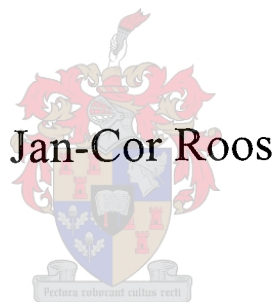




UNIVERSITEIT • STELLENBOSCH • UNIVERSITY  
jou kennisvennoot • your knowledge partner

# Autonomous Take-Off and Landing of a Fixed Wing Unmanned Aerial Vehicle



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 GARTH W. MILNE

CO-SUPERVISOR: MR IAIN K. PEDDLE

MARCH 2007

# Declaration

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

J.C. Roos:

Date: . 21 . . <sup>th</sup> February . 2007

# Abstract

The successful design and practical demonstration of controllers required for autonomous take-off and landing of a fixed wing unmanned aerial vehicle is presented. The project serves as an extension to the autonomous navigation autopilot developed in [1]<sup>1</sup>. All aspects of the controller design are presented, from modelling to practical testing and evaluation.

A non-linear model for the aircraft dynamics on the runway, a fixed pitch propeller model, a piston engine model and a wind gust model were developed. The conventional flight controllers were extended and improved to remove the single trim condition constraint. The avionics used for autonomous navigation were redesigned and extended to form a small, robust, reliable and professional hardware platform that includes hardware in the loop (HIL) simulation capabilities. Together with the non-linear models developed and the HIL capabilities from [22] a full non-linear hardware in the loop simulator was formed.

Throughout the development of the project the system was subjected to and verified through practical testing. The project saw the successful autonomous take-off and landing of the Reliance 0.46 size model aircraft.

---

<sup>1</sup> Iain K. Peddle. *Autonomous Flight of a Model Aircraft*. Masters dissertation, University of Stellenbosch. 2005.

# Uittreksel

Die suksesvolle ontwerp en praktiese demonstrasie van terugvoerbeheerders vir die outonome opstyg en land van 'n onbemande vastevlerkvliegtuig word aangebied. Die projek dien as 'n opvolg en uitbreiding van die vliegstuuroutomaat wat in [1]<sup>2</sup> ontwikkel is. Alle aspekte van die beheerderontwerp word aangebied, insluitend modulering, praktiese toetse en evaluering.

'n Nie-lineêre model van die vliegtuigdinamika op die aanloopbaan, 'n vaste-hellingpropellermodel, 'n klep enjinmodel en 'n windvraagmodel is ontwikkel. Die konvensionele vlugbeheerders is uitgebrei en verbeter om die enkele ewewigkondisiebeperking te verwyder. Die avionika wat gebruik is vir outonome navigasie is herontwerp om 'n klein, robuus, betroubare en professionele hardewareplatform te skep wat hardeware-in-die-lus simulaties kan uit voer. Die avionika, saam met die nie-lineêre modelle wat ontwikkel is en die hardeware-in-die-lus vermoë uit [22], word gekombineer om 'n volle nie-lineêre hardeware-in-die-lus simulator te vorm.

Met die ontwikkeling van die projek is die stelsel deurentyd blootgestel aan en geverifieer deur praktiese toetse. Die projek demostreer die suksesvolle outonome opstyg en land van die 0.46 skaal Reliance modelvliegtuig.

---

<sup>2</sup> Iain K. Peddle. *Autonomous Flight of a Model Aircraft*. Masters dissertation, University of Stellenbosch. 2005.



# Acknowledgements

This project would not have been possible without the support, guidance and investment of numerous people and institutions. I would like to thank everybody who assisted me. In particular I would like to extend my gratitude to.

- Prof. G.W. Milne, for your guidance and ideas that provided me with the solutions to a multitude of problems, as well as the support in aircraft construction.
- Mr. Iain Peddle, for your ideas, support and friendship. The hours sacrificed to help solve implementation problems and reading this document is very much appreciated. The foundation for this project was also of the highest standard. Thank you.
- Denel Aerospace Systems, for funding the project.
- Dr. Thomas Jones, for your support and advice.
- Willie van Rooyen, Ulrich Buttner, Johan Arendse and other staff members from the University of Stellenbosch. Your support is greatly appreciated.
- All of my friends in the ESL, especially Johan Bijker, Willem Hough and Keith Brown for the assistance during practical testing, discussion of topics and the many laughs.
- Dr. Kas Hamman, for doing an excellent job as safety pilot.
- Helderberg Radio Flyers Club, for allowing us to use their airfield.
- My parents, sister and other family members for the continuous and unbelievable love, patience and support.

# Contents

Chapter 1	<i>Introduction and Overview</i> .....	1
1.1	Background and History .....	1
1.2	Project Starting Platform.....	3
1.2.1	Aircraft and Model .....	3
1.2.2	Controllers .....	5
1.2.3	Navigation Algorithm.....	6
1.2.4	Avionics and Ground Station.....	6
1.3	Project Goals.....	9
1.4	Achievements.....	9
1.5	Thesis Outline .....	10
Chapter 2	<i>System Design</i> .....	11
2.1	Take-Off.....	11
2.1.1	Take-Off Sequence .....	12
2.1.2	Take-Off Controllers .....	13
2.2	Landing .....	15
2.2.1	Landing Sequence.....	16
2.2.2	Landing Controllers.....	17
2.2.3	Wind Estimation .....	19
2.2.4	Landing Path.....	21
2.3	Cross Track Error Algorithm .....	23
2.4	Controller Overview .....	24
2.5	Summary .....	26
Chapter 3	<i>Modelling</i> .....	27
3.1	Non-Linear Runway Model .....	27
3.1.1	Axis System Definition.....	27
3.1.2	Runway Forces .....	28
3.1.3	Resultant Forces and Moments.....	34
3.2	Linear Runway Models.....	36
3.2.1	Linear Longitudinal Runway Model.....	36
3.2.2	Linear Lateral Runway Model.....	38
3.3	Engine Model.....	41
3.3.1	Fixed Pitch Propeller Model.....	41
3.3.2	Non-Linear Engine Model.....	41
3.4	Wind Gust Model.....	43

3.5	Summary .....	43
Chapter 4	<i>Take-Off Control and Simulation</i> .....	44
4.1	$X_p$ -Acceleration Controller .....	44
4.2	Groundspeed Controller .....	47
4.3	Pitch Rate Damper .....	50
4.4	Climb Rate Controller .....	50
4.5	Runway Yaw Rate Controller .....	53
4.6	Turn Radius Interface .....	55
4.7	Runway Heading Controller .....	56
4.8	Runway Guidance Controller .....	59
4.9	Summary .....	62
Chapter 5	<i>Landing Control and Simulation</i> .....	63
5.1	Conventional Flight Controllers .....	63
5.1.1	Gain Scheduling .....	63
5.1.2	Dutch Roll Damper .....	66
5.1.3	Guidance Controller .....	67
5.2	Glide Path Controller .....	69
5.3	Flare Controller .....	71
5.4	De-Crab Controller .....	74
5.5	Summary .....	77
Chapter 6	<i>Avionics and Ground Station</i> .....	78
6.1	Avionics .....	78
6.1.1	Redesign and Packaging .....	79
6.1.2	Differential Pressure Sensor .....	81
6.1.3	Inertial Measurement Unit .....	82
6.1.4	PWM Receiver Interface .....	82
6.1.5	RF Module and GPS Signal Conditioning .....	84
6.1.6	Multiplexed Programming .....	84
6.1.7	Hardware in the Loop Interface .....	85
6.1.8	Ultrasonic Range Sensor and Magnetometer .....	87
6.1.9	Software .....	89
6.2	Ground Station .....	91
6.2.1	Ground Station Software .....	92
6.3	Summary .....	94
Chapter 7	<i>Non-linear Simulation (Hardware in the Loop)</i> .....	96
7.1	Hardware in the Loop Simulator .....	96
7.1.1	HIL Simulator Overview .....	96
7.1.2	Aircraft Model .....	98
7.1.3	Autopilot Model .....	98
7.1.4	Graphical Output .....	99
7.2	Controller Non-Linear Simulation .....	100

7.2.1	Take-Off Controllers .....	100
7.2.2	Take-Off Sequence .....	103
7.2.3	Landing Controllers .....	107
7.2.4	Landing Sequence .....	108
7.2.5	Wind Estimation .....	111
7.3	Summary .....	111
Chapter 8	<i>Flight Tests</i> .....	112
8.1	Flight Tests Overview .....	112
8.1.1	Flight Test 1 – Redesigned Avionics .....	113
8.1.2	Flight Test 2 – Conventional Flight Controllers, Sensors Validation and Wind Estimation .....	113
8.1.3	Flight Test 3 – Runway Controllers and Landing Approaches .....	114
8.1.4	Flight Test 4 – Autonomous Taxing and Lateral Assisted Take-Off .....	115
8.1.5	Flight Test 5 – Autonomous Take-Off and Landing .....	115
8.2	Sensor Evaluation .....	116
8.2.1	Global Positioning System Receiver .....	116
8.2.2	Ultrasonic Range Sensor .....	117
8.2.3	Absolute Pressure Sensor .....	118
8.3	Take-Off .....	120
8.3.1	Runway Yaw Rate Controller .....	120
8.3.2	Radius Interface .....	120
8.3.3	Runway Heading Controller .....	121
8.3.4	Groundspeed and Runway Guidance Controllers (Autonomous Taxing) .....	122
8.3.5	Autonomous Take-Off .....	124
8.4	Landing .....	127
8.4.1	Conventional Flight Controllers .....	127
8.4.2	Autonomous Landing .....	128
8.5	Summary .....	132
Chapter 9	<i>Summary and Recommendations</i> .....	133
9.1	Summary .....	133
9.2	Recommendations .....	133
Bibliography	.....	137
Appendix A	<i>Avionics Details</i> .....	140
A.1	Microcontrollers .....	140
A.2	Inertial Measurement Unit .....	141
A.3	Air Data .....	141
A.4	Telemetry Packets .....	142
A.5	System Control and Status Bytes .....	143
A.6	Circuit Diagrams .....	145
Appendix B	<i>Additional Controller Information</i> .....	146
B.1	Conventional Flight Controllers .....	146
B.1.1	Gain Scheduling .....	146

B.1.2	Guidance Controller.....	148
B.2	$X_B$ -Acceleration Controller.....	149
B.3	Runway Yaw Rate Controller.....	149
B.4	Runway Heading Controller.....	150
B.5	Runway Guidance Controller.....	150
B.6	Landing Sequence Parameters.....	151
Appendix C	<i>Aircraft Data</i> .....	152
C.1	Engine Data.....	152
C.2	Propeller Data.....	153
C.2.1	Inertia.....	154
C.3	Sensor Noise.....	154
Appendix D	<i>Stall Speed Calculations</i> .....	155
Appendix E	<i>Runway Model Initial CG and <math>\Theta</math> Calculations</i> .....	158
Appendix F	<i>Autonomous Flight of a Model Aircraft – Additional Information</i> .....	160
F.1	Axis System Definition.....	160
F.1.1	Inertial Axes.....	160
F.1.2	Body Fixed Axes.....	161
F.2	Equations of Motion.....	161
F.3	Linear Aircraft Model.....	162
F.3.1	Linear Equations of Motion.....	162
F.3.2	Linear Aircraft Model.....	162
F.4	Aircraft State Space Model.....	163
Appendix G	<i>Flight Test Plan</i> .....	165
G.1	Introduction.....	165
G.1.1	Background.....	165
G.1.2	Test Objectives.....	165
G.1.3	Description of the Test Item.....	166
G.1.4	Test Scope.....	167
G.2	Test Preparation.....	167
G.2.1	Project Team.....	167
G.2.2	Operations Support Requirements.....	167
G.2.3	Flight Test Engineering Support Requirements.....	167
G.2.4	Logistical Support.....	168
G.2.5	Test Operations Plan.....	168
G.2.6	Briefings and Debriefings.....	168
G.2.7	Safety.....	168
G.3	Details of the Test.....	168
G.3.1	Ground Controllers.....	168
G.3.2	Piloted Flight with Telemetry Logging and Runway Flybys.....	170
G.3.3	Longitudinal Controllers.....	172

G.3.4	Lateral Controllers .....	174
G.3.5	Autonomous Navigation and Wind Estimation .....	177
G.4	Test Schedule .....	179
G.5	Test Hazard Analysis .....	180

# List of Figures

Figure 1.1: <i>ESL UAV Fleet</i> .....	2
Figure 1.2: <i>Reliance 0.46 Size RC Trainer Aircraft [1]</i> .....	3
Figure 1.3: <i>Engine Thrust Geometry [1]</i> .....	4
Figure 1.4: <i>Original Autopilot Overview</i> .....	5
Figure 1.5: <i>Path Planner Graphical Output [1]</i> .....	6
Figure 1.6: <i>Original Avionics [1]</i> .....	7
Figure 1.7: <i>Original Avionics Block Diagram</i> .....	7
Figure 1.8: <i>Complete System from [1]</i> .....	8
Figure 1.9: <i>Thesis Chapter Flow</i> .....	10
Figure 2.1: <i>Take-Off Sequence Phases</i> .....	12
Figure 2.2: <i>Landing Sequence Phases</i> .....	16
Figure 2.3: <i>Wind Estimation Procedure</i> .....	20
Figure 2.4: <i>Landing Path Geometry (Side View)</i> .....	21
Figure 2.5: <i>Landing Path Geometry (Top View)</i> .....	21
Figure 2.6: <i>Cross Track Error Geometry</i> .....	23
Figure 2.7: <i>Complete Autopilot Overview</i> .....	24
Figure 3.1: <i>Top View of the Under Carriage with Friction Forces</i> .....	29
Figure 3.2: <i>Top View of the Under Carriage with Cornering Forces</i> .....	29
Figure 3.3: <i>Pictorial Descriptions of the Slip Angle</i> .....	30
Figure 3.4: <i>3D View of the Under Carriage with Normal Forces</i> .....	31
Figure 3.5: <i>Top View of the Under Carriage with Runway Forces</i> .....	34
Figure 3.6: <i>Rear View of the Under Carriage with Runway Forces</i> .....	35
Figure 3.7: <i>Side View of the Under Carriage with Runway Forces</i> .....	35
Figure 3.8: <i>Engine Thrust vs. Airspeed and Throttle Actuation</i> .....	42
Figure 3.9: <i>Wind Gust Model Block Diagram</i> .....	43
Figure 3.10: <i>Simulated Wind Gust</i> .....	43
Figure 4.1: <i><math>X_B</math>-Acceleration Controller Block Diagram</i> .....	45
Figure 4.2: <i><math>X_B</math>-Acceleration Controller Root Locus</i> .....	46
Figure 4.3: <i><math>X_B</math>-Acceleration Controller Linear Simulation</i> .....	47
Figure 4.4: <i>Groundspeed Controller Block Diagram</i> .....	48
Figure 4.5: <i>Groundspeed Controller Root Locus</i> .....	49
Figure 4.6: <i>Groundspeed Controller Linear Simulation</i> .....	49
Figure 4.7: <i>Pitch Rate Damper Block Diagram</i> .....	50
Figure 4.8: <i>Climb Rate Controller Block Diagram</i> .....	51
Figure 4.9: <i>Climb Rate Controller Root Locus (Top) with Origin Magnification (Bottom)</i> .....	52
Figure 4.10: <i>Climb Rate Controller Linear Simulation</i> .....	53



Figure 4.11: <i>Runway Yaw Rate Controller Block Diagram</i> .....	54
Figure 4.12: <i>Runway Yaw Rate Controller Linear Simulation</i> .....	55
Figure 4.13: <i>Turn Radius Interface Block Diagram</i> .....	55
Figure 4.14: <i>Turn Radius Linear Simulation</i> .....	56
Figure 4.15: <i>Runway Heading Controller Block Diagram</i> .....	57
Figure 4.16: <i>Runway Heading Controller Root Locus</i> .....	58
Figure 4.17: <i>Runway Heading Controller Linear Simulations</i> .....	59
Figure 4.18: <i>Runway Heading Controller Linear Simulation - Saturated Response</i> .....	59
Figure 4.19: <i>Runway Guidance Controller Block Diagram</i> .....	60
Figure 4.20: <i>Runway Guidance Controller Root Locus at 4m/s</i> .....	61
Figure 4.21: <i>Runway Guidance Controller Linear Simulation</i> .....	62
Figure 5.1: <i>Airspeed and Climb Rate Controller Block Diagram</i> .....	64
Figure 5.2: <i>Gain Scheduled Airspeed and Climb Rate Multi Root Locus Plots</i> .....	65
Figure 5.3: <i>Gain Scheduled Airspeed and Climb Rate Controller Linear Simulation</i> .....	65
Figure 5.4: <i>Yaw Orientation Controller Block Diagram</i> .....	66
Figure 5.5: <i>Gain Scheduled Altitude, Yaw Rate and Guidance Controllers Linear Simulation</i> .....	66
Figure 5.6: <i>Dutch Roll Damper Block Diagram</i> .....	67
Figure 5.7: <i>Heading and Guidance Controller Block Diagram</i> .....	68
Figure 5.8: <i>High-Gain and Original Guidance Controllers Linear Simulation</i> .....	68
Figure 5.9: <i>High-Gain and Original Guidance Controllers Linear Simulation with Gyroscope Biases</i> .....	69
Figure 5.10: <i>Glide Path Graphical Description</i> .....	69
Figure 5.11: <i>Glide Path Controller Block Diagram</i> .....	70
Figure 5.12: <i>Glide Path Controller Linear Simulation</i> .....	70
Figure 5.13: <i>Flare Path Graphical Description</i> .....	71
Figure 5.14: <i>Flare Controller Block Diagram</i> .....	71
Figure 5.15: <i>Sink Rate vs. Equivalent Drop Height</i> .....	73
Figure 5.16: <i>Ideal Flaring Path and Sink Rate</i> .....	73
Figure 5.17: <i>De-Crab Controller Block Diagram</i> .....	74
Figure 5.18: <i>De-Crab Controller Root Locus</i> .....	75
Figure 5.19: <i>De-Crab Controller Linear Simulation</i> .....	76
Figure 5.20: $\phi, r, \delta_a, v$ while de-Crabbing .....	76
Figure 5.21: <i>Magnetometer Heading Error with Roll and Pitch Angles</i> .....	77
Figure 6.1: <i>Avionics and Ground Station Block Diagram</i> .....	79
Figure 6.2: <i>Avionics Printed Circuit Board</i> .....	80
Figure 6.3: <i>Intermediate Board</i> .....	80
Figure 6.4: <i>Packaged Avionics</i> .....	81
Figure 6.5: <i>Motorola MPXV5004DP Differential Pressure Sensor [18]</i> .....	81
Figure 6.6: <i>Differential Pressure Signal Conditioning Block Diagram</i> .....	81
Figure 6.7: <i>Inertial Measurement Unit</i> .....	82
Figure 6.8: <i>Receiver Interface Block Diagram</i> .....	83
Figure 6.9: <i>Receiver Interface CPU Flowchart</i> .....	83
Figure 6.10: <i>MaxStream XStream OEM 2.4GHz RF Module</i> .....	84
Figure 6.11: <i>GPS Signal Conditioning Block Diagram</i> .....	84



Figure 6.12: <i>Multiplexed Programming Block Diagram</i> .....	85
Figure 6.13: <i>Programming Board</i> .....	85
Figure 6.14: <i>HIL Interface Block Diagram</i> .....	86
Figure 6.15: <i>HIL Board Connected to the Avionics</i> .....	86
Figure 6.16: <i>SensComp Ultrasonic Sensor [18]</i> .....	87
Figure 6.17: <i>Ultrasonic Interface Block Diagram</i> .....	87
Figure 6.18: <i>Ultrasonic Sensor Mounted in the Aircraft</i> .....	87
Figure 6.19: <i>Honeywell Magnetometer [18]</i> .....	88
Figure 6.20: <i>Magnetometer Interfacing Block Diagram</i> .....	88
Figure 6.21: <i>Flight Control Flow Diagram</i> .....	89
Figure 6.22: <i>Mission Control Flow Diagram</i> .....	90
Figure 6.23: <i>Mission Flight Logic Flow Diagram</i> .....	91
Figure 6.24: <i>Ground Station Page</i> .....	92
Figure 6.25: <i>Sensor and Actuator Pages</i> .....	93
Figure 6.26: <i>Conventional Flight + Take-Off and Landing Control Pages</i> .....	93
Figure 6.27: <i>Test Sequence Page</i> .....	94
Figure 6.28: <i>Avionics and Ground Station</i> .....	95
Figure 7.1: <i>Hardware in the Loop Simulation Setup</i> .....	97
Figure 7.2: <i>HIL Simulator Structure</i> .....	97
Figure 7.3: <i>Autopilot Model for HIL Simulator</i> .....	98
Figure 7.4: <i>Avionics Connected for HILS</i> .....	99
Figure 7.5: <i>Screenshot from the Graphical Display</i> .....	99
Figure 7.6: <i>Acceleration and Groundspeed Controller HIL Simulation</i> .....	100
Figure 7.7: <i>Runway Yaw Rate Controller HIL Simulation</i> .....	101
Figure 7.8: <i>Radius Interface HIL Simulation</i> .....	101
Figure 7.9: <i>Runway Heading Controller HIL Simulation</i> .....	102
Figure 7.10: <i>Runway Guidance Controller HIL Simulation</i> .....	102
Figure 7.11: <i>Heading and Turn Rate during Guidance Controller HIL Simulation</i> .....	103
Figure 7.12: <i>Take-Off Sequence Phases (Repeat)</i> .....	103
Figure 7.13: <i>Screenshot of the Aircraft shortly after Take-Off</i> .....	104
Figure 7.14: <i>Groundspeed Response during a HIL Take-Off</i> .....	104
Figure 7.15: <i>Longitudinal Control during a HIL Take-Off</i> .....	105
Figure 7.16: <i>Lateral Control during a HIL Take-Off</i> .....	105
Figure 7.17: <i>Longitudinal Control during a Crosswind HIL Take-Off</i> .....	106
Figure 7.18: <i>Lateral Control during a Crosswind HIL Take-Off</i> .....	107
Figure 7.19: <i>Glide Path Controller HIL Simulation</i> .....	107
Figure 7.20: <i>Landing Sequence Phases (Repeat)</i> .....	108
Figure 7.21: <i>Screenshots of the Aircraft just After Landing (Left) and on the Landing Approach (Right)</i> .....	108
Figure 7.22: <i>Longitudinal Control during a HIL Landing</i> .....	109
Figure 7.23: <i>Lateral Control during a HIL Landing</i> .....	109
Figure 7.24: <i>Ultrasonic Altitude and Pressure Climb Rate during a HIL Flare</i> .....	110
Figure 7.25: <i>Pitch Angle and Rate during HIL Flare</i> .....	110
Figure 8.1: <i>The Project Team</i> .....	113

Figure 8.2: Assistant Willem Hough and the Author Refuelling.....	115
Figure 8.3: Safety Pilot Kas Hamman and Assistant Willem Hough during Pre-Flight Checks.....	116
Figure 8.4: GPS Cross Track Error and Range after Three Flights .....	116
Figure 8.5: Ultrasonic Altitude and Climb Rate during a Manual Landing.....	117
Figure 8.6: Aircraft during a Manual Landing.....	118
Figure 8.7: Altitude Measurement during a Manual Runway Flyby .....	118
Figure 8.8: Aircraft during a Manual Runway Flyby .....	119
Figure 8.9: Pressure Recalibration with the Ultrasonic.....	119
Figure 8.10: Runway Yaw Rate Controller Step Response.....	120
Figure 8.11: Radius Interface Response to a 4m Command.....	121
Figure 8.12: Runway Heading Controller Step Response .....	121
Figure 8.13: Longitudinal Response during Autonomous Taxing .....	122
Figure 8.14: Lateral Response during Autonomous Taxing .....	123
Figure 8.15: Aircraft Autonomously Taxing.....	123
Figure 8.16: Take-Off Sequence Phases (Repeat) .....	124
Figure 8.17: Aircraft during an Autonomous Take-Off.....	124
Figure 8.18: Groundspeed and Acceleration Control during an Autonomous Take-Off.....	124
Figure 8.19: Longitudinal Control during an Autonomous Take-Off.....	125
Figure 8.20: Aircraft Autonomously Rotating .....	126
Figure 8.21: Lateral Control during an Autonomous Take-Off.....	126
Figure 8.22: Aircraft during an Autonomous Climb Out.....	127
Figure 8.23: Conventional Flight Controllers Responses at 20m/s.....	127
Figure 8.24: Conventional Flight Controllers Gain Scheduled Responses .....	128
Figure 8.25: Landing Sequence Phases (Repeat) .....	128
Figure 8.26: Aircraft on an Autonomous Landing Approach.....	129
Figure 8.27: Longitudinal Control during an Autonomous Landing.....	129
Figure 8.28: Glide Path Tracking and Flare Response during an Autonomous Landing.....	130
Figure 8.29: Aircraft just before Autonomous Touchdown .....	131
Figure 8.30: Lateral Control during an Autonomous Landing.....	131
Figure 8.31: Aircraft after a Successful Autonomous Landing.....	132
Figure A.1: Main Microcontroller Block Diagram .....	140
Figure A.2: Inertial Measurement Unit Block Diagram [1] .....	141
Figure A.3: Air Data Block Diagram [1] .....	142
Figure A.4: Circuit Diagram for Differential Pressure Signal Conditioning .....	145
Figure B.1: Yaw Orientation Controller Block Diagram .....	146
Figure B.2: Altitude Controller Block Diagram.....	146
Figure B.3: Airspeed and Climb Rate Controller Block Diagram .....	147
Figure B.4: Guidance Controller Block Diagram .....	148
Figure B.5: $X_p$ -Acceleration Controller Frequency Response .....	149
Figure B.6: Runway Yaw Rate Controller Frequency Response.....	150

LIST OF FIGURES

---

Figure C.1: <i>Coefficients of Thrust and Power vs. Advance Ratio</i> .....	153
Figure C.2: <i>Propeller Inertia Geometry</i> .....	154
Figure D.1: <i>Wind and Body Axes</i> .....	155
Figure E.1: <i>Geometry for Initial CG and Pitch Angle Calculation for the Runway Model</i> .....	158
Figure F.1: <i>Inertial Axes Definition [15]</i> .....	160
Figure F.2: <i>Body Axes Definition [15]</i> .....	161
Figure G.1: <i>ESL Micro Avionics</i> .....	166
Figure G.2: <i>Reliance 0.46 Size Model Aircraft</i> .....	167

# List of Tables

Table 6.1: <i>Avionics Package Specifications</i> .....	95
Table 7.1: <i>Wind Estimation Results</i> .....	111
Table A.1: <i>Autopilot Control Byte 0</i> .....	143
Table A.2: <i>Autopilot Control Byte 1</i> .....	143
Table A.3: <i>Autopilot Control Byte 2</i> .....	144
Table A.4: <i>Autopilot Control Byte 3</i> .....	144
Table A.5: <i>Sequence Control Byte 0</i> .....	144
Table A.6: <i>Sequence Control Byte 1</i> .....	145
Table B.1: <i>Altitude, Turn Rate and Guidance Controller Gains</i> .....	147
Table B.2: <i>Gain Scheduled Airspeed and Climb Rate Controller Gains</i> .....	148
Table B.3: <i>Original and High Gain Guidance Controller Gains</i> .....	148
Table B.4: <i>Runway Yaw Rate Proportional and Error Integral Gains</i> .....	149
Table B.5: <i>Runway Guidance Controller Gains</i> .....	150
Table B.6: <i>Landing Path Parameters</i> .....	151
Table B.7: <i>Approach Point Parameters</i> .....	151
Table C.1: <i>Engine Power Lookup Table</i> .....	152
Table C.2: <i>Coefficients of Thrust and Power vs. Advance Ratio</i> .....	153
Table C.3: <i>Sensor Noise Variances</i> .....	154
Table D.1: <i>Aircraft Parameters Used During Stall Speed Calculations</i> .....	156
Table G.1: <i>Data Required for Ground Controllers</i> .....	169
Table G.2: <i>Data Required for Piloted Flight with Telemetry Logging and Runway Flybys</i> .....	171
Table G.3: <i>Data Required for Longitudinal Controllers</i> .....	173
Table G.4: <i>Data Required for Lateral Controllers</i> .....	176
Table G.5: <i>Data Required for Autonomous Navigation and Wind Estimation</i> .....	178
Table G.6: <i>Test Schedule</i> .....	179
Table G.7: <i>Test Hazard Analysis Worksheet</i> .....	180

# Nomenclature

## Greek Letters

$\alpha$	Angle of attack, Slip angle or Angular acceleration
$\beta$	Angle of sideslip
$\varepsilon_T$	Thrust vector setting angle
$\delta_a$	Aileron perturbation from trim condition
$\delta_A$	Aileron total deflection
$\delta_e$	Elevator perturbation from trim condition
$\delta_E$	Elevator total deflection
$\delta_r$	Rudder perturbation from trim condition
$\delta_R$	Rudder total deflection
$\delta_s$	Steering wheel deflection angle
$\delta_i$	Throttle perturbation from trim condition
$\delta_T$	Throttle total deflection
$\delta_{i_c}$	Commanded throttle position
$\varepsilon_T$	Thrust vector setting angle
$\gamma$	Glide path angle
$\phi$	Roll angle perturbation
$\theta$	Pitch angle perturbation
$\rho$	Air density
$\tau$	Exponential time constant
$\tau_T$	Engine time constant
$\mu_f$	Rolling resistance coefficient
$\psi$	Yaw angle perturbation
$\Phi$	Roll angle
$\Theta$	Pitch angle
$\Psi$	Yaw angle
$\Omega$	Rotation speed

## Small Letters

$a$	Acceleration
$b$	Wing span
$\bar{c}$	Mean aerodynamic chord
$g$	Gravitational acceleration
$h$	Height (altitude)
$k_d$	Damper proportionality constant

## NOMENCLATURE

$k_s$	Spring proportionality constant
$l_m$	Main wheels moment arm in reference line body axis
$l'_m$	Main wheels moment arm in stability axis
$l_r$	Runway to centre of gravity length
$l_s$	Steering wheel moment arm in reference line body axis
$l'_s$	Steering wheel moment arm in stability axis
$l_{sm}$	Steering to main wheel moment arm in reference line body axis
$l'_{sm}$	Steering to main wheel moment arm in stability axis
$m$	Mass
$m_T$	Thrust vector moment arm
$p$	Roll rate perturbation or Static pressure
$q$	Pitch rate perturbation or Dynamic pressure
$r$	Yaw rate perturbation
$s$	Deformation or Displacement
$u$	Axial velocity perturbation
$v$	Lateral velocity perturbation or Compression rate
$w$	Normal velocity perturbation
$y$	Cross track error

## Capital Letters

$C_\alpha$	Cornering stiffness
$C_{\alpha\alpha}$	Cornering coefficient
$C_D$	Aerodynamic drag coefficient
$C_l$	Aerodynamic roll moment coefficient
$C_L$	Aerodynamic lift coefficient
$C_m$	Aerodynamic pitching moment coefficient
$C_n$	Aerodynamic yaw moment coefficient
$C_P$	Propeller coefficient of power
$C_T$	Propeller coefficient of thrust
$C_x$	Axial aerodynamic force coefficient
$C_y$	Lateral aerodynamic force coefficient
$C_z$	Normal aerodynamic force coefficient
$D$	Down displacement or spring deformation
$E$	East displacement
$F$	Force
$I_x$	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
$J$	Advance ratio
$L$	Roll moment
$M$	Pitch moment or Moment
$N$	Yaw moment, North displacement or Normal Force
$P$	Roll rate

## NOMENCLATURE

$P_D$	Position in the down direction
$P_E$	Position in the east direction
$P_N$	Position in the north direction
$P_T$	Engine power
$Q$	Pitch rate
$R$	Yaw rate, Range or Radius
$R_{GP}$	Glide path start range
$R_{flare}$	Flare range
$S$	Wing area or Steering wheel suspension point
$T$	Engine thrust magnitude or Torque
$U$	Axial velocity
$V$	Lateral velocity
$\bar{V}$	Airspeed
$W$	Normal velocity or Wind velocity component
$X$	Axial force or $X$ -axis
$Y$	Lateral force or $Y$ -axis
$Z$	Normal force or $Z$ -axis
$LM$	Left main wheel suspension point
$RM$	Right main wheel suspension point

## Subscripts

0	Trim or Initial condition
$app$	Approach point variable
$A$	Aerodynamic force or Moment
$B$	Body axis
$B_R$	Runway body axis
$c$	Cornering
$cg$	Around centre of gravity
$d$	Damper
$E$	Inertial axis, East component or Engine
$f$	Friction or Finish
$G$	Gravitational force or Moment
$lm$	Left main wheel
$LM$	Left main wheel suspension point
$m$	Main wheels
$N$	North component
$p$	Propeller
$R$	Runway
$rm$	Right main wheel
$RM$	Right main wheel suspension point
$s$	Steering wheel, Spring or Start
$T$	Engine force and moment as well as tail plane
$W$	Wind
$x$	Along the $X$ -axis
$y$	Along the $Y$ -axis
$z$	Along the $Z$ -axis

## Acronyms

3D	Three Dimensional
6DoF	Six Degrees of Freedom
ADC	Analog to Digital Converter
ASCR	Airspeed and Climb Rate Controller
CAN	Controller Area Network
CG	Centre of Gravity
CPU	Central Processing Unit
DCM	Direction Cosine Matrix
DC	Direct Current
DGPS	Differential Global Positioning System
DLD	Desired Landing Distance
EKF	Extended Kalman Filter
EMC	Electro Magnetic Coupling
ESL	Electronic Systems Laboratory
FC	Flight Control
FM	Frequency Modulation
GPS	Global Positioning System
GUI	Graphical User Interface
HIL	Hardware In the Loop
IMU	Inertial Measurement Unit
LAN	Local Area Network
LQR	Linear Quadratic Regulator
LPF	Low Pass Filter
MAP	Manifold Pressure
MC	Mission Control
MIMO	Multi-Input-Multi-Output
MISO	Master In Slave Out or Multi-Input-Single-Output
MOSI	Master Out Slave In
NiCad	Nickel Cadmium
PCM	Pulse Code Modulation
PI	Proportional Integral
PPM	Pulse Position Modulation
PWM	Pulse Width Modulation
RC	Radio Controlled RF Radio Frequency
RMS	Root Mean Square
RPM	Revolution Per Minute
RF	Radio Frequency
RSSI	Received Signal Strength Indicator
SHI	Simulation Hardware Interface
SISO	Single-Input-Single-Output
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver and Transmitter
UAV	Unmanned Aerial Vehicle
UDP	User Data Protocol
YOCS	Yaw Orientation Control System
ZOH	Zero Order Hold



# Chapter 1

## *Introduction and Overview*

### 1.1 Background and History

The University of Stellenbosch formed an Unmanned Aerial Vehicle (UAV) research group within the Electronic Systems Laboratory (ESL) in 2001. The group quickly produced several successful projects and has established itself as one of the top UAV research institutions in South Africa.

Successful projects include:

- Autonomous control of an electrical RC helicopter [17].
- Autonomous flight of a model aircraft [1].
- Development of a rotary-wing test bed for autonomous flight [18].
- Development of an experimental tilt-wing UAV.

Projects under development include:

- Autonomous acrobatic flight of a UAV [22].
- Autonomous take-off and landing of a rotary-wing UAV.
- Autonomous flight of a tilt wing UAV.

This project follows directly on the autonomous navigation autopilot developed in [1], with the goal to extend the current autopilot to include autonomous take-off and landing. This forced many decisions and restricted the project in many ways, but also provided a strong base from which to build.



**Figure 1.1:** *ESL UAV Fleet*

Some of the restrictions and decisions forced include:

- Aircraft selection – The aircraft used in [1] and in this project is shown in Figure 1.2. The small radio controlled aircraft is very susceptible to wind disturbances. This complicates take-off and landing. However, the aircraft is a trainer with a low stall speed and relatively stable natural dynamics, which is desirable for both take-off and landing.
- Sensor selection – Only low cost, off-the-shelf gyroscopes and accelerometers with a medium accuracy GPS (4m, 95% of the time [17]) were used in [1]. The gyroscope biases and GPS accuracy translates into a wide runway required for take-off and landing. The system did allow for additional, high accuracy sensors to be integrated, for example an ultrasonic range sensor that can provide an altitude measurement with 1cm accuracy to a maximum altitude of 5m.
- Microcontroller selection – The avionics developed in [1] made use of three 8-bit, non floating point microcontrollers. The microcontroller selection results in a low cost and light weight system, but closes many doors to algorithm development and thus limits the usable control strategies.
- Control strategy – The controllers in [1] were designed by a classical successive loop closure approach. This approach was adopted for the following reasons:
  1. It provides a good intuitive approach to controller design.
  2. The method typically leads to simple, easily tuneable controllers with good robustness properties.
  3. The resulting controllers are easily implemented on 8-bit microcontrollers.
  4. The strategy works well with low cost sensors.
  5. The strategy allows for only the inner attitude regulating loops to be gain scheduled.
  6. The limited accuracy aircraft model, low cost sensors and limited processing power makes the use of a full state optimal estimator undesirable. A Linear Quadratic Regulator or other full state feedback control strategies can therefore not be implemented.

For the reasons stated above, the successive loop closure design strategy is continued in this project.



But extending an existing autopilot has many advantages. Some of these advantages are:

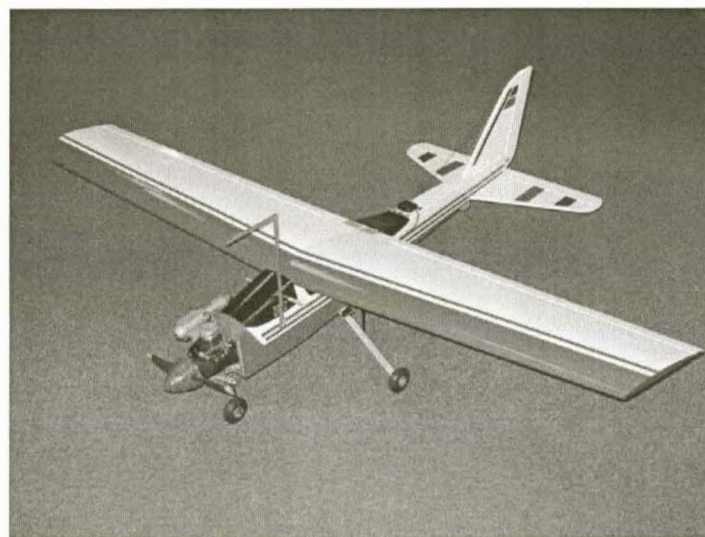
- Flight proven navigator and conventional flight controllers – The flight proven navigator and conventional flight controllers not only provide confidence in the control strategy, but will also greatly aid the landing process.
- Readily available aircraft model – The aircraft model can directly be used for the design of any airborne controllers. The model also provides a good starting point for the development of a runway model required for take-off and landing control and simulation.
- First iteration avionics – The avionics designed in [1] can be improved and extended, resulting in a second generation of hardware that was thoroughly tested and proved very reliable.
- Initial simulation environment – The non-linear simulation tools developed can be extended and improved to create a hardware in the loop (HIL) simulator.

## 1.2 Project Starting Platform

This section presents a very brief summary of the work done in [1] as it is the foundation on which this project is build. Appendix F contains additional detail on the work done in [1].

### 1.2.1 Aircraft and Model

The aircraft used in [1] and in this project is the Reliance 0.46 size model aircraft shown in Figure 1.2.



**Figure 1.2:** *Reliance 0.46 Size RC Trainer Aircraft [1]*

Three main sources contribute to the forces and moments acting on the aircraft. These three sources are the airfoils (aerodynamic forces and moments), the engine and gravity. The model for the aerodynamic forces and moments were developed using empirical data and equations, as well as the geometry of the aircraft.

The following assumptions were made about the aircraft airfoils:

- Only the wing and the tail plane contribute to the aircraft lift.
- Only the wing and the fuselage contribute to the aircraft drag.
- Only the lift force of the fin contributes to the aircraft side force.
- Only the wing and the fin contributes to the aircraft roll and yaw moments.

The engine forces and moments are modelled as being produced by a single, centrally positioned, rigid propeller engine. The propeller angular momentum is ignored, while gravitational force acting on the aircraft is modelled as a constant force per unit mass.

The forces and moments acting on the aircraft from all three sources are summarised in equations (1.1) to (1.6) [25]. All the forces and moments are referenced in the stability (wind) axis system. The parameters in parentheses indicate that the respective variable is a function (possibly non-linear) of the listed parameters.

$$X = X_A + X_T + X_G = qSC_x(\bar{V}, \alpha, \dot{\alpha}, \alpha_0, Q, \delta_E) + T(\bar{V}, \varepsilon_T, \alpha_0, \delta_T) - mg \sin \Theta \quad (1.1)$$

$$Y = Y_A + Y_G = qSC_y(\bar{V}, \beta, R, \delta_R) + mg \cos \Theta \sin \Phi \quad (1.2)$$

$$Z = Z_A + Z_T + Z_G = qSC_z(\bar{V}, \alpha, \dot{\alpha}, \alpha_0, Q, \delta_E) + T(\bar{V}, \varepsilon_T, \alpha_0, \delta_T) - mg \cos \Theta \cos \Phi \quad (1.3)$$

$$L = L_A = qSbC_l(\bar{V}, \alpha, \alpha_0, \beta, P, R, \delta_A, \delta_R) \quad (1.4)$$

$$M = M_A + M_T = qS\bar{c}C_m(\bar{V}, \alpha, \dot{\alpha}, \alpha_0, Q, \delta_E) + T(\bar{V}, m_T, \alpha_0, \delta_T) \quad (1.5)$$

$$N = N_A = qSbC_n(\bar{V}, \alpha, \alpha_0, \beta, P, R, \delta_A, \delta_R) \quad (1.6)$$

In the above equations  $\varepsilon_T$  is the thrust vector setting angle and  $m_T$  is the thrust vector moment arm as shown in Figure 1.3.

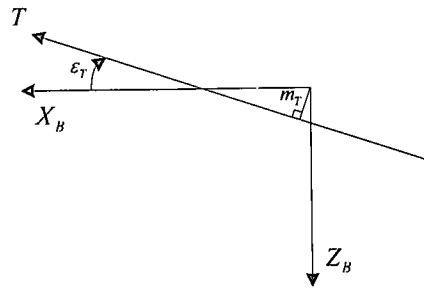


Figure 1.3: Engine Thrust Geometry [1]

The aerodynamic and engine forces and moments are linearised by expressing them as the sum of a truncated Taylor series, in the aircraft motion and control variables, about the trim condition. The linearisation is needed so that linear system analysis theory can be applied to develop initial control laws for the system. The linearisation is done by using small disturbance theory which has been found to give good practical results. All the stability and control derivatives are calculated from first principles where the aircraft geometry and approximate airfoil aerodynamic models are used. The linear aircraft model is given in section F.3.2.

The linear engine is modelled as a constant power source at a specific throttle setting with a first order thrust actuation lag. The constant power formula and first order differential equation is given in equation (1.7) and (1.8) respectively. The relationship between  $\delta_t$  and thrust is discussed in section 3.2.1.

$$T = \frac{P_T}{\bar{V}} \quad (1.7)$$

$$\dot{\delta}_t = -\frac{1}{\tau_T} \delta_t + \frac{1}{\tau_T} \delta_{t_c} \quad (1.8)$$

### 1.2.2 Controllers

All of the controllers are designed as linear controllers around a straight and level trim condition. The autopilot developed is capable of autonomous waypoint navigation within a conventional flight envelope. The controller design is broken into three sections:

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

An overview of the control system is shown in block diagram format in Figure 1.4 and the controllers developed are summarised below.

- Longitudinal:
  - Stability Augmentation: Pitch Rate Damper
  - Attitude Regulation: Airspeed and 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

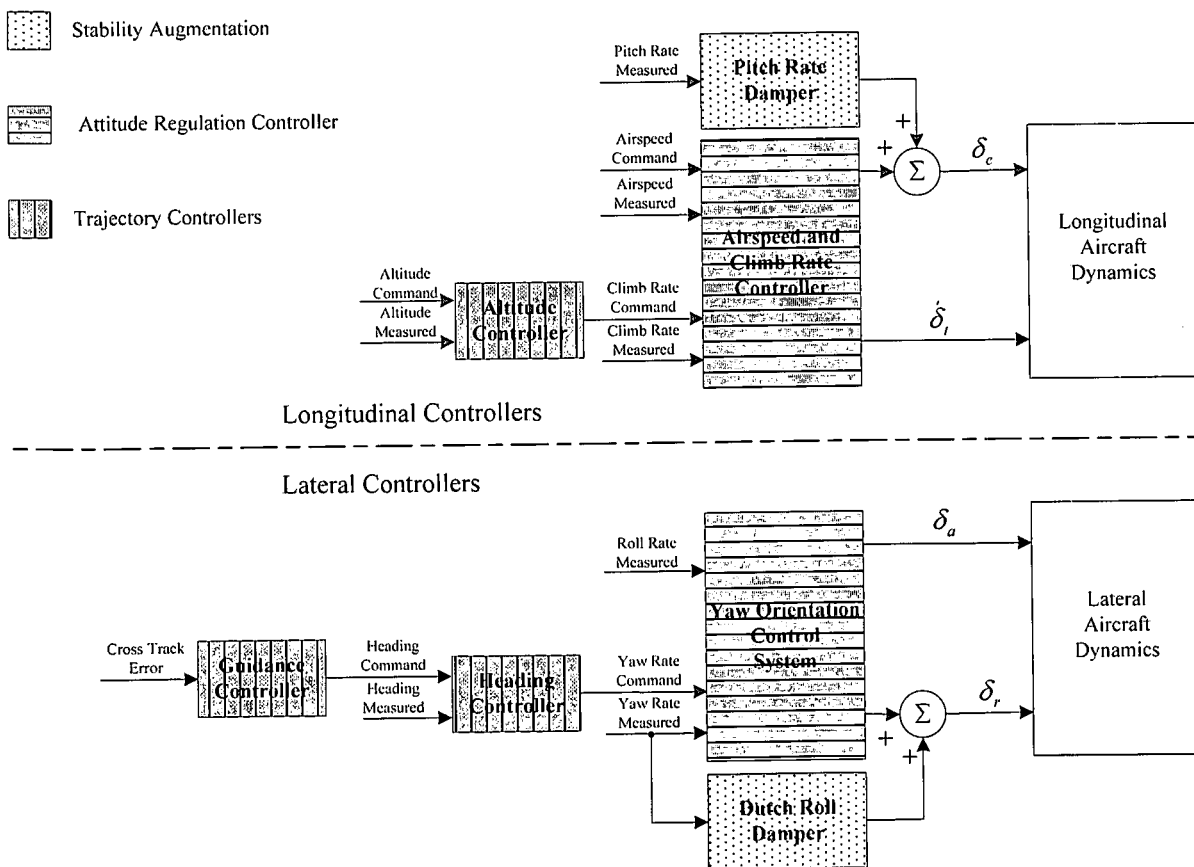


Figure 1.4: Original Autopilot Overview

### 1.2.3 Navigation Algorithm

A navigation algorithm capable of generating the cross track error required by the guidance controller was 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. A waypoint is defined by the following:

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

The path planner algorithm accepts two waypoints and determines a valid flight path between them. “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, which leads from one waypoint to the next, such that the starting and ending course tangents are coincidental with the respective heading directions”. The typical path outputs that the algorithm generates, are shown in Figure 1.5.

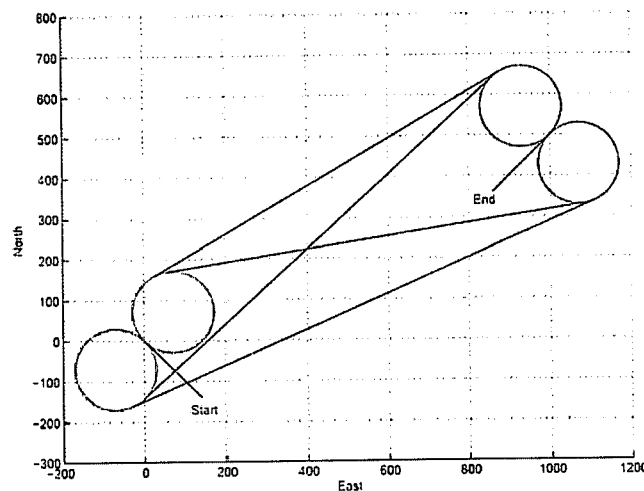


Figure 1.5: Path Planner Graphical Output [1]

### 1.2.4 Avionics and Ground Station

Figure 1.6 shows the avionics package developed for the practical implementation of the controllers and navigation algorithm discussed in the previous sections. Figure 1.7 shows a block diagram representation of the avionics, as well as the ground station.



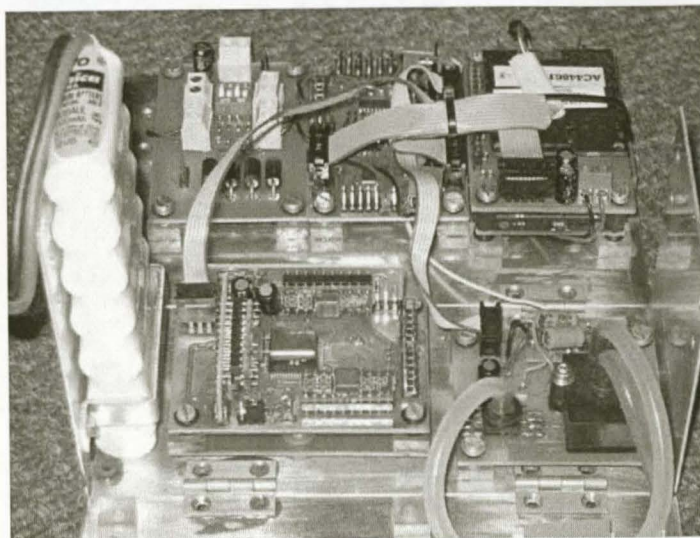


Figure 1.6: Original Avionics [1]

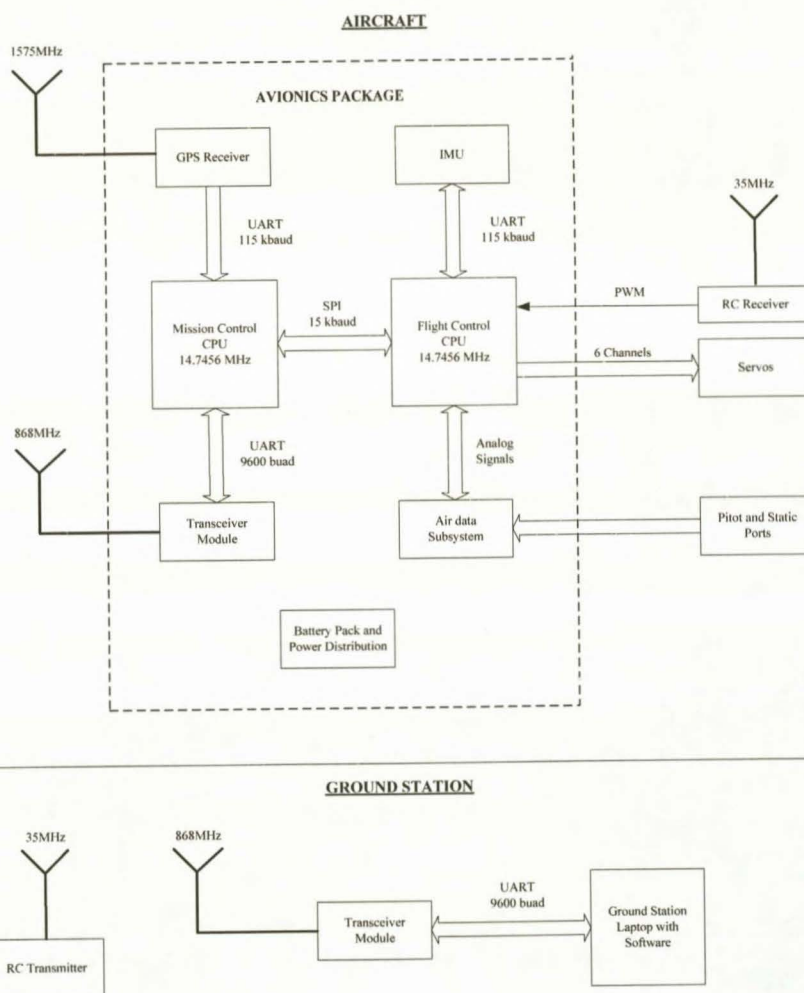


Figure 1.7: Original Avionics Block Diagram

The avionics package consisted of the following microcontrollers, sensors and OEM modules,

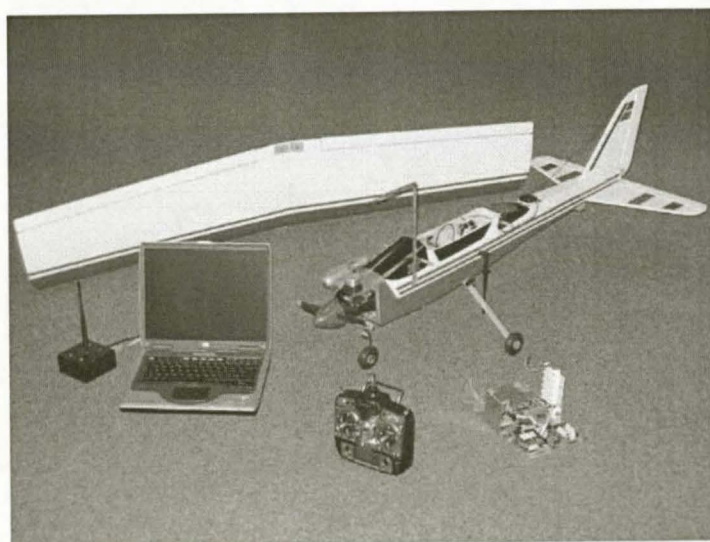
- 3 x ATmega 128 Microprocessors
  - Flight Control, Mission Control and IMU processor.
- 3 x Rate Gyroscopes
  - $\pm 60$  deg/s with 10Hz bandwidth and 0.1 deg/s digital resolution.

- 2 x Dual-Axes Accelerometers
  - $\pm 2g$  with 10 Hz bandwidth and 0.01g digital resolution.
- 1 x Differential Pressure Sensor
  - Measurable airspeed up to 45m/s with 0.07m/s digital resolution at 20m/s.
- 1 x Absolute Pressure Sensor
  - Rough altitude of 0 to 3000m range with 5Hz bandwidth and 2.8m digital resolution at sea level.
  - Local fine altitude of 0 to 300m range with 5Hz bandwidth and 0.28m digital resolution at sea level.
  - Climb rate signal with  $\pm 5m/s$  range with a 2Hz bandwidth and 0.01m/s digital resolution.
- Aerocomm OEM RF Transceiver
- U-Blox GPS Receiver
  - 4m CEP position accuracy 95% of the time at 4Hz update rate and a single cycle delay (0.25s).
  - 0.4 m/s CEP velocity accuracy at 4Hz update rate and a single cycle delay (0.25s).
- 8.4V, 2100mAh NiCad Battery Pack

The ground station consists of a RC transmitter and a laptop computer running the Graphical User Interface (GUI). The GUI allows for the following:

- Autopilot variables can be updated
- Waypoints can be manipulated and managed
- Sensor interaction
- Servo calibration and actuator monitoring, and
- Logging of telemetry and history information.

The complete system is shown in Figure 1.8.



**Figure 1.8:** Complete System from [1]



## 1.3 Project Goals

The *primary project goal* is to extend the autonomous navigating autopilot developed in [1] to include autonomous take-off and landing.

This goal is pursued by setting several *secondary project goals*:

- Re-package, re-design and extend the original avionics to a professional, small and easily reproducible product with hardware in the loop (HIL) simulation capability.
- Remove the single trim condition constraint to extend the flight envelope over a wider airspeed range as required for autonomous take-off and especially landing.
- Subject the system to more flight hours and testing to obtain confidence in the hardware and software, as well as a better understanding of the actual aircraft response and sensor performance.
- Complete the aircraft model by extending it to incorporate aircraft dynamics on the runway and a complete engine model.
- Combine the block diagram and graphical simulator developed in [1] and include HIL capability to form an excellent simulation and development environment.
- Design and implement the necessary controllers to perform autonomous take-off and landing.
- Practically demonstrate autonomous take-off and landing.

## 1.4 Achievements

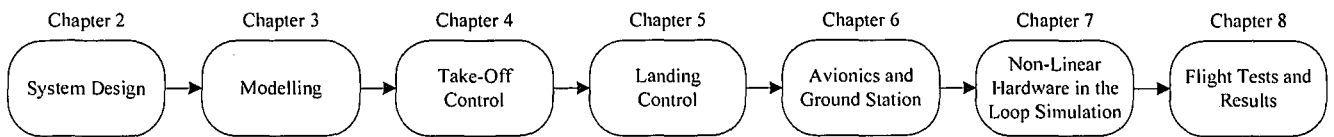
This project saw the successful autonomous taxiing, take-off and landing of the Reliance 0.46 size model aircraft.

The *secondary achievements* and output from this project included:

- The original avionics were redesigned and thoroughly tested in the laboratory and through practical flight tests. An ultrasonic range sensor and magnetometer were added to complete the sensor array.
- The original autopilot was extended and the flight envelope increased by removing the single trim condition constraint. The extended autopilot was verified through practical flight tests.
- Non-linear models were developed for the aircraft on the runway, as well as for the engine, to form a complete aircraft model when combined with the aerodynamic model developed in [1].
- The block diagram and graphical simulators developed in [1] were combined and the new simulator was extended with the non-linear models developed. The simulation was also modified to include the HIL capabilities as developed in [22].
- Controllers required for autonomous take-off and landing were designed, implemented on the avionics and verified by the non-linear hardware in the loop simulator.
- Practical results were also presented for the individual controllers. These practical results not only verified the controller designs but also the non-linear models created.

## 1.5 Thesis Outline

The chapter flow is illustrated by Figure 1.9.



**Figure 1.9:** *Thesis Chapter Flow*

The thesis starts in **CHAPTER 2** with a high level discussion on the design of the system. The chapter discusses the take-off and landing sequences implemented, as well as the controllers that make up these sequences. The calculation of the cross track error off the runway, the landing path and the wind estimation procedure is also discussed.

With an overview of the take-off and landing established, the focus shifts to the detail required for the design and implementation of the controllers. In **CHAPTER 3** the development of the non-linear runway, the engine and the wind gust models are discussed. The runway model is also linearised for use in Chapter 4.

The linear models developed in Chapter 3 are then used with the linear aircraft model developed in [1] to design the take-off controllers in **CHAPTER 4** and the landing controllers in **CHAPTER 5**. The linear simulation results are also presented in these two chapters.

**CHAPTER 6** completes the picture by the redesign and extension of the avionics and ground station to incorporate the new system functionality, sensors and hardware in the loop capability.

With the controllers and avionics designed, the stage is set for hardware in the loop simulation in **CHAPTER 7**, where the system developed is subjected to extensive non-linear simulation and evaluation.

Hereafter, the simulated system is subjected to practical testing. The results are presented in **CHAPTER 8**.

# Chapter 2

## *System Design*

This chapter discusses the high level design of the system. The discussion starts with a short general discussion on take-off in section 2.1. The take-off sequence implemented is then presented, defining the different control phases and the variables to be controlled in order to achieve autonomous take-off. With the take-off foundation established, the controllers required for the take-off sequence are presented. The reasons for the specific controller selection and controller function are discussed. Section 2.2 focuses on landing. It starts with a general discussion on landing, followed by a look at the sequence and controllers used during the landing. The wind estimation algorithm developed to ensure a head wind landing is also presented, followed by a discussion on the landing path.

The algorithm used to calculate the cross track error is discussed in section 2.3. This algorithm is required by both the take-off and the landing sequence.

Finally, all of the controllers are combined in section 2.4. This section contains a block diagram of the controllers and required sensors, together with a discussion on the implementation methods and design software used.

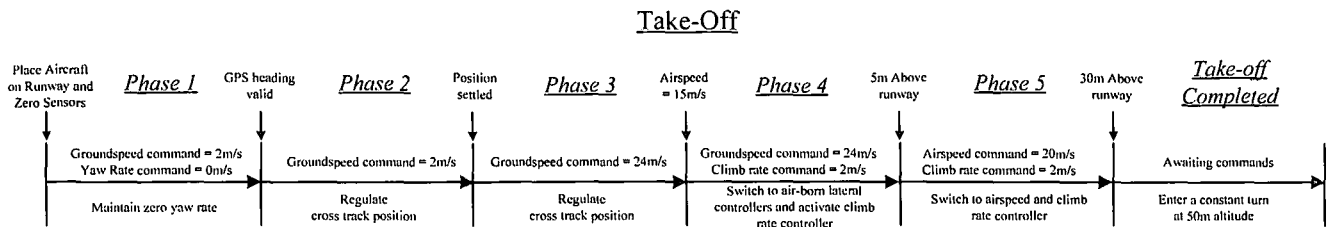
### 2.1 Take-Off

Take-off can essentially be defined as high speed taxiing down the runway (the ground roll) until the predefined take-off airspeed ( $2x$  stall speed) is reached. The take-off airspeed is the airspeed at which the wing generates enough lift to ensure a safe take-off in spite of wind disturbances. Once the take-off airspeed is reached, the aircraft is rotated and entered into a steady constant climb (the climb out) to gain altitude as quickly as possible without inducing a departure stall. It is necessary to control the forward velocity and lateral motion on the runway for the ground roll. Once the aircraft reached take-off airspeed climb rate can be controlled to initiate rotation and regulate the climb out. At the point of rotation lateral control should be shifted from the nose wheel and rudder to the ailerons, while still regulating the forward velocity (airspeed).

A discussion of the autonomous take-off sequence implemented in this project is presented in the following section. The take-off sequence serves as the framework and outline for the autonomous take-off.

Once the take-off sequence is defined, the focus shifts to a discussion on the individual controllers and the sensors required for realising the take-off sequence in section 2.1.2.

### 2.1.1 Take-Off Sequence



**Figure 2.1:** Take-Off Sequence Phases

The five phases that form the take-off sequence are shown in Figure 2.1. The take-off sequence starts by placing the aircraft on the runway and initialising all the sensors (can be automated). When the take-off command is issued, the system enters *Phase 1* where the groundspeed and yaw rate is controlled with a 2m/s groundspeed and 0 rad/s yaw rate command. *Phase 1* allows the aircraft to move so that the GPS heading becomes valid. A valid GPS heading is signalled when eight consecutive measurements all fall within a 50° cone.

Once a valid GPS heading is available, the system enters *Phase 2*. The cross track error is controlled to guide the aircraft down the runway while maintaining 2m/s groundspeed.

When the cross track error has settled, the system enters *Phase 3* which is the ground roll. The groundspeed command is stepped up to 24m/s while the lateral controllers continue to guide the aircraft down the runway.

The system enters *Phase 4* when the take-off airspeed of 15m/s ( $2 \times \bar{V}_{stall}^3$ ) is reached. In *Phase 4* the climb rate is controlled with a 2m/s command to initiate rotation and climb out. Lateral control is switched over from the runway controllers to the air-born controllers to keep on guiding the aircraft down the runway as it climbs. Should the aircraft naturally lift-off before the predefined take-off airspeed is reached the event will be detected by an ultrasonic range sensor onboard. The system will then also enter *Phase 4* in order to control the climb out.

Five meters above the runway, the system switches to *Phase 5*. The climb rate and groundspeed controllers are disabled and the airspeed and climb rate controller as designed in [1] are armed with an airspeed command of 20m/s and a climb rate command of 2m/s. The lateral controllers still guide the aircraft along the runway. Take-off is considered complete once the aircraft reaches an altitude of thirty meters. The altitude is then increased to 50m with the aircraft entering a constant turn, awaiting commands.

<sup>3</sup>  $\bar{V}_{stall}$  is calculated in Appendix D.

## 2.1.2 Take-Off Controllers

The controllers designed to autonomously perform the take-off sequence discussed in section 2.1.1 are,

- Longitudinal:
  - Pitch Rate Damper
  - $X_B$ -Acceleration and Groundspeed Controller
  - Climb Rate Controller
- Lateral:
  - Runway Yaw Rate Controller
  - Runway Heading Controller
  - Runway Guidance Controller
  - Turn Radius Interface (not part of the take-off sequence)

The main disturbances acting on the aircraft during the take-off process (other than sensor noise, drift and biases) that must be rejected by the controllers and kept in mind during controller designing are:

1. A constant crosswind and wind gusts that pushes the aircraft off the desired track.
2. Wind gusts that can induce a premature take-off or stall the aircraft shortly after take-off.
3. Ground effect phenomena caused by the interaction of the aircraft wingtip vortices with the ground. This interaction reduces the strength of these vortices. The weakened wingtip vortices reduce the wing downwash which increases the lift and decreases the induced drag [36]. The disturbance takes effect when the wing is within one wingspan length of the ground [16].
4. Propeller torque and helical slipstream increase the rolling friction on one of the main wheels and cause a side force on the rudder [16].

These effects will directly influence the selected controller design specifications. The disturbances, with the steps taken to reject them are further discussed in Chapter 4, as well as in the controller discussions that follows.

### Pitch Rate Damper

The pitch rate damper counters sudden pitch rate perturbations during the ground roll and rotation that can induce a premature take-off or stall the aircraft. It also serves as a short period damper and provides damping with respect to inertial space for wind gust rejection. The pitch rate damper provides the underlying stability around which the climb rate controller can be designed.

### Acceleration and Groundspeed Controllers

Two options exist for controlling the velocity of the aircraft on the ground: Airspeed or Groundspeed control. Airspeed is the more functional variable to control since it determines when the aircraft has enough lift for rotation and climb out. The digital resolution of the airspeed measurement obtained from a Pitot - static tube system and the differential pressure sensor is dependant on the airspeed and is given by equation (2.1) [1].

$$\text{Digital Resolution} = \left( \frac{\rho \bar{V} \cdot 5 \cdot 2^{10}}{1.25 \times 10^3 \cdot 5} \right)^{-1} \quad (2.1)$$

The digital resolution of the airspeed measurement at 1m/s is 1.25m/s. This makes the airspeed uncontrollable at low airspeeds (less than 2m/s). The airspeed measurement is also easily corrupted by wind gusts that can trigger a premature take-off.

The alternative, groundspeed control, is more attractive since the groundspeed measurement from the u-Blox GPS receiver is accurate to within 0.4m/s at any velocity and is not directly influenced by wind gusts. The groundspeed is controlled by an  $X_b$ -acceleration controller with an outer groundspeed loop. The acceleration controller is included for two reasons: The groundspeed measurement from the u-Blox GPS receiver is only updated at 4Hz, with a single cycle delay of 0.25 seconds [17] and the groundspeed is one natural integrator away from the acceleration that will be induced by a change in the throttle actuator. The  $X_b$ -acceleration can be sensed by a low cost accelerometer. Although the groundspeed is controlled during the ground roll, the rotation and climb out will still be initiated based on the airspeed measurement to prevent departure stalls.

Only acceleration control is sufficient to control the longitudinal motion during take-off. A groundspeed controller is designed, since it allows for autonomous taxiing. Groundspeed is also the more intuitive variable for a human operator to command.

### **Climb Rate Controller**

The climb rate controller is used to rotate the aircraft and perform the climb out when the take-off airspeed is reached. It is also used to flare the aircraft during the final stages of landing. The throttle has a major influence on the climb rate of the aircraft, but is not included in the design of the controller since the controller will only be operating when the throttle is either fully closed (flaring) or controlled by the groundspeed controller (rotation and climb out). The climb rate controller is therefore designed to provide control over the aircrafts climb rate with only the elevator.

The climb out rate during take-off is limited to 2m/s to ensure that the linear aircraft model remains valid.

### **Runway Yaw Rate Controller**

Control of the lateral motion of the aircraft on the runway is achieved by controlling the yaw rate of the aircraft. The yaw rate controller provides a good base for the design of a turn radius, heading and guidance controllers. The yaw rate can also easily be measured by a low cost gyroscope. On the runway, the yaw rate is controlled by the nose wheel and the rudder with the dominant actuator depending on the airspeed. The effectiveness of the nose wheel will decrease, while the effectiveness of the rudder will increase with airspeed. The lateral control of the aircraft throughout the ground roll is therefore non-linear with airspeed. One of the advantages of the successive loop closure technique is that only the runway yaw rate controller needs to be designed to incorporate the airspeed non-linearity. All successive loop closures can be done independently of airspeed since the non-linearity and aircraft dynamics are already encapsulated by the runway yaw rate controller.

Gain scheduling was chosen as the method to remove the airspeed non-linearity from the plant model. Feedback linearisation is preferred to gain scheduling, but implementation restrictions on the 8-bit non-floating point microcontrollers prevent the use of feedback linearisation. The disadvantage of gain scheduling is that the method requires redesigning the controller several times and does not allow for easy gain changes during practical testing.

### **Runway Heading and Guidance Controllers**

The heading and guidance controllers form two outer loops around the runway yaw rate controller and regulate the cross track error during the ground roll. Heading measurements from the u-Blox GPS receiver is fed back to form a heading error that becomes a turn rate command to the runway yaw rate controller. Position measurements from the u-Blox GPS receiver is used to calculate the cross track error that becomes a heading command for the heading controller (details on cross track error calculations are presented in section 2.3). The successive loop closure technique minimises the amount of dynamic compensation required in implementing the two lateral trajectory controllers [1]. The position accuracy of the u-Blox GPS receiver is 4m, 95% of the time [17]. This translates to a minimum runway width of 10m from which take-off can be guaranteed.

### **Radius Interface**

The turn radius interface is designed to remove the velocity dependency when issuing turning commands. This provides a more intuitive interface for a human operator to steer the aircraft. The turn radius interface is not used during the take-off sequence, but together with the runway heading and guidance controllers provide a platform to implement autonomous ground navigation algorithms in future projects. The path to be followed can be described in terms of the known turn radii of the runway.

## **2.2 Landing**

Landing consists of approaching the runway on a glide path, at the predefined glide slope. When the desired altitude and range from the runway is reached, the aircraft is transitioned from the glide path to the flare path. Flaring consists of flying an exponential path from the glide path to the runway so the aircraft can touch down gently. Tight lateral track regulation should be maintained throughout the landing approach to keep the aircraft in line with the runway. In the event of a cross-wind landing, it will be necessary to de-crab the aircraft before touchdown.

Landing is a difficult and dangerous manoeuvre and requires high precision longitudinal and lateral control as well as good timing from a human pilot. This is no different from when an autonomous landing is to be performed. Precision and high bandwidth lateral and longitudinal controllers are needed to perform an autonomous landing. Precision and high bandwidth translates into fast controllers requiring large feedback gains. High bandwidth is difficult to obtain with classical successive loop closures, especially in the outer loops. The full state vector of the aircraft is not available for feedback, limiting the obtainable dynamic response from the controllers. Multi-Input-Multi-Output (MIMO) controllers are often also required but are difficult to design with classical methods. One difficulty for example, is controlling both the climb rate and pitch angle of the aircraft with the throttle and elevator during landing. High demands are also placed on the accuracy, reliability and noise characteristics of the sensors when used with high bandwidth controllers. Qualities typically not obtained from the low cost off-the-shelf sensors used in this project.



It is clear from this discussion that autonomous landing is a difficult task. To circumvent system restrictions the available controllers and hardware are optimised and the landing procedure is kept as simple as possible.

A discussion of the autonomous landing sequence implemented in this project is presented in the following section. The landing sequence serves as the framework and outline for the autonomous landing. Once the landing sequence is defined, the focus shifts to a discussion on the individual controllers and the sensors required for realising the landing sequence in section 2.2.2. The wind estimation procedure necessary for a head wind landing is discussed in section 2.2.3. The landing path is then defined and discussed in section 2.2.4.

## 2.2.1 Landing Sequence

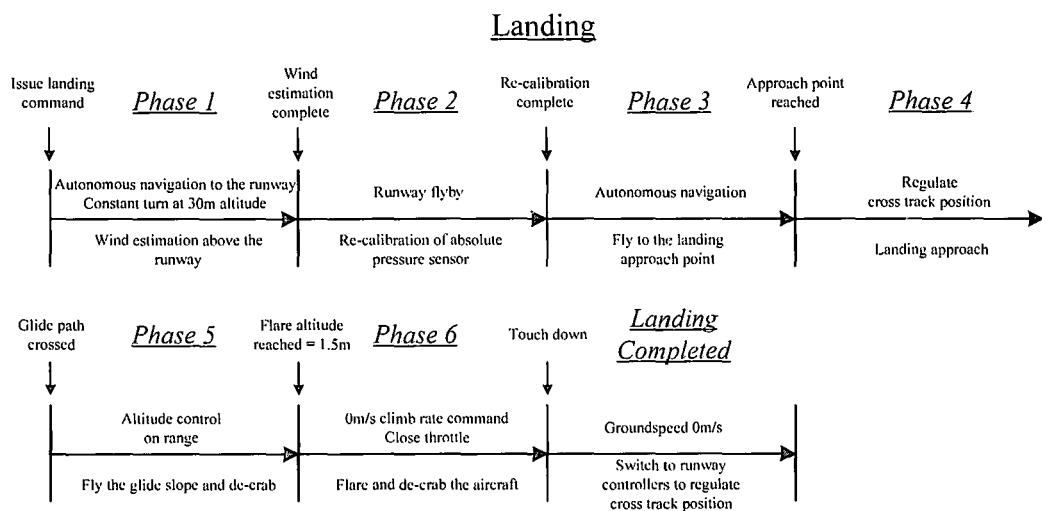


Figure 2.2: Landing Sequence Phases

Figure 2.2 shows the six phases that forms the landing sequence. When a landing command is issued, the system enters *Phase 1* where the aircraft autonomously navigates to the runway. Once above the runway, the wind estimation process is started (see section 2.2.3).

*Phase 2* is entered when a wind estimate is obtained. The aircraft then performs a flyby of the runway. The flyby is performed to re-calibrate the absolute pressure sensor with the very accurate ultrasonic range sensor. This is done as the absolute pressure sensor can display a drift in the altitude measurement of up to 4m during a fifteen minute flight. This drift becomes significant during the final stages of the landing. Re-calibration of the absolute pressure sensor also allows the aircraft to land on a different runway than the one on which the absolute pressure sensor was initialised (zeroed). The re-calibration is performed by guiding the aircraft down the runway with a constant sink rate of 1m/s. When a valid ultrasonic measurement is obtained (5m altitude), re-calibration is performed.

As soon as the absolute pressure sensor is re-calibrated, the system enters *Phase 3* where the aircraft is autonomously guided to the approach point (defined in section 2.2.4). When the approach point is reached, the system enters *Phase 4* and the landing approach starts. *Phase 4* allows the cross track error off the runway to settle.



The system enters *Phase 5* when the glide path is crossed. The glide path controller is armed and the glide path is flown until the desired flare altitude is reached, signalling the start of *Phase 6*. During *Phase 5* the de-crab controller is also armed to slowly de-crab the aircraft before touchdown.

When an altitude of 1.5m is reached on the glide path, the system transitions to *Phase 6*. The altitude measurement is obtained from an ultrasonic range sensor. In *Phase 6* the airspeed and climb rate controller, as well as the altitude controller, is disabled with the throttle fully closed. The climb rate controller is armed and a 0m/s climb rate command is issued to flare the aircraft. At touchdown, determined by the ultrasonic altitude, the lateral air-borne controllers are replaced by the lateral runway controllers to further guide the aircraft down the runway. The landing is complete when the aircraft comes to a standstill.

### 2.2.2 Landing Controllers

The controllers designed, re-designed or used to autonomously perform the landing sequence discussed in section 2.2.1, are:

- Longitudinal:
  - Glide Path Controller
  - Climb Rate Controller
  - Flare Controller
  - All the Longitudinal Controllers from [1]
- Lateral:
  - De-Crab Controller
  - All the Lateral Controllers from [1]

The runway guidance (cross track error) controller designed for taxiing and take-off are used just after touchdown to further regulate the trajectory of the aircraft.

The main disturbances acting on the aircraft during the landing process (other than sensor noise, drift and biases) that must be rejected by the controllers and kept in mind while designing, are:

1. Constant crosswind and especially wind gusts. Wind gusts are the major disturbance during landing, especially on a small light aircraft such as the one used in this project.
2. Ground effect phenomena (explained in section 2.1.2) that takes affect during to final moments of landing. When the induced drag is diminished, and the lift increases, the controllers should not counter the climb too aggressively, as this can push the aircraft into the ground.

These effects will directly influence the selected controller design specifications. The disturbances, with the steps taken to reject them are further discussed in Chapter 5, as well as in the controller discussions that follows.

### Conventional Flight Controllers

The autonomous navigation controllers developed in [1], were all designed using the linear aircraft model discussed in section 1.2.1 (details in section F.3). The design is only valid around a single, level flight, trim condition. All the conventional flight controllers will be used during the landing sequence where the trim

condition will continuously be changing. It is therefore necessary to extend the usable range (set of valid trim condition) of the conventional flight controllers through gain scheduling.

The airborne guidance controller is susceptible to gyroscope biases that are typical of the low cost sensors used in this project. Simulation results illustrated a worst case scenario steady state cross track error of 6m with roll and yaw rate gyroscope biases of 0.5deg/s. Although this is acceptable for conventional flight, improvement is called for if the controllers are to be used during landing. The guidance controller is therefore re-designed in an attempt to minimise the steady state position errors.

### **Glide Path Controller**

The glide path controller is responsible for guiding the aircraft along the glide path. Flying the glide path requires the aircraft to maintain a constant sink rate that depends on the forward velocity and the glide path angle. This is achieved by feeding forward a constant climb rate command to the airspeed and climb rate controller. The glide path controller is then used to generate an altitude command based on the range from the glide path origin point to regulate the glide path flight.

### **Flare Controller**

The final and most dangerous section of the landing sequence is the transition from the glide path to the runway through flaring. In this project the aircraft is flared with the climb rate controller developed for take-off, as explained in the landing sequence discussion.

An alternative and more controlled solution to flaring is to guide the aircraft along an exponential flare path onto the runway by providing climb rate commands to the climb rate controller. The commands are based on the altitude measurement obtained from an accurate altitude sensor such as an ultrasonic range sensor.

The exponential flare controller was not implemented in this project for the following reasons:

1. The bandwidth of the underlying climb rate controller is too low to track an exponential flaring path. The bandwidth cannot be increased due to the noise present on the climb rate measurement.
2. The exponential flare controller does not provide any guarantee of a positive pitch angle at touchdown. This can have catastrophic effects. Flaring the aircraft by commanding a 0m/s climb rate, ensures a positive pitch angle at touchdown and is the simplest and least error prone approach.
3. The exponential flare controller will not perform well in the presence of wind disturbances. Wind gusts can cause the aircraft to land very far down the runway or even before it. This is because the throttle will be closed while flying the flare path, with only the climb rate being controlled. The flaring method implemented ensures that the aircraft flies the glide path for as long as possible. This also ensures that the aircraft touches down on the runway, even in the presence of a strong head wind.

However, the flare controller design is still presented in Chapter 5 for the sake of completeness and possible use in a future project. A pitch angle estimate and a climb rate measurement with less noise can be used in a MIMO controller design to track the flare path while simultaneously controlling the pitch angle.

## De-Crab Controller

A de-crab controller used during a crosswind landing is the only additional lateral controller required.

The magnetometer onboard the aircraft can be used to calculate the body axes heading of the aircraft, i.e. the direction (heading) in which the aircraft is pointing. The GPS heading measurement is the course over ground heading of the aircraft, i.e. the heading in which the aircraft is moving. De-crabbing of the aircraft before touchdown can be achieved by feeding back the error between the body axes heading measurement from the magnetometer and the course over ground heading measurement from the GPS to the rudder.

If the magnetometer axis system is perfectly aligned with the body axis system of the aircraft, the roll and pitch angles of the aircraft must be zero in order to obtain an accurate heading measurement from the magnetometer. The roll angle of the aircraft is assumed zero during landing and the pitch angle very small ( $2^\circ$  to  $4^\circ$ ). This will satisfy the small angle requirement. However, the inclination angle of the earth's magnetic field at  $-33.91^\circ$  latitude and  $18.49^\circ$  longitude is approximately  $60^\circ$ . This means that the earth's magnetic field has a very large component along the  $Z_E$ -axis and relatively small components along the  $X_E$ -axis and  $Y_E$ -axis. The large component along the  $Z_E$ -axis causes the heading measurement obtained from the magnetometer to be very sensitive to changes in the roll and pitch angles of the aircraft (simulation results given in section 5.4). The large heading errors can cause instability and the de-crab controller is therefore not used in this project.

Nevertheless, the de-crab controller design is still presented in Chapter 5 for the sake of completeness and possible use in a future project where the body axis heading can be obtained from an optimal estimate and not the raw magnetic measurements.

### 2.2.3 Wind Estimation

It is desired to perform the take-off and especially the landing sequence into the wind. This will ensure that the airspeed is high enough to prevent the aircraft from stalling, while the groundspeed is as low as possible to limit damage to the undercarriage.

An estimate of the constant wind and average wind gusts is obtained by entering the aircraft into a constant turn and measuring the change in north and east displacement of the aircraft after a full circle has been flown. The position change, along with the time it took to complete a full circle, is used to calculate the north and east velocity components as well as the heading of the wind vector. By virtue of the measurement method, wind gusts are filtered and averaged during the constant turn to provide an accurate estimate of the wind.

## Wind Estimation Procedure

A graphical description of the wind estimation procedure is shown in Figure 2.3.

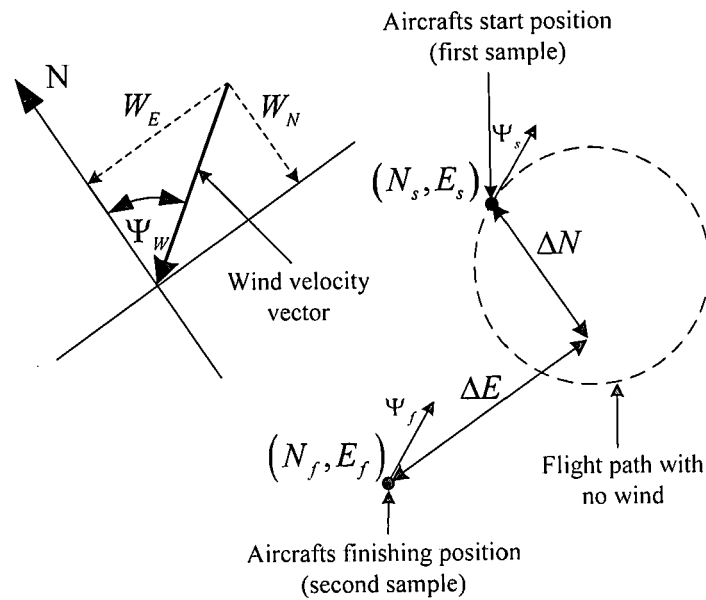


Figure 2.3: Wind Estimation Procedure

After the aircraft has settled into a constant turn, a sample of the aircraft's current north and east displacements,  $(N_s, E_s)$ , along with its heading ( $\Psi_s$ ), is taken and stored. The aircraft heading is then constantly compared to the initial heading and a full circle is signalled when the two headings are within  $4^\circ$  of each other. Once the heading match occurs, a second sample of the north and east displacements,  $(N_f, E_f)$  is taken. The change in position due to the constant wind is then given by:

$$\Delta N = N_f - N_s \quad (2.2)$$

$$\Delta E = E_f - E_s \quad (2.3)$$

The north and east velocity components of the constant wind are calculated from:

$$W_N = \frac{\Delta N}{t} \quad (2.4)$$

$$W_E = \frac{\Delta E}{t} \quad (2.5)$$

where,  $t$  is the time it took to complete the full rotation. The wind heading is calculated from:

$$\Psi_w = \tan^{-1} \left( \frac{\Delta E}{\Delta N} \right) \quad (2.6)$$

## 2.2.4 Landing Path

### Path Geometry

The landing path is made up of the final four phases of the landing sequence and is shown in Figure 2.4 and Figure 2.5. The zero point in Figure 2.4 is the point on the runway where the aircraft's sensors are initialised and serves as the reference point for the landing path and sequence.

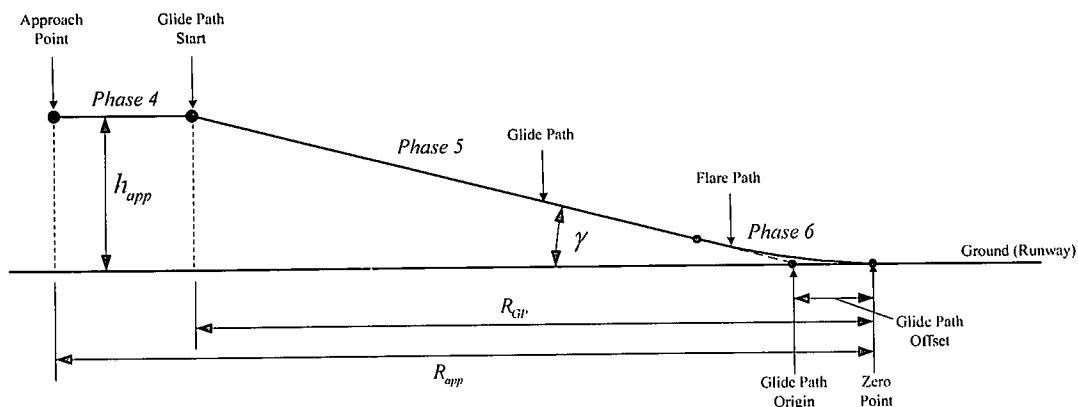


Figure 2.4: Landing Path Geometry (Side View)

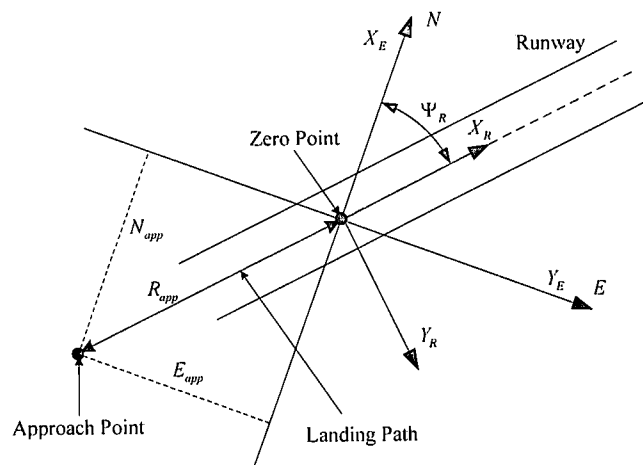


Figure 2.5: Landing Path Geometry (Top View)

The landing path variables are defined as:

- Approach Range ( $R_{app}$ ) – The range from the zero point where the approach point is placed.
- Glide Path Start Range ( $R_{GP}$ ) – The range from the zero point where the glide path starts.
- Glide Path Angle ( $\gamma$ ) – The desired angle of the glide path.
- Glide Path Offset – The range from the zero point where the glide path originates.
- Path heading ( $\Psi_R$ ) – The heading that the landing path lies in which must coincide with the runway heading.

The approach point is the waypoint to which the aircraft navigates during phase 3 of the landing sequence to start the landing approach. The approach waypoint parameters are:

Approach point:

- North displacement =  $N_{app} = R_{app} \cos(\Psi_R)$
- East displacement =  $E_{app} = R_{app} \sin(\Psi_R)$
- Heading =  $\Psi_R$
- Altitude =  $h_{app} = R_{app} \sin(\gamma)$

### Path Heading

The landing path of Figure 2.4 lies in the specified heading for the path that must coincide with the runway heading. This specified path heading is used as the reference heading in which the system will perform the take-off and landing sequences. If the result from the wind estimation procedure (discussed in section 2.2.3) indicates that the wind heading,  $\Psi_w$ , is greater than  $\Psi_R + 90^\circ$  or less than  $\Psi_R - 90^\circ$ , the landing path heading and zero point must be adjusted to ensure a head wind landing. Since the heading range is from  $0^\circ$  to  $360^\circ$ , the switching criteria's for the path heading and zero point are,

Runway heading in quadrants 2 or 3, switch when

$$(\Psi_w > \Psi_R + 90^\circ) \text{ OR } (\Psi_w < \Psi_R - 90^\circ) \quad (2.7)$$

Runway heading in quadrants 1 or 4, switch when

$$(\Psi_w > \Psi_R + 90^\circ) \text{ AND } (\Psi_w < \Psi_R - 90^\circ) \quad (2.8)$$

Switching the landing paths heading and zero point implies adding  $180^\circ$  to the specified path (runway) heading and moving the zero point to the opposite end of the runway. Moving the zero point is achieved by rotating the inertial axis system through the runway heading to form a new axis system, referred to as the runway inertial axes. The  $X$ -axis of the runway inertial axes system ( $X_R$  in Figure 2.5) is therefore aligned with the runway. The transformation of a point in the runway inertial axes to the inertial axes with reference to Figure 2.5 is given by:

$$\begin{bmatrix} X_E \\ Y_E \end{bmatrix} = \begin{bmatrix} \cos(\Psi_R) & -\sin(\Psi_R) \\ \sin(\Psi_R) & \cos(\Psi_R) \end{bmatrix} \begin{bmatrix} X_R \\ Y_R \end{bmatrix} \quad (2.9)$$

where  $(X_E, Y_E)$  is the inertial axes position,  $(X_R, Y_R)$  is the runway inertial axes position and  $\Psi_R$  is the runway heading defined in inertial axes.

The new inertial position of the zero point is therefore given by:

$$\begin{bmatrix} X_E \\ Y_E \end{bmatrix} = \begin{bmatrix} \cos(\Psi_R) & -\sin(\Psi_R) \\ \sin(\Psi_R) & \cos(\Psi_R) \end{bmatrix} \begin{bmatrix} l_R \\ 0 \end{bmatrix} \quad (2.10)$$

where  $l_R$  is the specified length of the runway.

## 2.3 Cross Track Error Algorithm

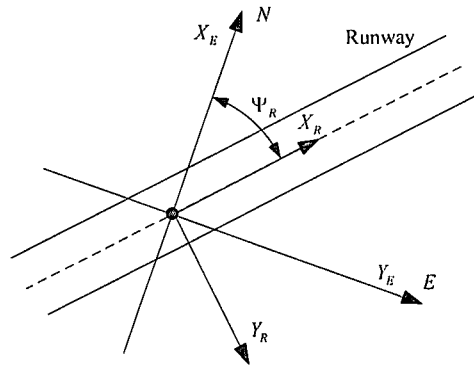


Figure 2.6: Cross Track Error Geometry

With reference to Figure 2.6, the cross track error of the aircraft off the runway for the take-off and landing can be obtained by rotating the inertial axis system through the runway heading angle to form the runway inertial axis system as was done in section 2.2.4. The  $X$ -axis of the runway inertial axis system is therefore aligned with the runway. The cross track error of the aircraft off the runway is then simply the  $Y$ -axis coordinate in the runway inertial axes.

The transformation from a position in inertial axes to the runway inertial axes is given by:

$$\begin{bmatrix} X_R \\ Y_R \end{bmatrix} = \begin{bmatrix} \cos(\Psi_R) & \sin(\Psi_R) \\ -\sin(\Psi_R) & \cos(\Psi_R) \end{bmatrix} \begin{bmatrix} X_E \\ Y_E \end{bmatrix} \quad (2.11)$$

where  $(X_E, Y_E)$  is the inertial axes position,  $(X_R, Y_R)$  is the runway inertial axes position and  $\Psi_R$  is the runway heading defined in inertial axes. The cross track error off the runway is then given by:

$$Y_R = Y_E \cos(\Psi_R) - X_E \sin(\Psi_R) \quad (2.12)$$

## 2.4 Controller Overview

With the take-off and landing sequences, as well as the underlying controllers discussed, the entire autopilot can be considered. A block diagram representation of the autopilot is shown in Figure 2.7.

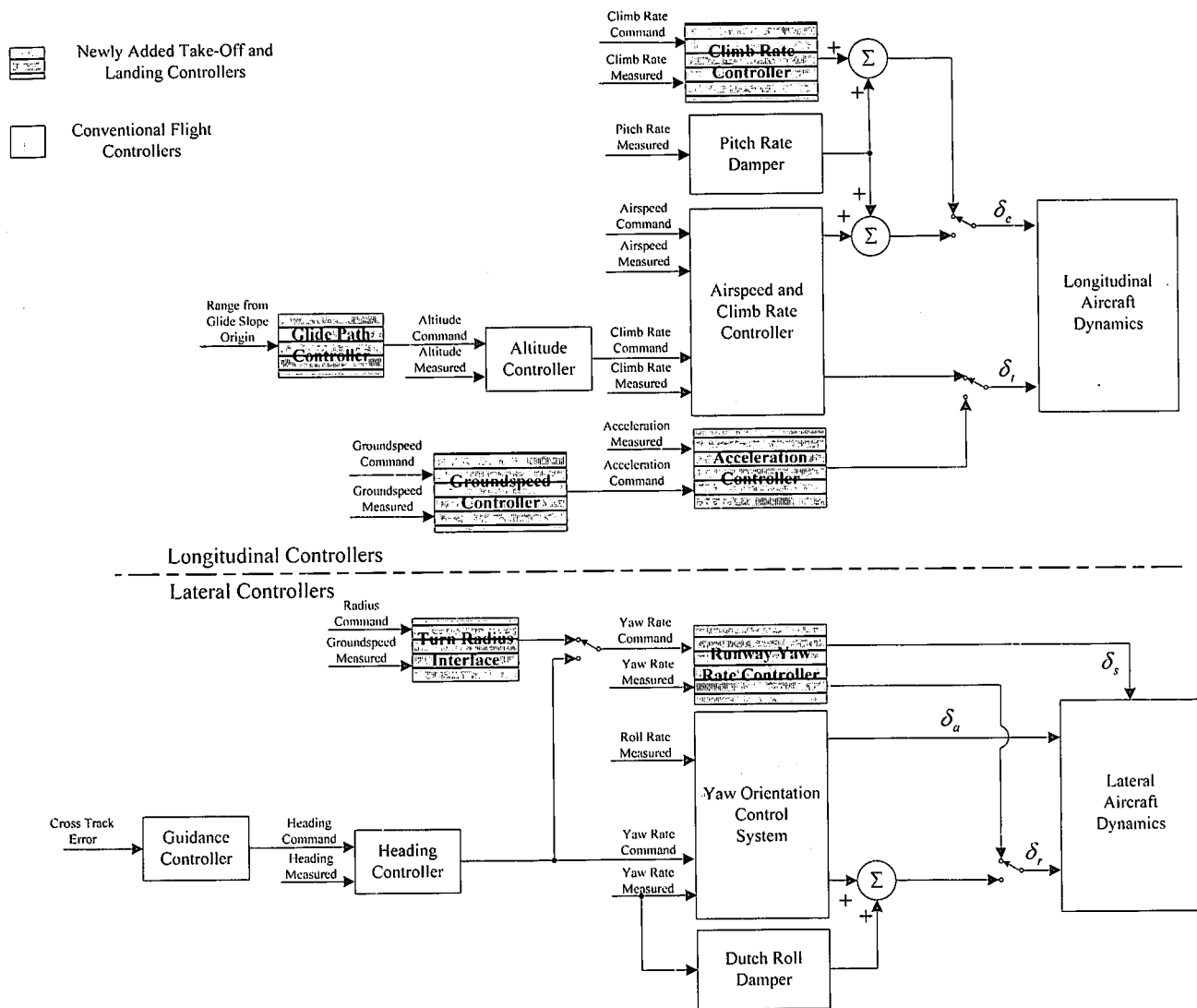


Figure 2.7: Complete Autopilot Overview

The system is capable of autonomous take-off, navigation and landing. This enables it to perform autonomous air missions with minimum human intervention. A human is only required in the loop to provide the runway heading, place the aircraft on the runway and issue the flight command as well as retrieving the aircraft after the mission is complete. All of this is accomplished with only the low cost sensors onboard the aircraft and does not require additional sensors on the runway for guidance during take-off or landing.



The controllers implemented in the autopilot are summarised below:

- Longitudinal:
  - Pitch Rate Damper
  - Airspeed and Climb Rate Controller
  - Climb Rate Controller
  - Altitude Controller
  - Glide Path Controller
  - Acceleration Controller
  - Groundspeed Controller
- Lateral:
  - Dutch Roll Damper
  - Airborne and Runway Yaw Rate Controllers
  - Heading Controller
  - Guidance Controller
  - Turn Radius Interface

Only two additional sensors need to be added to the set used for autonomous navigation in order to achieve autonomous take-off and landing; an ultrasonic range sensor and a single-axis accelerometer. The entire autopilot can therefore be implemented using the following low cost, off-the-shelf sensors:

- 3 Rate Gyroscopes
- 1 Single-Axis Accelerometer
- Absolute and Differential Pressure Sensors
- GPS Receiver
- Ultrasonic Range Sensor

All the airborne controllers are designed using the linear aircraft model developed in [1] as summarised in section F.3, while all the runway controllers are designed from the linear models developed in Chapter 3.

The heading, guidance and groundspeed controllers are designed in the discrete time domain, since the GPS receiver used is only capable of providing measurement updates at 4Hz. The rest of the controllers are designed in the continuous domain and discretised using Euler's method at a sample rate of 25Hz. The rate was chosen in [1] as it is a convenient rate for implementation, well above the frequencies of the aircraft's natural modes of motion. It also makes other methods such as Tustin's discretisation method unnecessary.

All the controllers are designed in MATLAB. The runway yaw rate controller is design using full state feedback methods whereas all other controllers are designed using root locus methods. All the linear simulation results were generated using Simulink.

## Smooth Transition

To ensure the smooth transition of the actuators when switching between autopilot and pilot control, a first order transition response was implemented in [1]. When switching control from the pilot to the autopilot, actuator jumps are avoided by setting up the integrator states such that the current autopilot actuator commands coincide with the pilot actuator commands at the transition. The same techniques are implemented in the newly added controllers to ensure actuator smooth transition.

During the take-off and landing sequence, actuator control will also be switched between different controllers. To avoid actuator jumps between controller to controller switching the integrator states of the controller to be activated are set up to coincide with the last actuator commands from the controller active at the point of transition.

## 2.5 Summary

This chapter discussed the higher level system functions and concepts. The take-off and landing processes with the different phases that the system goes through were also presented. The background and thought process behind the controllers were included, forming the base of the controller designs that follows in Chapter 4 and Chapter 5. Details on the landing path and wind estimation procedure were also discussed. Finally the algorithm used to calculate the cross track error off the runway was developed and an overview of the entire system provided.

# Chapter 3

## *Modelling*

In order to simulate the take-off and landing of an aircraft, a non-linear runway model is required and is developed in section 3.1. In section 3.2 this non-linear runway model is used to derive the linear runway models required for runway controller design in Chapter 4. The engine model developed in [1] is based on a constant power formula and cannot be used to simulate take-off and landing. A new engine model is therefore required and consequently developed in section 3.3. A wind gust model is developed in section 3.4.

### 3.1 Non-Linear Runway Model

In this section a model of the forces and moments that act on the aircraft while it is on the runway is developed to augment the non-linear aircraft model. The model is essential for simulating the take-off and landing process and the linearised model is used for controller design purposes in Chapter 4.

The runway model is developed for a three wheeled aircraft with a tricycle undercarriage, but can easily be transformed to a model for a tail dragger. The development starts by defining the axis system used, followed by the three main forces that act on the aircraft. The runway forces are then used to derive the desired resultant forces and moments that act on the aircraft.

In most cases of vehicle dynamic calculations, it is sufficient to use only the static vehicle weight for relatively accurate calculations [6]. However, it is essential to model the dynamic weight transfer (normal force on each wheel) because of the aerodynamic lift of the aircraft. It is especially important during take-off and landing where one wheel may touch down or lift-off before the others.

#### 3.1.1 Axis System Definition

In addition to the axis systems of section F.1, a third body fixed axis system is required for the convenient description and formulation of the forces and moments that the runway exerts on the aircraft. This axis system will be referred to as the *Runway Body Axis* system. The standard body axis system is referred to as *Reference Line Body Axes* to prevent confusion with the runway body axes.

## Runway Body Axes

The runway body axis system  $F_{B_R} = (O_{B_R}, X_{B_R}, Y_{B_R}, Z_{B_R})$  is defined as a right handed reference frame that is attached to the aircraft with its centre coinciding with the aircraft's centre of gravity. The  $X_{B_R}Z_{B_R}$ -plane lies in the plane of symmetry of the aircraft with the  $X_{B_R}$ -axis positive in the 'forward' direction. The  $Z_{B_R}$ -axis is perpendicular on the  $X_{B_R}$ -axis and positive in the 'down' direction. The  $Y_{B_R}$ -axis is perpendicular to the  $X_{B_R}Z_{B_R}$ -plane and positive to the right. The fundamental difference between the *Runway Body Axis* system and a conventional body axis system is that the *Runway Body Axis* system only yaws with the aircraft. It does not roll and pitch. The  $X_{B_R}Y_{B_R}$ -plane is therefore always parallel to the  $X_EY_E$ -plane (ground) and the  $Z_{B_R}$ -axis always points in the same direction as the  $Z_E$ -axis. The *Runway Body Axis* system is a *Locally Level Axis* system.

All the forces discussed in section 3.1.2 are referenced in the *Runway Body Axis* frame and the transformation of these forces to the *Stability Axis* frame is discussed in section 3.1.3.

### 3.1.2 Runway Forces

Three types of forces act on the aircraft due to the runway:

- Friction Forces
- Cornering Forces
- Normal Forces

#### Friction Forces

While the aircraft is taxiing on the runway, longitudinal friction forces act on it due to the rolling resistance between the tyre and ground surfaces and the bearing friction in the wheels. The other major aircraft resistance force is aerodynamic resistance (drag), but this only becomes significant at high speeds. At low taxiing speeds the rolling resistance is the primary retardation force.

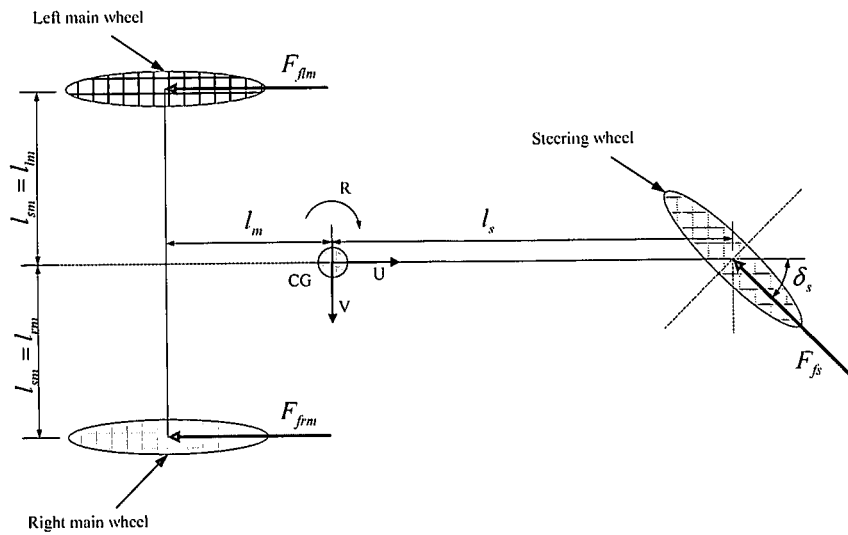
The friction force acting on a specific wheel is modelled as being *directly proportional to the weight* (normal force) on the wheel and is related by the *rolling resistance coefficient*. The friction forces acting on the three wheels of the aircraft is given by:

$$F_{fs} = \mu_f N_s \quad (3.1)$$

$$F_{flm} = \mu_f N_{lm} \quad (3.2)$$

$$F_{frm} = \mu_f N_{rm} \quad (3.3)$$

where  $\mu_f$  is the rolling resistance coefficient (assumed identical for all wheels),  $N_s$  is the normal force acting on the steering wheel, while  $N_{lm}$  and  $N_{rm}$  are the normal forces acting on the left and right main wheel respectively. The forces are explained graphically in Figure 3.1.

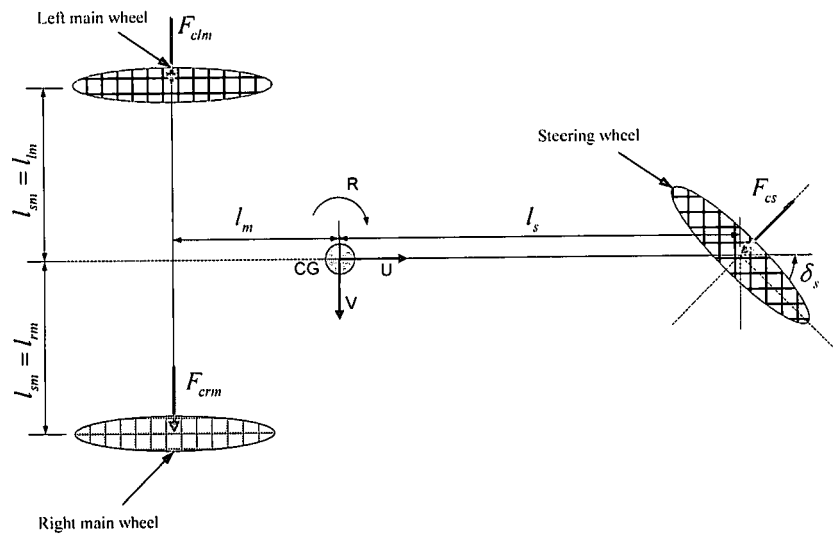


**Figure 3.1:** Top View of the Under Carriage with Friction Forces

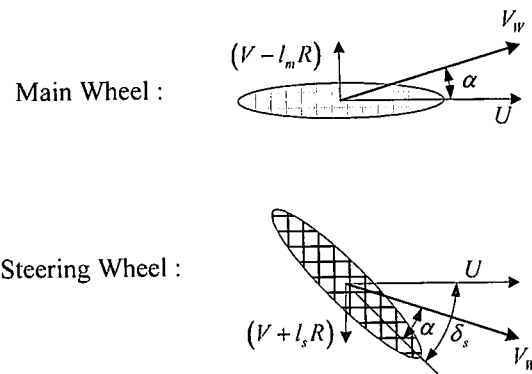
Factors affecting the rolling resistance are [6]: tyre temperature, tyre inflation pressure, velocity, tyre material and design and tyre slip (defined under the Cornering Forces section). The influence of these factors are small and will only complicate the modelling process. They are therefore ignored.

### Cornering Forces

The lateral forces acting on the aircraft (tyres) due to the runway are known as cornering forces. The slip angle  $\alpha$  of a wheel is defined as the angle between the direction of travel and the direction of heading of the wheel [6].



**Figure 3.2:** Top View of the Under Carriage with Cornering Forces



**Figure 3.3:** Pictorial Descriptions of the Slip Angle

The slip angles of the main wheels and steering wheel can be described, with reference to Figure 3.2 and Figure 3.3, by [7]:

$$\alpha_m = \arctan\left(\frac{V - l_m \cdot R}{|U|}\right) \quad (3.4)$$

$$\alpha_s = \arctan\left(\frac{V + l_s \cdot R}{|U|}\right) - \delta_s \quad (3.5)$$

where  $\alpha_m$  and  $\alpha_s$  is the slip angle of the main wheels and steering wheel respectively.  $l_m$  and  $l_s$  are the lengths from the main wheels to the aircraft's centre of gravity and the steering wheel to the aircraft's centre of gravity respectively.  $\delta_s$  is the steering angle and  $V_w$  is the wheels velocity vector.

The cornering force on the wheel is modelled as being directly proportional to the slip angle (at slip angles of less than five degrees) and is given by:

$$F_c = C_\alpha \alpha \quad (3.6)$$

The proportionality constant  $C_\alpha$  is known as the *cornering stiffness* and is dependent on many variables of which normal load and inflation pressure are the most influential. The inflation pressure is assumed constant and the change in cornering stiffness ( $C_\alpha$ ) with tyre load (normal force) is linear. The cornering stiffness can then be described in terms of the *cornering coefficient* and the tyre load:

$$C_\alpha = C_{\alpha\alpha} N \quad (3.7)$$

where  $C_{\alpha\alpha}$  is the *cornering coefficient* and  $N$  is the *normal force* on the wheel. Note that a positive slip angle produces a negative force on the tyre, implying that  $C_{\alpha\alpha}$  and  $C_\alpha$  is negative. The cornering force on a wheel is the given by:

$$F_c = C_{\alpha\alpha} N \alpha \quad (3.8)$$

Note from equations (3.4) and (3.5) that, at low forward velocities, the slip angles become large enough to violate the assumption that the cornering forces are linearly related to the slip angle and the accuracy of the model will suffer. However, it was found that this only becomes a problem at speeds of less than  $0.5\text{ m/s}$ . A model for low speed cornering is not required since the model is intended for use during take-off and landing at much higher speeds.

### Normal Forces

The normal forces acting on the aircraft are the reaction forces from the runway countering the weight of the aircraft at each wheel through the aircraft's suspension. These forces are the most important, since both the friction and cornering forces are modelled as being directly proportional to the normal force at each wheel.

The aircraft's suspension dynamics are modelled by a spring-damper system at each wheel. The tyre dynamics can also be modelled by a mass and spring systems attached to the suspension, but this was not done since it adds complexity with very little increase in model fidelity. It is assumed that the suspension systems at each wheel always remain vertical, i.e. that the roll and pitch angles of the aircraft are small.

The force developed by a linear elastic spring with a spring constant  $k_s$ , that is deformed (elongated or compressed) a distance  $s$ , is modelled as:

$$F_s = sk_s \quad (3.9)$$

The force developed by a linear viscous damper is modelled as being directly proportional to the rate (velocity) at which the damper is elongated or compressed with proportionality constant  $k_d$ :

$$F_d = vk_d \quad (3.10)$$

where  $v$  is the compression rate (velocity). The normal force acting on the aircraft at each wheel is then given by the sum of the forces produced by the spring and the damper. To obtain these forces, the deforming distance and compression rate from equations (3.9) and (3.10) must be calculated. It is therefore necessary to describe the position and velocity (in inertial axes) of every point on the aircraft's body where the normal forces will act (the point where the suspension connects to the body).

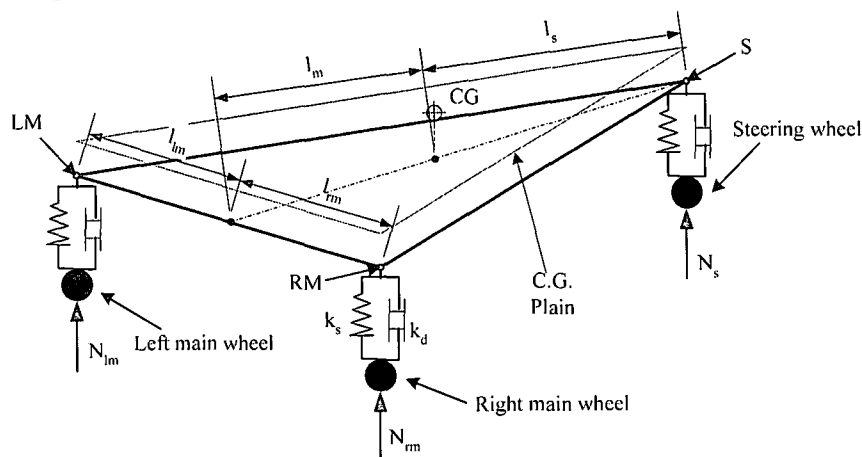


Figure 3.4: 3D View of the Under Carriage with Normal Forces

Figure 3.4 shows the system and the three points that need to be described. The points are labelled S for steering, LM for Left Main and RM for Right Main. The position vectors for the three points in reference line body axes are given by:

$$S_B = [l_s \quad 0 \quad 0]^T \quad (3.11)$$

$$RM_B = [-l_m \quad l_{sm} \quad 0]^T \quad (3.12)$$

$$LM_B = [-l_m \quad -l_{sm} \quad 0]^T \quad (3.13)$$

The following assumptions are made about the three suspension points:

- The three points (S, LM and RM), as well as the aircraft's centre of gravity, all lie in the same plane.
- The aircraft is symmetrical around the  $X_B Z_B$ -plane, i.e. the distance from LM to the CG is equal to the distance from RM to the CG.

The increased complexity introduced when a  $Z_B$ -position offset are added to the three points (S, LM and RM) is not justified by the increase in model fidelity and are therefore ignored.

The next step in determining the position of points S, LM and RM in inertial axes is to coordinate the known reference line body axes position components to inertial axes components by means of the DCM. The origin of the re-coordinated inertial position vectors is still the centre of gravity. To obtain the absolute position of each point in inertial axes, the centre of gravity position of the aircraft must be added after the transformation. Note that the DCM of section F.2 will transform between the *inertial axis system* and the *stability axis system* and not the *reference line body axis system*. Since the trim angle of attack ( $\alpha_0$ ) is typically less than four degrees, the error introduced is ignored for the purpose of developing the runway model.

The desired absolute position vectors of points S, LM and RM in inertial axes are described by the following equations:

$$S_E = [DCM]^T [S_B] + CG \quad (3.14)$$

$$RM_E = [DCM]^T [RM_B] + CG \quad (3.15)$$

$$LM_E = [DCM]^T [LM_B] + CG \quad (3.16)$$

With the absolute positions of S, LM and RM known in inertial axes, the deformation of the spring part of the suspension can be calculated. The deformation of the spring at each point is given by:

$$D_S = (Z_S + |Z_{S0}|) \quad (3.17)$$

$$D_{RM} = (Z_{RM} + |Z_{RM0}|) \quad (3.18)$$

$$D_{LM} = (Z_{LM} + |Z_{LM0}|) \quad (3.19)$$



where  $Z_S$  is the Z position of point S in inertial axes (the Z component of equation (3.14)) and  $|Z_{S0}|$  is the absolute value of the unloaded Z position of point S in inertial axes. Similar definitions apply for points LM and RM. Note that the Z-axis of the inertial axis system is defined as positive downwards, therefore the Z position in equations (3.17) to (3.19) is negative.

Now that the spring component of the normal force can be calculated, the damping component still needs to be calculated. The Z velocities of points S, LM and RM in inertial axes are required for this component. The velocities of these points in reference line body axes are given by [4]:

$$\begin{bmatrix} U_{S_B} \\ V_{S_B} \\ W_{S_B} \end{bmatrix} = \begin{bmatrix} U_B \\ V_B + R_B l_s \\ W_B - Q_B l_s \end{bmatrix} \quad (3.20)$$

$$\begin{bmatrix} U_{RM_B} \\ V_{RM_B} \\ W_{RM_B} \end{bmatrix} = \begin{bmatrix} U_B - R_B l_{sm} \\ V_B + R_B (-l_m) \\ W_B - Q_B (-l_m) + P_B l_{sm} \end{bmatrix} \quad (3.21)$$

$$\begin{bmatrix} U_{LM_B} \\ V_{LM_B} \\ W_{LM_B} \end{bmatrix} = \begin{bmatrix} U_B - R_B (-l_{sm}) \\ V_B + R_B (-l_m) \\ W_B - Q_B (-l_m) + P_B (-l_{sm}) \end{bmatrix} \quad (3.22)$$

The velocity components in inertial axes are obtained by a DCM transformation and given by:

$$\begin{bmatrix} U_{S_E} \\ V_{S_E} \\ W_{S_E} \end{bmatrix} = [DCM]^T \begin{bmatrix} U_{S_B} \\ V_{S_B} \\ W_{S_B} \end{bmatrix} \quad (3.23)$$

$$\begin{bmatrix} U_{RM_E} \\ V_{RM_E} \\ W_{RM_E} \end{bmatrix} = [DCM]^T \begin{bmatrix} U_{RM_B} \\ V_{RM_B} \\ W_{RM_B} \end{bmatrix} \quad (3.24)$$

$$\begin{bmatrix} U_{LM_E} \\ V_{LM_E} \\ W_{LM_E} \end{bmatrix} = [DCM]^T \begin{bmatrix} U_{LM_B} \\ V_{LM_B} \\ W_{LM_B} \end{bmatrix} \quad (3.25)$$

The Z components of each velocity vector given by equation (3.23) to (3.25) are the desired Z velocities in inertial space of the points S, LM and RM. With both the deformation and velocity for respectively the spring and damper in the suspension system known, the only step that remains is forming the normal forces acting on the body of the aircraft by using equations (3.9) and (3.10).

The normal forces at points S, LM and RM are described by:

$$N_S = (Z_{S_E} + |Z_{S0_E}|)k_s + W_{S_E}k_d \quad (3.26)$$

$$N_{RM} = (Z_{RM_E} + |Z_{RM0_E}|)k_s + W_{RM_E}k_d \quad (3.27)$$

$$N_{LM} = (Z_{LM_E} + |Z_{LM0_E}|)k_s + W_{LM_E}k_d \quad (3.28)$$

The normal forces are with respect to the runway body axis frame.

### 3.1.3 Resultant Forces and Moments

Now that the forces that act on the aircraft due to the runway have been identified and their associated equations developed, the resultant forces and moments that act on the aircraft can be obtained and coordinated in the stability axis system. This is a two-step process, where the resultant forces and moments are first obtained in the runway body axis system and then transformed to the stability axis system.

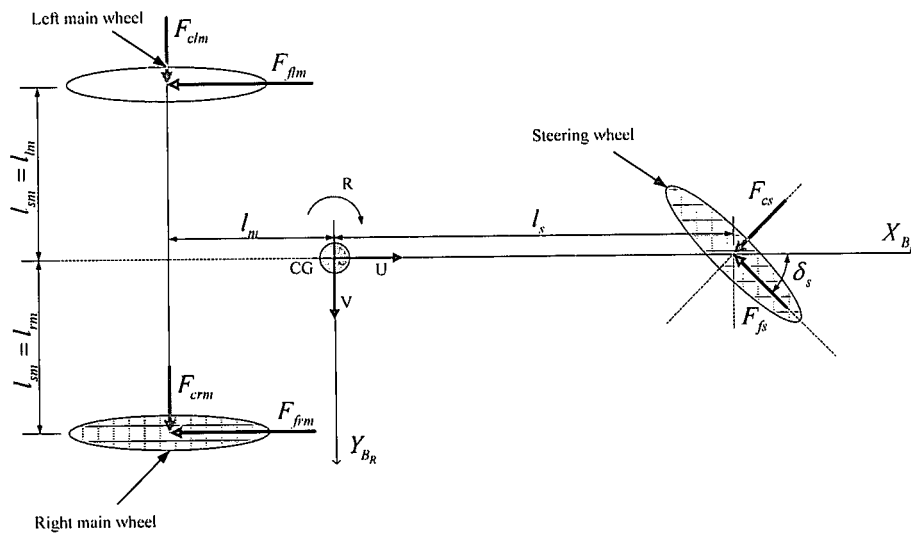


Figure 3.5: Top View of the Under Carriage with Runway Forces

With reference to Figure 3.5, the resultant forces acting on the aircraft in the  $X_{B_R}$ -axis and the  $Y_{B_R}$ -axis, as well as the yaw moment acting around the  $Z_{B_R}$ -axis, are calculated by:

$$\sum F_x = -F_{f/m} - F_{c/s} \sin(\delta_s) - F_{f/s} \cos(\delta_s) - F_{f/r/m} = F_{x_{B_R}} \quad (3.29)$$

$$\sum F_y = F_{c/m} + F_{c/s} \cos(\delta_s) - F_{f/s} \sin(\delta_s) + F_{c/r/m} = F_{y_{B_R}} \quad (3.30)$$

$$\sum N_{cg} = F_{c/s} \cos(\delta_s) l_s - F_{c/m} l_m - F_{f/m} l_{sm} - F_{f/s} \sin(\delta_s) l_s - F_{c/r/m} l_m + F_{f/r/m} l_{sm} = N_{B_R} \quad (3.31)$$

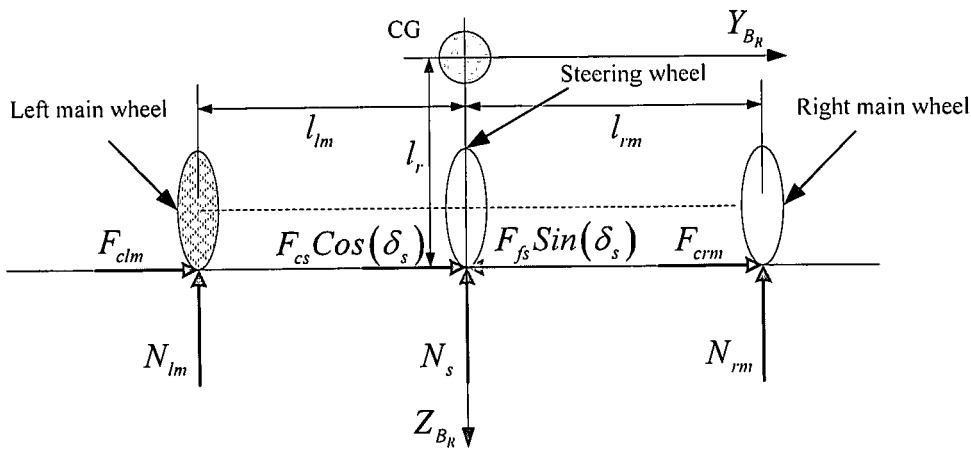


Figure 3.6: Rear View of the Under Carriage with Runway Forces

Figure 3.6 shows the rear view of the aircraft under carriage. The resultant force acting in the  $Z_{B_R}$ -axis and the rolling moment around the  $X_{B_R}$ -axis are calculated by:

$$\sum L_{cg} = -F_{csm}l_r + N_{lm}l_{sm} - F_{cs} \cos(\delta_s)l_r + F_{fs} \sin(\delta_s)l_r - F_{crm}l_r - N_{rm}l_{sm} = L_{B_R} \tag{3.32}$$

$$\sum F_z = -N_s - N_{rm} - N_{lm} = F_{z_{B_R}} \tag{3.33}$$

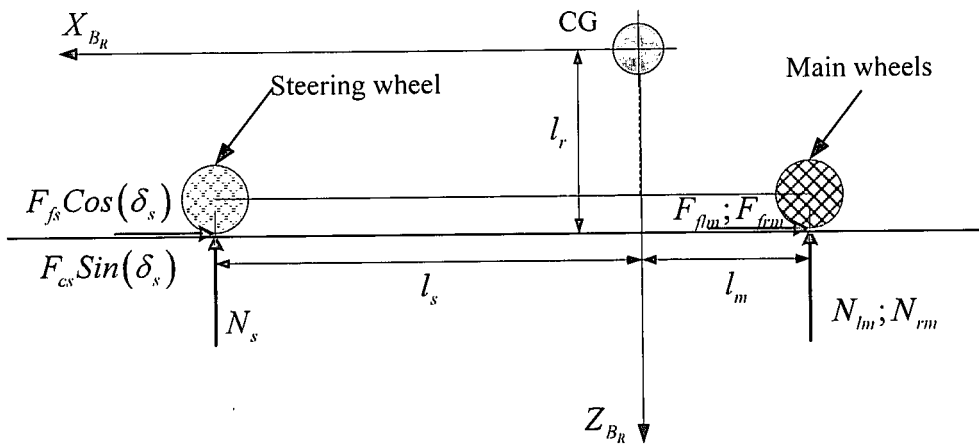


Figure 3.7: Side View of the Under Carriage with Runway Forces

The last equation that is required, is the pitching moment around the  $Y_{B_R}$ -axis. From the side view of the aircraft shown in Figure 3.7, the pitching moment is seen to be:

$$\sum M_{cg} = N_s l_s - N_{lm} l_m - F_{cs} \sin(\delta_s) l_r - N_{rm} l_m = M_{B_R} \tag{3.34}$$

### Axis Transformation

The final step in the development is to coordinate the resultant forces and moments from the runway body axis system into the stability axis system. The only difference between the runway body axis system and the inertial axis system is the aircrafts yaw angle (as explained in section 3.1.1). The transformation from runway body

axes to stability axes can therefore easily be achieved by the conventional DCM with the yaw angle set to zero. The forces acting on the aircraft coordinated in stability axes due to the runway are then given by:

$$\begin{bmatrix} X_R \\ Y_R \\ Z_R \end{bmatrix} = [DCM_{\psi=0}] \begin{bmatrix} F_{x_{BR}} \\ F_{y_{BR}} \\ F_{z_{BR}} \end{bmatrix} \quad (3.35)$$

where  $F_{x_{BR}}$ ,  $F_{y_{BR}}$  and  $F_{z_{BR}}$  are given by equations (3.29), (3.30) and (3.33) respectively.

## 3.2 Linear Runway Models

This section develops linear longitudinal and lateral models of the runway to be used for controller design purposes. Simplified longitudinal and lateral non-linear models are developed and these models are then linearised.

The following assumptions are made to simplify the non-linear models:

- The steering angle  $\delta_s$  is small ( $-10^\circ \leq \delta_s \leq 10^\circ$ ).
- The roll and pitch angles ( $\Phi$  and  $\Theta$ ) are zero on the runway.
- The airspeed during taxiing is low enough, and the error introduced during the final moments of take-off is small enough, to assume that the engine velocity derivative  $\frac{\partial T}{\partial u} = 0$ . The acceleration loop discussed in section 2.1.2 and designed in section 4.1 will also minimise the effect of the  $\frac{\partial T}{\partial u}$  term.

Additional assumptions for the lateral model:

- Only the steering wheel and the rudder-fin combination contribute to the yaw moment acting on the aircraft while it is on the runway. All other aerodynamic, engine and runway effects that contribute to the yaw moment acting on the aircraft are ignored. These effects include:
  - Differential drag on the wing.
  - Difference in the friction coefficients of the main wheels.
  - Gyroscopic and torque effects in the engine inducing differential friction on the main wheels.
  - Helical slipstream from the propeller inducing a side force on the fin.
- The roll and pitch rates are assumed small.

### 3.2.1 Linear Longitudinal Runway Model

#### Simplified Longitudinal Non-Linear Plant Model

The force acting in the  $X_{BR}$ -axis of the aircraft due to the runway is given by equation (3.29) and will be referred to as the  $X_R$ -force.

By applying the small steering angle assumption, the  $X_R$ -force reduces to:

$$X_R = -(F_{fm} + F_{fm} + F_{fs}) \quad (3.36)$$

where the  $F_{cs}\delta_s$  term is the product of two small values and is therefore ignored. Replacing the three friction forces by equations (3.1), (3.2) and (3.3), and combining the three normal forces into a single normal force ( $N_R$ ), results in the  $X_R$ -force given by:

$$X_R = -\mu_f N_R \quad (3.37)$$

The normal force,  $N_R$ , is the sum of the gravitational force ( $Z_G$ ) and the aerodynamic lift force ( $Z_A$ ) acting on the aircraft and is given by:

$$N_R = Z_G + Z_A \quad (3.38)$$

The gravitational and aerodynamic lift forces are respectively described by [1]:

$$Z_G = mg \cos(\Theta) \cos(\Phi) \quad (3.39)$$

$$Z_A = \frac{1}{2} \rho \bar{V}^2 S C_z \quad (3.40)$$

where the non-dimensional proportionality constant  $C_z = -C_L$  and  $S$  is the surface area of the wing. Applying the small roll and pitch angle assumption results in the  $X_R$ -force reducing to:

$$X_R = -\mu_f \left[ mg - \frac{1}{2} \rho \bar{V}^2 S C_L \right] \quad (3.41)$$

The aerodynamic, gravitational and thrust forces acting on the aircraft in the  $X_b$ -axis are (from [1]):

$$X_A = \frac{1}{2} \rho \bar{V}^2 S C_x \quad (3.42)$$

$$X_G = -mg \sin(\Theta) \quad (3.43)$$

$$X_T = T_0 + \frac{\partial T}{\partial u} u + \frac{\partial T}{\partial \delta_i} \delta_i \quad (3.44)$$

where the non-dimensional proportionality constant  $C_x = -C_D$ , the pitch angle  $\Theta$  is zero and the trim thrust is  $T_0 = 0$ . The units of  $\delta_i$  and  $T$  are set the same and with  $\frac{\partial T}{\partial u} = 0$ , the engine control derivative  $\frac{\partial T}{\partial \delta_i}$  reduces to unity.

The desired non-linear model that describes the  $X_B$ -axis acceleration of the aircraft on the runway, is obtained by combining equations (3.41) to (3.44) with the  $X_B$ -axis equation of motion (equation (F.9)) to yield:

$$\delta_t - \mu_f \cdot mg + \frac{1}{2} \rho S (\mu_f C_L - C_D) \bar{V}^2 = m\dot{U} \quad (3.45)$$

### Plant Linearisation

Two methods of linearisation were considered. The first is to add the  $\mu_f \cdot mg$  and  $\frac{1}{2} \rho S (\mu_f C_L - C_D) \bar{V}^2$  terms in equation (3.45) to the throttle commands generated by a controller designed from a plant model given by:

$$\delta_t' = m\dot{U} \quad (3.46)$$

The second method is to simply ignore the  $\mu_f \cdot mg$  and  $\frac{1}{2} \rho S (\mu_f C_L - C_D) \bar{V}^2$  terms in equation (3.45). This is motivated by the friction forces being very small with the actual value uncertain due to the unknown friction coefficient,  $\mu_f$ . Even if the friction forces were known, it will still be very difficult to cancel them with an open loop throttle command due to the unknown mapping of throttle position to obtained thrust. Therefore the friction terms are ignored, resulting in the plant model being given by equation (3.46). This approach will force a high phase margin controller for robustness, with integral control to push out any disturbances due to unmodelled friction.

## 3.2.2 Linear Lateral Runway Model

### Simplified Lateral Non-Linear Plant Model

The assumptions made to simplify the non-linear lateral runway model were given at the beginning of section 3.2. The only aerodynamic effect of any real significance that must be included in the model, is the lift from the wing causing a reduction in the steering wheel effectiveness.

The yaw moment acting on the aircraft due to the runway is given by equation (3.31) and will be referred to as the  $N_R$ -moment. Applying the simplification assumptions to equation (3.31) and replacing the cornering force by equation (3.8), reduces the  $N_R$ -moment to:

$$N_R = C_{\alpha\alpha} \alpha_s N_s l_s \quad (3.47)$$

Further assuming that the side velocity on the runway ( $v$ ) is zero and that the ratio  $\frac{R}{U}$  in equation (3.5) is small, results in the  $N_R$ -moment reducing to:

$$N_R = C_{\alpha\alpha} \left( \frac{l_s R}{U} - \delta_s \right) N_s l_s \quad (3.48)$$

The normal force acting on the steering wheel is given by:

$$N_s = \left( \frac{l_m}{l_s + l_m} \right) \left( mg - \frac{1}{2} \rho \bar{V}^2 S C_L \right) \quad (3.49)$$

The first bracket describes the weight transfer between the steering wheel and main wheels. The second bracket is the weight of the aircraft that the undercarriage must support. This weight equals the total aircraft weight minus the lift with small roll and pitch angles, where  $C_L$  is the lift coefficient of the wing. Substituting equation (3.49) into equation (3.48), the  $N_R$ -moment is given by:

$$N_R = \frac{l_m l_s}{(l_s + l_m)} C_{\alpha\alpha} \left( mg - \frac{1}{2} \rho \bar{V}^2 S C_L \right) \delta_s \quad (3.50)$$

Note that  $C_{\alpha\alpha}$  is a negative coefficient. Positive  $\delta_s$  deflections will induce negative yaw moments. This satisfies the convention that a positive actuator deflection causes a negative body moment [4].

The aerodynamic yaw moment due to the rudder and fin are (from [1]),

$$N_A = \frac{1}{2} \rho \bar{V}^2 S b C_n \quad (3.51)$$

where

$$C_n = C_{n_w} + \left( \frac{S_F l_F}{S b} \right) C_{L_F} \cos(\beta) \quad (3.52)$$

$C_{n_w}$  is the yaw moment coefficient of the wing and  $C_{L_F}$  is the lift coefficient of the fin.  $C_{n_w}$  is zero under static symmetric flow conditions and no aileron deflection. For controller design purposes, it is assumed that the crosswind on the runway is zero. The side slip angle ( $\beta$ ) is therefore also assumed zero on the runway. Crosswinds will be the main disturbance to be rejected by a controller. The lift coefficient of the fin is then given by:

$$C_{L_F} = \frac{\partial C_{L_F}}{\partial \delta_r} \delta_r \quad (3.53)$$

where  $\frac{\partial C_{L_F}}{\partial \delta_r}$  is the rudder control derivative and defined as negative, again satisfying the convention that a positive actuator deflection causes a negative body moment [4]. Substituting equations (3.52) and (3.53) into equation (3.51), gives the yaw moment as a function of the airspeed due to the rudder deflection ( $\delta_r$ ):

$$N_A = \frac{1}{2} \rho \bar{V}^2 S_F l_F \frac{\partial C_{L_F}}{\partial \delta_r} \delta_r \quad (3.54)$$

The equation of motion, describing the yaw motion of the aircraft, is linearised in Appendix F, section F.3.1 and is given by:

$$N_A + N_G + N_T = \dot{r}I_z - \dot{p}I_{xz} \quad (3.55)$$

where  $N_A$ ,  $N_G$  and  $N_T$  are the yaw moments acting on the aircraft due to aerodynamics, gravity and the engine respectively. The yaw moment acting on the aircraft due to gravity ( $N_G$ ) is by definition zero. Since the engine is assumed to lie in the  $X_B Z_B$ -plane of the aircraft, the yaw moment due to the engine ( $N_T$ ) is zero. Due to the symmetry of the aircraft, the cross product of inertia  $I_{xz}$  is negligibly small and thus ignored. After adding the  $N_R$ -moment, the simplified non-linear model is given by:

$$N_A + N_R = \dot{r}I_z \quad (3.56)$$

where  $N_A$  and  $N_R$  are given by equations (3.54) and (3.50) respectively.

The steering wheel and rudder actuator are related by a 2:1 ratio. This ratio is chosen since it is the same ratio used by the human pilot. The steering wheel actuator command can therefore be written as:

$$\delta_s = 0.5\delta_r \quad (3.57)$$

in order to form a single input model. Expanding equation (3.56) and using equation (3.57) results in the non-linear model given by:

$$\left[ \frac{1}{2} \rho \bar{V}^2 S_F l_F \frac{\partial C_{L_F}}{\partial \delta_r} + \frac{0.5 \cdot l_m l_s}{(l_s + l_m)} C_{\alpha\alpha} \left( mg - \frac{1}{2} \rho \bar{V}^2 S C_L \right) \right] \delta_r = \dot{r}I_z \quad (3.58)$$

### Plant Linearisation

The non-linear model of equation (3.58) is only non-linear with respect to airspeed. The expression in square brackets in equation (3.58) is a constant value at a specific airspeed and can be written as:

$$\frac{K_{rr}(\bar{V})}{I_z} \delta_r = \dot{r} \quad (3.59)$$

Equation (3.59) is the desired linear model relating the aircrafts yaw rate on the runway to the rudder (and indirectly the steering wheel) actuation command at a specific airspeed. Equation (3.59) can now be evaluated at several airspeeds to design a gain scheduled controllers. The choice of gain scheduling over feedback linearization was discussed in section 2.1.2 under the "Runway Yaw Rate Controller".



### 3.3 Engine Model

A simple constant power engine model with first order lag dynamics from throttle commands was developed in [1]. This model is developed around the level flight trim condition and is only accurate for small perturbations. The model is summarised in section 1.2.1 and from equation (1.7) it can be seen that the thrust from a constant power model will be infinite at zero airspeed. It is clear that a different engine model is required to simulate the take-off and landing process. For controller design purposes, the engine is simply modelled as having a first order delay from commanded to obtained thrust.

#### 3.3.1 Fixed Pitch Propeller Model

The thrust and moment produced by a fix pitch propeller is given by [12]:

$$F_p = \frac{4}{\pi^2} \rho R_p^4 \Omega^2 C_T \quad (3.60)$$

$$M_p = \frac{4}{\pi^3} \rho R_p^5 \Omega^2 C_p \quad (3.61)$$

$\rho$  is the air density,  $R_p$  is the propeller radius and  $\Omega$  is the propeller rotation speed in radians per second (rad/s).  $C_T$  and  $C_p$  are respectively the coefficients of thrust and power for the propeller. The coefficients of thrust and power are functions of the current advance ratio ( $J$ ) given by [12]:

$$J = \frac{\pi \bar{V}}{\Omega R_p} \quad (3.62)$$

where  $\bar{V}$  is the airspeed in meters per second (m/s). The coefficients of thrust and power versus the advanced ratio used in this project are given in Appendix C. With the coefficients of thrust and power known, only the rotation speed ( $\Omega$ ) is required to calculate the force and moment produced by the propeller. The rotation speed is determined by the engine.

#### 3.3.2 Non-Linear Engine Model

The power delivered by the engine is a function of the rotations per minute (RPM) and the manifold pressure (MAP) [12]. MAP is determined by the current airspeed and throttle setting through the intake model of the engine and is given by [12]:

$$MAP = \left[ \left( \frac{p}{1000} - MAP_{\min} \right) \cdot \delta_T \right] + MAP_{\min} \quad (3.63)$$

$p$  is the air pressure and  $MAP_{\min}$  is the minimum manifold pressure.  $\delta_T$  is the throttle setting between zero and one, where one corresponds to full throttle. The RPM and MAP ranges used in this project are given in Appendix C, as well as the lookup table used to obtain the engine power from the current RPM and MAP.

The torque delivered by the engine is given by [12]:

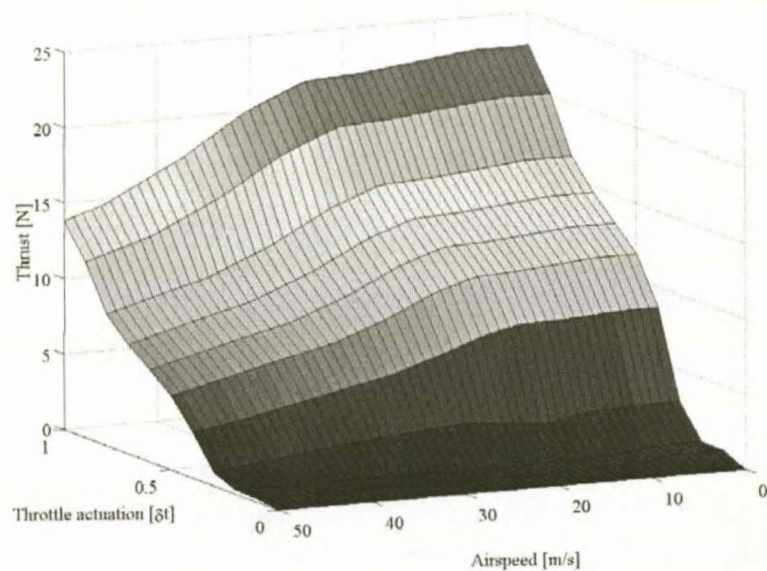
$$T_E = \frac{P_E}{\Omega} \quad (3.64)$$

where  $P_E$  is the power delivered by the engine. The rotation speed of the engine ( $\Omega$ ) is obtained by integrating the angular acceleration of the engine described by [12]:

$$\alpha_E = \dot{\Omega} = \frac{(T_E + M_p)}{(I_E + I_p)} \quad (3.65)$$

$M_p$  is the propeller momentum given by equation (3.61).  $I_E$  is the engine moment of inertia and  $I_p$  is the propeller moment of inertia.  $I_E$  and  $I_p$  are given in Appendix C. With the rotation speed known, the desired force and moment from the propeller is calculated from equations (3.60) and (3.61).

Figure 3.8 shows the output thrust obtained from the engine model developed. The output thrust is shown as a function of the airspeed and throttle actuation.



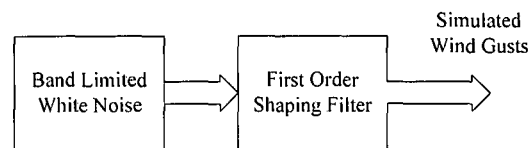
**Figure 3.8:** *Engine Thrust vs. Airspeed and Throttle Actuation*

Note that the coefficients of thrust and power, as well as the RPM and MAP ranges used, are not the actual values for the engine used in this project. They are only typical values for a similar engine obtained from [13]. The engine model is therefore not a precise model of the engine used in this project, but will have very similar characteristics and is well suited for simulation purposes.

### 3.4 Wind Gust Model

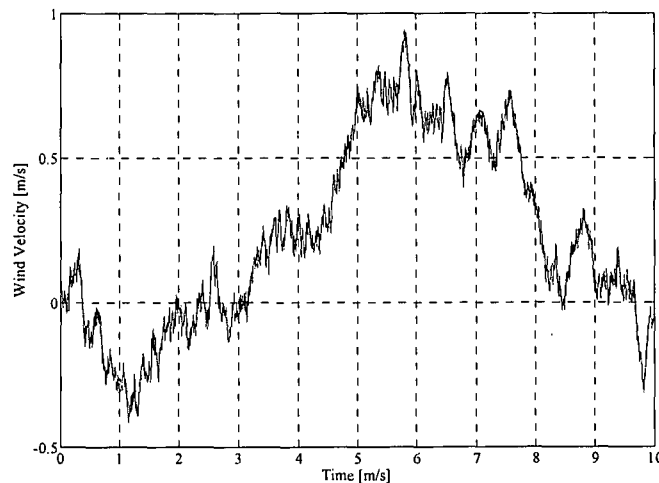
Wind, especially wind gusts, is the main disturbance acting on the aircraft during take-off and landing. It is therefore very important to simulate the effects of wind gusts on the controllers designed in Chapter 4 and Chapter 5. For this purpose a wind gust model is developed.

A simulated wind gust is generated by shaping band limited white noise with a first order filter as shown in Figure 3.9.



**Figure 3.9:** *Wind Gust Model Block Diagram*

The power of the band limited white noise is adjusted until the variance of the simulated wind gust is the desired value. In this project a wind gust variance of 0.8 m/s was chosen, resulting in the height of the power spectral density of the noise source to be 0.5 Watt/Hz. The time constant of the shaping filters determines the duration of a wind gust. A time constant of 2s was found to give good results. The sample time of the band limited white noise is set as high as possible to ensure that the sample time of the noise is much higher than the cut-off frequency of the shaping filter. A wind gust generated by the model is shown in Figure 3.10.



**Figure 3.10:** *Simulated Wind Gust*

A wind shear model to simulate constant wind disturbances were also implemented for simulation purposes.

### 3.5 Summary

In this chapter non-linear models for the runway and engine were developed, as well as a wind gust model. These models are used in the hardware in the loop simulator discussed in Chapter 7. Linear longitudinal and lateral runway models were derived from the non-linear runway model. These linear models will be used to design the runway controllers in Chapter 4.

# Chapter 4

## *Take-Off Control and Simulation*

This chapter presents the detailed design and linear simulation of the take-off controllers. Sections 2.1.1 and 2.1.2 contains the controller choices and the role that each controller plays in the bigger picture. Each controller design is presented by starting with a general description of the controller and the task that it should fulfil. Based on the task, system constraints and the disturbances that should be rejected the design specification are set. The actual design of the controller is then carried out and finally the linear simulation results are presented and discussed. All of the designed feedback gains, with addition information on the controllers, are given in Appendix B.

### 4.1 $X_B$ -Acceleration Controller

The acceleration controller controls the  $X_B$ -axis acceleration of the aircraft with the throttle and forms the first level of control over the longitudinal motion of the aircraft on the runway.

The aircraft used in this project has no active method of braking and relies on friction and drag forces to slow down. The system will therefore recover slowly from any overshoot, resulting in a long settling time and a much degraded response. A low overshoot specification is therefore required for the controller. The low overshoot specifications will also ensure a high phase margin. The high phase margin is desired, since the linear plant model used to design the controller is of limited accuracy.

A slow rise time is necessary to further support the low overshoot requirement. More importantly, it limits the bandwidth of the system. The low cost accelerometers used in this project display poor noise characteristics - engine vibration further elevates the problem. Measurements from the  $X_B$ -axis accelerometer showed a zero mean noise standard deviation of approximately  $0.15 \text{ m/s}^2$ .

Due to the uncertainty that exists in the mapping from the controller commanded thrust (in Newton's) to the throttle position that will generate the thrust, an integral term is required in the compensator. Measuring the throttle position to output thrust mapping is possible but time consuming and the results will depend on atmospheric conditions. It will also result in a controller that is more engine specific.

The controller design approach is therefore to design a general, engine independent but robust (high phase margin) controller. The thrust commands from the controller are linearly mapped over the available actuator range.

The design specifications for the acceleration controller are:

- Overshoot of less than 5%
- Rise time of less than 3s
- Zero steady state error

## Controller Design

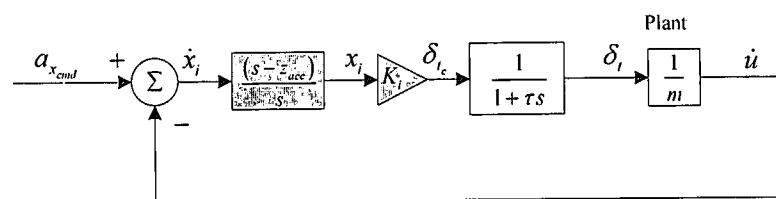


Figure 4.1:  $X_B$  -Acceleration Controller Block Diagram

Figure 4.1 shows the block diagram representation of the acceleration controller. The constant plant dynamics are given by equation (3.46). The throttle displays a significant delay from commanded thrust to output thrust. This can cause instability in the system if not modelled. The engine dynamics is therefore modelled as a first order delay as given by equation (1.8). Note the integral control with  $x_i$  the integrator state. The plant model with the engine lag dynamics included is given by:

$$[\ddot{u}] = \left[ -\frac{1}{\tau_T} \right] [\dot{u}] + \left[ \frac{1}{m\tau_T} \right] [\delta_{t_c}] \quad (4.1)$$

where  $\tau_T$  is the engine's time constant given by equation (F.29).  $m$  is the mass of the aircraft and  $\delta_{t_c}$  is the commanded throttle position. The compensator zero is iteratively placed at  $s = z_{acc} = -2.5$ , resulting in the following compensator:

$$D_{acc}(s) = \frac{(s + 2.5)}{s} \quad (4.2)$$

Figure 4.2 shows the root locus for the variations in the feedback gain  $K_i$  with integral control, with the design specifications superimposed.

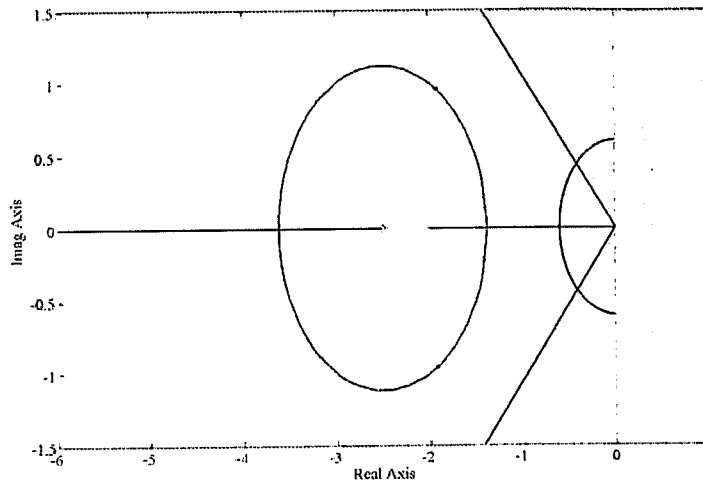


Figure 4.2:  $X_B$ -Acceleration Controller Root Locus

A feedback gain of

$$K_i = 2.2 \quad (4.3)$$

is found to yield the desired dynamic response with all the design specifications met. The closed loop system dynamics are described by:

$$\begin{bmatrix} \dot{\underline{x}} \\ \dot{\underline{x}}_c \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B}K_i\mathbf{C}_c \\ -\mathbf{B}_c\mathbf{C} & \mathbf{A}_c \end{bmatrix} \begin{bmatrix} \underline{x} \\ \underline{x}_c \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{B}_c \end{bmatrix} a_{x_{cmd}} \quad (4.4)$$

$\mathbf{A}_c$ ,  $\mathbf{B}_c$ ,  $\mathbf{C}_c$  and  $\mathbf{D}_c$  are the state space representation matrices and  $\underline{x}_c$  is the state vector of the compensator of equation (4.2).  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{C}$  are the state space representation matrices and  $\underline{x}$  is the state vector of the plant dynamics described by equation (4.1).

### Linear Simulation

Figure 4.3 shows the response of the  $X_B$ -acceleration controller to a  $1m/s^2$  step command in acceleration, as well as the associated thrust command from the controller. Note the slow rise time and the very low overshoot (high phase margin) in the response.



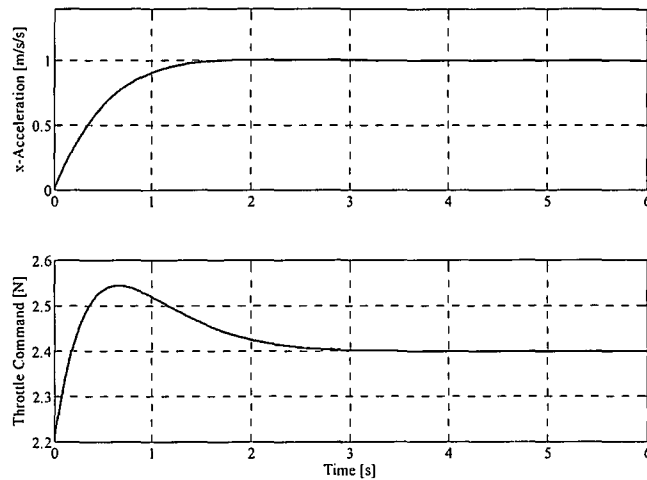


Figure 4.3:  $X_B$ -Acceleration Controller Linear Simulation

## 4.2 Groundspeed Controller

The groundspeed controller forms an outer loop around the  $X_B$ -acceleration controller by generating an acceleration command from the groundspeed error. It is therefore only necessary to stabilise the extra integral state introduced by the natural integrator.

The groundspeed measurement is obtained from the GPS receiver, which produces measurement updates at 4Hz with a single cycle delay of 0.25s [17]. The groundspeed controller is designed directly in the discrete time domain for two reasons: To easily incorporate the measurement delay and because the measurement update rate is only a factor 9 higher than the 0.44Hz bandwidth of the  $X_B$ -acceleration controller (shown in Figure B.5). The measurement update rate should at least be a factor 30 higher than the bandwidth for accurate design by emulation [9].

A slow rise time is necessary for the groundspeed controller for similar reasons given during the design of the  $X_B$ -acceleration controller. The rise time is also required to be slower than that of the  $X_B$ -acceleration controller, as the groundspeed controller forms the outer loop. The overshoot is again required to be low for similar reasons as given in the design of the  $X_B$ -acceleration controller. The phase margin is however slightly relaxed, as the groundspeed controller is being implemented around a well behaved and stable system with all of the aircraft and engine dynamics already encapsulated by the  $X_B$ -acceleration controller.

The natural integrator will ensure that the system can track groundspeed commands with zero steady state error. However, the biases on, or misalignment of the accelerometer used to sense the acceleration will cause a steady state error in groundspeed. If no steady state error is required an integrator term will be required in the compensator. This integral term is not implemented as the steady errors introduced were found to be acceptably small.

The design specifications for the groundspeed controller are:

- Rise time of under 5s
- Overshoot of less than 10%
- Zero steady state error

## Controller Design

Figure 4.4 shows the block diagram representation of the groundspeed controller. Plant dynamics are the closed loop dynamics of the  $X_B$ -acceleration controller given by equation (4.4), with the output matrix ( $C$ ) set to extract the acceleration state. The acceleration passes through a natural integrator to become the groundspeed state  $u$ . The groundspeed is sampled at 4Hz and delayed by a single cycle (0.25s) to become the delayed discreet groundspeed,  $u_D(k)$  that is equivalent to the measurement obtained from the GPS.

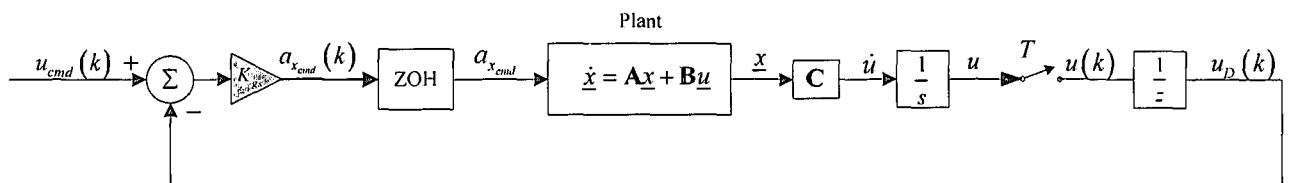


Figure 4.4: Groundspeed Controller Block Diagram

Augmenting the continuous groundspeed state to the plant dynamics of equation (4.4) results in:

$$\begin{bmatrix} \dot{x} \\ \dot{u} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ 1 & \mathbf{0} \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} a_{x_{cmd}} \quad (4.5)$$

$\mathbf{A}$  and  $\mathbf{B}$  are the state space representation matrices and  $\underline{x}$  is the state vector for the plant given by equation (4.4). Discretising the plant given by equation (4.5) at a sample time ( $T$ ) of 0.25s and augmenting the discreet equivalent of the delayed groundspeed state, results in:

$$\begin{bmatrix} \underline{x}(k+1) \\ u_D(k+1) \end{bmatrix} = \begin{bmatrix} \Phi & \mathbf{0} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \underline{x}(k) \\ u_D(k) \end{bmatrix} + \begin{bmatrix} \Gamma \\ 0 \end{bmatrix} a_{x_{cmd}}(k) \quad (4.6)$$

$\Phi$  and  $\Gamma$  are the discreet state space representation matrices and  $\underline{x}(k)$  is the discreet state vector for the system described by equation (4.5).  $u_D(k)$  is the discreet delayed groundspeed state and  $a_{x_{cmd}}(k)$  is the discreet acceleration input.

Figure 4.5 shows the root locus plot for variations in the feedback gain  $K_{gs}$ , with only proportional feedback and the superimposed design specifications.

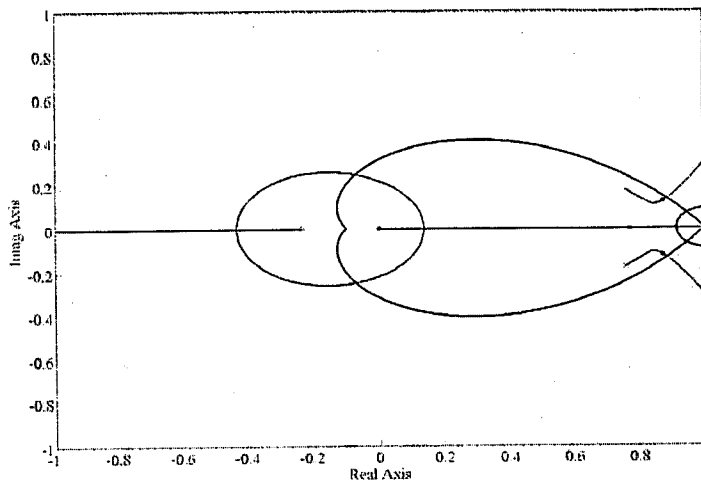


Figure 4.5: Groundspeed Controller Root Locus

A feedback gain of

$$K_{gs} = 0.5 \tag{4.7}$$

is found to yield the desired dynamic response with all the design specification met. The closed loop system dynamics are described by:

$$\underline{x}(k+1) = [\Phi - \Gamma K_{gs} \mathbf{H}] \underline{x}(k) + [\Gamma K_{gs}] u_{cmd}(k) \tag{4.8}$$

$\Phi$  and  $\Gamma$  are the discrete state space representation matrices and  $\underline{x}(k)$  is the discrete state vector for the system described by equation (4.6).

### Linear Simulation

Figure 4.6 shows the response of the groundspeed controller to a 1m/s step command in groundspeed, as well as the associated acceleration command. It is clear from the response that all the design specifications are met.

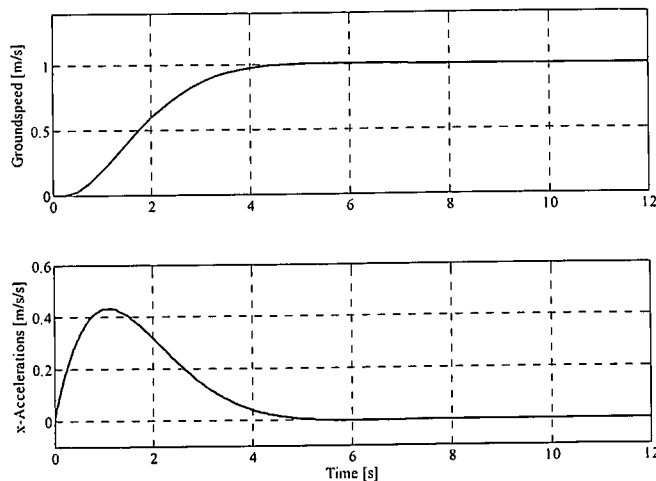


Figure 4.6: Groundspeed Controller Linear Simulation

### 4.3 Pitch Rate Damper

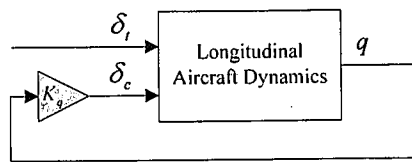


Figure 4.7: Pitch Rate Damper Block Diagram

Figure 4.7 shows the block diagram of the pitch rate damper designed in [1]. A similar pitch rate damper with a larger feedback gain for increased damping is used during the take-off and landing sequences. The feedback gain is increased from -0.02 to

$$K_q = -0.06 \quad (4.9)$$

The closed loop dynamics for the system is described by:

$$\dot{\underline{x}} = \left[ \mathbf{A} - \mathbf{B} \begin{bmatrix} K_q & 0 \end{bmatrix}^T \right] \underline{x} + \mathbf{B}u \quad (4.10)$$

where  $\mathbf{A}$  is the dynamics matrix and  $\mathbf{B}$  the input matrix for the system described by equation (F.23).  $\underline{x}$  is the state vector given by equation (F.27).

### 4.4 Climb Rate Controller

The climb rate controller will be used during flaring and rotation where any excessive overshoot is undesirable. An excessive overshoot can stall the aircraft during take-off or cause damage to the undercarriage during landing.

It is best to obtain the fastest possible rise time from the controller within the physical limitations of the aircraft and without excessively increasing the bandwidth of the system.

The design specifications for the climb rate controller are:

- Overshoot of less than 10%
- Rise time of under 2s

#### Controller Design

Figure 4.8 shows the block diagram for the climb rate controller. Equation (F.23) describes the plant dynamics and the output matrix  $\mathbf{C}$  is set to extract the climb rate in meters per second (m/s). The aerodynamic model is linearised about a trim velocity of  $14\text{m/s}$  and  $K_q$  in the plant dynamics is given by equation (4.9).

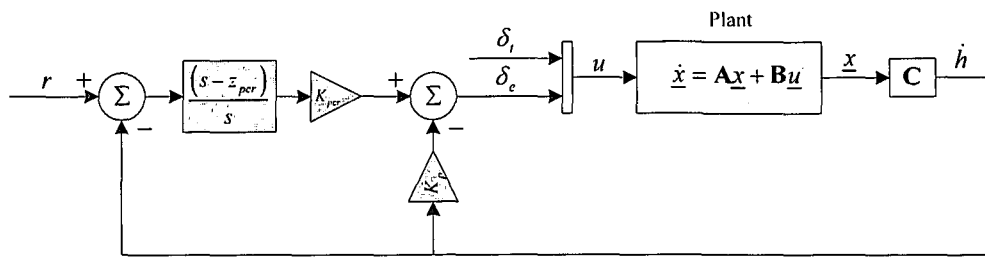


Figure 4.8: Climb Rate Controller Block Diagram

After closing a proportional loop around the plant, the system has very little DC gain. This leads to poor steady state command following. Simply increasing the gain of the proportional controller is not sufficient and degrades the dynamic response. An integral term is added to provide the infinite DC gain for a zero steady state error. However, adding the integrator causes the phugoid mode poles to easily go unstable. The phugoid poles are pulled back into the left half plane by adding a compensator zero and forming a PI controller (lead compensator). The dynamic response improves dramatically after the zero has been placed, but the response suffers from excessive overshoot due to a plant zero close to the origin ( $s = -0.111$ ). The overshoot is reduced by directly feeding back the climb rate measurement to the elevator. If the climb rate therefore increases excessively, the elevator command will be reduced. The slow plant zero keeps the integrator pole close to the origin, which results in a dynamic response that very slowly approaches the commanded steady state value. Nevertheless, this is not a problem, as only the first ten seconds of the response are of interest, because the controller will only be used for short periods during rotation and flaring. The compensator zero is placed at  $s = z_{per} = -2.5$ , resulting in the following compensator:

$$D_{per}(s) = \frac{(s + 2.5)}{s} \quad (4.11)$$

A climb rate feedback gain of

$$K_p = -0.9 \quad (4.12)$$

is found to sufficiently damp the excessive overshoot of the system. With the compensator and climb rate feedback in place, the root locus for the variation in compensator gain ( $K_{per}$ ) is shown in Figure 4.9. The design specifications are superimposed. Note the troublesome pole and zero close to the origin.

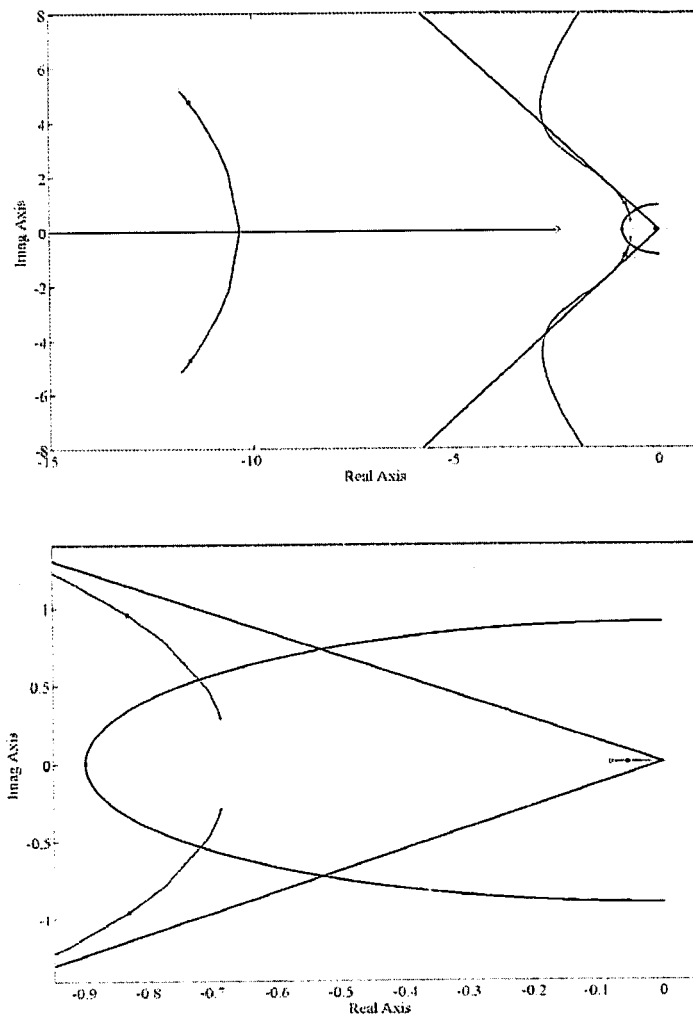


Figure 4.9: Climb Rate Controller Root Locus (Top) with Origin Magnification (Bottom)

A compensator gain of

$$K_{pcr} = -0.3 \quad (4.13)$$

is found to produce the desired dynamic response with all the design specifications met. The closed loop system dynamics are given by:

$$\begin{bmatrix} \dot{\underline{x}} \\ \dot{\underline{x}}_c \end{bmatrix} = \begin{bmatrix} \mathbf{A} + \mathbf{BK}_p\mathbf{C} - \mathbf{BK}_{pcr}\mathbf{D}_c\mathbf{C} & \mathbf{BK}_{pcr}\mathbf{C}_c \\ -\mathbf{B}_c\mathbf{C} & \mathbf{A}_c \end{bmatrix} \begin{bmatrix} \underline{x} \\ \underline{x}_c \end{bmatrix} + \begin{bmatrix} \mathbf{BK}_{pcr}\mathbf{D}_c \\ \mathbf{B}_c \end{bmatrix} r \quad (4.14)$$

$\mathbf{A}_c$ ,  $\mathbf{B}_c$ ,  $\mathbf{C}_c$  and  $\mathbf{D}_c$  are the state space representation matrices and  $\underline{x}_c$  is the state vector of the compensator of equation (4.11).  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{C}$  are the state space representation matrices and  $\underline{x}$  is the state vector of the plant dynamics described by equation (4.10).



## Linear Simulation

The response of the climb rate controller to a  $1\text{ m/s}$  step command in climb rate is shown in Figure 4.10. Note the offset from the commanded value, but the good dynamic response during the time frame of interest (first ten seconds).

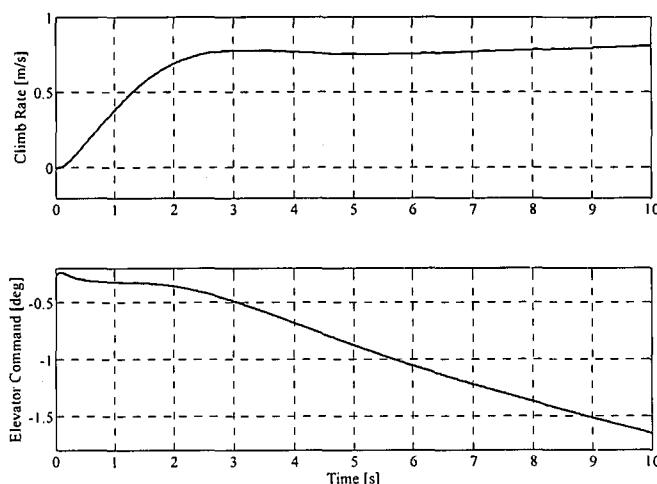


Figure 4.10: Climb Rate Controller Linear Simulation

## 4.5 Runway Yaw Rate Controller

The runway yaw rate controller forms the first level of lateral control on the runway. Since the non-linear lateral runway model reduces to a first order system when linearised at a specific airspeed (linearised in section 3.2.2), no overshoot specification is required. The rise time is chosen to make the aircraft respond as quickly as possible to a yaw rate step, within the physical limitations of the aircraft and without saturating the nose wheel or the rudder. A fast response time is desired for maximum disturbance (wind gust) rejection during take-off and will allow for fast outer-loop controllers. The upper bound on the rise time is enforced by the susceptibility of the controller to the high frequency noise present on the low cost gyroscopes.

An integral term is essential in the compensator to negate effects of the aircraft's asymmetry and non-linearity's in the nose wheel actuator. The integrator will also serve to counter propeller torque and helical slipstream effects (discussed in section 2.1.2), as well as constant wind disturbances. It is also desired to command a certain yaw rate with zero steady state error.

The design specifications for the runway yaw rate controller are:

- Rise time of under 2s
- Zero steady state error
- To follow yaw rate commands of up to  $\pm 30\text{ deg/s}$

Actuator limits are the main limiting factor on the response of the runway yaw rate controller. The limits for the steering wheel and rudder actuators to ensure a linear actuation model are:

- Maximum steering wheel deflection of 10° from the zero position
- Maximum rudder deflection of 20° from the zero position

### Controller Design

The yaw rate controller is designed using a full state feedback architecture and the “Place” function in MATLAB. This allows a script to be written that can automate the gain scheduling design over the desired airspeed range. Yaw rate feedback with a root locus design, followed by emulation, will require several gains to be selected manually.

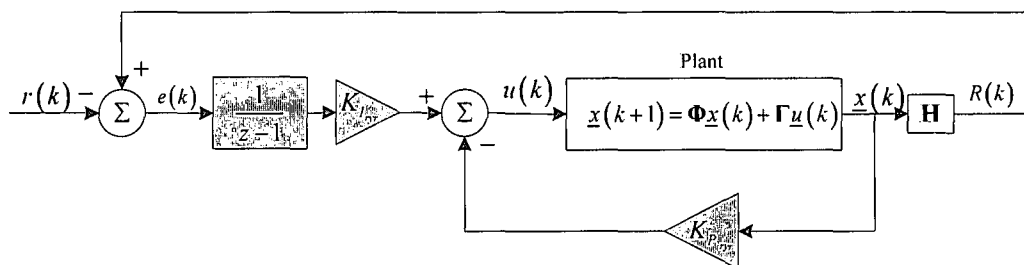


Figure 4.11: Runway Yaw Rate Controller Block Diagram

Consider the block diagram for the runway yaw rate controller in Figure 4.11 where the plant model is the discrete equivalent of equation (3.59) in state space form. The discrete output matrix ( $\mathbf{H}$ ) is set to extract the yaw rate in rad/s. The integrator dynamics are described by:

$$x_i(k+1) = x_i(k) + e(k) = x_i(k) + \mathbf{H}x(k) - r(k) \tag{4.15}$$

where  $x_i$  is the integration error state and  $e(k)$  is the yaw rate error.  $r(k)$  is the reference input yaw rate command. The plant model augmented by the integrator dynamics are:

$$\begin{bmatrix} \underline{x}(k+1) \\ x_i(k+1) \end{bmatrix} = \begin{bmatrix} \Phi & 0 \\ \mathbf{H} & 1 \end{bmatrix} \begin{bmatrix} \underline{x}(k) \\ x_i(k) \end{bmatrix} + \begin{bmatrix} \Gamma \\ 0 \end{bmatrix} u(k) - \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(k) \tag{4.16}$$

$\Phi$ ,  $\Gamma$  and  $\mathbf{H}$  are the discrete scalar state space representation matrices for the plant given by equation (3.59) discretised at 25Hz.

The control law can be written as:

$$u(k) = - \begin{bmatrix} K_{p_{rr}} & K_{i_{rr}} \end{bmatrix} \begin{bmatrix} \underline{x}(k) \\ x_i(k) \end{bmatrix} \tag{4.17}$$

where  $K_{p_{rr}}$  and  $K_{i_{rr}}$  are the proportional and integrated error feedback gains respectively. The integrator pole is placed at  $z = 0.8$  (fast enough not to interfere with the system response) and the system pole at the location

resulting in the desired rise time of 2s. The values for  $K_{p_{rr}}$  and  $K_{i_{rr}}$  over an airspeed range of 20m/s at 1m/s intervals are given in Table B.4. The dynamics of the closed loop system is given by:

$$\begin{bmatrix} \underline{x}(k+1) \\ x_r(k+1) \end{bmatrix} = \begin{bmatrix} \Phi - \Gamma K_{p_{rr}} & -\Gamma K_{i_{rr}} \\ \mathbf{H} & 1 \end{bmatrix} \begin{bmatrix} \underline{x}(k) \\ x_r(k) \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} r(k) \tag{4.18}$$

**Linear Simulation**

The linear simulation result to a step command of 30 deg/s in yaw rate at airspeeds of 2m/s, 5m/s and 8m/s are shown in Figure 4.12. It is clear from the response that all the design specifications are met. Neither the steering wheel nor the rudder were saturated during the response. The effect of gain scheduling to produce similar yaw rate responses at any airspeed within the design range is clear.

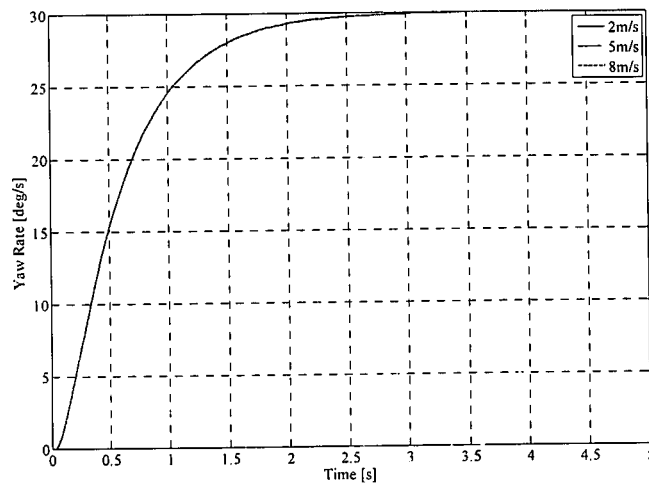


Figure 4.12: Runway Yaw Rate Controller Linear Simulation

**4.6 Turn Radius Interface**

The radius interface provides a yaw rate command to the runway yaw rate controller based on the current groundspeed measurement and radius command.

**Controller Design**

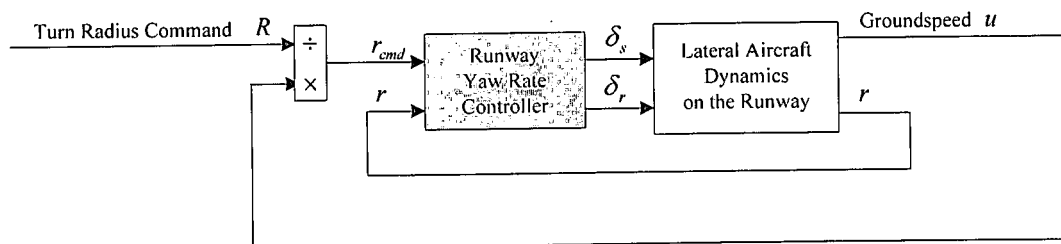


Figure 4.13: Turn Radius Interface Block Diagram

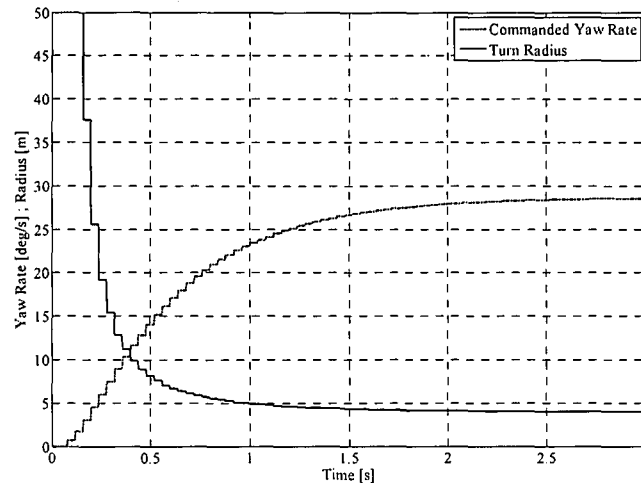
A block diagram for the turn radius interface is illustrated in Figure 4.13.

Conversion of the radius command to a yaw rate command for the runway yaw rate controller is achieved by the following relationship:

$$r = \frac{U_B}{R} \quad (4.19)$$

$U_B$  is the body axes axial velocity (groundspeed) and  $R$  is the desired turn radius.

### Linear Simulation



**Figure 4.14:** *Turn Radius Linear Simulation*

Figure 4.14 shows the turn radius response and the associated yaw rate to a 4m turn radius command at 2m/s. Note the infinite turn radius at zero yaw rate.

## 4.7 Runway Heading Controller

The runway heading controller is the ground counterpart of the heading controller designed in [1]. The controller generates a yaw rate command for the runway yaw rate controller from a heading error. This means that only the extra natural integrator due to the heading state needs to be stabilised by the feedback. The aircraft and runway dynamics are already incorporated and stabilised by the runway yaw rate controller. The heading measurement, used to form the heading error, is obtained from the GPS receiver. Similar to the groundspeed controller design, the runway heading controller is designed directly in the discrete time domain to easily incorporate the single cycle measurement delay of the GPS. The measurement update rate is also only a factor 14 higher than the 0.28Hz bandwidth of the runway yaw rate controller (shown in Figure B.6). It falls short of the factor 30 required for accurate design by emulation [9].

The overshoot is not critical in the controller design, but a low overshoot will ensure a high phase margin and more importantly, a dynamic response suitable for control in a confined area such as a runway with limited width.

The rise time should be as fast as possible to ensure the maximum rejection of disturbances and allow for the fastest possible guidance controller. The upper bound of the rise time is enforced by measurement noise

requiring a limited bandwidth. The rise time should also preferably be slower than the rise time of the runway yaw rate controller.

The natural integrator introduced by the heading state will ensure that the runway heading controller can track a command with zero steady state error. However, if a bias is present on the yaw rate gyroscope the heading controller will no longer track a command with zero steady state error, since the runway yaw rate controller will require a constant command input to oppose the bias. A compensator integral term is therefore required in the runway heading controller to provide the constant input necessary to appose the bias and maintain a zero steady state error. For a runway length of 100m and a width of 10m, the maximum heading error that will ensure that the aircraft stays on the runway for the entire runway length is  $2.86^\circ$ . Simulation results in Figure 4.17 showed that a yaw rate gyroscope bias of  $0.5\text{deg/s}$  (typical value for the sensor used in this project) results in a steady state heading error of  $1^\circ$ . The steady state heading error due to a gyroscope bias is therefore acceptable and the integral term is not added to the compensator.

The runway heading controller design specifications are:

- Overshoot of less than 10%
- Rise time of under 4s
- Zero steady state error

### Controller Design

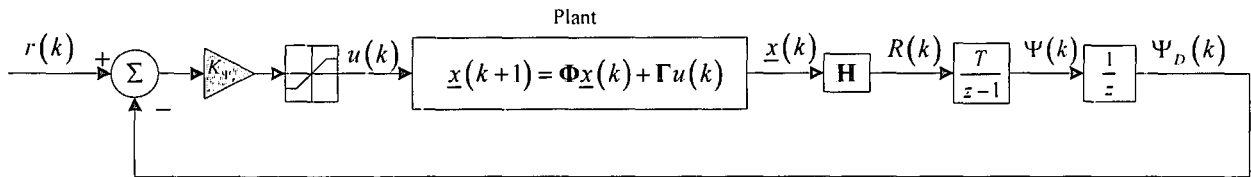


Figure 4.15: Runway Heading Controller Block Diagram

Figure 4.15 shows the block diagram for the heading controller. The plant dynamics are given by equation (4.18), with the discrete output matrix ( $\mathbf{H}$ ) set to extract the yaw rate state in rad/s. The yaw rate passes through the discrete equivalent of the natural integrator to become the discrete heading state in radians (rad). The discrete heading state then passes through a single cycle delay block to become the discrete delayed heading measurement ( $\Psi_D(k)$ ). Note that the yaw rate command is passed through a saturation block to ensure it does not exceed the design limits of  $\pm 30$  deg/s of the runway yaw rate controller.

Augmenting the closed loop plant dynamics with the discrete heading and delayed heading states, results in the following open loop plant model for the heading controller:

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \Psi(k+1) \\ \Psi_D(k+1) \end{bmatrix} = \begin{bmatrix} \Phi & \mathbf{0} & \mathbf{0} \\ T\mathbf{H} & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \Psi(k) \\ \Psi_D(k) \end{bmatrix} + \begin{bmatrix} \Gamma \\ 0 \\ 0 \end{bmatrix} u(k) \quad (4.20)$$

$\Phi$ ,  $\Gamma$  and  $\mathbf{H}$  are the discrete state space representation matrices for the plant given by equation (4.18).  $T$  is the sample time (GPS update rate) of 0.25s,  $\Psi(k)$  is the discrete heading state,  $\Psi_D(k)$  is the delayed discrete heading state and  $u(k)$  is the discrete yaw rate command input.

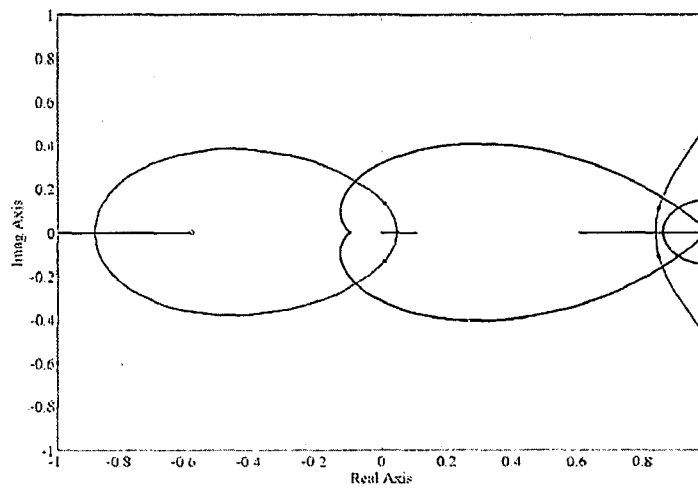


Figure 4.16: Runway Heading Controller Root Locus

Figure 4.16 shows the root locus for a variation in the feedback gain  $K_\psi$ , with the design specifications superimposed. All the design specifications are met with a proportional gain of:

$$K_\psi = 0.4 \quad (4.21)$$

The closed loop system dynamics are described by:

$$\underline{x}(k+1) = [\Phi - \Gamma K_\psi, \mathbf{H}] \underline{x}(k) + [\Gamma K_\psi, ] r(k) \quad (4.22)$$

$\Phi$ ,  $\Gamma$  and  $\mathbf{H}$  are the discrete state space representation matrices and  $\underline{x}(k)$  is the discrete state vector for the system described by equation (4.20).  $r(k)$  is the commanded discrete heading input.

### Linear Simulation

Figure 4.17 and Figure 4.18 shows the response of the runway heading controller to 45° and 270° step commands in heading at a groundspeed of 2m/s. Figure 4.17 shows the response to a 45° step command that does not saturate the yaw rate command to the runway yaw rate controller. Also shown in Figure 4.17, is the same response but with a yaw rate gyroscope bias of 0.5deg/s that result in a steady state heading error of 1°.

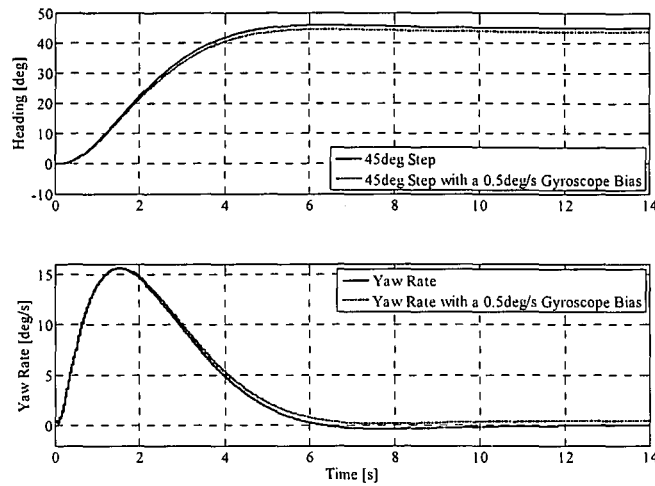


Figure 4.17: Runway Heading Controller Linear Simulations

Figure 4.18 shows the response of the runway heading controller to a  $270^\circ$  heading command that saturates the yaw rate command to the runway yaw rate controller. Note that the aircraft enters a constant turn for the majority of the response.

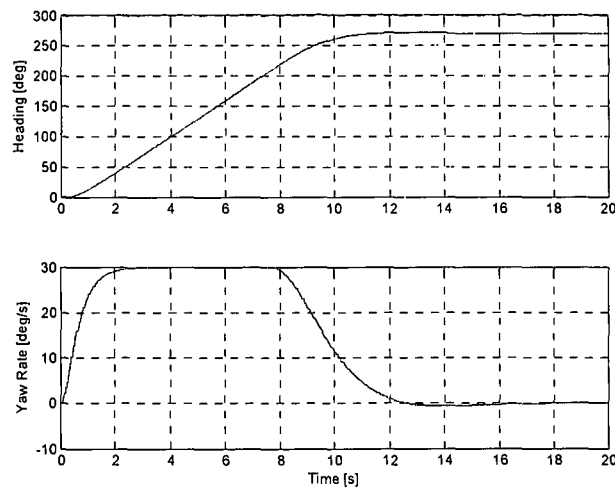


Figure 4.18: Runway Heading Controller Linear Simulation - Saturated Response

## 4.8 Runway Guidance Controller

The runway guidance controller is the final controller designed for use on the runway. It generates a heading command from the cross track error measurement obtained from the GPS. This means that only the extra integrator, due to the cross track error state, needs to be stabilised. Details cross track error calculation were given in section 2.3. The runway guidance controller is also designed directly in the discrete time domain for the same reasons stated during the design of the runway heading controller.

The cross track error is dependant on the forward velocity of the aircraft. It is therefore necessary to introduce a gain scheduled design approach for the runway guidance controller to remove the velocity dependency from the system.



The overshoot design arguments for the guidance controller are similar to the ones stated during the design of the runway heading controller. The rise time of the controller is not critical, since the runway guidance controller serves as a slower outer loop.

The natural integrator introduced by the cross track position state will ensure a zero steady state error. However, a bias on the yaw rate gyroscope will however result in a steady state heading error and consequently a steady state position error for the same reasons as explained during the design of the runway heading controller. To maintain a zero steady state error, a compensator integral term is required in the runway guidance controller. A 0.5deg/s bias on the yaw rate gyroscope causes a cross track position error of just over 1m (refer to simulation results in Figure 4.21). This position offset is acceptable for use on a 10m wide runway and the compensator integrator is therefore not implemented.

The design specifications for the runway guidance controller are:

- Overshoot of less than 10%
- Rise time of under 10s
- Zero steady state error

## Controller Design

Figure 4.19 shows the block diagram for the runway guidance controller. Equation (4.22) gives the plant dynamics with the discrete output matrix ( $\mathbf{H}$ ) set to extract the cross track velocity in meters per second (m/s). The cross track velocity then passes through the discrete equivalent of the natural integrator to become the discrete cross track position state in meters. The cross track position is delayed a single cycle to become the discrete delayed cross track position measurement ( $y_D(k)$ ).

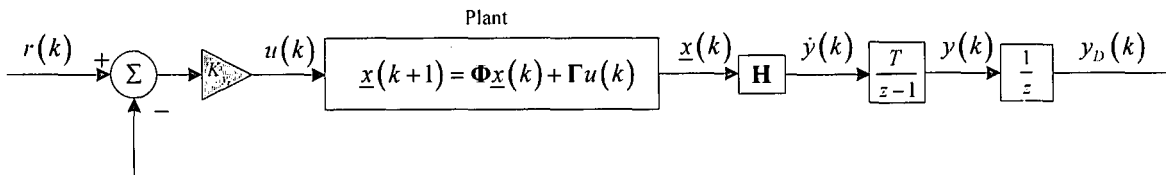


Figure 4.19: Runway Guidance Controller Block Diagram

Augmenting the plant dynamics with the discrete cross track error and delayed cross track error states, results in the following open loop plant model:

$$\begin{bmatrix} \underline{x}(k+1) \\ y(k+1) \\ y_D(k+1) \end{bmatrix} = \begin{bmatrix} \Phi & \mathbf{0} & \mathbf{0} \\ T\mathbf{H} & 1 & 0 \\ \mathbf{0} & 1 & 0 \end{bmatrix} \begin{bmatrix} \underline{x}(k) \\ y(k) \\ y_D(k) \end{bmatrix} + \begin{bmatrix} \Gamma \\ \mathbf{0} \\ 0 \end{bmatrix} u(k) \quad (4.23)$$

$\Phi$ ,  $\Gamma$  and  $\mathbf{H}$  are the discrete state space representation matrices and  $\underline{x}(k)$  is the discrete state vector for the system described by equation (4.22).  $T$  is the sample time (GPS update rate) of 0.25s,  $y(k)$  is the discrete cross track error state,  $y_D(k)$  is the delayed discrete cross track error state and  $u(k)$  is the discrete heading command input.

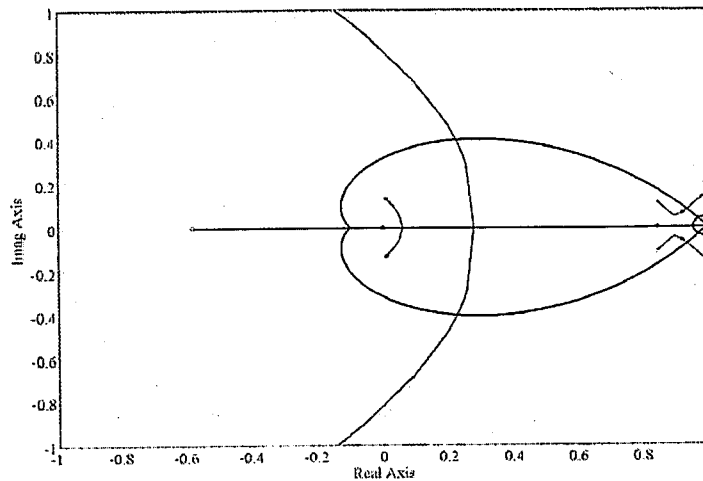


Figure 4.20: Runway Guidance Controller Root Locus at 4m/s

Figure 4.16 shows the root locus for a variation in the feedback gain  $K_{y_r}$ , with the design specifications superimposed. The values of  $K_{y_r}$  over a 12m/s airspeed range at 1m/s intervals are given in Table B.4. The closed loop system can then be written as:

$$\underline{x}(k+1) = [\Phi - \Gamma K_{y_r} \mathbf{C}] \underline{x}(k) + [\Gamma K_{y_r}] r(k) \quad (4.24)$$

$\Phi$  and  $\Gamma$  are the discrete state space representation matrices and  $\underline{x}(k)$  is the discrete state vector for the system described by equation (4.23).

### Linear Simulation

The top half of Figure 4.21 shows the response of the runway guidance controller to a 20m step command in cross track position at different groundspeeds.

Note however that the response does not meet the design specifications for groundspeeds less than 2m/s. This is due to the fact that the feedback gains required for meeting the design specifications grow very rapidly as the groundspeed decreases. This will result in large gain jumps as the controller schedules that will cause undesired jumps in the response. Therefore, the feedback gains are limited to smaller values to ensure a smooth response at the cost of dynamic performance.

The bottom graph shows the response to a 20m step command in cross track position at 4m/s and the cross track position error due to a 0.5deg/s bias on the yaw rate gyro. The cross track position error is just over a meter, which is acceptable for use on a 10m wide runway.

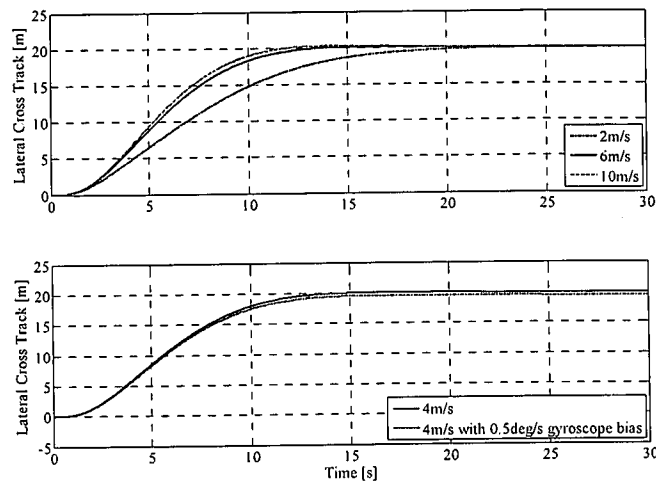


Figure 4.21: Runway Guidance Controller Linear Simulation

## 4.9 Summary

This chapter presented the design of all the controllers that form the building blocks of the take-off sequence discussed in section 2.1.1. A robust design approach was followed for the runway controllers to ensure stability throughout the very non-linear operating range on the runway and to compensate for modelling uncertainties. All the controller designs were verified by linear simulation with very satisfactory results.

# Chapter 5

## *Landing Control and Simulation*

This chapter presents the detail design and linear simulation of the controllers required during landing. Sections 2.2.1 and 2.2.2 discussed the role that each controller plays in the bigger picture. The chapter starts by revisiting the conventional flight controllers to extend their usable airspeed range and improve on some of the responses and characteristics required for landing. The glide path controller follows and finally the flare and de-crab controller designs are discussed. Linear simulation results for all the controller designs are also presented. All of the designed feedback gains, with addition information on the controllers, are given in Appendix B.

### 5.1 Conventional Flight Controllers

#### 5.1.1 Gain Scheduling

Due to the successive loop closure design mythology implemented, it is only necessary to gain schedule the inner control loops responsible for regulating the attitude of the aircraft. The outer guidance loops will interact with a similar system at any trim condition because of the scheduling and will therefore remain unchanged. However, the cross track error used in the guidance controller is dependent on the forward velocity (trim airspeed), as can be seen from equation (F.34). The outer loop guidance controller will thus also require scheduling. In addition to the inner longitudinal attitude regulating airspeed and climb rate controller, the outer altitude controller will also be gain scheduled. This is done to ensure that the altitude response is as good as possible during landing.

The four conventional flight controllers that are gain scheduled are:

- The airspeed and climb rate controller
- The altitude controller
- The yaw orientation control system
- The guidance (cross track error) controller

## Airspeed and Climb Rate Controller

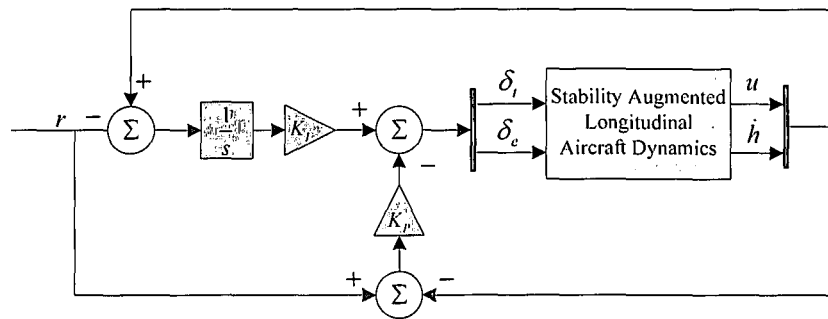


Figure 5.1: Airspeed and Climb Rate Controller Block Diagram

Figure 5.1 shows the block diagram for the airspeed and climb rate controller. The Multi-Input-Multi-Output (MIMO) airspeed and climb rate controller was designed in [1], using an intuitive, graphical approach. This involves an iterative method. The current closed loop poles, along with a set of root loci corresponding to the change in each element of the feedback gain matrices ( $K_p$  and  $K_I$ ), are plotted on the complex frequency plane.

The iterative process is summarised below:

1. Plot the poles of the current closed loop system.
2. Vary the feedback gains of  $K_p$  and  $K_I$ , 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.
5. 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”.

Figure 5.2 shows the resulting multi root locus plots for trim conditions with trim airspeeds ranging from  $14m/s$  to  $20m/s$  at  $2m/s$  intervals. The corresponding gain matrices are given in Table B.2. The airspeed range is only airspeeds obtained during landing and not the entire flight envelope due to the difficult design process of the airspeed and climb rate controller. The interval size of  $2m/s$  provides a good balance between successive gain increments and the number of design iterations required to cover the desired airspeed range.

“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 each root locus of Figure 5.2 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”.

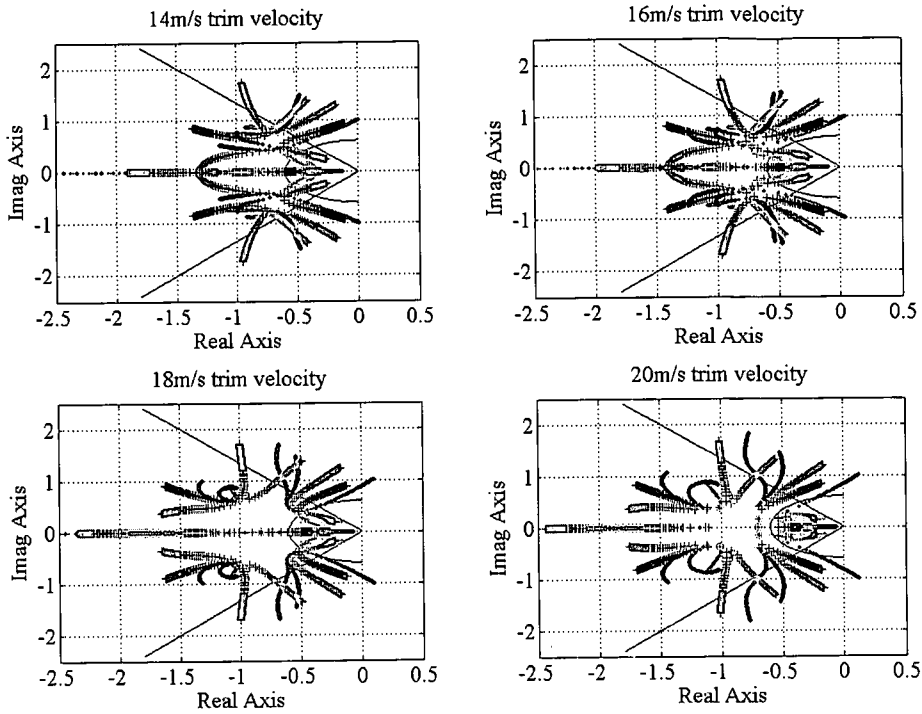


Figure 5.2: Gain Scheduled Airspeed and Climb Rate Multi Root Locus Plots

Figure 5.3 shows the linear simulation results for the gain scheduled airspeed and climb rate controller. The left two graphs in Figure 5.3 show the response to a 5m/s step in airspeed, along with the consequent climb rate response at four different trim airspeeds. The two graphs on the right hand side of Figure 5.3 show the response to a 2m/s step command in climb rate, along with the consequent airspeed response at the same four trim airspeeds.

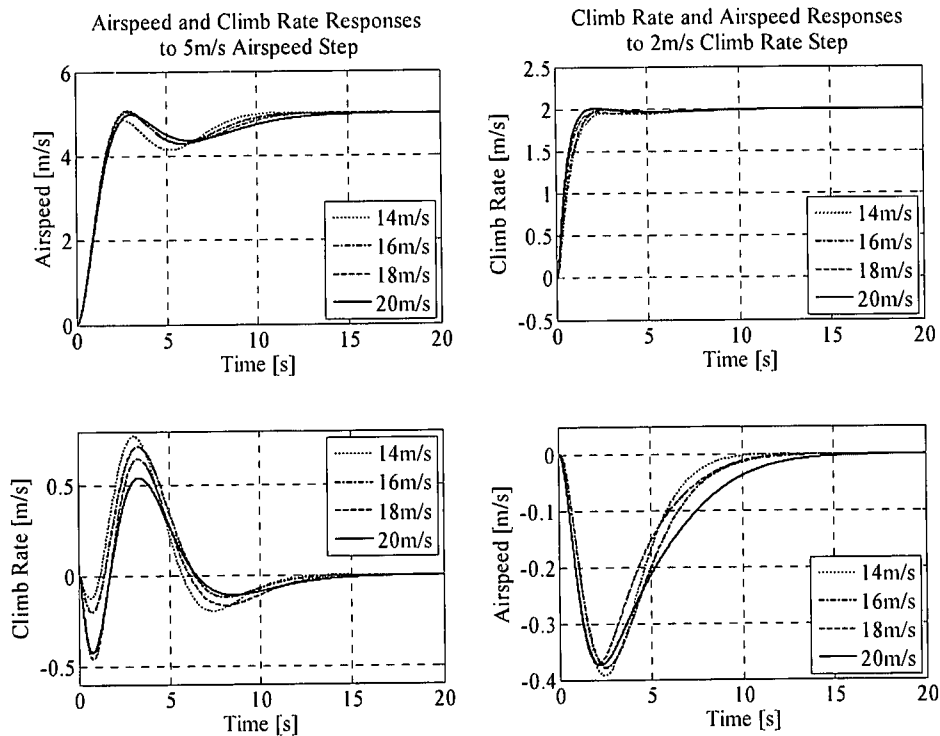


Figure 5.3: Gain Scheduled Airspeed and Climb Rate Controller Linear Simulation

## Altitude, Yaw Orientation and Guidance Controllers

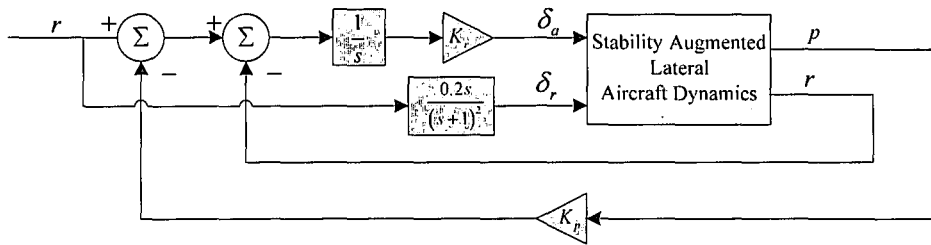


Figure 5.4: Yaw Orientation Controller Block Diagram

The yaw orientation control system is scheduled by keeping the yaw rate feedback gain fixed and only scheduling the roll rate feedback gain. Figure 5.4 shows the block diagram of the yaw orientation control system.

Table B.1 contains the scheduled gains for the altitude, yaw orientation and guidance controllers. The gains are for trim conditions with trim airspeeds ranging from  $10\text{m/s}$  to  $32\text{m/s}$  at  $2\text{m/s}$  intervals. The airspeed range is selected to cover the entire flight envelope of the aircraft used in this project. The interval size of  $2\text{m/s}$  provides a good balance between successive gain increments and the number of design iterations required to cover the desired airspeed range. The block diagrams for the altitude and guidance controllers are shown in Figure B.2 and Figure B.4 respectively. Figure 5.5 shows the linear simulation results for the altitude, yaw rate and guidance controllers at airspeeds of  $12\text{m/s}$ ,  $20\text{m/s}$  (the original response) and  $28\text{m/s}$ .

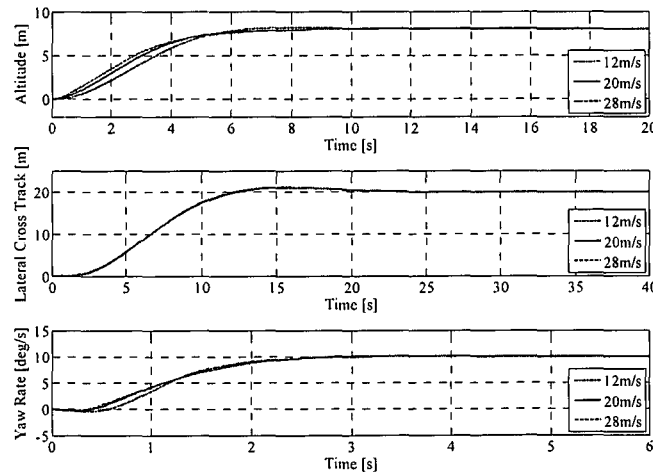


Figure 5.5: Gain Scheduled Altitude, Yaw Rate and Guidance Controllers Linear Simulation

### 5.1.2 Dutch Roll Damper

The dutch roll damper designed in [1] was found to be unstable during practical flight tests. The instability is caused by modelling errors in the rudder's effectiveness at the trim airspeed and can be removed by simply decreasing the feedback gain. The dutch roll damper was not critical for autonomous navigation, but will play a more significant role during autonomous landing due to its ability to reject wind gust disturbances. The redesign of a stable dutch roll damper is therefore necessary.



## Controller Design

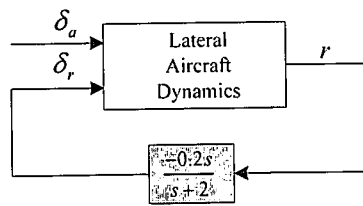


Figure 5.6: Dutch Roll Damper Block Diagram

The dutch roll damper is implemented by feeding back the yaw rate signal through a high pass filter to the rudder, as shown in the block diagram in Figure 5.6. “The yaw rate signal is high passed filtered to prevent the damper from countering constant turn motion and will negate any bias on the rate gyroscope”.

The cut-off frequency selection and design mythology was discussed in [1] and is left unchanged. Slightly decreasing the gain, and consequently the damping of the dutch roll damper, produces a stable dutch roll damper. The new dutch roll damper washout filter is given by:

$$D_w(s) = \frac{-0.08s}{s+2} \quad (5.1)$$

The root locus plot and linear simulation results for the new dutch roll damper are very similar to the ones obtained in [1] and therefore not repeated.

The closed loop dynamics for the system in Figure 5.6 are described by,

$$\begin{bmatrix} \dot{x} \\ \dot{x}_w \end{bmatrix} = \begin{bmatrix} \mathbf{A} - \mathbf{B}D_w\mathbf{C} & -\mathbf{B}C_w \\ B_w\mathbf{C} & A_w \end{bmatrix} \begin{bmatrix} x \\ x_w \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} u \quad (5.2)$$

where  $\mathbf{A}$  is the dynamics matrix and  $\mathbf{B}$  the input matrix for the system described by equation (F.32) and  $\underline{x}$  is the state vector given by equation (F.35).

### 5.1.3 Guidance Controller

It was mentioned in the discussion on landing controllers in section 2.2.2 that the airborne guidance controller is susceptible to gyroscope biases that are typical on the low cost sensors used in this project. These biases, with the controllers designed in [1], can cause steady state errors in lateral position of up to 6m.

One solution to remove the steady state errors is to introduce an integrator term in the heading controller to push out the steady state errors of both the heading and guidance controllers. This solution introduces additional heading controller dynamics that complicates the design. The controllers have also already been practically demonstrated. For these reasons it was deemed best to leave the design unaltered. The alternative is to allow for steady state errors, but to increase the loop gains of the heading and guidance controllers without sacrificing too much of the dynamic response characteristics until the steady state errors are acceptable. This solution was implemented.

### Controller Design

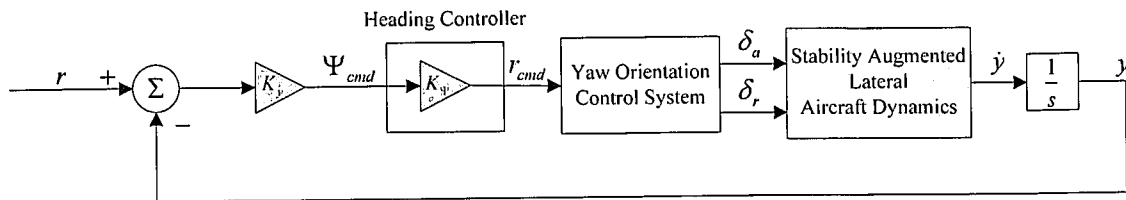


Figure 5.7: Heading and Guidance Controller Block Diagram

Figure 5.7 shows the block diagram for the heading and guidance controllers as designed in [1]. Increasing the controller gain degrades the controller dynamic response. This places an upper limit on the gain. The heading controller feedback gain is increased to:

$$K_{\psi} = 0.4 \tag{5.3}$$

The increased feedback gains for the guidance controller are given in Table B.3.

### Linear Simulation

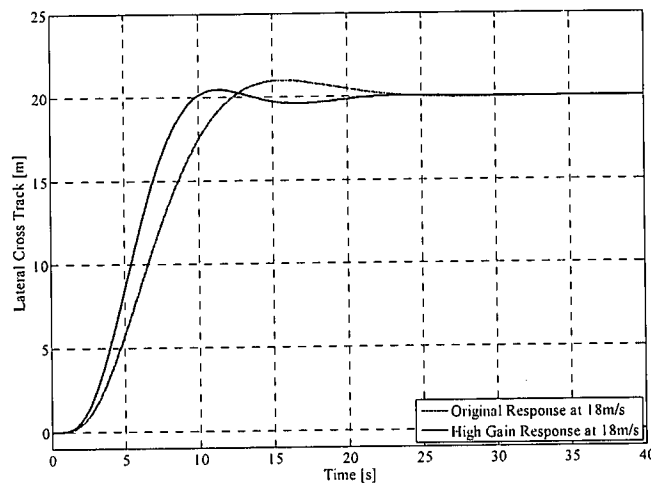


Figure 5.8: High-Gain and Original Guidance Controllers Linear Simulation

Figure 5.8 shows the step response to a 20m step command in track position for the original and high-gain guidance controllers. Note that the high-gain guidance controller has a faster rise time due to the increased gain, with slightly less overshoot but more undershoot than the original guidance controller. However, the settling time of the two designs are very similar. The effect of the increased gain on the steady state error in cross track position is illustrated in Figure 5.9. The responses were obtained with biases of 0.5deg/s on both the yaw rate and roll rate gyroscopes.

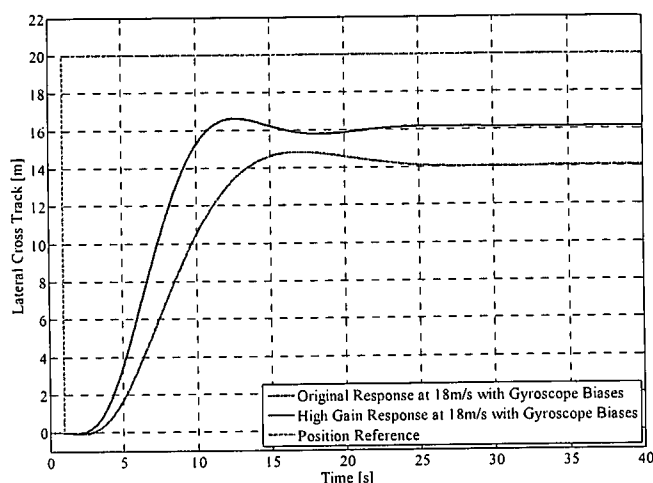


Figure 5.9: High-Gain and Original Guidance Controllers Linear Simulation with Gyroscope Biases

The original guidance controller has a steady state error of just less than 6m, whereas the high-gain guidance controller has a steady error of just less than 4m.

## 5.2 Glide Path Controller

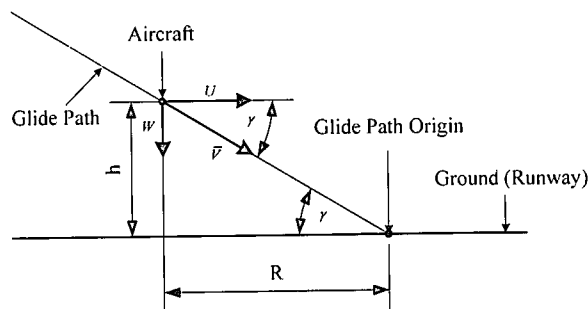


Figure 5.10: Glide Path Graphical Description

Figure 5.10 shows a graphical description of the glide path that the aircraft is required to fly during the landing approach.  $\gamma$  is the glide path angle.  $R$  is the range of the aircraft from the glide path origin point and  $h$  is the altitude. The range and altitude are respectively given by:

$$R = \sqrt{P_N^2 + P_E^2} \tag{5.4}$$

$$h = R \cdot \text{Tan}(\gamma) \tag{5.5}$$

$P_N$  is the north displacement and  $P_E$  the east displacement of the aircraft in inertial space.

### Controller Design

Flying the glide path requires the aircraft to maintain a constant sink rate. With reference to Figure 5.10, this sink rate can be obtained by differentiating equation (5.5) and is given by:

$$\dot{h} = \dot{R} \cdot \text{Tan}(\gamma) = U \cdot \text{Tan}(\gamma) = W \tag{5.6}$$

However, the aircraft will be disturbed and it is therefore necessary to regulate the position of the aircraft onto the glide path in addition to the constant sink rate command. This regulation is achieved by supplying an altitude command to the altitude controller based on the range of the aircraft from the glide path origin point. The altitude command is given by equation (5.5). The constant sink rate command and the regulating climb rate command from the altitude controller is then added and supplied to the airspeed and climb rate controller. Figure 5.11 shows the block diagram of the glide path controller.

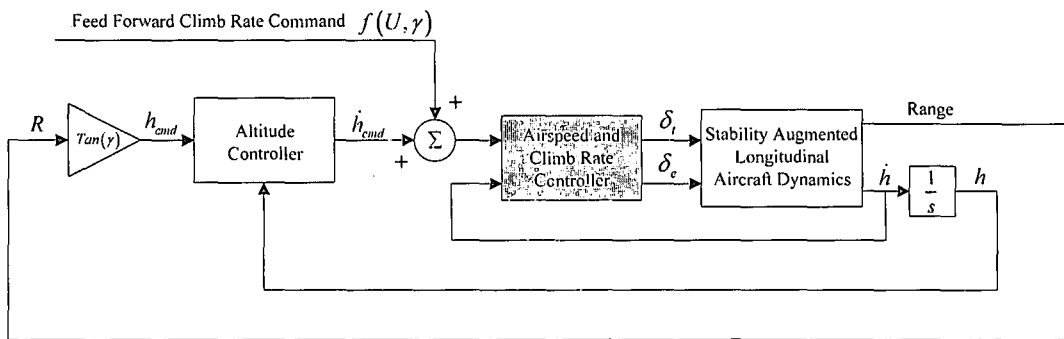


Figure 5.11: *Glide Path Controller Block Diagram*

### Linear Simulation

Figure 5.12 shows a linear simulation of the glide path controller. The altitude response along with the desired glide path (desired altitude) is shown. The aircraft initially flies straight and level. Once the glide path is crossed, the feed forward climb rate command is supplied to the airspeed and climb rate controller and the altitude controller is armed.

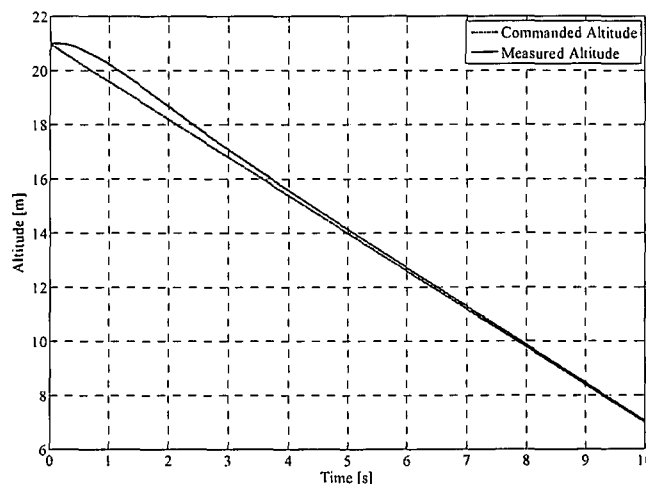


Figure 5.12: *Glide Path Controller Linear Simulation*

### 5.3 Flare Controller

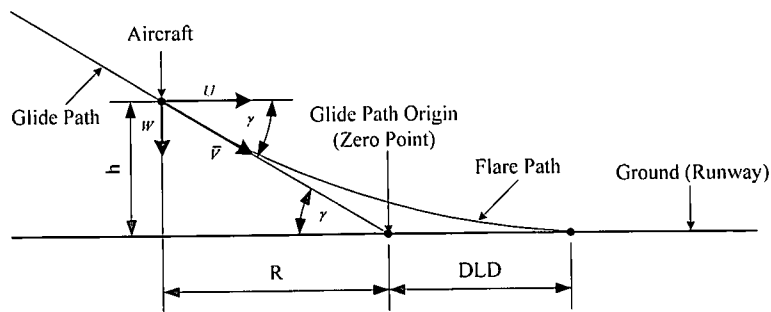


Figure 5.13: Flare Path Graphical Description

Figure 5.13 shows a graphical description of the flare path that the aircraft is required to fly during the final moments of landing.  $\gamma$  is the glide path angle.  $R$  is the range of the aircraft from the glide path origin point where the flare is initiated.  $h$  is the altitude and DLD is the desired landing distance from the glide path origin. The desired flare path is an exponential altitude function described by:

$$h = h_0 e^{-\frac{t}{\tau}} \tag{5.7}$$

where  $h_0$  is the initial altitude (the altitude at the start of the flare path at range  $R$ ) and  $\tau$  is the time constant for the exponential function. The sink rate on this path is obtained by differentiating equation (5.7) and is given by:

$$\dot{h} = -\frac{h_0}{\tau} e^{-\frac{t}{\tau}} = -\frac{h}{\tau} \tag{5.8}$$

#### Controller Design

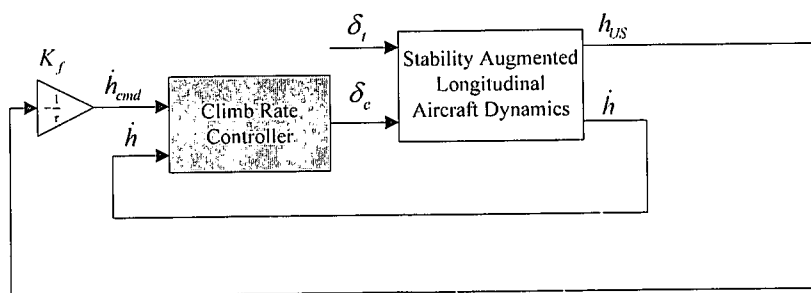


Figure 5.14: Flare Controller Block Diagram

Flying the flare path based on climb rate can be achieved by providing a climb rate command to the climb rate controller, based on equation (5.8). The block diagram for the controller is shown in Figure 5.14 where the plant dynamics are described by equation (4.10). The accurate altitude measurement ( $h_{US}$ ), required to generate the climb rate command, can be obtained from the ultrasonic range sensor.

If it is assumed that the airspeed of the aircraft remains constant throughout the flare ( $U_0$ ) and that the exponential functions dies away in four time constants, the time constant can be obtained from:

$$R + DLD = U_0(4\tau) \quad (5.9)$$

$R$  is the range at which the flare is initiated and  $DLD$  is the desired landing distance. In order for the gradients of the glide slope and exponential flare path to co-inside the initial flaring climb rate must equal the climb rate on the glide path. The initial climb rate is therefore given by:

$$\dot{h}_0 = -U_0 \sin(\gamma) \quad (5.10)$$

The flaring range can be written as

$$R = \frac{h_0}{\tan(\gamma)} = \tau U_0 \cos(\gamma) \quad (5.11)$$

after equation (5.8) and (5.10) is substituted for  $h_0$ . Substituting equation (5.11) into equation (5.9), the time constant can be calculated from the flaring airspeed, the glide slope angle and the desired landing distance:

$$\tau = \frac{DLD}{U_0[4 - \cos(\gamma)]} \quad (5.12)$$

The flaring airspeed is 12m/s and the desired landing distance ( $DLD$ ) is chosen as 50m. With a glide slope angle of  $4^\circ$  the time constant is calculated from equation (5.12) as:

$$\tau = 1.388 \quad (5.13)$$

The flare controller feedback gain is then:

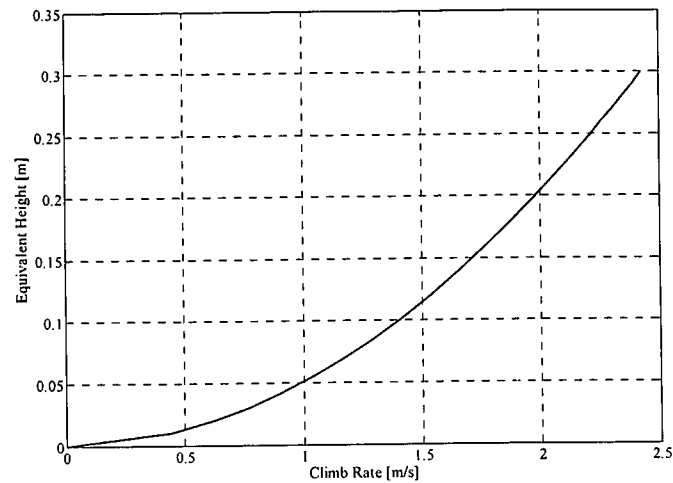
$$K_f = -\frac{1}{\tau} = -0.72 \quad (5.14)$$

### Linear Simulation

The parameter that will cause damage at touchdown is the sink rate of the aircraft. The sink rate can be quantified by converting the sink rate to an equivalent drop height, i.e. the height from which the aircraft must be dropped to attain the same sink rate when hitting the ground. The conversion from a sink rate to an equivalent drop height is done by the constant acceleration equation of motion given by:

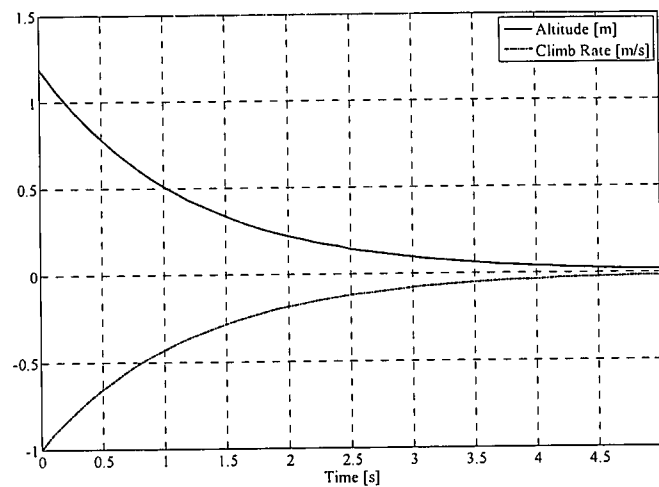
$$v^2 = 2as \rightarrow v = \sqrt{2as} \quad (5.15)$$

where the initial velocity  $u = 0$ . Figure 5.15 shows the relationship between sink rate and an equivalent drop height.



**Figure 5.15:** Sink Rate vs. Equivalent Drop Height

A sink rate of 0.5m/s at touchdown is equivalent to dropping the aircraft from 1.1cm above the ground and will not cause any damage. Any sink rate below 0.5m/s at touchdown can therefore be considered as a good landing.



**Figure 5.16:** Ideal Flaring Path and Sink Rate

Figure 5.16 shows the desired flare path and associated sink rate for  $K_f = -0.72$ . From the graph it is clear that the underlying climb rate controller requires a rise time of approximately 0.5s to track the flare path with reasonable precision. The rise time of the climb rate controller designed in section 4.4 is 2s. This is far too slow to track the exponential flare path. The bandwidth of the climb rate controller cannot be increased due to the excessive noise on the climb rate signal and the proportional feedback of the climb rate measurement to the elevator. The flare controller is therefore not used in this project.



## 5.4 De-Crab Controller

The yaw rate generated by de-crabbing the aircraft will stimulate a response through the yaw orientation control system (YOCS) to counter the yaw rate. A MIMO design is therefore required to regulate both the yaw rate and the error between the body and course over ground headings (side slip angle) with the rudder and ailerons. Since the de-crabbing of the aircraft can be done gradually along the landing path, it is possible to design a slow de-crabbing controller that will limit the amount of coupling into the YOCS. It will also allow the design of the controller using Single-Input-Single-Output (SISO) methods. The sensitivity of the heading measurements from the magnetometer to specifically the roll angle of the aircraft further encourages a slow de-crab controller. This will minimise the coupling into the YOCS to minimise the roll angles obtained while de-crabbing.

A very low overshoot is desired from the controller to ensure a high phase margin and to approximate a first order response. A slow rise time is required to decouple the de-crab controller from the yaw orientation control system as explained. An integrator term will be required in the compensator to provide a non-zero steady state command to the rudder to reject the cross-wind disturbances and ensure a zero steady state error.

The design specifications for the de-crab controller are:

- Rise time of under 6s
- Zero steady state error
- Overshoot of less than 5%

### Controller Design

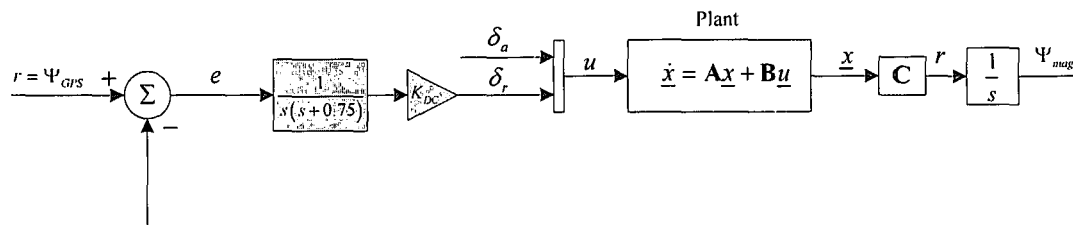


Figure 5.17: De-Crab Controller Block Diagram

Figure 5.17 shows the block diagram for the de-crab controller. The plant dynamics are given by equation (B.1) with the aerodynamic model linearised at 14m/s and the output matrix  $\mathbf{C}$  is set to extract the yaw rate state of the aircraft in deg/s. The yaw rate state then passes through a natural integrator to become the body axis heading state ( $\Psi_{mag}$ ) in degrees. The course over ground heading from the GPS ( $\Psi_{GPS}$ ) serves as the reference input to the system. Augmenting the heading state to the plant results in the following system:

$$\begin{bmatrix} \dot{x} \\ \dot{\Psi}_{mag} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} x \\ \Psi_{mag} \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} u(k) \quad (5.16)$$

The compensator in Figure 5.17 includes an integrator term and an additional slow pole that act as a low pass filter for the rudder command at  $s = -0.75$ . The pole ensures the long rise time desired for the de-crab controller.

The compensator dynamics are described by:

$$C_{DC}(s) = \frac{1}{s(s+0.75)} \quad (5.17)$$

Figure 5.18 shows the root locus for the variation in the feedback gain  $K_{DC}$  with the compensator in the feedback path. The root locus on the right shows a magnified view of the origin of the root locus on the left. Four of the nine poles correspond to the natural dynamics of the aircraft (the dutch roll mode, spiral mode and roll mode). Two poles are added by the compensator as described by equation (5.17). One pole is due to the heading state and another to the washout filter state added during the dutch roll damper design. The last pole is the integrator state added by the yaw orientation control system.

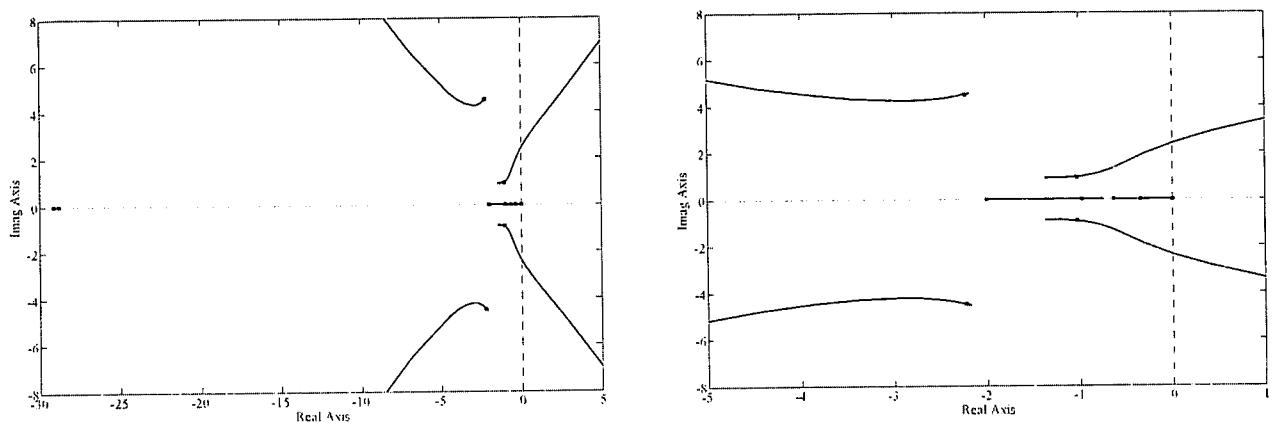


Figure 5.18: De-Crab Controller Root Locus

A compensator gain of

$$K_{DC} = -0.5 \quad (5.18)$$

is found to satisfy all the design specifications. Augmenting the system of equation (5.16) with the compensator dynamics result in the closed loop system given by:

$$\begin{bmatrix} \dot{\underline{x}} \\ \dot{\underline{x}}_c \end{bmatrix} = \begin{bmatrix} \mathbf{A} - K_{DC} \mathbf{B} \mathbf{D}_c \mathbf{C} & K_{DC} \mathbf{B} \mathbf{C}_c \\ -\mathbf{B}_c \mathbf{C} & \mathbf{A}_c \end{bmatrix} \begin{bmatrix} \underline{x} \\ \underline{x}_c \end{bmatrix} + \begin{bmatrix} K_{DC} \mathbf{B} \mathbf{D}_c \\ \mathbf{B}_c \end{bmatrix} r \quad (5.19)$$

$\mathbf{A}_c$ ,  $\mathbf{B}_c$ ,  $\mathbf{C}_c$  and  $\mathbf{D}_c$  are the state space representation matrices and  $\underline{x}_c$  is the state vector of the compensator given by equation (5.17).  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{C}$  are the state space representation matrices and  $\underline{x}$  the state vector of the plant dynamics described by equation (5.16).

### Linear Simulation

Figure 5.19 shows the body axes heading response of the de-crab controller to a  $5^\circ$  step in heading error. In the simulation results the aircraft is crabbed rather than de-crabbled with no crosswind present. The results were generated by starting with the aircraft flying straight and level. A step command in the GPS heading reference

( $\Psi_{GPS}$  in Figure 5.17) was then given. The yaw rate caused by the rudder deflection will be countered by the ailerons until the system reaches a steady state of zero yaw rate and the commanded heading. When de-crabbing, the rudder will be used to straighten the aircraft (counter the weather cock effect into the wind) while a constant aileron deflection will counter the crosswind and keep the aircraft flying the desired heading. The responses during an actual de-crab scenario will therefore be slightly different from the simulated responses. However, the simulation still provides a good general idea as to what the controller response entails. Note the constant rudder command that keeps the body axes heading at 5 degrees.

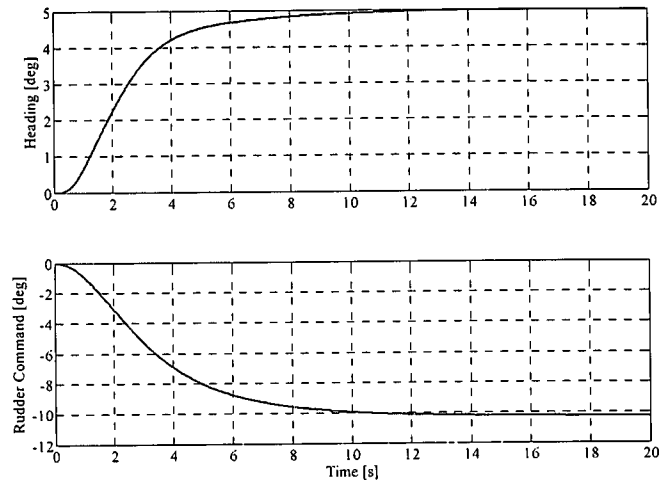


Figure 5.19: De-Crab Controller Linear Simulation

Figure 5.20 shows the dynamic responses of the bank angle ( $\phi$ ), yaw rate ( $r$ ), aileron deflection ( $\delta_a$ ) and side slip velocity ( $v$ ) during the heading error step. Note the constant aileron deflection and resulting bank angle to counter the yaw rate generated by the constant rudder deflection.

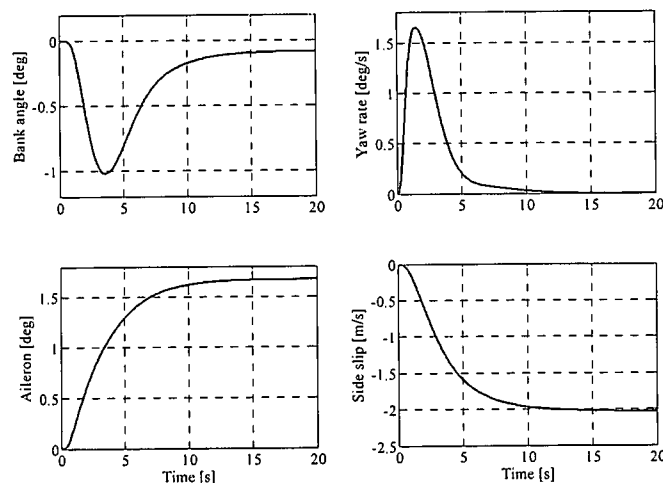
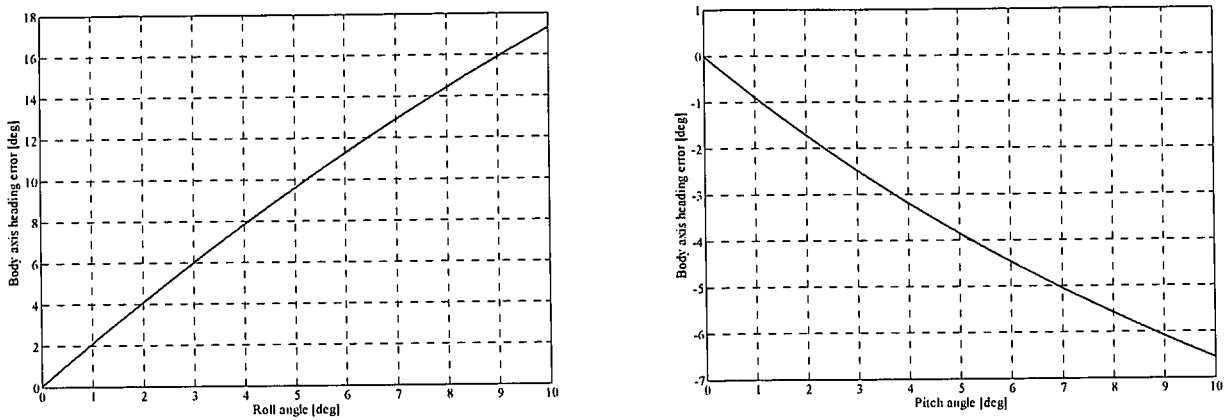


Figure 5.20:  $\phi, r, \delta_a, v$  while de-Crabbing

The sensitivity of the magnetometer heading measurement to roll and pitch angles was discussed in section 2.1.2. Figure 5.21 shows the magnetometer (body axis) heading errors to variations in roll and pitch angles.



**Figure 5.21:** Magnetometer Heading Error with Roll and Pitch Angles

For small roll angle perturbations of  $10^\circ$ , the body axis heading error is  $17^\circ$ . This error becomes significant and can induce an oscillation while the aircraft is de-crabbed. The de-crab controller is therefore not implemented in this project.

## 5.5 Summary

The design and re-design of all the controllers that form the building blocks of the landing sequence discussed in section 2.2.1 was presented in this chapter. The trim constraint was removed from the conventional flight controllers to allow operation over the entire airspeed range required during landing. The steady state error of the guidance controller was improved to allow for better lateral control during a landing and the dutch roll damper was stabilised for wind gust rejection. The flare and de-crab controller designs were presented for the sake of completeness and possible use in future projects. All the controller designs were verified by linear simulation.

# Chapter 6

## *Avionics and Ground Station*

This chapter discusses the modifications and extensions to the avionics package and ground station. A high level, block diagram approach is adopted to present the hardware development with the essential information in section 6.1. The ground station is discussed in section 6.2 where an overview of the additional Graphical User Interface (GUI) pages is presented. A summary of the avionics package details is given in section 6.3.

### 6.1 Avionics

The avionics used for practical implementation and hardware in the loop simulation is based on the avionics designed in [1]. The original avionics is shown and discussed in section 1.2.4, with a picture and block diagram shown in Figure 1.6 and Figure 1.7 respectively. The modifications and improvements made to the avionics, as well as the extra functionality and sensors added, are listed below:

- New layout and packaging
- New differential pressure sensor with signal conditioning
- Modified inertial measurement unit (IMU)
- New PWM RC receiver interface
- GPS signal conditioning
- Multiplexed programming through a single header
- Hardware in the loop interface – sensor multiplexing
- New RF module
- Ultrasonic range sensor and magnetometer added
- Updated software flow

Each of the points is briefly discussed in the sections to follow. Appendix A contains additional information and more detail on the avionics. Figure 6.1 shows a block diagram of the updated avionics with the modified or added blocks shaded for easy identification. Fully shaded blocks represent new additions, while partly shaded blocks indicate that modifications were made to the existing system.

The aircraft consists of the avionics package and aircraft hardware (an RC receiver, servos and a pitot-static system). The ground station consists of the RC transmitter, a second RF transceiver and a laptop running the ground station software.

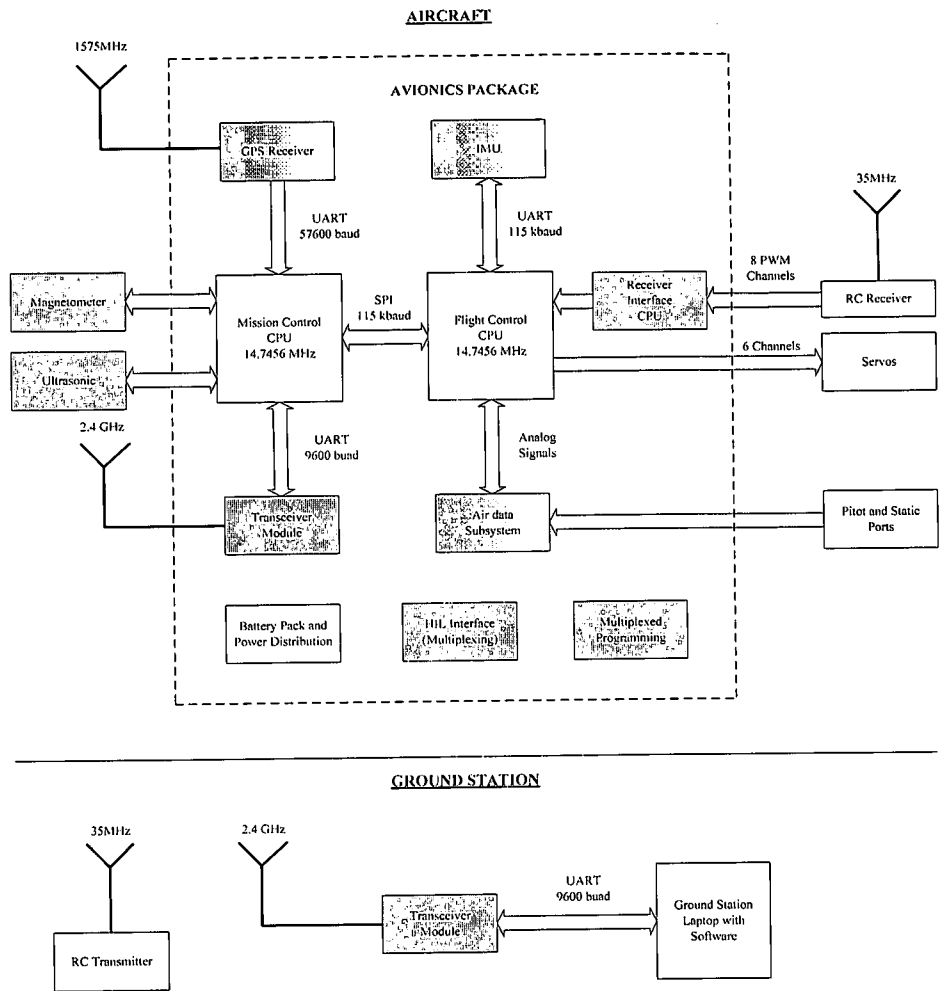
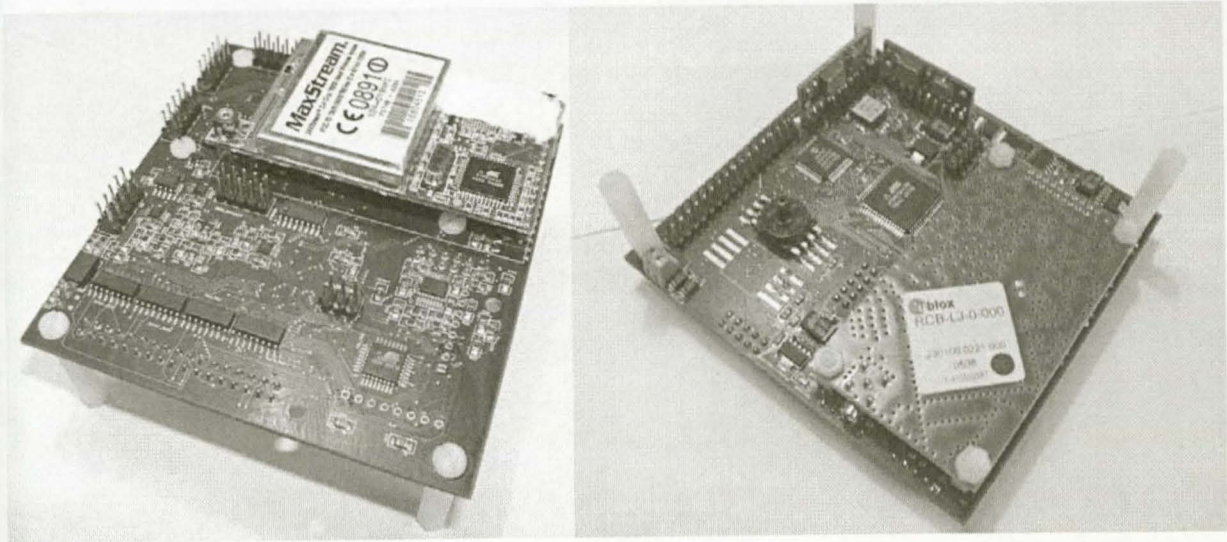


Figure 6.1: Avionics and Ground Station Block Diagram

### 6.1.1 Redesign and Packaging

The entire avionics package is redesigned so the hardware required for conventional flight fits onto a single multi-layer PC104 sized printed circuit board. The redesign of the avionics package resulted in a small, robust and very reliable hardware platform. Together with the bill of materials and production documentation, the newly developed avionics package can easily be duplicated for use in future UAV projects, eliminating the frustrating and time consuming process of hardware development. The avionics package is now also on the level of a commercial product and is shown in Figure 6.2.



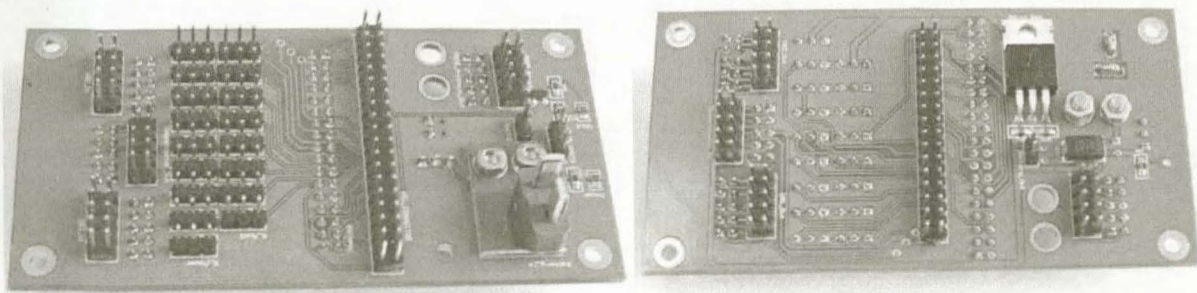


**Figure 6.2:** *Avionics Printed Circuit Board*

The avionics is re-packaged into a very rigid and robust aluminium case with one side covered by an intermediate board to serve as an interface to the avionics inside. The intermediate board provides the following connection points:

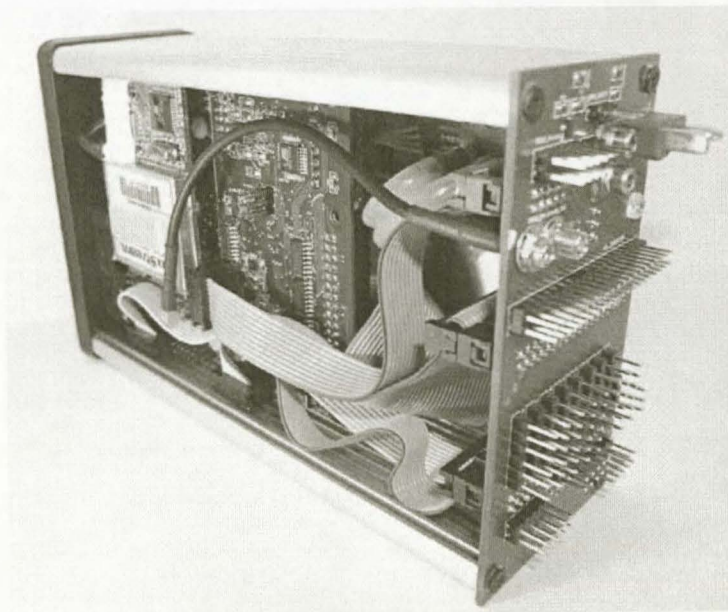
- Output to servos
- PWM signal inputs from the RC receiver
- 2 x Ultrasonic connections
- Magnetometer connection
- HIL connection header
- GPS and RF transceiver antenna connections
- Microcontroller programming header
- Battery pack connector (Deans connector)

The intermediate board also contains a 3A, 5V regulator to provide power to the RC receiver and servos. The regulator is sufficient to provide current for six servos under normal operation. Figure 6.3 shows the intermediate board while Figure 6.4 shows the packaged avionics.



**Figure 6.3:** *Intermediate Board*



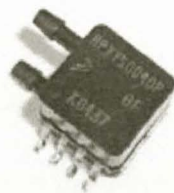


**Figure 6.4:** Packaged Avionics

The entire avionics package fits perfectly into the aircraft used in this project. Vibration isolation is provided by sponge padding.

### 6.1.2 Differential Pressure Sensor

The new differential pressure sensor is the *Motorola MPXV5004DP* shown in Figure 6.5.

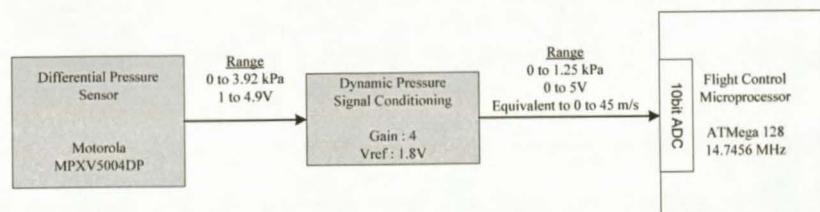


**Figure 6.5:** Motorola MPXV5004DP Differential Pressure Sensor [18]

A new differential pressure sensor is desired for two reasons:

1. Size: the sensor used in [1] takes up an unacceptable portion of the new PC104 sized avionics.
2. Cost: The new differential pressure sensors costs R300 compared to R1000 for the old sensor.

Figure 6.6 is a block diagram of the signal conditioning for the new sensor.



**Figure 6.6:** Differential Pressure Signal Conditioning Block Diagram

The raw output of the differential pressure sensor is a 0 to 3.92kPa signal mapped over an output voltage range of 1V to 4.9V.

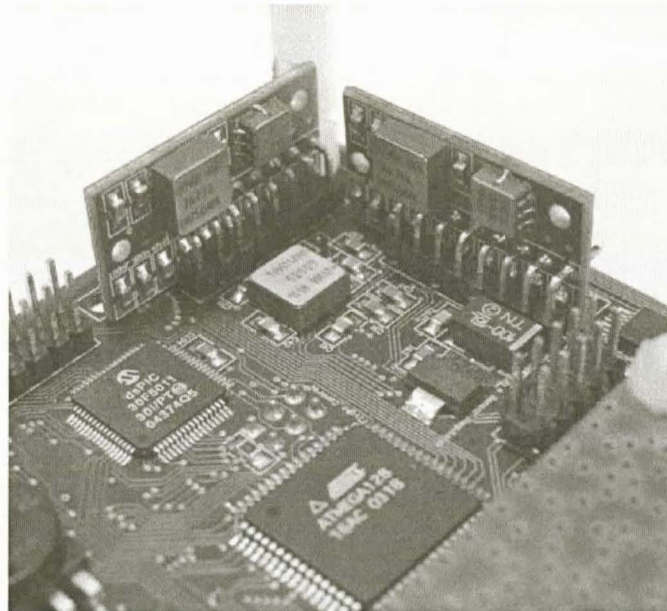
It is desired to measure airspeeds of up to 45m/s, translating into a dynamic pressure range of 0kPa to 1.25kPa. The signal conditioning block maps the desired pressure range over the entire 5V range of the ADC to maximise the digital resolution. The digital resolution is obtained by differentiating the equation for dynamic pressure and converting it to a 10bit ADC value using the full 5V voltage range. The result is [1]:

$$\text{Digital Resolution} = \left( \frac{\rho \bar{V} \cdot 5 \cdot 2^{10}}{1.25 \times 10^3 \cdot 5} \right)^{-1} \quad (6.1)$$

where  $\rho$  is the air density and  $\bar{V}$  is the airspeed. The digital resolution is 0.05 m/s at 20m/s at sea level. A circuit diagram for the signal conditioning block is shown in section A.6.

### 6.1.3 Inertial Measurement Unit

The inertial measurement unit in [1] was implemented using gyroscopes on development boards and an Atmel ATmega128 microprocessor. The gyroscope integrated circuit and the required peripheral electronics are now directly integrated onto the avionics board and two perpendicular boards to reduce the space required. The same gyroscopes, accelerometers and signal conditioning as discussed in section A.2 (with the block diagram shown in Figure A.2) are still used. The IMU microprocessor is changed to an *Atmel ATmega8*. It is much smaller in size and has only the necessary peripheral devices. Figure 6.7 shows the new IMU integrated onto the avionics board.



**Figure 6.7:** *Inertial Measurement Unit*

### 6.1.4 PWM Receiver Interface

The original avionics interfaced with the RC receiver by drilling a hole in the receiver, tapping out the Pulse Position Modulated (PPM) signal that contains the servo position information and routing it to Flight Control. A new receiver interface is developed where up to eight output channels from any receiver can plug into the avionics package. The eight receiver channels are then routed to a *Microchip dsPic30F5011* microprocessor where they are timed in. The receiver interface can work with both Pulse Code Modulated (PCM) and Frequency Modulated (FM) receivers.



The dsPic30F5011 was selected since it has eight dedicated ‘interrupt on input capture’ pins which are required to capture the pulse widths of eight Pulse Width Modulated (PWM) receiver channels. As soon as a single channel is timed in, the pulse width is placed on a 12bit parallel data bus to Flight Control, the 3bit channel address is written out and Flight Control is interrupted to read the data. The interrupt from the dsPic simply replaces the original RC receiver PPM signal interface interrupt and therefore keeps the software structure the same. A block diagram of the receiver interface is shown in Figure 6.8.

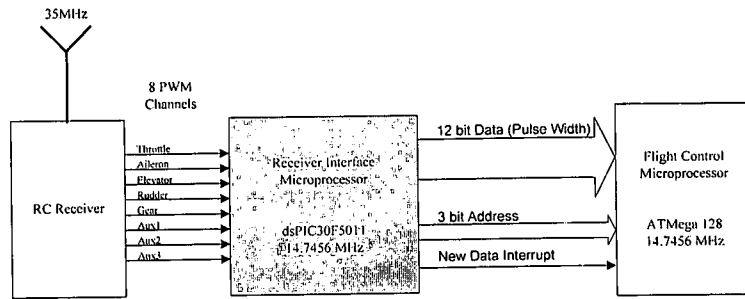


Figure 6.8: Receiver Interface Block Diagram

Figure 6.9 shows a flow diagram of the code running on the dsPic30F5011.

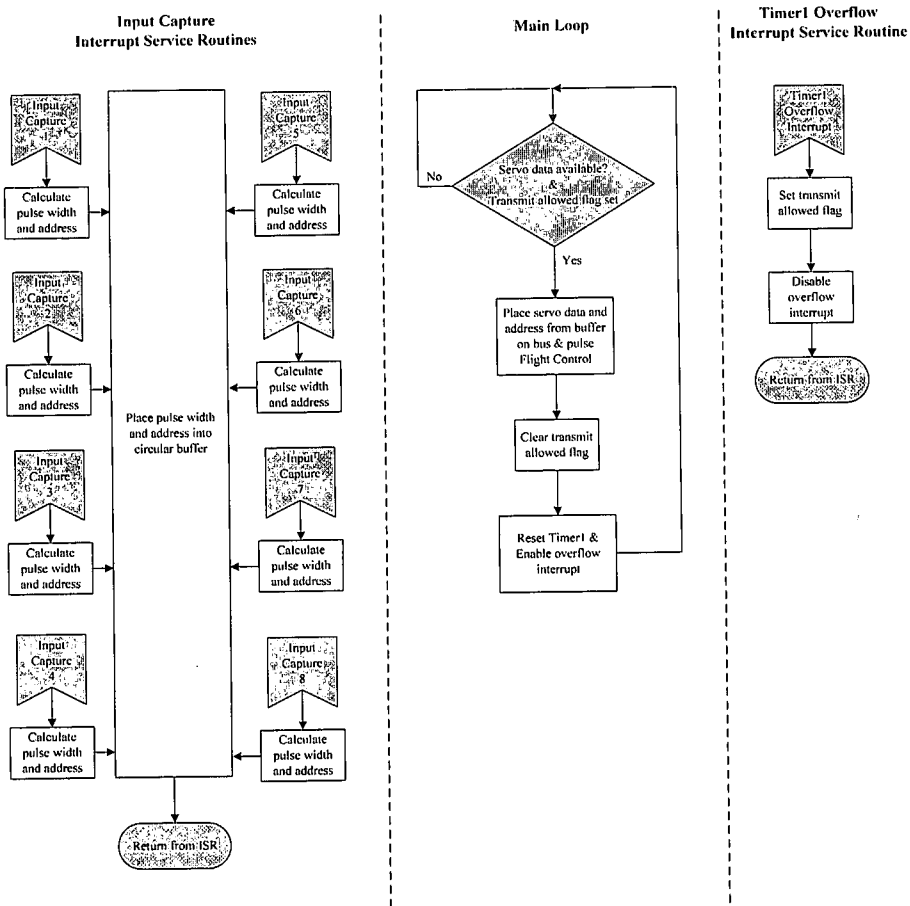
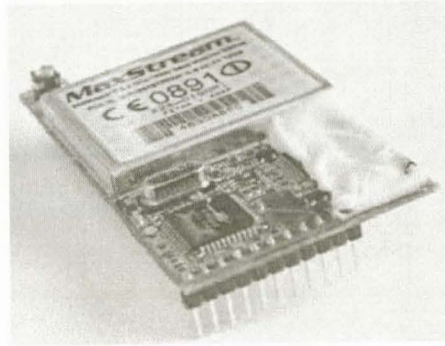


Figure 6.9: Receiver Interface CPU Flowchart

## 6.1.5 RF Module and GPS Signal Conditioning

### RF Module

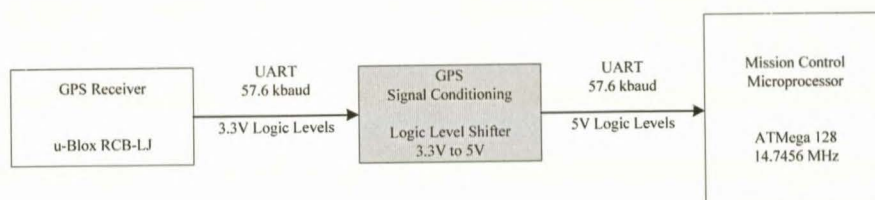
The AeroComm RF transceiver module used in the original avionics proved to be very unreliable and showed poor communication capabilities. A new RF module, the *MaxStream XStream OEM 2.4GHz* module was integrated into the system and showed excellent results. The module is shown in Figure 6.10.



**Figure 6.10:** *MaxStream XStream OEM 2.4GHz RF Module*

### GPS Signal Conditioning

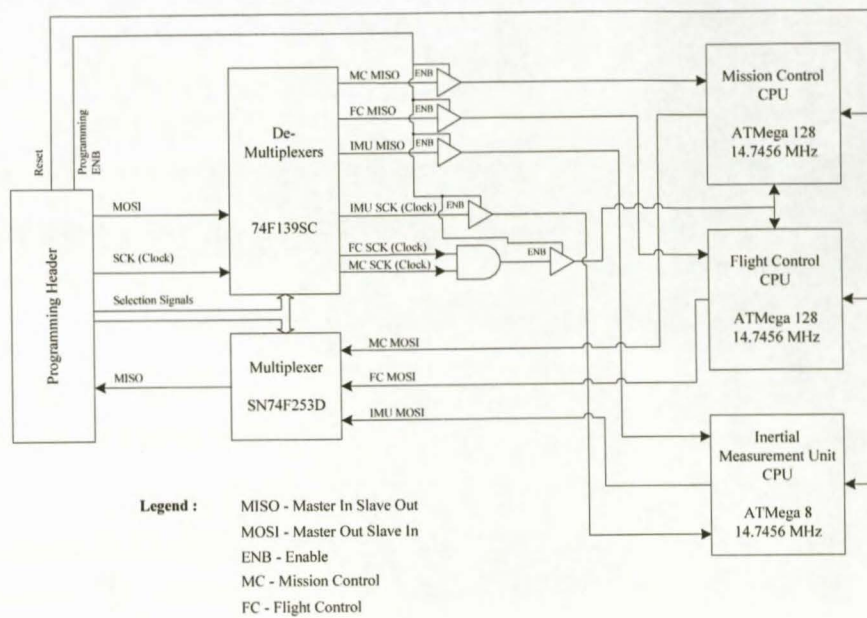
The GPS receiver outputs serial UART data at 3.3V levels. In order to improve the reliability of the system by increasing the noise margin, the GPS data is sent through a signal conditioning block, translating the UART signals to 5V levels. Figure 6.11 shows a block diagram of the signal conditioning.



**Figure 6.11:** *GPS Signal Conditioning Block Diagram*

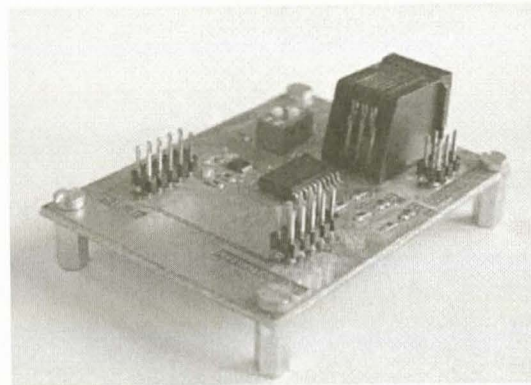
## 6.1.6 Multiplexed Programming

The original avionics made use of three separate programming headers for the main microcontrollers: Flight Control, Mission Control and the IMU processor. The programming of the microcontrollers was also limited to a specific order. The programming process is improved by adding a multiplexed programming scheme to the avionics. This allows for programming of all three microcontrollers through a single header and in any desired order, easing the development process and saving space on the new avionics board. A block diagram of the scheme is shown in Figure 6.12. Refer to [21] for details on AVR Programming and signal definitions.



**Figure 6.12:** *Multiplexed Programming Block Diagram*

A separate programming board serves as a user interface from which the desired microcontroller to program can be selected. The board is also used to program the dsPic30F5011 for the receiver interface and is shown in Figure 6.13.

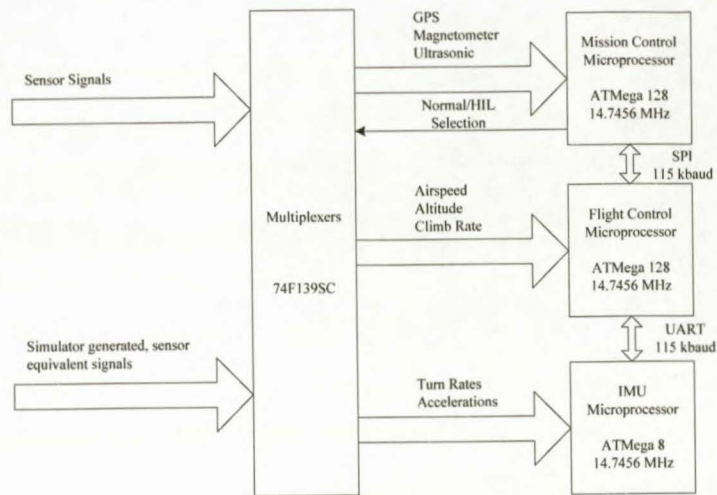


**Figure 6.13:** *Programming Board*

### 6.1.7 Hardware in the Loop Interface

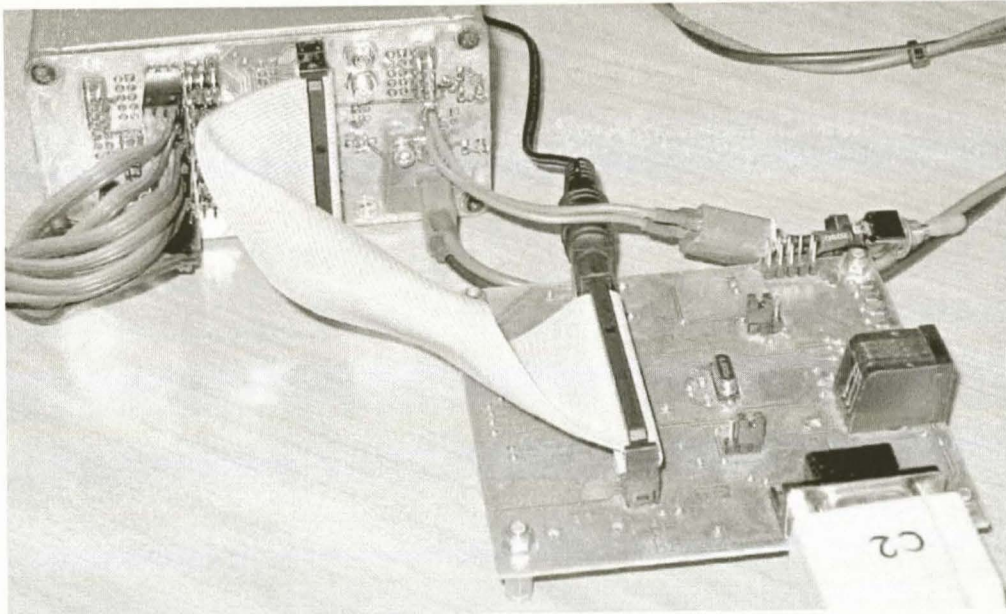
An HIL interface was developed to allow the avionics package to function with the hardware in the loop (HIL) simulator. The interface consists of multiplexing all the sensor inputs to the microcontrollers with the option to select either the actual sensor signal or the HIL signal. Selection is controlled by a single line from Mission Control. Figure 6.14 provides a block diagram of the HIL interface.





**Figure 6.14:** *HIL Interface Block Diagram*

The servo commands generated by the avionics package are also routed out to the HIL board, which serves as an interface to the HIL simulator. Figure 6.15 shows the intermediate HIL board connected to the avionics. The HIL simulator and intermediate board is discussed in section 7.1.

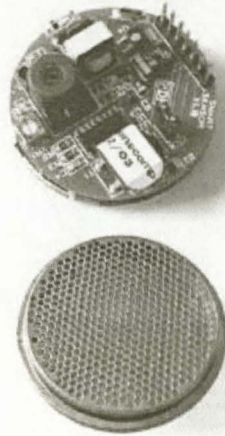


**Figure 6.15:** *HIL Board Connected to the Avionics*

## 6.1.8 Ultrasonic Range Sensor and Magnetometer

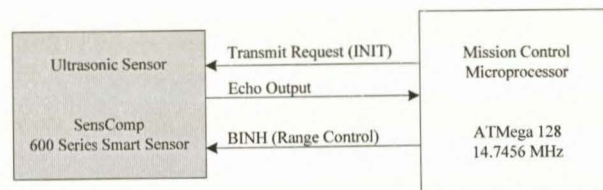
### Ultrasonic Range Sensor

The ultrasonic range sensor used is the *SensComp 600 Series Smart Sensor* shown in Figure 6.16.

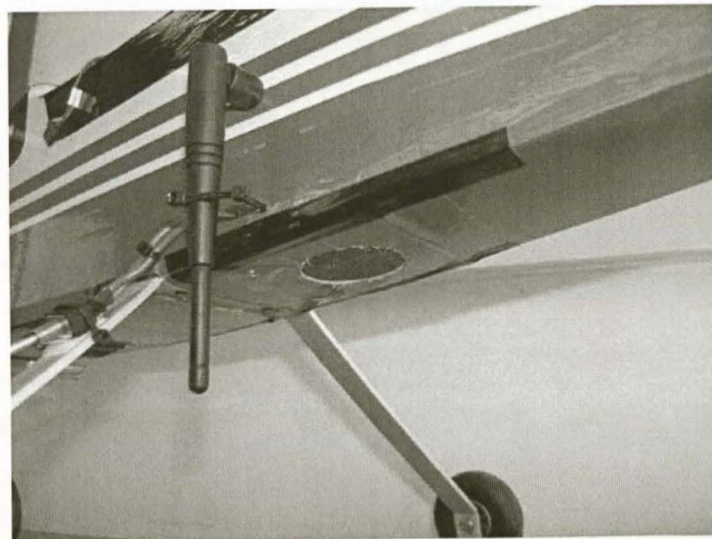


**Figure 6.16:** *SensComp Ultrasonic Sensor [18]*

The sensor is set up to measure altitudes between 12cm and 5m and is capable of 1cm resolution. No additional circuitry is required. The interface is with Mission Control through three control lines. The sensor provides measurement updates at 10Hz, with a single cycle delay of 0.1s. The sensor is powered directly from the battery pack and is mounted under the aircraft as shown in Figure 6.18. Figure 6.17 shows a block diagram of the Mission Control-sensor interface.



**Figure 6.17:** *Ultrasonic Interface Block Diagram*



**Figure 6.18:** *Ultrasonic Sensor Mounted in the Aircraft*



The ultrasonic range sensor was tested and its performance evaluated on a moving platform such as an aircraft. The results are given in Chapter 8 and proved very satisfactory as the sensor performed well during conventional flight and landing. The high pitch angle during take-off does not allow for an accurate measurement from the sensor, but it can still be used to detect lift-off.

## Magnetometer

The magnetometer used is the *Honeywell HMC2003* shown in Figure 6.19.

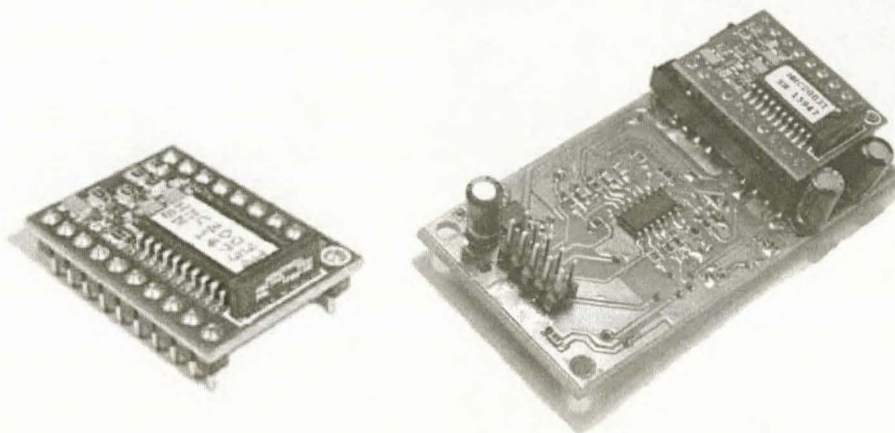


Figure 6.19: *Honeywell Magnetometer* [18]

The magnetometer provides magnetic field measurements in three axes. The raw output of the magnetometer maps a  $\pm 2$  gauss range over a voltage span of 4V with a 1 kHz bandwidth. Signal conditioning reduces the bandwidth to 100 Hz and provides an amplification factor of 4. After signal conditioning a  $\pm 0.625$  gauss range is mapped over a voltage span of 5V. Set/Reset pulses to the magnetometer are used to re-align the magnetic domains and in so doing, re-calibrating the sensor. Section F.2.2 in [20] provides the detailed design and schematic diagrams of the signal conditioning and Set/Reset circuitry.

The three outputs are sampled by the onboard 10bit ADC's on Mission Control at 10Hz, resulting in a digital resolution of 1.22 mille gauss. Figure 6.19 shows the separate board developed for the magnetometer and accompanied signal conditioning. Figure 6.20 shows the block diagram for the magnetometer with signal conditioning.

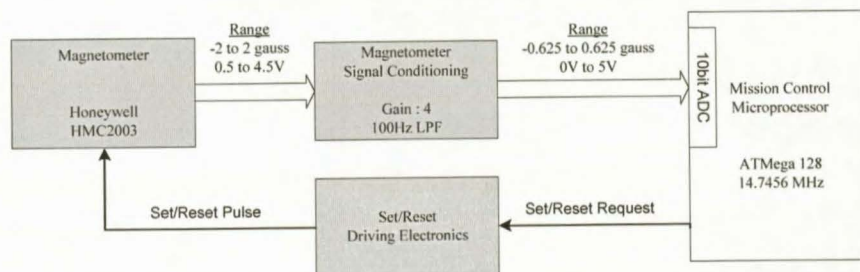


Figure 6.20: *Magnetometer Interfacing Block Diagram*

### 6.1.9 Software

The software on Flight Control and Mission Control is modified in order to incorporate new sensors, execute the mission sequences and improve the execution of the control algorithms.

#### Flight Control

Figure 6.21 shows the updated flow diagram for Flight Control. Shaded blocks indicate updated or newly added blocks.

Modifications to Flight Control are:

- Driving the servos after the control algorithms is executed. This modification ensures the servos are updated with the current commanded values and removes a 20ms delay that was present in the system.
- Removing the 4Hz controllers from the 25Hz loop by implementing a separate 4Hz loop for the controllers. This modification removes a delay of up to 40ms from the 4Hz controllers.

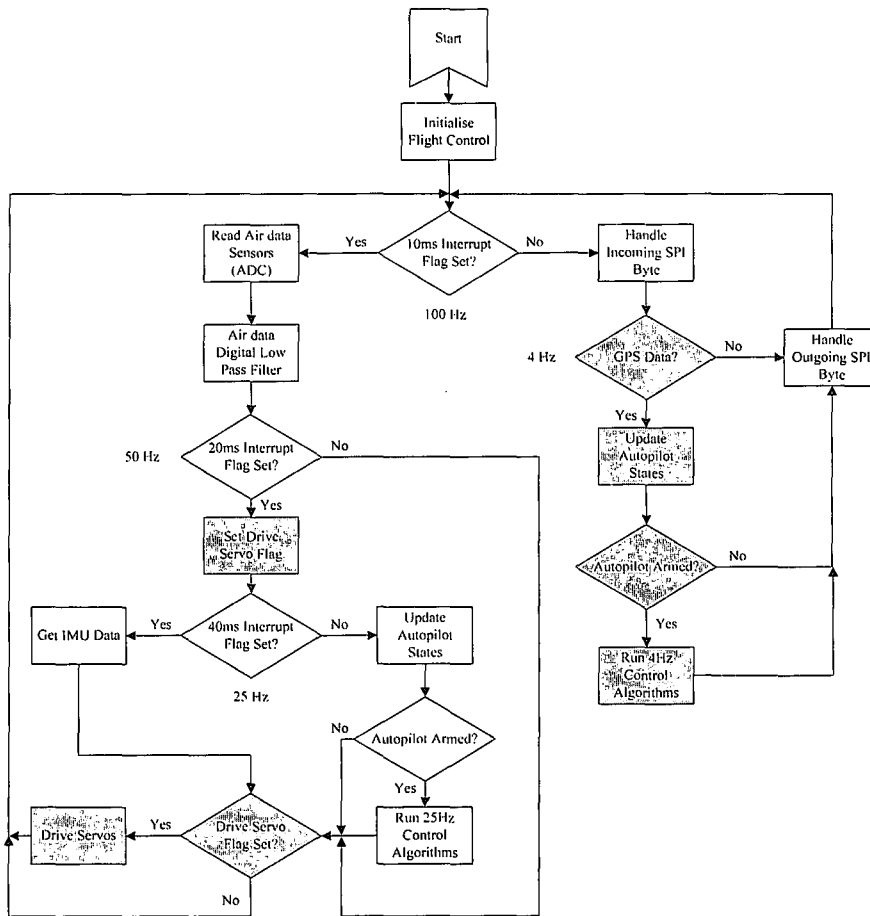


Figure 6.21: Flight Control Flow Diagram

### Mission Control

Figure 6.22 shows the updated flow diagram for Mission Control. Shaded blocks indicate updated or newly added blocks.

Modifications to Mission Control are:

- Including mission sequence selection and execution at 10Hz.
- Handling the ultrasonic range sensor and reading the magnetometer signals at 10Hz.

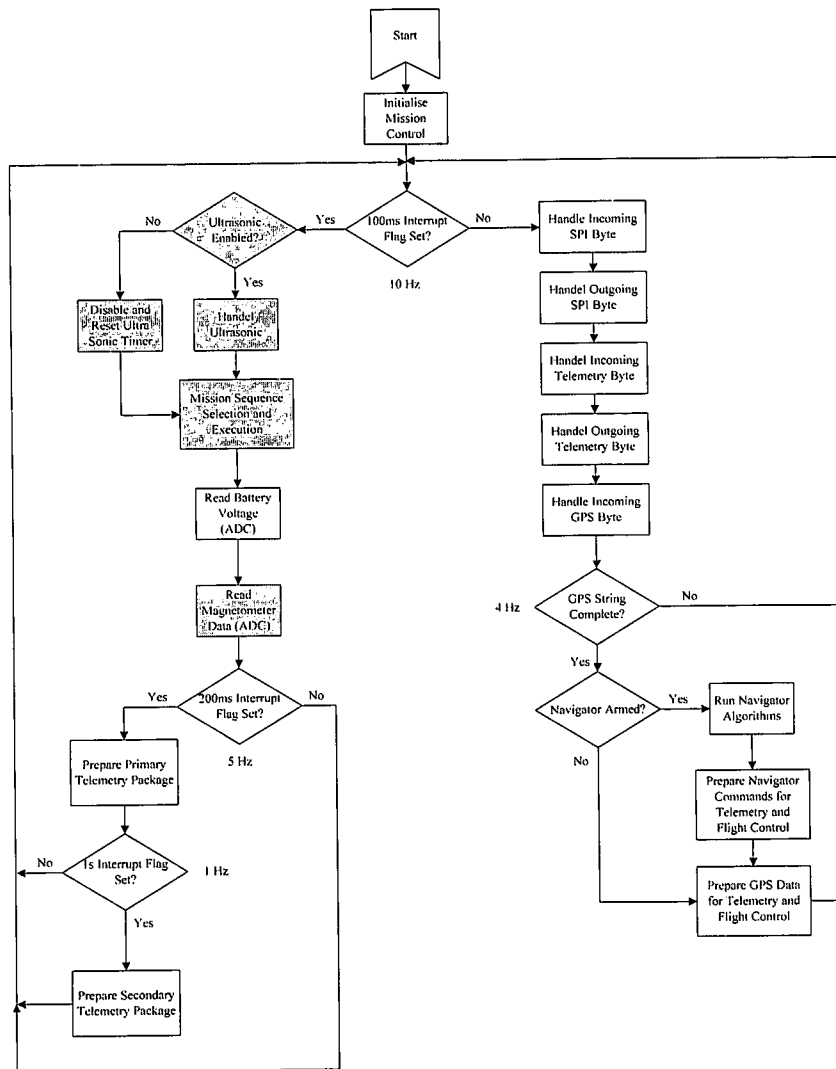


Figure 6.22: Mission Control Flow Diagram

Mission sequence selection is a high level of system control that enables the user to command a specific sequence to be executed. The user can choose between three sequences:

- Take-Off
- Navigation
- Landing

The system will determine the current flight sequence and performs the tasks required to execute the commanded sequences. Figure 6.23 shows the mission sequence selection and execution flow diagram.

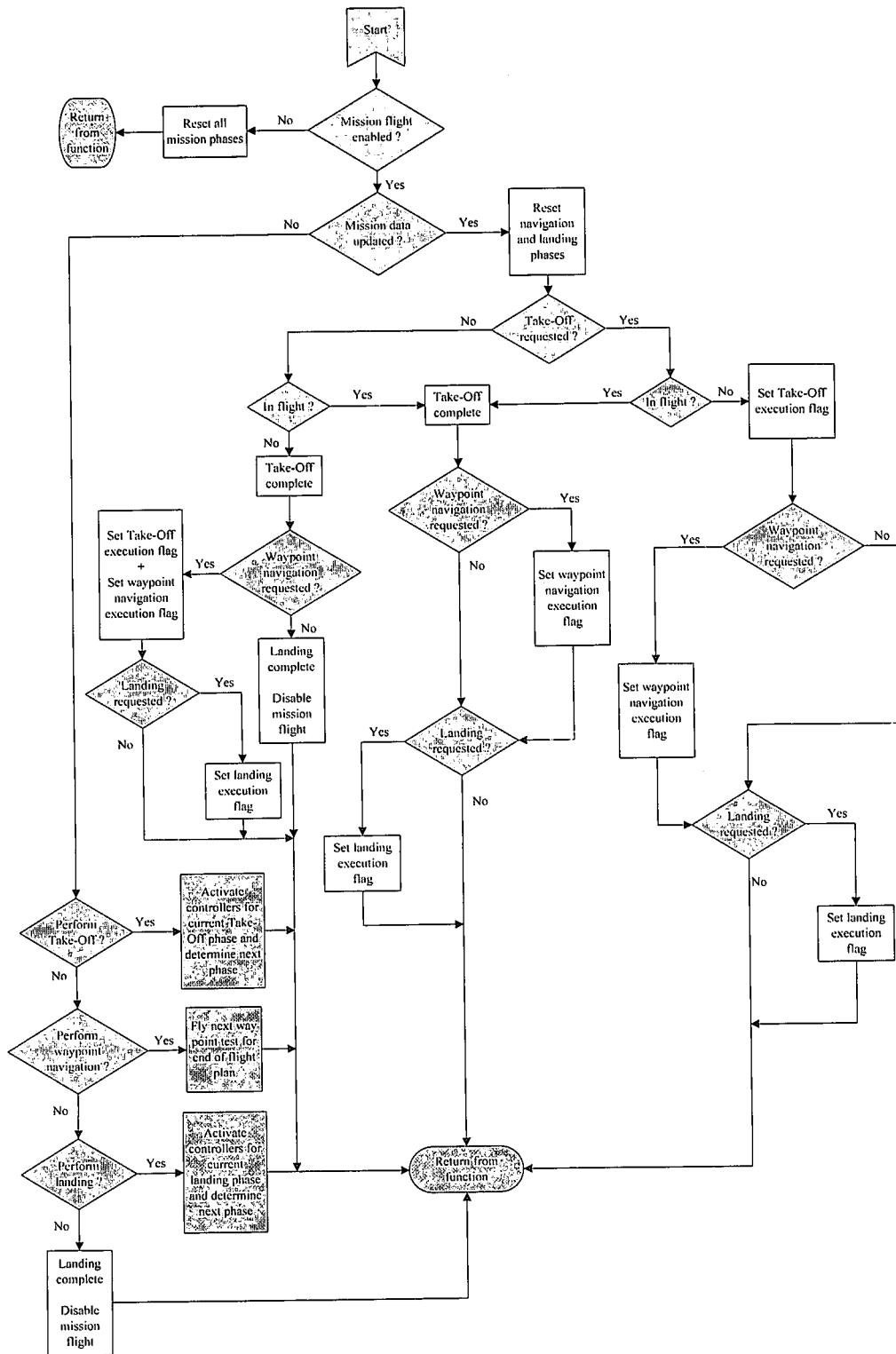


Figure 6.23: Mission Flight Logic Flow Diagram

## 6.2 Ground Station

The original ground station was developed in [1] and is briefly discussed in section 1.2.4. Figure 6.1 shows the updated block diagram. The new RF transceiver module, shown in Figure 6.10, is integrated into the ground station to form a new wireless link to the aircraft at 2.4GHz.

The GUI software is extended to include the following functionality:

- Mission flight (sequence) control
- Updating of controller parameters
- Interfacing with the ultrasonic and magnetometer
- Manipulation of the landing path and approach point
- Selection and execution of test sequences

Section 6.2.1 discusses the added functionality in more detail. The flight planner page is not discussed since no changes were made.

## 6.2.1 Ground Station Software

### Ground Station Page

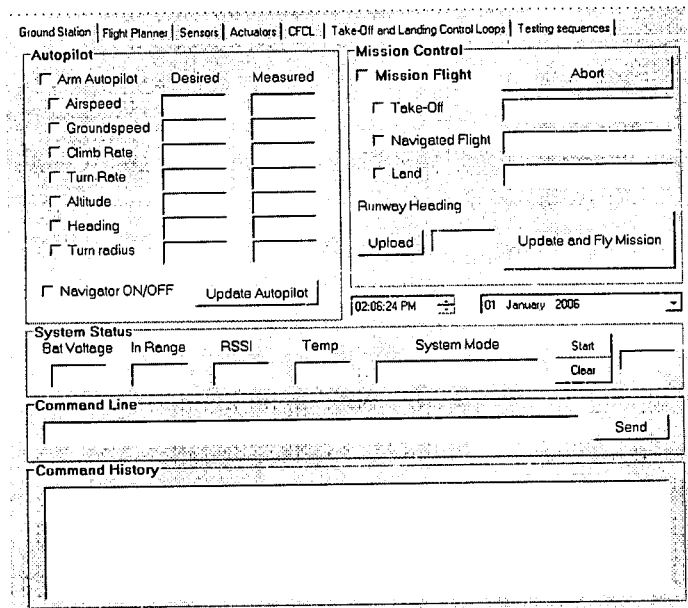


Figure 6.24: Ground Station Page

The ground station page is the main page of the GUI and is shown in Figure 6.24. The page allows for the selection and arming of controllers and displays the commanded values with the measured values for easy comparison. Selection between the runway and air-borne controllers are done by arming the specific control loops on the conventional flight control page and the take-off and landing control page.

Mission sequence selection is also controlled from the ground station page. The user can select either the Take-Off or Land checkbox and click on the “Upload and Fly Mission” button to start execution of the relevant sequence. Selecting the Navigated Flight checkbox will arm the navigator and start autonomous navigation. The page also displays critical system information like battery voltage, temperature and the system mode (normal or HIL).



Sensor and Actuator Pages

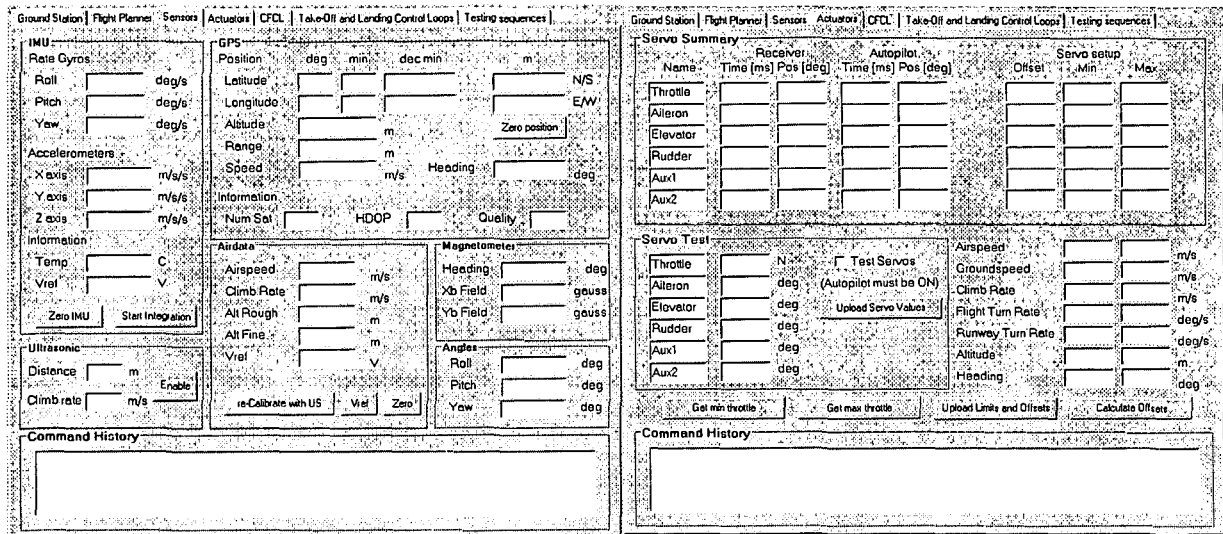


Figure 6.25: Sensor and Actuator Pages

The sensor page, displaying all the measured data from the aircraft, is shown on the left in Figure 6.25. New additions to the page include the magnetometer heading and field strengths, the GPS range measurement and the ultrasonic distance and climb rate with a button to enable the sensor.

Figure 6.25 shows the actuator page on the right. The actuator page allows for the calibration of the servos, limiting the command variables and displays actuator information during flight. New additions to the page are the “Get min throttle” and “Get max throttle” buttons used to capture the full throttle and throttle closed position. These positions are used to map a Newton command from the acceleration controller to an actuator position as discussed in section 4.1.

Conventional Flight and Take-Off and Landing Control Pages

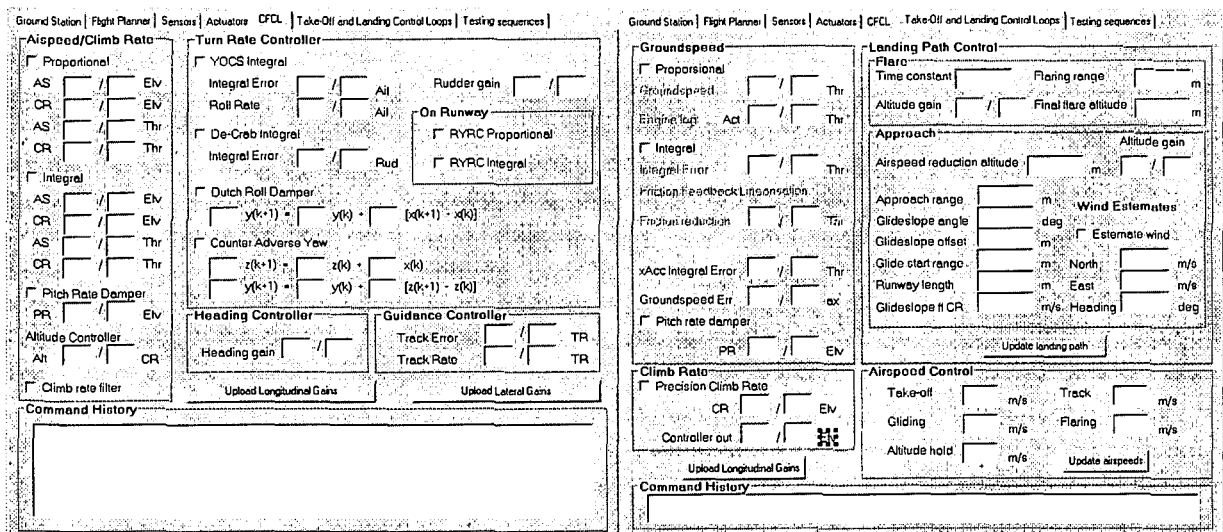


Figure 6.26: Conventional Flight + Take-Off and Landing Control Pages

The conventional flight page is shown on the left and the take-off and landing control page is shown on the right in Figure 6.26. The pages allow for specific control loops or a specific section of a loop to be enabled. It also allows the controller gains to be changed and uploaded to the avionics package to fine tune controllers during a flight test.

The take-off and landing control page provides control over the landing path, as well as the airspeeds, during the different sections. The wind estimation procedure can be enabled from the take-off and landing control page where the result will also be displayed.

## Test Sequence Page

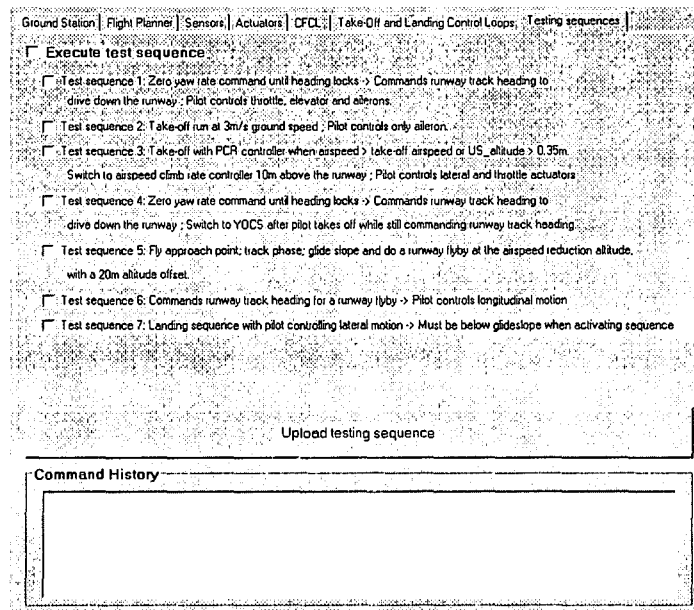


Figure 6.27: Test Sequence Page

Figure 6.27 shows the test sequence page. This page allows for certain sections of the take-off and landing sequence to be performed to allow for step by step practical testing of the sequences. For example, only the lateral control during the landing sequence can be controlled by the autopilot, while the safety pilot controls the longitudinal motion to evaluate lateral performance.

## 6.3 Summary

This chapter provided a high level discussion of the avionics and ground station. The final result is a very reliable, robust and user friendly avionics package that can easily be duplicated for future projects. The ground station software provides an easy and user friendly interface to the avionics package. Table 6.1 summarizes the avionics package specifications.



**Table 6.1:** Avionics Package Specifications

SPECIFICATIONS: AVIONICS PACKAGE		
POWER REQUIREMENTS	Supply Voltage	8.4 [V]
	Supply Current	<500 [mA] (with RF transmission)
	Power Consumptions	< 4.2 [W]
PERFORMANCE	RF Range	5 [km]
	RF Frequency	2.4 [GHz]
	Turn Rate	±60 [deg/s]
	Acceleration	±2 [g]
	Ultrasonic Distance	6 [m]
	Altitude	300 [m]
	GPS Position	4 [m]
	GPS Velocity	0.4 [m/s]
	GPS Heading	2 [deg] <sup>4</sup>
	Magnetometer	±0.625 [gauss]
GENERAL	Weight	460 [g] (no battery)
	Dimensions	180x52x100 [mm]
	GPS antenna	MCX
	Battery connector	Standard Deans
	MaxStream Antenna	MMCX
	Price	±8000 [R]

**Figure 6.28:** Avionics and Ground Station<sup>4</sup> Above 5m/s groundspeed

# Chapter 7

## *Non-linear Simulation (Hardware in the Loop)*

This chapter presents the hardware in the loop (HIL) simulator and the simulation results of the controllers and sequences developed in sections 2.1.1 and 2.2.1, as well as Chapter 4 and Chapter 5. The chapter starts with an overview of the simulator. In section 7.2 the simulation results obtained are presented and discussed. The simulation verifies the controller implementation and evaluates the controller performance under full non-linear conditions, including noise and wind effects. The hardware in the loop simulator is an essential tool to gain confidence in the controller implementation and dramatically reduces the risk during practical testing. The simulator also provides a fair level of confidence that the controllers will be able to perform their desired tasks in practise.

### 7.1 Hardware in the Loop Simulator

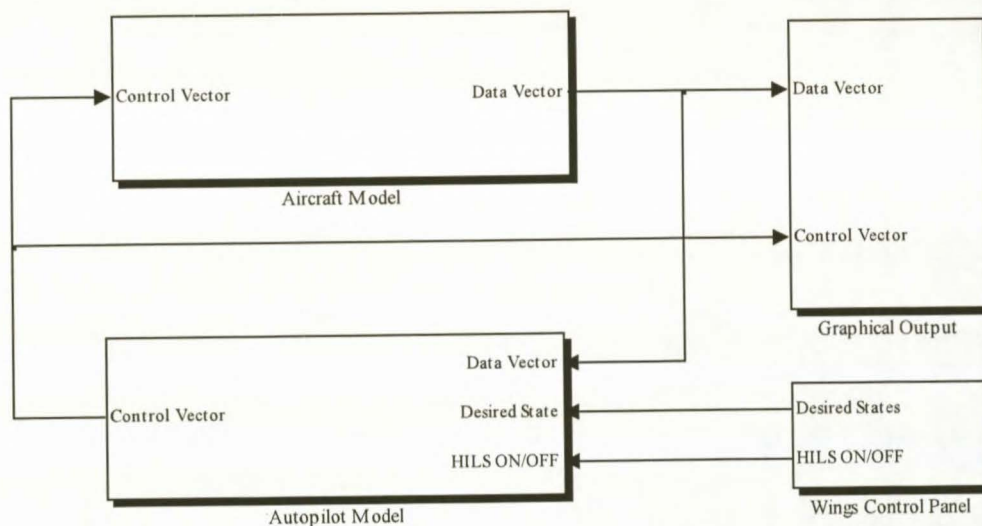
#### 7.1.1 HIL Simulator Overview

In order to thoroughly simulate the controllers and sequences the hardware in the loop (HIL) simulator was developed. The HIL simulator consists of Simulink models of the aircraft, runway, environment, sensors, actuators, an interface for the avionics to communicate with these models and a graphical display output. The interface to the avionics allows the aircraft model to be flown with the RC transmitter and the control algorithms to be armed through the ground station in the exact same manner as they would be out in the field. This ensures a high degree of trust in the control algorithms and ensures the system is thoroughly tested before a flight test is attempted, minimising the risk. Figure 7.1 shows a typical HIL simulation setup.



**Figure 7.1:** *Hardware in the Loop Simulation Setup*

The simulator is based on the block diagram simulator developed in [1]. Figure 7.2 shows the Simulink structure of the hardware in the loop simulator.



**Figure 7.2:** *HIL Simulator Structure*

The *Aircraft Model*<sup>5</sup> uses the actuator commands (control vector) and the current state of the aircraft to propagate the states through to the next Simulink time step. The output of the Aircraft Model is the aircraft states contained in the data vector. The current aircraft states from the aircraft model, as well as the commanded states, serve as input to the *Autopilot Model* where the control algorithms calculate the actuator commands. The control algorithms used to calculate the actuator commands are either the algorithms simulated in Simulink or the algorithms implemented on the avionics. The *Wings Control Panel* serves as a user interface to the simulator. In the control panel the desired states are commanded and the HIL capability is switched. The *Graphical Output* block receives the aircraft states along with the actuator commands and uses it to display the aircraft in an OpenGL three-dimensional (3D) environment.

<sup>5</sup> The first instance of each simulator block is written in italic font for easy identification.



## 7.1.2 Aircraft Model

The aircraft model calculates the next state of the aircraft based on the current state and the actuator commands to the model. The aircraft states are calculated by a six degrees of freedom (6DoF) block that contains the six degrees of freedom dynamic equations. The forces and moments supplied to the 6DoF block are generated by five blocks, each modelling forces and moments from a specific source. The five models are:

- Aerodynamic model
- Runway model
- Engine model
- Gravitational model
- Wind model

The runway, engine and wind models were developed in Chapter 3 and the aerodynamic and gravitational models were obtained from [1].

## 7.1.3 Autopilot Model

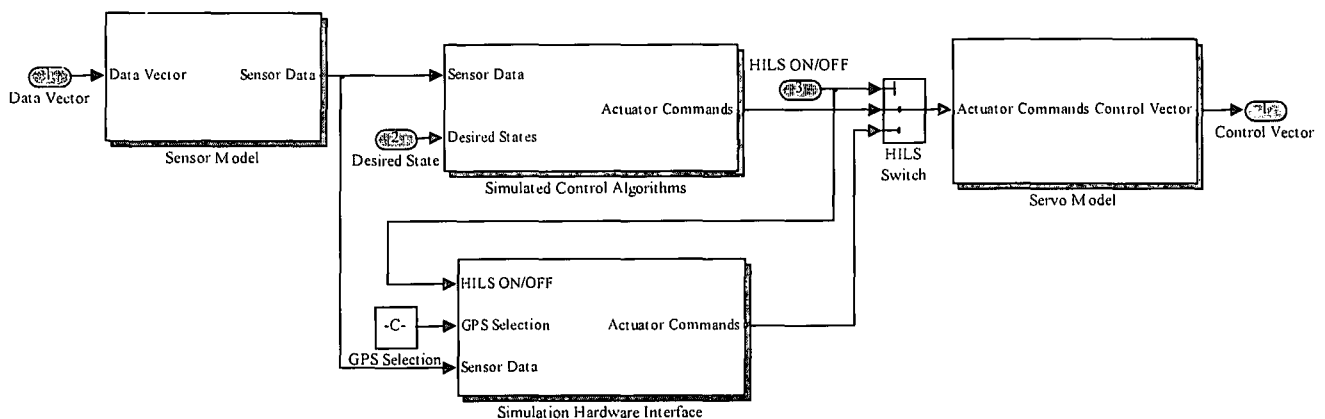
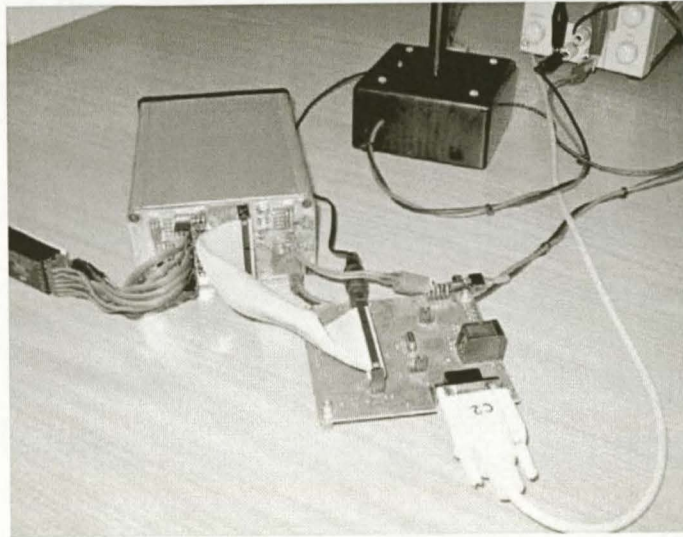


Figure 7.3: Autopilot Model for HIL Simulator

Figure 7.3 shows the expansion of the autopilot block in Figure 7.2. The autopilot model includes the sensor models, control algorithms and servo model. The block inputs are the aircraft states and the desired states. The output is the servo positions calculated by the control algorithms. The *Sensor Model* converts the aircraft states to equivalent sensor measurements and adds noise to the measurements. The noise figures used in the simulator are summarised in section C.3. The simulated sensor data is then used by the control algorithms to calculate the actuator commands. A software switch selects either the actuator commands from the simulated control algorithms implemented in Simulink or the control algorithms running on the avionics. The *Simulation Hardware Interface* (SHI) block was developed in [22]. The SHI block packages the sensor data and writes it out via the serial RS232 port of the computer to the HIL board. The HIL board then decodes the data and writes it out to the avionics package through the HIL interface multiplexers discussed in section 6.1.7. The avionics reads the sensor data, calculates the servo commands and writes it back to the HIL board. The board then packages the servo data and sends it back to Simulink via the serial RS232 port completing the loop. Figure 7.4

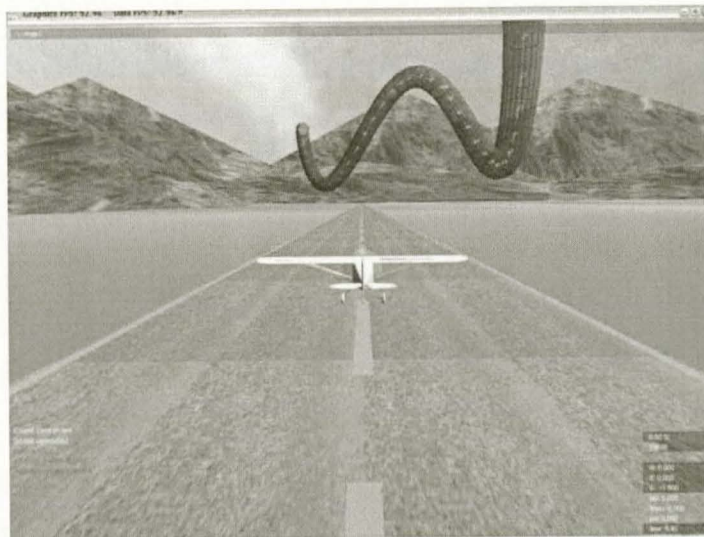
shows the avionics connected to the HIL board with its serial RS232 connection to a personal computer. The *Servo Model* block models the servo dynamics.



**Figure 7.4:** *Avionics Connected for HILS*

#### 7.1.4 Graphical Output

An OpenGL graphical display package for an aircraft was developed in [23]. This display package uses the aircraft states (position and Euler attitude) to display the aircraft in a 3D environment. The 3D environment is excellent to see the controllers at work and makes it possible to observe controller responses in real time. The graphics engine can run on the same or another computer as the one running the HIL simulation. In both cases the states required for display is sent to the graphics engine via a local area network (LAN) using the user data protocol (UDP). The display package includes functionality to draw 3D paths, place waypoints and display aircraft states on graphs and in text boxes, making it a very powerful tool for development and evaluation. A screenshot from the display package is shown in Figure 7.5 with the aircraft on the runway and a 3D graphics object shown in the distance.



**Figure 7.5:** *Screenshot from the Graphical Display*

In this project the Simulink interface to the display package was developed and is shown as the *Graphical Output* block in Figure 7.2. The graphical output block packages the states and other data to be displayed and sends it via the UDP protocol over the network using the code provided by the display package.

## 7.2 Controller Non-Linear Simulation

The controllers designed in Chapter 4 and Chapter 5, as well as the take-off and landing sequences discussed in section 2.1.1 and 2.2.1 are simulated with the HIL simulator to prepare the controllers and avionics for flight tests. The HIL simulation is essential to verify the controllers and quantify effects like:

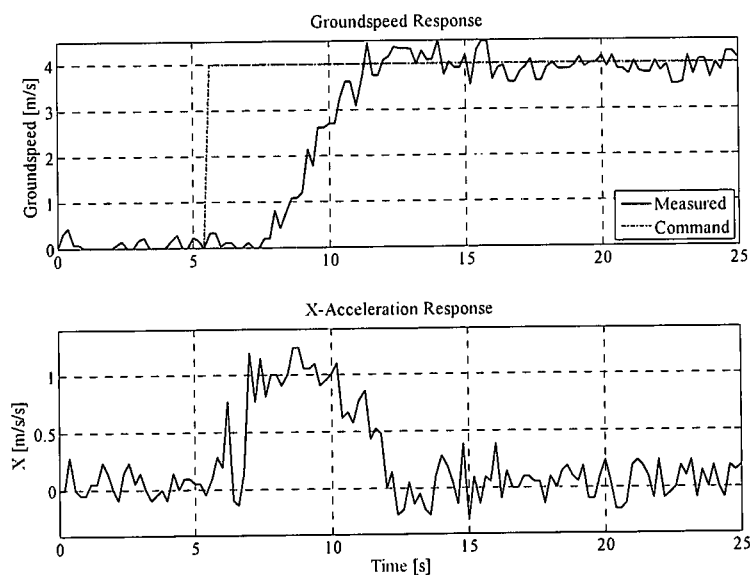
- Sensor noise
- Microprocessor quantisation
- Wind
- Actuator delay and saturation
- Longitudinal and lateral coupling
- Implementation correctness
- Severity of linearisation assumptions

The RMS noise value used for each sensor in the HIL simulator is summarised in Table C.3.

### 7.2.1 Take-Off Controllers

#### Acceleration and Groundspeed Controllers

Figure 7.6 shows the response of the groundspeed controller with inner acceleration loop to a 4m/s step command.



**Figure 7.6:** Acceleration and Groundspeed Controller HIL Simulation

The step response is very similar to the linear response shown in Figure 4.6, with a slightly faster rise time of approximately four seconds. The slight steady state error in the response is due to the small pitch angle of the

aircraft on the runway. The pitch angle causes the  $X$ -axis accelerometer to measure a portion of the specific force acting on the aircraft from the runway. Note the delay in controller response to a step command. This delay is a direct effect of the dead band present in the engine model as can be seen from Figure 3.8.

### Runway Yaw Rate Controller

Figure 7.7 shows the step response of the runway yaw rate controller to a 30deg/s step command at 3m/s airspeed.

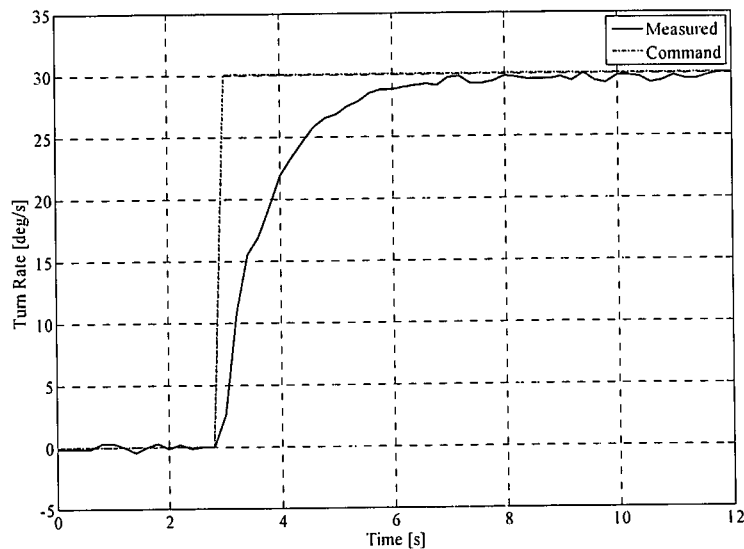


Figure 7.7: Runway Yaw Rate Controller HIL Simulation

The HIL response is very similar to the linear response shown in Figure 4.12. Note the 2s rise time as designed for and the classical first order system response. The fact that the first order response is also seen in the full non-linear simulation justifies the linearisation assumptions made in section 3.2.

### Radius Interface

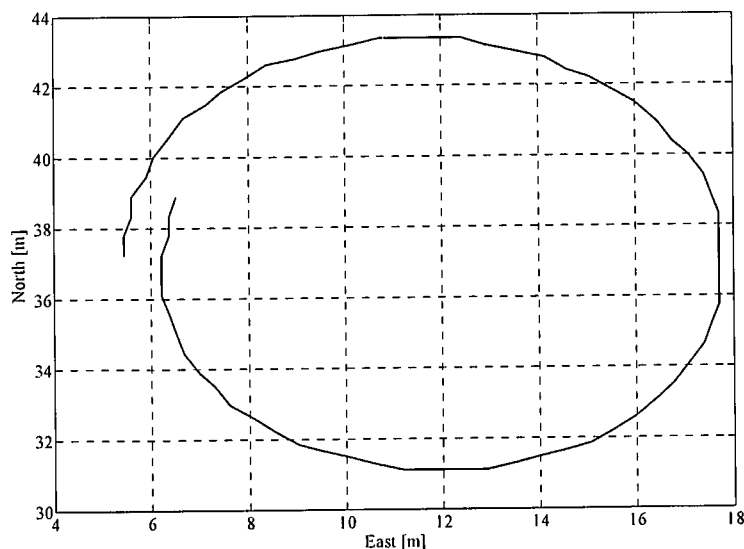


Figure 7.8: Radius Interface HIL Simulation



Figure 7.8 shows the two-dimensional position response of the radius interface to a 6m radius command at 2m/s. GPS position noise was disabled to clarify the position response. The desired circle with a radius of 6m is clear from the response.

### Runway Heading Controller

Figure 7.9 shows the step response of the runway heading controller to a  $90^\circ$  step command at 3m/s airspeed. The turn (yaw) rate associated with the heading step is also shown.

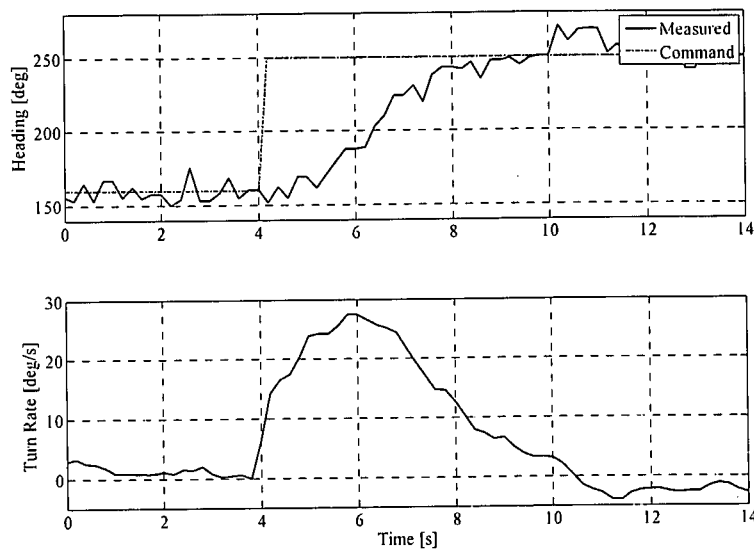


Figure 7.9: Runway Heading Controller HIL Simulation

The GPS receiver heading measurement is poor at low forward velocities with a variance of approximately six degrees. Band limited white noise with a variance of eight degrees was added to the simulation. It is clear that all the design specifications are still satisfied in spite of the noise. The four second rise time and slight overshoot corresponds very well with the linear response.

### Runway Guidance Controller

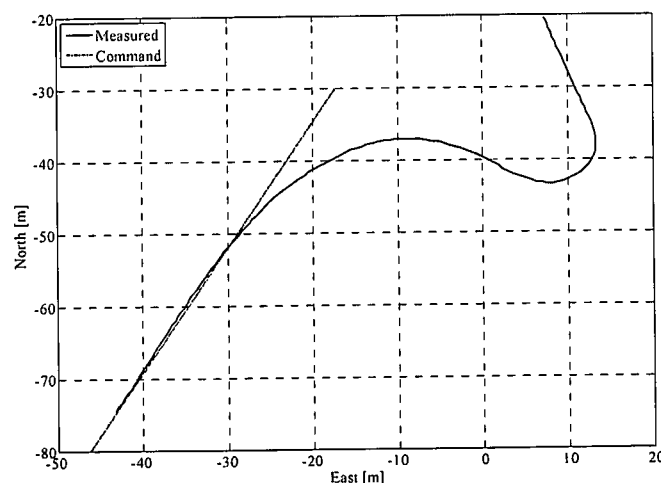


Figure 7.10: Runway Guidance Controller HIL Simulation

Figure 7.10 shows the two-dimensional position response of the runway guidance controller at 3m/s, while Figure 7.11 shows the heading and yaw rate responses of the aircraft while tracking the desired path. Note the excessive heading noise in Figure 7.11. The aircraft is travelling in a heading of 160° when the path to track is commanded. The non-linear response again corresponds excellently with the linear response of Figure 4.21. For the simulation GPS position noise was disabled so the position response is better visible. GPS position noise will only cause an offset in the position that will slowly drift around with time. The excessive heading noise does have a slight influence on the yaw rate response of the aircraft, but it is still well within acceptable limits.

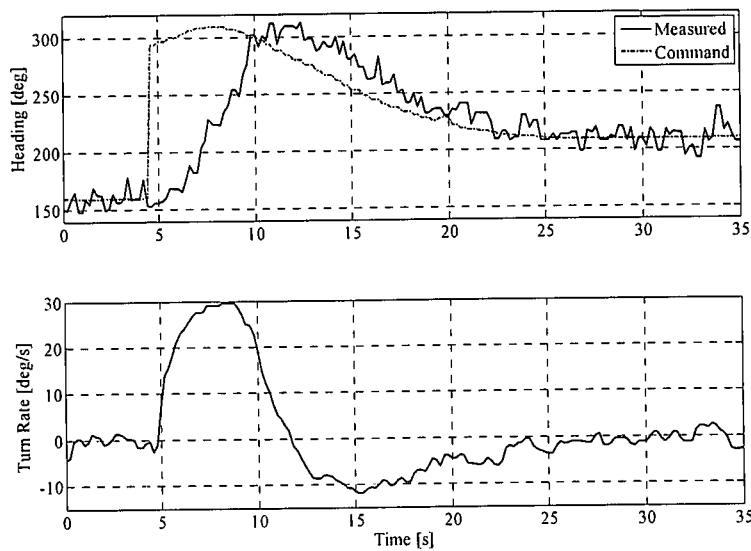


Figure 7.11: Heading and Turn Rate during Guidance Controller HIL Simulation

### 7.2.2 Take-Off Sequence

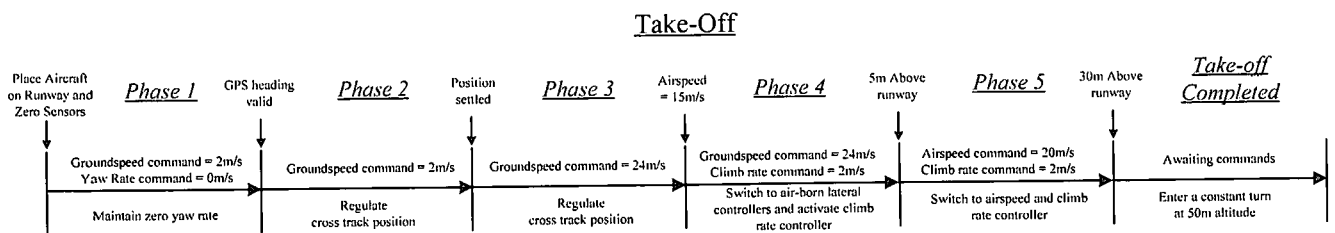


Figure 7.12: Take-Off Sequence Phases (Repeat)

The take-off sequence time line is repeated in Figure 7.12 for convenience. A screenshot from the 3D graphical output shortly after take-off is shown in Figure 7.13. Figure 7.14 to Figure 7.16 shows the controller commands and simulated system responses during the take-off sequence. The sequence is performed without any constant wind or wind gust disturbances.

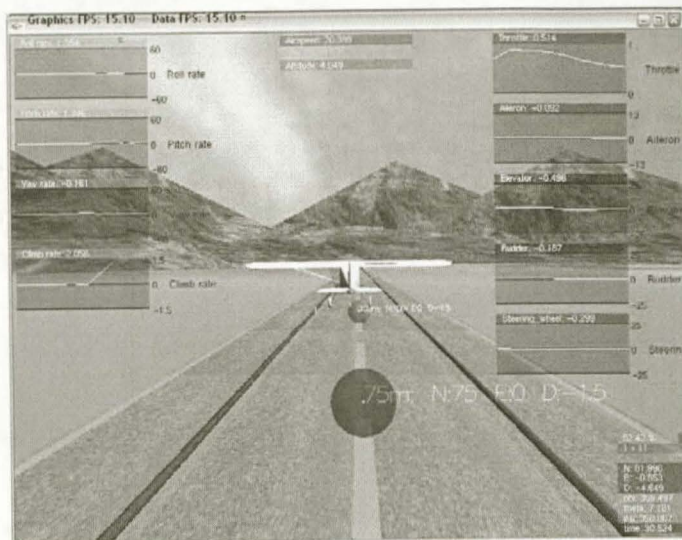


Figure 7.13: Screenshot of the Aircraft shortly after Take-Off

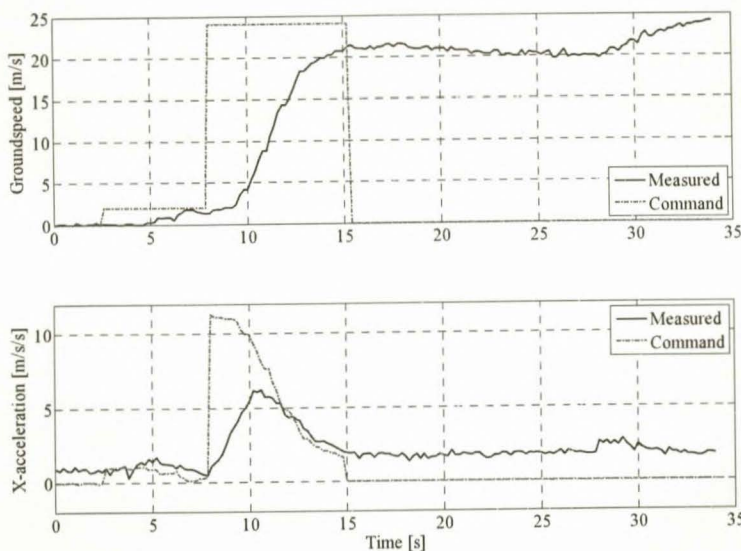


Figure 7.14: Groundspeed Response during a HIL Take-Off

The take-off sequence starts in phase 1 by commanding zero yaw rate and 2m/s groundspeed, corresponding to the first step in the top half of Figure 7.14. Again note the lag in response due to the dead band in the engine model. When the GPS heading is valid, phase 2 is entered and the guidance controller is armed to guide the aircraft down the runway. Lateral control during the take-off is shown in Figure 7.16. When the position of the aircraft has settled phase 3 is entered and the groundspeed command is step up to 24m/s in Figure 7.14, starting the ground roll.

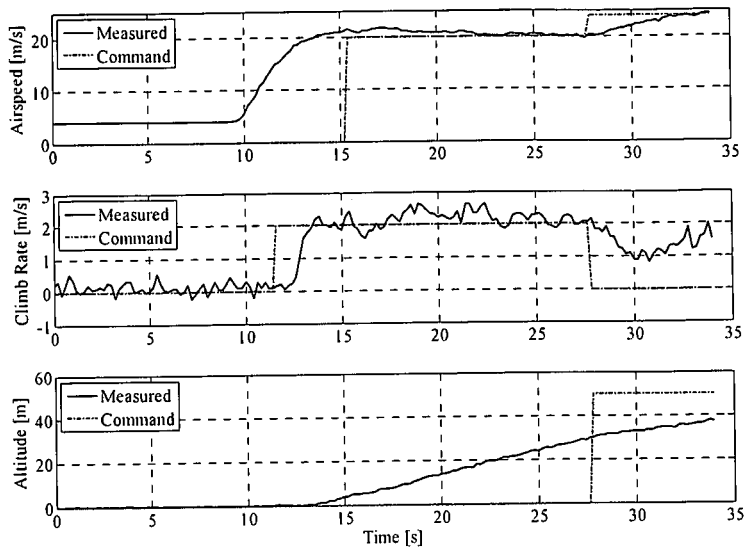


Figure 7.15: Longitudinal Control during a HIL Take-Off

Once the take-off airspeed of 15m/s is reached at approximately twelve seconds, phase 4 is entered where the climb rate controller is armed with a 2m/s command. The step command in climb rate can be seen in the centre graph of Figure 7.15 at approximately 12s. At this point the aircraft rotates and starts to climb. Lateral control is also switched from the runway controllers to the air-born controllers.

When the aircraft reaches an altitude of 5m, the airspeed and climb rate controller designed in [1] replace the groundspeed and climb rate controllers. This transition marks the start of phase 5 and takes place at approximately fifteen seconds. The step in airspeed can be seen in Figure 7.15, with the climb rate command still at 2m/s. When an altitude of 30m is reached, the take-off is considered complete and the aircraft is commanded to enter a constant turn at 50m altitude. The arming of the altitude controller and the step in altitude command can be seen in Figure 7.15 at approximately 27s.

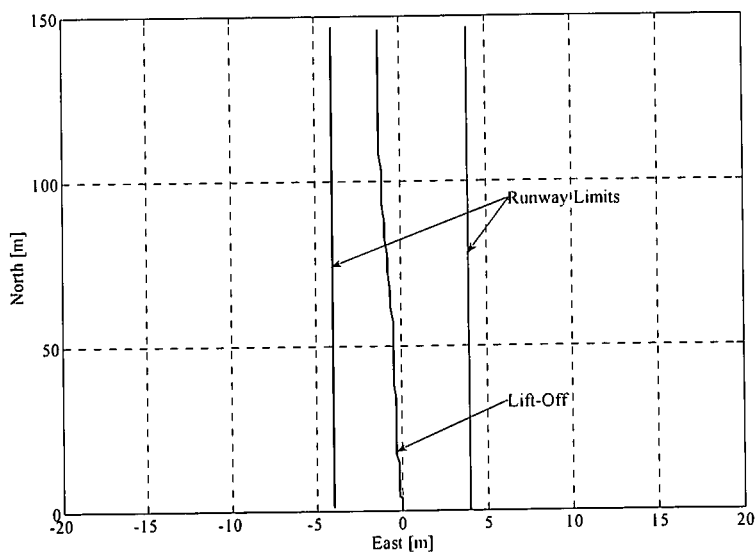


Figure 7.16: Lateral Control during a HIL Take-Off

Figure 7.16 shows the two dimensional GPS position of the aircraft during take-off. The runway limits for a runway width of 8m and the point of take-off is also shown. The lateral controllers keep the aircraft on and over

the runway during the entire take-off sequence. The take-off point in Figure 7.16 shows the aircraft took off at 20m north. The take-off sequence started at 0m north. This translates into a minimum runway length of 40m required for a safe take-off with no wind.

### Crosswind Take-Off

Figure 7.17 and Figure 7.18 shows the simulated results for the take-off sequence being performed in a crosswind of 3m/s (10.8 km/h) at a 50° angle to the runway (the runway is north aligned). A crosswind of 3m/s is a significant disturbance if the size of the aircraft used in this project is considered. Wind gusts as the one shown in Figure 3.10 also act on the aircraft during the take-off sequence.

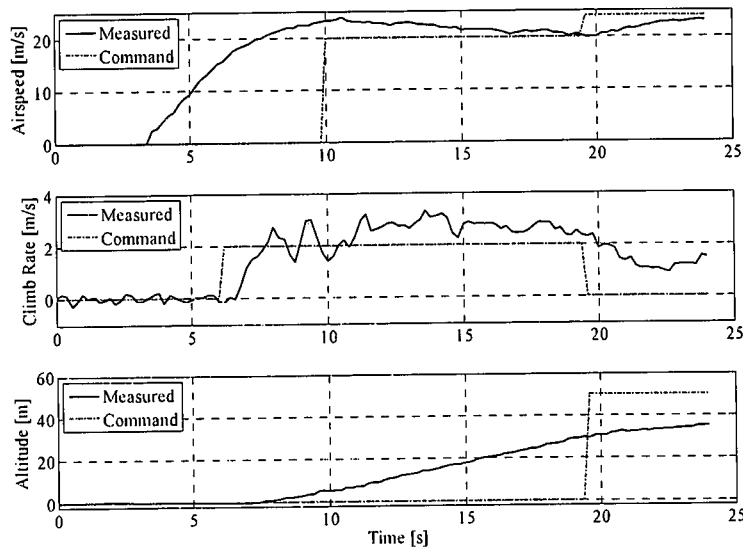
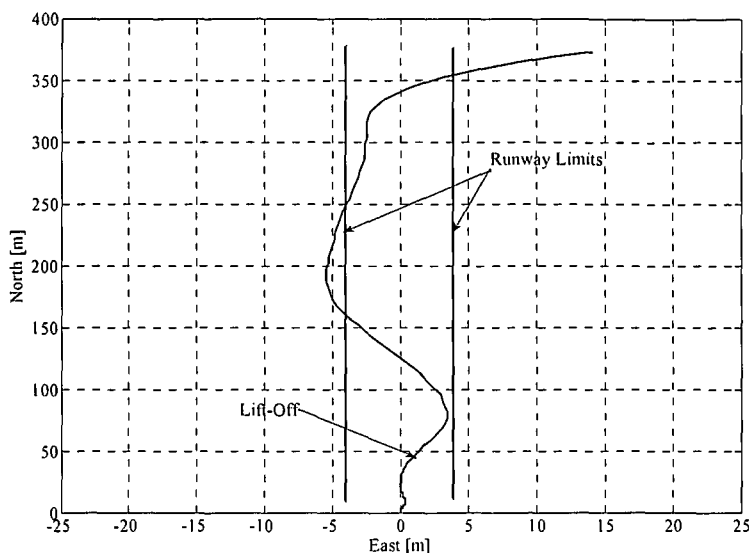


Figure 7.17: Longitudinal Control during a Crosswind HIL Take-Off

Figure 7.17 shows the longitudinal responses of the aircraft during a crosswind take-off. The responses are very similar to the ones obtained during a no-wind take-off, shown in Figure 7.15. However, there is an increase in airspeed (groundspeed is the variable controlled) during the crosswind take-off. The climb rate also increase when the airspeed and climb rate controller is armed at 25s. This is due to the airspeed and climb rate controller issuing a negative elevator command to increase the pitch angle of the aircraft to regulate the airspeed down to the command 20m/s. It is this elevator command that causes the aircraft to climb.

Figure 7.18 shows the lateral GPS position of the aircraft during the crosswind take-off. During the first three phases (position settling and ground roll) the position of the aircraft is very well regulated by the runway guidance controller in spite of the strong wind disturbance. From the GPS position response it can be seen that the aircraft drifts to the right just before take-off. This is due to the nose wheel losing effectiveness just before take-off, causing the aircraft to weathercock into the wind.

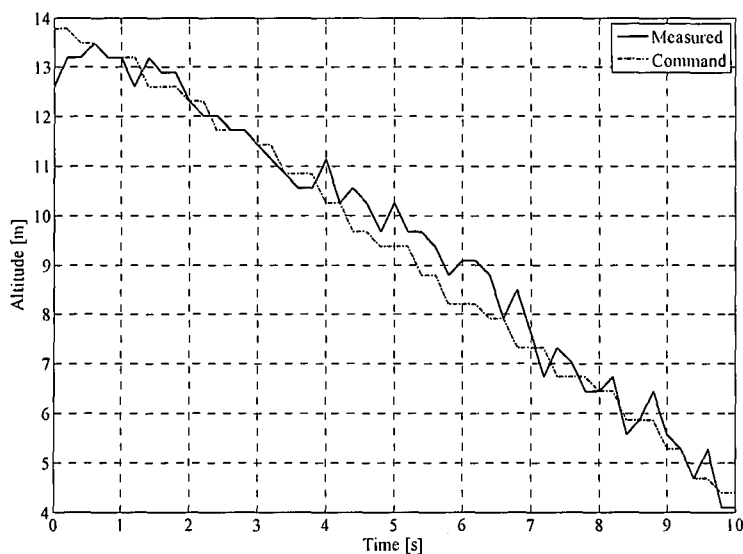


**Figure 7.18:** Lateral Control during a Crosswind HIL Take-Off

The yaw orientation control system is armed just after take-off and pulls the aircraft back to the runway track. The crosswind then causes the aircraft to overshoot the runway track as is clear from the GPS position plot. The guidance controller brings the aircraft back over the runway, but the constant wind causes an offset of approximately 3m in lateral position. The constant turn at the end of the take-off sequence is entered at 34s.

### 7.2.3 Landing Controllers

#### Glide Path Controller

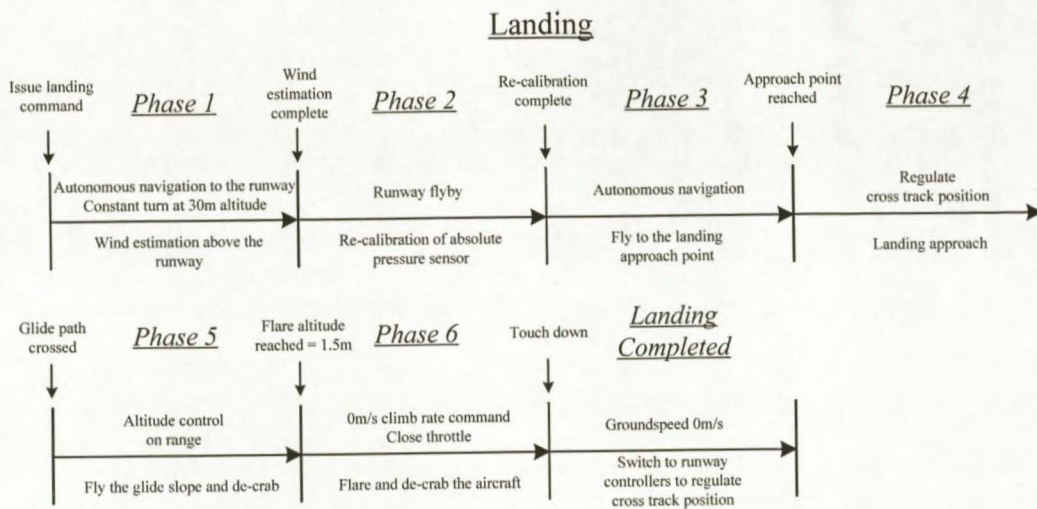


**Figure 7.19:** Glide Path Controller HIL Simulation

Figure 7.19 shows the response of the glide path controller. The response is very similar to the linear response obtained. The offset in altitude from the commanded value from 4s to 7s was most likely caused by a wind gust. The controller recovers very well and continues to track the commanded glide path.

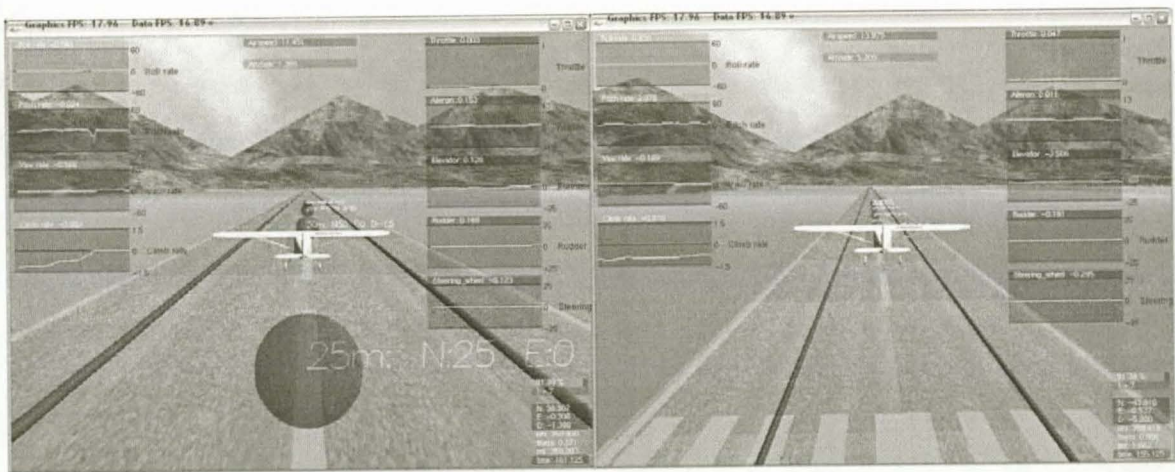


### 7.2.4 Landing Sequence



**Figure 7.20:** Landing Sequence Phases (Repeat)

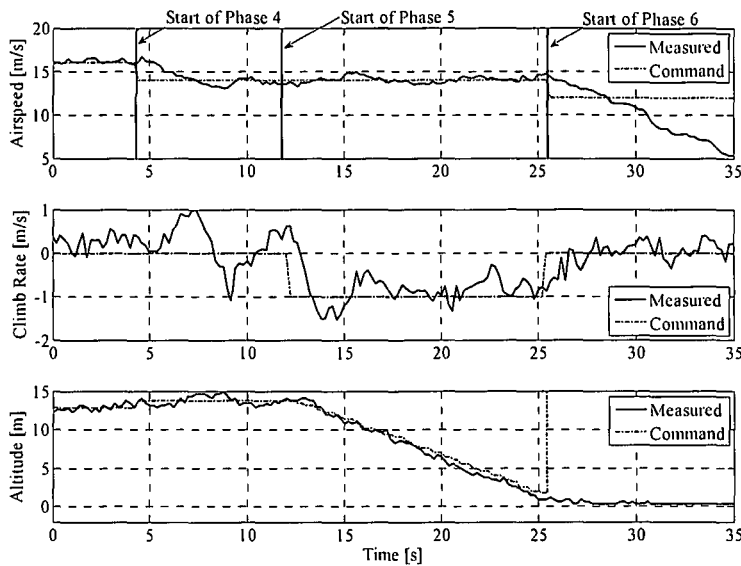
The landing sequence time line is repeated in Figure 7.20 for convenience. Figure 7.22 to Figure 7.25 shows the controller commands and simulated system responses during the landing sequence. The sequence is performed with crosswind gust disturbances as the one shown in Figure 3.10 at an angle of 50° to the runway (the runway is north aligned). Figure 7.21 shows screenshots from the 3D graphical output shortly after landing and the final moments of the landing approach.



**Figure 7.21:** Screenshots of the Aircraft just After Landing (Left) and on the Landing Approach (Right)

The first two phases of the landing sequence presented in Figure 7.20 does not form part of the simulation. The landing path parameters used during the simulation are summarised in section B.6.



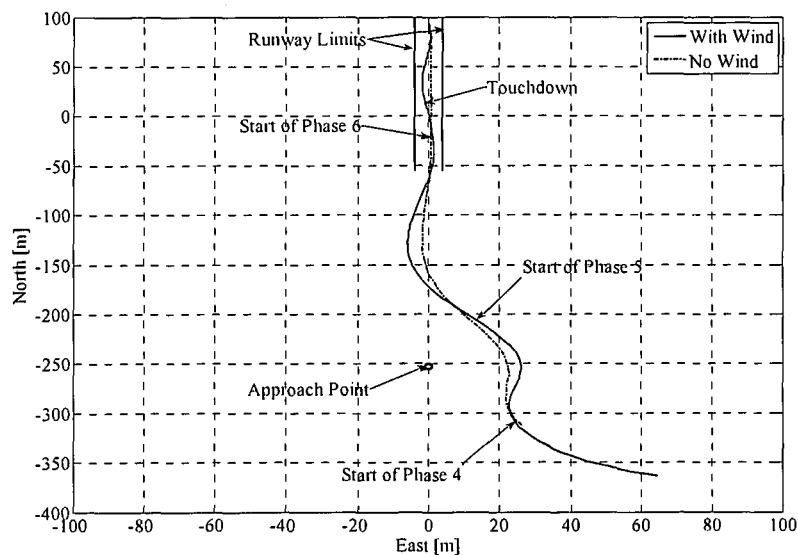


**Figure 7.22:** Longitudinal Control during a HIL Landing

The presented simulation data starts in phase 3 during the final moments of navigation at an airspeed of 16m/s and an altitude of 13m, as can be seen in Figure 7.22. Figure 7.23 shows the lateral position of the aircraft during landing along with the start of every phase. From Figure 7.23 it is clear that the system switches to phase 4 just before the approach point is reached and starts to regulate the cross track position off the runway.

During phase 4 an airspeed command of 12m/s is issued to reduce the kinetic energy of the aircraft. When the glide path is crossed (just after 12s) the system enters into phase 5 and starts to track the glide path. The airspeed command is slightly increased to 14m/s. This is done to achieve a sink rate of 1m/s as indicated by equation (5.6) for a glide slope angle of 4°.

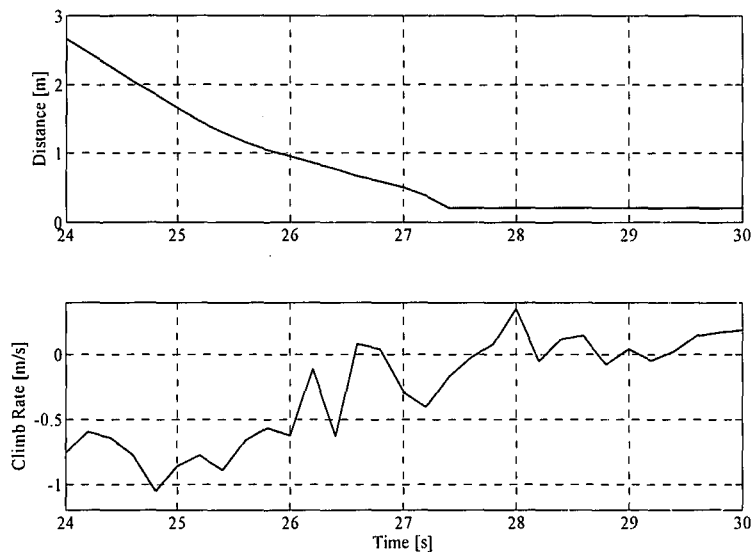
Figure 7.23 shows the simulated lateral response with and without wind gust disturbances. Note how the runway guidance controller pulls the aircraft back on to the runway track after touchdown.



**Figure 7.23:** Lateral Control during a HIL Landing

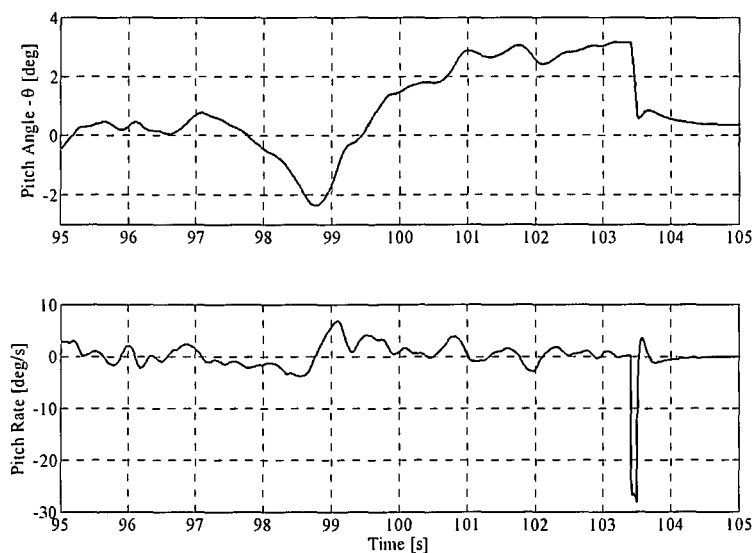
The system enters phase 6 when the aircraft reaches an altitude of 1.5m. In phase 6 the airspeed and climb rate controller is replaced by the climb rate controller and the throttle is closed. A 0m/s climb rate command is issued to flare the aircraft, reducing the sink rate and ensuring a positive pitch angle at touch down. Figure 7.24 shows the ultrasonic altitude and pressure climb rate during the phase 6 flare.

Touchdown occurs at approximately 27.5 seconds. Note the sink rate increasing from -1m/s to almost 0m/s at touchdown. The flare shown in Figure 7.24 will be considered excellent by any human pilot.



**Figure 7.24:** *Ultrasonic Altitude and Pressure Climb Rate during a HIL Flare*

Figure 7.25 shows the pitch angle ( $\theta$ ) in degrees and the pitch rate in degrees per second during the flare. The pitch angle increases to 3 degrees during the flare and the negative spike in pitch rate indicates that the main wheels touched down first. Figure 7.25 was generated from simulation output data and not telemetry data as all the other responses since pitch angle data is not available in the telemetry stream. Therefore the time scale in Figure 7.25 has no relation to the time scales of the other landing sequence graphs.



**Figure 7.25:** *Pitch Angle and Rate during HIL Flare*

## 7.2.5 Wind Estimation

Table 7.1 shows the simulated results of the wind estimation procedure, providing repeated simulation results for three different headings and wind speeds. The results were obtained with only a constant wind present i.e. no wind gusts.

**Table 7.1: Wind Estimation Results**

ACTUAL VALUES			ESTIMATED VALUES			ERRORS		
Heading [deg]	North Velocity [m/s]	East Velocity [m/s]	Estimated Heading [deg]	Estimated North Velocity [m/s]	Estimated East Velocity [m/s]	Heading Error [deg]	North Velocity Error [m/s]	East Velocity Error [m/s]
35°	-3.2	-2.28	39.4°	-3.2	-2.6	4.4°	0	-0.3
35°	-3.2	-2.28	35°	-3.1	-2.1	0°	0.1	0.2
35°	-3.2	-2.28	33.3°	-3	-2	-1.7°	0.2	0.3
120°	2	-3.43	120.3°	2.1	-3.5	0.3°	0.1	-0.1
120°	2	-3.43	119.7°	2	-3.5	-0.3°	0	-0.1
120°	2	-3.43	118.4°	1.7	-3.2	-1.6°	-0.3	0.2
120°	2	-3.43	122.4°	2	-3.2	2.4°	0	0.2
330°	-3.43	1.98	335.4°	-3.6	1.6	5.4°	-0.2	-0.4
330°	-3.43	1.98	325°	-3.4	2.4	-5°	0	0.4
330°	-3.43	1.98	330°	-3.5	2	0°	-0.1	0
Average Errors						2.11 °	0.1	0.27

The results in Table 7.1 are extremely good. These results can be attributed to the zero mean sensor noise used in the simulation. The noise effectively cancels out during the long wind estimation procedure. Wind gusts will be smoothed out by the wind estimation procedure in both the simulation environment and practice. The absence of wind gusts will therefore not have a dramatic effect on the simulated results. The practical results are expected to be far less accurate than those obtained from simulation. Typical results expected are:

- Heading variance =  $\sigma_{\psi} = 10^{\circ}$
- North velocity variance:  $\sigma_N = 1m/s$
- East velocity variance:  $\sigma_E = 1m/s$

The accuracy of the wind estimation procedure is however not crucial due to the 180° error bounds associated with a landing direction decision.

## 7.3 Summary

This chapter discussed the hardware in the loop simulator and presented the simulation results for the controllers and sequences developed. The simulator is an essential tool to gain confidence in the implementation correctness and performance of the controllers. With the controllers and system thoroughly simulated, the system is ready for practical testing.

# Chapter 8

## *Flight Tests*

Throughout the development of the project the system was subjected to, and verified through practical testing. The successful autonomous take-off and landing of the Reliance 0.46 size model aircraft was demonstrated.

Section 8.1 provides an overview of the practical testing that built the system to the point of autonomous take-off and landing. Section 8.2 evaluates the data obtained from the low cost GPS and the newly added ultrasonic range sensor. The magnetometer was not flown as it is not used and is an expensive piece of equipment to risk unnecessarily. Section 8.3 presents all the data for the take-off controllers and sequence. Section 8.4 presents the data obtained for autonomous landing. The autonomous landing data includes the gain scheduled responses of the conventional flight controllers, the glide path tracking, the landing sequence and the flare response.

All the telemetry data presented are the states of the aircraft as measured by the onboard sensors.

### 8.1 Flight Tests Overview

Instead of subjecting the complete system to vigorous flight testing at the end of the project, a total of five flight tests were conducted as the system developed. Each consisted of several flights. This approach was adopted to identify any hardware or controller constraints and problems early on to avoid unwanted surprises. The approach also provides confidence in the subsystems developed up to the last flight test and allows for more informed decisions to be made on the future of the project. However, this approach introduces a risk that the aircraft and hardware can be lost early on in the project and consequently severely hinder progress. The HIL simulator, a detailed flight test plan and a checklist are essential tools to minimise the risks during a flight test. A typical flight test plan is given in Appendix G.

All of the flight tests were conducted at Helderberg Radio Flyers near Somerset West with Dr. Kas Hamman, the chief flying instructor as the safety pilot.



**Figure 8.1:** *The Project Team*

The five flight tests tested:

1. The redesigned avionics package and sensors.
2. Conventional flight controllers, sensor validation and wind estimation.
3. Runway controllers and landing approaches ending in runway flybys.
4. Autonomous taxiing and lateral assisted take-off.
5. Autonomous take-off and landing.

### 8.1.1 Flight Test 1 – Redesigned Avionics

The first flight test of the project was conducted after the avionics package was extended and repackaged. It established a hardware and experience base from which to build. The flight test verified that the aircraft is able to fly with the new avionics package and allowed the safety pilot to familiarise himself with the system. All the sensors, with their associated signal conditioning required for conventional flight and autonomous navigation, were tested and found to work. The flight test also served as an opportunity to draw up and round off a detailed flight test plan and checklist to ensure the smooth execution of future flight tests.

### 8.1.2 Flight Test 2 – Conventional Flight Controllers, Sensors Validation and Wind Estimation

The first objective of the second flight test was to verify the conventional flight controllers on the new avionics and to subject the system to more flight hours. The gain scheduling of the conventional flight controllers were also tested, with very satisfactory results.

The second objective of the test was to verify the suitability of the sensors for high precision tasks such as landing. The verification specifically focused on the GPS and absolute pressure sensor accuracy and drift. It was found that the GPS accuracy is just sufficient for the lateral guidance of the aircraft during both the take-off and the landing. Thus no additional sensors for guidance are required on the runway. The absolute pressure



sensor was found to be accurate enough to guide the aircraft to the runway, but displays a significant amount of drift due to temperature.

The ultrasonic range sensor was also evaluated to determine its accuracy on a moving platform and on different ground surfaces. The ultrasonic provided very good measurements on both grass and concrete surfaces. Based on the good measurements the ultrasonic sensor was chosen to recalibrate the absolute pressure sensor in order to remove the temperature drift just before the landing approach is started.

The third goal of the flight test was to test and evaluate the wind estimation procedure. Since there is no accurate reference of wind speed and direction available, it is difficult to determine the exact performance of the estimation procedure. The obtained results did however correspond very well with the judged wind speed and direction on the day.

### 8.1.3 Flight Test 3 – Runway Controllers and Landing Approaches

The third flight test tested the runway controllers, as well as the landing sequence and glide slope altitude response.

Initially, the runway controllers provided some problems. The yaw rate controller work very well for anti-clockwise turning commands (negative yaw rate), but was unstable for clockwise turning commands. The command dependent instability can be attributed to the very non linear nose wheel actuator. Since the integrator in the runway yaw rate controller is responsible for pushing out the non-linear effects, the integrator pole position was moved (speed of response decreased) until the system was stable. With a stable and working runway yaw rate controller in place, the responses obtained from the higher level heading and guidance controllers were excellent.

The groundspeed controller also provided some initial problems by having a large steady state error. This is due to the misalignment of the x-accelerometer, causing it to measure a component of the specific force acting on the aircraft from the runway. After the misalignment was corrected and compensated for, the groundspeed controller responded as predicted.

The landing sequence was performed with a 20m altitude offset and was ended just before the flare (phase 6). Instead of the flare the altitude was held at 20m and the aircraft performed a runway flyby. The main goal of these tests was to evaluate the response and accuracy of the glide path tracking, as well as evaluating the lateral position responses and accuracy during the landing approach. The landing approaches also confirmed the correct execution of the landing sequence.





**Figure 8.2:** Assistant Willem Hough and the Author Refuelling

#### 8.1.4 Flight Test 4 – Autonomous Taxing and Lateral Assisted Take-Off

With all the individual controllers verified and the initial test of the landing sequence completed, the focus shifted to the take-off sequence. During flight test four all of the runway controllers were strung together in order to perform autonomous taxing down the runway. The controllers performed excellently, resulting in very satisfactory autonomous taxing results and established the first stepping stone towards autonomous take-off.

With the autonomous taxing verified, a lateral assisted take-off was performed. During the lateral assisted take-off the pilot controlled the throttle and elevator actuators, while the autopilot controlled the nose wheel, rudder and aileron actuators. The lateral assisted take-off was performed very successfully, verifying the lateral control, sections of the take-off sequence and the smooth transitioning between controllers during a take-off.

Unfortunately the positive results came to an abrupt end when the pilot lost control of the elevator due to a glitch in the servo time in code. The glitch was triggered by the RF link to the aircraft being interrupted, a problem that would never have been found in the HIL simulator. The avionics did not suffer any damage and the aircraft was quickly repaired back in the laboratory, with the problem corrected so that flight testing can continue.

#### 8.1.5 Flight Test 5 – Autonomous Take-Off and Landing

Flight test five demonstrated autonomous take-off and landing. The flight test started by retesting the conventional flight controllers and autonomous taxing after the crash of flight test four.

With the controllers reverified the take-off and landing was attempted. Four successful autonomous take-offs and two successful autonomous landings were performed during the test. Successful autonomous take-off and landing in the same flight was also demonstrated.

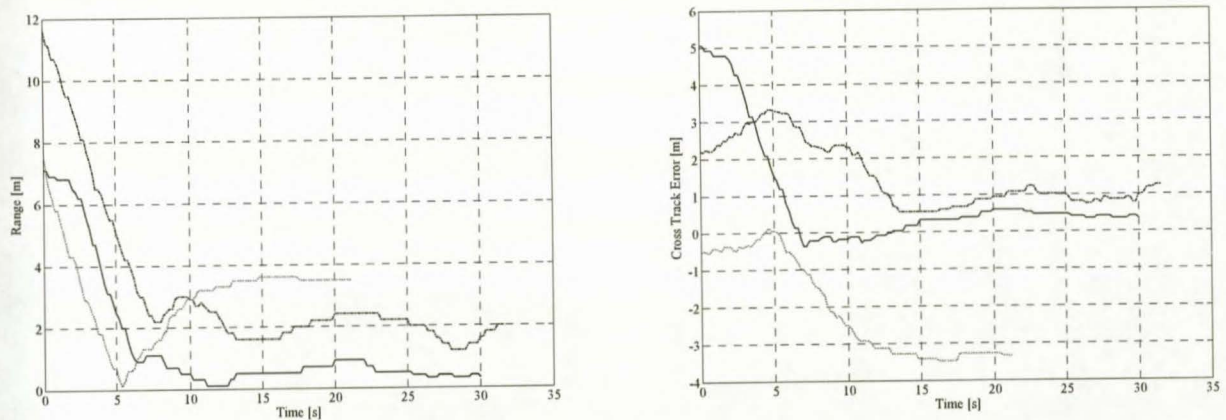


**Figure 8.3:** Safety Pilot Kas Hamman and Assistant Willem Hough during Pre-Flight Checks

## 8.2 Sensor Evaluation

### 8.2.1 Global Positioning System Receiver

The GPS accuracy (drift) after a flight was evaluated by placing the aircraft on the runway and marking the location where the GPS is zeroed. After the flight the aircraft was returned to the zero point on the runway. Figure 8.4 shows the GPS range and cross track error measurements after the aircraft has been returned to the zero point for three different flights.



**Figure 8.4:** GPS Cross Track Error and Range after Three Flights

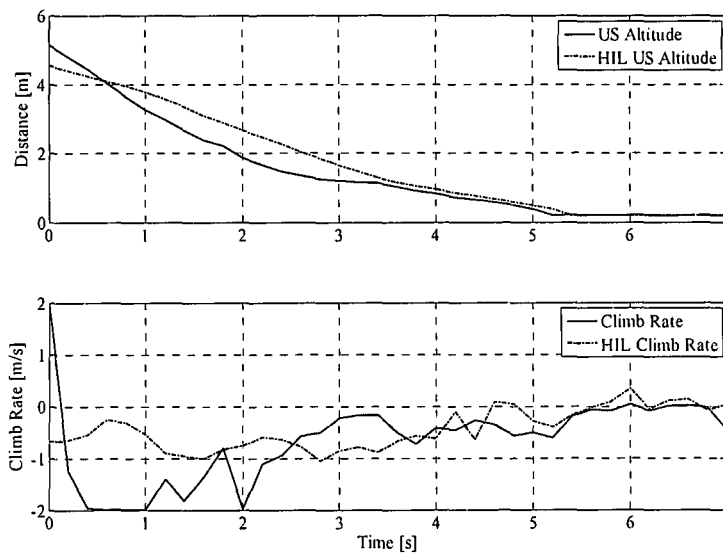
The GPS position error is seen to be up to 3.5m, translating into a runway width of 10m required for landing. However, the airborne guidance controller will display a steady state error in cross track position due to gyroscope biases of up to 4m (see simulation results in Figure 5.9). This increases the required runway width to 20m. Possible solutions to reduce the required runway width are discussed as part of the recommendation made in section 9.2.



## 8.2.2 Ultrasonic Range Sensor

The correct functionality of the ultrasonic range sensor was verified by performing several manual runway flybys and manual landings with the ultrasonic enabled. The goal was to determine if the sensor provides a reliable reading when used on a moving platform and on different ground surfaces. The ultrasonic was found to provide very reliable and accurate measurements from a moving platform, with no visible difference between measurements on grass and concrete surfaces.

Figure 8.5 shows the measured altitude and climb rate from the ultrasonic sensor during a manual landing. The simulated altitude and climb rate from a HIL landing is also shown in Figure 8.5 for the purpose of comparison.



**Figure 8.5:** *Ultrasonic Altitude and Climb Rate during a Manual Landing*

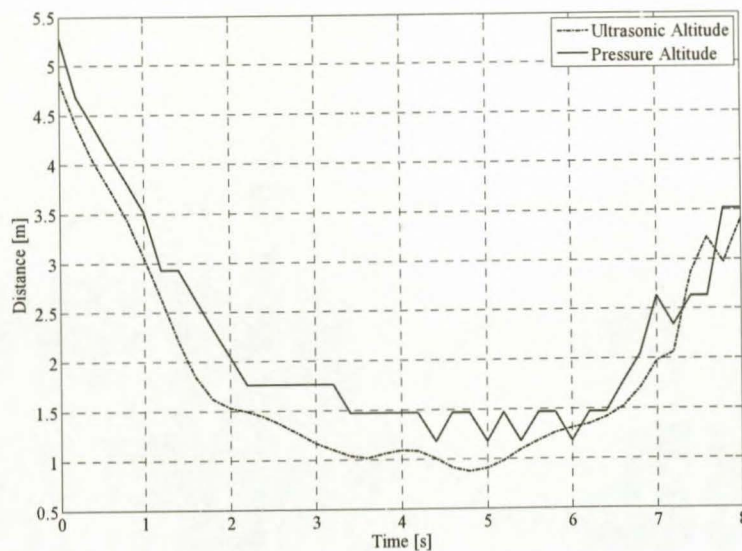
It is clear that the ultrasonic provides a smooth and noiseless altitude measurement. The simulated landing corresponds very well with the manual landing, especially during the final two crucial seconds. The ultrasonic climb rate is obtained by differentiating the altitude measurement and can provide a very accurate climb rate measurement. However, the measurement does display climb rate spikes as can be seen at 2s. For this reason the ultrasonic climb rate is not used in the project. Figure 8.7 shows the ultrasonic measurement during a runway flyby.



**Figure 8.6:** Aircraft during a Manual Landing

### 8.2.3 Absolute Pressure Sensor

The system relies on the absolute pressure sensor for an altitude measurement during landing. The sensor is used to guide the aircraft to within 1.5m of the ground. A fairly accurate measurement is required from the absolute pressure sensor in order to achieve this. The measurement from the absolute pressure sensor during a runway flyby is shown in Figure 8.7, together with the ultrasonic measurement for a reference of the true altitude.



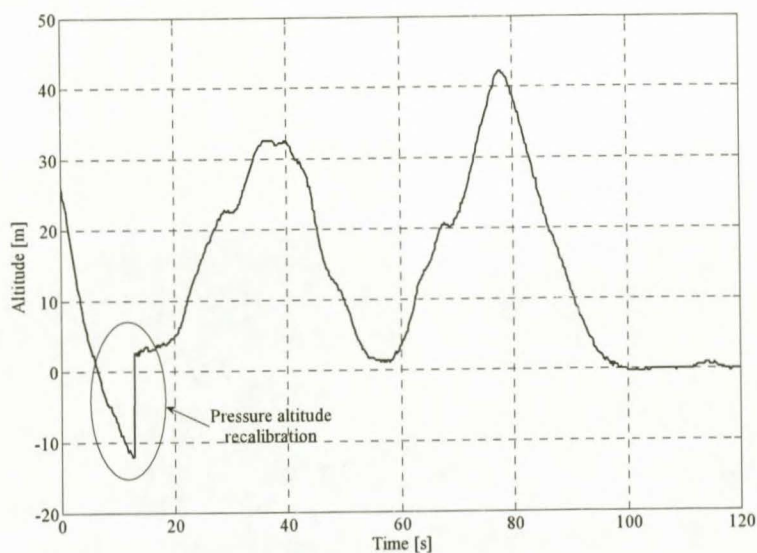
**Figure 8.7:** Altitude Measurement during a Manual Runway Flyby

The graph shows the pressure altitude is fairly accurate and never deviates more than 0.5m from the ultrasonic altitude. Note that the pressure altitude has a resolution of 0.28m (refer to section A.3). The absolute pressure sensor is therefore accurate enough to be used for altitude control during landing.



**Figure 8.8:** Aircraft during a Manual Runway Flyby

However, the absolute pressure sensor can display a drift of up to 4m over a 15 minute flight due to temperature which will render it useless for altitude control during a landing. Since the GPS altitude is only accurate to within 5m (from uBlox data sheet), the absolute pressure sensor is the only sensor available to provide an altitude measurement. It is therefore necessary to recalibrate the absolute pressure sensor. The accurate measurement from the ultrasonic is well suited for the task. Recalibration of the absolute pressure sensor with the ultrasonic is shown in Figure 8.9. The absolute pressure altitude measurement was deliberately manipulated to be erroneous before recalibration. Note the zero altitude measurement after landing.

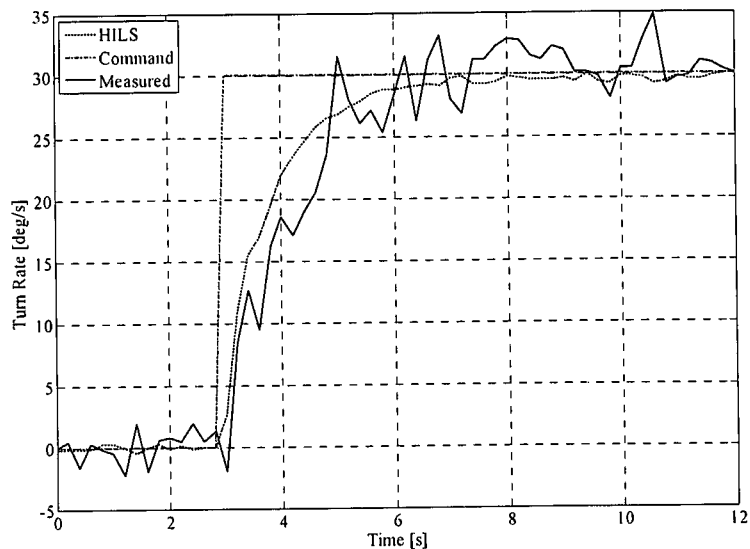


**Figure 8.9:** Pressure Recalibration with the Ultrasonic

## 8.3 Take-Off

### 8.3.1 Runway Yaw Rate Controller

Figure 8.10 shows the response of the runway yaw rate controller to a 30deg/s step command in yaw rate, together with the theoretical response from the HIL simulator. The actual and theoretical yaw rate responses correspond extremely well. The exceptional correlation is a great vote of confidence in the accuracy of the runway model and proves the linearisation assumptions made for controller design purposes are valid.



**Figure 8.10:** Runway Yaw Rate Controller Step Response

Unfortunately the yaw rate measurement is contaminated by a lot of noise. This is due to the small wheel size and the absence of suspension in the undercarriage of the aircraft used in this project. Any disturbance from the runway will only be damped by the vibration isolation of the avionics package. The noise on the yaw rate measurement can be reduced by improving the vibration isolation of the avionics package.

### 8.3.2 Radius Interface

After the excellent response from the runway yaw rate controller, the radius interface is also expected to have good results with the results mainly dependant on the GPS velocity accuracy. Figure 8.11 shows the response generated by the radius interface to a 4m turn radius command.



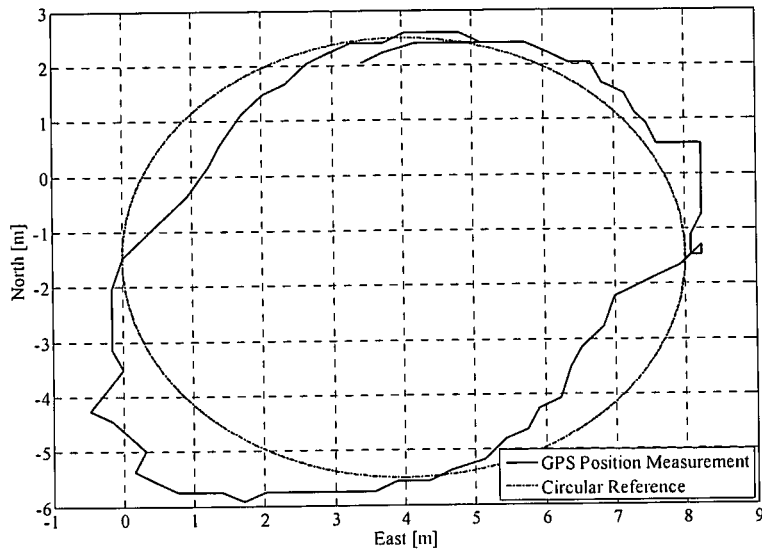


Figure 8.11: Radius Interface Response to a 4m Command

It is difficult to quantify the actual response since the position measurement is obtained from the GPS receiver with 4m accuracy. However, the general circular motion of the aircraft is clear.

### 8.3.3 Runway Heading Controller

Figure 8.12 shows the response of the runway heading controller to an  $80^\circ$  step in heading, as well as the associated yaw rate during the heading step. The theoretical HIL response to the same step command is also shown.

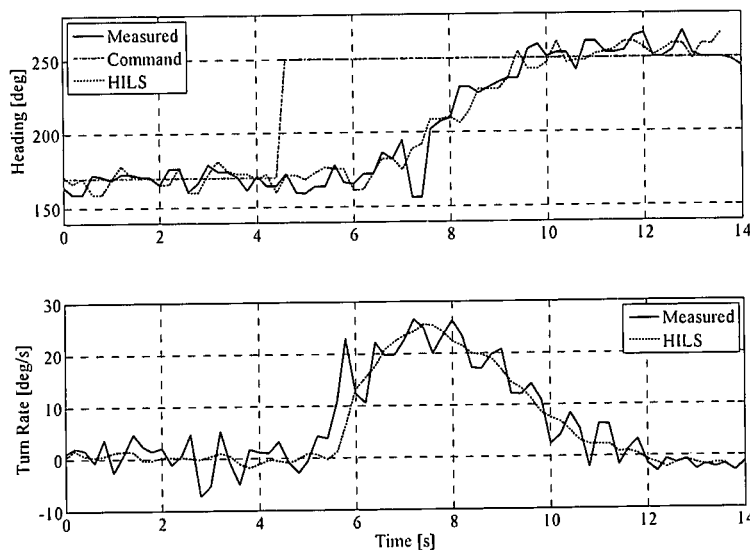


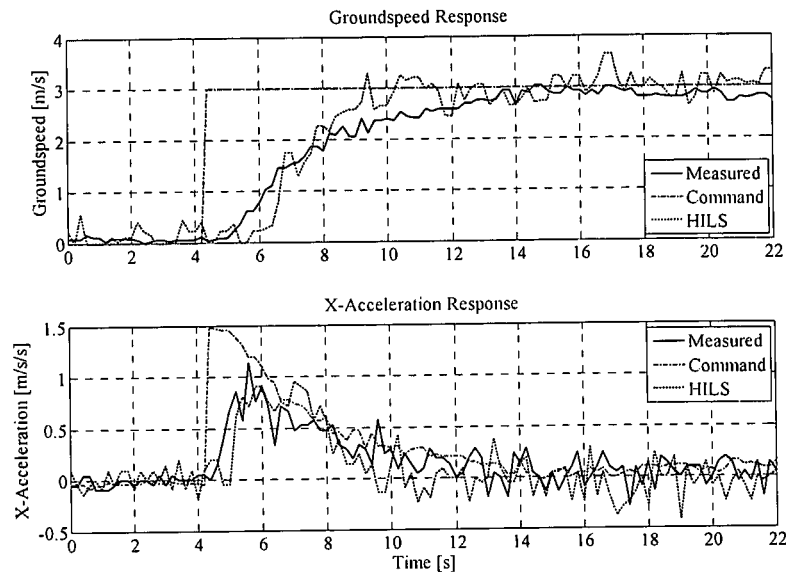
Figure 8.12: Runway Heading Controller Step Response

The actual heading step response again corresponds closely with the simulated theoretical response. This excellent response is expected since the runway yaw rate controller has an excellent response that matches its theoretical response. The heading controller will therefore interact with a very similar system to the one used for designing the controller and the actual and theoretical responses will be close to identical. Note the high

noise levels on the GPS heading measurement at the low groundspeed. In spite of the noise, the controller still performs excellently.

### 8.3.4 Groundspeed and Runway Guidance Controllers (Autonomous Taxing)

Figure 8.13 shows the longitudinal response during an autonomous taxing (groundspeed controller response) to a 3m/s step command in groundspeed. Figure 8.13 also shows the associated acceleration command and response, as well as the theoretical groundspeed and acceleration responses from a HIL simulation.



**Figure 8.13:** Longitudinal Response during Autonomous Taxing

The measured and theoretical responses correspond very well. However, the actual response has a rise time of approximately 6s, where the theoretical response has a rise time of 3s with an initial dead band of 2s. The difference can be attributed to several sources of which the most influential are errors in the engine model used for design and the non-linear engine model in the HIL simulator. The effect of the friction forces (that were ignored for design purposes) also differs out in the field and can cause the response to degrade. The small steady state groundspeed error is due to the absence of an integral term in the controller and the misalignment of the accelerometer. Note the sharp increase in engine induced noise on the accelerometer measurement as the throttle is opened. The overall performance of the groundspeed controller is very satisfactory.

Figure 8.14 shows the lateral response during an autonomous taxing with the runway (8m wide) and direction of travel indicated.

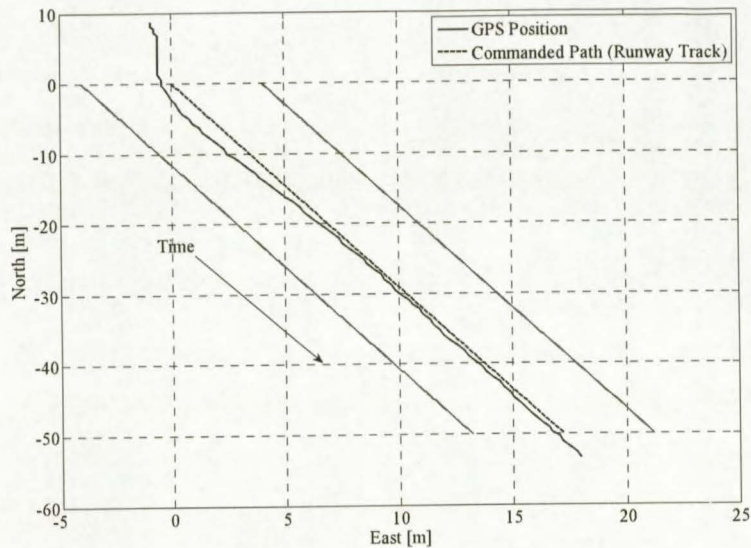


Figure 8.14: *Lateral Response during Autonomous Taxing*

The guidance controller response is again exceptionally good and the runway centre line is tracked with great precision. This response is expected since the runway yaw rate and heading controllers also have excellent responses that match their theoretical response. The outer guidance loop will therefore interact with a very similar system as the one used for design purposes. However, the good response was obtained right after the GPS position measurement has been zeroed. Any drift in the GPS position measurement will cause an offset from the runway centre line. The aircraft performing the autonomous taxing is shown in Figure 8.15.



Figure 8.15: *Aircraft Autonomously Taxing*



### 8.3.5 Autonomous Take-Off

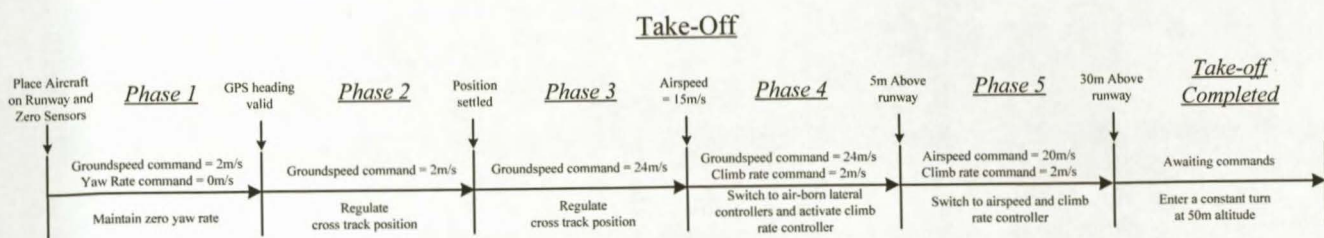


Figure 8.16: Take-Off Sequence Phases (Repeat)

The take-off sequence time line is repeated in Figure 8.16 for convenience. Figure 8.17 shows the aircraft during an autonomous take-off.



Figure 8.17: Aircraft during an Autonomous Take-Off

Figure 8.18 and Figure 8.19 shows the longitudinal reference commands and system responses during the take-off sequence.

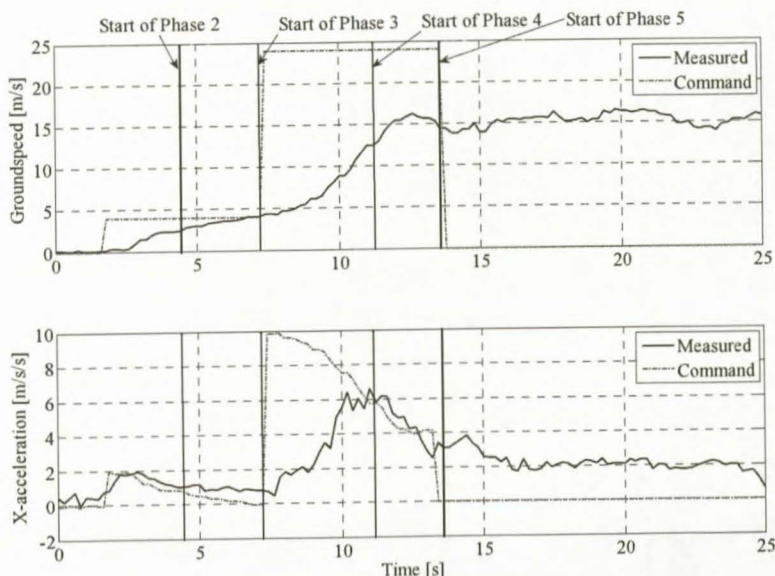
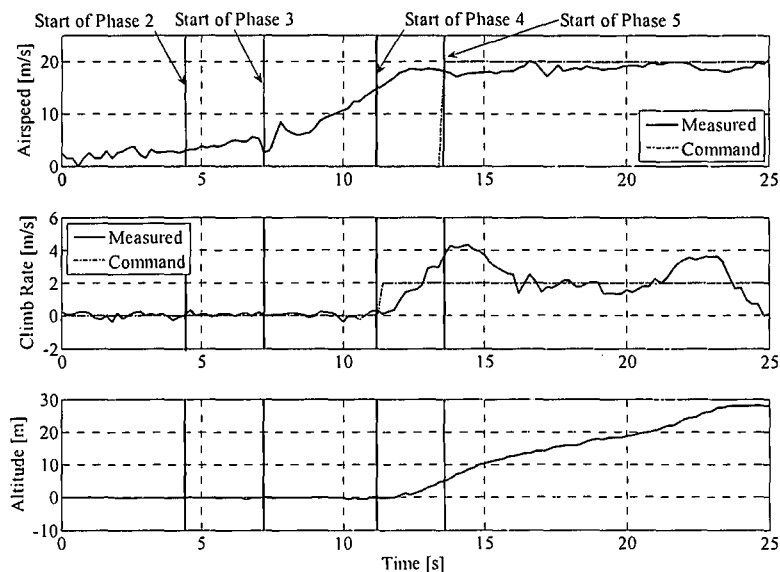


Figure 8.18: Groundspeed and Acceleration Control during an Autonomous Take-Off

Figure 8.18 shows the groundspeed and acceleration control during an autonomous take-off. The take-off sequence starts when the groundspeed command of 4m/s is issued. The groundspeed command was increased from 2m/s to obtain better heading measurements from the GPS. At 4.5s a valid GPS heading is signalled and the system entered phase 2 in which the cross track error is regulated. When the position settled, phase 3 is entered and the groundspeed command is stepped up to 24m/s, starting the ground roll. The aircraft lifts off at the start of phase 4 (13.6s). The  $X_B$ -acceleration is the main variable being controlled. The pitch angle obtained during climb out in phase 4 causes the groundspeed to remain at 15m/s instead of increasing to the command 24m/s.



**Figure 8.19:** Longitudinal Control during an Autonomous Take-Off

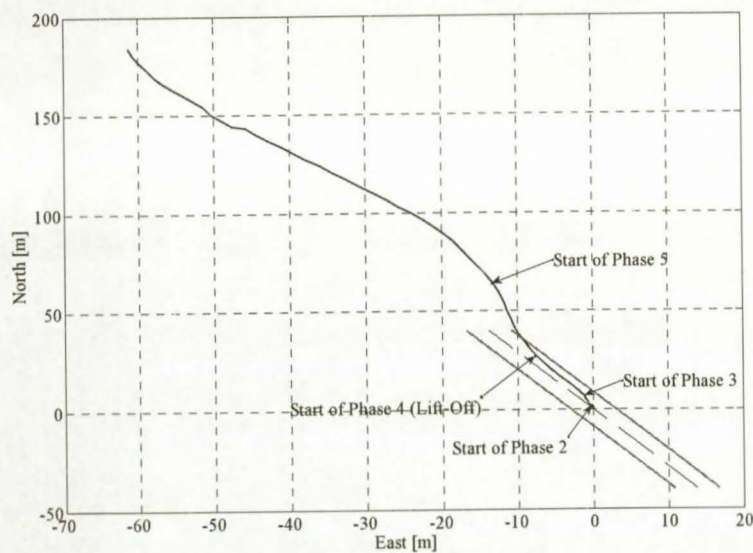
However, the airspeed still increases after take-off as is clear from Figure 8.19. When an altitude of 5m is reached the system enters phase 5 and the airspeed and climb rate controller is armed, continuing the climb out. Figure 8.19 shows a sharp increase in the airspeed measurement just as phase 3 is entered. This increase is caused by airflow from the propeller coupling into the pitot-static tube system as the throttle is opened for the ground roll.

Figure 8.19 also shows excessive overshoot in the climb rate during climb out in phases 4 and 5. The excessive overshoot is caused by the climb rate controller and is most likely due to modelling errors of the elevator, with the elevator being more effective in practise than in simulation.



**Figure 8.20:** Aircraft Autonomously Rotating

Figure 8.21 shows the two dimensional GPS position of the aircraft during take-off with the runway limits for a 6m wide runway superimposed.



**Figure 8.21:** Lateral Control during an Autonomous Take-Off

The system displays good lateral control during the take-off sequence. However, Figure 8.21 shows that the aircraft slightly yaws to the right during take-off. This is due to the nose wheel losing effectiveness as the lift increases, and the aircraft weather cocked into the crosswind present during the flight test. The effect was also seen during HIL simulation.





Figure 8.22: Aircraft during an Autonomous Climb Out

## 8.4 Landing

### 8.4.1 Conventional Flight Controllers

Figure 8.23 shows the responses of the airspeed, altitude and turn rate controllers, at the original trim airspeed of 20m/s, as designed for in [1].

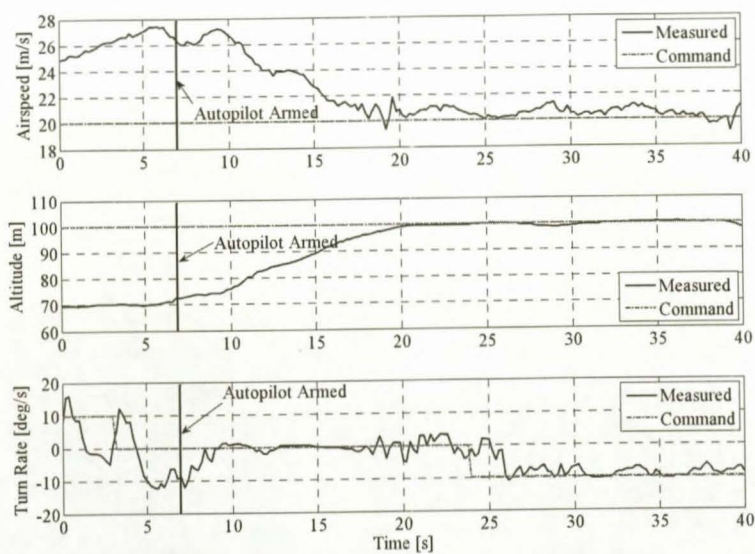


Figure 8.23: Conventional Flight Controllers Responses at 20m/s

Figure 8.24 shows the responses of the airspeed, altitude and turn rate controllers at airspeeds of 24 and 16m/s. It is clear that the responses in Figure 8.23 and Figure 8.24 are almost identical. The single trim condition constraint is therefore successfully removed and the flight envelope extended.

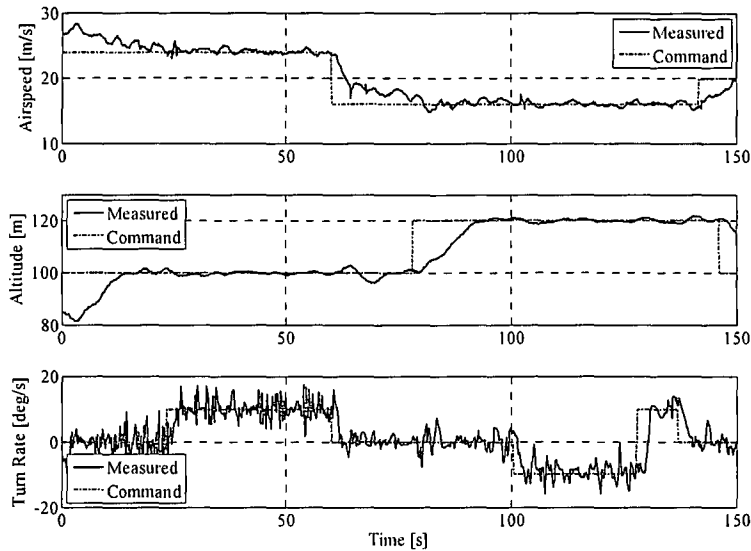


Figure 8.24: Conventional Flight Controllers Gain Scheduled Responses

### 8.4.2 Autonomous Landing

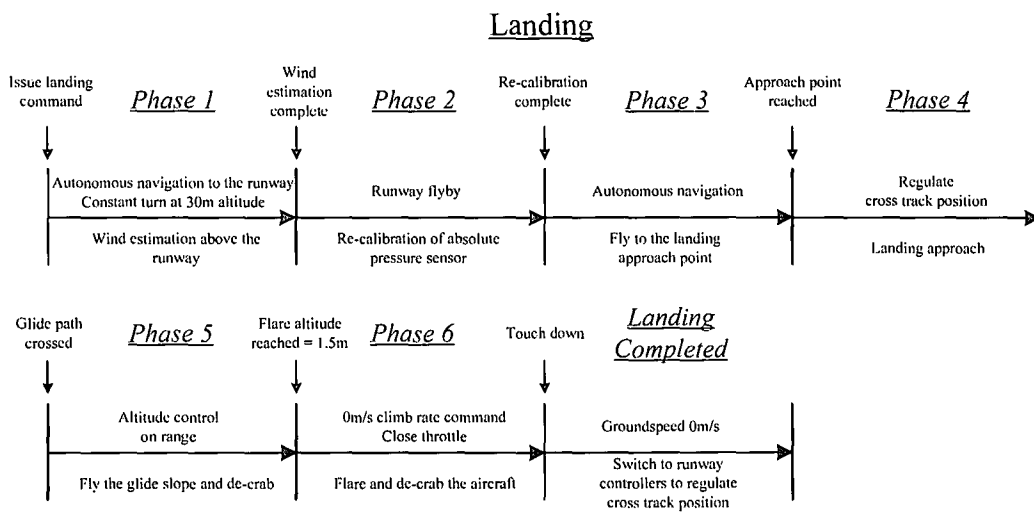


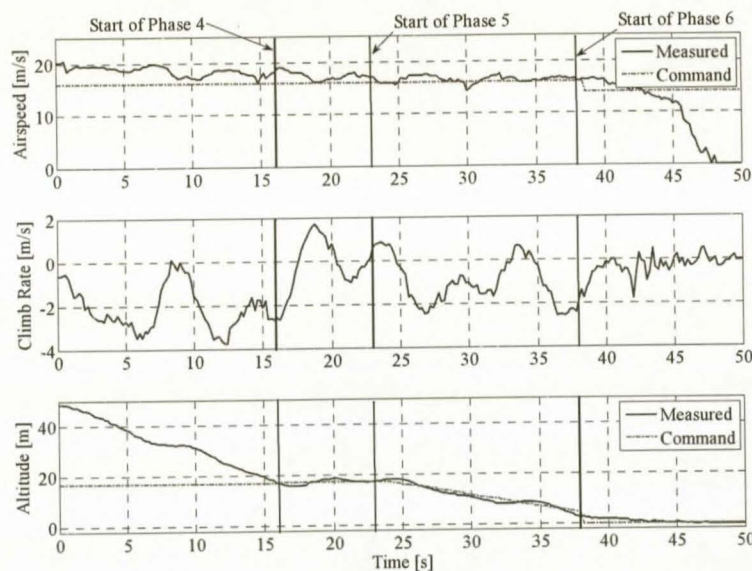
Figure 8.25: Landing Sequence Phases (Repeat)

The landing sequence time line is repeated in Figure 8.25 for convenience. Figure 8.27, Figure 8.28 and Figure 8.30 shows the reference commands and system responses during the landing sequence. Figure 8.26 shows the aircraft on an autonomous landing approach. The first two phases of the landing sequence presented in Figure 8.25 was not executed during the flight test. The landing path parameters used are summarised in section B.6.



**Figure 8.26:** Aircraft on an Autonomous Landing Approach

Figure 8.27 shows the longitudinal control of the aircraft during the landing sequence, with the start of each phase indicated.



**Figure 8.27:** Longitudinal Control during an Autonomous Landing

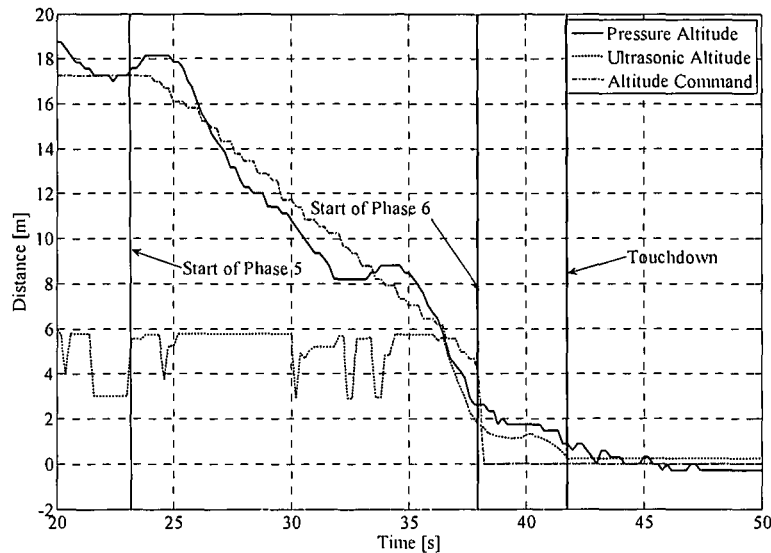
The presented flight data starts in phase 3 during the final moments of navigation. The aircraft is descending to the approach altitude of 17m, and the airspeed is regulated down to 16m/s. At 16s the approach point is reached and the system switches to phase 4 where the cross track position off the runway is regulated. Just after 23s the glide path is crossed and the system enters into phase 5 where the glide path is flown. The system enters phase 6 when the aircraft reaches an altitude of 1.5m just after 38s.

Figure 8.27 show that the airspeed response has a slight positive oscillation from the command reference. The oscillation is caused by airflow from the propeller coupling into the pitot-static tube system, corrupting the airspeed measurement. From equation (B.2) and the associated gain values in Table B.2, it is seen that a large feedback gain exist for airspeed error perturbation to throttle actuation. The throttle is therefore the main actuator controlling airspeed.



As the throttle is opened the airspeed measurement will immediately increase due to the coupling into the pitot-static tube system. This in turn will cause the throttle to be closed. The sequence of events will repeat resulting in the oscillation seen in the measured airspeed response.

Figure 8.28 shows an enhanced view of the altitude response during phase 5 and 6. The ultrasonic altitude is superimposed and is only valid below 5m.

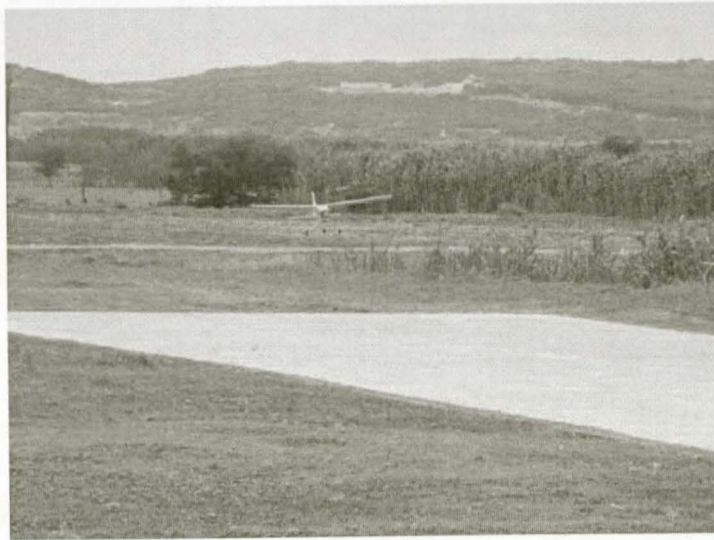


**Figure 8.28:** *Glide Path Tracking and Flare Response during an Autonomous Landing*

Throughout the landing sequence the altitude reference is tracked with acceptable precision, never deviating more than 2m from the reference command. However, the altitude response also displays periodic motion. From equation (B.2) and (B.3), as well as the associated gain values in Table B.2 it is seen that large feedback gains exist for climb rate error perturbation to elevator actuation, and integrated climb rate error to throttle actuation. As the altitude error around 25s grows, the integrated climb rate error will build up to a negative value causing the throttle to be closed. The climb rate error will be fed back to the elevator causing the aircraft to descend. The fast descend will result in the integrated climb rate error reversing and building up to a positive value, opening the throttle. The increased thrust will cause the aircraft to climb as can be seen from 32 to 35s. The climb rate error and integrated climb rate errors will then again reverse, and the sequence of events will repeat.

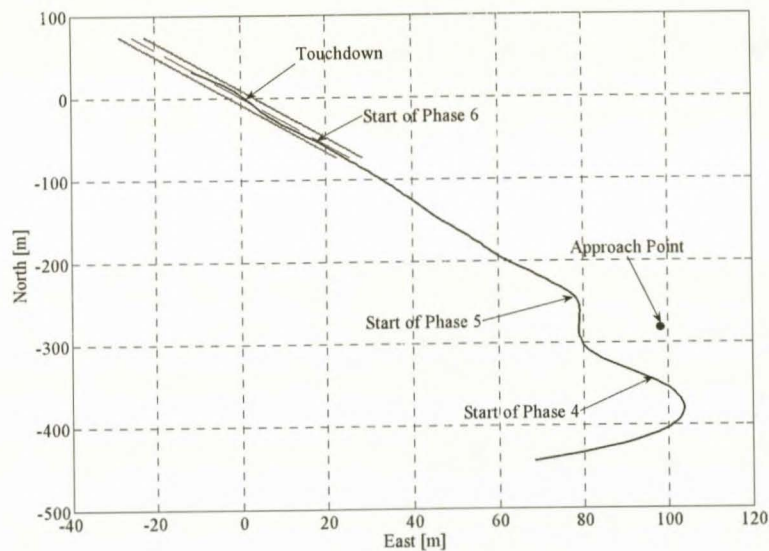
The altitude response can be improved by reducing the feedback gains of climb rate error and integrated climb rate error to the elevator and throttle respectively.

The flare (phase 6) was initiated at 38.4s. The climb rate controller response was faster than expected, causing the aircraft to flare sharply and hold an altitude of 1m. The sharp decrease in sink rate can be seen in Figure 8.27. Since the throttle was closed the aircraft glided the final meter and touchdown occurred at 42s. The sharp flare was most likely caused by modelling errors of the elevator, with the elevator being more effective in practise than in simulation.



**Figure 8.29:** Aircraft just before Autonomous Touchdown

Figure 8.29 shows the aircraft just before an autonomous touchdown. Figure 8.30 shows the lateral position of the aircraft during the landing sequence with the start of every phase indicated.



**Figure 8.30:** Lateral Control during an Autonomous Landing

From Figure 8.30 it is seen that the system switches to phase 4 as the approach point is reached and starts to regulate the cross track position off the runway. Just as the cross track error settled, phase 5 is entered and the glide path is flown. The aircraft entered phase 6 (flared) prematurely at approximately 20m from the touchdown point. The premature flare was caused by the error in altitude as shown in Figure 8.28. The sharp flare induced the long gliding section between phase 6 and touchdown. Note how the runway guidance controller pulls the aircraft back on to the runway track after touchdown.

The cross track error is seen to settle faster than expected. The approach point can therefore be moved much closer to the runway. Figure 8.31 shows the aircraft after a successful autonomous landing.



**Figure 8.31:** *Aircraft after a Successful Autonomous Landing*

## 8.5 Summary

This chapter presented the data from five flight test in which the sensors were evaluated, the take-off and landing controllers from Chapter 4 and Chapter 5 were tested, and the take-off and landing sequences were shown to work in practise. The results were evaluated and discussed with possible explanations given for any deviations of the practical results from the theoretical results. The overall performance of the system is very satisfactory. Recommendations for possible improvements are discussed in the next chapter.



# Chapter 9

## *Summary and Recommendations*

### 9.1 Summary

This thesis presented the design and successful practical demonstration of controllers required for autonomous take-off and landing of a fixed wing unmanned aerial vehicle.

The project served as an extension to the autopilot develop in [1] for autonomous navigation. A non-linear model for the aircraft dynamics on the runway, a fixed pitch propeller model, a piston engine model and a wind gust model were developed. These models complement the non-linear aircraft model to form a complete model of the dynamics of the aircraft on the ground and in the air. The non-linear models were linearised and used to design the feedback controllers required to perform the take-off and landing sequences. The conventional flight controllers were also extended and improved to remove the single trim condition constraint. The avionics used for autonomous navigation was redesigned and extended to form a small, robust, reliable and professional hardware platform for use in future UAV projects. The avionics was designed to include hardware in the loop (HIL) simulation capabilities and, together with the non-linear models developed and the HIL capabilities from [22], a full non-linear hardware in the loop simulator was formed. Once the controllers were implemented on the avionics, they were subjected to rigorous non-linear simulation with excellent results.

Throughout the development of the project the system was subjected to and verified through practical testing. The project saw the successful autonomous take-off and landing of the Reliance 0.46 size model aircraft.

### 9.2 Recommendations

This thesis was the first in the ESL UAV group to venture into the unknown world of autonomous take-off and landing. The following is a list of recommendations on how the current systems can be improved and extended.

#### **General**

- In the current system only the GPS position information is used to calculate the cross track error off the runway during take-off and landing. Combining GPS position, rate gyroscope measurements and gyroscope biases in an optimal filter can provide a better estimate of the cross track error to improve the lateral control of the system.

- The yaw orientation controller uses only the yaw rate gyroscope measurement to control turning motion. If an estimate of the roll angle can be obtained, the Euler angle differential equations (kinematic equations) as given by equation (F.7) can be used to calculate  $\dot{\Psi}$  from the yaw and pitch rate measurements. Using this measurement in the yaw orientation controller will greatly improve turning motion and also allow for steeper bank angles. An estimate of the roll angle can be obtained from a complementary filter combining the roll and yaw rate measurements as explained in lecture 15 of [24].
- The introduction of an extended Kalman filter (EKF) into the system can be considered. The optimal filter can provide the roll angle and cross track error estimates required for the previous two recommendations. Pitch angle information will also be valuable while flaring and a more accurate altitude estimate will greatly aid the glide path tracking accuracy. The availability of the full state vector of the aircraft also opens the door for faster LQR controllers with good dynamic responses. However, it is important to remember that the EKF is not essential and that the increased complexity will greatly increase the risk of system failure.

### Hardware

- The hardware (electronics) used in this project, although simple and cost-effective, is not suited for precision tasks such as landing. It is strongly recommended that an avionics system based on a single microprocessor with floating point capabilities is designed for future projects. The floating point capabilities will open many doors to different control strategies and other algorithms.
- It is recommended that the IMU be implemented as a stand alone unit with its own CPU. All the signal conditioning can be integrated into the unit with a CAN interface to the avionics. A small Kalman filter can be implemented as part of the IMU for gyroscope bias estimation. The separate IMU will allow for placement directly on the aircraft's centre of gravity as well as for vibration and temperature isolation.
- The standard RC equipment used in this project is very susceptible to EMC interference and is not suitable for use in conjunction with additional electronics. The EMC interference degrades the RF control link, severely reducing the obtainable flight range. It also causes glitches on the control surfaces and autopilot arm/disarm status that poses a great risk to the aircraft. It is recommended that a dedicated RF control link is developed and integrated into the avionics that operates at a much higher frequency than the 35 MHz of standard RC equipment. 2.4 GHz is recommended.
- Climb rate is a very important parameter during the take-off and even more important during the landing. An accurate and low noise climb rate sensor will greatly aid the take-off and landing control. One possible option is a laser altimeter to provide a low noise and accurate altitude measurement. A climb rate measurement can then be obtained by differentiating the altitude measurement.

### Take-Off

- The nose wheel and actuator combination used in this project are very non-linear and require a robust controller to stabilise and control the system. The high phase margin requirement reduces the obtainable bandwidth from the controller resulting in a slow response and average wind gust rejection during a cross wind take-off. It is recommended that the current nose wheel be replaced by a well behaved linear actuator.

- The dynamic response of the climb rate controller needs improvement. One possible solution is to design a Multi-Input-Single-Output (MISO) controller where not only the elevator, but also the throttle actuator, is used to control the climb rate. Longitudinal control during take-off can then simply consist of commanding 75% throttle and controlling the climb rate with the MISO climb rate controller around this throttle offset. It should be noted that the throttle offset must be included in the controller design. The MISO climb rate controller might also be well suited for the flaring manoeuvre during landing.
- In this project, rotation and climb out is initiated by a 2m/s climb rate command. It is recommended that this command be increased to 4m/s to achieve a steeper take-off that clears the runway faster. The faster climb out will ensure that the aircraft is less susceptible to wind gusts and will provide for better obstacle clearance. The steeper climb out was not implemented in this project due to the fact it will start to violate small disturbance theory assumption made in the linear aircraft model. Further investigation is required.

## Landing

- The steady state lateral cross track error of the current guidance controller can be up to 10m, translating into a required runway width of 20m for a guaranteed landing. Several possible solutions exist to improve the steady state cross track error in order to reduce the required runway width. The first solution would be to introduce an integrator term into the airborne heading controller that can provide a non-zero steady state yaw rate command to oppose the gyroscope bias and consequently push out the cross track error. This solution was attempted in this project, but a stable system with an acceptable dynamic response could not be obtained. The steady state cross track error can significantly be reduced if the bias on the yaw rate gyroscope can be eliminated or at least be controlled. It is therefore recommended to incorporate an estimate of the gyroscope bias into the current yaw orientation controller. The steady state cross track error can further be improved by making use of the differential capabilities of the GPS receiver, which will increase its accuracy. It is estimated that the required runway width can be reduced to 8m with the gyroscope bias estimate and DGPS enabled. This is far more acceptable. Providing guidance commands straight to the lower level controllers and not through the heading controller, might also increase overall lateral performance.
- Performing the flare manoeuvre at the correct height is very important. If the height is too low, the descent rate at touchdown will be excessive, risking damage to the undercarriage. Conversely, an excessive flare height results in an extended touchdown point and can also cause excessive airspeed loss during the flare since the throttle is closed. It is therefore recommended that the flare initiation height, for the flare manoeuvre as implemented in this project, be set adaptively based on the sink rate of the aircraft. Adaptive flaring can also be applied to the flare controller presented in section 5.3.
- The current system has no method of de-crabbing. This will result in excessive side loads and yaw transients at touchdown during a cross wind landing. If an accurate measure of the side slip angle can be obtained with either a side slip vane or from an optimal filter estimate, the de-crab controller presented in section 5.4 can be used to de-crab the aircraft before touchdown. The speed of response of the controller can be increased if feed forward commands are issued to the ailerons to prevent coupling into the yaw orientation control system. These commands can be issued in a similar fashion to the feed forward rudder commands from the yaw orientation control system to counter adverse yaw in [1].
- Improving the navigator will allow the aircraft to settle onto the runway track faster during a landing approach. This will allow the approach point (defined in section 2.2.4) to be placed much closer to the runway. It is recommended that the heading control loop be removed and the guidance controller

adjusted to directly provide yaw rate commands to the yaw orientation controller. This will allow for a faster guidance controller. To improve the circular path tracking, a feed forward turn rate command based on the circle radius and aircraft velocity can be provided as suggested in [1].

# Bibliography

- [1] Iain K. Peddle. *Autonomous Flight of a Model Aircraft*. Masters dissertation, University of Stellenbosch. 2005.
- [2] A.C. Kermode. *Mechanics of Flight*. London: Pitman. 1979.
- [3] McRuer, Ashkenas and Graham. *Aircraft Dynamics and Automatic Control*. Princeton University Press. 1973.
- [4] M.V. Cook. *Flight Dynamics Principles*. Elsevier Butterworth-Heinemann. 1997.
- [5] J.R. Ellis. *Vehicle Dynamics*. London Business Books Limited. 1969.
- [6] Thomas D. Gillespie. *Fundamentals of Vehicle Dynamics*. Society of Automotive Engineers Inc. 1992.
- [7] Marco Monster. *Car Physics for Games*.  
<http://home.planet.nl/~monstrous/tutcar.html>. 2003.
- [8] Gene F. Franklin, J. David Powell, Abbas Emami-Naeini. *Feedback Control of Dynamic Systems*, 3rd ed. Addison-Wesley Publishing Company. 1994.
- [9] Gene F. Franklin, J. David Powell, Michael Workman. *Digital Control of Dynamic Systems*. Addison Wesley Longman. 1998.
- [10] John H. Blakelock. *Automatic Control of Aircraft and Missiles*, 2nd ed. Wiley-Interscience. 1991.
- [11] Aeronautics Learning Laboratory for Science, Technology, and Research. *Flight Performance – Level 3, Take-Off and Land*.  
<http://www.allstar.fiu.edu/aero/TO&Land.htm>. 2004.
- [12] Unmanned Dynamics. *AeroSim, Aeronautical simulation block set version 1.1: User's Guide*.  
<http://www.u-dynamics.com>. 2004.
- [13] JSBSim. *Aeromatic version 0.7*.  
<http://jsbsim.sourceforge.net>. 2005
- [14] R.C. Hibbeler. *Engineering Mechanics, Dynamics*. Prentice Hall. 1997.



- [15] NASA History Office web page.  
<http://history.nasa.gov/SP-367/appendc.htm>. 2004.
- [16] Fly Light Ground School – Theory of Flight. *Take-off Considerations*.  
<http://www.auf.asn.au/groundschool/umodule11.html#tosequence>. 2006.
- [17] 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.
- [18] Fanus Groenewald. *Development of a Rotary-Wing Test Bed for Autonomous Flight*. Masters dissertation, University of Stellenbosch. 2004.
- [19] Arthur E. Bryson, Jr. *Control of Spacecraft and Aircraft*. Princeton University Press. 1994.
- [20] Johan Bijker. *Development of an Attitude Heading Reference System for an Air Ship*. Masters dissertation, University of Stellenbosch. 2006.
- [21] Atmel. *AVR910: In-System Programming*. Atmel application note. 2000.
- [22] Willem Hough. *Autonomous Acrobatic Flight of an Unmanned Aerial Vehicle*. Masters dissertation, University of Stellenbosch. 2006.
- [23] Mark C. Silberbauer. *Simulation Visualisation System*. Final year under graduate project, University of Stellenbosch. 2005.
- [24] MIT. *Open Course Ware – Aircraft Stability and Control*, Lecture 1 to 17. 2005.
- [25] I.K. Peddle and G.W. Milne. *Development of a Waypoint Navigation Autopilot for an Unmanned Aerial Vehicle*. SAIMEchE R&D Journal (in review). 2006.
- [26] Cloud Cap Technology. *A Highly Integrated UAV Avionics System*. 2003.
- [27] P. Riseborough. *Automatic Take-Off and Landing Control for Small UAV's*. BAE Systems Australia, Melbourne. 2004.
- [28] Hal Stoen. *Cross Wind Landings*. <http://stoenworks.com>. 2002.
- [29] John S. Denker. *See How It Flies*, Chapter 13 – Take-Off. <http://www.av8n.com/>. 2003.
- [30] Garth W. Milne. *Simplifying Vector Rotations and Coordination*. AIAA 2001-4310. 2001.
- [31] M.C. Koen. *Modelling and Simulation of an RPV for Flight Control System Design Purposes*. Department of Electronic and Computer Engineering, University of Pretoria.
- [32] Micropilot web page. <http://micropilot.com/>. 2005.

[33] Unmanned Dynamics web page. <http://u-dynamics.com/>. 2005.

[34] Kentron web page. <http://www.kentron.co.za/>. 2005.

[35] Advanced Technologies and Engineering (ATE) web page.  
<http://www.sadid.co.za/edition7/ate/ate.html>. 2005.

[36] Stephen Corda, Mark T. Stephenson, Frank W. Burcham and Robert E. Curry. *Dynamic Ground Effects Flight Test of an F-15 Aircraft*. Nasa Technical Memorandum 4604. 1994.

# Appendix A

## Avionics Details

### A.1 Microcontrollers

The heart of the avionics package consists of the two microcontrollers, *Flight Control* and *Mission Control*. Figure A.1 shows a detailed interconnection block diagram for the two microcontrollers.

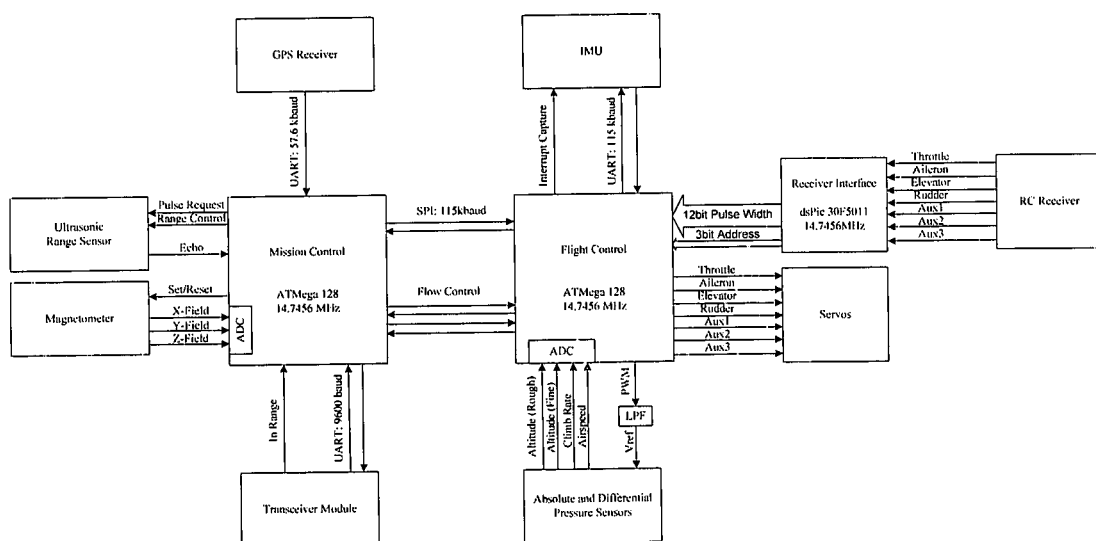


Figure A.1: Main Microcontroller Block Diagram

“Flight Control is responsible for executing all of the control code, interfacing with the high bandwidth sensors such as those on the IMU and air data subsystem 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 57.6 kHz Pulse Width Modulation (PWM) signal generated by Flight Control, to extract only the DC component with acceptable ripple” [1].

“To allow Flight Control to switch between safety pilot and autopilot, the servos are driven directly from the microprocessor. Manually driving the servos allows single control loops to be turned on at a time, while the safety pilot controls the actuators not in use by the autopilot” [1].

“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. The GPS and RF modules interface serially with Mission Control, via the UART protocol, at 57.6kB and 9600 baud respectively” [1].

## A.2 Inertial Measurement Unit

A block diagram representation of the IMU showing the sensors, signal conditioning and microprocessor are shown in Figure A.2.

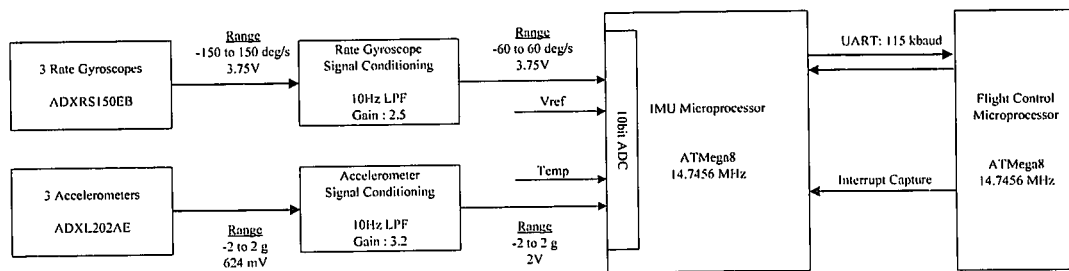


Figure A.2: Inertial Measurement Unit Block Diagram [1]

With no signal conditioning, the rate gyroscopes are capable of sensing angular rates of up to  $\pm 150$  deg/s at a bandwidth of 40Hz, mapped over a 2.5V centred voltage span of 3.75V. The signal conditioning block reduces the bandwidth of the signal from the rate gyroscopes to 10Hz and the measurable range to  $\pm 60$  deg/s, removing unwanted noise and increasing the obtainable resolution from a 10bit ADC to 0.1deg/s.

The raw output of the accelerometers is a  $\pm 2g$  signal at a bandwidth of 6 kHz, mapped over a voltage range of 0.624V centred about 2.5V. The signal conditioning block reduces the bandwidth of the signal from the accelerometers to 10Hz, removing unwanted noise and increasing the obtainable resolution from a 10bit ADC to 0.01g.

“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 is an Atmel ATMega8 that runs at 14MHz and is responsible for providing further filtering of the signals, removing biases and interfacing with the Flight Control microprocessor” [1].

## A.3 Air Data

The air data subsystem is responsible for generating airspeed, climb rate and altitude signals from a pitot-static system with differential and absolute pressure sensors. Figure A.3 shows a block diagram for the air data subsystem.

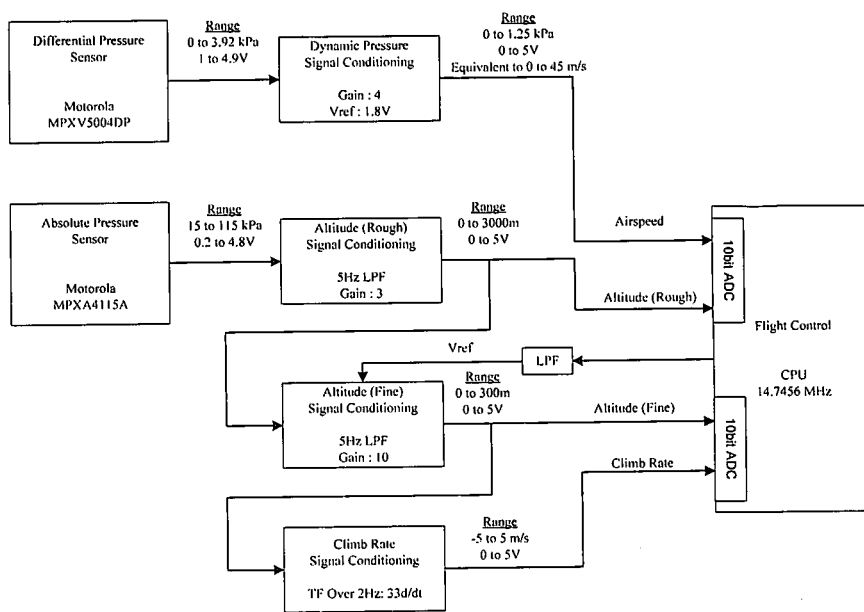


Figure A.3: Air Data Block Diagram [1]

$\rho$  is the air density and  $\bar{V}$  is the airspeed. “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. It is desired to measure altitudes ranging from 0 to 3000m translating into a desired pressure range of 70 to 105kPa. The *Rough Altitude Signal Conditioning Block* limits the signal bandwidth to 5Hz and maps the entire pressure range over the ADC input resulting in a digital resolution of 2.8m”.

“The *Fine Altitude Signal Conditioning Block* provides a high resolution altitude measurement over a local 300m range for any altitude between 0 and 3000m with a digital resolution of 0.28m at sea level. The amplifier offset voltage needed to zoom in on the local altitude range, is obtained from the low pass filtered PWM signal from Flight Control”.

## A.4 Telemetry Packets

The telemetry structure developed in [1] is used directly in this project. Telemetry is transmitted serially at 9600 BAUD at 5Hz, translating into a maximum of 192 bytes that can be transmitted per cycle. Additional data were added to the primary, secondary and GPS packets and are summarised below. The total telemetry packet size is expressed as a percentage of the maximum 192 bytes that can be transmitted when all of the telemetry packets are sent during the same transmitting cycle.

### Primary 5Hz packet

- 47 bytes: Primary telemetry from [1]
- 4 bytes: Ultrasonic altitude and climb rate
- Total = 51 bytes  $\Rightarrow$  27% of 5Hz transmit cycle



### Secondary 1Hz packet

- 18 bytes: Secondary telemetry from [1]
- 1 byte: System Status
- Total = 19 bytes  $\Rightarrow$  10% of 5Hz transmit cycle

### GPS 4Hz packet

- 15 bytes: GPS telemetry from [1]
- 2 bytes: GPS Range
- Total = 17 bytes  $\Rightarrow$  9% of 5Hz transmit cycle

## A.5 System Control and Status Bytes

### Autopilot Bytes

Table A.1: Autopilot Control Byte 0

DATA AP_CMD[0]		
Binary Value	Hexadecimal Value	Function
00000001	0x01	Autopilot Armed
00000010	0x02	User/Navigator Select
00000100	0x04	Airspeed Control
00001000	0x08	Climb Rate Control
00010000	0x10	Turn Rate Control
00100000	0x20	Climb Rate/Altitude Select
01000000	0x40	Turn Rate/Heading Select
10000000	0x80	Groundspeed Control

Table A.2: Autopilot Control Byte 1

DATA AP_CMD[1]		
Binary Value	Hexadecimal Value	Function
00000001	0x01	Turn Rate/Radius Select
00000010	0x02	Reserved for Future Use
00000100	0x04	Reserved for Future Use
00001000	0x08	Reserved for Future Use
00010000	0x10	Reserved for Future Use
00100000	0x20	Reserved for Future Use
01000000	0x40	Reserved for Future Use
10000000	0x80	Reserved for Future Use

**Table A.3: Autopilot Control Byte 2**

<b>DATA_AP_CMD[2]</b>		
<i>Binary Value</i>	<i>Hexadecimal Value</i>	<i>Function</i>
00000001	0x01	Airspeed Climb Rate Proportional
00000010	0x02	Airspeed Climb Rate Integral
00000100	0x04	Pitch Rate Damper
00001000	0x08	Climb Rate Filter
00010000	0x10	Yaw Orientation Integral
00100000	0x20	Dutch Roll Damper
01000000	0x40	Counter Adverse Yaw
10000000	0x80	De-Crab Integral

**Table A.4: Autopilot Control Byte 3**

<b>DATA_AP_CMD[3]</b>		
<i>Binary Value</i>	<i>Hexadecimal Value</i>	<i>Function</i>
00000001	0x01	Runway Yaw Rate Proportional
00000010	0x02	Runway Yaw Rate Integral
00000100	0x04	Reserved for Future Use
00001000	0x08	Reserved for Future Use
00010000	0x10	Groundspeed Pitch Rate Damper
00100000	0x20	Climb rate
01000000	0x40	Reserved for Future Use
10000000	0x80	Reserved for Future Use

**Mission Control and Test Sequence Bytes****Table A.5: Sequence Control Byte 0**

<b>DATA_MC_GS[0]</b>		
<i>Binary Value</i>	<i>Hexadecimal Value</i>	<i>Function</i>
00000001	0x01	Mission Flight Selected
00000010	0x02	Take-Off Sequence
00000100	0x04	Navigation
00001000	0x08	Landing Sequence
00010000	0x10	Reserved for Future Use
00100000	0x20	Reserved for Future Use
01000000	0x40	Reserved for Future Use
10000000	0x80	Reserved for Future Use

**Table A.6: Sequence Control Byte 1**

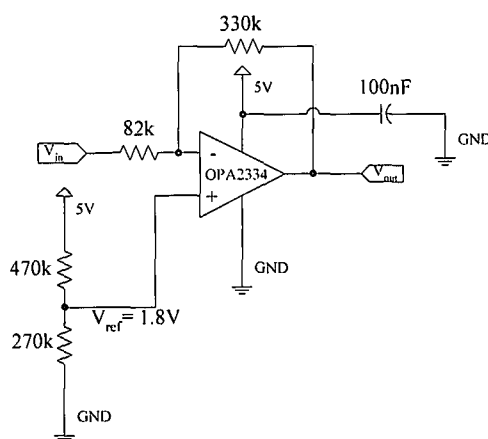
DATA_MC_GS[1]		
Binary Value	Hexadecimal Value	Function
10000001	0x81	Test Sequence 1
10000010	0x82	Test Sequence 2
10000100	0x84	Test Sequence 3
10001000	0x88	Test Sequence 4
10010000	0x90	Test Sequence 5
10100000	0xA0	Test Sequence 6
11000000	0xC0	Test Sequence 7

## A.6 Circuit Diagrams

### Differential Pressure Signal Conditioning

Figure A.4 shows the circuit diagram for the differential pressure signal conditioning. The output equation for the amplifier is given by:

$$V_{out} = 1.8 - \frac{330}{82}(V_{in} - 1.8) \quad (\text{A.1})$$



**Figure A.4: Circuit Diagram for Differential Pressure Signal Conditioning**

# Appendix B

## Additional Controller Information

### B.1 Conventional Flight Controllers

#### B.1.1 Gain Scheduling

#### Turn Rate, Altitude and Guidance Controllers

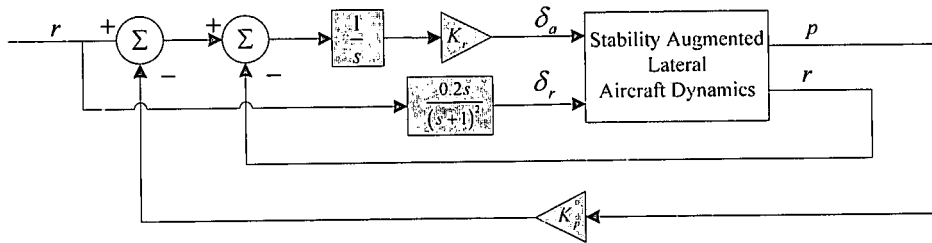


Figure B.1: Yaw Orientation Controller Block Diagram

Figure B.1 shows the block diagram for the yaw orientation controller. The closed loop dynamics of the system in Figure B.1 with the integrator state augmented and the two control loops in place are described by:

$$\begin{bmatrix} \dot{x} \\ \dot{x}_i \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B}[K_r \ 0]^T \\ -([0 \ 1]\mathbf{C} + K_p[1 \ 0]\mathbf{C}) & 0 \end{bmatrix} \begin{bmatrix} x \\ x_i \end{bmatrix} + \begin{bmatrix} 0 & \mathbf{B}[0 \ 1]^T \\ 1 & 0 \end{bmatrix} \begin{bmatrix} r \\ \delta_r \end{bmatrix} \quad (\text{B.1})$$

where  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$  and  $\underline{x}$  are the state space representation matrixes and state vector for the system given by equation (5.2). The second order high pass filter dynamic are not included in equation (B.1).

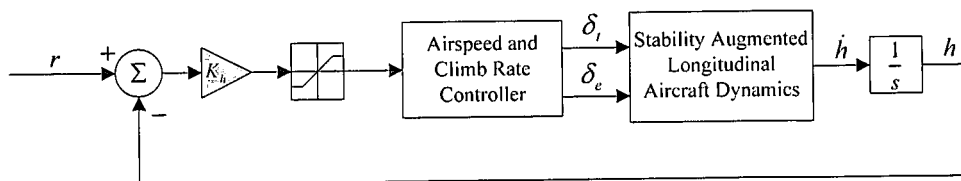


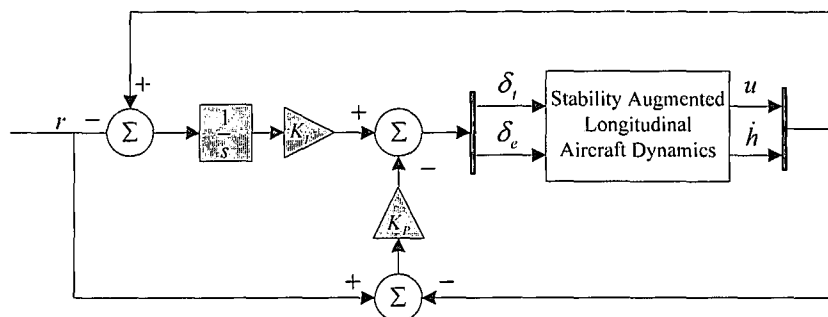
Figure B.2: Altitude Controller Block Diagram

Table B.1 presents the trim airspeed with the associated feedback gain for the altitude, turn rate and guidance controllers. Figure B.1, Figure B.2 and Figure B.4 shows the block diagrams of each of the controller and [1] provides the details on the design of each controller.

**Table B.1: Altitude, Turn Rate and Guidance Controller Gains**

	ALTITUDE CONTROLLER	TURN RATE CONTROLLER	GUIDANCE CONTROLLER
Airspeed [m/s]	Altitude Error Feedback Gain ( $K_h$ )	Roll Rate Feedback Gain ( $K_p$ )	Cross Track Error Feedback Gain ( $K_y$ )
10	0.34	1.4	0.85
12	0.44	1.1	0.7
14	0.48	0.9	0.6
16	0.48	0.77	0.52
18	0.48	0.67	0.47
20	0.5	0.6	0.42
22	0.52	0.54	0.38
24	0.52	0.49	0.35
26	0.52	0.45	0.32
28	0.54	0.42	0.3
30	0.54	0.39	0.28
32	0.54	0.36	0.26

**Airspeed and Climb Rate Controller**



**Figure B.3: Airspeed and Climb Rate Controller Block Diagram**

Figure B.3 shows the block diagram for the airspeed and climb rate controller. The feedback gain matrices are defined in equations (B.2) and (B.3). Table B.2 provides the gain scheduled feedback gain matrices for the airspeed and climb rate controller.

$$K_p = \begin{bmatrix} K_{(\delta_c \leftarrow u)} & K_{(\delta_c \leftarrow \dot{h})} \\ K_{(\delta \leftarrow u)} & K_{(\delta \leftarrow \dot{h})} \end{bmatrix} \tag{B.2}$$

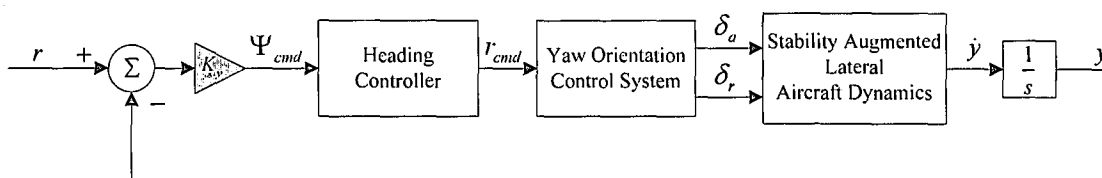
$$K_I = \begin{bmatrix} K_{(\delta_c \leftarrow e_u)} & K_{(\delta_c \leftarrow e_{\dot{h}})} \\ K_{(\delta \leftarrow e_u)} & K_{(\delta \leftarrow e_{\dot{h}})} \end{bmatrix} \tag{B.3}$$



**Table B.2: Gain Scheduled Airspeed and Climb Rate Controller Gains**

AIRSPEED [m/s]	PROPORTIONAL FEEDBACK GAINS ( $K_p$ )	INTEGRAL ERROR FEEDBACK GAINS ( $K_i$ )
14	$\begin{bmatrix} 0.08 & -0.98 \\ 0.88 & 0.18 \end{bmatrix}$	$\begin{bmatrix} 0.24 & -0.18 \\ 0.14 & 0.64 \end{bmatrix}$
16	$\begin{bmatrix} 0.08 & -0.84 \\ 0.84 & 0.16 \end{bmatrix}$	$\begin{bmatrix} 0.16 & -0.16 \\ 0.16 & 0.60 \end{bmatrix}$
18	$\begin{bmatrix} 0.1 & -0.66 \\ 0.84 & 0.10 \end{bmatrix}$	$\begin{bmatrix} 0.12 & -0.12 \\ 0.10 & 0.63 \end{bmatrix}$
20	$\begin{bmatrix} 0.08 & -0.6 \\ 0.80 & 0.10 \end{bmatrix}$	$\begin{bmatrix} 0.08 & -0.12 \\ 0.10 & 0.618 \end{bmatrix}$

**B.1.2 Guidance Controller**



**Figure B.4: Guidance Controller Block Diagram**

Figure B.4 shows the block diagram for the guidance controller. The gain scheduled feedback gains for the original guidance controller as designed in [1] and the high-gain guidance controller is given in Table B.3.

**Table B.3: Original and High Gain Guidance Controller Gains**

	ORIGINAL GUIDANCE CONTROLLER	HIGH GAIN GUIDANCE CONTROLLER
Airspeed [m/s]	Cross Track Error Feedback Gain ( $K_y$ )	Cross Track Error Feedback Gain ( $K_y$ )
10	0.85	1.1
12	0.7	0.82
14	0.6	0.72
16	0.52	0.62
18	0.47	0.54
20	0.42	0.54
22	0.38	0.54
24	0.35	0.54
26	0.32	0.54

The guidance controller is only redesigned for airspeeds less than 18m/s since the high gain guidance controller will only be used during landing.

## B.2 $X_B$ -Acceleration Controller

Figure B.5 shows the bode plot for the closed loop  $X_B$ -acceleration control system given by equation (4.4).

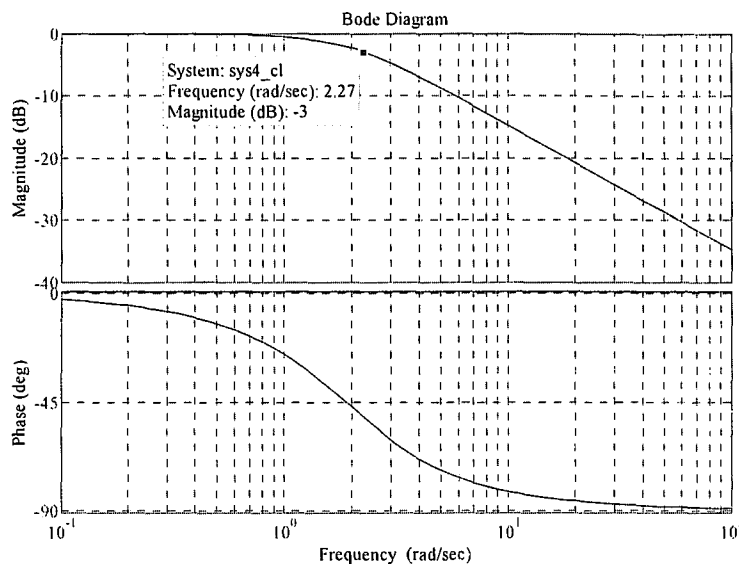


Figure B.5:  $X_B$ -Acceleration Controller Frequency Response

## B.3 Runway Yaw Rate Controller

Table B.4 gives the proportional and error integral gains for the runway yaw rate controller.

Table B.4: Runway Yaw Rate Proportional and Error Integral Gains

RUNWAY YAW RATE CONTROLLER					
Airspeed [m/s]	Proportional Feedback Gain ( $K_{p_{yr}}$ )	Error Integral Feedback Gain ( $K_{I_{yr}}$ )	Airspeed [m/s]	Proportional Feedback Gain ( $K_{p_{yr}}$ )	Error Integral Feedback Gain ( $K_{I_{yr}}$ )
1	-0.5588	-0.0310	11	-0.2623	-0.0146
2	-0.5434	-0.0302	12	-0.2381	-0.0132
3	-0.5196	-0.0289	13	-0.2164	-0.0120
4	-0.4896	-0.0272	14	-0.1970	-0.0109
5	-0.4558	-0.0253	15	-0.1797	-0.0100
6	-0.4203	-0.0233	16	-0.1643	-0.0091
7	-0.3848	-0.0214	17	-0.1505	-0.0084
8	-0.3507	-0.0195	18	-0.1382	-0.0077
9	-0.3187	-0.0177	19	-0.1273	-0.0071
10	-0.2892	-0.0161	20	-0.1174	-0.0065

## B.4 Runway Heading Controller

Figure B.6 shows the bode plot for the closed loop system runway yaw rate control system given by equation (4.18).

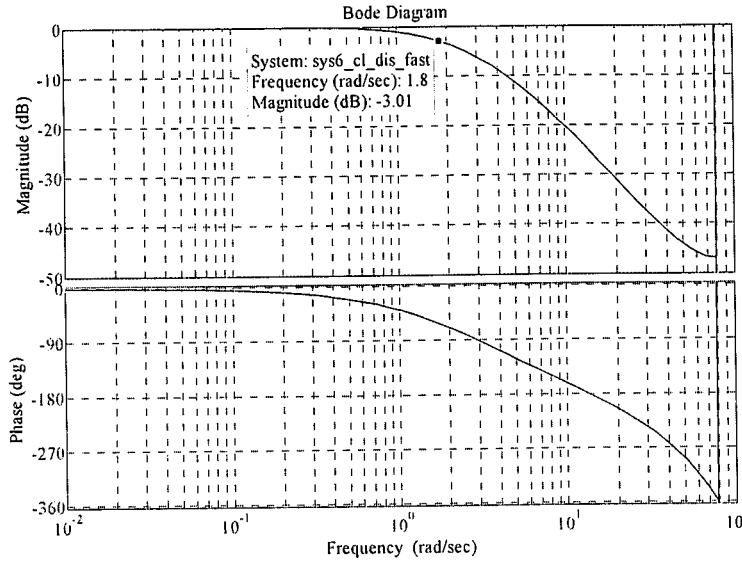


Figure B.6: Runway Yaw Rate Controller Frequency Response

## B.5 Runway Guidance Controller

The set of feedback gains that satisfies the design specifications for the runway guidance controller at different groundspeeds are listed in Table B.5.

Table B.5: Runway Guidance Controller Gains

RUNWAY GUIDANCE CONTROLLER			
Groundspeed [m/s]	Cross Track Error Feedback Gain ( $K_y$ )	Groundspeed [m/s]	Cross Track Error Feedback Gain ( $K_y$ )
1	5.2	7	1.4
2	4.2	8	1.2
3	3.2	9	1.1
4	2.4	10	1
5	1.9	11	0.9
6	1.6	12	0.8

## B.6 Landing Sequence Parameters

The following variable values were used for the landing path (Figure 2.4):

**Table B.6:** *Landing Path Parameters*

SIMULATION		FLIGHT TESTING	
Approach Range ( $R_{app}$ )	250m	Approach Range ( $R_{app}$ )	300m
Glide Path Start Range ( $R_{GP}$ )	200m	Glide Path Start Range ( $R_{GP}$ )	250m
Glide Path Angle ( $\gamma$ )	4°	Glide Path Angle ( $\gamma$ )	4°
Glide Path Offset	0m	Glide Path Offset	0m
Path heading ( $\Psi_R$ )	0°	Path heading ( $\Psi_R$ )	341°

The approach point is located at:

**Table B.7:** *Approach Point Parameters*

SIMULATION		FLIGHT TESTING	
North displacement ( $N_{app}$ )	-250m	North displacement ( $N_{app}$ )	-283.66m
East displacement ( $E_{app}$ )	0m	East displacement ( $E_{app}$ )	97.67m
Heading	0°	Heading	341°
Altitude ( $h_{app}$ )	13m	Altitude ( $h_{app}$ )	17m

# Appendix C

## Aircraft Data

### C.1 Engine Data

The engine used in this project is the Irvine 0.53. The RPM and manifold pressure (MAP) ranges used for an engine similar to the Irvine 0.53 are:

$$\text{RPM} = [2550 \ 3570 \ 4760 \ 5950 \ 7650 \ 8670 \ 9350 \ 10200 \ 11900] \quad (\text{C.1})$$

$$\text{MAP} = [60 \ 70 \ 80 \ 90 \ 92 \ 94 \ 96 \ 98 \ 100] \text{ kPa} \quad (\text{C.2})$$

These typical values were obtained from [13]. The power lookup table used for the Irvine 0.53 engine is given in Table C.1. These values are again not the actual values for the Irvine 0.53 engine, but rather typical values for a similar engine obtained from [13].

**Table C.1:** Engine Power Lookup Table

RPM	2550	6.5565	8.1948	57.365	58.73	60.104	58.73	58.73	60.104	75.122
	3570	7.7452	8.6052	110.91	130.03	131.95	139.6	154.9	174.02	195.05
	4760	8.1591	13.003	163.18	206.52	216.72	221.83	267.72	318.71	346.76
	5950	14.343	14.023	213.54	267.72	283.65	305.97	366.52	427.08	462.13
	7650	15.776	31.962	354.04	457.3	467.14	516.31	590.08	585.97	672.03
	8670	15.79	32.044	406.82	561.48	597.69	596.31	633.46	715.19	863.8
	9350	16.027	38.314	420.7	605.51	644.57	649.09	716.2	831.38	921.53
	10200	14.205	48.08	439.27	639.25	688.97	721.2	823.37	983.45	1038.1
	11900	10.837	43.344	436	662.92	729.21	795.51	953.59	1083.6	1243
		60	70	80	90	92	94	96	98	100
		MAP								

The engines moment of inertia is taken as a typical value:

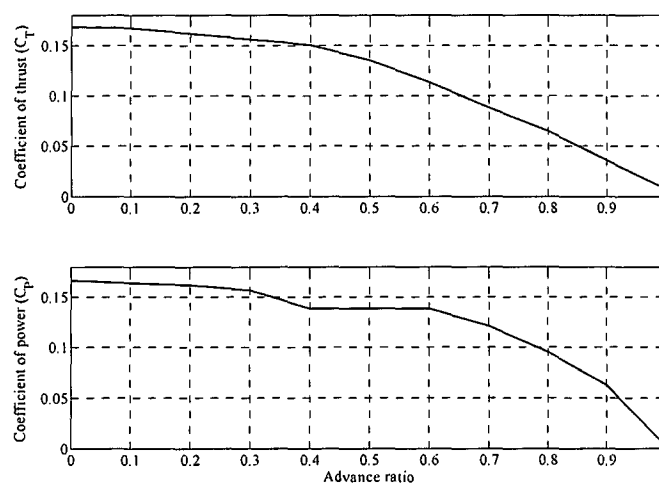
$$I_E = 0.1 \times 10^{-3} \text{ kgm}^2 \quad (\text{C.3})$$

## C.2 Propeller Data

The coefficients of thrust and power versus the advance ratio used for the propeller, are given in Table C.2. The values are shown graphically in Figure C.1. These coefficients were not measured and are derived from typical thrust and power coefficients for a similar propeller and engine combination. These typical values were obtained from [13] and were scaled until the RPM range and max thrust matched those of the Irvine 0.53 engine and propeller combination.

**Table C.2:** *Coefficients of Thrust and Power vs. Advance Ratio*

ADVANCE RATIO ( $J_p$ )	COEFFICIENT OF THRUST ( $C_T$ )	COEFFICIENT OF POWER ( $C_P$ )
0	0.103	0.1188
0.1	0.102	0.117
0.2	0.101	0.1152
0.3	0.1	0.1116
0.4	0.091	0.108
0.5	0.0812	0.1044
0.6	0.0693	0.099
0.7	0.0539	0.0864
0.8	0.0396	0.0684
0.9	0.022	0.045
1	0.0044	0.018



**Figure C.1:** *Coefficients of Thrust and Power vs. Advance Ratio*



The propeller radius is:

$$R_p = 0.115 \text{ [m]} \quad (\text{C.4})$$

### C.2.1 Inertia

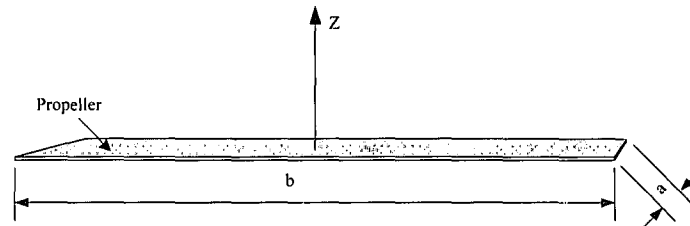


Figure C.2: Propeller Inertia Geometry

The propeller is approximated as a thin flat plate as shown in Figure C.2. The moment of inertia around the Z-axis of a fin plate is given by [14]:

$$I_{zz} = \frac{1}{12} m_p \cdot (a^2 + b^2) \quad (\text{C.5})$$

The dimensions in meters are:  $a = 0.025$  and  $b = 0.23$ . The propeller mass ( $m_p$ ) is 0.022 kg. The moment of inertia around the Z-axis is therefore:

$$I_{zz} = I_p = 0.1 \times 10^{-3} \quad (\text{C.6})$$

## C.3 Sensor Noise

The noise RMS (variance) value used for each sensor for simulation purposes is summarised in Table C.3. The values were obtained from practical measurements, datasheets and [1].

Table C.3: Sensor Noise Variances

SENSOR	NOISE VARIANCE	UNITS	SENSOR	NOISE VARIANCE	UNITS
GPS groundspeed	0.2	m/s	Y-axis Gyro	0.4	deg/s
GPS longitude	391.9639e-9	deg	Z-axis Gyro	0.4	deg/s
GPS latitude	391.9639e-9	deg	X-axis Accelerometer	0.15	m/s <sup>2</sup>
GPS meters above sea level	5	m	Y-axis Accelerometer	0.34	m/s <sup>2</sup>
GPS heading	2	deg	Z-axis Accelerometer	0.2	m/s <sup>2</sup>
Pressure airspeed	0.5	m/s	X-axis Magnetometer	4e-6	Gauss
Pressure altitude	0.5	m	Y-axis Magnetometer	4e-6	Gauss
Pressure climb rate	1	m/s	Z-axis Magnetometer	4e-6	Gauss
X-axis Gyro	0.4	deg/s	Ultrasonic altitude	0.025	m

# Appendix D

## Stall Speed Calculations

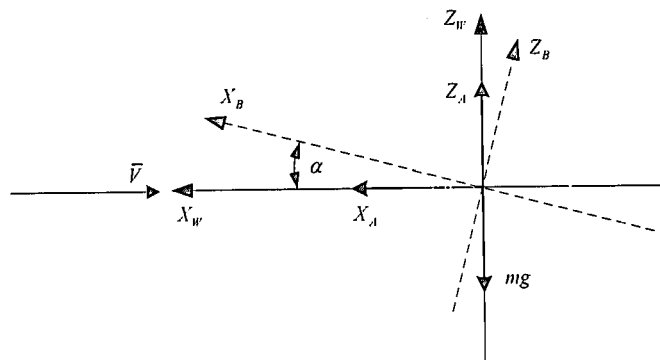


Figure D.1: Wind and Body Axes

In steady level trimmed flight, the lift generated by the wing must equal the gravitational force acting on the aircraft for the aircraft to stay in the air. Figure D.1 shows the geometry. In equation form this statement becomes:

$$Z_A = \frac{1}{2} \rho \bar{V}^2 S C_z = mg \quad (\text{D.1})$$

where  $C_z$  is the non-dimensional proportionality constant for the  $Z_w$ -axis force and is given by [1]:

$$C_z = -C_D \sin(\alpha) - C_L \cos(\alpha) \quad (\text{D.2})$$

with

$$C_L = C_{L_w} + \left( \frac{S_T}{S} \right) C_{L_T} \quad (\text{D.3})$$

$$C_D = C_{D_w} + \left( \frac{S_B}{S} \right) C_{D_B} \quad (\text{D.4})$$

The subscripts  $W$ ,  $T$  and  $B$  indicate wing, tail plane and fuselage (body) respectively.  $C_L$  and  $C_D$  is the lift and drag coefficients of the respective airfoils and  $S$  the surface area. The lift and drag coefficients of the respective surfaces are given by [1]:

$$C_{L_T} = \frac{\partial C_{L_T}}{\partial \alpha} \left[ \alpha_T + \left( 1 - \frac{\partial \varepsilon}{\partial \alpha} \right) \alpha \right] \quad (D.5)$$

$$C_{L_W} = \frac{\partial C_{L_W}}{\partial \alpha} \alpha + C_{L_{W_0}} + \left( \frac{S_T}{S} \right) C_{L_T} \quad (D.6)$$

$$C_{D_W} = C_{D_{W_0}} + k_W C_{L_W}^2 \quad (D.7)$$

$$C_{D_B} = C_{D_{B_0}} \quad (D.8)$$

where  $\frac{\partial C_{L_W}}{\partial \alpha}$  and  $\frac{\partial C_{L_T}}{\partial \alpha}$  are the wing and tail plane lift curve slopes respectively.  $C_{L_{W_0}}$  is the zero angle of attack lift coefficient of the wing,  $\alpha_T$  is the tail plane setting angle and  $\frac{\partial \varepsilon}{\partial \alpha}$  is the change in down wash angle with angle of attack.  $C_{D_{W_0}}$  is the wing parasitic drag coefficient,  $k_W$  is the wing induced drag coefficient and  $C_{D_{B_0}}$  is the fuselage parasitic drag coefficient.

The parameter values for the aircraft used in this project are summarised in Table D.1.

**Table D.1:** Aircraft Parameters Used During Stall Speed Calculations

PARAMETER	VALUE	PARAMETER	VALUE
$\frac{S_T}{S}$	0.1976	$\alpha_T$	0
$\frac{S_B}{S}$	0.8072	$C_{L_{W_0}}$	0
$\frac{\partial C_{L_T}}{\partial \alpha}$	3.8318	$C_{D_{W_0}}$	0.01347
$\frac{\partial C_{L_W}}{\partial \alpha}$	4.658	$k_W$	0.00601
$\frac{\partial \varepsilon}{\partial \alpha}$	0.2381	$C_{D_{B_0}}$	0.004898

With a stall angle of attack of  $15^\circ$  the non-dimensional proportionality constant,  $C_z$  is given by:

$$C_z = -2.09454 \quad (D.9)$$

The relationship of equation (D.1) then becomes:

$$0.561 \bar{V}^2 = (3)(9.81) \quad (D.10)$$

The stall speed of the aircraft used in this project is therefore:

$$\bar{V}_{stall} = 7.3 \quad [m/s] \quad (D.11)$$

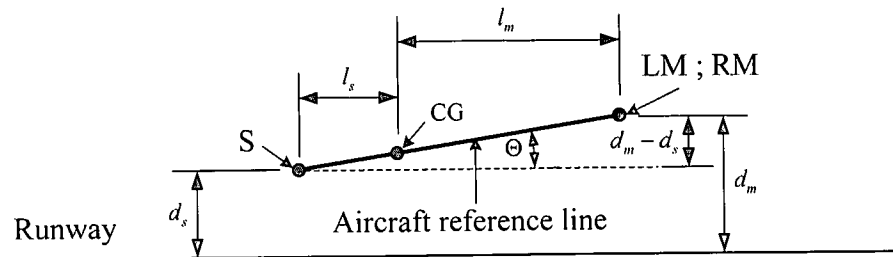
The take-off and landing airspeed is then:

$$2 \cdot \bar{V}_{stall} = 15 \quad [m/s] \quad (D.12)$$

## Appendix E

### *Runway Model Initial CG and $\Theta$ Calculations*

In order for the aircraft to start in its natural position on the runway and not display transient motion due to the suspension when the hardware in the loop simulation starts, the initial centre of gravity position and pitch angle of the aircraft must be calculated. These values are then the initial conditions supplied to the 6DoF block.



**Figure E.1:** *Geometry for Initial CG and Pitch Angle Calculation for the Runway Model*

With reference to Figure E.1, the force produced by the suspension at the steering wheel and a single main wheel is given by:

$$F_s = (l_r - d_s)k_s \quad (\text{E.1})$$

$$F_m = (l_r - d_m)k_s \quad (\text{E.2})$$

$l_r$  is the z-position of each suspension point (S, LM and RM in Figure 3.4) in the uncompressed state.  $d_s$  and  $d_m$  are the z-positions of the steering and main wheels suspension points in the compressed state (z-position when resting on the runway) and  $k_s$  is the suspension spring constant. The force acting on the steering wheel because of the weight of the aircraft is given by:

$$F_s = \left( \frac{l_m}{l_m + l_s} \right) \cdot mg \quad (\text{E.3})$$

The expression in brackets describes the portion of the weight transferred to the steering wheel.

$l_s$  is the length from the centre of gravity to the steering wheel suspension point and  $l_m$  the length from the centre of gravity to the main wheels centre line (refer to Figure 3.4). The force acting on a single main wheel due to the weight of the aircraft is given by:

$$F_m = \frac{1}{2} \cdot \left( \frac{l_s}{l_m + l_s} \right) \cdot mg \quad (\text{E.4})$$

Here the expression in brackets describes the portion of the weight transferred to the main wheels. By substituting equation (E.3) and (E.4) into equation (E.1) and (E.2), the z-positions in the compressed state can be described as:

$$d_s = l_r - \left( \frac{l_m}{l_m + l_s} \right) \frac{mg}{k_s} \quad (\text{E.5})$$

$$d_m = l_r - \left( \frac{l_s}{l_m + l_s} \right) \frac{mg}{2k_s} \quad (\text{E.6})$$

The pitch angle of the aircraft, when resting on the runway, can then be calculated as:

$$\Theta_0 = \sin^{-1} \left( \frac{|d_s - d_m|}{l_s + l_m} \right) \quad (\text{E.7})$$

And the initial z-position for the centre of gravity when the aircraft is resting on the runway, is given by:

$$Z_{CG_0} = l_m \sin(\Theta_0) + d_m \quad (\text{E.8})$$



# Appendix F

## *Autonomous Flight of a Model Aircraft – Additional Information*

This appendix provides more detail on the work done in [1], since all the work in this thesis builds directly on it. The appendix begins by defining the axis systems used and a summary of the equations of motion. The equations of motion and the forces and moments are then linearised and combined to form a linear aircraft model. This linear model is expressed in state space form and is the model used to design all the airborne controllers of Chapter 4 and Chapter 5.

### F.1 Axis System Definition

#### F.1.1 Inertial Axes

Figure F.1 defines the inertial reference frame used in [1] and in this project for the application of Newton's laws. The inertial axis system  $F_E = (O_E, X_E, Y_E, Z_E)$  is right handed and earth fixed. "The earth is considered flat and non-rotating, justified by the limited flight range and low translation velocity of a RC aircraft. Its origin coincides with at the runway location below the aircraft centre of gravity at the start of the motion studied".

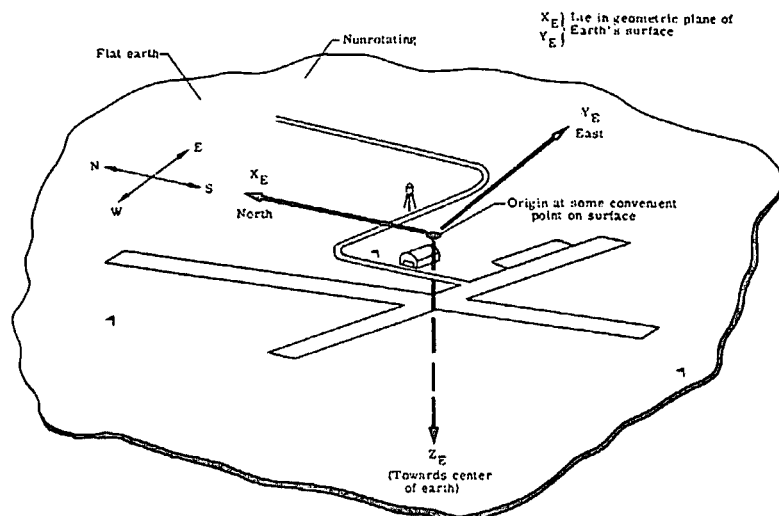


Figure F.1: Inertial Axes Definition [15]

## F.1.2 Body Fixed Axes

Two of the body fixed axis systems used in this project is *Body Axes* and *Stability (Wind) Axes*. The body axis system  $F_B = (O_B, X_B, Y_B, Z_B)$  is defined in Figure F.2. Both body axis systems are right handed with the centre coinciding with the aircraft's centre of gravity.

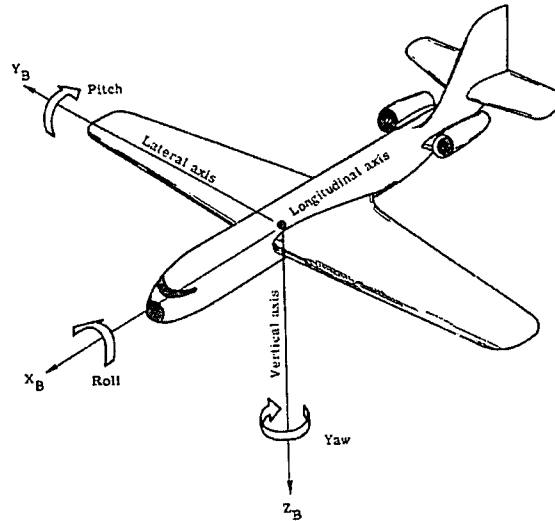


Figure F.2: Body Axes Definition [15]

## F.2 Equations of Motion

Equations (F.1) to (F.6) provides the set of six, non-linear, coupled differential equations for general six degree of freedom motion. Equation (F.7) gives the set of differential equations that allow the dynamic calculation of the Euler angles. Equation (F.8) relates the body axes velocities to the inertial axes velocities through the Direction Cosine Matrix (DCM). It is assumed that the aircraft is symmetric about the  $X_B Z_B$ -plane, resulting in the products of inertia,  $I_{xy}$  and  $I_{yz}$  being zero. It is also assumed that the inertial properties do not change over time.

$$X = m(\dot{U} + WQ - VR) \quad (F.1)$$

$$Y = m(\dot{V} + UR - WP) \quad (F.2)$$

$$Z = m(\dot{W} + VP - UQ) \quad (F.3)$$

$$L = I_x \dot{P} - I_{xz} \dot{R} + (I_z - I_y) QR - I_{xz} PQ \quad (F.4)$$

$$M = I_y \dot{Q} + (I_x - I_z) PR + I_{xz} (P^2 - R^2) \quad (F.5)$$

$$N = I_z \dot{R} - I_{xz} \dot{P} + (I_y - I_x) PQ - I_{xz} QR \quad (F.6)$$

$$\begin{bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} 1 & \cos \Phi \tan \Theta & \cos \Phi \tan \Theta \\ 0 & \cos \Phi & -\sin \Phi \\ 0 & \sin \Phi \sec \Theta & \cos \Phi \sec \Theta \end{bmatrix} \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \quad (F.7)$$

$$\begin{bmatrix} \dot{N} \\ \dot{E} \\ \dot{D} \end{bmatrix} = \begin{bmatrix} C_\psi C_\theta & (C_\psi S_\theta S_\phi - S_\psi C_\phi) & (C_\psi S_\theta C_\phi + S_\psi S_\phi) \\ S_\psi C_\theta & (S_\psi S_\theta S_\phi + C_\psi C_\phi) & (S_\psi S_\theta C_\phi - C_\psi S_\phi) \\ -S_\theta & C_\theta S_\phi & C_\theta C_\phi \end{bmatrix} \begin{bmatrix} U \\ V \\ W \end{bmatrix} \quad (\text{F.8})$$

### F.3 Linear Aircraft Model

The equations of motion are linearised and then the full linear aircraft model is presented with the linear aerodynamic, gravitational and engine forces and moments included. The state space representation is given in section F.4.

#### F.3.1 Linear Equations of Motion

Applying small perturbation theory and assumptions to the non-linear equations of motion presented in section F.2, results in the following linear dynamic equations for the aircrafts motion:

$$X_A + X_G + X_T = m(\dot{u}) \quad (\text{F.9})$$

$$Y_A + Y_G + Y_T = m(\dot{v} + U_0 r) \quad (\text{F.10})$$

$$Z_A + Z_G + Z_T = m(\dot{w} - U_0 q) \quad (\text{F.11})$$

$$L_A + L_G + L_T = \dot{p}I_x - \dot{r}I_{xz} \quad (\text{F.12})$$

$$M_A + M_G + M_T = \dot{q}I_y \quad (\text{F.13})$$

$$N_A + N_G + N_T = \dot{r}I_z - \dot{p}I_{xz} \quad (\text{F.14})$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (\text{F.15})$$

$$\begin{bmatrix} \dot{P}_N \\ \dot{P}_E \\ \dot{P}_D \end{bmatrix} = \begin{bmatrix} U_0 + u \\ U_0 \psi + v \\ -U_0 \theta + w \end{bmatrix} \quad (\text{F.16})$$

In the equations above, the aerodynamic, gravitational and engine forces and moments are still non-linear functions and need to be linearised about the trim condition.

#### F.3.2 Linear Aircraft Model

Equations (F.17) to (F.22) form the linear aircraft model used for airborne controllers designed in Chapter 4 and Chapter 5.

$$X_{A_u} u + X_{A_w} w + X_{A_q} q + X_{A_{\dot{w}}} \dot{w} + X_{A_{\delta_e}} \delta_e - mg\theta + X_{T_u} u + X_{T_{\delta_t}} \delta_t = m(\dot{u}) \quad (\text{F.17})$$

$$Y_{A_v} v + Y_{A_p} p + Y_{A_r} r + Y_{A_{\delta_a}} \delta_a + Y_{A_{\delta_r}} \delta_r + mg\phi = m(\dot{v} + U_0 r) \quad (\text{F.18})$$

$$Z_{A_u} u + Z_{A_w} w + Z_{A_q} q + Z_{A_{\dot{w}}} \dot{w} + Z_{A_{\delta_e}} \delta_e + Z_{T_u} u + Z_{T_{\delta_t}} \delta_t = m(\dot{w} - U_0 q) \quad (\text{F.19})$$

$$L_{A_v} v + L_{A_p} p + L_{A_r} r + L_{A_{\delta_a}} \delta_a + L_{A_{\delta_r}} \delta_r = \dot{p}I_x - \dot{r}I_{xz} \quad (\text{F.20})$$

$$M_{A_u} u + M_{A_w} w + M_{A_q} q + M_{A_{\dot{w}}} \dot{w} + M_{A_{\delta_e}} \delta_e + M_{T_u} u + M_{T_{\delta_i}} \delta_i = \dot{q} I_y \quad (\text{F.21})$$

$$N_{A_v} v + N_{A_p} p + N_{A_r} r + N_{A_{\delta_a}} \delta_a + N_{A_{\delta_r}} \delta_r = \dot{r} I_z - \dot{p} I_{xz} \quad (\text{F.22})$$

The following example indicates the partial derivative notation used:

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

## F.4 Aircraft State Space Model

The linear longitudinal aerodynamic model for the aircraft in state space form as developed in [1], is given by:

$$\dot{\underline{x}}_l = \mathbf{A}_l \underline{x}_l + \mathbf{B}_l \underline{u}_l \quad (\text{F.23})$$

$$\dot{h} = \mathbf{C}_{l_h} \underline{x}_l \quad (\text{F.24})$$

$$\dot{p}_N = \mathbf{C}_{l_u} \underline{x}_l \quad (\text{F.25})$$

$$\dot{\delta}_i = -\frac{1}{\tau_T} \delta_i + \frac{1}{\tau_T} \delta_{i_c} \quad (\text{F.26})$$

with

$$\underline{x}_l = [u \quad w \quad q \quad \theta]^T \quad (\text{F.27})$$

$$\underline{u}_l = [\delta_e \quad \delta_i]^T \quad (\text{F.28})$$

$$\tau_T = 0.5 \quad (\text{F.29})$$

$$\mathbf{C}_{l_h} = \begin{bmatrix} 0 & -1 & 0 & \frac{U_0 \cdot \pi}{180} \end{bmatrix} \quad (\text{F.30})$$

$$\mathbf{C}_{l_u} = [1 \quad 0 \quad 0 \quad 0] \quad (\text{F.31})$$

where  $U_0$  is the trim velocity. The linear lateral aerodynamic model for the aircraft in state space form as developed in [1], is given by:

$$\dot{\underline{x}}_l = \mathbf{A}_l \underline{x}_l + \mathbf{B}_l \underline{u}_l \quad (\text{F.32})$$

$$\dot{\psi} = \mathbf{C}_{l_r} \underline{x}_l \quad (\text{F.33})$$

$$\dot{y} = k_{l_\psi} \psi + \mathbf{C}_{l_v} \underline{x}_l \quad (\text{F.34})$$

with

$$\underline{x}_l = [v \quad p \quad r \quad \phi]^T \quad (\text{F.35})$$

$$\underline{u}_l = [\delta_a \quad \delta_r]^T \quad (\text{F.36})$$

$$k_{l_\psi} = \frac{U_0 \cdot \pi}{180} \quad (\text{F.37})$$

$$C_{t_\psi} = [0 \ 0 \ 10 \ 0] \quad (\text{F.38})$$

$$C_{t_v} = [1 \ 0 \ 0 \ 0] \quad (\text{F.39})$$

where  $U_0$  is the trim velocity.

# Appendix G

## *Flight Test Plan*

### G.1 Introduction

#### G.1.1 Background

The Autopilot initially developed by Iain Peddle for his master's degree has been extended and re-implemented to form a next generation of the autopilot system. Testing is required for verification of the new hardware, as well as the accumulation of essential flight hours.

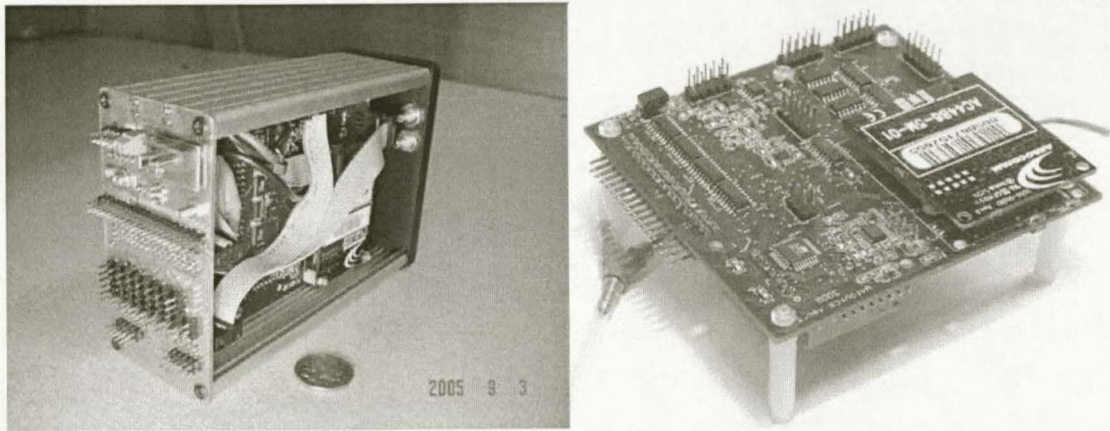
In order to extend the system to perform autonomous take-off and landing, it is necessary to establish the performance criteria of the new ultrasonic sensor in flight and on different ground surfaces, as well as initial testing of the new ground controllers and wind estimation procedure.

#### G.1.2 Test Objectives

- Collecting take-off and landing telemetry data.
- Piloted flight to ensure that all the sensors and specifically the ultrasonic are performing as required.
- Lateral and longitudinal controller testing (gain scheduling and no trim).
- Full autonomous flight.
- Wind estimation.
- Ground controller testing.



### G.1.3 Description of the Test Item



**Figure G.1:** *ESL Micro Avionics*

The system subjected to testing is version two of the Micro Autopilot shown in Figure G.1 that is being developed in the ESL.

The system is capable of full autonomous navigation between specified waypoints with each way point consisting of:

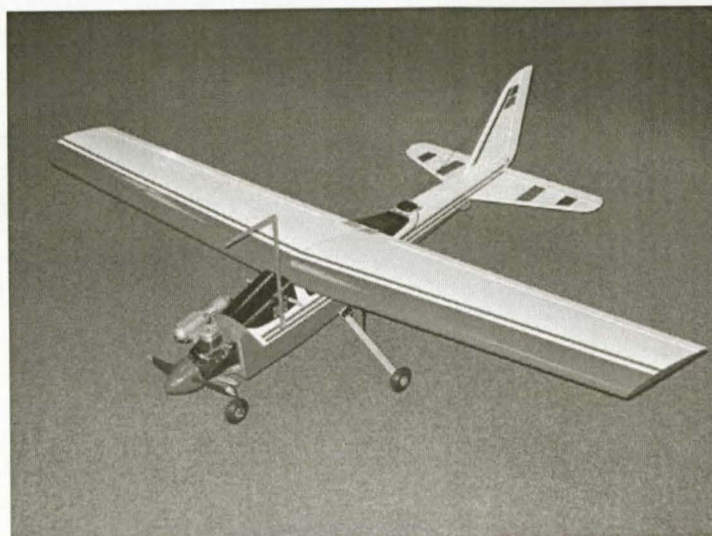
- 3D location in inertial axes.
- Heading.
- Airspeed.

The system is also theoretically capable of yaw rate control on the runway, as well as estimating the wind speed and direction. This needs to be verified through testing.

The hardware necessary to achieve full autonomous navigation and currently implemented includes:

- Four microcontrollers (2x ATmega128; 1x ATmega8; 1x Microchip DSpic 30F5011).
- Inertial measurement unit (IMU).
  - 3 x ADXRS150EB Rate Gyroscopes.
  - 2 x ADXL202AE Dual-Axis Accelerometers.
- Pressure sensors:
  - One absolute (MPX4115A).
  - One differential (MPXV5004G).
- u-Blox GPS.
- MaxStream RF module.
- SensComp ultrasonic sensor.

All system software is version 3 code as completed on 2006-06-09.



**Figure G.2:** *Reliance 0.46 Size Model Aircraft*

The aircraft used for the test is the reliance 0.46 size radio controlled trainer with an Irvine 0.53 piston engine and a fix pitch propeller as shown in Figure G.2.

### G.1.4 Test Scope

The tests will range from hardware (sensor) testing to controller testing for full autonomous flight and runway ground speed and yaw rate control. The hardware test will include a full system functionality test with specific focus on the ultrasonic readings and the GPS accuracy. The controller testing will follow a systematic procedure to ensure that all the controllers are working. Finally, the full autonomous navigation capability of the new system will be tested and the wind estimation procedure verified.

## G.2 Test Preparation

### G.2.1 Project Team

Study leaders: Prof G.W. Milne and Mr I.K. Peddle.

Safety Pilot: Dr. Kas Hamman.

System Engineer: Jan-Cor Roos.

Flight Test Coordinator: Willem Hough.

### G.2.2 Operations Support Requirements

None

### G.2.3 Flight Test Engineering Support Requirements

None.

## G.2.4 Logistical Support

Toyota Hi-Ace Panel Van.

## G.2.5 Test Operations Plan

None.

## G.2.6 Briefings and Debriefings

Full briefings and debriefings are to be held for each test.

## G.2.7 Safety

- The relevant test hazard analysis will be briefed before flight.
- First aid kit must be available.
- Fire services (extinguishers) must be available.
- General safety around the UAV will be briefed on the flight day.
- Refer and adhere to the checklist.

# G.3 Details of the Test

## G.3.1 Ground Controllers

### G.3.1.1 Objective

- To determine whether the runway yaw rate controller is working and to measure the performance of the controller.
- To determine the GPS heading accuracy at low speeds and yaw angle accuracy from integrating the yaw rate gyroscope.

### G.3.1.2 Test Methods

#### **Yaw Rate Controller**

Place the aircraft on the side of the runway (left for a positive yaw rate test and right for a negative yaw rate test). Arm the yaw rate controller from the ground station with a 30 deg/s yaw rate command.

Allow the safety pilot to slowly drive the aircraft forward and arm the yaw rate controller.

Once control is achieved, the aircraft can be steered from the ground station.



### GPS Heading and Yaw Angle Accuracy (Drift)

Allow the safety pilot to drive up and down the runway while logging telemetry data and enabling gyro integration.

#### G.3.1.3 Test Conditions

- Fine clear weather conditions with a wind speed of less than 15km/h.
- A smooth clear runway with no obstacles on the side and a width of at least 6m.

#### G.3.1.4 Data Required

**Table G.1:** *Data Required for Ground Controllers*

PARAMETER	UNITS	SOURCE
Groundspeed	m/s	GPS
Body axes yaw rate	deg/s	Rate Gyroscopes (IMU)
Heading	deg	GPS
Yaw angle	deg	Integrated yaw rate
Servo positions	deg	Servos

#### G.3.1.5 Data Analysis

Post data processing in MATLAB is required to generate time plots of the logged data. The plots are generated by opening “PlotTelemetryData.m” and selecting the data required for plotting and post analysis. After the required plots are specified, run the m-file. The following graphs are required over time:

- Ground speed vs. time.
- GPS heading vs. time.
- Yaw rotation rate (in body axes).
- Yaw angle vs. time.
- Actual and Pilot servo commands.

#### G.3.1.6 Acceptance Criteria

##### Yaw Rate Controller

- Rise time of less than 4s.
- Overshoot of less than 10%.
- Zero steady state error.
- Yaw rate steps of up to 30deg/s are allowed.

## G.3.2 Piloted Flight with Telemetry Logging and Runway Flybys

### G.3.2.1 Objective

- To determine the flying ability of the reliance 0.46 aircraft with the avionics set onboard.
- Allowing the safety pilot to familiarise himself with the aircraft and system.
- During flight all telemetry data from the avionics pack will be logged to determine if all the sensors onboard, with their signal conditioning are working.
- Collection of telemetry data during take-off and landing.
- Determining the accuracy and reliability of the ultrasonic sensor during flight and on different ground surfaces.
- Testing the recalibration of the absolute pressure sensor from the ultrasonic measurement.

### G.3.2.2 Test Method

Take-off and manual flight using standard procedures (non-acrobatic flight) while all telemetry data is logged for post processing and analysis.

Allow time for the absolute pressure sensor (altitude measurement) to drift (10 min should be sufficient).

Let the pilot do a couple of runway flybys with the ultrasonic sensor enabled and recalibration of the absolute pressure sensor requested.

Allow a couple of flybys (off the runway) with the ultrasonic sensor enabled and recalibration of the absolute pressure sensor requested for comparison with the previous procedure.

Land the aircraft manually and ensure that the ultrasonic sensor is active during landing.

### G.3.2.3 Test Conditions

- Fine clear weather condition with a wind speed of less than 15km/h is required.
- Flight of the reliance 0.46 aircraft must be performed in a non-acrobatic fashion with an airspeed limit of 30m/s, an altitude limit of 250m and a range limit of 2000m (2km).
- A clear runway with no other aircraft is required for runway flybys.

### G.3.2.4 Data Required

**Table G.2:** *Data Required for Piloted Flight with Telemetry Logging and Runway Flybys*

PARAMETER	UNITS	SOURCE
Airspeed	m/s	Differential pressure sensor
Groundspeed	m/s	GPS
Altitude	m	Absolute pressure sensor and GPS
Body axes rates (Roll, Pitch, Yaw)	deg/s	Rate Gyroscopes (IMU)
Accelerations (X,Y,Z axes)	m/s <sup>2</sup>	Accelerometers (IMU)
Climb rate	m/s	Absolute pressure sensor
Heading	deg	GPS
Ultrasonic altitude	m	Ultrasonic
Ultrasonic climb rate	m/s	Ultrasonic
Servo positions	deg	Servos

### G.3.2.5 Data Analysis

Post data processing in MATLAB is required to generate time plots of the logged flight data. The plots are generated by opening “PlotTelemetryData.m” and selecting the data required for plotting and post analysis. After the required plots are specified, run the m-file. The following graphs are required over time:

- Airspeed, climb rate and altitude.
- Airspeed vs. ground speed.
- Pressure altitude vs. GPS altitude.
- Pressure altitude vs. ultrasonic altitude.
- Pressure climb rate vs. ultrasonic climb rate.
- Accelerations (in body axes).
- Rotation rate (in body axes).
- Pilot servo commands.

### G.3.2.6 Acceptance Criteria

Since the test is a general flight test, no specific values for the data are required. However, all flight data must adhere to the expected results that correspond with the flight performed by the safety pilot. The safety pilot must be satisfied with the flying ability of the aircraft with the avionics on board.



### G.3.3 Longitudinal Controllers

#### G.3.3.1 Objective

- To test the functionality and performance of the longitudinal controllers onboard the avionics set with specific focus on the controllers that are gain scheduled. The controllers to be tested are:
  - Airspeed and climb rate controller.
  - Altitude controller.
- Collecting telemetry data during take-off and landing, specifically ultrasonic readings

#### G.3.3.2 Test Method

Take-off must be performed by the safety pilot. After take-off and before testing any one of the controllers the aircraft must be stabilised in level flight at an approximate airspeed of 24m/s, an altitude of 100m and above the runway. Ensure that the aircraft remains in a 2000m (2km) range of the ground station throughout the testing of any controller.

Each subsequent controller test as specified is dependant on the correct functionality of the previous controllers. If any controller should fail during testing, it must not be used in the next controller test. Only the working controllers and the current controller being tested should be armed.

After the completion of the test the safety pilot is responsible for the safe landing of the aircraft.

The ground station operator must be notified when the landing approach is started so the ultrasonic can be enabled.

#### **Airspeed and Climb Rate Controller**

Once level flight is reached, arm the airspeed and climb rate controller with an airspeed command of 24m/s and a climb rate command of 0m/s. Allow the aircraft to settle in the commanded state before issuing further commands. After the aircraft has settled, issue positive step commands in both airspeed and climb rate. The recommended commands are: Airspeed of 30m/s; Climb Rate of 2m/s. Again allow the aircraft to settle in the new commanded state. After the aircraft has settled, issue negative step commands in both airspeed and climb rate. The recommended commands are: Airspeed of 24m/s; Climb Rate of -2m/s.

#### **Altitude Controller**

When the aircraft reaches level flight, arm the airspeed and climb rate controller with an airspeed command of 24m/s and a climb rate command of 0m/s. After the aircraft has settled, arm the altitude controller with an altitude command of 100m. Allow the aircraft to settle in the commanded state before issuing further commands. After the aircraft has settled, issue a positive step command in altitude. An altitude of 150m is recommended. Again allow the aircraft to settle in the new commanded state. After the aircraft has settled, command a negative step in altitude. An altitude of 100m is recommended.

To test the gain scheduling of the controller, let the aircraft settle at 100m altitude at 16m/s. Step the altitude controller to 150m altitude command. Once the altitude and airspeed have settled, command the airspeed to 14m/s and allow settling time. An altitude command of 100m should then be given.

### G.3.3.3 Test Conditions

- Fine clear weather condition with a wind speed of less than 15km/h is required.
- Flight of the reliance 0.46 aircraft must be performed in a non-acrobatic fashion with an airspeed limit of 30m/s, an altitude limit of 250m and a range limit of 2000m (2km).
- Before testing any one of the controllers, the aircraft must be stabilised in level flight at an approximate airspeed of 24m/s at an altitude of 100m.

### G.3.3.4 Data Required

**Table G.3:** Data Required for Longitudinal Controllers

PARAMETER	UNITS	SOURCE
Airspeed	m/s	Differential pressure sensor
Groundspeed	m/s	GPS
Altitude	m	Absolute pressure sensor and GPS
Body axes rates (Roll, Pitch, Yaw)	deg/s	Rate Gyroscopes (IMU)
Accelerations (X,Y,Z axes)	m/s <sup>2</sup>	Accelerometers (IMU)
Ultrasonic altitude	m	Ultrasonic
Ultrasonic climb rate	m/s	Ultrasonic
Climb rate	m/s	Absolute pressure sensor
Servo positions	deg	Servos

### G.3.3.5 Data Analysis

Post data processing of all the required data in MATLAB is required to generate time plots of the logged flight data. The plots are generated by opening "PlotTelemetryData.m" and selecting the data required for plotting and post analysis. After the required plots are specified, run the m-file. The following graphs are required over time:

- Airspeed, Climb rate and Altitude.
- Airspeed vs. Ground speed.
- Pressure altitude vs. GPS altitude.
- Accelerations (in body axes).
- Rotation rate (in body axes).
- Actual, autopilot and pilot servo commands.
- Longitudinal controller data (Airspeed, Climb rate, Altitude and Pitch rate).
- Longitudinal controller performance (Commanded vs. sensed airspeed, climb rate and altitude).
- Pressure altitude and climb rate versus ultrasonic altitude and climb rate.

### G.3.3.6 Acceptance Criteria

The controllers must reach the following design specifications:

#### **Airspeed and Climb Rate Controller**

- Rise time of less than 3s.
- Overshoot of less than 20%.
- Zero steady state error.

#### **Altitude Controller**

- Rise time of less than 6s (for an unsaturated step).
- Overshoot of less than 20%.
- Zero steady state error.

The controller performance should remain relatively unchanged at the different airspeeds.

## G.3.4 Lateral Controllers

### G.3.4.1 Objective

- To test the functionality and performance of the lateral controllers onboard the avionics set with specific focus on the controllers that are gain scheduled. The controllers to be tested are:
  - Turn rate controller (YOCS).
  - Heading controller.
- Collecting telemetry data during take-off and landing, specifically ultrasonic readings.

### G.3.4.2 Test Method

Take-off must be performed by the safety pilot. After take-off and before testing any one of the controllers the aircraft must be stabilised in level flight at an approximate airspeed of 24m/s and an altitude of 100m. Ensure that the aircraft remains in a 2000m (2km) range of the ground station throughout the testing of any controller.

Each subsequent controller test as specified is dependant on the correct functionality of the previous controllers. If any controller should fail during testing it must not be used in the next controller test and only the working controllers and the current controller being tested should be armed.

After completion of the tests the safety pilot is responsible for the safe landing of the aircraft.

The ground station operator must be notified when the landing approach is started so the ultrasonic can be enabled.

### Turn Rate Controller (YOCS)

Once level flight is reached, arm the airspeed and climb rate controller with an airspeed command of 24m/s and a climb rate command of 0m/s. The altitude controller must be armed with an altitude command of 100m. After the aircraft has settled, arm the Dutch Roll Damper to ensure that it is stable. The turn rate controller with a turn rate command of 0deg/s can now be armed. Allow the aircraft to settle in the commanded state before issuing further commands. After the aircraft has settled, issue a positive step command in turn rate. The recommended command is: Turn Rate of 10deg/s. Again allow the aircraft to settle in the new commanded state. Command a turn rate of 0deg/s to bring the aircraft back to level flight and allow settling time. Command a negative step in turn rate. The recommended command is: Turn Rate of -10deg/s.

The above procedure should be repeated at 16m/s and 14m/s airspeeds to test the gain scheduling of the controller.

### Heading Controller

Once level flight is reached, arm the airspeed and climb rate controller with an airspeed command of 24m/s and a climb rate command of 0m/s. The altitude controller must be armed with an altitude command of 100m. After the aircraft has settled, arm the heading controller with a heading command of 90°. Allow the aircraft to settle in the commanded state before issuing further commands. After the aircraft has settled, issue a positive step command in heading. The recommended command is: Heading of 270°. Again allow the aircraft to settle in the new commanded state. After the aircraft has settled, command a negative step in heading. The recommended command is: Heading of 90°.

#### G.3.4.3 Test Conditions

- Fine clear weather condition with a wind speed of less than 15km/h is required.
- Flight of the reliance 0.46 aircraft must be performed in a non-acrobatic fashion with an airspeed limit of 30m/s, an altitude limit of 250m and a range limit of 2000m (2km).
- Before testing any one of the controllers, the aircraft must be stabilised in level flight at an approximate airspeed of 24m/s and an altitude of 100m.

### G.3.4.4 Data Required

**Table G.4:** Data Required for Lateral Controllers

PARAMETER	UNITS	SOURCE
Airspeed	m/s	Differential pressure sensor
Groundspeed	m/s	GPS
Altitude	m	Absolute pressure sensor and GPS
Body axes rates (Roll, Pitch, Yaw)	deg/s	Rate Gyroscopes (IMU)
Accelerations (X,Y,Z axes)	m/s <sup>2</sup>	Accelerometers (IMU)
Climb rate	m/s	Absolute pressure sensor
Heading	deg	GPS
Turn rate	deg/s	Yaw rate gyro (IMU)
Ultrasonic altitude	m	Ultrasonic
Ultrasonic climb rate	m/s	Ultrasonic
Servo positions	deg	Servos

### G.3.4.5 Data Analysis

Post data processing of all the required data in MATLAB is required to generate time plots of the logged flight data. The plots are generated by opening “PlotTelemetryData.m” and selecting the data required for plotting and post analysis. After the required plots are specified, run the m-file. The following graphs are required over time:

- Airspeed, Climb rate and Altitude.
- Airspeed vs. Ground speed.
- Pressure altitude vs. GPS altitude.
- Accelerations (in body axes).
- Rotation rate (in body axes).
- Actual, autopilot and pilot servo commands.
- Lateral controller data (Roll rate, Yaw rate, Turn rate and heading).
- Longitudinal controller data (Airspeed, Climb rate, Altitude and Pitch rate).
- Lateral controller performance (Commanded vs. sensed Turn rate and heading).
- Longitudinal controller performance (Commanded vs. sensed airspeed, climb rate and altitude).
- Pressure altitude vs. ultrasonic altitude.
- Pressure climb rate vs. ultrasonic climb rate.

### G.3.4.6 Acceptance Criteria

The controllers must reach the following design specifications:

### Turn Rate Controller

- Rise time of less than 2s.
- Overshoot of less than 20%.
- Zero steady state error.
- Yaw rate steps of up to 15deg/s are allowed.

### Heading Controller

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

The controller performance should remain relatively unchanged at the different airspeeds.

## G.3.5 Autonomous Navigation and Wind Estimation

### G.3.5.1 Objective

- To test full autonomous navigated flight of the reliance 0.46 aircraft through a series of way points where each way points consists of:
  - 3D location in inertial axes.
  - Heading.
  - Airspeed.
- Testing the wind estimation procedure and the activation of the landing sequence.
- Collecting telemetry data during take-off and landing, specifically ultrasonic readings.

### G.3.5.2 Test Method

Take-off is done manually. Stabilise the aircraft after take-off in level flight at an approximate airspeed of 24m/s and an altitude of 100m. Update way points if desired; arm the navigator from the ground station and allow the safety pilot to arm the autopilot when ready.

Once the navigator testing is complete, bring the aircraft to level flight at an approximate airspeed of 24m/s and 100m altitude above the runway. Enable wind estimation and send a landing request so that the landing sequence is pending. Once the pilot is comfortable, the autopilot can be armed for the wind estimation procedure to begin.

When the procedure is complete the pilot should regain command and control of the aircraft and is responsible for the safe landing of the aircraft.

The ground station operator must be notified when the landing approach is started so the ultrasonic can be enabled.



### G.3.5.3 Test Conditions

- Fine clear weather condition with a wind speed of less than 15km/h is required.
- Flight of the reliance 0.46 aircraft must be performed in a non-acrobatic fashion with an airspeed limit of 30m/s, an altitude limit of 250m and a range limit of 2000m (2km).

### G.3.5.4 Data Required

**Table G.5:** Data Required for Autonomous Navigation and Wind Estimation

PARAMETER	UNITS	SOURCE
Airspeed	m/s	Differential pressure sensor
Groundspeed	m/s	GPS
Altitude	m	Absolute pressure sensor and GPS
Body axes rates (Roll, Pitch, Yaw)	deg/s	Rate Gyroscopes (IMU)
Accelerations (X,Y,Z axes)	m/s <sup>2</sup>	Accelerometers (IMU)
Climb rate	m/s	Absolute pressure sensor
Heading	deg	GPS
Turn rate	deg/s	Yaw rate gyro (IMU)
Ultrasonic altitude	m	Ultrasonic
Ultrasonic climb rate	m/s	Ultrasonic
Servo positions	deg	Servos

### G.3.5.5 Data Analysis

Post data processing of all the required data in MATLAB is required to generate time plots of the logged flight data. The plots are generated by opening “PlotTelemetryData.m” and selecting the data required for plotting and post analysis. After the required plots are specified, run the m-file.

The following graphs are required over time:

- Airspeed, Climb rate and Altitude.
- Airspeed vs. Ground speed.
- Pressure altitude vs. GPS altitude.
- Accelerations (in body axes).
- Rotation rate (in body axes).
- Actual, autopilot and pilot servo commands.
- Lateral controller performance (Commanded vs. sensed Turn rate and heading).
- Longitudinal controller performance (Commanded vs. sensed airspeed, climb rate and altitude).
- 2D GPS position plot.
- Pressure altitude vs. ultrasonic altitude.
- Pressure climb rate vs. ultrasonic climb rate.

### G.3.5.6 Acceptance Criteria

The guidance controller must reach the following design specifications:

- Rise time of less than 8s.
- Overshoot of less than 20%.
- Zero steady state error.

The wind estimation procedure should have the following accuracies:

- Wind speed accuracy within 2m/s.
- Wind direction accuracy within 10deg.

## G.4 Test Schedule

The test schedule is dependant on the weather and turbulent level. Adjustments to the program will be made on a daily basis.

**Table G.6:** *Test Schedule*

NO	OBJ. NO	TEST DATE	FLIGHT TIME [H]	TEST DESCRIPTION
A	B	C	D	E
1	3.1	15 June 2006	0.25	Ground controller testing.
2	3.2	15 June 2006	0.25	General piloted flight with telemetry logging and runway flybys for ultrasonic testing.
3	3.3	15 June 2006	0.25	Longitudinal controller testing.
4	3.4	15 June 2006	0.25	Lateral controller testing.
5	3.5	15 June 2006	0.25	Autonomous navigation and wind estimation.

Total flight time = 1.25 [h]

## G.5 Test Hazard Analysis

Table G.7: Test Hazard Analysis Worksheet

TEST HAZARD ANALYSIS WORKSHEET						
TEST TITLE:						
MICRO AVIONICS AUTONOMOUS FLIGHT	SUBJECTIVE PROBABILITY OF OCCURRENCE					
	Hazard Category	frequent	probable	uncertain	remote	improbable
	catastrophic		HIGH			
AIRCRAFT/SYSTEM: Reliance 0.46 / Micro avionics system	critical					
	marginal			MEDIUM	X	
	negligible					LOW
HAZARD:	Damage to property by crashing aircraft.					
CAUSE:	Micro avionics system failure.					
EFFECT:	Uncontrollable unmanned aerial vehicle (UAV).					
MINIMISING PROCEDURES:	Work through preset checklist to ensure system is safe and set up correctly. In case of partial system failure, attempt to land safely or steer away to a clear area. Conduct flight test on a day of low activity on the airfield.					
EMERGENCY PROCEDURES:	Warn spectators of uncontrollable UAV when safety pilot loses control. Select a safe landing area.					
PREPARED BY	SIGNATURE		DATE			
Jan-Cor Roos						
Risk Level (after minimizing procedures taken into account)						
High	Medium	Low	X			