consider the altitude controller block diagram of Figure 5.8 where the plant model is taken from equation (5.15). The output matrix ($\boldsymbol{C}$) is set to extract the climb rate state in metres per second (m/s). The climb rate then enters the natural integrator to become the height state in metres (m). It is this integrator that makes the system type 1. Note the saturation block that ensures the maximum climb rate command is not exceeded.

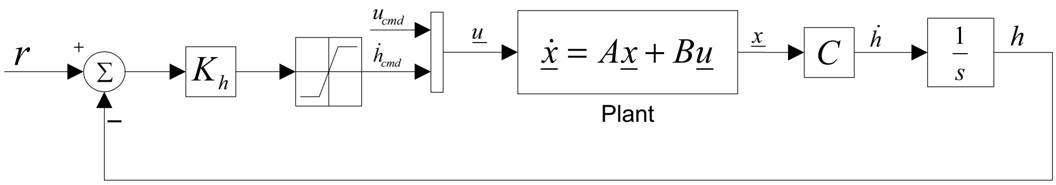


Figure 5.8: *Altitude Controller Block Diagram*

A root locus plot for the variation in the feedback gain (K_h) is shown in Figure 5.9. Note that the six poles shown correspond to the five poles of Figure 5.5 and the height state pole. A proportional feedback gain of,

$$K_h = 0.4 \tag{5.18}$$

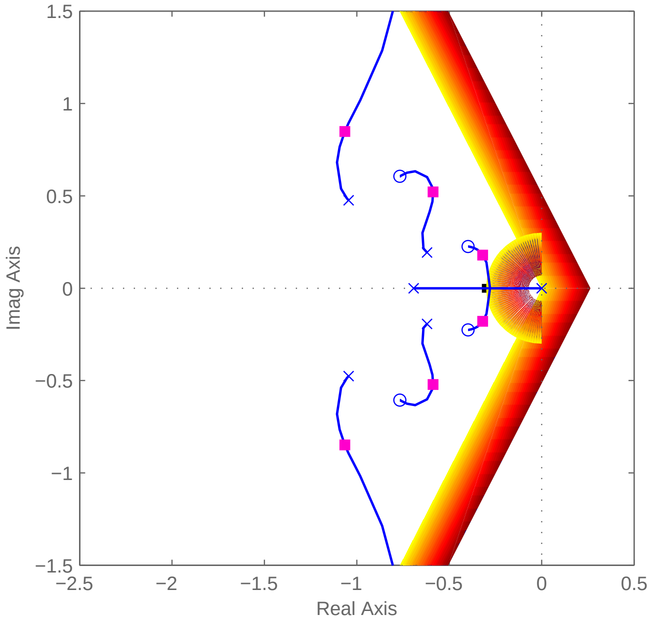


Figure 5.9: *Altitude Controller Root Locus*

is found to yield acceptable closed loop pole locations. This is a pure negative feedback loop which makes sense, since a positive altitude offset requires a negative climb rate command to stabilise the system. The closed loop system with the augmented height state is described by,

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} & -\boldsymbol{B}[0 \quad K_h]^T \\ \boldsymbol{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ h \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}[0 \quad K_h]^T \\ 0 \end{bmatrix} \mathrm{r} \tag{5.19}$$

where, r is the altitude reference command variable.

Linear Simulation

The responses to a 10m and a 40m step command in altitude are shown in Figure 5.10. The response to the 10m step command does not saturate the climb rate command signal at any time. It is clear from this plot that all specifications have been met. The response to the 40m step command shows how the saturation block automatically switches the aircraft into a constant climb rate mode for the majority of the step.

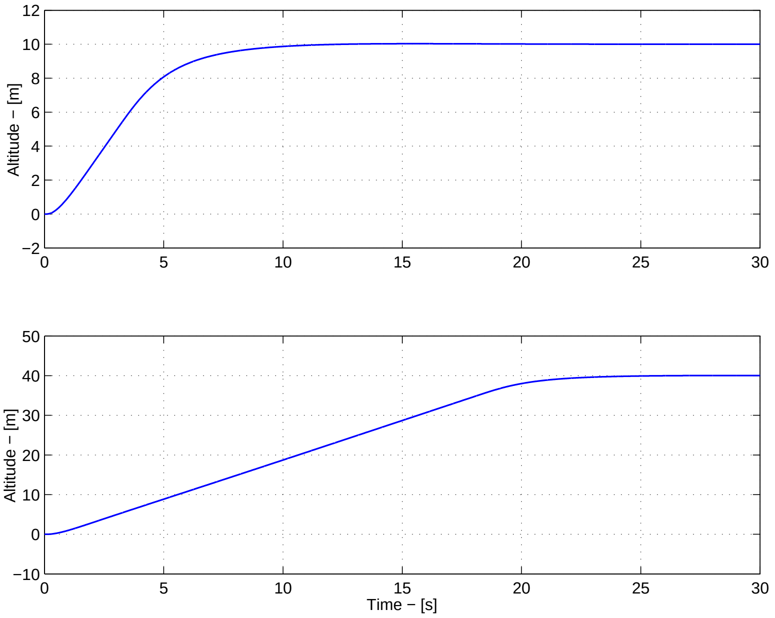


Figure 5.10: *Responses to 10m and 40m Step Commands in Altitude*

5.3 Lateral Controllers

The design and simulation of the aircraft's lateral controllers is considered in this section. Each controller design begins with a discussion of the design approach before the details of the controller are considered and the results verified by linear simulation.

5.3.1 Dutch Roll Damper

Modal analysis showed that the natural damping of the dutch roll mode is low. This natural damping is dependent on the fin moment arm length (l_F) and the aircraft velocity, both of which vary during flight. Thus a dutch roll damper is implemented in order to suppress the natural dynamics of the dutch roll mode, desensitise the overall yaw damping to variation in flight parameters and provide damping with respect to inertial space for wind gust disturbance.

Continuing along the same lines as the design of the pitch rate damper, the dutch roll damper is implemented by feeding a yaw rate signal back to the rudder. The rudder is used to provide the damping because it was shown by modal analysis to be 4 times more effective at controlling the dutch roll mode than the ailerons are. The yaw rate signal must however, be high pass filtered before it is fed to the rudder so that the damper does not counter constant turn rate motions. A secondary advantage of the high pass filter is that it negates the bias effects of low cost rate gyroscopes.

Controller Design

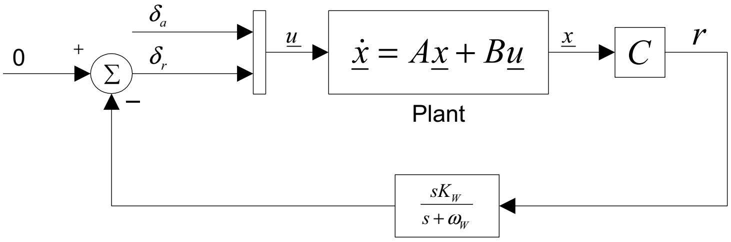

0
+
−
Σ
δ_a
δ_r
$\underline{u}$
$\dot{\underline{x}} = A\underline{x} + B\underline{u}$
Plant
$\underline{x}$
C
r
$\frac{sK_W}{s+\omega_W}$

Figure 5.11: *Dutch Roll Damper Block Diagram*

The control system is summarised in the block diagram of Figure 5.11 where the plant dynamics are taken from equation (4.27). The output matrix ($\boldsymbol{C}$) is set to extract the yaw rate state in degrees per second (deg/s). Note that the yaw rate signal moves through a washout [1] filter before being fed back to the rudder in a proportional fashion.

Choosing the washout filter cutoff frequency is an iterative process. However, it must be chosen low enough to ensure the frequency of the dutch roll mode lies within its passband (without too much phase error), but high enough so as not to interfere with constant turn rate motions. With the natural frequency of the dutch roll mode at 8.34rad/s, a washout filter cutoff frequency of 2rad/s is found to work well. With the filter dynamics included,

[1] High pass

the root locus plot for the variation of the filter feedback gain (K_W) is shown in Figure 5.12. The five poles visible are the natural modes of the aircraft and the washout state.

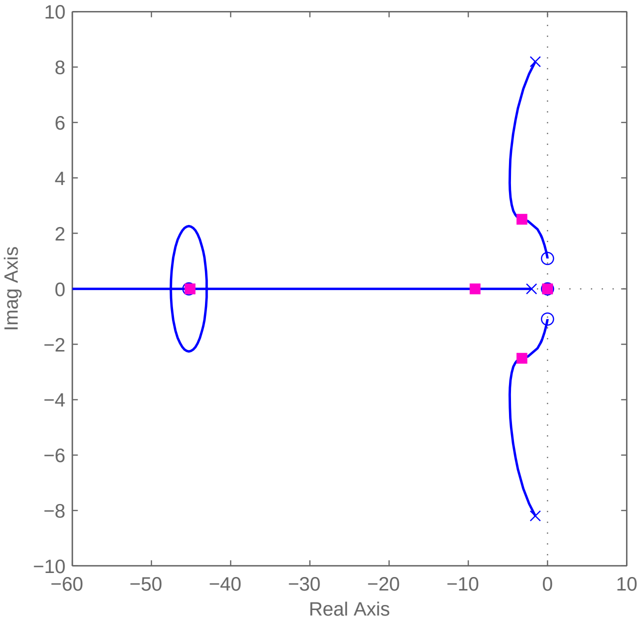


Figure 5.12: *Dutch Roll Damper Root Locus*

The filter gain is increased to provide maximum damping of the dutch roll mode for the cutoff frequency chosen. The complete washout filter with the feedback gain included is stated below,

$$D_W(s) = \frac{-0.2s}{s+2} \tag{5.20}$$

The closed loop system dynamics with the augmented washout state (x_W) are,

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathrm{x}}_W \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} - \boldsymbol{B}D_W\boldsymbol{C} & -\boldsymbol{B}C_W \\ B_W\boldsymbol{C} & A_W \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathrm{x}_W \end{bmatrix} + \begin{bmatrix} \boldsymbol{B} \\ \mathbf{0} \end{bmatrix} \mathbf{u} \tag{5.21}$$

where, A_W, B_W, C_W and D_W are the scalar state space representation matrices of equation (5.20).

Linear Simulation

The effect of the dutch roll damper can be verified by linear simulation. The yaw rate response of the aircraft with the dutch roll damper both armed and disarmed, and the lateral state vector initialised with the real part of the dutch roll mode eigenvector, is shown in Figure 5.13. It is clear that the dutch roll mode dynamics are well suppressed by the damper.

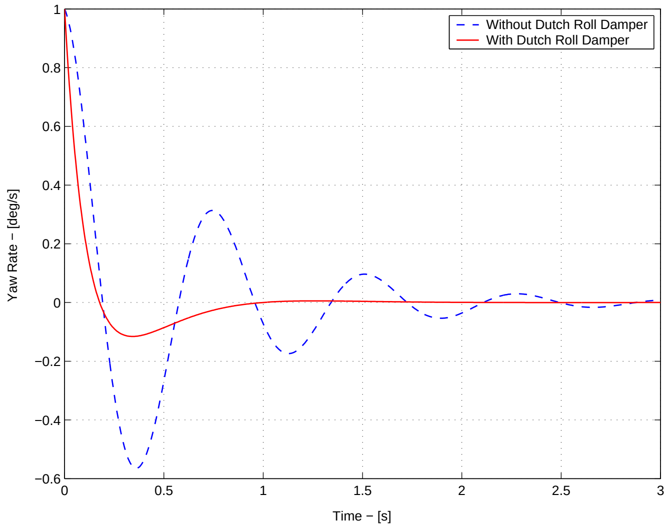


Figure 5.13: *Dutch Roll Damper Simulation*

5.3.2 Yaw Rate Controller

The yaw rate controller indirectly regulates the roll angle of the aircraft. The ailerons provide the best control over pure roll motions, as indicated by modal analysis of the roll mode. This suggests that a simple, effective approach to controlling the yaw rate, would be to feed a yaw rate signal back to the ailerons. However, it is desired to be able to command a certain yaw rate with zero steady state error. In practice, the aircraft is not entirely symmetrical and coefficients such as C_l will not be zero under symmetric flow conditions. Thus, integral control is required to negate the effects of the aircraft's asymmetry. However, root locus analysis shows that simple yaw rate feedback with integral control forces the system unstable. This can easily be remedied by providing roll rate feedback to the ailerons, to counter the large aileron commands that the integrator builds up and that drive the system unstable [4]. This issue will be treated in more detail in the controller analysis to follow.

The major limitation on the performance of the yaw rate controller is the aileron actuation limit. The aileron actuation limit is stated in Table 4.4 and restated below for convenience. This limit ensures that the linear actuation model remains valid.

- Maximum aileron deflection of 10° from the trim condition

The specifications for the yaw rate controller are listed below, where the maximum step response capability is chosen to ensure that the linear relationship between yaw rate and roll angle remains valid.

- To cater for yaw rate steps of up to 15deg/s
- Rise time of under 2s
- Overshoot of less than 20%
- Zero steady state error

The speed of response performance is again not critical for the scope of this project. It is desired to make the controller respond as quickly as possible within the physical limitations of the aircraft, so as to provide maximum disturbance rejection without saturating the ailerons or stimulating unmodeled dynamics. The specification is set slightly faster than that of the airspeed and climb rate controller because lateral manoeuvres merely require the direction of the lift vector to be altered and do not require any addition of energy to the system. The overshoot specification was imposed to ensure a high phase margin, to allow for modeling uncertainties in the plant.

Controller Design

Consider the controller block diagram of Figure 5.14 where the plant dynamics are obtained from equation (5.21). The output matrix ($\boldsymbol{C}$) is set to extract the roll and yaw rate states in degrees per second (deg/s). Note the integrator in the forward path that ensures the system is of type 1 and the two successive loop closures.

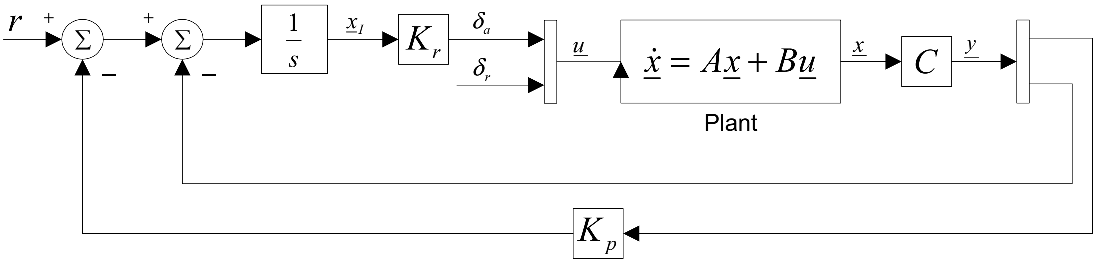


Figure 5.14: *Yaw Rate Controller Block Diagram*

The analysis begins with the yaw rate loop. The integrator dynamics are described by,

$$\dot{\mathrm{x}}_I = \mathrm{r} - [0 \quad 1]\boldsymbol{C}\mathbf{x} \tag{5.22}$$

where, x_I is the integrator state and r is the reference yaw rate command.

With the integrator dynamics included, the root locus plot for the variation in the yaw rate feedback gain (K_r) is shown in Figure 5.15. The five poles in the plot correspond to

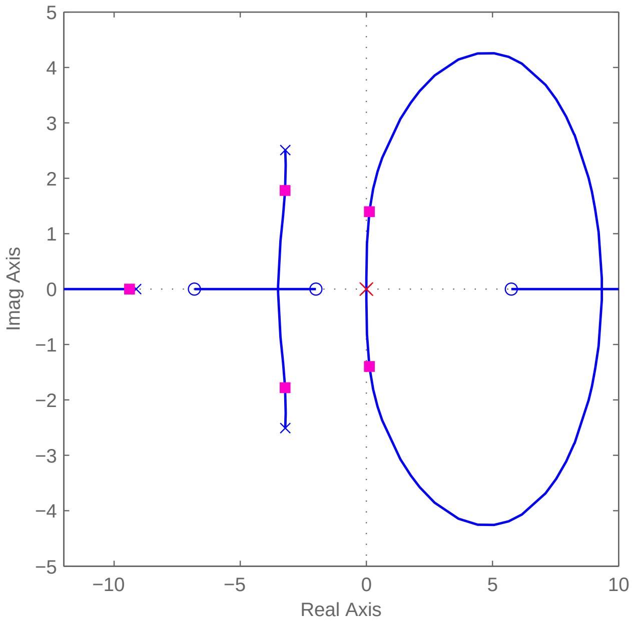


Figure 5.15: *Yaw Rate Loop Root Locus*

the spiral mode poles, the integrator state, the two damped dutch roll mode poles and the washout filter state. The roll mode pole is not visible in the plot. Note how pure yaw rate feedback drives the system unstable. If the yaw rate feedback gain is increased, more bandwidth is gained but it becomes more difficult for the roll rate loop to stabilise the system. Thus, in choosing the feedback gain, a compromise between the bandwidth gained and the degree of instability must be found. The yaw rate feedback gain was chosen in an iterative process with the roll rate feedback gain as,

$$K_r = -0.34 \tag{5.23}$$

With the yaw rate loop closed, the root locus plot for variation in the roll rate feedback gain (K_p) is shown in Figure 5.16.

Note how the roll rate feedback pulls the unstable poles back into the left half plane and that the bandwidth gained from the previous loop is largely maintained. With a roll rate feedback gain of,

$$K_p = 0.6 \tag{5.24}$$

the controller specifications are satisfied. With the integral state augmented and the two control loops in place, the closed loop dynamic equations can be written as,

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathrm{x}}_I \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}[K_r \quad 0]^T \\ -(\boldsymbol{C_r} + K_p\boldsymbol{C_p}) & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathrm{x}_I \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \boldsymbol{B}[0 \quad 1]^T \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \mathrm{r} \\ \delta_r \end{bmatrix} \tag{5.25}$$

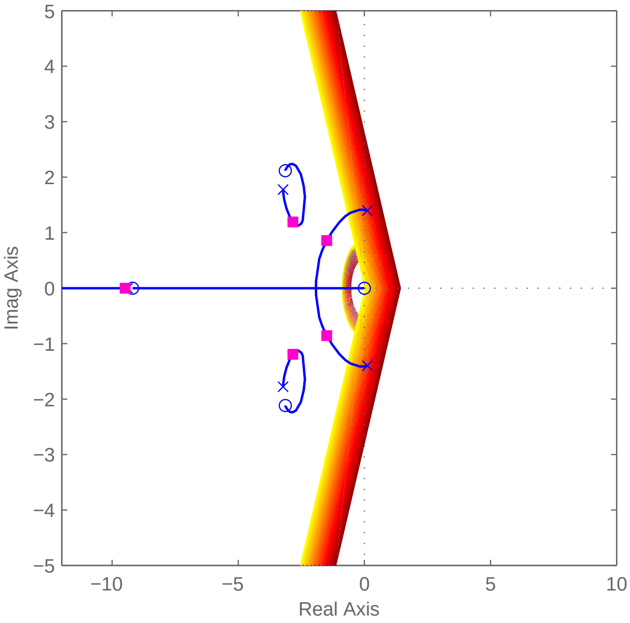


Figure 5.16: *Roll Rate Loop Root Locus*

where,

$$\boldsymbol{C_p} = [1 \quad 0]\boldsymbol{C} \tag{5.26}$$

$$\boldsymbol{C_r} = [0 \quad 1]\boldsymbol{C} \tag{5.27}$$

Integrator antiwindup was also implemented in this controller for the reasons stated in Section 5.2.2.

Linear Simulation

Figure 5.17 shows the yaw rate and aileron responses to a maximum yaw rate step command of 15deg/s.

Note that all specifications have been met but that there is adverse yaw at the onset of the turn. This is caused by differential drag due to the ailerons and quantified by the yaw moment due to aileron deflection control derivative (N_{δ_a}). To counter this effect, a human pilot typically provides a transient rudder input at the onset of a turn. Thus, all the controller need do is simulate this input.

Consider the block diagram of Figure 5.18 where the plant dynamics are defined by equation (5.25). Note that the yaw rate command is now also fed through a second order

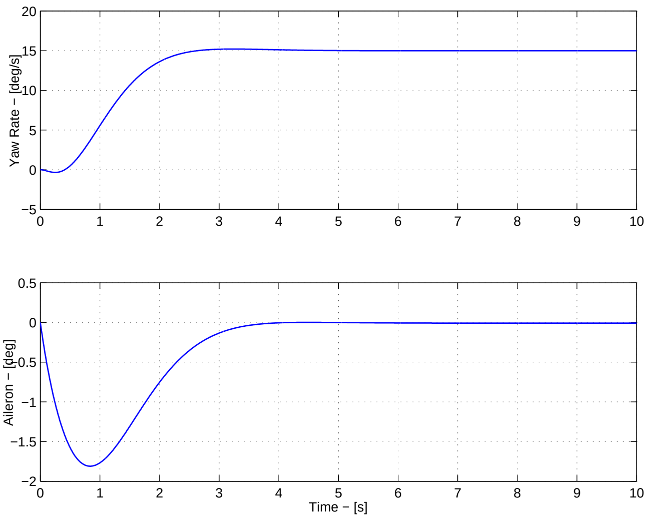


Figure 5.17: *Yaw Rate Controller Simulation*

washout filter to the rudder input. The filter is designed by iterative simulation of the yaw rate response until satisfactory cancellation of adverse yaw is attained. The following filter is found to work well,

$$D_W(s) = -\frac{0.2s}{(s+1)^2} \tag{5.28}$$

Note, that this controller solution to the adverse yaw problem is an example of open loop control. Thus, the controller performance depends heavily on the aircraft's dynamic model. However, the controller is not essential to the functioning of the yaw rate controller and can easily be fine tuned using flight test data.

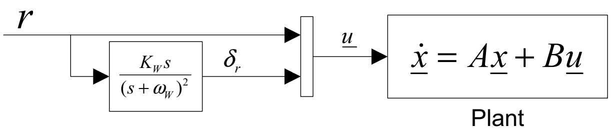

r
$\frac{K_W s}{(s+\omega_W)^2}$
δ_r
u
$\dot{\underline{x}} = A\underline{x} + B\underline{u}$
Plant

Figure 5.18: *Counter Adverse Yaw Block Diagram*

With the filter in place, the lateral system reduces to a single input system commonly know as a *yaw orientational control system*. The dynamics of this system are described by,

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{x}}_W \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}[0 \quad 1]^T \boldsymbol{C_W} \\ \boldsymbol{A_W} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_W \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}[1 \quad 0]^T + \boldsymbol{B}[0 \quad 1]^T \boldsymbol{D_W} \\ \boldsymbol{B_W} \end{bmatrix} \mathrm{r} \tag{5.29}$$

where, $\boldsymbol{A_W}$, $\boldsymbol{B_W}$, $\boldsymbol{C_W}$ and $\boldsymbol{D_W}$ are the matrices of the state space representation of equation (5.28). A plot of the yaw rate and rudder responses of the yaw orientational control system to a maximum yaw rate step command (15deg/s) is shown in Figure 5.19. It is clear that the adverse yaw has been cancelled with a rudder command that is far from its saturation limits stated in Table 4.4 (±10°).

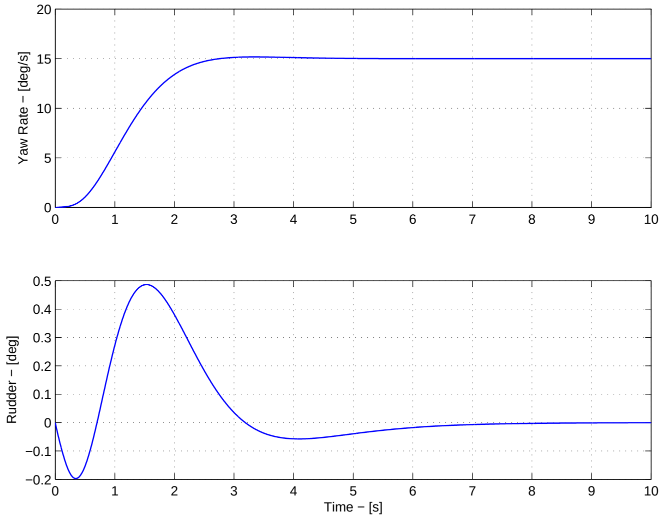


Figure 5.19: *Yaw Orientational Control System Simulation*

5.3.3 Heading Controller

The heading and guidance controllers form the two lateral trajectory controllers. Both controllers obtain their measurement information from the GPS receiver which has a maximum update rate of 4Hz with a measurement delay on the order of 1 cycle (0.25s) [23]. The controllers are designed directly in the discrete time domain because the measurement update rate is not orders of magnitude higher than the bandwidth of the closed loop yaw orientational control system. Figure 5.16 shows that the bandwidth of the yaw orientational control system is approximately 0.25Hz which is a factor 16 less than the 4Hz update rate, compared to the factor 30 suggested by [10] for accurate design by emulation. Designing in the discrete time domain also allows the measurement delay to be accurately accounted for.

The heading controller is the lateral counterpart of the altitude controller. The design approach is to generate a yaw rate command from the heading error signal, which means that only the extra integrator due to the heading state, need be stabilised by the feedback. This approach eases the design of the heading controller because all of the dynamics specific to the aircraft, are already encapsulated by the yaw orientational control system.

The heading error command is also sent through a saturation block, to ensure that the maximum yaw rate command of ±15deg is not exceeded. When large heading step commands are issued, the saturation block automatically enters the aircraft into a constant yaw rate mode for the majority of the turn.

The specifications for an unsaturated heading step response are,

- Overshoot of less than 20%
- Rise time of under 4s
- Zero steady state error

The rise time and overshoot constraints are chosen for the same reasons as stated in the altitude controller design. If biases on the rate gyroscopes used in the implementation of the yaw rate controller are ignored then the heading state's natural integrator implies that no compensator integrator term is required to maintain zero steady state heading error. However, if there is a bias on either the roll or yaw rate gyroscopes, then to maintain zero yaw rate, the yaw rate controller will require a constant command input to oppose the bias magnitude. Thus, to counter the gyroscope bias effects, a compensator integrator term is needed to provide the non-zero yaw rate command required to hold a constant heading. However, in this project, the compensator integrator was not implemented because the closed loop heading controller was able to suppress the heading errors due to the biases sufficiently (see simulation results later).

Controller Design

Consider the controller block diagram of Figure 5.20, where the plant dynamics are taken from equation (5.29). The output matrix ($\boldsymbol{C}$) is set to select the yaw rate perturbation state in degrees (deg/s). This state then enters the natural integrator to generate the heading state in degrees (deg).

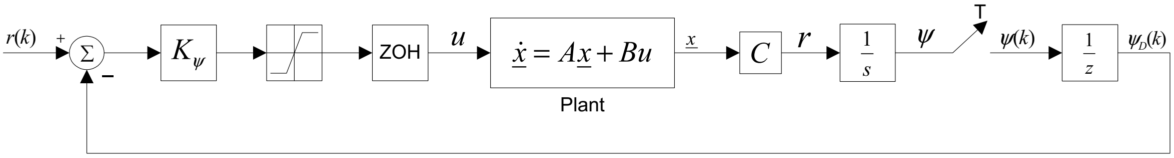


Figure 5.20: *Heading Controller Block Diagram*

Using MATLAB to incorporate the effect of the ZOH circuit, for a sample time (T) of 0.25s, into the continuous plant model and augmenting the discrete equivalent of the heading and delayed heading states to this model gives,

$$\begin{bmatrix} \mathbf{x}(\mathrm{k}+1) \\ \psi(\mathrm{k}+1) \\ \psi_D(\mathrm{k}+1) \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi} & \mathbf{0} & \mathbf{0} \\ T\boldsymbol{C} & 1 & 0 \\ \mathbf{0} & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(\mathrm{k}) \\ \psi(\mathrm{k}) \\ \psi_D(\mathrm{k}) \end{bmatrix} + \begin{bmatrix} \mathbf{\Gamma} \\ 0 \\ 0 \end{bmatrix} \mathrm{u}(\mathrm{k}) \tag{5.30}$$

where, $\mathbf{x}$(k) is the discrete state vector, which is exactly equivalent to the continuous state vector at each sample instant, u(k) is the discrete yaw rate command input, $\mathbf{\Phi}$ is the plant transition matrix, $\mathbf{\Gamma}$ is the discrete input matrix, ψ(k) is the discrete heading state and ψ_D(k) is the single cycle delayed discrete heading state.

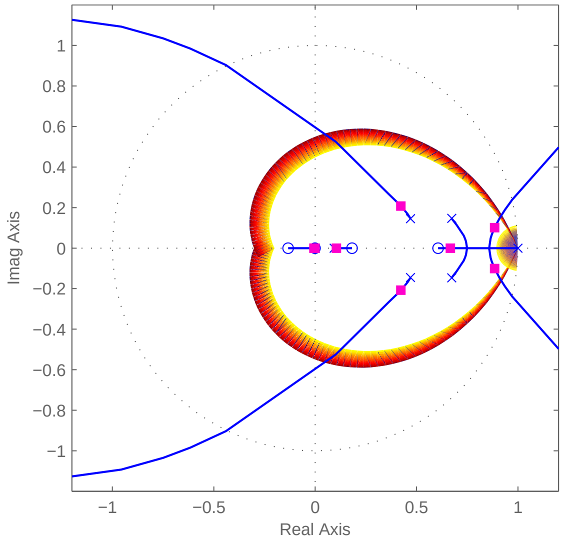


Figure 5.21: *Heading Controller Root Locus*

A discrete root locus, plotted in the z-plane, for variation in the heading feedback gain (K_ψ) is shown in Figure 5.21. The seven poles correspond to the five poles of Figure 5.16, the roll mode pole and the heading state. The system specifications are also shown on the plot and it is clear that with simple proportional feedback, the system poles are easily stabilised within the specification boundaries. A feedback gain of,

$$K_\psi = 0.3 \tag{5.31}$$

is found to satisfy the specifications. The closed loop system is described by,

$$\mathbf{x}(\mathrm{k}+1) = \left[\ \mathbf{\Phi} - \mathbf{\Gamma} K_\psi \boldsymbol{C}\ \right] \mathbf{x}(\mathrm{k}) + \left[\ \mathbf{\Gamma} K_\psi\ \right] \mathrm{r}(\mathrm{k}) \tag{5.32}$$

where, $\mathbf{x}$, $\mathbf{\Phi}$ and $\mathbf{\Gamma}$ are the state vector, system matrix and input matrix of equation (5.30) respectively and r(k) is the discrete heading command variable.

Linear Simulation

The responses to a 45° and a 180° step command in heading are shown in Figure 5.22. The 45° step command plot also shows the heading offset due to 0.5deg/s biases on the roll and yaw rate gyroscopes. These bias values are representative of the biases on low cost rate gyroscopes. With the biases, the heading offset is approximately 2°, which is acceptable for the purposes of this project. The response to the 180° step command shows how the saturation block automatically enters the aircraft into a constant turn rate mode for the majority of the step.

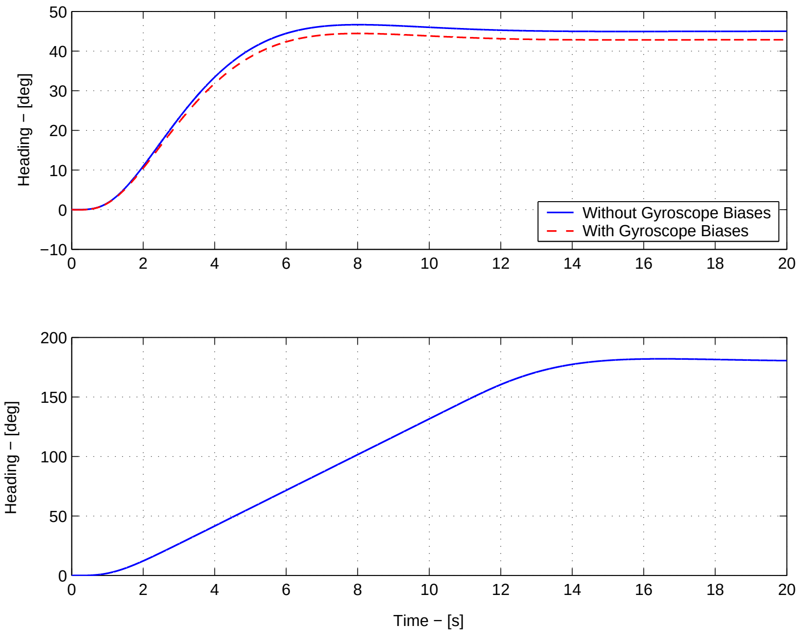


Figure 5.22: *Heading Controller Simulation*

5.3.4 Guidance Controller

The final controller to be designed is the guidance controller. The design approach is to generate a heading command from a cross track error signal, which means that only the extra integrator due to the cross track error state need be stabilised by the feedback. The cross track error is calculated using the GPS receiver's position measurements. The details of how the cross track error is calculated are presented in the Navigation section of the next chapter.

The specifications for the guidance controller are,

- Rise time of under 8s

- Overshoot of less than 20%
- Zero steady state error

The time and overshoot specifications are set for reasons similar to those of the heading controller. The lateral track state adds a natural integrator into the forward path that ensures zero steady state errors under zero gyroscope bias conditions. However, when the rate gyroscopes of the yaw rate controller have non-zero biases, then there will be a steady state cross track error that can be cancelled by the addition of an integrator term. An additional integrator term was not included in this project because the track errors with bias effects included are acceptably small. However, with the current successive loop closure controller architecture, the susceptibility of both the heading and guidance controllers to gyroscope biases can be nullified by the addition of a single, slow integrator term to the heading controller alone.

Controller Design

Consider the controller block diagram of Figure 5.23 where the plant dynamics are the continuous equivalent of equation 5.32. The output matrix ($\boldsymbol{C}$) is set to extract the derivative of the track state in metres per second (m/s), where it then moves through a natural integrator to become the track state in metres (m).

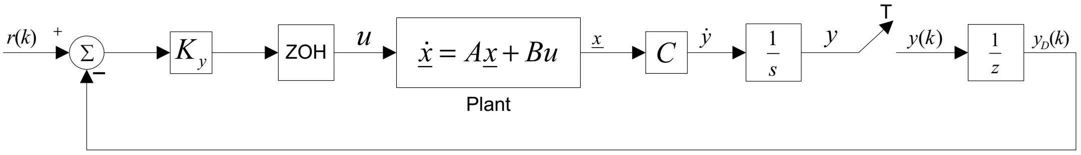


Figure 5.23: *Guidance Controller Block Diagram*

Using MATLAB to incorporate the effect of the ZOH circuit, for a sample time (T) of 0.25s, into the continuous plant model and augmenting the discrete equivalent of the lateral track and delayed lateral track states to this model gives,

$$\begin{bmatrix} \mathbf{x}(\mathrm{k}+1) \\ y(\mathrm{k}+1) \\ y_D(\mathrm{k}+1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi} & \mathbf{0} & \mathbf{0} \\ T\boldsymbol{C} & 1 & 0 \\ \mathbf{0} & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(\mathrm{k}) \\ y(\mathrm{k}) \\ y_D(\mathrm{k}) \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Gamma} \\ 0 \\ 0 \end{bmatrix} \mathrm{u}(\mathrm{k}) \qquad (5.33)$$

where, $\mathbf{x}$(k) is the discrete state vector, u(k) is the discrete heading command input, $\boldsymbol{\Phi}$ is the plant transition matrix, $\boldsymbol{\Gamma}$ is the discrete input matrix, y(k) is the discrete cross track error state and y_D(k) is the single cycle delayed discrete cross track error state.

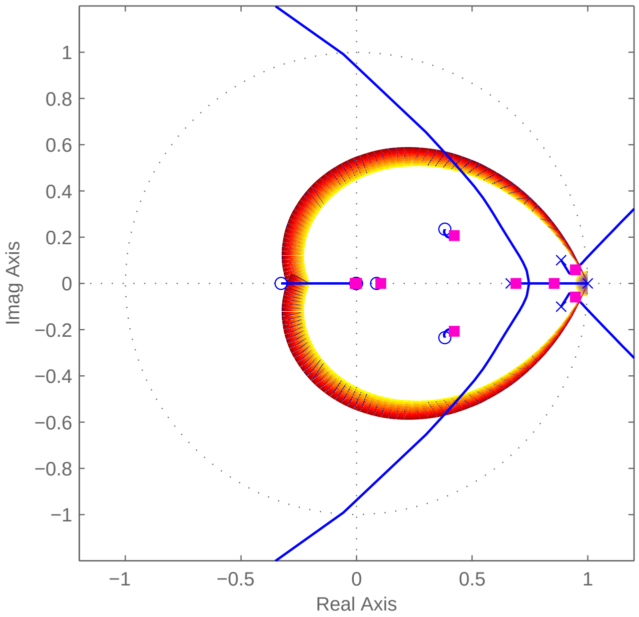


Figure 5.24: *Guidance Controller Root Locus*

A discrete root locus for the variation in compensator feedback gain is shown in Figure 5.24. The visible poles are those of Figure 5.21 and the cross track error state. A proportional feedback with a gain of,

$$K_y = 0.42 \tag{5.34}$$

is shown to meet the pole location specifications. The closed loop system can be written as,

$$\mathbf{x}(\mathrm{k}+1) = \left[\ \mathbf{\Phi} - \mathbf{\Gamma} K_y \boldsymbol{C}\ \right] \mathbf{x}(\mathrm{k}) + \left[\ \mathbf{\Gamma} K_y\ \right] \mathrm{r}(\mathrm{k}) \tag{5.35}$$

where, $\mathbf{x}$, $\mathbf{\Phi}$ and $\mathbf{\Gamma}$ are the state vector, system matrix and input matrix of equation (5.33) respectively and r(k) is the discrete lateral track command variable.

Linear Simulation

Responses to a 50m step command in track position are shown in Figure 5.25. The track offset due to roll and yaw rate gyroscope biases of 0.5deg/s is shown to be approximately 5m. This offset is acceptable for the purposes of this project, where manoeuvres such as landing that require strict cross track error control are not required.

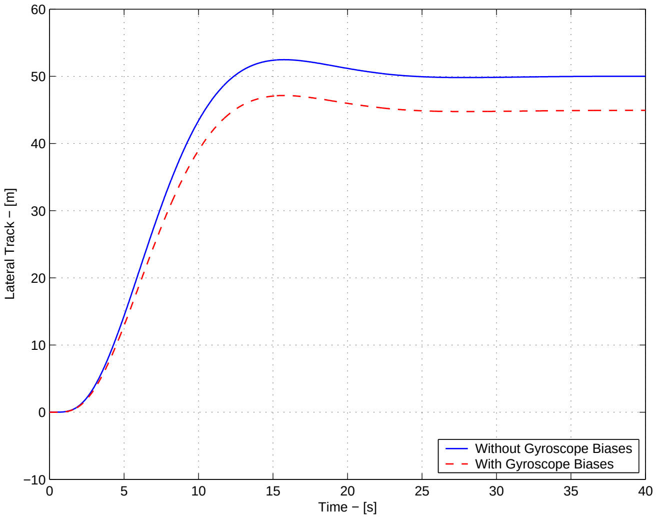


Figure 5.25: *Guidance Controller Simulation*

5.4 Summary

The design of all of the controllers was presented in this chapter. A successive loop closure technique was used beginning with stability augmentation, then attitude regulation and finally trajectory control. All controllers were designed to work with low cost sensors and to minimise the dependency of the controller performance on the accuracy of the aircraft's dynamic model. The control algorithms developed require little processing power for implementation and are capable of regulating all motion variables desired for conventional flight.

Chapter 6

Navigation and Non-linear Simulation

This chapter can be divided into two parts. In Section 6.1, a navigation algorithm capable of generating the cross track error required by the guidance controller is developed. The algorithm consists of a path planner that generates a valid path between two waypoints and a cross track error algorithm that calculates the error from the generated path. In Section 6.2, the block diagram and graphical non-linear simulators that were developed during this project, are discussed with reference to the implementation problems they helped solve. These simulators played a key role in the preparing the controllers of Chapter 5 for practical implementation and are essential tools required for the successful achievement of autonomous flight in a minimum number of test flights.

6.1 Navigation

6.1.1 Path Planner Algorithm

The path planner algorithm is responsible for determining a valid path between two waypoints. The definition of a waypoint, for the context of this project, is a structure containing the following information,

- A 3D location in inertial axes
- A heading
- An airspeed

The path planner algorithm should be capable of accepting two waypoints and determining a valid flight path between them. An algorithm developed in [33] offers a simple, yet effective solution and was used directly in this project. Given a starting and ending waypoint, the algorithm finds the shortest path made up of an initial circle, then a straight line and finally another circle, that leads from one waypoint to the next, such that the starting and ending course tangents are coincidental with the respective heading directions. The circle radius can be set arbitrarily and was made 100m in this project, so as not to exceed the maximum yaw rate command of 15deg/s at the aircraft trim speed of 20m/s.

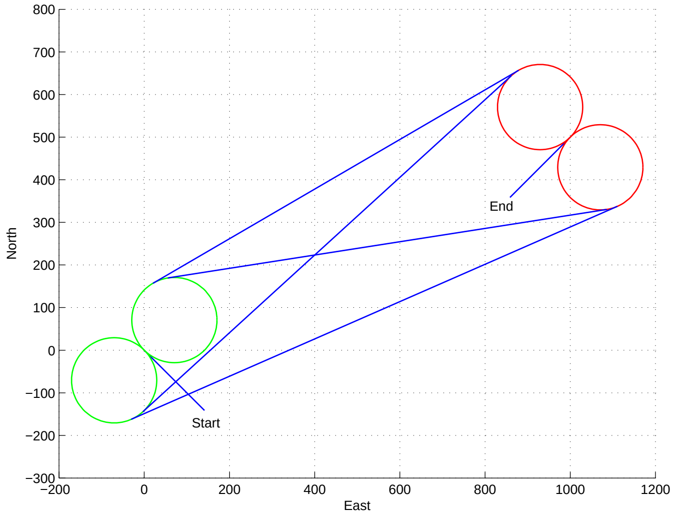


Figure 6.1: *Path Planner Graphical Output*

Given two waypoints, the typical path outputs that the algorithm generates are shown in Figure 6.1. Note that a maximum of four possible path options is generated and the algorithm automatically chooses the shortest route and then returns the following information.

- C_{CD}: 2D Circle centre for departure
- C_{CA}: 2D Circle centre for arrival
- Rad: Circle radius
- P_D: 2D Point of departure from initial circle to straight line
- P_A: 2D Point of arrival from straight line to final circle
- L_{SL}: Length of the straight line
- H_{SL}: Straight line heading unit vector

- H_F: Final heading unit vector

The information fully describes the path to be followed and provides the data required by the the cross track error algorithm.

6.1.2 Cross Track Error Algorithm

Now that a valid path between two waypoints can be calculated, it is required to develop an algorithm capable of determining the aircraft's cross track error from the path, so that it can be fed to the guidance controller of Chapter 5. Given the path planner data, the cross track error algorithm must be capable of identifying which section of the path the aircraft is on (initial circle, straight line or final circle) as well as the corresponding cross track error. These two items are discussed qualitatively in the next two sections.

Path Section Identification

The key to identifying the path section, lies in identifying the transition from one section to another. Only two types of transitions ever need be identified,

1. Transition from a circular section
2. Transition from a straight line section

Identifying the transition from a circular section is extremely simple. The aircraft's heading vector can constantly be compared to the departure heading of the circle (H_{SL} or H_F) and the transition made when the heading difference is within an acceptably small bound. A bound of 10° was used in this project.

Transition from a straight line can be identified by centering the inertial coordinate system at the start of the straight line (P_D) and rotating it such that the straight line heading lies along the new inertial OX-axis. Then, the OX-axis coordinate of the aircraft's transformed location can be monitored and the transition made when it exceeds the length of the straight line (L_{SL}).

Cross Track Error Calculation

With the current path section identified, the cross track error needs to be calculated. Again, two scenarios exist,

1. Cross track error calculation for a circular section
2. Cross track error calculation for the straight line section

The cross track error calculation for the circular section can be attained by subtracting the straight line distance between the current aircraft location and the circle centre location (current radius) from the desired radius (Rad). Care must be taken with the sign of the error as it depends of the direction of turn of the circle.

The calculation of the cross track error for the straight line section can again be done by centering the coordinate system at the start of the straight line (P_D) and rotating it such that the straight line heading lies along the new inertial OX-axis. The cross track error is then merely the OY-axis coordinate of the transformed aircraft location.

6.1.3 Navigator

The algorithms described in the previous two sections provide the final tools necessary for autonomous navigation. An overview of the navigation algorithms is shown in the flow chart of Figure 6.2 and the complete system is referred to as the *Navigator*.

Note that when the path planner algorithm is called, the *current* aircraft location and heading are used as the data for the initial waypoint which allows the navigator to be turned on anywhere in flight. The airspeed and altitude values are commanded from the starting waypoint data each time the path planner algorithm is called. It is clear from Figure 6.2 that, together with the control algorithms of Chapter 5, all that need be provided for autonomous navigation is a list of waypoints.

6.2 Non-linear Simulation

The control and navigation algorithms developed up to now are still a long way from the implementation stage. The entire autopilot system needs to be thoroughly simulated if success is to be achieved in a minimum number of flights tests. Two non-linear simulators were built for this purpose.

1. Block Diagram Simulator
2. Graphical Simulator

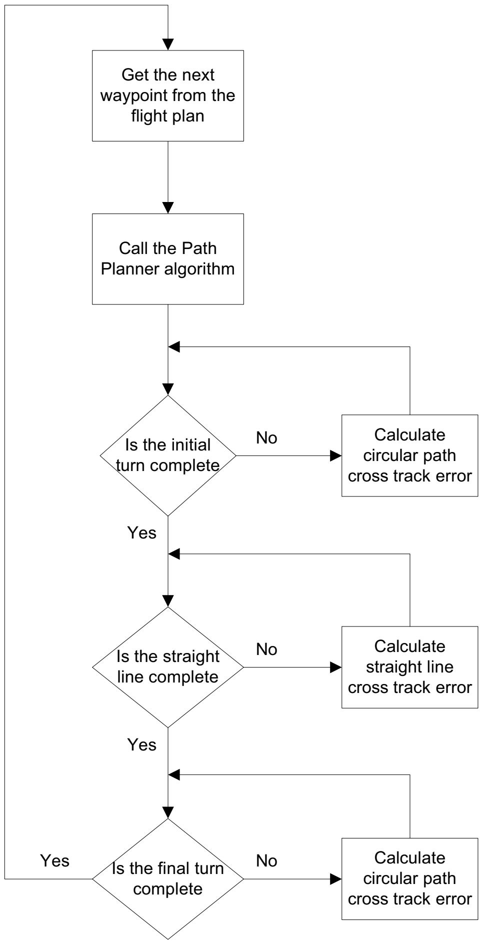

Get the next waypoint from the flight plan
Call the Path Planner algorithm
Is the initial turn complete
No
Calculate circular path cross track error
Yes
Is the straight line complete
No
Calculate straight line cross track error
Yes
Yes
Is the final turn complete
No
Calculate circular path cross track error

Figure 6.2: *Navigator Flow Chart*

The block diagram simulator was built in Simulink and is based on the non-linear aircraft dynamic model developed in Chapter 2, summarised by equations (2.67) to (2.74), with the airfoil coefficient models of equations (3.58) to (3.64). A non-linear wing lift curve slope was used to include stall effects. The advantage of this simulator over the graphical simulator lies in the block diagram structure, which gives good visual insight into how the complete aircraft model and control algorithms fit together. The graphical simulator uses the same non-linear equation core as the block diagram simulator, but was written in C-code and uses the OpenGL graphics library to visually display the aircraft as it moves through its surroundings. This simulator is an extension of a previous glider simulator by [34], that essentially displayed a tug plane towing a glider on the screen, with an extremely simplified motion model. The major advantage with this type of simulator lies in its visual interface. The interface allows for real time interaction with the aircraft and for the autopilot to be visually scrutinised.

Each simulator is now discussed in more detail with emphasis placed on the specific implementation problems that it solved.

6.2.1 Block Diagram Simulator

The lowest level of the block diagram simulator is shown in Figure 6.3. The *Control Processor* block is shown on the left with the aircraft sensor data entering it and the servo commands leaving. The *Aircraft, Actuator and Sensor Model* block is shown on the right with the servo commands entering and the sensor data leaving.

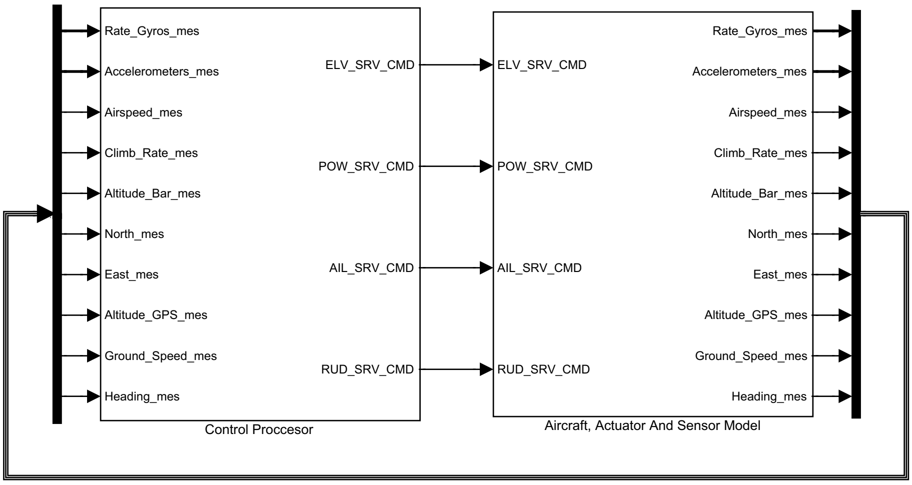


Figure 6.3: *Lowest Level of Block Diagram Simulator*

The expansion of the *Control Processor* block is shown in Figure 6.4. The *Analog to Digital Conversion* block is shown on the left and contains scaling, offset, discretisation and quantisation blocks specific to the analog to digital conversion process of the control microprocessor used. The *Autopilot* block contains all of the control algorithms developed in Chapter 5.

The expansion of the *Aircraft, Actuator and Sensor Model* block of Figure 6.3 is shown in Figure 6.5. The *Servo Actuator Model* and *Sensor Model* blocks contain the noise and dynamic models of the servos and sensors respectively. The *Non-Linear Aircraft Model* block is the core of the simulator and contains the full aerodynamic, gravitational and engine force and moment models feeding the non-linear aircraft dynamic equations of Chapter 2.

The expansion of the *Non-Linear Aircraft Model* block is shown in Figure 6.6. This figure gives an excellent overview of the aircraft dynamic model. The expansion of the blocks in Figure 6.6 would reveal the details of the aircraft model equations developed in Chapters

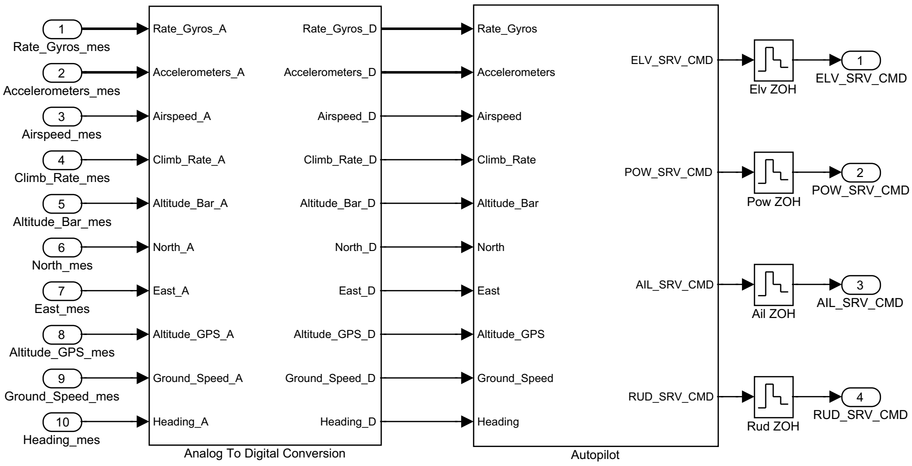


Figure 6.4: *Control Processor Block of Block Diagram Simulator*

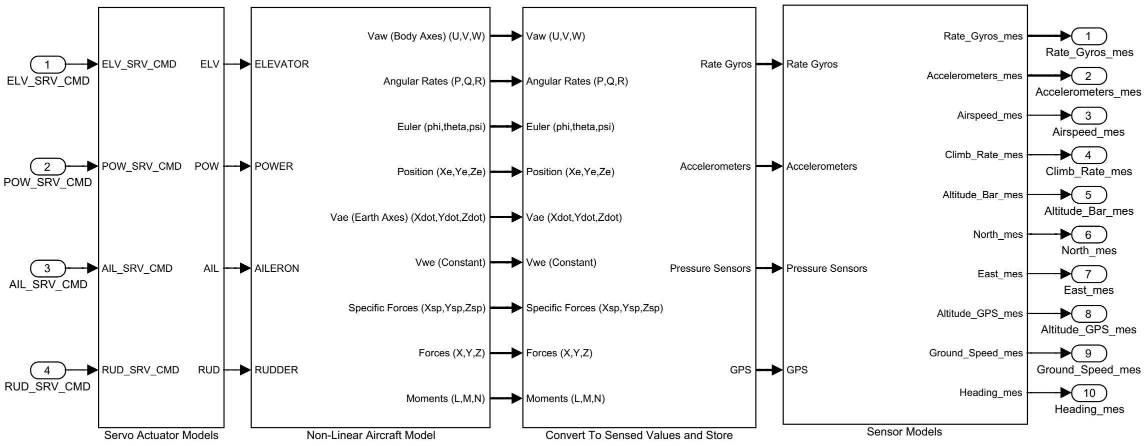


Figure 6.5: *Aircraft, Actuator and Sensor Model Block of Block Diagram Simulator*

2 and 3.

Armed with a brief overview of the simulator, some of the implementation problems that it was used to solve are discussed.

Longitudinal and Lateral Coupling Effects

The block diagram simulator makes it possible to simultaneously test the longitudinal and lateral controllers and note any cross coupling effects. Because the aircraft is modeled as symmetric about the OX_BZ_B-plane, pure longitudinal motion does not induce any lateral aircraft motion. Thus, the simulation of the longitudinal controllers is not shown here since the results are very similar to those obtained in Chapter 5. However, when any lateral manoeuvre is commanded, the aircraft's lift vector moves out of the plane

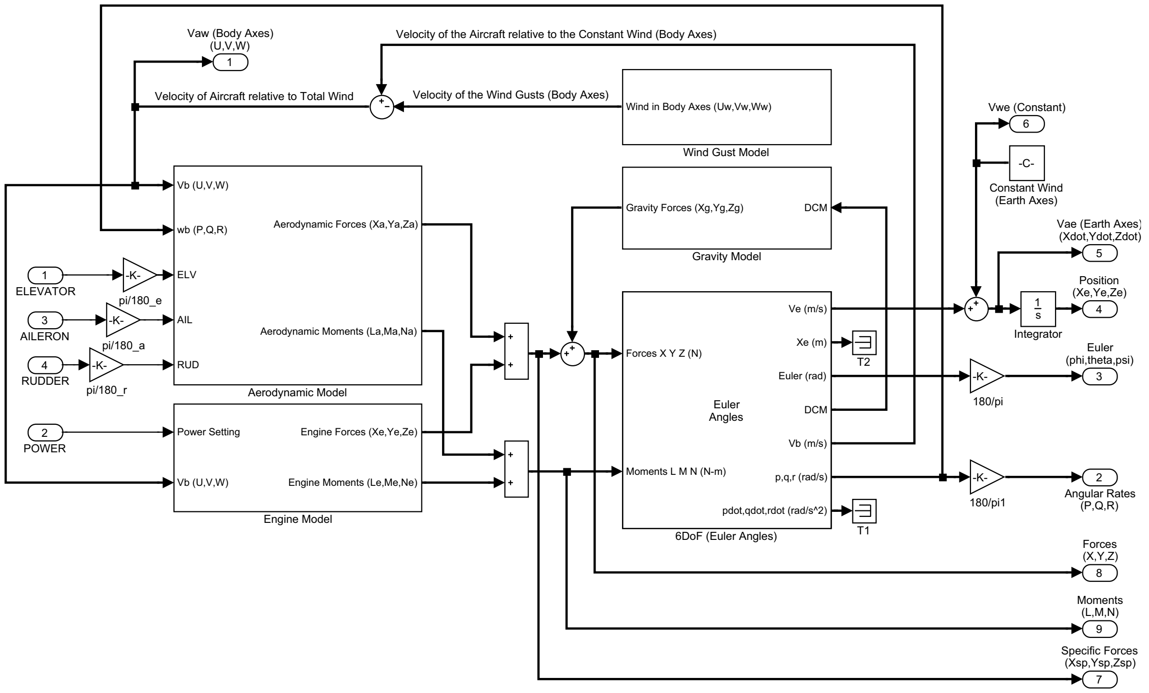


Figure 6.6: *Aircraft Model Block of Block Diagram Simulator*

of symmetry to provide the required lateral acceleration, and consequently more lift is required if the same airspeed and climb rate (or altitude) are to be maintained.

Figure 6.7 shows the yaw rate, airspeed and climb rate responses to a maximum step command in yaw rate (15deg/s).

There are two major differences between the yaw rate response of Figure 6.7 and that of Figure 5.19. Firstly, there is a slight deterioration in the transient response of Figure 6.7, as the airspeed and climb rate controller responds to restore level flight during the turn. Secondly, there is a very small steady state error on the yaw rate response of Figure 6.7. To gain insight into the origin of the steady state error, consider Figure 6.8, where all of the aircraft's angular rates for a 15deg/s yaw rate step response are shown plotted on the same set of axes.

In order to provide the extra lift required for the turn while maintaining the trim velocity, the angle of attack must be increased. In the steady state, the angle of attack change corresponds directly to a pitch angle change. With a steady state pitch angle, a constant inertial turn rate ($\dot{\psi}$) resolves into all three body axis angular rates. This is shown by equation (2.34) and can be seen in Figure 6.8. The non-zero roll rate is what causes the steady state offset in yaw rate. With reference to the yaw rate controller block diagram of Figure 5.14, the system will only settle when the input to the integrator is zero. With a constant reference input, this implies that a steady state roll rate will cause a steady

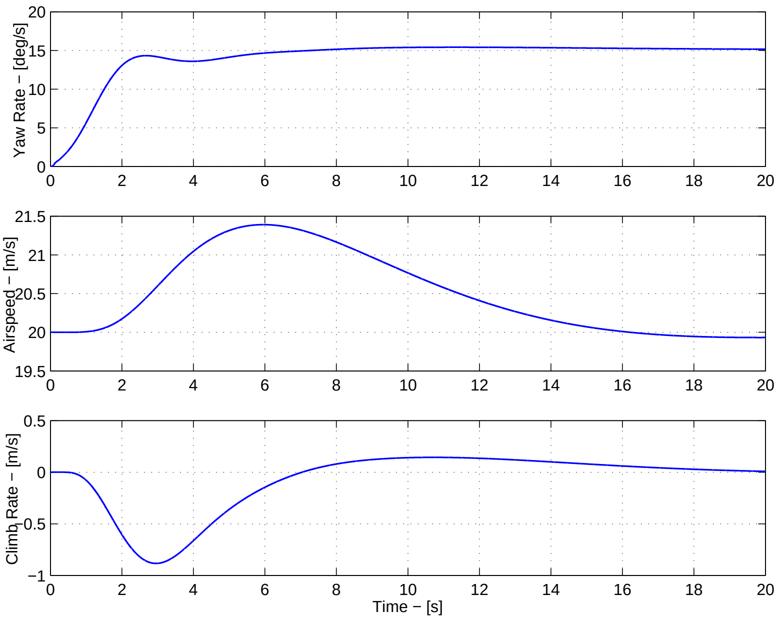


Figure 6.7: *Yaw Rate, Airspeed and Climb Rate Responses to a 15deg/s Yaw Rate Step Command*

state yaw rate error. Note, that the yaw rate offset will be amplified during a climb or decent due to its dependence on the aircraft's pitch angle.

Despite the slight deteriorations in the yaw rate response, the controllers appear to perform well together under coupled, non-linear conditions. This verifies the usefulness and applicability of the small disturbance theory developed in Chapter 3 and instills confidence in the controller design.

Controller Gain Conversion

The controller gains calculated in Chapter 5 are designed to work with sensor measurement and actuator command values in the units defined in Chapter 4 and in the controller development of Chapter 5. However, when the controller is implemented practically on a microprocessor, the sensor signals that enter the microprocessor and the actuator signals that leave it, are not likely to be in the desired units. Thus either the sensor and actuator signals must be scaled in the microprocessor, such that they work directly with the controller gains designed, or the control algorithms must be modified to work directly with the natural microprocessor units.

The first approach is simpler and safer since the control gains do not have to be altered and the signal units are well known. However, the drawback to this approach, is that it most often forces the use of floating point calculations to maintain precision throughout

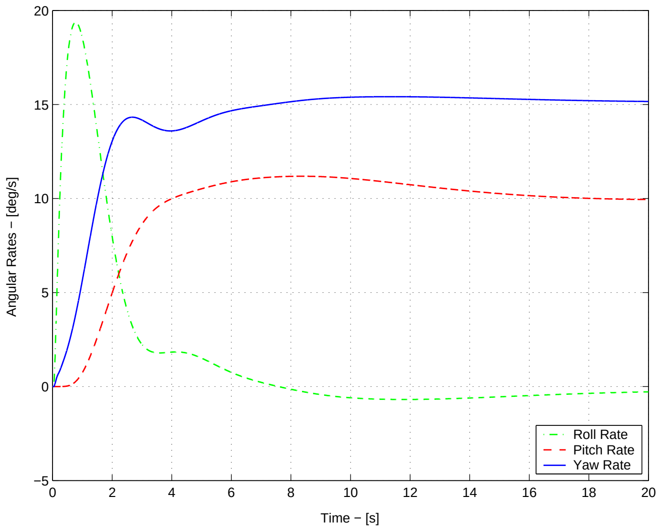


Figure 6.8: *Roll, Pitch and Yaw Rate Responses to a 15deg/s Yaw Rate Step Command*

the controller code execution. The control microprocessor used in this project (discussed in detail in Chapter 7), does not have hardware floating point calculation capabilities and thus this approach would introduce unacceptable latency into the control system. Thus, the second approach was adopted because it avoids the necessity for floating point calculations. This approach does however require the control algorithm gains and offsets to be altered. The block diagram simulator presents an excellent environment in which to develop and verify the necessary scaling and offset values for the controller.

The *Analog to Digital Conversion* block of Figure 6.4 is shown expanded in Figure 6.9. Note that the sensor signals that enter the block are in the units required by the control algorithms of Chapter 5. These signals are then converted to the microprocessor units using a single gain and offset and then enter the control system. An extract from the *Airspeed and Climb Rate Controller* block, found inside the *Autopilot* block of Figure 6.4, is shown in Figure 6.10. This figure shows the feedback gains from the pitch rate damper and airspeed and climb rate controller to the elevator. Note how the controller gains have been modified and are now preceded by the inverse of the sensor gain and succeeded by the inverse of the actuator gain. The actuator offset is then added to produce the signal that leaves the microprocessor to drive the servos. All of the autopilot control gains and offsets were modified in a similar fashion. An expansion of the *Servo Actuator Model* block of Figure 6.5 is shown in Figure 6.11. Note how the actuator offsets are removed and the signal scaled by the actuator gains to produce the actuator command in the units of the control system design of Chapter 5. These values then continue to the *Non-Linear Aircraft Model* block of Figure 6.5.

The block diagram simulator was used to debug the system with the implementation

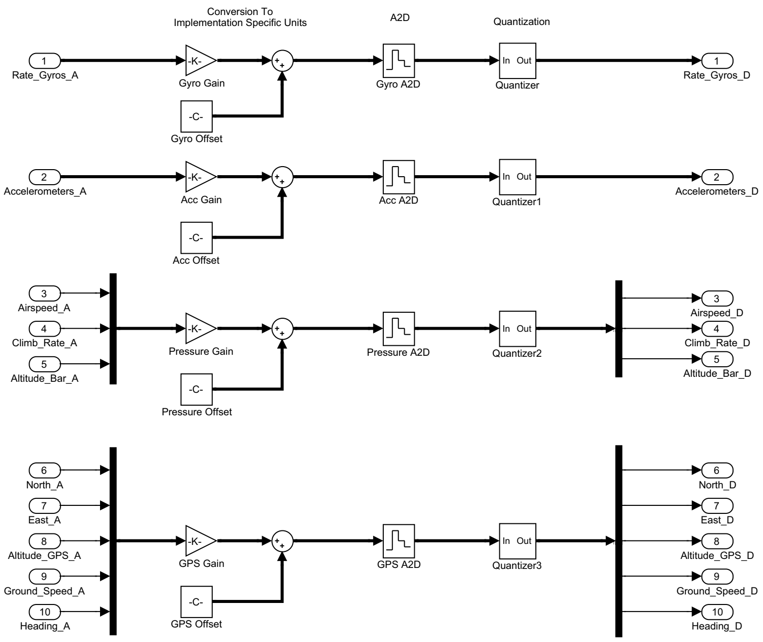


Figure 6.9: *Analog To Digital Conversion Block of Block Diagram Simulator*

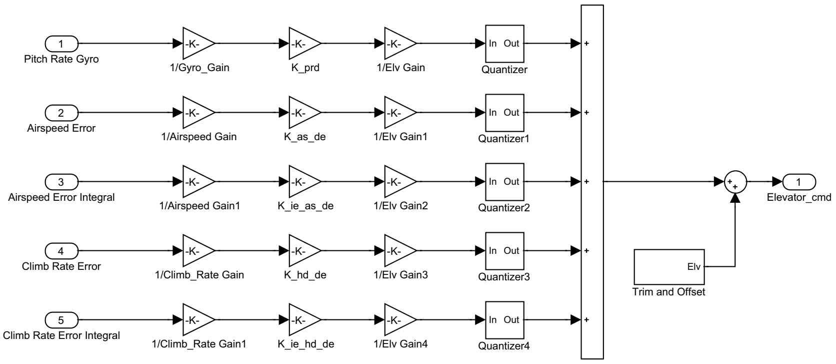


Figure 6.10: *Elevator Gains Block of Block Diagram Simulator*

specific scaling gains and offsets included, until exactly the same results were attained as the results before their inclusion. This verified the correctness of the new controller gains and offsets, and brought it one step closer to its final implementation.

Quantisation Effects

The controller modifications of the previous section allow the entire control system to be implemented without the need for floating point calculations. However, not using floating point numbers means that every time a calculation is completed, a quantisation error is made. These errors can degrade the performance of the controller if intermediate control signals are not properly scaled. To investigate the effects of quantisation errors,

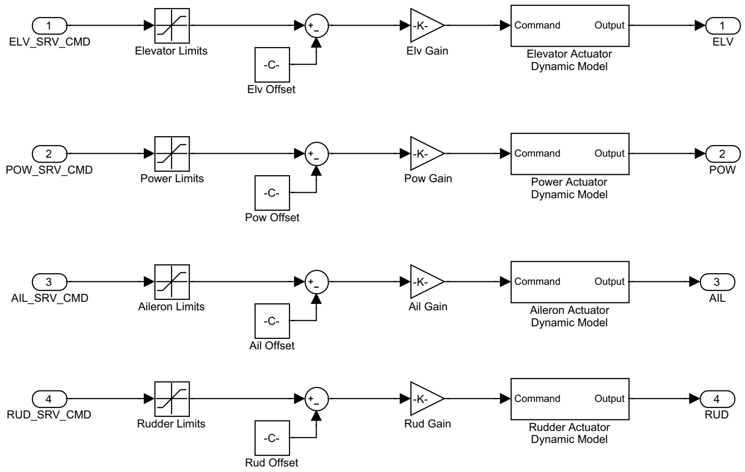


Figure 6.11: *Actuator Model Block of Block Diagram Simulator*

quantisation blocks were placed after every gain sequence. These blocks are visible in Figures 6.9 and 6.10. With these blocks in place, the control system was analysed and appropriate signals scaled until its performance with quantisation error effects included was deemed acceptable.

The response to a climb rate step command of 2m/s, with quantisation effects both enabled and disabled is shown in Figure 6.12. Note that when quantisation effects are enabled, the response is slightly degraded but still acceptable. The simulation result indicates that the controller signals are well scaled for fixed point calculations.

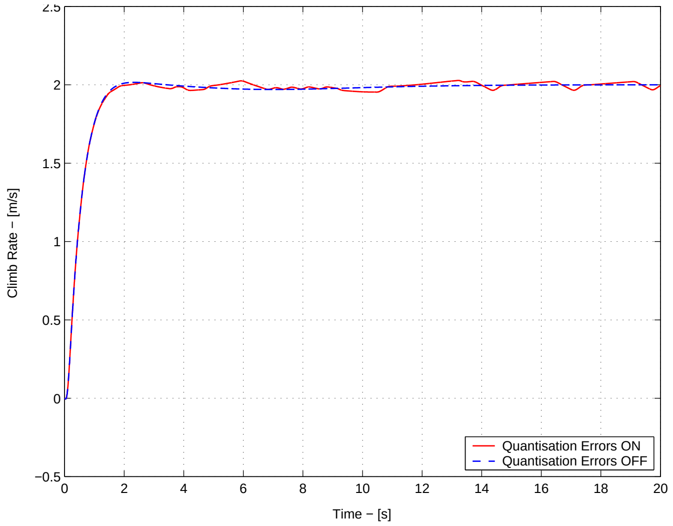


Figure 6.12: *Quantisation Example - Climb Rate Step Response*

Wind Effects

The effects of constant wind and wind gusts on the controllers can be easily analysed in the block diagram simulator. However, constant winds have no dynamic effect on the aircraft and so only wind gusts are considered. A stochastic model for aircraft wind gusts, coordinated in body axes, is presented in [3] and used directly in the simulator. The wind gust entry point is shown in Figure 6.6. With the airspeed, altitude and heading controllers armed and a RMS wind gust velocity in each of the body axis directions of 2m/s (chosen as 10% of the trim airspeed), the regulated airspeed, altitude and heading is shown in Figure 6.13. The results show an RMS airspeed error of 1.5m/s, a RMS altitude error of 0.6m and an RMS heading error of 4.4deg. With the airspeed perturbations posing no threat to the aircraft stalling (stalls at approximately 10m/s), the results are deemed acceptable for the purposes of this project.

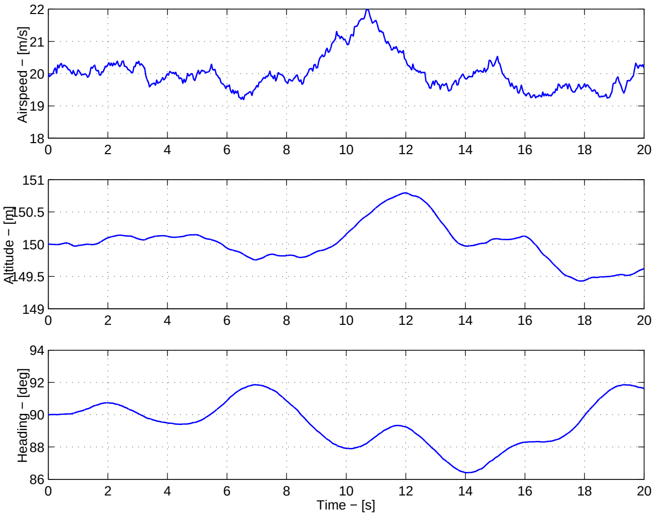


Figure 6.13: *Airspeed, Altitude and Heading Responses to RMS Wind Gusts of 2m/s*

Sensor Noise

The effect of sensor noise is also easily analysed in the simulator. The *Sensor Model* block of Figure 6.5 contains the noise models of all of the sensor used for feedback in the controllers. The noise models are formed from band limited white noise blocks, the characteristics of which were obtained from datasheets, practical measurements and [23]. The RMS noise value used for each sensor is stated in point form below.

- Rate Gyroscopes: 0.2 deg/s
- Airspeed Sensor: 0.5 m/s

- Climb Rate Sensor: 1 m/s
- Altitude Sensor: 0.5 m
- GPS Heading: 0.5 deg
- GPS 2D Position: 2 m

The simulation results, with the noise models activated and the airspeed, altitude and heading controllers armed are shown in Figure 6.14. The results show an RMS airspeed error of 0.07m/s, a RMS altitude error of 0.13m and an RMS heading error of 0.06deg. These controller results are more than acceptable for the purposes of this project. Note however, that the noise models do not include the vibration induced sensor noise that will be evident during practical test flights.

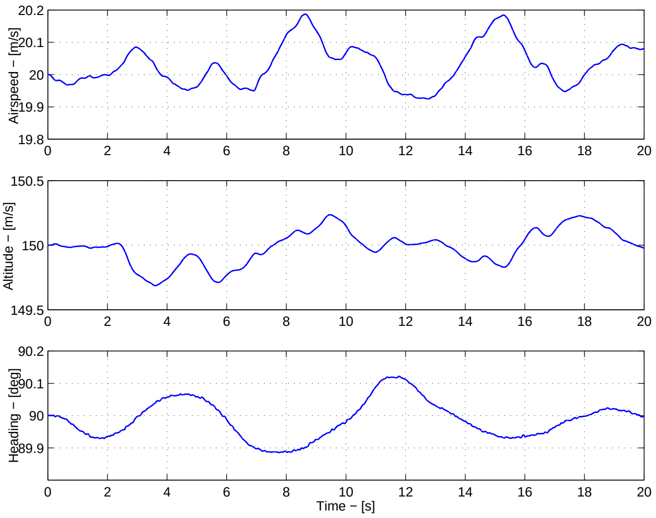


Figure 6.14: *Airspeed, Altitude and Heading Responses with Sensor Noise*

6.2.2 Graphical Simulator

The graphical simulator is written in C-code and uses the OpenGL graphics library to display the aircraft's motion though a virtual surrounding. The internal aircraft model is exactly the same as that used in the block diagram simulator. The major advantage of the graphical simulator over the block diagram simulator is its interface, which makes it possible to interact with the aircraft in real time and visually see the controllers at work. The virtual world makes it possible to fly the aircraft from takeoff to landing, making it a very useful tool for the future development of a takeoff and landing autopilot. Finally, the simulator also provides a good graphical presentation tool for demonstrating the controller at work. Some screenshots of the simulator are shown in Figures 6.15, 6.16

and 6.17. The simulator is now discussed further with reference to the specific problems it helped solve in preparing the controller for implementation.

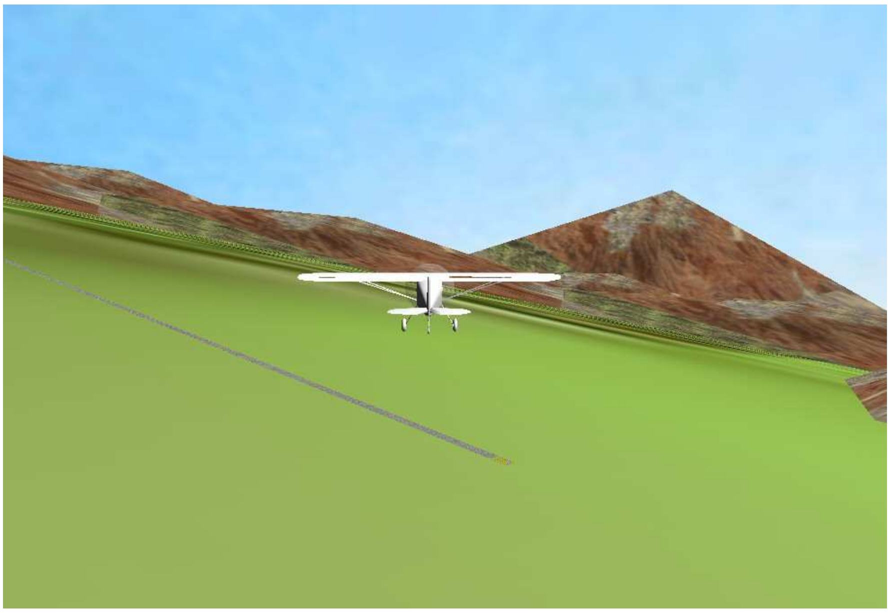


Figure 6.15: *Graphical Simulator Screenshot - Aircraft Following View*

Aircraft Model Verification

The graphical simulator makes it possible for a human pilot to verify the aircraft model. This can be done by manually flying the aircraft in the virtual world and seeing whether it responds similarly to a real model aircraft. Although it is not possible to quantify much with this type of testing, it still provides good assurance that the model developed does in fact produce responses similar to the actual aircraft's. The aircraft model was thoroughly tested in the simulator (one of the more enjoyable tasks!) and was found to produce the expected responses for a model aircraft of its size.

Code Conversion

One of the most important advantages of this simulator is that it is written in C-code - the same language that the controllers are implemented in on the control microprocessor. This made it possible to fully test and debug the C-code implementation of the controllers, before loading it onto the microprocessors. This technique minimised the amount of practical debugging work required and provides reassurance during flight tests, that the control code has at least been implemented correctly.

Figure 6.16: *Graphical Simulator Screenshot - Ground Observer View*

Smooth Transition

It is important to ensure the smooth transition of the actuators between autopilot and pilot control because these switching points can potentially produce large actuator steps which can damage the aircraft. To smooth the switching of an actuator from autopilot control back to pilot control, a first order exponential transition response from the autopilot's commanded position to the pilot's commanded position was implemented. The time constant of the response was chosen as 2s to provide a safe transition speed. When switching actuator control from safety pilot to autopilot a technique called integrator smooth transition was used. This technique involves setting the attitude controller's integrator states such that the current autopilot actuator commands coincide with the pilot actuator commands at the transition. The smooth transition algorithms are attained by re-arranging equation (5.12) to make the integral states the subject of the formula, and by inspection of the block diagram of Figure 5.14. The algorithms are summarised below with the airspeed, climb rate and yaw rate error integrals states as the subjects of the formulas.

$$\begin{bmatrix} \int e_u \\ \int e_{\dot{h}} \end{bmatrix} = -K_I^{-1}\left(K_P \begin{bmatrix} u - r_u \\ \dot{h} - r_{\dot{h}} \end{bmatrix} - \begin{bmatrix} \delta_e \\ \delta_{t_c} \end{bmatrix}_{SP}\right) \tag{6.1}$$

$$\int e_r = \frac{[\delta_a]_{SP}}{K_r} \tag{6.2}$$

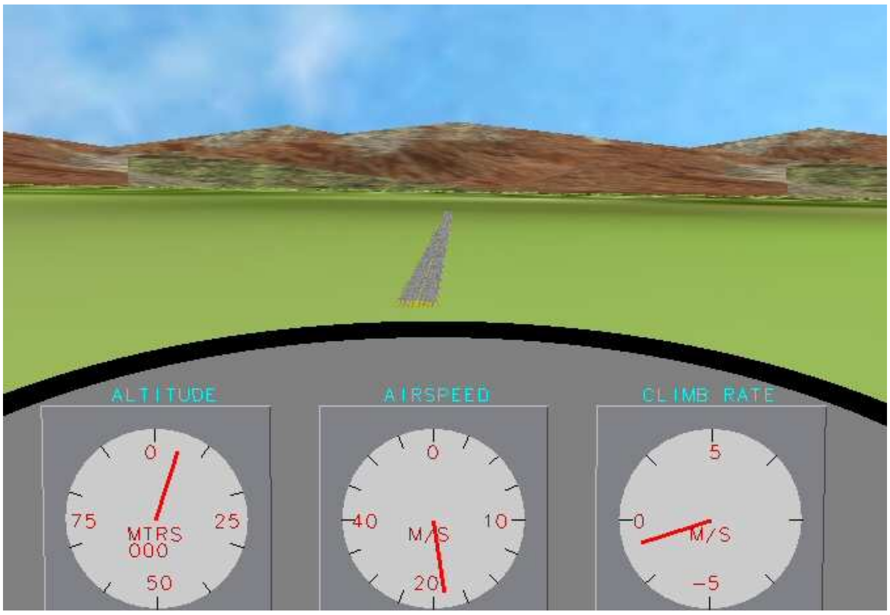


Figure 6.17: *Graphical Simulator Screenshot - Inside Aircraft View*

where, r in equation (6.1) indicates the reference command for the subscripted variable and the subscript SP indicates the safety pilot's actuator command. These smooth transition algorithms were tested in the graphical simulator and practically implemented with great success.

Autopilot Overview

Finally, a good overview of the autopilot performance can be formed with the graphical simulator. The complete system including the effects of all of the topics discussed in Section 6.2.1 as well as the topics discussed in this section, can be tested simultaneously and visually scrutinised. This makes it possible to see the autopilot at work and allows the switching between various controllers to be fully tested. This final stage of testing provides good confidence in the complete autopilot package.

6.3 Summary

In this chapter, a navigation algorithm capable of generating the cross track error signal required by the guidance controller of Chapter 5 was developed. The two non-linear simulators were then addressed with specific attention paid to the problems they helped solve. The chapter showed the importance of the two simulators in fully preparing the control algorithms for practical implementation. These simulators are essential tools for achieving the goal of successful autonomous flight in a minimum number of test flights.

This chapter marks the end of the theoretical development of the autopilot, as it is now ready for implementation. The avionics design and ground station software are the last topics that need to be considered before a practical implementation is possible. These topics are the subject of the next chapter.

Chapter 7

Avionics and Ground Station Design

This chapter discusses the hardware and software involved in the avionics and ground station of this project. A block diagram overview of the system is shown in Figure 7.1.

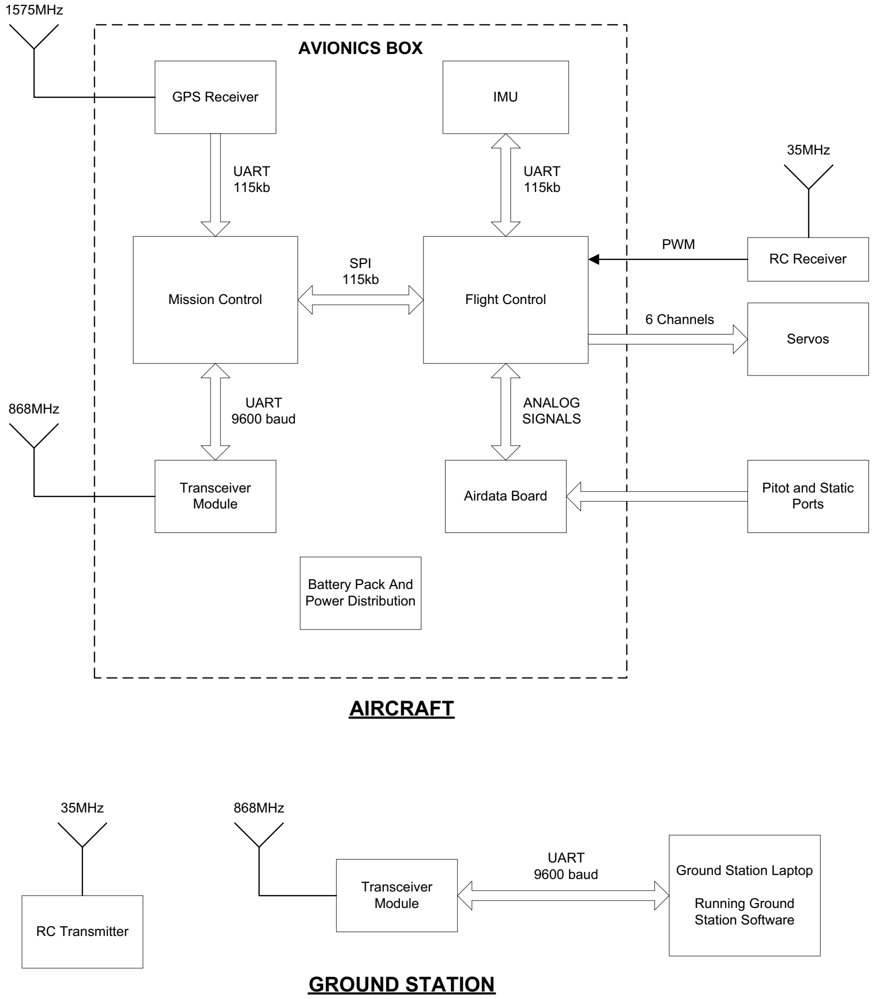


Figure 7.1: *Avionics and Ground Station Overview*

The block diagram shows that the system can be divided into two logical parts: the aircraft and the ground station. The aircraft consists of a RC (Radio Controlled) receiver, a set of servos as well as the avionics package developed. The ground station consists of a Personal Computer (PC) running the ground station software, a Radio Frequency (RF) module and a RC transmitter. The avionics package and ground station designs are now discussed in more detail and a summary of the system including a cost, power and weight analysis is presented in the final section.

7.1 Avionics

A picture of the avionics package is shown in Figure 7.2. All of the electronics are mounted in an aluminium box with the side walls hinged, which allowed for easy access to the components during the debugging phase. The entire aluminium box is mounted on rubber grommets and padded with soft foam inside the aircraft to provide vibration isolation. A picture of the avionics package mounted inside the aircraft is shown in Figure 7.3.

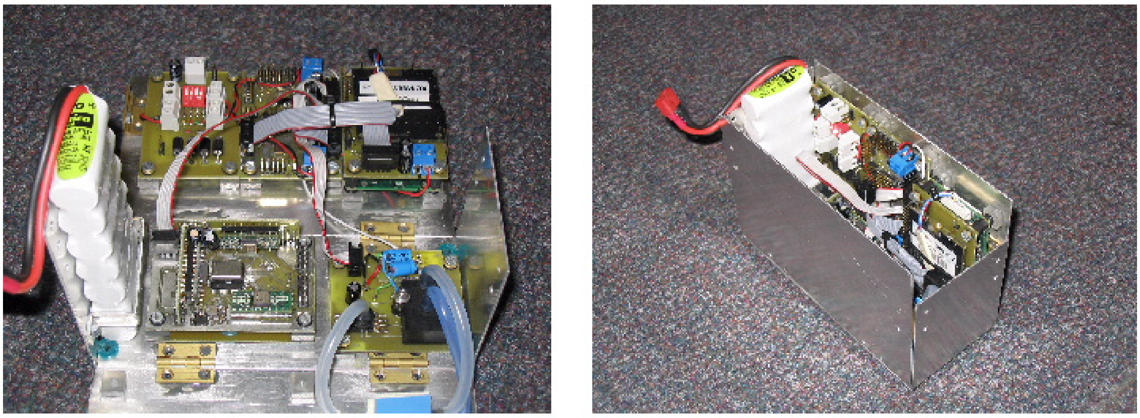


Figure 7.2: *Avionics Box Shown Open on the Left and Closed on the Right*

The avionics comprises of the following Printed Circuit Boards (PCB),

- Controller Board
- Inertial Measurement Unit (IMU)
- Airdata Board
- RF Module
- GPS Receiver

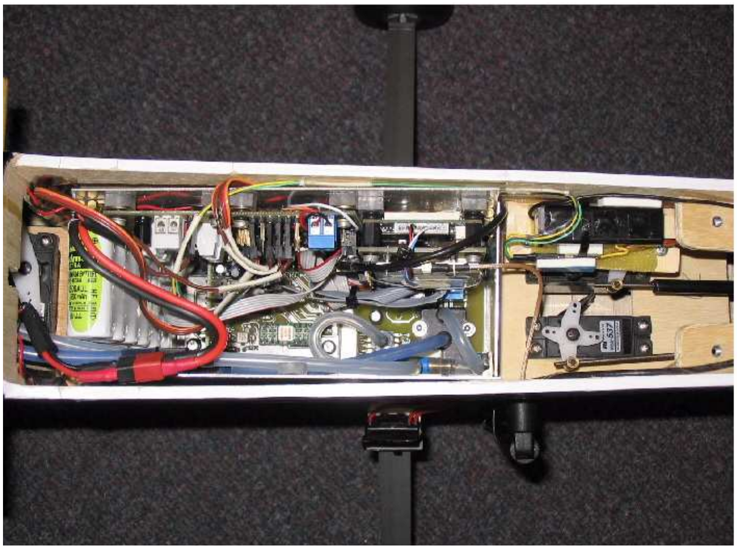


Figure 7.3: *Avionics Package Mounted Inside Aircraft*

- Battery and Power Distribution Board

Each of these boards is now discussed in more detail.

7.1.1 Controller Board

A block diagram overview of the Controller Board is shown in Figure 7.4.

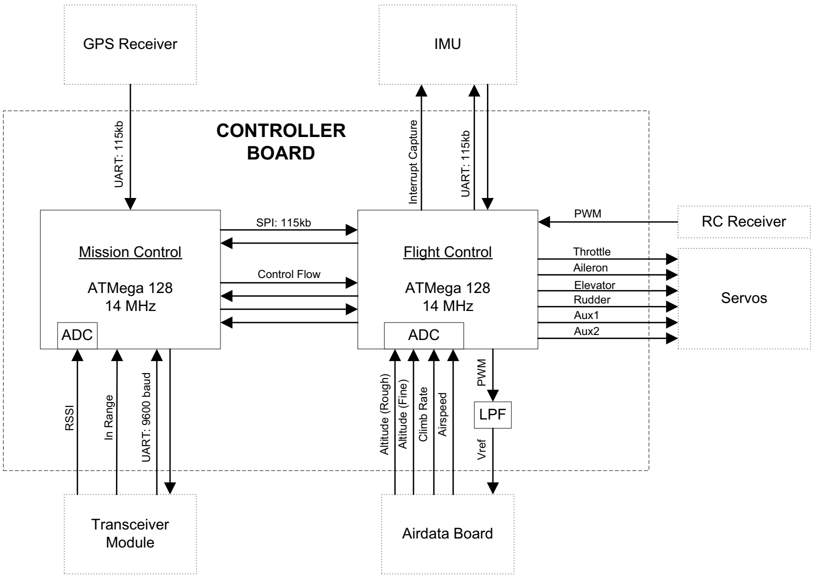


Figure 7.4: *Controller Board Block Diagram*

At the heart of this board are,

- 2 × ATMega 128 Microprocessors

These microprocessors were chosen because they met all of the communications, sampling and processing needs of the project at a very low cost.

As indicated in Figure 7.4, the two microprocessors were given glamorous names: *Flight Control* and *Mission Control*, and are referred to as such for convenience throughout the text. Both microprocessors run at 14MHz and communicate serially, via the SPI protocol, at 115kB, as well as via four direct connection lines used for data flow control. Each microprocessor and its duties in the avionics package are now discussed in more detail.

Flight Control

Flight Control is responsible for executing all of the control code, interfacing with the high bandwidth sensors such as those on the IMU and Airdata Board and driving the servos. The IMU is interfaced serially, via the UART protocol, at 115kB as well as via a single 'interrupt on input capture' line to allow Flight Control to determine when the IMU sensors are sampled. The signals from the airdata board are sampled directly by Flight Control using its on-board 8 channel, 10bit Analog to Digital Converter (ADC). The airdata board also requires a digitally variable voltage reference signal to bias one of its amplifiers. The voltage reference signal is attained by low pass filtering a 57.6kHz Pulse Width Modulation (PWM) signal generated by Flight Control, to extract only the DC component with acceptable ripple.

To allow Flight Control to switch between safety pilot and autopilot, the servos are driven directly from the microprocessor. This is done by re-routing the RC receiver's PWM signal, containing all of the servos' position information and capturing it using an 'interrupt on input capture' pin. Manually driving the servos also allows single control loops to be turned on at a time, while the safety pilot controls the actuators not in use by the autopilot.

A flow chart of the code running on Flight Control is presented in Figure 7.5. The timing dependent processor tasks are shown on the left leg of the flow chart. Once Flight Control has completed these tasks, it continues to process the incoming and outgoing communication buffers, one byte at a time, as necessary.

Following the left leg of the flow chart, it is clear that when the autopilot is armed, the

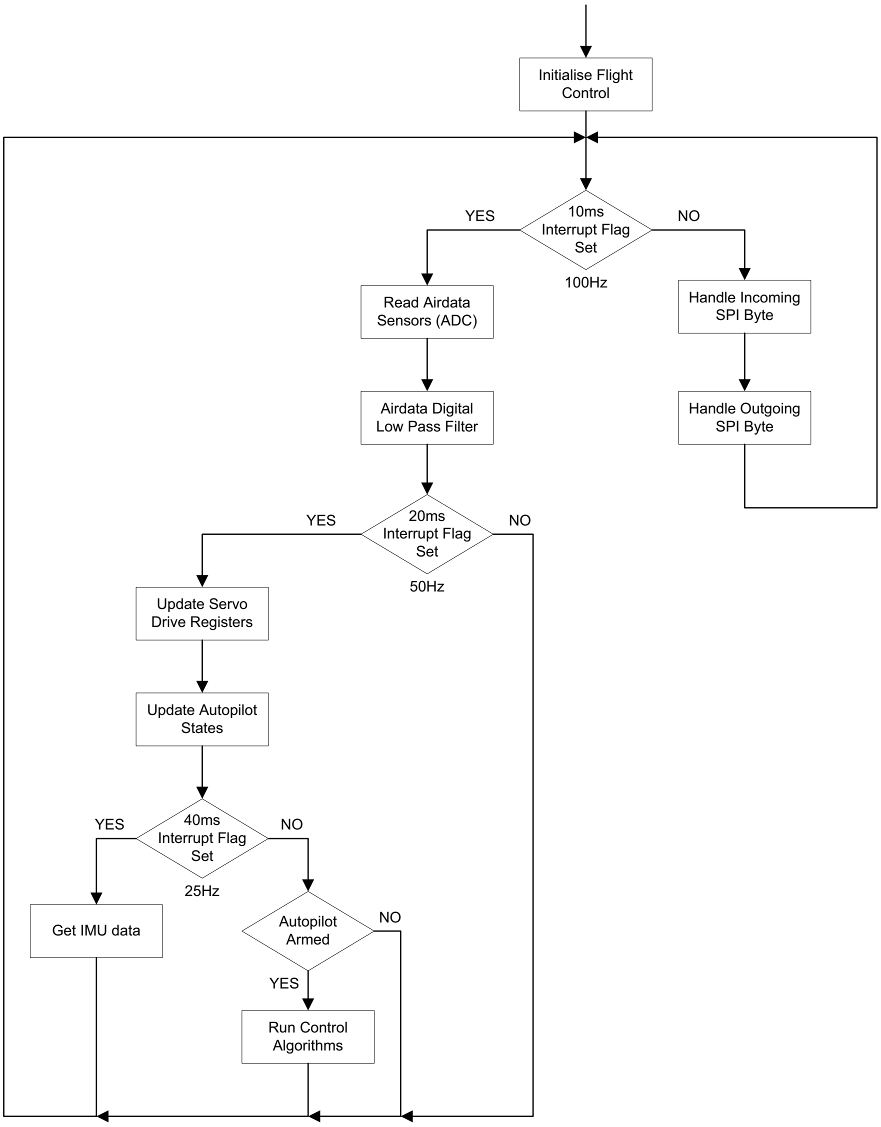


Figure 7.5: *Flight Control Flow Chart*

controller is executed at every odd 20ms interval (25Hz). The IMU data is prepared every even 20ms interval, to ease the processing load when the controller is executed. This setup introduces an unmodeled 20ms delay into the control system. However, its effects on the control system performance are negligible considering that the low level controllers only have a bandwidth on the order of 0.25Hz.

The servos are driven at 50Hz using an interrupt technique. A timer is used to measure out a servo pulse length by preloading it appropriately and waiting for the overflow interrupt. All autopilot variables are also polled for any ground station updates at 50Hz. The airdata is oversampled at 100Hz and then digitally low-pass filtered down to a bandwidth of 2Hz for use in the control algorithms at 25Hz. The lag effect introduced by the filter is also negligible. The oversampling helps to average out any inherent noise in the ADC process.

Flight Control's interrupts and their execution delays are summarised below.

- A timer interrupt to control code execution
  Delay ≈ 2.4μs every 5ms ⇒ 0.05% of 40ms period
- An interrupt on input capture used for the RC receiver PWM signal
  Delay ≈ 9×5.7μs every 22ms ⇒ 0.23% of 40ms period
- An interrupt on timer overflow used in driving the servos
  Delay: ≈ 6×6.0μs every 20ms ⇒ 0.18% of 40ms period
- General communication interrupts for the serial interfaces
  Delay ≈ 100×2.0μs every 40ms ⇒ 0.5% of 40ms period

It is clear that all of the interrupts present negligible delays compared to the controller execution period.

Mission Control

Mission control is responsible for handling the navigation algorithms, interfacing with the GPS, handling the communication link with the ground station and sampling the battery voltage and Received Signal Strength Indicator (RSSI) of the RF module. Provision for an ultrasonic range finder and a magnetometer was also made but is not discussed further because the sensors were not used in the project. The GPS and RF modules interface serially with Mission Control, via the UART protocol, at 115kB and 9600 baud respectively.

A flow chart of the code running on Mission Control is presented in Figure 7.6. It is clear that the fastest timing loop is an order of magnitude slower than that on Flight Control and consequently, an interrupt analysis is not presented. The flow chart indicates that every 200ms (5Hz), a primary telemetry packet is prepared for sending and that every 1s (1Hz), a secondary telemetry packet is prepared. The battery voltage and the RF module's RSSI are monitored every 100ms (10Hz). Although not implemented in this project, the RSSI can be used to actively set the RF module's transmit power to maintain a reliable wireless connection.

When Mission Control is not performing its time dependent tasks, it continues to process the incoming and outgoing communication buffers, one byte at a time, as necessary. The incoming GPS buffer is also handled during this time and when the GPS information is ready, the Navigation algorithm summarised in Figure 6.2 is executed (if the Navigator is active) and the GPS and Navigator information packets are prepared. Each time the path planner algorithm of Figure 6.2 is called, there is a 22ms execution time delay. However, this delay is acceptable, because none of Mission Control's tasks are strictly time dependant and all of the incoming and outgoing communications are buffered to handle the delay.

The primary, secondary, GPS and Navigator telemetry packets as well as the communication header, trailer and checksum protocol used throughout the system, are shown in Appendix E. Of importance, is the worst case scenario when all of the telemetry data is sent in the same transmit cycle. This is shown to take 62% of a 5 Hz transmit cycle at 9600 baud.

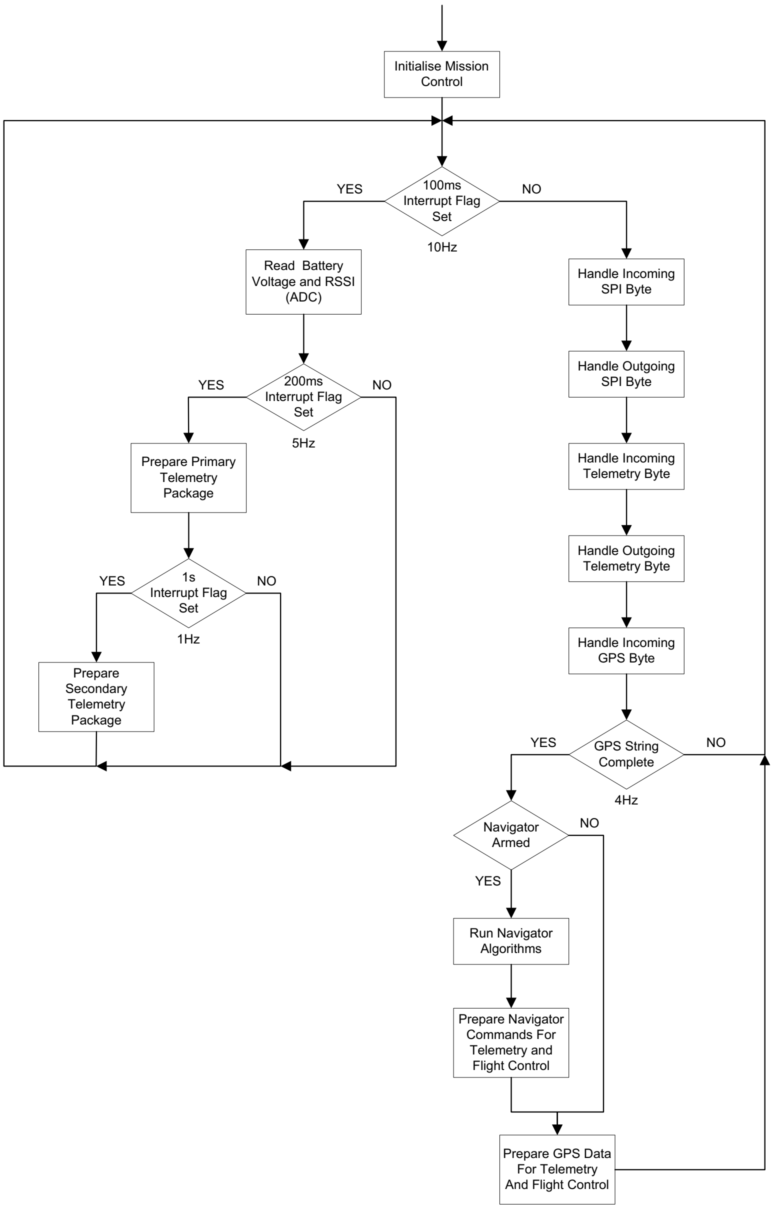

Initialise Mission Control
100ms Interrupt Flag Set
YES
NO
10Hz
Read Battery Voltage and RSSI (ADC)
Handle Incoming SPI Byte
Handle Outgoing SPI Byte
200ms Interrupt Flag Set
YES
NO
5Hz
Prepare Primary Telemetry Package
Handle Incoming Telemetry Byte
Handle Outgoing Telemetry Byte
1s Interrupt Flag Set
YES
NO
1Hz
Prepare Secondary Telemetry Package
Handle Incoming GPS Byte
GPS String Complete
YES
NO
4Hz
Navigator Armed
NO
YES
Run Navigator Algorithms
Prepare Navigator Commands For Telemetry and Flight Control
Prepare GPS Data For Telemetry And Flight Control

Figure 7.6: *Mission Control Flow Chart*

7.1.2 Inertial Measurement Unit

The IMU board houses the following inertial sensors and microprocessor,

- 3 × ADXRS150EB Rate Gyroscopes
- 2 × ADXL202AE Dual-Axis Accelerometers
- 1 × ATMega 128 Microprocessor

The sensors were selected because of their combination of low cost and good performance and the microprocessor was chosen to remain consistent with the Controller Board. Although the accelerometers are not required for feedback purposes in the control system designed, they were included in the design of the IMU to extend its applicability to future UAV projects.

A picture and a block diagram overview of the IMU board are shown in Figures 7.7 and 7.8 respectively. The sensors are mounted orthogonally so that angular rate and specific force in each aircraft axis can be sensed. The entire IMU is also mounted on soft rubber grommets to provide vibration isolation. The block diagram is now discussed in more detail.

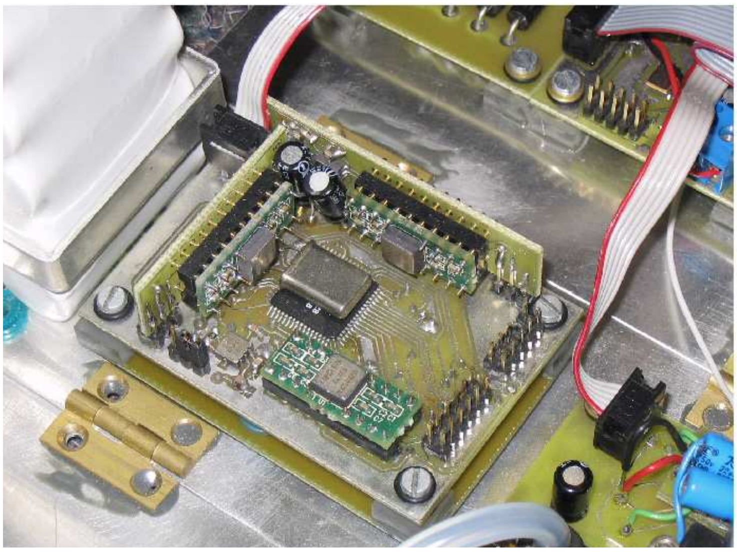


Figure 7.7: *Picture of IMU*

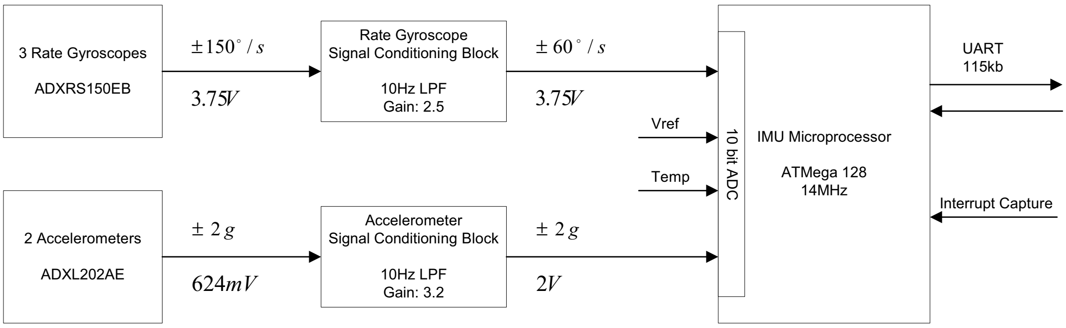


Figure 7.8: *Block Diagram of IMU*

Rate Gyroscopes

With no signal conditioning, the rate gyroscopes are capable of sensing angular rates of up to ±150 deg/s at a bandwidth of 40Hz, mapped over a 2.5V centred voltage span of 3.75V. The bandwidth is unnecessarily high and would have introduced unwanted noise into the system if left unaltered. Also, given the output voltage span, the microprocessor's 10bit ADC used to sample the signal would have only been able to provide a resolution of around 0.4deg/s. During conventional flight, high angular rates are not of interest and resolution of the signal is far more important.

For the above reasons, the *Rate Gyroscope Signal Conditioning Block* of Figure 7.8 was designed. It is responsible for amplifying the rate gyroscope's signal by 2.5, for ensuring that its zero rate output is close to 2.5V and for reducing the signal's bandwidth to 10Hz. The reduction in bandwidth removes a lot of the unwanted signal noise but still allows flexibility for further digital filtering in the microprocessor at a later stage. The gain increases the digital resolution of the signal to around 0.1deg/s and reduces the measurable range to ±60deg/s. Although the rate signal may saturate during sudden movements of the aircraft, the control system is insensitive to rate signal saturation, and thus no problems are expected. A circuit diagram of the block is shown in Appendix E.

Accelerometers

The raw output of the accelerometers is a $\pm 2g$ signal at a bandwidth of 6kHz, mapped over a voltage range of 0.624V centred about 2.5V. The bandwidth is again far to high and would have introduced large amounts of unwanted noise if left unaltered. The digital signal resolution using a 10bit ADC would have been $0.03g$, which is acceptable but was improved to make better use of the ADC.

The *Accelerometer Signal Conditioning Block* was designed similarly to the rate gyroscope signal conditioning block. Its purpose is to reduce the bandwidth of the signal to 10Hz as well as to provide a gain of 3.2 that increases the digital resolution of the signal to around 0.01g. A circuit diagram of the block is shown in Appendix E.

IMU Microprocessor

The conditioned rate gyroscope and accelerometer signals are sampled together with a temperature and voltage reference signal (from one of the rate gyroscopes) by the IMU microprocessor's 8-channel 10bit ADC. The microprocessor runs at 14MHz and is responsible for providing further filtering of the signals, removing biases and interfacing with the flight control microprocessor. A flow chart of the code running on the IMU microprocessor is shown in Figure 7.9. The 200Hz sample rate is again to average out any inherent noise in the ADC process.

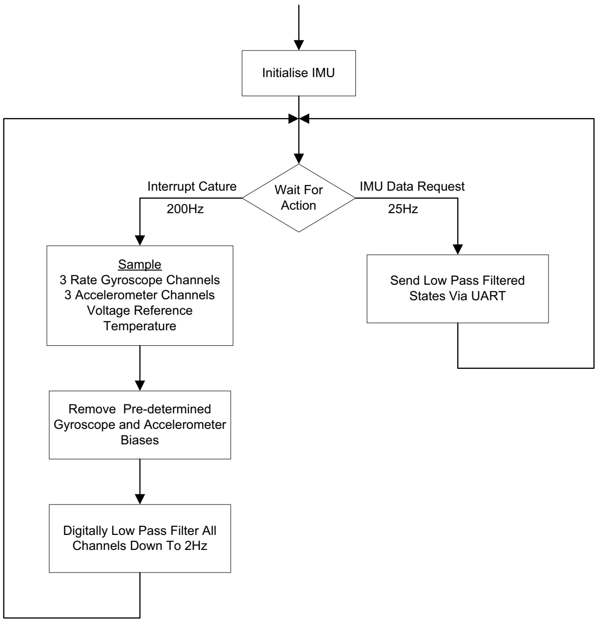

Initialise IMU
Interrupt Cature
200Hz
Wait For Action
IMU Data Request
25Hz
Sample
3 Rate Gyroscope Channels
3 Accelerometer Channels
Voltage Reference
Temperature
Send Low Pass Filtered States Via UART
Remove Pre-determined Gyroscope and Accelerometer Biases
Digitally Low Pass Filter All Channels Down To 2Hz

Figure 7.9: *Flow Chart of the IMU Microprocessor Code*

7.1.3 Airdata Board

A picture and a block diagram overview of the Airdata Board are shown in Figures 7.10 and 7.11 respectively.

The Airdata board houses the following sensors,

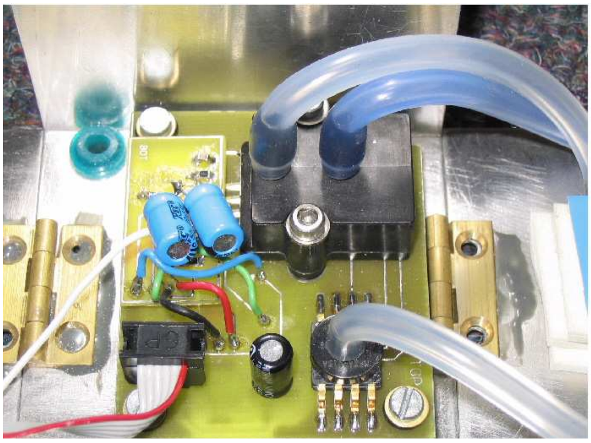


Figure 7.10: *Picture of Airdata Board*

- 1 × DC005NGC4 Differential Pressure Sensor
- 1 × MPXA4115A Absolute Pressure Sensor

The differential pressure sensor was chosen because of its high sensitivity which meant that very little signal conditioning was required. The absolute pressure sensor was chosen because of it low cost and good performance.

The Airdata Board is responsible for generating airspeed, climb rate and altitude signals. How these signals are generated from a differential and absolute pressure sensor is the subject of the next two sections.

Differential Pressure Sensor

Bernoulli's equation (2.36) shows that airspeed is related to dynamic pressure ($\frac{1}{2}\rho\bar{V}^2$) and that dynamic pressure is the difference between total (p_T) and static (p) pressure. A pitot-static tube system, connected to the differential pressure sensor, was used to measure this pressure difference. A schematic diagram and picture of the system used in this project is shown in Figure 7.12.

The pitot tube is essentially an open ended pipe that faces into the oncoming airflow with the other end terminated into the differential pressure sensor's high pressure port. The pressure in this tube is equal to the total pressure of the airflow. The static port was made by closing up and streamlining the front end of a tube and then drilling holes further back along it, perpendicular to the airflow direction, where the static pressure well approximates the static pressure of the freestream flow (see Figure 7.12). The other

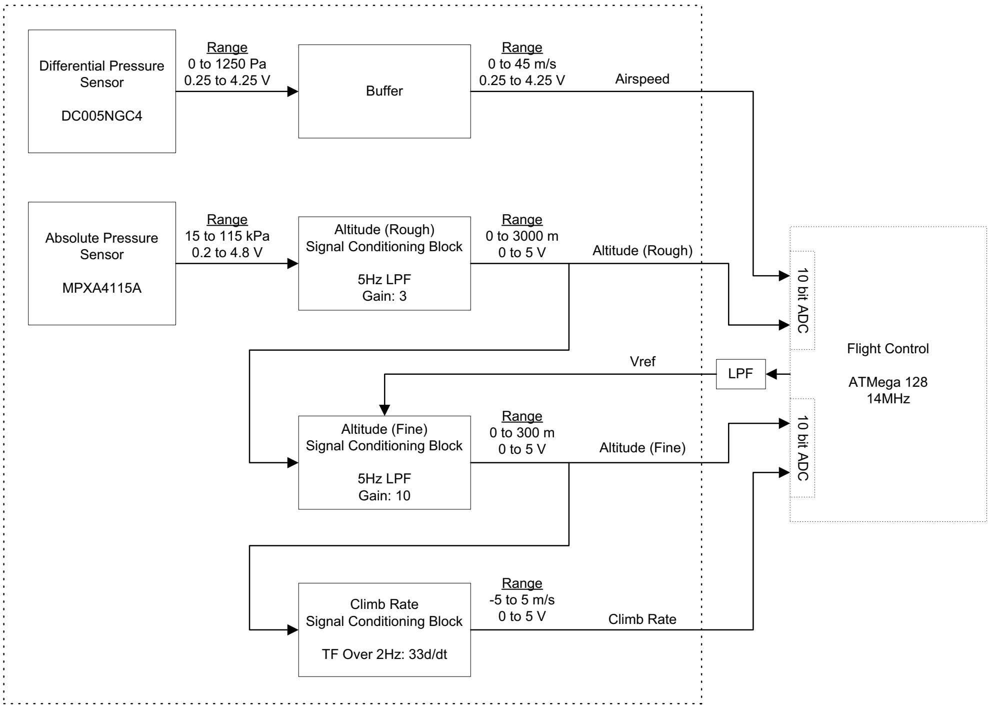


Figure 7.11: *Block Diagram of Airdata Board*

end of this pipe was then attached to the differential pressure sensor's low pressure port to form a device capable of measuring dynamic pressure.

The raw output of the differential pressure sensor is a 0 to 1.25kPa signal mapped over a voltage span of 4V. This means that the sensor is capable of measuring airspeeds of up to 45m/s at sea level which provides adequate leeway given the aircraft trim velocity of 20m/s. Of more interest however, is the airspeed digital resolution at the trim velocity. Differentiating the equation for dynamic pressure and substituting the trim airspeed and sea level density reveals,

$$\left.\frac{\partial q}{\partial \bar{V}}\right]_{\bar{V}=U_0} = \rho U_0 = 24.5 \text{ Pa/ms}^{-1} \tag{7.1}$$

Converting this to ADC values,

$$\text{Digital Resolution} = \left(\frac{24.5 \cdot 4 \cdot 2^{10}}{1.25 \times 10^3 \cdot 5}\right)^{-1} \approx 0.07 \text{ ms}^{-1} \tag{7.2}$$

This means that on a 10bit ADC, the airspeed can be resolved to within 0.07m/s at 20m/s at sea level. This resolution is satisfactory and means that no further signal processing is required. As shown in Figure 7.11 the differential pressure sensor signal is merely buffered and then sent directly to Flight Control where it is converted from dynamic pressure to airspeed.

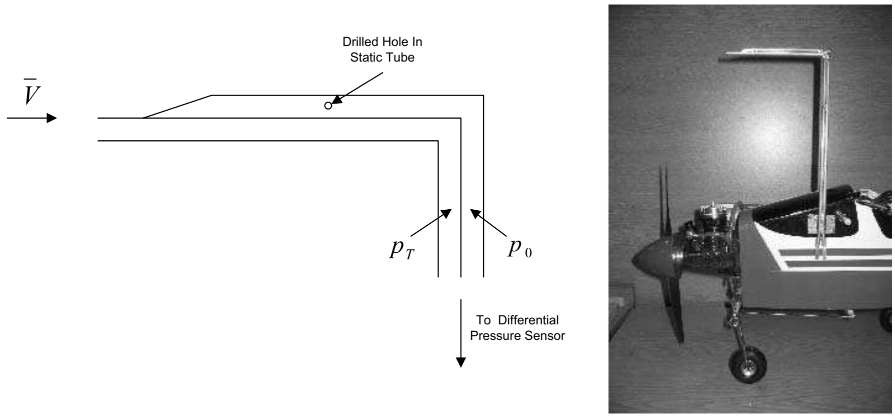


Figure 7.12: *Pitot-Static Tube System Schematic and Picture*

Absolute Pressure Sensor

In a standard atmosphere, static pressure is related to altitude (h) by the following equation [39],

$$p(h) = p_0 e^{-\frac{Mgh}{RT_0}} \tag{7.3}$$

where, p_0 is the average static pressure at sea level (101.325 kPa), M is the average molecular mass of the atmosphere (0.02897 kg/mole), T is the average temperature at sea level (288 K) and R is the gas law constant (8.314510 J/K/mole). Thus, the measurement of static pressure and its time derivative provide a means of determining the aircraft's altitude and climb rate respectively.

The absolute pressure sensor provides a raw static pressure signal of 15 to 115kPa mapped over a voltage range of 0.2 to 4.8V. For portability reasons, it was desired to measure altitude between 0 and 3000m above sea level which, on substitution into equation (7.3) and accounting for high pressure days at sea level, gives the pressure range of interest as: 70 to 105kPa. To make full use of the 10bit ADC, the *Rough Altitude Signal Conditioning Block* of Figure 7.11 was designed. This block is responsible for limiting the sensor signal bandwidth to 5Hz and amplifying it such that the 70 to 105kPa pressure range extends over the entire ADC input. The amplification provides a digital resolution of 2.8m at sea level. A circuit diagram of the block is shown in Appendix E.

The altitude resolution attainable from the signal of the *Rough Altitude Signal Conditioning Block* is still unacceptable and led to the design of the *Fine Altitude Block*. This block is responsible for amplifying the rough altitude signal by a factor of 10 such that the sea level resolution is increased to 0.28m. The amplifier offset voltage is obtained from the low pass filtered PWM signal from Flight Control. The PWM signal is set at the beginning of a flight as a function of the ground level rough altitude signal. The *Fine Altitude Signal Conditioning Block* thus provides a high resolution altitude measurement over a local 300m range for any altitude between 0 and 3000m. A circuit diagram of the block along with the equation for calculating the reference voltage is provided in Appendix E.

The final task was to develop circuitry to extract a climb rate signal. This was done by taking the fine altitude signal and sending it through the *Climb Rate Signal Conditioning Block* of Figure 7.11. This block acts as a differentiator for signals below 2 Hz and then 'rolls off' into a low pass filter so as not to amplify high frequency noise. The block has a gain of 33 which results in a ±5m/s climb rate swing at a resolution of 0.01 m/s on a 10bit ADC. The circuit diagram is shown in Appendix E.

7.1.4 RF Module

The RF module and antenna used are listed below,

- Aerocomm AC4486-500M-3 OEM Transceiver
- Active dipole antenna with 5dB gain

A picture of the RF module board is shown in Figure 7.13. The module was chosen because of its high power (Effective Isotropic Radiated Power (EIRP) = 500mW), ease of implementation and relatively low cost.

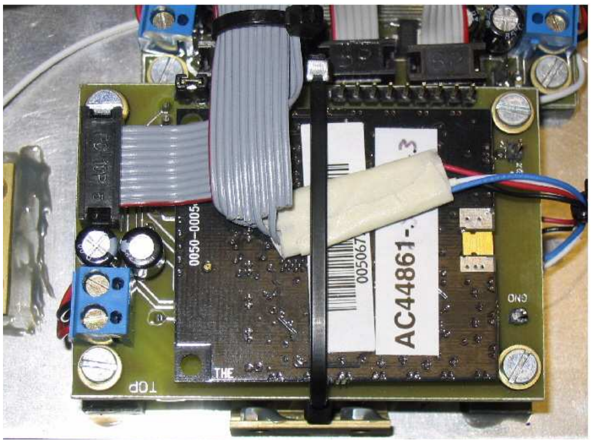


Figure 7.13: *Picture of the RF Module*

The module was set up in the following manner.

- 9600 baud serial interface
- Full duplex mode
- Acknowledge packets with 3 times retry
- Addressed packets
- Transmit power of 250mW

The module works excellently for telemetry and communication purposes and its long range capabilities make it well suited for autonomous navigation, where the aircraft has the potential to fly long distances from the ground station. A picture of the RF module's antenna mounting is shown in Figure 7.15.

7.1.5 GPS Receiver

The GPS receiver and antenna used are listed below,

- u-Blox RCB-LJ
- Active antenna with 25dB gain

A picture of the GPS receiver board is shown in Figure 7.14. The receiver was chosen based on the results of extensive testing in [23], which show that the receiver provides an excellent combination of low cost and measurement accuracy.

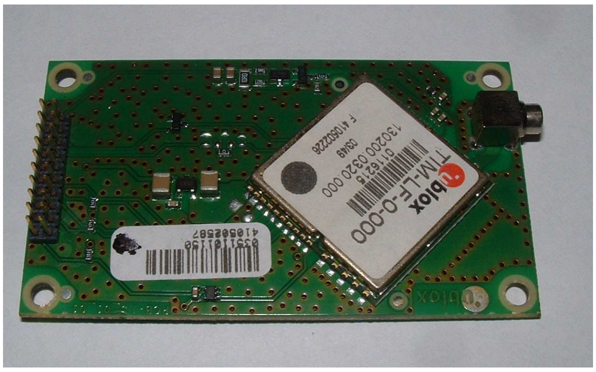


Figure 7.14: *Picture of the GPS Receiver*

The receiver was set up to deliver information on the aircraft's longitude, latitude, altitude, ground speed and heading as well as information on the quality of the satellite fix.

The information is sent over a serial link, via the UART protocol, at 115kB to Mission Control, where it is used in the control and navigation algorithms.

A picture of the GPS receiver's antenna mounting is shown in Figure 7.15.

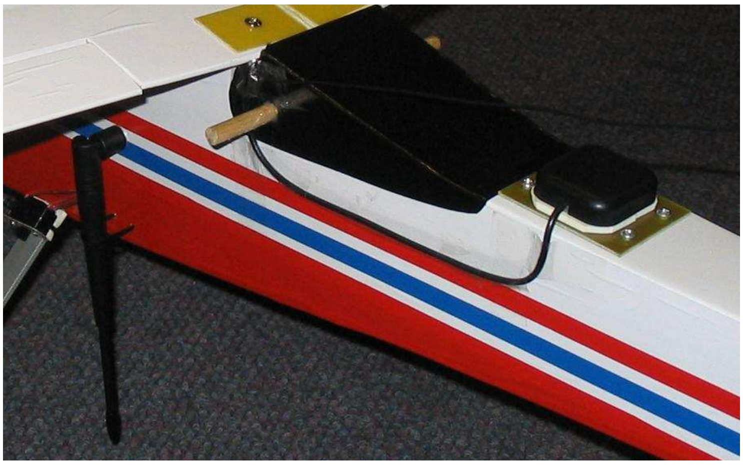


Figure 7.15: *Picture of the RF Module Antenna (on the side of the aircraft) and the GPS Receiver Antenna (on the tail of the aircraft)*

7.1.6 Power Distribution

The avionics power source is a,

- 8.4V, 1500mAh NiCd Battery Pack

The battery is connected directly to the Power Distribution Board where it is routed as shown in the block diagram of Figure 7.16.

Only the servos and RC receiver are powered by an unregulated voltage supply. This ensures that the servos always have access to a high current source that may be required in the event of a stuck servo. If a standard regulator were used during this event, it would either shut down or fail, rendering the aircraft uncontrollable.

The flow chart also displays the average power dissipation in each block and reveals that the total avionics power consumption is approximately 5W. This means that the 1500mAh battery pack is able to power the system for about $2\frac{1}{2}$ hours.

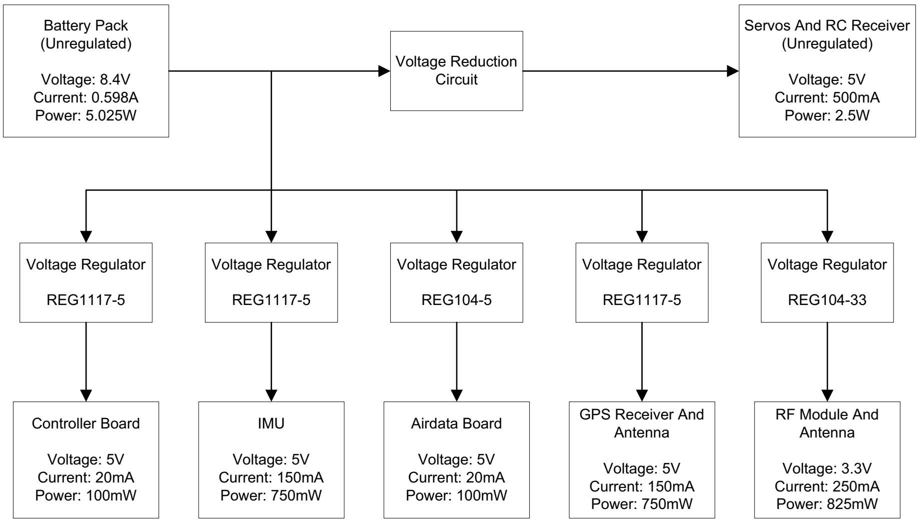


Figure 7.16: *Block Diagram of Avionics Power Distribution*

7.2 Ground Station

Figure 7.1 shows that the ground station is made up of a PC running the ground station software, a RF communications module and a RC Transmitter. The RF module provides a wireless link to the aircraft used for telemetry and communications purposes. The ground station PC and the RF module communicate serially, via the UART protocol, at 9600 baud. The ground station RF module is set up in exactly the same way as the avionics RF module is. The only modification to the RC transmitter is that one of its channels is used, in conjunction with a switch in the ground station software, to turn the autopilot on and off. This allows the safety pilot to regain full control over the aircraft at any time.

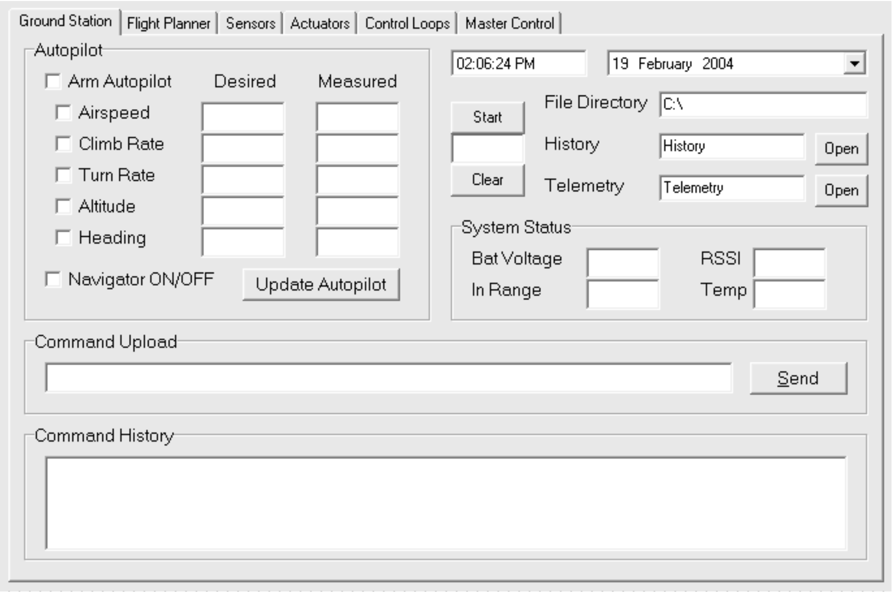
The rest of this section is devoted to describing the ground station software.

7.2.1 Ground Station Software

The ground station software is a Graphical User Interface (GUI) to the aircraft's on-board avionics package, that allows the autopilot variables to be updated without the need to reprogram the control microprocessors. Before a flight, the ground station and avionics package must both be reset, so that their initial states are synchronised. From then on, the ground station is able to upload new information which is acknowledged by the autopilot once received. If the acknowledge signal is not received by the ground station, the command can merely be sent again until the reception is confirmed.

The ground station software is best explained by graphically presenting all of the interface pages and briefly describing the functions that the various group boxes, buttons and text boxes perform.

The Ground Station Page



02:06:24 PM

19 February 2004

File Directory: C:\

History: History

Telemetry: Telemetry

Figure 7.17: *Ground Station Page*

Consider Figure 7.17. The *Autopilot* group box allows the various autopilot control loops to be armed and their reference commands to be set. The current measurement values are displayed next to the references for ease of comparison. The *Command History* list box displays a running history of all the ground station commands issued as well as any errors that occur. The *Command Upload* text box allows any predefined command to be uploaded to the autopilot. This page also allows telemetry and history logs to be created and terminated. It displays critical information such as the battery voltage, temperature and RF module signal strength. The time and date are displayed and a stop watch is provided to time flights.

The Flight Planner Page

Consider Figure 7.18. The flight planner page displays the data concerned with autonomous navigation. Among other things, it shows a list of the current navigation waypoints as well as indicating the waypoint that the aircraft is flying towards and what section of the path the aircraft is on. It allows waypoints to be added, edited and removed as well providing the ability to force the Navigator to jump to a certain waypoint in the

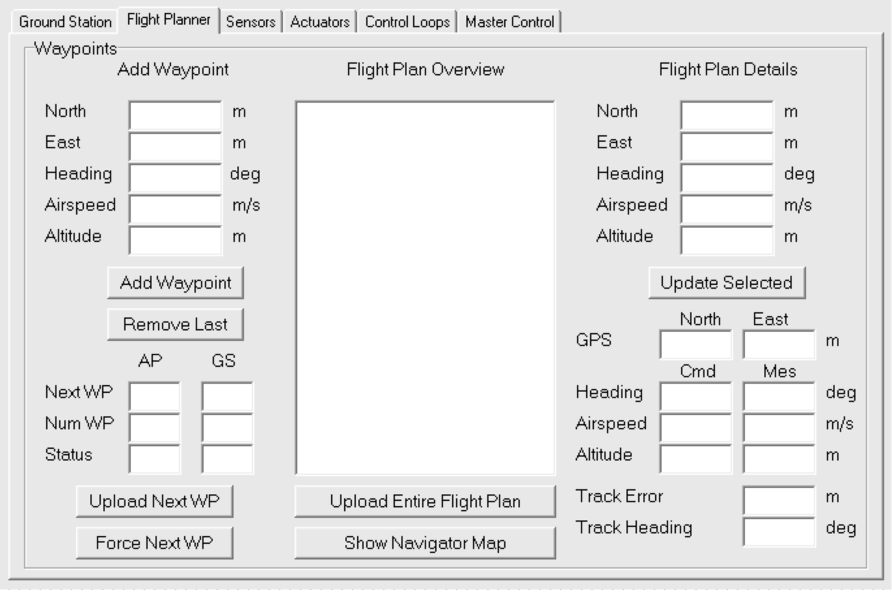


Figure 7.18: *Flight Planner Page*

list. There is also an option to display a 2D map of the aircraft, along with the waypoints and the path to be followed to the next waypoint.

The Sensors Page

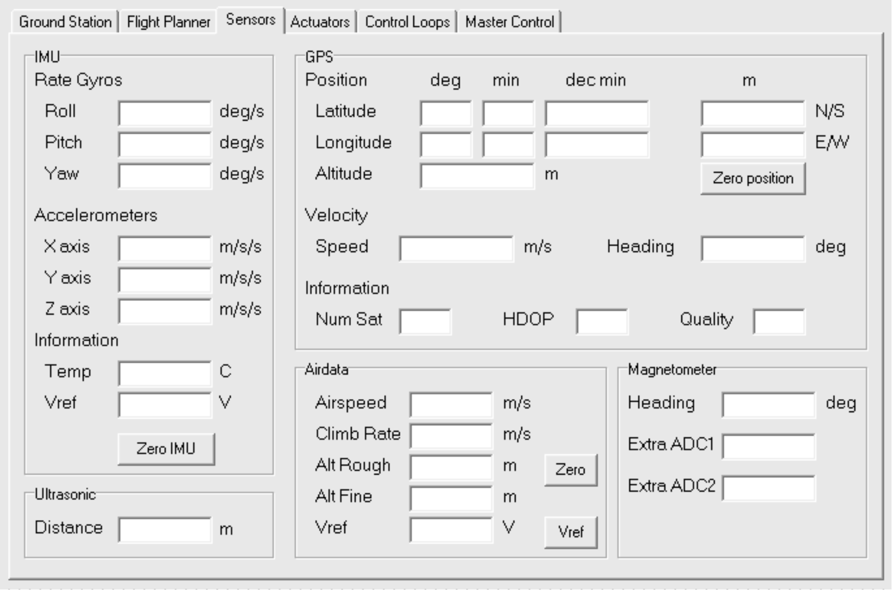


Figure 7.19: *Sensors Page*

Consider Figure 7.19. This page displays all of the avionics sensor data. Only the *IMU*, *Airdata* and *GPS* group boxes were used in this project. The page allows the voltage reference signal required by the *Fine Altitude Signal Conditioning Block* to be set up.

It also allows the rate gyroscope biases, pressure altitude offset and GPS location to be zeroed. The altitude zeroing is done on the runway just before the start of a flight to minimise the altitude drift.

The Actuators Page

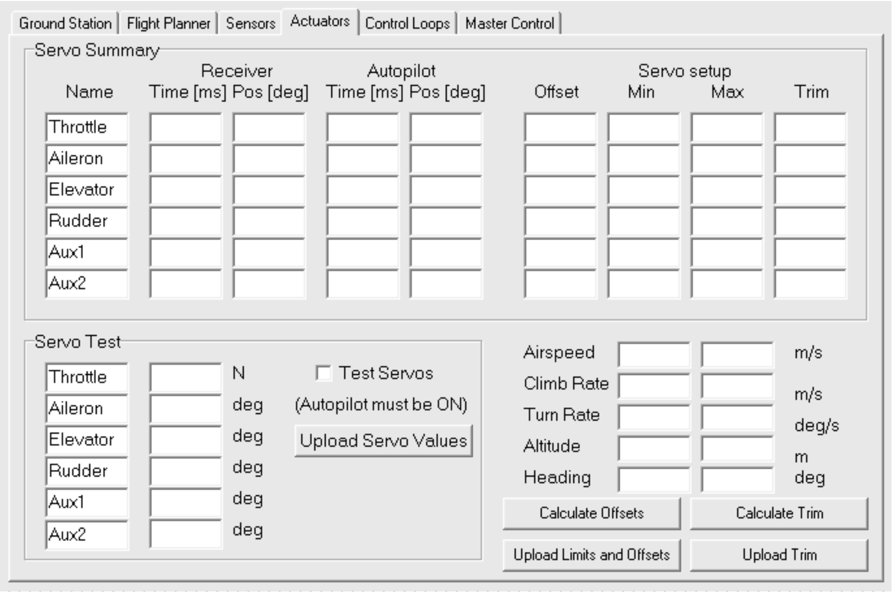


Figure 7.20: *Actuators Page*

Consider Figure 7.20. This page displays the safety pilot and autopilot servo positions as well as the servo limits, offsets and trim condition positions. The *Servo Test* group box provides a means for calibrating the servos by allowing a servo location to be directly commanded. This page also allows the ground station to synchronise with the RC transmitter limits and offsets as well as the safety pilot trim condition while in flight. These values can then be uploaded to the autopilot to set the controller bias points.

The Control Loops Page

Consider Figure 7.21. This page displays the longitudinal and lateral control loop gains and allows new gains to be entered and uploaded in flight. The gains are expressed as fractions as this is how they are implemented on Flight Control, given the lack of hardware floating point capability. The page also provides check boxes that make it possible to enable only selected parts of the control loops. This feature together with the in flight gain updates, makes the control loop page a good interface for fine tuning the controller during flight tests.

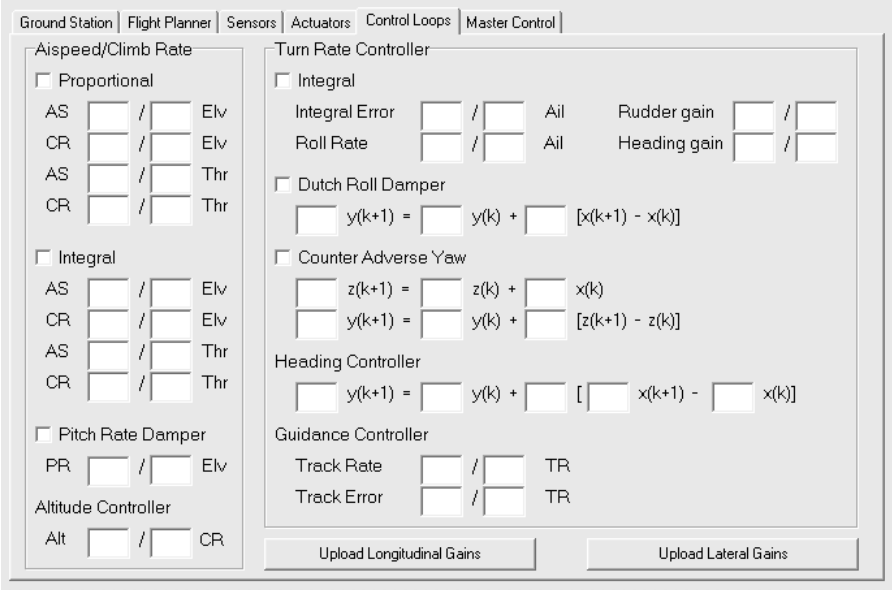


Figure 7.21: *Control Loops Page*

The Master Control Page

Finally, one last page was created for safety purposes (and for fun!). It involved using the mouse location on the page to control the elevator and aileron positions while setting the throttle and rudder to their trim positions. In the event of a RC link failure, this page can be used to bring the aircraft closer to the runway in the hope that the RC link is reacquired. Luckily, this page was never used!

7.3 Summary

This chapter presented an overview of the avionics and ground station developed for the practical implementation of the controller and navigation algorithms of Chapters 5 and 6. A cost, mass and power consumption summary of the avionics package is presented in Table 7.1.

Table 7.1 shows that the total cost of the avionics package is only R5000. The total extra weight that it adds to the aircraft is 845g. The added weight is acceptable because model aircraft typically have very low initial wing loadings and a high thrust to weight ratio. The avionics set increases the wing loading of the aircraft used in this project from 5.5 to 7.4kg/m^2 and decreases the thrust to weight ratio from approximately unity to 0.74. These values are perfectly acceptable for conventional flight. The power budget has already been discussed in Section 7.1.6 and it was shown that the aircraft is capable of flying under

Item	**Cost (R)**	**Mass (g)**	**Power (mW)**
Controller Board	350	25	100
IMU	1850	40	750
Airdata Board	1100	25	100
RF Module and Antenna	750	55	825
GPS Receiver and Antenna	550	90	750
Battery Pack	400	230	-
Pitot and Static Tubes	-	100	-
Avionics Box	-	280	-
Total	R 5000	845g	2525mW

Table 7.1: *Cost, Mass and Power Consumption Summary of Avionics Package*

autopilot control for approximately $2\frac{1}{2}$ hours on a single battery pack. Considering that a fuel tank only lasts on the order of 15mins, this is perfectly acceptable.

With the controller and navigation algorithms fully designed and the avionics and ground station hardware and software ready, the autopilot is complete and ready for flight tests. A picture of the complete system is shown in Figure 7.22.

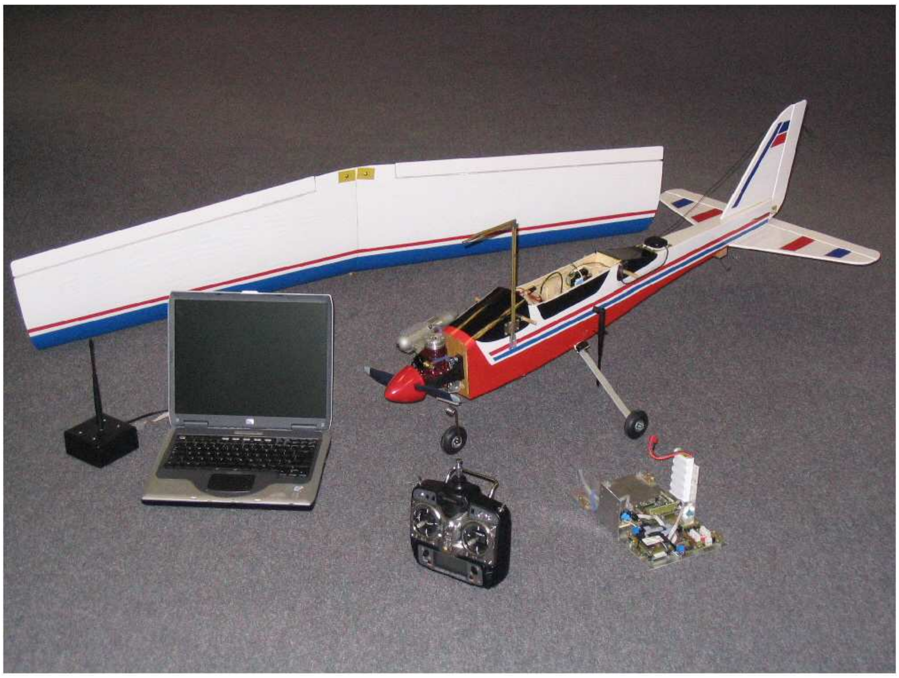


Figure 7.22: *Avionics, Ground Station and Aircraft*

Chapter 8

Flight Tests

Over a period of less than two weeks, three days of extremely successful flight tests were conducted. In this time, the longitudinal and lateral controllers as well as the navigation algorithm were shown to work successfully with minimal flight day complications. The lack of complications can largely be attributed to the vast amount of simulation that took place before the flight tests.

Section 8.1 begins by presenting an overview of what was achieved during the three flight test days. Then in Section 8.2 telemetry data describing the performance of the longitudinal and lateral controllers and the navigation algorithm is presented and analysed.

8.1 Flight Test Overview

A checklist was created to improve the chances of success for each flight test. The checklist was religiously consulted before each flight test day, to ensure that nothing was overlooked and that the aircraft was still fully intact. The checklist was also consulted just before individual flight tests to once again ensure the integrity of the aircraft.

All flight tests were held at the Helderberg Radio Flyers Club in Somerset West, with the Chief Flying Instructor as safety pilot.

8.1.1 Day 1 - Sunday the 15^{th} of August 2004

Figure 8.1: *Aircraft Pre-flight Checks*

Sunday the 15^{th} of August 2004 was the first time that the aircraft ever flew. Five flight tests were conducted on the day, the first three of which involved trimming the actuators for steady flight and allowing the safety pilot to familiarise himself with the aircraft. The last two flights involved the testing of the yaw rate controller. After a few minor complications, the yaw rate controller was found to work excellently, with the safety pilot controlling only the elevator and the throttle. The control appeared very smooth and as expected, no effect of the rate gyroscope biases was experienced.

8.1.2 Day 2 - Wednesday the 18^{th} of August 2004

Figure 8.2: *Aircraft Shortly After Takeoff*

The aim of this day was to test the airspeed and climb rate controller with the safety pilot controlling the ailerons and the rudder. Two flight tests were conducted and the airspeed and climb rate controller was found to work immediately. From the ground station, the control appeared extremely smooth and the aircraft pitch angle was well maintained by the control of the aircraft's airspeed and climb rate. The turn rate controller was then armed and the first fully autonomous flight was performed!

During the second flight, the altitude and heading controllers were armed and also worked immediately. These controllers were expected to work without much effort because of the layered controller architecture. The excitement of all of the controller success led to the second test flight continuing for over 15 minutes. The flight ended when the aircraft ran out of fuel and crash landed 100m short of the runway! The landing gear broke off when it dug into boggy ground caused by previous rain, but the rest of the fuselage remained fully intact.

8.1.3 Day 3 - Saturday the 28^{th} of August 2004

Figure 8.3: *Aircraft Landing*

With the aircraft landing gear re-attached, the purpose of this day was to test the Navigator. Five flight tests were held and after some minor complications, the Navigator was armed and found to work excellently. It successfully generated and followed the shortest path made up of an initial circle, a straight line and a final circle between two respective waypoints in the waypoint list. It was also possible to add, remove and edit waypoints and to force the aircraft to skip waypoints.

Many more months could have been absorbed in fine tuning and testing all of the details of the autopilot, but with the goal of autonomous navigation in a minimum number of test flights achieved and the impending arrival of the two year deadline (set as a design goal in Chapter 1), the flight tests were concluded.

8.2 Autopilot Results

In this section telemetry data representative of the performance of the autopilot is provided. It must be noted however, that the data presented is what the sensors were measuring during the flight and not what the actual state of the aircraft was. The sensor data provides a good indication of the controller performance but paints a noisier picture than the true response.

8.2.1 Longitudinal Controllers

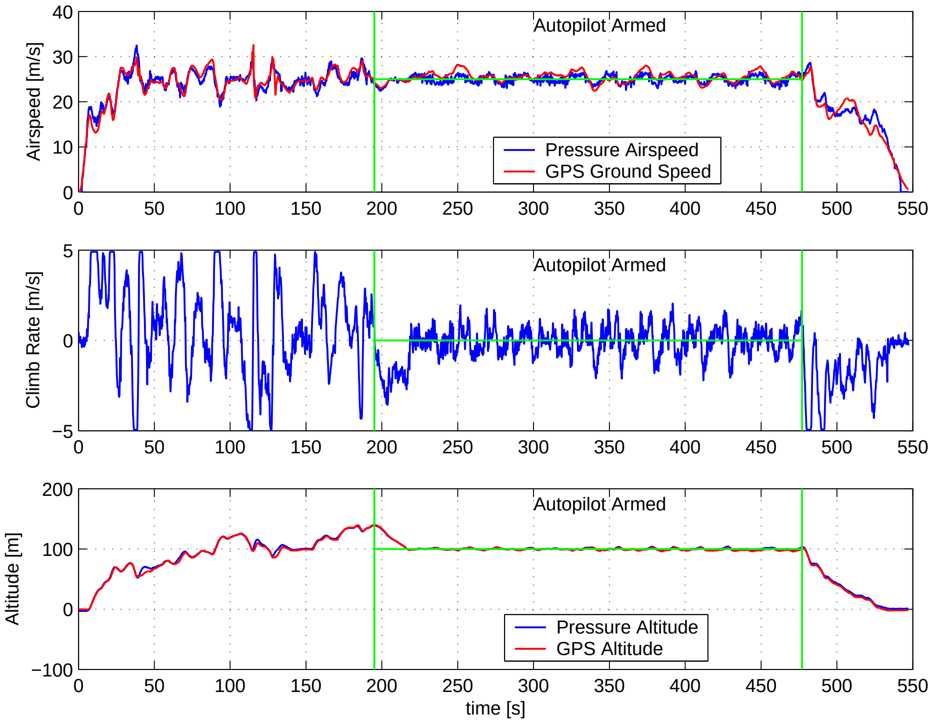


Figure 8.4: *Airspeed, Climb Rate and Altitude over an Entire Flight*

Typical longitudinal flight data accumulated over an entire test flight, is shown in Figure 8.4. The aircraft is under pilot control for the first 195s, after which the autopilot is armed with the airspeed and altitude controllers active. The airspeed command was set at 25m/s while the altitude command was set at 100m. With the autopilot armed, the RMS error in airspeed is calculated as 0.86m/s while the RMS error in altitude is calculated as 1.38m. These errors are based on the sensor measurement data and thus include the errors due to sensor noise. Also, changes in the aircraft's heading during the flight account for some of the 'noise' seen on the response plots. The autopilot is disarmed after 477s and the plane landed by the pilot. The barometric altitude is found to have drifted 3.8m over the entire flight time of 550s (9min 10s).

The GPS receiver's ground velocity and altitude measurements, used to calibrate the airspeed and barometric altitude, have also been plotted on the airspeed and altitude plots of Figure 8.4 respectively. The airspeed and ground speed are very similar with the differences arising from wind gusts, sensor noise and the aircraft's orientation (when ascending or descending, the airspeed will be higher than the ground speed). The barometric altitude and the GPS altitude are almost identical. The disadvantage of course with the GPS altitude measurement, is that updates are only available at 4Hz. However, the results do indicate that the aircraft could be controlled without the need for the airdata sensors. Further investigation into this subject is advised for future projects.

Airspeed Controller

A 5m/s step response in airspeed, together with the theoretical response from the block diagram simulator, is shown in Figure 8.5. The actual airspeed response displays more overshoot that the simulated response. This is most likely due to errors in the engine model because, as shown by the feedback gains of equation (5.16), the transient airspeed response is largely controlled by the engine thrust. To improve this response, more effort must be invested into the modeling of the engine. However, even with the slightly degraded response, the controller was found to regulate the airspeed satisfactorily.

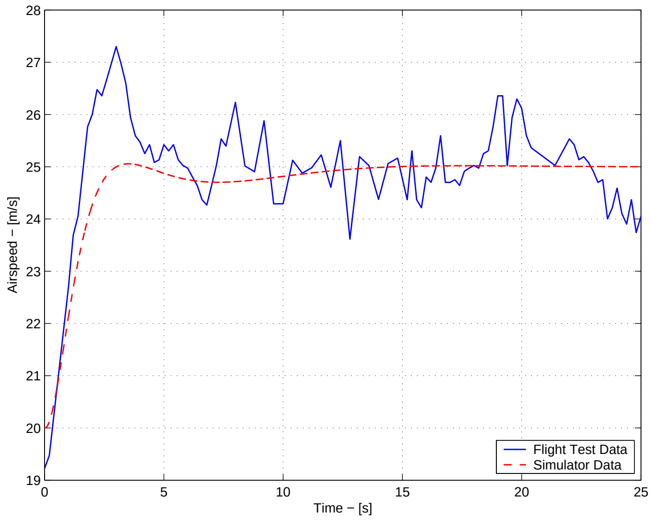


Figure 8.5: *Airspeed Controller Step Response*

Climb Rate Controller

The climb rate and altitude responses to a 2m/s step command in climb rate, are shown in Figure 8.6. The theoretical responses from the block diagram simulator are also shown plotted on the same axes.

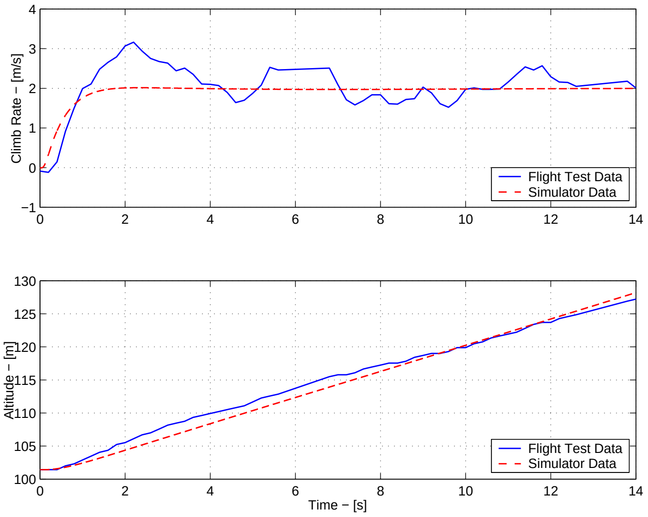


Figure 8.6: *Climb Rate Controller Step Command*

The inherent noise of the climb rate sensor makes it difficult to identify the aircraft's true response to the climb rate step command. However, a strong similarity between the actual and simulated step response can be seen. A clearer picture is obtained by looking at the altitude response to the climb rate step command. Here it is clear that the aircraft is ascending at approximately 2m/s.

Altitude Controller

The climb rate and altitude responses to a 10m step command in altitude are shown in Figure 8.7, together with the theoretical responses from the block diagram simulator. The step command is issued after 5s and it is clear that the actual altitude response is very similar to the theoretical response.

The climb rate and altitude responses to a 40m step command in altitude are shown in Figure 8.8. The theoretical results are plotted on the same axes. Note how the control system automatically enters the aircraft into a constant climb rate mode for the majority of the altitude step. The altitude response clearly verifies the working of the altitude controller.

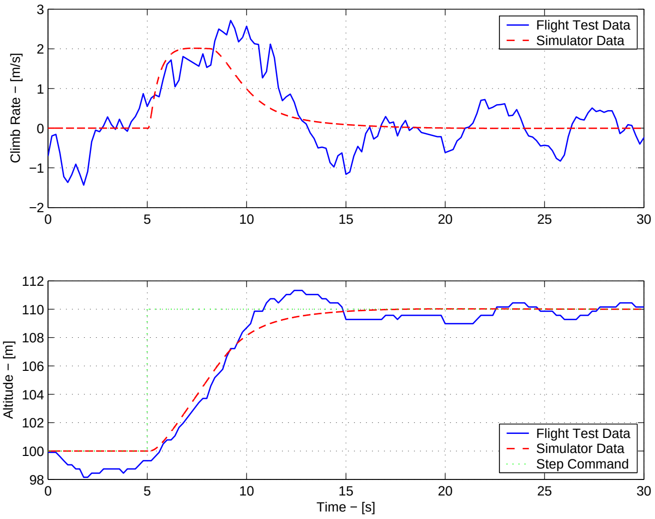


Figure 8.7: *Altitude Controller Step Response Without Constant Climb Rate Phase*

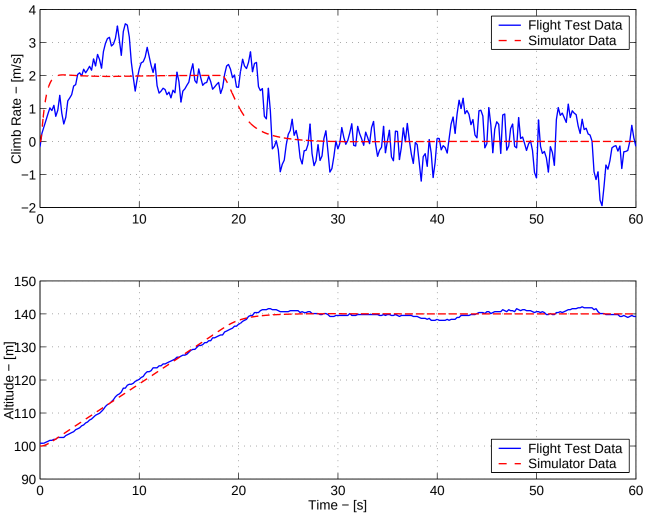


Figure 8.8: *Altitude Controller Step Response With Constant Climb Rate Phase*

8.2.2 Lateral Controllers

Yaw Rate Controller

A yaw rate step response is shown in Figure 8.9. The aircraft is initially flying with zero yaw rate and is issued a 15deg/s yaw rate command at 0s. This command is then returned to 0deg/s after 9.2 seconds. Except for the adverse yaw at the onset of the turn, the actual yaw rate response is very similar to the theoretical response. The fact that there is still adverse yaw illustrates the sensitivity of open loop control to the aircraft dynamic model. To counter the effect of adverse yaw, the washout filter of equation (5.28) must be adjusted until a satisfactory practical response is attained. Another, more robust approach, would be to provide feedback of the yaw rate signal to the rudder at the onset of the turn.

The airspeed and altitude controllers were also armed for the duration of the yaw rate step commands. Their responses are also plotted together with the theoretical responses in Figure 8.9. As discussed in Section 6.2.1, when the aircraft turns by banking, the vertical component of the lift vector is decreased and the aircraft begins to descend and build up airspeed. The opposite effect occurs when the aircraft returns from a banked state to straight and level flight. Figure 8.9 shows how the actual airspeed and altitude responses follow the theoretical responses. The practical altitude drop is approximately double that of the theoretical value. Without further investigation, it is difficult to determine the source of the degraded altitude controller performance. However, it is most likely to be due to the effects of asymmetric flow, experienced at the onset of the turn, at the pitot-static tube and on the aircraft's aerodynamic surfaces.

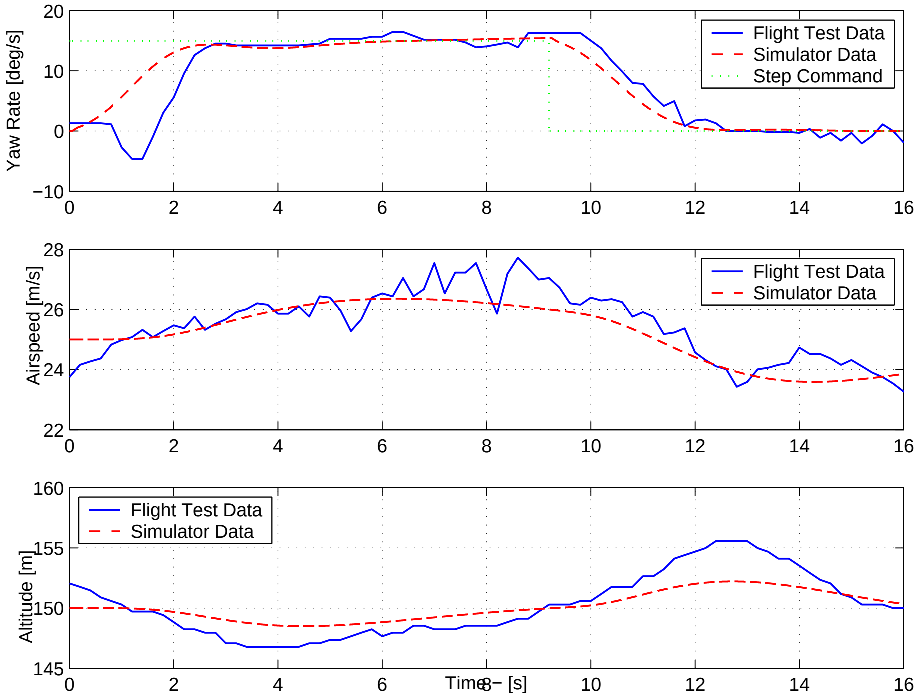


Figure 8.9: *Longitudinal and Lateral Controller Coupling*

A second yaw rate response to a 15deg/s step command is shown in Figure 8.10, together with the corresponding GPS receiver heading measurement. Note, that a yaw rate of 15deg/s is commanded after 5s but the steady state yaw rate value settles at approximately 14deg/s. Further investigation as to the origin of the offset, revealed that the yaw rate step command was issued while the aircraft was descending at 2m/s. Section 6.2.1 discussed the steady state errors that can occur in yaw rate when the aircraft's pitch angle is non-zero. In this example, the aircraft's descent implies a negative steady state pitch angle,

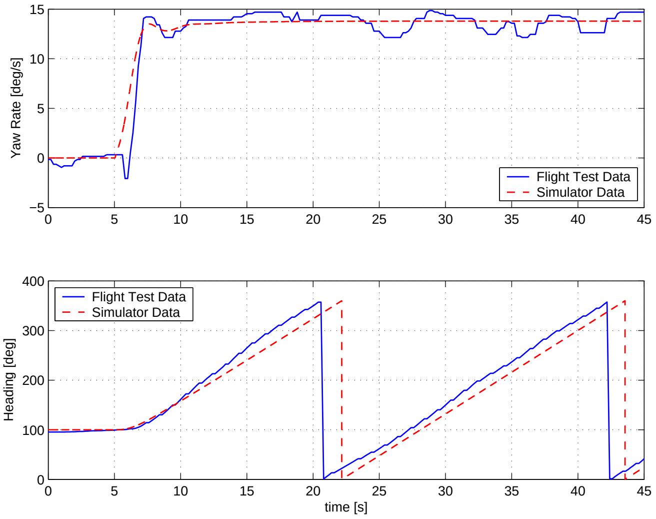


Figure 8.10: *Yaw Rate Controller*

which is responsible for the steady state error in yaw rate. A simulation was run with the aircraft descending at 2m/s and a yaw rate command of 15deg/s issued after 5s. The results are also plotted in Figure 8.10 and agree excellently with the practical results. The heading plot provides a clearer picture of the constant turn rate of the aircraft and shows that the practical turn rate agrees excellently with the simulated turn rate.

Heading Controller

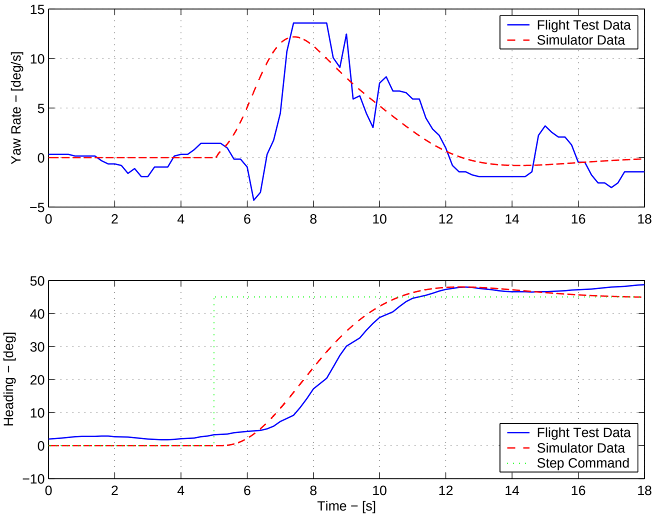


Figure 8.11: *Heading Controller Step Response Without Constant Yaw Rate Phase*

The yaw rate and heading responses, to a 45° step command in heading, are shown in Figure 8.11. The simulation responses are shown plotted on the same axes. The step command is issued after 5s and it is clear that the practical heading response closely matches the theoretical response. The small heading offset, visible before the step command is issued, is most likely due to biases on the roll and yaw rate gyroscopes as discussed in Section 5.3.3.

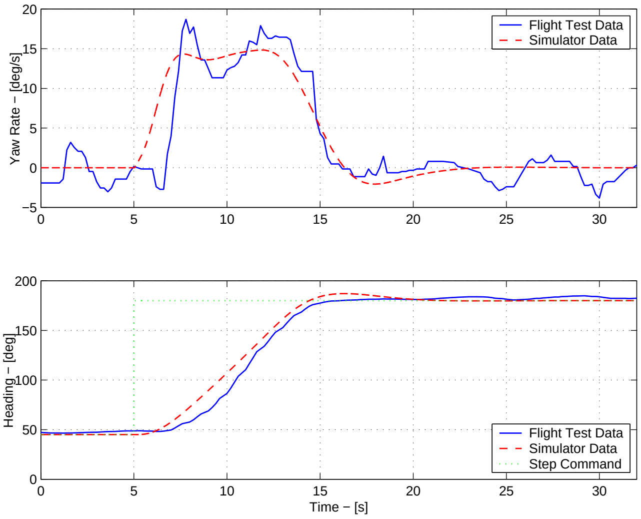


Figure 8.12: *Heading Controller Step Response With Constant Yaw Rate Phase*

The yaw rate and heading responses to a 135° step command in heading are shown in Figure 8.7, with the theoretical results plotted on the same axes. Note how the control system automatically enters the aircraft into a constant yaw rate mode for the majority of the heading step. The results clearly verify the working of the heading controller.

8.2.3 Navigator

A two dimensional (north-east) trajectory extract from the flight test data where the Navigator was armed is shown in Figure 8.13. The plot also shows the two relevant waypoints marked by stars, as well as the paths generated to navigate from one to the other. The aircraft begins at the ‘Start’ text of Figure 8.13 and moves to the ‘End’ Text.

It is clear that overall, the aircraft follows the desired trajectory. However, the accuracy with which it maintains the desired course is poor. The poor accuracy can largely be attributed to the errors built up during the constant turn rate sections. The first step to rejecting these errors would be to increase the cross track error feedback gain. However, simple proportional feedback will not be capable of tracking the circular path with zero steady state error, because a non-zero yaw rate command is required. An integrator term can be added to the compensator to provide this command. However, even with the addition of a pure integrator term, the dynamic response, when transitioning to a circular path section, is still expected to be poor. This is because the integrator state will require

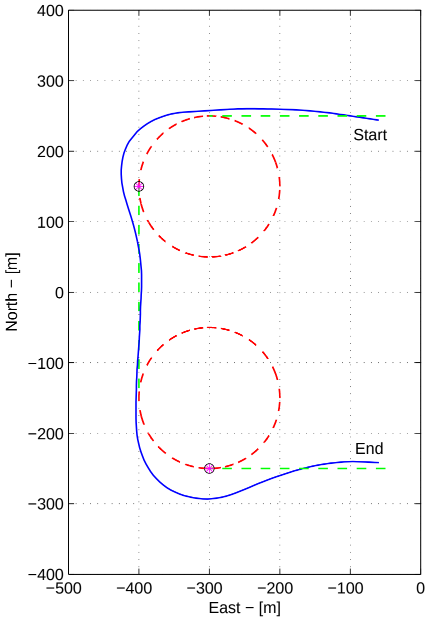


Figure 8.13: *Navigator*

time to build up to its steady state value if it is not to drive the system unstable. Two alternative solutions are discussed below.

The first solution involves using the fact that the circle radius and aircraft velocity are know. This knowledge allows the steady state yaw rate command, required for a circular section, to be manually provided in a feed-forward manner. This removes the need for a slow integrator term to provide the steady state yaw rate command and the approach is expected to yield satisfactory dynamic performance. A second solution is to use the non-linear guidance algorithm presented in [15]. This algorithm provides zero steady state error while tracking a circular path and is capable of providing excellent wind disturbance rejection.

8.3 Summary

This chapter described the three days of flight tests in which all of the controllers and the navigation algorithm of Chapters 5 and 6 were tested and shown to work in practice. Longitudinal, lateral and navigation flight test data was presented. In general, the practical results were found to match the theoretical results extremely well. Explanations for any deviations of the practical results from the theoretical results were provided and methods to improve the deficiencies in the guidance algorithm were also discussed.

Chapter 9

Summary and Recommendations

9.1 Summary

This thesis has reported the design and practical implementation of an autopilot for a conventional model aircraft. The aircraft's dynamic model was developed as a function of its physical parameters alone, which allows it to be readily applied to other conventional model aircraft. This model was then analysed and critically evaluated before being used in the autopilot controller design. A controller architecture suitable for implementation using low cost, off-the-shelf sensors was developed. The design was validated by a rigorous set of non-linear simulations. Finally, an avionics package and ground station was developed and the autopilot's excellent flight test results were presented.

The practical success of the autopilot can be largely attributed to the controller architecture and the extensive validation by non-linear simulations. The controller architecture made it possible to implement the autopilot using affordable hardware and minimised its performance dependency on the accuracy of the aircraft model. The two non-linear simulators developed allowed the controller code to be fully debugged and prepared for implementation on the on-board control microprocessors. The success of this process was demonstrated in achieving autonomous navigation in only three days of flight tests.

The following hardware and software developed during this project will be of value to future UAV work.

- MATLAB code for calculating an aircraft's stability and control derivatives from its physical parameters alone. This software automates the entire process of generating a linear aircraft model and simply requires the aircraft's parameters to be entered into a user friendly Excel file.

- Simulink block diagram non-linear flight simulator
- OpenGL based graphical non-linear flight simulator
- Comprehensive ground station GUI software
- Inertial Measurement Unit using solid state rate gyroscopes and accelerometers

9.2 Recommendations

Recommendations of how the current autopilot can be improved and extended are discussed in point form below. It is important however, to remember that any added complexity to a system also tends to increase its chances of failure. Thus, the additions must always be considered in light of what the final autopilot goals are.

- The autopilot developed up to this point has undergone three days of flight tests in which it performed autonomous navigation with no anomalies. Extensive testing and further study is still required to identify how the autopilot responds to unwanted situations such as stalls, spins, actuator saturation, engine failure and other potential scenarios. Thus, comprehensive practical testing is recommended to consolidate the autopilot design.

- The current autopilot has the hardware capability to perform an autonomous take-off. The yaw rate gyroscope and GPS heading measurements (once there is ground velocity) can be used in a complementary fashion to control the aircraft's rudder and nose wheel, to steer the aircraft down the runway. The airspeed sensor can be used to determine when to issue the elevator command for lift off. Once in the air, the airspeed and climb rate controller designed in this project can be used to guide the aircraft away from the ground. The graphical simulator will prove very useful during the development phase of this controller.

- To extend the autopilot to perform an autonomous landing would at least require the addition of a high precision altitude sensor for the final stages of landing. Provision is made in the current hardware and software design, for the addition of a Polaroid ultrasonic range finder. This sensor is capable of providing sub-centimetre accuracy altitude measurements over a range of 10m, and will thus compliment the current barometric altitude measurement well, for the purposes of landing. Strict lateral control for landing will also require additional sensors such as vision based or RF sensors. However, if lateral location is not critical (wide runway), then the controllers designed in this project will suffice. Again the graphical simulator will prove very useful in the design of this controller.

- Gain scheduling is commonly used to enhance the performance of autopilots. During conventional flight, the performance benefits are often negligible and the added effort of implementation makes the approach unattractive. However, if the autopilot is extended to perform automatic takeoffs and landings, then gain scheduling is recommended. This will ensure that the feedback gains remain valid for the aircraft's airspeed and orientation during either manoeuvre.

- The autopilot telemetry data indicated that the accuracy of the ground speed and altitude measurements, provided by the current GPS receiver, is similar to that of the differential and absolute pressure sensors (when there is little wind). The GPS receiver's measurement accuracy suggests that the autopilot could be implemented without the need for the airdata sensors. It is recommended that this approach is considered because it minimises the sensors required for the autopilot and removes the need for a pitot-static tube structure. However, the GPS only provides a measurement of *ground speed* and thus care must be taken to ensure the autopilot does not stall the aircraft in windy conditions.

Appendix A

Body Axes Transformation Equations

The equations used to transform the aircraft's inertial and geometric properties from the Reference Line body axis system to the Stability axis system are presented in Sections A.1 and A.2 respectively. Figure A.1 illustrates that the only difference between the two axis systems is the rotation through the trim angle of attack (α_0), about the OY_B-axis. The subscript b is used to indicate a variable quantified in the Reference Line body axis system, while variables without subscripts are coordinated in Stability axes. For further details on the transformation equations see [1].

A.1 Moments Of Inertia

$$I_x = I_{x_b} \cos^2 \alpha_0 + I_{z_b} \sin^2 \alpha_0 - 2I_{xz_b} \sin \alpha_0 \cos \alpha_0 \tag{A.1}$$

$$I_y = I_{y_b} \tag{A.2}$$

$$I_z = I_{z_b} \cos^2 \alpha_0 + I_{x_b} \sin^2 \alpha_0 + 2I_{xz_b} \sin \alpha_0 \cos \alpha_0 \tag{A.3}$$

$$I_{xy} = I_{xy_b} \cos \alpha_0 + I_{yz_b} \sin \alpha_0 \tag{A.4}$$

$$I_{xz} = I_{xz_b}(\cos^2 \alpha_0 - \sin^2 \alpha_0) + (I_{x_b} - I_{z_b}) \sin \alpha_0 \cos \alpha_0 \tag{A.5}$$

$$I_{yz} = I_{yz_b} \cos \alpha_0 - I_{xy_b} \sin \alpha_0 \tag{A.6}$$

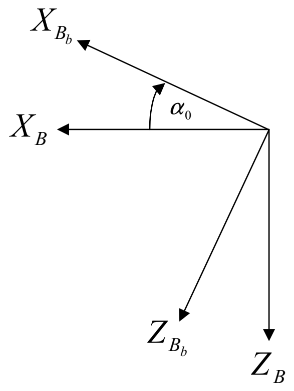


Figure A.1: *Stability and Reference Line Body Axes*

A.2 Geometric Properties

Aerodynamic

$$h = h_b \cos\alpha_0 \tag{A.7}$$

$$h_0 = h_{0_b} \cos\alpha_0 \tag{A.8}$$

$$l_T = l_{T_b} \cos\alpha_0 \tag{A.9}$$

$$l_F = l_{F_b} \cos\alpha_0 \tag{A.10}$$

$$h_F = h_{F_b} \cos\alpha_0 - l_{F_b} \sin\alpha_0 \tag{A.11}$$

Engine

$$\epsilon_T = \epsilon_{T_b} + \alpha_0 \tag{A.12}$$

$$m_T = m_{T_b} \tag{A.13}$$

Appendix B

Empirical Airfoil Equations

Empirical equations for the calculation of an airfoil's lift curve slope (due to angle of incidence), parasitic drag coefficient, induced drag coefficient and downwash variation with angle of incidence are provided in Section B.1. These equations can be used to calculate the respective airfoil coefficients for the aerodynamic model of Chapters 2 and 3.

It is difficult to accurately calculate the change in an airfoil's lift, roll moment and yaw moment coefficients due to actuator deflections using empirical data only. These coefficients are important because they directly affect the feedback gains of the control system. A vortex lattice solution method was used to calculate the required coefficients and was found to provide results with satisfactory accuracy. A program called AVL [15] was used by [29] to perform the vortex lattice calculations. The program accepts the aircraft's geometric information and trim flight condition, and calculates the required coefficients.

B.1 Empirical Equations

The following empirical equations are valid for airfoils experiencing subsonic, symmetric airflow at angles of incidence below the stall angle.

B.1.1 Lift Curve Slope

From [1], the theoretical maximum value of the lift curve slope for a rectangular flat plate airfoil of infinite span is given by,

$$\left.\frac{\partial C_L}{\partial \alpha}\right]_{\infty} = 2\pi \cos \Lambda_{le} \tag{B.1}$$

where, Λ_{le} is the airfoil leading edge sweep angle. For an airfoil of finite thickness, the equation becomes,

$$\left.\frac{\partial C_L}{\partial \alpha}\right]_{\infty} = 1.8\pi\left(1 + 0.8\frac{t}{c}\right) \cos \Lambda_{le} \tag{B.2}$$

where, t is the airfoil thickness and c is the standard mean chord. Finally, for an airfoil of finite span, the lift curve slope is further reduced as a function of the aspect ratio A,

$$\frac{\partial C_L}{\partial \alpha} = \frac{\frac{\partial C_L}{\partial \alpha}]_{\infty}}{\left(1 + \frac{\frac{\partial C_L}{\partial \alpha}]_{\infty}}{\pi A}\right)} \tag{B.3}$$

B.1.2 Parasitic Drag Coefficient

An equation for the parasitic drag coefficient of an airfoil [5] is,

$$C_{D_0} = \frac{K C_f S_{WET}}{S} \tag{B.4}$$

where,

- K = correction factor for pressure drag. (Empirical)
- C_f = skin friction coefficient. (Empirical)
- S_{WET} = airfoil wetted area: the actual area in contact with the air
- S = airfoil reference area

This equation is also easily adapted to calculate the parasitic drag of the fuselage by merely using the correct empirical data and setting S_{WET} to the fuselage wetted area. The wing area is used as the fuselage reference area.

B.1.3 Induced Drag Coefficient

An empirical equation for an airfoil's induced drag coefficient [5] is,

$$k = \frac{1}{\pi A e} \tag{B.5}$$

where, e is the empirically calculated airfoil *efficiency factor*.

B.1.4 Downwash Angle

An empirical equation for the downwash angle as a function of the lift coefficient is,

$$\epsilon = \frac{C_L}{\pi A} \tag{B.6}$$

Differentiating with respect to the angle of attack gives,

$$\frac{d\epsilon}{d\alpha} = \frac{1}{\pi A}\frac{\partial C_L}{\partial \alpha} \tag{B.7}$$

which provides a formula, as a function of the airfoil lift curve slope and aspect ratio, for the change in downwash angle with angle of attack.

Appendix C

Aircraft Data

The data provided in this section is based on the Reliance 0.46 size trainer aircraft, with an Irvine 0.53 cu.in engine, used in this project.

C.1 Aircraft Physical Data

C.1.1 Inertial

The aircraft inertial properties with the avionics set included are,

$$m = 3.25 \text{ kg} \tag{C.1}$$

$$\boldsymbol{I_B} = \begin{bmatrix} 0.12 & 0 & 0 \\ 0 & 0.17 & 0 \\ 0 & 0 & 0.30 \end{bmatrix} \text{kgm}^2 \tag{C.2}$$

The moments of inertia were found experimentally using the double pendulum setup shown in Figure C.1. The aircraft is suspended by two equally long strings such that the strings are parallel to the moment of inertia axis of concern. The aircraft is then set in motion by rotating it by a small angle about this axis, and the period of oscillation timed. The equation relating the period of oscillation to the moment of inertia is [15],

$$I = \frac{mgd^2}{4\pi^2 l} T^2 \tag{C.3}$$

where, T is the period of oscillation, l is the length of each string and d is the distance between each string and the axis of concern. For accurate results, it is important to ensure that the suspension strings are thin, that $d \ll l$ and that the perturbation angle is small [41].

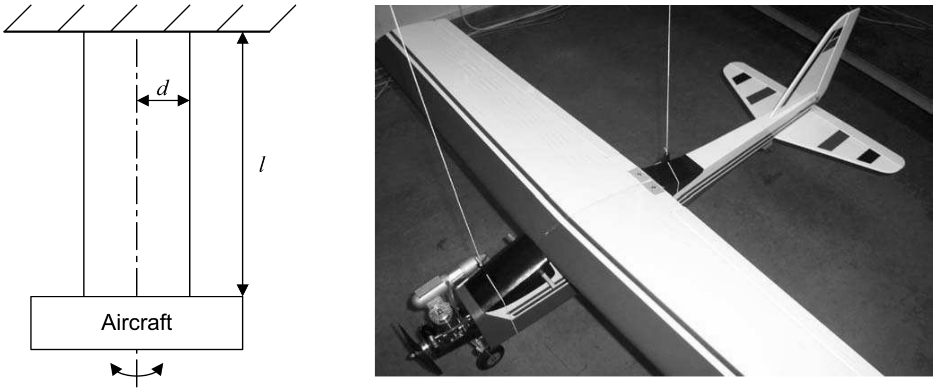


Figure C.1: *Diagram and Picture of Moment Of Inertia Measurement Procedure*

C.1.2 Airfoil Geometry

Each airfoil is assumed a straight edged, tapered surface. Given the measured airfoil data,

- b = Span
- t = Thickness
- Λ_{le} = Leading edge sweep angle
- c_r = Root chord
- c_t = Tip chord

the chord length variation over a semi-span can be written as,

$$c_y = c_r + \frac{2y}{b}(c_t - c_r) \tag{C.4}$$

where, y is the semi-span location from the root chord. The airfoil semi-span (s), planform area (S), mean aerodynamic chord ($\bar{c}$), standard mean chord (c), aspect ratio (A) and quarter chord sweep angle ($\Lambda_{\frac{1}{4}}$) are then calculated using the following equations,

$$s = \frac{b}{2} \tag{C.5}$$

$$S = \int_{-s}^{s} c_y dy \tag{C.6}$$

$$\bar{c} = \frac{1}{S}\int_{-s}^{s} c_y^2 dy \tag{C.7}$$

$$c = \frac{S}{b} \tag{C.8}$$

$$A = \frac{b}{c} \tag{C.9}$$

$$\Lambda_{\frac{1}{4}} = \tan^{-1}\left[\tan\Lambda_{le} + \frac{1}{4}\cdot\frac{c_t - c_r}{s}\right] \tag{C.10}$$

Table C.1 presents the geometric data for each aircraft airfoil where the wing, tailplane and fin subscripts (W,T,F) have been left out for clarity. The abbreviations (NA) and (ND) stand for Not Applicable and Non-dimensional respectively.

Symbol	**Wing**	**Tailplane**	**Fin**	**Unit**
b	1.650	0.595	NA	m
t	0.027	0.005	0.0050	m
Λ_{le}	-1.00	20.00	40.00	deg
c_r	0.290	0.205	0.270	m
c_t	0.240	0.085	0.100	m
s	0.825	0.298	0.270	m
S	0.437	0.086	0.050	m^2
$\bar{c}$	0.266	1.000	0.198	m
c	0.265	0.1450	0.185	m
A	6.226	4.103	1.460	ND
$\Lambda_{\frac{1}{4}}$	-1.87	14.74	34.28	deg

Table C.1: *Airfoil Geometric Data*

The moment arms for the wing, tailplane and fin are presented below,

$$h = \frac{0.097}{\bar{c}} = 0.365 \tag{C.11}$$

$$l_T = 0.685 \text{ m} \tag{C.12}$$

$$l_F = 0.715 \text{ m} \tag{C.13}$$

$$h_F = 0.114 \text{ m} \tag{C.14}$$

C.1.3 Engine Geometry

The engine model data is presented below,

$$\epsilon_T = 0^\circ \tag{C.15}$$

$$m_T = 0 \text{ m} \tag{C.16}$$

C.2 Trim Condition Data

The trim velocity was set at,

$$U_0 = 20 \text{ m/s} \tag{C.17}$$

Setting all small perturbation variables to zero, substituting the appropriate equations into the three trim condition equations (3.87) (3.88) and (3.89), and substituting the trim velocity, the remaining three non-zero trim states can be calculated,

$$\alpha_0 = 2.58^\circ \tag{C.18}$$

$$\delta_{E_0} = -1.70^\circ \tag{C.19}$$

$$T_0 = 2.338 \text{ N} \tag{C.20}$$

C.3 Aerodynamic Model Data

C.3.1 Aerodynamic Airfoil Data

The aerodynamic airfoil coefficients of the various aircraft surfaces are quantified using the empirical equations presented in Appendix B. All angles are in radians (rad).

Wing

$$C_{L_{W_0}} = 0 \qquad \frac{\partial C_{L_W}}{\partial \alpha} = 4.658 \tag{C.21}$$

$$C_{D_{W_0}} = 0.019 \tag{C.22}$$

$$\frac{\partial C_{l_W}}{\partial \delta_a} = -0.357 \tag{C.23}$$

$$C_{m_{W_0}} = 0 \tag{C.24}$$

$$\frac{\partial C_{n_W}}{\partial \delta_a} = 0.009 \tag{C.25}$$

where, the empirical data and wetted area used is,

$$K = 1.214 \qquad C_f = 5.9 \times 10^{-3} \qquad e = 0.850 \qquad S_{WET} = 0.892 \text{ m}^2 \tag{C.26}$$

The downwash angle change with angle of attack is,

$$\frac{d\epsilon}{d\alpha} = 0.238 \tag{C.27}$$

Tailplane

$$\frac{\partial C_{L_T}}{\partial \alpha} = 3.832 \tag{C.28}$$

$$\alpha_T = 0 \tag{C.29}$$

$$\frac{\partial C_{L_T}}{\partial \delta_e} = 2.828 \tag{C.30}$$

Fin

$$\frac{\partial C_{L_F}}{\partial \beta} = 2.250 \tag{C.31}$$

$$\frac{\partial C_{L_F}}{\partial \delta_r} = 1.791 \tag{C.32}$$

Fuselage

$$C_{D_{B_0}} = 0.0043 \tag{C.33}$$

where, the empirical data and wetted area is,

$$K = 1.200 \qquad C_f = 4.5 \times 10^{-3} \qquad S_{WET} = 0.353 \text{ m}^2 \tag{C.34}$$

and the wing reference area is used,

$$S_B = S_W = 0.437 \text{ m}^2 \tag{C.35}$$

C.3.2 Longitudinal Stability Derivatives

The aircraft's longitudinal stability derivatives, valid about the trim condition of Section C.2 are presented in Table C.2. The units of X and Z are in Newtons (N), M in Newton-metres (Nm), u and w in metres per second (m/s), q in radians per second (rad/s) and δ_e in radians (rad).

	$\boldsymbol{u}$	$\boldsymbol{w}$	$\boldsymbol{q}$	$\boldsymbol{\dot{w}}$	$\boldsymbol{\delta_e}$
$\boldsymbol{X}$	-0.350	0.543	0	0	0
$\boldsymbol{Z}$	-2.339	-29.115	-2.774	-0.033	-59.783
$\boldsymbol{M}$	0	-1.352	-1.900	-0.023	-40.951

Table C.2: *Longitudinal Stability Derivatives*

Referring to equations (3.97) and (3.98), the derivatives Z_q, $Z_{\dot{w}}$ and $M_{\dot{w}}$ were set to zero because,

- $Z_q \ll mU_0$
- $Z_{\dot{w}} \ll m$
- $M_{\dot{w}} \ll m\bar{c}$

C.3.3 Lateral Stability Derivatives

The aircraft's lateral stability derivatives, valid about the trim condition of Section C.2 are presented in Table C.3. The units of Y are in Newtons (N), L and N in Newton-metres (Nm), v in metres per second (m/s), p and r in radians per second (rad/s) and δ_a and δ_r in radians (rad).

	$\boldsymbol{v}$	$\boldsymbol{p}$	$\boldsymbol{r}$	$\boldsymbol{\delta_a}$	$\boldsymbol{\delta_r}$
$\boldsymbol{Y}$	-1.377	-0.157	0.985	0	21.912
$\boldsymbol{L}$	-0.780	-5.418	0.566	-63.0944	1.798
$\boldsymbol{N}$	0.985	-0.107	-0.755	1.5896	-15.677

Table C.3: *Lateral Stability Derivatives*

Referring to equation (3.106) the derivative Y_r was set to zero because,

- $Y_r \ll mU_0$

C.4 Gravitational Model Data

The gravitational force per unit mass is,

$$g = 9.81 \text{ m/s}^2 \tag{C.36}$$

C.5 Engine Model Data

The engine geometric data reduces the normal force and pitching moment stability and control derivatives to zero,

$$\frac{\partial Z_T}{\partial u} = \frac{\partial Z_T}{\partial \delta_t} = \frac{\partial M_T}{\partial u} = \frac{\partial M_T}{\partial \delta_t} = 0 \tag{C.37}$$

and the axial force stability and control derivatives become,

$$\frac{\partial X_T}{\partial u} = \frac{\partial T}{\partial u} = -\frac{T_0}{U_0} = 0.117 \text{ N/ms}^{-1} \tag{C.38}$$

$$\frac{\partial X_T}{\partial \delta_t} = 1 \tag{C.39}$$

The engine time constant was found by experiment to be,

$$\tau_T = 0.5 \text{ s} \tag{C.40}$$

Appendix D

Modal Decomposition

Consider the general state space system below,

$$\dot{\mathbf{x}} = \boldsymbol{A}\mathbf{x} + \boldsymbol{B}\mathbf{u} \tag{D.1}$$

$$\mathbf{y} = \boldsymbol{C}\mathbf{x} + \boldsymbol{D}\mathbf{u} \tag{D.2}$$

where $\mathbf{x}$ is an arbitrary state vector of length n, $\mathbf{u}$ is the input vector of length m and $\mathbf{y}$ is the output vector of length l. $\boldsymbol{A}$ is the $n \times n$ system matrix, B is the $n \times m$ input matrix, C is the $l \times n$ output matrix and D is a $l \times m$ matrix that is zero for all finite bandwidth systems.

The natural response of the system can be analysed by setting the input vector of equation (D.1) to zero ($\mathbf{u} = 0$) and recognising the well know solution for a set of linear, first order differential equations as,

$$\mathbf{x}(t) = \mathbf{x_0}e^{\lambda t} \tag{D.3}$$

To solve for λ, equation (D.3) is substituted into (D.1) to give,

$$\lambda\mathbf{x_0} = \boldsymbol{A}\mathbf{x_0} \tag{D.4}$$

which simplifies to,

$$(\lambda\boldsymbol{I} - \boldsymbol{A})\mathbf{x_0} = 0 \tag{D.5}$$

Equation (D.5) has non-trivial solutions when,

$$det(\lambda \boldsymbol{I} - \boldsymbol{A}) = 0 \tag{D.6}$$

Equation (D.6) is the *characteristic equation* of the system $\boldsymbol{A}$. It provides solutions for λ called eigenvalues, which are either real or occur in complex conjugate pairs. These solutions, when substituted back into equation (D.3), define the *natural modes* of motion of the system. Each eigenvalue is associated with a corresponding eigenvector, which can be visualised as a trajectory in the system space. The trajectory describes the ratios of the various states that make up the specific modes of motion and thus provides a physical interpretation of each mode.

Equation (D.4) can be written in matrix form for all n eigenvalue-eigenvector solutions,

$$\begin{bmatrix} \mathbf{e_1} & \mathbf{e_2} & \cdots & \mathbf{e_n} \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \lambda_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} \end{bmatrix} \begin{bmatrix} \mathbf{e_1} & \mathbf{e_2} & \cdots & \mathbf{e_n} \end{bmatrix} \tag{D.7}$$

where $\mathbf{e_k}$ represents the eigenvector corresponding to the eigenvalue λ_k where k runs from 1 to n. Equation (D.7) can be summarised as,

$$\boldsymbol{V}\boldsymbol{D} = \boldsymbol{A}\boldsymbol{V} \tag{D.8}$$

where, $\boldsymbol{V}$ is a matrix of eigenvectors and $\boldsymbol{D}$ is the corresponding matrix of eigenvalues. The elements of $\boldsymbol{V}$ and $\boldsymbol{D}$ are complex for each complex conjugate eigenvalue-eigenvector pair. To avoid dealing with complex matrices, it desired to write the eigenvalue and eigenvector matrices with real elements only. This can be done using a linear transformation to transform each complex conjugate eigenvalue-eigenvector pair to a real form as shown below,

$$\boldsymbol{V'} = \boldsymbol{V}\boldsymbol{T} \tag{D.9}$$

$$\boldsymbol{D'} = [\boldsymbol{T}^{-1}\boldsymbol{D}\boldsymbol{T}] \tag{D.10}$$

where $\boldsymbol{T}$ is the transformation matrix, $\boldsymbol{V'}$ and $\boldsymbol{D'}$ are matrices with only real elements, and the relationship of equation (D.8),

$$\boldsymbol{V'}\boldsymbol{D'} = \boldsymbol{A}\boldsymbol{V'} \tag{D.11}$$

still holds. If the 2 × 2 diagonal blocks of $\boldsymbol{T}$ are set to [3],

$$\boldsymbol{T} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2j} & 0 \\ \frac{1}{2} & -\frac{1}{2j} & 0 \\ 0 & 0 & \ddots \end{bmatrix} \tag{D.12}$$

for each complex conjugate pair of eigenvalues and eigenvectors, and it is recognised that,

$$Re(c) = \frac{c + c^*}{2} \tag{D.13}$$

$$Im(c) = \frac{c - c^*}{2j} \tag{D.14}$$

where c is an arbitrary complex number, then the transformed complex conjugate eigenvalue and eigenvector pairs of $\boldsymbol{V}$ and $\boldsymbol{D}$ become,

$$\boldsymbol{V'} = \begin{bmatrix} Re(\mathbf{e_1}) & Im(\mathbf{e_1}) & \cdots \end{bmatrix} \tag{D.15}$$

$$\boldsymbol{D'} = \begin{bmatrix} Re(\lambda_1) & Im(\lambda_1) & 0 \\ -Im(\lambda_1) & Re(\lambda_1) & 0 \\ 0 & 0 & \ddots \end{bmatrix} \tag{D.16}$$

Of course, the diagonal elements of the transformation matrix are merely unity for each real eigenvalue-eigenvector. Note that after the transformation, the matrices $\boldsymbol{V'}$ and $\boldsymbol{D'}$ have real elements only and that $\boldsymbol{D'}$ has 2 × 2 blocks on the diagonal for each complex conjugate eigenvalue-eigenvector pair.

Making $\boldsymbol{A}$ the subject of equation (D.11) and substituting it into equation (D.1),

$$\dot{\mathbf{x}} = \boldsymbol{V'}\boldsymbol{D'}\boldsymbol{V'}^{-1}\mathbf{x} + \boldsymbol{B}\mathbf{u} \tag{D.17}$$

Now, let

$$\mathbf{z} = \boldsymbol{V'}^{-1}\mathbf{x} \qquad \Rightarrow \qquad \mathbf{x} = \boldsymbol{V'}\mathbf{z} \tag{D.18}$$

then, the modal form of equations (D.1) and (D.2) is,

$$\dot{\mathbf{z}} = \boldsymbol{D'}\mathbf{z} + \boldsymbol{V'}^{-1}\boldsymbol{B}\mathbf{u} \tag{D.19}$$

$$\mathbf{y} = \boldsymbol{C}\boldsymbol{V}'\mathbf{z} + \boldsymbol{D}\mathbf{u} \tag{D.20}$$

where, $\mathbf{z}$ is the modal state vector and the system matrix is now diagonal for all real eigenvalues and blockwise diagonal if there are complex eigenvalues. Note that the matrix $\boldsymbol{D}$ in equation (D.20) is the $l \times m$ matrix from equation (D.2) and *not* the diagonal eigenvalue matrix of equation (D.8). The modal form provides maximum insight into the properties of the different system modes.

Appendix E

Avionics Details

E.1 Communication

E.1.1 Protocol

The string layout protocol used throughout the project is summarised below.

$\$AA\ data * CS$

where,

Identifier	**Details**
$	Start of string delimiter
AA	Two matching letters string identifier
data	Binary data string
*	End of string delimiter
CS	8 bit checksum

Table E.1: *String Layout Protocol Identifiers*

subject to the following rules,

- The checksum is formed by taking the exclusive OR of all characters in the string between but not including the start and end of string delimiters.
- If any byte in the data string is a start or end of string delimiter, then a second data character is placed in succession so as to avoid ambiguity.

- If the checksum is calculated to be an end of string delimiter character, then it is replaced by a ‘+’ to avoid confusion with the previous rule.

The ‘data’ part of the string is in binary form to increase the information density, especially for telemetry purposes where the baud rate is low.

E.1.2 Telemetry Packets

The various telemetry packets are summarised below. Telemetry packets are transmitted serially every 200ms at 9600 baud via the UART protocol. Including the effects of start and stop bits, this means that approximately 960 bytes can be transmitted per second, which means that a maximum of 192 bytes can be transmitted at 5Hz (200ms). The total telemetry packet size is expressed as a percentage of this maximum for the worst case scenario calculation that follows.

Primary

The Primary packet is sent at 5Hz.

- 3 bytes: Time stamp
- 12 bytes: Gyroscope and Accelerometer data
- 8 bytes: Rough altitude, Fine altitude, Airspeed, Climb rate
- 12 bytes: Autopilot servo commands
- 12 bytes: Safety Pilot servo commands
- TOTAL = 47 bytes ⇒ 24% of 5Hz transmit cycle

Secondary

The Secondary packet is sent at 1Hz.

- 2 bytes: Battery Voltage, RSSI, In range
- 6 bytes: Temperature, IMU reference voltage, Airdata reference voltage
- 4 bytes: Extra ADC channels

- 6 bytes: Magnetometer data
- TOTAL = 18 bytes ⇒ 9% of 5Hz transmit cycle

GPS

The GPS packet is sent at 4Hz.

- 4 bytes: Longitude
- 4 bytes: Latitude
- 2 bytes: Altitude
- 2 bytes: Velocity
- 2 bytes: Heading
- 1 bytes: Quality Information
- TOTAL = 15 bytes ⇒ 8% of 5Hz transmit cycle

Navigation

The Navigation packet is sent at 4Hz.

- 6 bytes: Aircraft North, East and Heading
- 2 bytes: Track error
- 4 bytes: Track heading and turn rate
- 6 bytes: Heading, airspeed and altitude command
- 3 bytes: Navigator status, next waypoint, number of waypoints
- TOTAL = 21 bytes ⇒ 11% of 5Hz transmit cycle

Worst Case

In the worst case, all of the telemetry packets are sent during the same transmit cycle and, including the communications protocol overhead, 52% of the cycle is used. This is acceptable when there is a strong RF link and no transmit retries are required. However, if the RF link becomes weak and the RF module begins to re-send packets, then some data will be lost.

E.2 Inertial Measurement Unit

The circuit diagram of the *Rate Gyroscope Signal Conditioning Block* and the *Accelerometer Signal Conditioning Block* of Figure 7.8 is shown in Figure E.1 below. The corresponding resistor and capacitor values are listed in Table E.2.

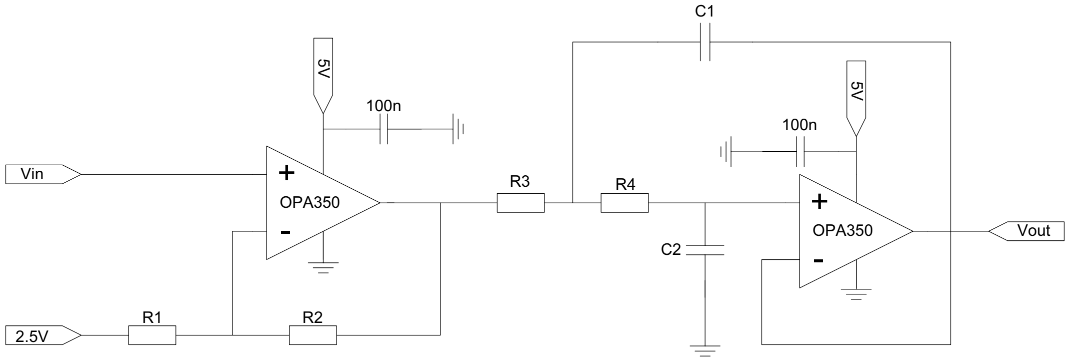


Figure E.1: *Circuit Diagram for Rate Gyroscope and Accelerometer Signal Conditioning Blocks*

Symbol	Rate Gyroscope	Accelerometer
$R1$	220k	220k
$R2$	330k	470k
$R3$	100k	100k
$R4$	100k	100k
$C1$	330n	330n
$C2$	100n	100n

Table E.2: *Rate Gyroscope and Accelerometer Signal Conditioning Block Component Values*

E.3 Airdata Board

E.3.1 Rough Altitude Signal Conditioning Block

The circuit diagram of the *Rough Altitude Signal Conditioning Block* of Figure 7.11 is shown in Figure E.2.

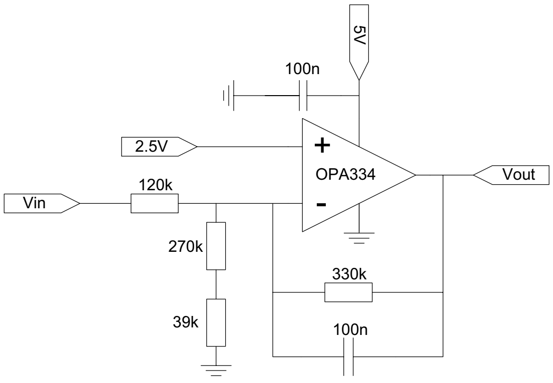


Figure E.2: *Circuit Diagram for Rough Altitude Signal Conditioning Block*

E.3.2 Fine Altitude Signal Conditioning Block

The circuit diagram of the *Fine Altitude Signal Conditioning Block* of Figure 7.11 is shown in Figure E.3.

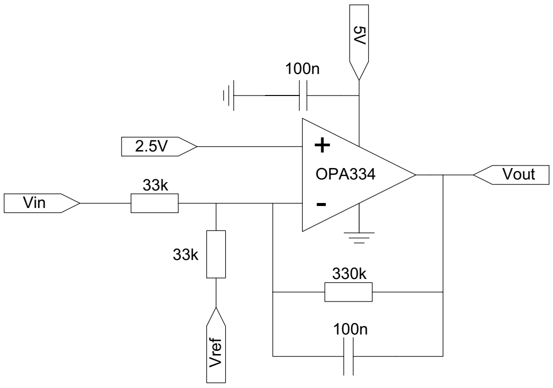


Figure E.3: *Circuit Diagram for Fine Altitude Signal Conditioning Block*

The voltage reference signal used to bias the amplifier (V_{ref}) is attained from Flight Control using the following formula,

$$V_{ref} = 5 - (V_{rough} + 0.2) \tag{E.1}$$

where, V_{rough} is the rough altitude measurement in volts. The formula biases the amplifier such that the output altitude can swing from -30m to 270m.

E.3.3 Climb Rate Signal Conditioning Block

The circuit diagram of the *Climb Rate Signal Conditioning Block* of Figure 7.11 is shown in Figure E.4.

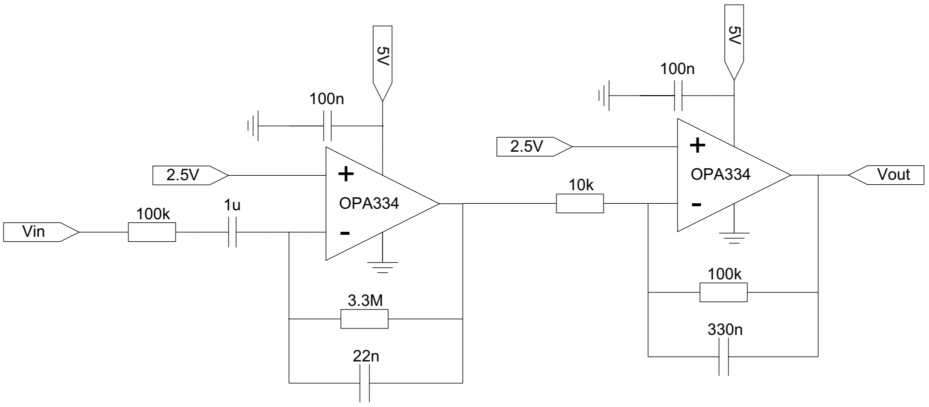


Figure E.4: *Circuit Diagram for Climb Rate Signal Conditioning Block*

Bibliography

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

[2] Louis V. Schmidt. *Introduction to Aircraft Flight Dynamics.* AIAA education series, 1998.

[3] Arthur E. Bryson, Jr. *Control of Spacecraft and Aircraft.* Princeton University Press, 1994.

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

[5] Richard S. Shevell. *Fundamentals of Flight.* Prentice-Hall, 1983.

[6] Bernard Etkin, Lloyd Duff Reid. *Dynamics of Flight, Stability and Control, 3rd ed.* John Wiley & Sons, 1996.

[7] Arthur Gelb. *Applied Optimal Estimation.* MIT Press, 1974.

[8] Paul Zarchan, Howard Musoff. *Fundamentals of Kalman Filtering - A Practical Approach.* Volume 190 of Progress in Astronautics and Aeronautics. AIAA, 200.

[9] Gene F. Franklin, J. David Powell, Abbas Emami-Naeini. *Feedback Control of Dynamic Systems, 3rd ed.* Addison-Wesley Publishing Company, 1994.

[10] Gene F. Franklin, J. David Powell, Michael Workman. *Digital Control of Dynamic Systems* Addison Wesely Longman, 1998.

[11] A.C. Kermode. *Mechanics of Flight.* London: Pitman, 1979.

[12] A.W. Babister. *Aircraft Stability and Control.* Pergamon Press, 1961.

[13] McRuer, Ashkenas and Graham. *Aircraft Dynamics and Automatic Control.* Princeton University Press, 1973.

[14] Paul Horowitz, Winfield Hill. *The Art of Electronics, 2nd ed.* Cambridge University Press, 1989.

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

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

[17] Craig Hennessey. *Autonomous Control of a Scale Airplane.* Bachelors of Engineering Science Thesis, Simon Fraser University, 2000.

[18] Max Baarspul. *A Review Of Flight Simulation Techniques.* Pergamon Press, 1990.

[19] Garth W. Milne. *Simplifying Vector Rotations and Coordination.* AIAA 2001-4310, 2001.

[20] Eric H. Biass. *Complete Guide to Drones and Sensors.* Armada International, 2001.

[21] B. Taylor, C.Bil, S. Watkins. *Horizon Sensing Attitude Stabilisation: A VMC Autopilot.* Presented to the 18^{th} International UAV Systems Conference, Bristol, UK, 2003.

[22] Scott M. Ettinger, Michael C. Nechyba, Peter G. Ifju, Martin Waszak. *Vision-Guided Flight Stability and Control for Micro Air Vehicles.* University of Florida, 2001.

[23] Nicol 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.

[24] Gray Welch, Gary Bishop. *An Introduction to the Kalman Filter.* UNC-Chapel Hill, TR 95-041, 2002.

[25] Dan Simon. *Kalman Filtering.* Innovatia Software, 1998.

[26] Alonzo Kelly. *A 3D State Space Formulation of a Navigation Kalman Filter for Autonomous Vehicles.* Carnegie Mellon University, CMU-RI-TR-94-19, 1994.

[27] Cloud Cap Technology web page. http://www.uavautopilots.com/. 2003.

[28] Sven Rönnbäck. *Development of a INS/GPS Navigation Loop for a UAV.* Masters Thesis, Luleå University of Technology, 2000.

[29] Duncan Palmer. Vacation work using AVL. 2004.

[30] Derek Kingston, Randal Beard, Timothy McLain, Michael Larsen, Wei Ren. *Autonomous Vehicle Technologies for Small Fixed Wing UAVS.* AIAA, 2003.

[31] Uy-Loi Ly *Stability and Control of Flight Vehicle.* Uy-Loi Ly, University of Washington, 1997.

[32] Eric N. Johnson, Sébastien G. Fontaine, Aaron D. Kahn. *Minimum Complexity Uninhabited Air Vehicle Guidance and Flight Control System.* Georgia Institute of Technologies, 2001.

[33] Garth W. Milne. *Navigation Work Package.* University of Stellenbosch, GWM-RPV-1, 2004.

[34] Gath W. Milne. *OpenGL Glider Simulator Code.* University of Stellenbosch, 2003.

[35] Graham C. Goodwin, Stefan F. Graebe, Mario E. Salgado. *Control System Design.* Prentice Hall, 2000.

[36] Jean-Jacques E.Slotine, Weiping Li. *Applied Nonlinear Control.* Prentice Hall, 1991.

[37] David C. Lay. *Linear Algebra and its Applications, 2nd ed.* Addison Wesely Longman, 1998.

[38] Donald A. Neamen. *Electronic Circuit Analysis and Design.* McGraw-Hill Companies. 1996.

[39] S.E. Van Bramer. *Pressure vs Altitude Relationships.* http://rcboat.com/tech-notes/multifilar.html, 2004.

[40] Albert-Jan Baerveldt, Robert Klang. *A Low-Cost and Low-Weight Attitude Estimation System for an Autonomous Helicopter.* Halmstad University, 1999.

[41] Brian Callahan. *Multifilar Suspension Method for Measuring Mass Moment of Inertia.* http://science.widener.edu/, 2004.

[42] Edward A. Haering, Jr. *Airdata Measurement and Callibration.* NASA - 104316, 1995.

[43] T.L. Clark, M.B. McCollum, D.H. Trout, K. Javor. *Marshall Space Flight Center Electromagnetic Compatibility Design and Interference Control (MEDIC) Handbook.* NASA, Reference Publication 1368, 1995.

[44] Micropilot web page. http://micropilot.com/. 2004.

[45] Unmanned Dynamics web page. http://u-dynamics.com/. 2004.

[46] Kentron web page. http://www.kentron.co.za/. 2004.

[47] Advanced Technologies and Engineering (ATE) web page. http://www.sadid.co.za/edition7/ate/ate.html. 2004.

[48] NASA History Office web page. http://history.nasa.gov/SP-367/appendc.htm. 2004.

[49] Marcel J. Sidi. *Spacecraft Dynamics & Control - A Practical Engineering Approach.* Cambridge University Press, 1997.