Modeling and Simulation of Autonomous Thermal Soaring with Horizon Simulation Framework

A Thesis

Presented to

The Faculty of California Polytechnic State University,
San Luis Obispo

In Partial Fulfillment

Of the Requirement for the Degree of
Master of Science in Aerospace Engineering

By

Zhenhua Morton Li

December 2010

©Copyright 2010
Zhenhua Morton Li
ALL RIGHTS RESERVED

Committee Membership

Title: Modeling and Simulation of Autonomous Thermal Soaring
with Horizon Simulation Framework
Author: Zhenhua Morton Li
Date Submitted: December 2010

Committee Chair: Dr. Eric A. Mehiel
Department Chair/ Assistant Professor
Department of Aerospace Engineering
California Polytechnic State University, San Luis Obispo

Committee Member: Dr. Daniel J. Biezad
Professor Emeritus
Department of Aerospace Engineering
California Polytechnic State University, San Luis Obispo

Committee Member: Dr. Robert A. McDonald
Assistant Professor
Department of Aerospace Engineering
California Polytechnic State University, San Luis Obispo

Committee Member: Dr. Alberto Jiménez
Lecturer
Department of Mathematics
California Polytechnic State University, San Luis Obispo

Abstract

Modeling and Simulation of Autonomous Thermal Soaring with Horizon Simulation Framework

Zhenhua Morton Li

A thermal is a column of warm rising air triggered by differential heating on the ground. In recent studies UAVs were programmed to exploit this free atmospheric energy from thermals to improve their range and endurance. Researchers had successfully flown UAVs autonomously with thermal soaring method. Most research involved some form of flight simulation. Improvements to the aircraft and thermal models for simulation purpose would enable researchers to better design their UAVs and explore any potential flaws in their designs. An aircraft simulation with a thermal environment was created in Horizon Simulation Framework, a modeling and verification framework that was developed by Cal Poly Space Technologies and Applied Research laboratory. The objective of this study is to enhance the fidelity of existing modeling and simulation methods on autonomous thermal soaring, and to advance and demonstrate the capabilities of Horizon Simulation Framework through such implementation. The geometry of a small remote controlled glider was used in this simulation. Aerodynamic prediction programs DATCOM+ and AVL were used to obtain stability and control derivatives for this glider. The induced roll effect caused by the asymmetric vertical velocity distribution of a thermal was included in the aerodynamic roll moment calculation. The autonomous guidance algorithm for the glider included a turn logic which would determine the correct turn direction for the glider when a thermal is detected. The thermal model developed in this thesis included the capabilities to vary the time dependent location, height, radius, and vertical velocity characteristics of naturally occurring thermals.

Acknowledgments

I want to first thank my thesis advisor Dr. Eric Mehiel for giving me this opportunity. Thank you for your patient and insightful explanation to every question I asked. I felt a step closer to finishing my research every time I talked to you. I am so glad that Dr. Dan Biezad introduced me to the world of flight simulation and control theory, and welcomed me to the family of the Flight Simulation Group. I am always grateful for your guidance and support. I like to thank Dr. Al Jimenez for your wonderful ideas and insightful feedbacks on my research. The Numerical Analysis you taught me is very useful. I would also like to thank Dr. Robert McDonald for taking his time to be in my thesis committee. Your comments helped shape my research, I really appreciate it. I want to thank Bill Galbraith for making such a wonderful program and helping me understanding DATCOM+. I also want to thank Dr. Natasha for proofreading my thesis, and corrected so many of my mistakes.

To all the faculty and staff in the Aerospace department, thank you for providing me such a great educational experience. I will treasure my time here at Cal Poly. To my friends who are always there for me, thank you. Looking back, you have done so much to shape the person who I am today. I especially want to thank Tanner Sims and Brian Fujizawa for thoroughly reviewing my paper. You are the ones that got me interested in Control theory, and I would not have made it through grad school without your help.

Finally, to my family who unconditionally support me, you are the reason that I make it this far in life, thank you for everything.

Table of Contents

List of Tables.....	ix
List of Figures	x
Nomenclature	xii
Chapter 1 Introduction.....	1
1.1 Soaring Flights	1
1.1.1 Gust Soaring	2
1.1.2 Dynamic Soaring	3
1.1.3 Static Soaring	4
1.1.4 Previous and Related Work	6
1.2 Modeling and Simulation of Flight	8
1.2.1 Coordinate Frames	9
1.2.2 Euler Angles and Quaternion	10
1.2.3 Equations of Motion	12
1.2.4 Aerodynamic Prediction Methods	12
1.2.5 Gravity.....	13
1.2.6 Thermal Model.....	13
1.2.7 Guidance and Control.....	14
1.2.8 Standard Atmosphere	15
1.2.9 Previous and Related Work	15
1.3 Horizon Simulation Framework	16
1.3.1 Development Principles.....	17
1.3.2 Position Types in Horizon Simulation Framework	17
1.3.3 Previous and Related Work	18
1.4 Objective	19
1.5 Organization of Thesis	20
Chapter 2 Aircraft Models.....	22
2.1 Quaternion Method for Aircraft Attitude	22
2.2 Equations of Motion	24
2.3 Aerodynamic Forces and Moments.....	24
2.4 Stability and Control Derivatives	25
2.4.1 Glider Geometry.....	26
2.4.2 DATCOM+	27
2.4.3 Athena Vortex-Lattice	29
2.4.4 Comparison and Validation.....	30
2.5 Thermal Induced Roll.....	32
2.5.1 Moment Arm	33
2.5.2 Angle of Attack and Freestream Velocity	34

2.5.3 Induced Roll Moment	34
2.6 Gravitational Model	35
2.7 Atmospheric Model	35
Chapter 3 Thermal Model	38
3.1 Allen's Thermal Model	38
3.2 Time Dependent Parameters.....	39
3.3 Drift 40	
3.4 Improved Thermal Model	41
Chapter 4 Autopilot Design.....	42
4.1 Linear Control Law Design	42
4.1.1 Small Perturbation Theory	42
4.1.2 Longitudinal Analysis	43
4.1.3 Lateral Analysis.....	47
4.2 Nonlinear Control Law Design	52
4.2.1 Longitudinal Analysis	52
4.2.2 Lateral Analysis.....	54
Chapter 5 Thermal Soaring Guidance	58
5.1 Mode Logic	58
5.2 Turn Logic.....	59
5.3 Thermal Identification.....	61
5.4 Soaring Controller	61
Chapter 6 Simulation Implementation.....	63
6.1 Interpolation	63
6.2 Frequency to state space domain	67
6.2.1 Example.....	67
6.2.2 Derivative approximation	68
6.2.3 Verification.....	68
Chapter 7 Simulation Result.....	70
7.1 Turn Logic.....	70
7.2 Thermal Soaring Race	72
7.3 Shape, Strength, and Drift	74
Chapter 8 Conclusion	76
Bibliography 77	
Appendix A Stability and Control Derivatives Definition	79
Appendix B Glider Geometry Measurements	81
Appendix C DATCOM+ Input and Output.....	83
Appendix D AVL Input and Output	87
Appendix E Updraft Model Parameter Inputs	92
Appendix F Pseudo Codes for Interpolation Methods	93

Appendix G Transfer Function Routine	97
--	----

List of Tables

Table 1.1	Origin and axes definition of the coordinate frames.....	9
Table 2.1	Modeled glider's general information	27
Table 2.2	ARDC 1959 standard atmosphere properties	35
Table 4.1	Glider natural longitudinal response and handling qualities	43
Table 4.2	Pitch rate feedback gain effects on time domain analysis	46
Table 4.3	PID compensator designs for different rate feedback gains	47
Table 4.4	Glider natural lateral response and handling qualities.....	48
Table 4.5	Compensator designs for roll angle to aileron loop.....	49
Table 4.6	Compensator design for coordinate turn	51
Table 4.7	Yaw rate PID controllers performance.....	51
Table 4.8	Heading controllers comparison of forward path proportional gain.....	52
Table 4.9	Nonlinear simulation controller performance comparison	52
Table 6.1	Drag coefficient as function of angle of attack.....	63
Table 6.2	Change in drag coefficient due to angle of attack and elevator deflection	64
Table 6.3	Computational performance of the interpolation methods	66
Table 7.1	Summary of the soaring flight race	73

List of Figures

Figure 1.1 Dryden Wind Turbulence model.....	2
Figure 1.2 Shear flow on the leeward side of a ridge ²⁰	3
Figure 1.3 Thermal Model ⁴	4
Figure 1.4 Aircraft model diagram.....	9
Figure 1.5 Reference frames (working on a better one...)	10
Figure 1.6 Aircraft Euler angles ⁵ (working on another one...)	11
Figure 1.7 Six degrees-of-freedom motion of an aircraft	12
Figure 1.8 Computational updraft model	14
Figure 2.1 Aircraft attitude calculation implementation using Quaternions.....	22
Figure 2.2 HYPER-DL 1.5M sailplane ³⁶	26
Figure 2.3 Glider manufacturer's plan drawing with partially digitized body outline	27
Figure 2.4 Portion of the example input file for DATCOM+.....	28
Figure 2.5 Graphical representation of the glider geometry in DATCOM+	28
Figure 2.6 Graphical representation of the glider geometry in AVL.....	29
Figure 2.7 Datcom and AVL comparison of lift and pitch moment S&C derivatives.....	31
Figure 2.8 Datcom and AVL comparison of drag and roll moment coefficients	31
Figure 2.9 Datcom and AVL comparison of side force coefficients	31
Figure 2.10 Datcom and AVL comparison of yaw moment coefficients	32
Figure 2.11 Comparison between published and calculated temperature as altitude is increased.....	37
Figure 3.1 Implementation of the time dependent variables	39
Figure 3.2 Example of the time dependent height gain interpolation table	39
Figure 3.3 Implementation of drift	40
Figure 3.4 Example of time dependent drift values in y direction.....	40
Figure 3.5 Time lapse images of three thermals.....	41
Figure 4.1 Load-sensitivity factor flying qualities	43
Figure 4.2 Longitudinal control loop.....	44
Figure 4.3 Root-contour of different pitch rate feedback gains, Phugoid mode.....	45
Figure 4.4 Bode-contour plot of different pitch rate feedback gains	45
Figure 4.5 Step response of pitch angle controllers.....	47
Figure 4.6 Roll angle to aileron and body yaw rate to rudder root loci.....	48
Figure 4.7 Roll angle to aileron and body yaw rate to rudder unity feedback step responses	48
Figure 4.8 Roll attitude angle controllers comparison.....	49
Figure 4.9 Lateral-directional controller block diagram.....	50
Figure 4.10 Step response of various rate feedback gains for β to rudder	50
Figure 4.11 Yaw rate controllers	51

Figure 4.12 Step response in non-linear simulation	53
Figure 4.13 Elevator control activity in non-linear simulation.....	53
Figure 4.14 Step response with Dryden Turbulence Model in non-linear simulation	54
Figure 4.15 Elevator control activity with Dryden Turbulence Model	54
Figure 4.16 Linear and nonlinear roll angle controller step responses	55
Figure 4.17 Linear and nonlinear yaw rate step response comparison	55
Figure 4.18 Nonlinear simulation lateral-directional control activity	56
Figure 4.19 Heading angle controllers comparison.....	56
Figure 5.1 Updraft velocities felt by glider at CG, left wing, and right wing locations	59
Figure 5.2 Glider flying through a thermal without autopilot	60
Figure 5.3 Glider flying through a thermal with autopilot on	61
Figure 5.4 Soaring controller.....	62
Figure 6.1 Interpolation of the one-dimension drag coefficient table	65
Figure 6.2 Interpolation of the two-dimensional drag coefficient table	65
Figure 6.3 Transfer function routine verification	69
Figure 7.1 Clockwise turn demonstration flight.....	70
Figure 7.2 Counterclockwise turn demonstration flight	71
Figure 7.3 Three-dimensional view of the clockwise turn demonstration flight.....	71
Figure 7.4 Twenty set of thermals randomly positioned in the race course	72
Figure 7.5 Flight #10 of thermal soaring race	74
Figure 7.6 Flight #15 of thermal soaring race	74
Figure 7.7 Demonstration flight to show thermal's time dependent properties.....	75

Nomenclature

UAV	Uninhabited aerial vehicle
M&S	Modeling and Simulation
HSF	Horizon Simulation Framework
CPSTAR	Cal Poly Space Technologies and Applied Research
EOM	Equations-of-motion
NED	North East Down
6DOF	Six degrees-of-freedom
AVL	Athena Vortex Lattice
CFD	Computational fluid dynamic
LLA	Latitude, longitude, altitude
ECI	Earth center inertia
DCM	Direction cosine matrix
S&C	Stability and control
SAS	Stability augmentation system
PID	Proportional-integral-derivative
PI	Proportional-integral
ILC	Integral lag compensator

Chapter 1 Introduction

Birds have developed soaring techniques for conserving energy, extending range, and lengthening their flight time by taking advantage of atmospheric phenomena such as gust, shear flow, and thermals. This study focuses on the modeling of a thermal - a column of upward moving warm air. Glider pilots have learned to use thermals to improve their range and endurance. In the recent decade, research has shown that uninhabited aerial vehicles (UAVs) can utilize autonomous soaring techniques to help reduce the vehicle's fuel consumption, extend range, and improve endurance. Researchers have also relied on modeling and simulation (M&S) of soaring flight to test, verify, and fine-tune the guidance and control algorithms of UAVs systems. A high fidelity model of the aircraft and the soaring environment would allow researchers to better prepare for unforeseen problems in their design and allow them to experiment using more reliable data.

Horizon Simulation Framework (HSF), a system-level modeling and verification tool developed by Cal Poly Space Technologies and Applied Research (CPSTAR) laboratory, was used to implement the programming aspect of this study. Previous HSF projects have demonstrated many designed functions and improved the utility of the framework. The soaring simulation developed for this study would demonstrate one of the HSF defined position type trajectory features. Furthermore, additional functionality was added to the HSF library through the implementation of this simulation.

In this chapter, an introduction to thermal and other soaring methods are first introduced, followed by a description of aircraft simulation methods and an autonomous thermal soaring algorithm. The programming framework HSF and its development are then presented. In the context of the research topics and tools introduced, the objective of this thesis is finally discussed.

1.1 Soaring Flights

In the autonomous soaring field of study, three different soaring methods are actively researched - gust, dynamic, and static soaring.¹ Gust soaring extracts energy from a turbulent condition to improve flight performance, utilized by vultures and petrels to exhibit better flight performance in turbulence condition compared to their normal gliding.² Dynamic soaring extracts energy from the shear flow in the atmospheric

boundary layer; albatrosses are known to trail ships in the open sea for days, by using dynamic soaring, almost without flapping their wings.³ Finally, static soaring utilizes upward moving air mass (updraft) to sustain flight, exemplified by condors and vultures that have used updraft mostly in the form of thermals to migrate and forage.⁴ Information about turbulence, shear flow, and updraft is given along with their soaring methods in this section. Even though the focus of this study is thermal soaring, other soaring methods are considered here for completeness.

1.1.1 Gust Soaring

A gust is a strong sudden burst of wind. When the profile is continuous, the gust structure is generally referred to as turbulence.⁵ Turbulence is also observed in between thermals, whereby the duration of gust is usually only a few seconds. The motion of a gust is unpredictable, but a statistically representative (stochastic) model can be used for simulation analysis. Figure 1.1 shows an example of Dryden Wind Turbulence model.

Figure 1.1 Dryden Wind Turbulence model

Gust soaring is becoming an increasingly active research area. Implementation of such a soaring method on an aircraft is difficult due to the short duration and randomness of gusts. A necessary fast response from the aircraft is hard to achieve; birds, on the other hand, are naturally equipped with such agile reactions. One method, suggested by Patel,² is to align the lift vector of an aircraft to the direction of the gust during flight, enabling the aircraft to utilize gust energy. However, no successful autonomous flight is currently known to successfully employ this method.

1.1.2 Dynamic Soaring

Shear flow is defined as horizontal wind in the boundary layer that has a velocity gradient profile due to the frictional force from the surface. Wind closer to the surface is slowed down by friction, thus velocity increases with altitude. Shear flow is common in the open sea, and is successfully used by albatross to sustain flight. In contrast, on land, shear flow that is suitable for dynamic soaring is restricted to mountain ridges that satisfy a certain condition – a specific strength and profile of the moving air over the mountain ridge capable of creating a well defined boundary, as illustrated in Figure 1.2 (adapted from Sachs⁶).

Figure 1.2 Shear flow on the leeward side of a ridge⁶

Dynamic soaring methods extract energy from shear flow by flying in a pattern of diving downwind, turning, and climbing upwind, then turning and diving downwind again.³ When diving downwind, aircraft can increase their ground speed by combining their constant airspeed with the wind velocity. Kinetic energy is gained this way while the energy loss due to drag remains minimal. In the turning segment, the previously gained kinetic energy can be used to travel to the intended heading, whereby energy loss in this portion is due to aerodynamic drag. As the aircraft subsequently climbs into the wind gradient, it trades ground speed for altitude while maintaining airspeed again. The cycle is then repeated. This soaring technique requires a highly agile flight maneuver, as the aircraft is in close proximity to the ground, making this flight technique very risky. No successful applications of autonomous dynamic soaring flight currently exist; however, enthusiasts have successfully used radio controlled airplanes to perform dynamic soaring.¹

1.1.3 Static Soaring

An updraft is a vertical current of rising air. There are three forms of updraft - orographic lift, mountain wave, and thermal. Orographic lift is the result of horizontal wind deflected upward by discontinuities on the Earth's surface. If temperature and wind condition are met, on rare occasions, mountain wave can be formed from orographic lift. Thermal updraft is a parcel of buoyant rising air triggered by differential heated-surface. Among the three, thermal is the most abundant and widely used by birds and glider pilots.⁷ In this section, only the structure of thermal and the soaring method used by glider pilots are further described in detail. The terms updraft and thermal are used interchangeably in this thesis.

The formation process of a thermal is explained here. As the Earth absorbs energy from our sun, the surface dissipates this energy to the parcel of air that is directly above. When this parcel of air is heated, it starts to expand, becomes less dense, and rises to form a column of warm air; as a result, temperature within the thermal decreases. At a certain altitude, the temperature inside the thermal would reach an equilibrium point with its surroundings and will lose its driving force to rise; marking the ceiling for a thermal. The immediate air mass below would continue to rise and push the air above out and down to the side, similar to a glass of soda overflowing. The air that was pushed down is called downdraft or entrainment, and is often encountered by glider pilots as they first enter a thermal. The path of the airflow and the generalized shape of a thermal are shown in Figure 1.3.

Figure 1.3 Thermal Model⁷

Some qualitative characteristics of a thermal are given here. Thermal is wider at the top and narrower at the root because it expands as it rises. The vertical velocity distribution across a thermal resembles a normal distribution with the peak velocity at the center; this velocity profile matches what glider pilots experience during their soaring flight.⁴ Thermals have irregular sizes and shapes which change constantly due to solar and wind conditions. Quantitatively, thermals have diameter that can range up to 1,000 meters wide; height varying from 500 to 3700 meters; lasting from 5 to 30 minutes; and are characterized by vertical air velocities of 1 to 4 m/s.⁴

The strength of a thermal is primary governed by the amount of available sunlight. Secondary factors are time of day, atmosphere conditions, geographic location, and ground surface properties. The movement and longevity of a thermal is affected by the presence of prevailing wind. Thermals tend to tilt in a steady wind, and can be distorted or broken apart if the wind conditions are strong. Naturally occurring thermals also tend to merge to form a bigger thermal as they get closer at high altitudes.⁹

Soaring methods given here are based on glider pilots' experiences from the *Glider Flying Handbook*.⁷ The procedures of thermal soaring can be divided into four steps: locating a thermal, entering, centering, and leaving a thermal.

Most of the techniques used by glider pilots to locate thermals rely on visual observation. For example, often a formation of cumulus clouds signals that a thermal is right underneath. If the air inside a thermal contains enough moisture, and the thermal reaches high enough altitude, cumulus cloud would form. Another indicator of a thermal is the presence of other circling gliders and birds. Haze dome caused by particulates that have been carried aloft by thermal is another visual indication. On the other hand, without relying on visual cues to search for thermals, glider pilots often visit locations where they encountered thermals repeatedly; these reoccurring thermals are given a name – house thermals.⁷

When a glider is close enough to a thermal, the first indication of the thermal's presence is, oddly enough, increased sink caused by the entrainment of a thermal. Next, a positive G-force should be felt by the glider; this is caused by the increasing updraft velocity as the glider moves closer to the center of a thermal. Furthermore, due to the stronger updraft velocity toward the center, the wing closer to the thermal center would experience a stronger lift force than the outer wing. Under these conditions, a glider would

roll away by a thermal, if no action is taken by the glider pilot. Glider pilots usually start turning into a thermal either at the strongest updraft location or just right pass that point.

After a thermal is located, a glider pilot needs to determine which way to turn in order to enter the thermal properly. This can be determined by the initial roll motion of the glider. If a glider is banked to the right upon entering the thermal, the glider pilot should make a left turn to enter. Once a glider enters the thermal, pilot should straighten or tighten the turn toward the stronger lift direction. The goal of centering a thermal is to determine the location of the best lift and move the glider into it for the most consistent climb.⁷ A good set of rules-of-thumb used by glider pilots is from Reichmann:

As climb improves, flatten the circle (~15-20° bank angle)

As climb deteriorates, steepen the circle (~50° bank angle)

If climb remains constant, keep constant bank (~15-20° bank angle)

The criterion for leaving a thermal depends on whether the goal is to maximize altitude or shorten cross country flight time. If the goal is to maximize altitude, the glider should stay as long as the climb rate is not negative. In contrast, if the goal is to shorten cross country flight time, the glider should leave as soon as lift is weakened.

1.1.4 Previous and Related Work

After the success of the first powered flight by the Wright brothers in 1903, interests in soaring techniques to sustain flight were shifted away from the early aviation pioneers. Much of the early soaring method research was continued on by a small group of aeroecologists, who applied aerodynamic principles to avian ecology.³ Presently, many of the existing soaring techniques are based on the actual flight strategy of migrating birds.

In 1998, Wharington first proposed the use of autonomous thermal soaring to extend the range of an UAV. In his research, he used a neural-based thermal locator algorithm to estimate a thermal's location, strength, and radius. He also used reinforce-learning algorithm for his soaring guidance system. Much of the subsequent work done by other researchers on autonomous thermal soaring can be traced back to Wharington's initial designs.

The progress of autonomous thermal soaring made a huge leap with Michael Allen's research at NASA Dryden Flight Research Center. Allen had first developed a three degrees-of-freedom flight model along with a basic updraft model to assess the annual performance benefits from updraft. Results from his

simulation indicate that UAV can gain significant endurance improvement using thermal soaring.⁸ In 2006, Allen presented a high fidelity updraft model that more accurately represented the naturally occurring thermal.⁹ In 2007, Allen published detail implementation of a guidance algorithm on autonomous thermal soaring along with successful autonomous soaring flight test results. Instead of using a neural-based algorithm from Wharington's research, Allen implemented a less computationally intensive iterative fit thermal locator algorithm developed by Staffan Kjerrström.¹⁰

Another breakthrough came from Daniel Edwards¹¹ of North Carolina State University. In 2008, Edwards had successfully sustained 1.5 hours of autonomous thermal soaring flight. Later that year he had unofficially beaten the world record maximum range for radio controlled glider with 60.4 miles distance traveled in about 4 hours of flight. Edwards had chosen to focus his effort on robustness and flight testing during the development of his UAV and bypassed the option of a high fidelity simulation. Edwards used Allen's work as a starting point, but he chose Wharington's neural-based method instead of Allen's iterative fit algorithm. Allen did not use the neural-based method because it was too computationally intensive for the onboard computer to handle. Edward solved this problem by using ground based processing and relayed the control and telemetry data wirelessly to his UAV. Edwards also suggested that the iterative fit algorithm was limited to thermal center prediction within the measurements of the UAV's flight path; whereas the neural-based algorithm does not have such restriction and can make better thermal location prediction.

With the successful development and flight test from Allen and Edwards, other aspects of thermal soaring are also being investigated. In 2007, Kahveci¹² from University of Southern California investigated the benefits of using trajectory optimization for thermal soaring. In 2009, Andersson and other researchers at Naval Postgraduate School showed that cooperating UAVs can increase the likelihood of finding thermals in simulation.^{13,14,15} Researchers at Naval Postgraduate School planned to use flight test results to verify their simulation finding; but no published data was found at the time this thesis was written. The soaring guidance algorithm used by Andersson was based on Allen's work with modification. The most recent research found in this field came in 2010, from Matthew Hazard¹⁶ of North Carolina State University. His research was based on both Allen and Edwards' work, with focus on improving the thermal

center locator algorithm. His solution was to use Unscented Kalman Filter, a nonlinear recursive estimator. Simulation of his method showed good convergence and phase lag performance.

1.2 Modeling and Simulation of Flight

Fighter, bomber, and surveillance aircraft played important roles in World War I. The military realized that they needed to find a way to train new pilots faster, safer, and more efficiently. The use of a flight simulator suited those needs. With the advancement of analog computers in the 1940s and digital computers in the 1960s, flight simulators became standard training devices for the aircraft industry. As the designs of new aircrafts became more advanced and complex, computer flight simulations were used by designers in the early stages of development to safely test and verify their ideas to prevent costly modification later on. Flight simulation encompasses many different systems in order to replicate the behavior of an aircraft in flight; visual display, motion cue, computational model of an aircraft are some examples. The present study focuses on the computational models that represent an aircraft system and the thermal environment. A small glider is used as the aircraft platform for this simulation.

In this section the tools used to construct the flight simulation are introduced. Three coordinate frames represent the vector information of different models. Quaternions represent aircraft attitude information. Equations-of-motion (EOM) are a system of ordinary differential equations that describes the motion of an aircraft with different forces and moments acting upon the airframe. Some of the major forces and moments that are usually included in a flight simulation come from aerodynamics, propulsion, landing gear, gravity, and wind models. An aerodynamic model predicts the force and moment that an aircraft's geometric shape generates as it moves through the airstream. Propulsion and undercarriage models are omitted in this work, as the modeled aircraft is a small glider and does not have such features. Gravitational model of Earth estimates the gravitational force applied to the aircraft as a function of altitude. In this simulation thermals were used as the wind model. An autonomous guidance and control system was used to fly this UAV; since no pilot was available to fulfill this task. Earth's atmospheric model provides pressure, temperature, and density information to the other systems. Figure 1.4 shows the interaction of these systems.

Figure 1.4 Aircraft model diagram

1.2.1 Coordinate Frames

A flat Earth model was used in this simulation. Geometric, rotational, and magnetic properties of Earth were neglected. These assumptions were acceptable since the distance traveled in this simulation was small. Three coordinate frames were used in this study – body, stability, and local ground reference frame. Body frame is a common frame for forces and moments of the aerodynamics, propulsion, and landing gear models to be expressed in. Stability frame was used to express the stability and control derivatives from aerodynamic model calculations. Local ground reference frame, or just local frame, was used to express the position of the glider and was treated as the inertial reference frame. Origin and axes definition of the three frames are shown in Table 1.1 and Figure 1.5.

Table 1.1 Origin and axes definition of the coordinate frames

Frame	Origin	Definition
Body	Aircraft CG	i_b points out of aircraft nose, j_b points right wing, k_b points down
Stability	Aircraft CG	i_{sb} is the projected vector on the xz plane by the flight path vector, j_{sb} points right wing, k_{sb} points down
Local	Fixed point on Earth	i_{local} points north, j_{local} points east, k_{local} points down

k axis of all three frames is the vector cross product of i axis crosses j axis. The local frame is also referred to as the north-east-down frame (NED). i_a is the direction that the glider is flying.

Figure 1.5 Reference frames

1.2.2 Euler Angles and Quaternion

The orientation of an aircraft can be described by a sequence of rotational angles called the Euler angles. The order of the rotation sequence from the local frame to the body frame, is important with the proper sequence being yaw-pitch-roll. In an actual aircraft, the pilot can obtain the Euler angles either by looking outside of the aircraft or from the attitude indicator and the heading indicator. Figure 1.6 illustrates the rotation sequence and the Euler angles.

Figure 1.6 Using yaw-pitch-roll sequence to obtain aircraft Euler angles

However, the mathematical calculation of the Euler angles would encounter a singularity solution when the pitch angle is 90 degrees. This also happens to the rotational rate measurement device of an actual aircraft; such phenomenon is called gimbal lock. In simulation, Quaternion parameters can be used instead, to keep track of an aircraft's attitude information and to solve for the corresponding Euler angles without running into the gimbal lock problem.

1.2.3 Equations of Motion

If the forces and moments acted upon an aircraft by its subsystems and the environment were known, the aircraft motion as a function of time progressed can be represented by the equations-of-motion (EOM). The EOM used in this simulation accounted for motion with six degrees-of-freedom (6DOF), which are the three translational movements in the x , y , and z direction and the three rotational motions of roll, pitch, and yaw; as illustrated in Figure 1.7. The EOM in this study also included the nonlinear and coupled effects. EOM can be described as nonlinear if the equations account for the nonlinear combination of aircraft states. For example, forward acceleration calculation in nonlinear EOM is depended on the combination of yaw rate and the side velocity. Furthermore, EOM can be described as coupled if the motion in the lateral-directional axis affects the longitudinal axis, and vice versa.

Figure 1.7 Six degrees-of-freedom motion of an aircraft¹⁷

1.2.4 Aerodynamic Prediction Methods

There are many different methods that can be used to predict the aerodynamics of an aircraft, including manual calculation of aerodynamic theories, semi-empirical method such as Digital Datcom and DATCOM+, computational numerical method such as AVL and Fluent, wind tunnel results, and system

identification method from flight test data. DATCOM+ and AVL were used to complement each other and provide aerodynamic stability and control derivatives for the glider model used in this study.

1.2.5 Gravity

The gravitational model used in this simulation is based on Newton's law of universal gravitation. The force exerted on the aircraft was calculated as a function of the mass of Earth, the mass of the glider, and the distance between the Earth's center and the glider. This simple model assumes that the Earth is a perfectly uniform sphere and does not account for rotational effects of Earth.

1.2.6 Thermal Model

The thermal model used in this simulation is based on Allen's research.⁹ Modification was made to the original model to include additional characteristics of naturally occurring thermals. Figure 1.8 shows a graphical representation of the thermal model used in this simulation; MATLAB's built-in function *contourslice* was used to create this figure. The color of the contour lines that form the thermal model represents the updraft vertical velocity. The velocity is stronger at the center of a thermal and weaker at the outer radius, which resembles a normal distribution described earlier. Downdraft is found at the top portion of the thermal model, which is represented by blue contour lines.

Figure 1.8 Computational updraft model

Some additional characteristics of a thermal were incorporated in this simulation, such as height, radius and vertical velocity of a thermal are time dependent variables; they are affected by the amount of sunlight, time of day, atmospheric condition, geographic location, and ground surface properties. Also the position of a thermal tends to drift in the prevailing wind direction.

1.2.7 Guidance and Control

The guidance and control of this glider are comprised of two parts – the autopilot and autonomous soaring guidance system. The autopilot was used to stabilize the glider and accepted maneuver command from either the operator or the autonomous guidance system. Classical control theory was used to design the autopilot. Sims' work¹⁸ was used as a reference during this designing process, whereas the autonomous guidance system was based on Allen's research. The guidance system used in this simulation is consists of four main parts – the mode logic, turn logic, thermal identification, and soaring controller. Mode logic algorithm decides whether a glider should engage or disengage to/from autonomous thermal soaring mode. Turn logic determines which side a thermal relative to the glider is, and controls the turn direction if the

soaring mode is engaged. Thermal identification algorithm estimates a thermal's location, height, radius, and vertical velocity. The soaring controller calculates the optimal turn rate that a glider should use to center a thermal based on a heuristic approach.

1.2.8 Standard Atmosphere

The first practical table for a standard atmosphere suitable for aeronautical research use was compiled by Walter S. Diehl in 1925. This model treats the air mass as a homogenous mixture of gas that rotates with Earth. Measurements used for this atmospheric model were averaged over a full year at different regions around the world. Many variants of the original model had been created. All subsequent versions of the 1925 standard atmospheric model have good agreement with the original data up to 20 kilometers altitude. This simulation used the ARDC 1959 standard Atmosphere model.^{19,20}

1.2.9 Previous and Related Work

Since constructing a simulation-architecture and modeling the aerodynamics of a glider are a major aspect of this study, related research are mentioned here.

Turevskiy²¹ proposed an aircraft design process that uses both free and commercial off-the-shelf programs to construct flight simulation and to design flight control system. Numerous prebuilt simulation system models from MATLAB/Simulink's Aerospace Blockset were used to construct the aircraft model in his research. Digital Datcom was used to estimate the aerodynamic characteristics of a light weighted monoplane.

Leong²² developed a 6DOF, nonlinear, coupled UAV simulation and validated his result using flight test data. Roskam's AAA and Drela's AVL aerodynamic prediction programs were used in his study. Flight test results suggested that the nonlinear 6DOF EOM provides more accurate aircraft response and coupling effect than the linear simulation.

Ghoreyshi²⁴ automated a Computer Fluid Dynamic (CFD) process to generate stability and control derivative tables to model the aerodynamics of an aircraft. Reynolds-averaged Navier-Stokes method (RANS) was used for the CFD calculation. When CFD result, DATCOM, Tornado and wind tunnel test data were compared, a good match was found between the CFD results and the wind tunnel measurements.

Ghoreyshi's work provided a highly accurate and robust way of computing the aircraft aerodynamic characteristics without the costly wind tunnel testing.

Foster²⁵ proposed a new method of determining the longitudinal handling qualities for small UAVs. His research suggested that small UAV has a higher natural frequency on all modes of oscillation. Based on this finding, he developed a new load-factor-sensitivity flying qualities criterion for UAVs. His work was based on the MIL-F-8785C military specification handling qualities. The new handling qualities suggested from his work were also used to evaluate the modeled glider of this simulation.

1.3 Horizon Simulation Framework

Horizon Simulation Framework (HSF) is the underlying programming environment used to create the flight simulation in this thesis. HSF is a C++ based modeling and simulation framework, whereby software application, as well as operation of the simulation is implemented in Microsoft Visual Studio.

HSF was developed by the Cal Poly Space Technologies and Applied Research (CPSTAR) laboratory in the summer of 2006 with support from Cutting Edge Communication, LLC.²⁶ The initial thrust of the program was toward the application of systems engineering principals as applied to space systems. However, the HSF design supports modeling and simulation of any system, such as ground troops, blimps, or UAVs. HSF was built as a modeling and simulation framework used to perform system level requirement verification and validation that designers could use throughout the design process. More information on Horizon Simulation Framework can be found in the HSF reference manual.²⁷

HSF uses a standard test scenario to verify the integrity of each of its new stable releases, such as the Aeolus test case scenario of an extreme-weather imaging satellite mission. The framework provides the necessary modeling and calculation tools which allowed the designer of HSF to model the subsystems of an extreme-weather imaging satellite. The Aeolus satellite system was simulated over three revolutions around the Earth, and HSF's exhaustive-schedulers were used to select the optimal schedule that would collect the maximum number of images. Through the simulation of such scenario, possible subsystems constraints or failures can be identified and allow the designer to make appropriate changes.

The main advantage of HSF is its computational performance. Since the simulation process in HSF is implemented through compiled codes, it can run faster than if higher level programming languages or tools,

such as MATLAB or Visual Basic, were used. One possible drawback of HSF is the steep learning curve required from the users new to C++ programming. Since HSF is written in C++, it is also computer platform independent. HSF is an open source research development project and its important benefit is the free access to its source code for researchers.

Development and improvement of HSF have been the work of many projects and graduate theses. The continued improvement on HSF makes it more competitive to similar commercial software and allows students to research and contribute to the development of such simulation tool. The work here contributes to the growth of HSF.

1.3.1 Development Principles

HSF was developed with three principles in mind – modularity, flexibility, and utility. Original developers designed the interaction between the scheduling algorithm and the system model, as well as the subsystems interaction within the system model to be modular. The interactions are rigorously defined such that they can only interact through a very limited set of protocols. This allows the scheduling algorithm or the subsystems to be replaced easily. Consequently, this promotes the users to build their subsystems in a modular way, so that can be included in the HSF library and be reused later.²⁶

The subsystem modeling in HSF is flexible. Since no internal limits are placed on how a subsystem is modeled, only the interaction between subsystems is specified; the HSF user has a full freedom in their modeling. Hence, the simulation of a glider soaring within a thermal is possible in HSF.

The original HSF developers aimed for the framework to be useful and reusable. Thus, a library of tools is included in the framework to ease the workload for the programmers. Different integration methods, coordinate-frame-transformations, matrix, Quaternion, XML file format parser, and more are included in the HSF library. The library is growing, as new projects contribute to the development and testing of the framework.

1.3.2 Position Types in Horizon Simulation Framework

The classification of the type of positions that HSF uses to associate with each system is of particular importance because the results of this thesis demonstrate the dynamic-ECI position type of HSF. In HSF, the position of the system can be represented by latitude, longitude, and altitude (LLA) coordinates or Earth center inertia (ECI) coordinates. The trajectory of the system is further specified as static, predetermined, or

dynamic. Static object does not move or is nearly stationary for the duration of the simulation; examples are buildings on Earth and distant stars. Predetermined objects' trajectory can be calculated before the start of simulation, such as for satellites in orbits and trains on rails. Dynamic type systems describe objects that can vary their trajectory in response to events or the environment they reside in, as exemplified by the thermal soaring simulation. The combinations of the two coordinate representations and the three trajectory types yield the six position types that are available in HSF.

1.3.3 Previous and Related Work

Some of the past and current projects will be discussed here to provide perspective on HSF development progress. As mentioned, each new version of HSF is validated through the test case scenario of the Aeolus mission to verify the integrity of the new development. In the latest stable release of the framework, the number of imaging satellites had increased from one to three simultaneous satellites in operation, demonstrating the multi-asset capability of the framework.

Cory O'Connor²⁶ had meticulously defined the interaction between various modules of the framework and, thus, determined the underlying structure of HSF. In his study, Derek Siebel implemented a new scheduling algorithm for HSF, showcasing the modularity principle.

One of the previous projects resulted in a simulation of a blimp interacting with wind field information obtained from online weather database. This simulation was the first to demonstrate the Dynamic-LLA position type. However, this simulation did not fully utilize the HSF library and did not adhere to the modularity principle. As result the blimp system was not able to interact with other systems in HSF.

More recently, Brian Patrick developed a graphical user interface for HSF using C# programming language. His work did not modify the core codes of HSF, but would make using HSF more convenient for new users. Brian Butler is currently working on creating a scripting interface for HSF. The scripting feature would be similar to the scripting capability of MATLAB and would create a higher level user interface that would allow using HSF without the need to learn C++. The learning curve for the scripting language would be easier.

1.4 Objective

The motivation for this study came from the opportunity to work with CPSTAR in developing an aircraft simulation that would dynamically reacts to the environment. The inspiration for research in the field of thermal soaring originated from the author's daily sighting of turkey vultures soaring in the sky of San Luis Obispo. Moreover, the idea of modeling a glider and a thermal environment came from the ongoing research in the field of the autonomous soaring, especially from Michael Allen's work. Many different aspects of thermal soaring have been studied by other researchers, such as thermal locator algorithm, soaring guidance algorithm, trajectories optimization, and cooperating UAVs. One common aspect of previous studies is that they all involve some form of computational flight simulation. Thus, improvement to the simulation process can possibly help the researchers better design their aircrafts.

The construction of such soaring simulations can be broken down into three major components: aircraft model, thermal model, and the aircraft guidance and control. Improvement on the aircraft model can be obtained from a higher fidelity aerodynamic prediction method, which accounts for the interaction between the aircraft and the thermal model. For example, when a glider flies through a thermal without any active control, the glider should be rolled away by the nonuniform lift distribution within a thermal. Thermal models can be improved by incorporating more characteristics of naturally occurring thermals. The guidance and control algorithm should then take advantage of the higher fidelity aircraft and thermal models to develop better soaring algorithms.

In order to positively contribute to the development of HSF, the soaring flight simulation of this study should adhere to the design principles of the framework. The aircraft system should follow the modularity principle and be constructed in such a way that it would interact with the framework. By successfully simulating the soaring flight of a glider, the flexibility principle and the Dynamic-ECI position type of HSF will be demonstrated. Supporting algorithms used in the flight simulation should be written in ways that they could be added to the HSF library and be reused later.

The objective of this study is twofold; first is to enhance the fidelity of existing modeling and simulation methods on autonomous thermal soaring; second is to advance and demonstrate the capabilities of Horizon Simulation Framework through the implementation and exercise of this glider thermal soaring simulation.

1.5 Organization of Thesis

The aircraft model, thermal model, and the guidance and control designs are introduced in Chapter 3 – 5. Additional algorithms that were used to help implement the simulation in HSF are discussed in Chapter 6 . Autonomous thermal soaring flight simulation results are shown in Chapter 7 . Finally conclusion is given in Chapter 8 .

Chapter 2 describes the tools and systems used to build the aircraft simulation. The method of using Quaternion parameters to represent the aircraft attitude is introduced first. The EOM that govern the dynamics of the aircraft are given next. Aerodynamic models are explained in detail with the emphasis on DATCOM+ and AVL aerodynamic prediction programs. Next, the induced roll interaction of a glider and a thermal is derived, with gravity and atmospheric models covered last.

Chapter 3 briefly introduces the thermal model from Allen’s research. Improvements to the thermal model are covered next. In the present study, additional thermal characteristics such as time dependent height, radius, vertical velocity, and drift were incorporated to the thermal model through the use of lookup tables. Each parameter was stored in lookup tables as functions of time and the corresponding thermals.

In Chapter 4 , the glider’s autopilot is designed based on classical control theory. A linearized and decoupled glider model is used to design the longitudinal and lateral-directional controllers separately. Nonlinear simulation is used to verify the controller designs. The bare-frame glider dynamic modes are evaluated and compared to the flying qualities listed in Nelson and Foster. A longitudinal axis controller is designed to hold pitch angle command. Lateral-directional axes controllers are designed to take yaw rate command from either the heading controller or the autonomous soaring guidance controller.

In Chapter 5 the four parts of the guidance system are presented. Mode logic is used to determine whether the glider should be in normal flying mode or autonomous soaring mode. Turn logic is used to determine if the glider should make a clockwise or counterclockwise turn when a thermal is detected. Thermal properties, such as the thermal center location, radius, and vertical velocity information are estimated by the thermal identification algorithm. This information is subsequently used to help determine how steep a turn the glider should bank to center a thermal in the soaring controller.

Chapter 6 gives supporting algorithms of interpolation and a transfer function routine. Interpolation methods are used to interpolate the predetermined aerodynamic stability & control derivatives tables. Four interpolation methods with different curve smoothness results and computational performances are included in the HSF library. The transfer function routine is used to convert parts of the controller designs to the time domain state space form. Verifications of both methods are also given in this chapter.

In Chapter 7 a series of simulation flights are used to demonstrate the result of the improved aircraft and thermal models. The turn logic algorithm is validated through simple glider flights, making the correct turn direction into thermals on either side of the glider. The benefit of this additional algorithm is explored by the soaring race result of two gliders, one with and one without the turn logic. The improvement on the thermal model is validated through another flight with three thermals varying their shapes, strengths, and locations.

Chapter 8 summarizes the improvements made to the aircraft model, thermal model, and the guidance system. It also reviews this project's contribution to Horizon Simulation Framework. Some possible future work is also suggested.

Chapter 2 Aircraft Models

This chapter covers the tools used to construct an aircraft simulation. Modeling methods for aircraft simulation are extensively published; thus, their detailed derivations are not repeated here. The aircraft simulation methods used in this research are based on the work of Rolfe²³ and Diston.²⁰ Aircraft aerodynamics prediction methods vary greatly throughout the published literature. DATCOM+ and AVL were used in this project because these two methods are computationally fast, prediction results are well acknowledged by the academic community, and they are free. The implementation and derivation of the thermal induced roll moment are given next. Basic Earth's gravitational and atmospheric models are included in the last portion of this chapter. The detailed description of the coding implementation of this project can be found through CPSTAR.²⁷

2.1 Quaternion Method for Aircraft Attitude

The procedure of using Quaternion method to determine the aircraft orientation is outlined in Figure 2.1. First, the Quaternion rates $\dot{e}_0, \dot{e}_1, \dot{e}_2, \text{ and } \dot{e}_3$ are calculated using the previous time step Quaternion parameters $e_0, e_1, e_2, \text{ and } e_3$ and aircraft rotational rates $p, q, \text{ and } r$. The Quaternion rates are integrated to obtain the next time step Quaternion parameters. Next, the intermediate variables direction cosine functions are calculated and used to calculate the Euler angles and the direction cosine matrix (DCM), used for coordinate frame transformation.

Figure 2.1 Aircraft attitude calculation implementation using Quaternions

At the beginning of a simulation, if the Euler angles are known, the Quaternion parameters can be initialized by using the set of equations given by Eqs. (2.1).

$$\begin{aligned}
 e_0 &= \cos\left(\frac{1}{2}\psi\right) \cos\left(\frac{1}{2}\theta\right) \cos\left(\frac{1}{2}\phi\right) + \sin\left(\frac{1}{2}\psi\right) \sin\left(\frac{1}{2}\theta\right) \sin\left(\frac{1}{2}\phi\right) \\
 e_1 &= \cos\left(\frac{1}{2}\psi\right) \cos\left(\frac{1}{2}\theta\right) \sin\left(\frac{1}{2}\phi\right) - \sin\left(\frac{1}{2}\psi\right) \sin\left(\frac{1}{2}\theta\right) \cos\left(\frac{1}{2}\phi\right) \\
 e_2 &= \cos\left(\frac{1}{2}\psi\right) \sin\left(\frac{1}{2}\theta\right) \cos\left(\frac{1}{2}\phi\right) + \sin\left(\frac{1}{2}\psi\right) \cos\left(\frac{1}{2}\theta\right) \sin\left(\frac{1}{2}\phi\right) \\
 e_3 &= -\cos\left(\frac{1}{2}\psi\right) \sin\left(\frac{1}{2}\theta\right) \sin\left(\frac{1}{2}\phi\right) + \sin\left(\frac{1}{2}\psi\right) \cos\left(\frac{1}{2}\theta\right) \cos\left(\frac{1}{2}\phi\right)
 \end{aligned} \tag{2.1}$$

Rotational rates are obtained from the integration of the EOM, and the Quaternion parameters are obtained from the integration of the Quaternion rates. If these variables are known, Quaternion rates can be solved by Eqs. (2.2). Note that Quaternion parameters need to be normalized first to eliminate numerical error from the previous integration process.

$$\begin{aligned}\dot{e}_0 &= -\frac{1}{2}(e_1 p + e_2 q + e_3 r) \\ \dot{e}_1 &= \frac{1}{2}(e_0 p + e_2 r - e_3 q) \\ \dot{e}_2 &= \frac{1}{2}(e_0 q + e_3 p - e_1 r) \\ \dot{e}_3 &= \frac{1}{2}(e_0 r + e_1 q - e_2 p)\end{aligned}\tag{2.2}$$

Once the Quaternion rates are integrated to yield the Quaternion parameters, the direction cosines can be calculated

$$\begin{aligned}l_1 &= e_0^2 + e_1^2 - e_2^2 - e_3^2 \\ l_2 &= 2(e_1 e_2 + e_0 e_3) \\ l_3 &= 2(e_1 e_3 - e_0 e_2) \\ m_1 &= 2(e_1 e_2 - e_0 e_3) \\ m_2 &= 2(e_0^2 - e_1^2 + e_2^2 - e_3^2) \\ m_3 &= 2(e_2 e_3 + e_0 e_1) \\ n_1 &= 2(e_0 e_2 + e_1 e_3) \\ n_2 &= 2(e_2 e_3 - e_0 e_1) \\ n_3 &= e_0^2 - e_1^2 - e_2^2 + e_3^2\end{aligned}\tag{2.3}$$

Direction cosines are then used to calculate the Euler angles

$$\begin{aligned}\phi &= \tan^{-1}(l_2 / l_1) \\ \theta &= \sin^{-1}(-l_3) \\ \psi &= \tan^{-1}(m_3 / n_3)\end{aligned}\tag{2.4}$$

Coordinate transformations between the local frame and the body frame can be performed through Eqs.

(2.5). The matrix that converts one frame to another frame is called the direction cosine matrix.

$$\begin{aligned}\begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} &= \begin{bmatrix} l_1 & l_2 & l_3 \\ m_1 & m_2 & m_3 \\ n_1 & n_2 & n_3 \end{bmatrix} \begin{bmatrix} x_{local} \\ y_{local} \\ z_{local} \end{bmatrix} \\ \begin{bmatrix} x_{local} \\ y_{local} \\ z_{local} \end{bmatrix} &= \begin{bmatrix} l_1 & m_1 & n_1 \\ l_2 & m_2 & n_2 \\ l_3 & m_3 & n_3 \end{bmatrix} \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix}\end{aligned}\tag{2.5}$$

MATLAB/Simulink's built-in blocks *Quaternions to Rotational Angles*, *Quaternion to Directional Cosine Matrix*, and *Rotational Angles to Quaternions* were used to verify the implementation of the

Quaternion method in HSF. A series of inputs were used in both Simulink's built-in blocks and HSF's Quaternion implementation; both programs yielded the same result.

2.2 Equations of Motion

Derived from Newton's law, the kinematic equations-of-motion that govern the translational and rotational movements of a rigid aircraft are given in Eqs. (2.6). Assumptions used in their derivation are:

- Local frame is treated as an inertial frame
- The aircraft has rigid structure and is treated as a point mass
- The mass of the aircraft is constant with respect to time
- The aircraft is symmetric about the *body* xz plane; hence $I_{xy} = 0$ and $I_{yz} = 0$

$$\begin{aligned}
 \dot{u} &= X/m + rv - qw \\
 \dot{v} &= Y/m + pw - ru \\
 \dot{w} &= Z/m + qu - pv \\
 \dot{p} &= \frac{LI_z + NI_{xz} + pq(I_x I_{xz} - I_y I_{xz} + I_z I_{xz}) + qr(I_y I_z - I_{xz}^2 - I_z^2)}{I_x I_z - I_{xz}^2} \\
 \dot{q} &= [M + pr(I_z - I_x) + (r^2 - p^2)I_{xz}]/I_y \\
 \dot{r} &= [N + pq(I_x - I_y) + (\dot{p} - qr)I_{xz}]/I_z
 \end{aligned} \tag{2.6}$$

Where:

$\dot{u}, \dot{v}, \dot{w}$	Accelerations in the i_b, j_b , and k_b direction
u, v, w	Velocities in the i_b, j_b , and k_b direction
p, q, r	Body roll rate, pitch rate, and yaw rate
X, Y, Z	Forces in the i_b, j_b , and k_b direction
L, M, N	Moments about the i_b, j_b , and k_b vectors
I_x, I_y, I_z, I_{xz}	Moments of inertia
m	Glider mass

2.3 Aerodynamic Forces and Moments

The aerodynamic forces and moments would be first solved in the stability frame using Eqs. (2.7) and then converted to body frame using Eqs. (2.8).

$$\begin{aligned}
 X_{sb} &= -C_D QS \\
 Y_{sb} &= C_Y QS \\
 Z_{sb} &= -C_L QS \\
 L_{sb} &= C_l QSb \\
 M_{sb} &= C_m QSc \\
 N_{sb} &= C_n QSb \\
 Q &= \frac{1}{2} \rho V^2
 \end{aligned} \tag{2.7}$$

$$\begin{aligned}
\begin{bmatrix} X_b \\ Y_b \\ Z_b \end{bmatrix} &= \begin{bmatrix} \cos(\alpha) & 0 & -\sin(\alpha) \\ 0 & 1 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix} \begin{bmatrix} X_{sb} \\ Y_{sb} \\ Z_{sb} \end{bmatrix} \\
\begin{bmatrix} L_b \\ M_b \\ N_b \end{bmatrix} &= \begin{bmatrix} \cos(\alpha) & 0 & -\sin(\alpha) \\ 0 & 1 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix} \begin{bmatrix} L_{sb} \\ M_{sb} \\ N_{sb} \end{bmatrix}
\end{aligned} \tag{2.8}$$

Where:

X_{sb}, Y_{sb}, Z_{sb}	Forces in the stability axis
L_{sb}, M_{sb}, N_{sb}	Moments in the stability axis
X_b, Y_b, Z_b	Forces in the body axis
L_b, M_b, N_b	Moments in the body axis
Q	Dynamic pressure, N/m^2
α	Angle of attack, rad
C_L	Lift coefficient
C_D	Drag coefficient
C_y	Side force coefficient
C_l	Roll moment coefficient
C_m	Pitch moment coefficient
C_n	Yaw moment coefficient

The non-dimensional stability coefficients C_L, C_D, C_y, C_l, C_m , and C_n are calculated by Eqs. (2.9). Next section would cover the methods of obtaining the stability and control derivatives used to calculate the nondimensional stability coefficients. The definition of the nondimensional stability and control derivatives are given in Appendix A .

$$\begin{aligned}
\text{Lift:} \quad C_L &= C_{L_{wbh}}\{\alpha\} + (C_{L_{\dot{\alpha}}} \dot{\alpha} + C_{L_q} q) c / (2V_{\infty}) + C_L\{\delta_e\} \\
\text{Drag:} \quad C_D &= C_{D_{wbh}}\{\alpha\} + C_D\{\alpha, \delta_e\} \\
\text{Side Force:} \quad C_y &= C_{y_{\beta}}\beta + (C_{y_p} p + C_{y_r} r) b / (2V_{\infty}) + C_{y_{\delta_r}} \delta_r \\
\text{Roll Moment:} \quad C_l &= C_{l_{\beta}}\beta + (C_{l_p} p + C_{l_r} r) b / (2V_{\infty}) + C_l\{\delta_a\} + C_{l_{\delta_r}} \delta_r \\
\text{Pitch Moment:} \quad C_m &= C_{m_{wbh}}\{\alpha\} + (C_{m_{\dot{\alpha}}} \dot{\alpha} + C_{m_q} q) c / (2V_{\infty}) + C_m\{\delta_e\} \\
\text{Yaw Moment:} \quad C_n &= C_{n_{\beta}}\beta + (C_{n_p} p + C_{n_r} r) b / (2V_{\infty}) + C_n\{\alpha, \delta_a\} + C_{n_{\delta_r}} \delta_r
\end{aligned} \tag{2.9}$$

2.4 Stability and Control Derivatives

This study focused on methods of modeling the aircraft aerodynamic characteristics, rather than on optimizing the design of an UAV. Hence, the choice of the aircraft model was based on the completeness of the available geometry information. The chosen aircraft is the HYPER-DL 1.5M sailplane. Methods

used to estimate the aerodynamic characteristics are DATCOM+ and AVL. Details of these two programs are given here along with their limitations. They were chosen to complement each other and to validate their results. For this simulation, the lift, drag, pitch moment, and roll moment derivatives are obtained from DATCOM+, whereas the side force and yaw moment derivatives are obtained from AVL.

2.4.1 Glider Geometry

HYPER-DL is a small, remote controlled, hand launched, sailplane made of balsa wood, fiberglass, resin, foam, and black poplar veneer wood sheeting; shown in Figure 2.2.²⁸

Figure 2.2 HYPER-DL 1.5M sailplane²⁸

A general summary of the glider's information is given in Table 2.1. Detailed geometry information of the HYPER-DL was obtained from its manufacturer's website and by using scaled digitized measurements from its 3-view plan. Detail digitized measurements of the HYPER-DL are given in Appendix B . Verifications of these measurements were performed through visual comparison; an example is given in Figure 2.3.

Adjustments were made to the aircraft model, as the information on the HYPER-DL was incomplete. The original airfoil of the HYPER-DL is the *HN1033* modified, instead a similar airfoil *S7012* with more support data was used. The value of the moment of inertia for the HYPER-DL was not available so the inertia value of Mark Drela's *SuperGee*, which has similar dimensions and weight, was used instead.²⁹

Table 2.1 Modeled glider's general information

HYPER-DL 1.5M sailplane	
Wing Span	1500 mm
Wing Area	22.0 dm ²
Body Length	975 mm
Glider Weight	230 g
Wing Profile	S7012

Figure 2.3 Glider manufacturer's plan drawing with partially digitized body outline

2.4.2 DATCOM+

Aerodynamics prediction can be manually calculated based on classical aerodynamic theories. However, if a detailed solution is required, this calculation process can become tedious. Back in the times when computers were not accessible to designers, the USAF commissioned McDonnell Douglas to create a systematic summary of methods for estimating aircraft aerodynamic stability and control (S&C) derivatives. This collection of papers resulted in currently available USAF Stability and Control Datcom. Datcom is a semi-empirical method that uses both aerodynamic theories and aircraft historical trends to estimate the aircraft's S&C derivatives.

Digital Datcom, a computerized version of the original Datcom was made available in 1978. Digital Datcom has the same implementation as the original Datcom, but dramatically shortens the time requires for the calculation process. Digital Datcom was written in FORTRAN. Its input file requires the geometric dimensions of the modeled aircraft. Its output text file stores the S&C derivatives. The interface of Digital Datcom is not very user friendly. The input text file is difficult to decipher and is thus easy to make mistakes. The output is a single text file which is not easy to manipulate or to interface with other programs.

Instead of using Digital Datcom, DATCOM+³⁰ developed by Bill Galbraith was used in this simulation. DATCOM+ is wrapper program that improved the input and output interfaces of the original

Digital Datcom. The original code of Digital Datcom is retained in DATCOM+ with few minor execution errors corrected. Unlike Digital Datcom, the input file of DATCOM+ allows comments to be included, making inputting the geometry of an aircraft model much easier. For example, the meaning of the reference parameters would not be so apparent to the designer without the comments in Figure 2.4.

```
*****
* Reference Parameters *   pg 29
*****
* SREF   Reference area value of theoretical wing area used by program
*         if not input
* CBARR   Longitudinal reference length value of theoretical wing
*         Mean Aerodynamic Chord used by program if not input
* BLREF   Lateral reference length value of wing span used by program
* ROUGFC  Surface roughness factor, equivalent sand roughness, default
*         to 0.16e-3 inches (Natural sheet metal)
*         0.02/0.08E-3 - Polished metal or wood
*         0.16E-3 - Natural sheet metal
*         0.25E-3 - Smooth matte paint, carefully applied
*         0.40E-3 - Standard camouflage paint, average application
$OPTINS SREF=320.8, CBARR=6.75, BLREF=51.7, ROUGFC=0.25E-3$
```

Figure 2.4 Excerpt from the example input file for DATCOM+

The output of DATCOM+ consists of five files. The first file is the original Digital Datcom output file. The second file is a XML format file which stores the S&C derivatives; this file can be used by JSBSim or FlightGear directly. The third file, which was used for this research, is an Excel format file, which also stores the S&C derivatives. It is provided for the user convenience as the data can be easily manipulated in Excel or exported to other software; the author used this to extract the needed S&C derivatives to construct the linearized glider model. The fourth file generates custom plots of the S&C derivatives. The last file is an AC3D format file that shows the three-dimensional view of the modeled aircraft. This visual representation helped identify many incorrect geometry configuration inputs for this project. An example image of the HYPER-DL is shown in Figure 2.5. The input and output files of DATCOM+ used in this simulation are shown in Appendix C .

Figure 2.5 Graphical representation of the glider geometry in DATCOM+

Operational limitations of DATCOM+, similar to those of Digital Datcom, are discussed here. Prediction capability of Datcom is limited to traditional wing-body-tail aircraft configuration. Resulting S&C derivatives do not include rudder to yawing and rolling moment terms. For Reynolds number below 100,000, the result validity is questionable. Unfortunately, the Reynolds number for the flight condition of this simulation is around 100,000; thus the output from AVL is used to verify the result from DATCOM+. If the results from the two programs are in close agreement, the validity of the stability and control derivatives used in this simulation can be justified.

2.4.3 Athena Vortex-Lattice

Another category of aerodynamic S&C derivatives prediction method is based on Computational Fluid Dynamics (CFD). The fundamental goal of CFD is to solve the Navier-Stokes equations that describe the fluid dynamics. Generally it is not feasible to solve the closed form equations directly; thus, a numerical method is used to solve the simplified form of the Navier-Stokes equations. If the air flow is assumed to be inviscid and irrotational, the Navier-Stokes equations can be simplified to potential-flow theory.

Athena Vortex Lattice (AVL), developed by Mark Drela and Harold Youngren at MIT, is used here.²⁹ AVL is one of the existing computer programs that are based on potential-flow theory. The vortex lattice method divides the surface of an aircraft into small panels with varying strength horseshoe vortex filaments placed on each panel. The program then proceeds to simultaneously solve the circulation on each vortex filament, and calculates the forces and moments generated by each panel. Unlike Datcom, AVL allows designers to model unconventional aircraft geometries. The AVL input and output files are included in Appendix D .

Figure 2.6 Graphical representation of the glider geometry in AVL

Operational limitations of AVL include: AVL does not provide derivatives relating to drag or the rate of change of angle of attack; the validation of the result is limited to small angles of attack and sideslip; the oscillatory motion of the aircraft must be slow enough so that the period of oscillation is much longer than the time it takes the flow to traverse an airfoil chord;³⁸ finally, the results from AVL do not capture any nonlinear effects, such as stall.

2.4.4 Comparison and Validation

Using the same geometry, inertia, and flight conditions as inputs to both DATCOM+ and AVL, the aerodynamics prediction of these two programs show similar results, given in Figure 2.7 to Figure 2.10. Significant differences were observed in the side force and yaw moment terms. These can be attributed to the prediction methods and the limitations of the two programs. DATCOM+'s prediction on lateral and directional aerodynamic characteristics are also limited.

One common difference that is seen in C_L , C_m , C_D , C_{l_p} , C_{l_r} , C_{n_p} , C_{n_r} , and $C_{n_{\delta a}}$ comparison plots is that the stall effect was captured by DATCOM+, whereas AVL showed linear result. The difference in C_{L_q} and C_{m_q} plots might be caused by the different horizontal tail moment arms interpretation of the two programs. The lower drag values predicted by AVL did not account for induced drag, whereas DATCOM+ does not demonstrate this shortcoming.

In conclusion, noting the exceptions above, the results from DATCOM+ and AVL were in good agreement. Nevertheless, CFD, wind tunnel, or flight test would be needed if more accurate results are desired.

Figure 2.7 Datcom and AVL comparison of lift and pitch moment S&C derivatives

Figure 2.8 Datcom and AVL comparison of drag and roll moment coefficients

Figure 2.9 Datcom and AVL comparison of side force coefficients

Figure 2.10 Datcom and AVL comparison of yaw moment coefficients

2.5 Thermal Induced Roll

When a glider enters a thermal region, the impact of the updraft velocity will be stronger on the wing closer to the center of the thermal, than on the outer wing. To model this effect, each wing's lift force was calculated separately to account for the roll moments induced by the thermal. Asymmetrical lift distribution on the wing in the spanwise direction would cause a roll moment on the glider. Equivalent lift vector is used instead of the lift distributions for ease of calculation; illustration of this concept is given in Figure 2.11.

Figure 2.11 Uneven lift from left and right wing

The calculation of the individual lift force of each wing requires the non-dimensional stability coefficients C_L of the wing alone configuration, each wing's angle of attack, and freestream velocity. Individual wing C_L can be obtained from DATCOM+. Each wing's angle of attack and freestream velocity are calculated using the glider and the updraft velocities.

2.5.1 Moment Arm

In order to find the induced roll moments, the moment arm of the location of the equivalent lift vector is needed. It was assumed that the glider wing has an elliptical circulation distribution; according to Anderson,³¹ this is a reasonable approximation for a high aspect ratio tapered wing. Two assumptions were made in deriving the equivalent spanwise location for the lift vectors.

- Circulation distribution of the wing is elliptical
- Changes of V_∞ is negligible along the spanwise direction

Starting with the elliptical circulation distribution³¹

$$\Gamma(y) = \Gamma_0 \sqrt{1 - (2y/b)^2} \quad (2.10)$$

where Γ_0 is the circulation at wing root, the elliptical lift distribution can be calculated as

$$L'(y) = \rho_\infty V_\infty \Gamma(y) \quad (2.11)$$

Then the individual wing lift force is given by:

$$L_{right} = \int_0^{b/2} \rho_{\infty} V_{\infty} \Gamma(y) dy \quad (2.12)$$

$$L_{right} = \rho_{\infty} V_{\infty} \int_0^{b/2} \Gamma(y) dy$$

Individual wing moment about the C.G. of the aircraft can be found in similar way,

$$M_{right} = \rho_{\infty} V_{\infty} \int_0^{b/2} \Gamma(y) y dy \quad (2.13)$$

Finally the spanwise location of the equivalent lift vector was solved, yielding

$$\bar{b} = M_{right} / L_{right}$$

$$\bar{b} = \frac{\rho_{\infty} V_{\infty} \int_0^{b/2} \Gamma(y) y dy}{\rho_{\infty} V_{\infty} \int_0^{b/2} \Gamma(y) dy} = \frac{b^2 \Gamma_0 / 12}{b \Gamma_0 \pi / 8} = \frac{2}{3\pi} b \quad (2.14)$$

2.5.2 Angle of Attack and Freestream Velocity

The lift coefficient is a function of angle of attack that each respective wing sees in the freestream airflow.

The angle of attack and the freestream velocity of each wing can be found by solving Eqs. (2.20)

$$\alpha_{left} = \tan^{-1} \frac{w + w_{UL}}{u}$$

$$\alpha_{right} = \tan^{-1} \frac{w + w_{UR}}{u} \quad (2.15)$$

$$V_{left} = \sqrt{u^2 + v^2 + (w + w_{UL})^2}$$

$$V_{right} = \sqrt{u^2 + v^2 + (w + w_{UR})^2}$$

Where:

$\alpha_{left}, \alpha_{right}$	Angles of attack that the left and right wings see
u, v, w	Body frame velocities
w_{UL}, w_{UR}	Body frame vertical updraft velocities of the left and right wings

Wing-only lift stability coefficients $C_{L_{right}}$ and $C_{L_{left}}$ are obtained from Datcom as a function of angle of attack.

2.5.3 Induced Roll Moment

Finally, the roll moment caused by these asymmetrical lift forces of each wing can be calculated by solving Eqs. (2.16).

$$\begin{aligned}
L_{left} &= C_{L_{left}} \left(\frac{1}{2} \rho V_{left}^2 \right) \left(\frac{1}{2} S \right) \\
L_{right} &= C_{L_{right}} \left(\frac{1}{2} \rho V_{right}^2 \right) \left(\frac{1}{2} S \right) \\
Roll\ Moment &= L_{left} \bar{b} - L_{right} \bar{b}
\end{aligned} \tag{2.16}$$

2.6 Gravitational Model

Newton's law of universal gravitation is presented in Eqs. (2.17).

$$F_g = \frac{GMm}{r^2} \tag{2.17}$$

Where:

F_g	Gravitational force between the Earth and the glider
G	Gravitational constant, $m^3 kg^{-1} s^{-2}$
M	Earth mass, kg
m	Glider mass, kg
r	Distance between the center of gravity of Earth and the glider
GM	$3.986004418 \times 10^{14} m^3 s^{-2}$

2.7 Atmospheric Model

The ARDC 1959 standard atmosphere model given in Anderson³¹ is used in this aircraft simulation. The key defining parameter of the atmospheric model is the variation of the temperature as altitude increases. Regions of the atmosphere can be classified as isothermal or gradient regions. In isothermal regions, temperature remains constant as altitude increases, whereas in the gradient regions, temperature varies linearly with the change in altitude. The rate of change of temperature as altitude increases is called the lapse rate. Table 2.2 shows the atmospheric properties at the beginning of each isothermal and gradient region.

Table 2.2 ARDC 1959 standard atmosphere properties

Geopotential Altitude (m)	Temperature (K)	Pressure (Pa)	Density (kg/m ³)	Lapse Rate (K/m)
0	288.2	1.013E+05	1.225E+00	0.0000
11000	216.7	2.263E+04	3.639E-01	-0.0065
25000	216.7	2.489E+03	4.003E-02	0.0000
47000	282.7	1.205E+02	1.485E-03	0.0030
53000	282.7	5.835E+01	7.191E-04	0.0000
79000	165.7	1.010E+00	2.124E-05	-0.0045
90000	165.7	1.045E-01	2.198E-06	0.0000
105000	225.7	7.460E-03	1.152E-07	0.0040

The geopotential altitude is a fictitious altitude used to simplify the calculation of atmospheric properties.

$$h = \frac{r}{r + h_G} h_G \quad (2.18)$$

Where:

h	Geopotential altitude, m
h_G	Geometric altitude, m
r	Radius of Earth, m . $r = 6.356766 \times 10^6$

For the isothermal regions, once the temperature is obtain from Table 2.2, with the corresponding altitude, the pressure and density can be calculated from Eqs. (2.19).

$$\begin{aligned} p/p_1 &= e^{-g_0(h-h_1)/(RT)} \\ \rho/\rho_1 &= e^{-g_0(h-h_1)/(RT)} \end{aligned} \quad (2.19)$$

Similarly, for the gradient regions, once the temperature is interpolated from Table 2.2, the pressure and density can be calculated from Eqs. (2.20).

$$\begin{aligned} p/p_1 &= (T/T_1)^{-g_0/(aR)} \\ \rho/\rho_1 &= (T/T_1)^{-\{[g_0/(aR)]+1\}} \end{aligned} \quad (2.20)$$

Where:

T_1, P_1, ρ_1	Temperature, pressure, and density at the base altitude of the region of interest
g_0	Gravitational constant at sea level, m/s^2 . $g_0 = 9.80665$
h	Geopotential altitude, m
h_1	Geopotential altitude at the beginning of the layer of interest, m
R	Gas constant, $287.058 J/kg^{-1}K^{-1}$
a	Lapse rate, K/m

The atmospheric model implemented in HSF is verified against the published data from Anderson; the results are identical, as shown in Figure 2.12.

Figure 2.12 Comparison between published and calculated temperature as altitude is increased

This chapter covered the tools that were needed to construct the base of the aircraft simulation, explained the rationale in choosing DATCOM+ and AVL as the aerodynamic tools, and showed the derivation of the thermal induced roll moment calculation.

Chapter 3 Thermal Model

The equations used to construct Allen's thermal model are not repeated here; detailed derivation and implementation can be found in his research paper.⁹ Thus, a brief overview of Allen's thermal model is given first, followed by the methods of modeling the additional thermal characteristics.

3.1 Allen's Thermal Model

Convective mixing-layer thickness z_i is the maximum above-ground-altitude an updraft can reach; it is calculated by combining rawinsonde balloon readings and ground property measurements. Convective scale velocity w^* is a scaling parameter, based on the surface heat condition at the location of thermal. These two parameters are used in the calculation of the thermal's radius and vertical velocity.

A thermal's shape is defined by its height z_i and radius r_2 at each altitude, whereas its strength is defined by the vertical velocity w . The outer radius of a thermal is a function of altitude and convective mixing-layer thickness.

$$r_2 = function\{z, z_i\} \quad (3.1)$$

The updraft vertical velocity is a function of the altitude, convective mixing-layer thickness, convective scale velocity, distance from the updraft center, and updraft outer radius.

$$w = function\{z, z_i, w^*, r, r_2\} \quad (3.2)$$

According to conservation of mass, an equal amount of downward moving air mass (environmental sink) is needed to balance out the upward moving air mass of a thermal. Environmental sink is presented everywhere in the test area outside of the thermal.

$$w_e = function\{z, z_i, r, r_2, w_D, N, A\} \quad (3.3)$$

Where:

A	Test area, m^2
N	Number of thermals inside the test area

3.2 Time Dependent Parameters

The time dependent shape and strength properties of a thermal were simulated by multiplying the height, radius, and vertical velocity of the original thermal. A top level diagram in Figure 3.1 demonstrates that the original thermal model properties were modified by the time dependent gains.

Figure 3.1 Implementation of the time dependent variables

The final height, radius, and vertical velocity of a thermal are calculated using Eqs. (3.4).

$$\begin{aligned} z_i &= z_{i_{original}} \times \Delta z_i \\ r_2 &= r_{2_{original}} \times \Delta r_2 \\ w &= w_{original} \times \Delta w \end{aligned} \quad (3.4)$$

Interpolation tables were created to specify the time dependent gains. Each table specified gain values at different point in time for each thermal; an example table is showed in Figure 3.2. Time dependent gain tables for thermal height, radius, and vertical velocity along with other thermal model parameters are given in Appendix E .

<zGain>				
Time, s	0.0	60.0	240.0	300.0
Thermal #1	0.0	1.0	1.0	1.0
Thermal #2	1.5	1.5	1.2	1.0
Thermal #3	0.0	0.8	1.0	1.2
</zGain>				

Convective mixing-layer
thickness gains

Figure 3.2 Example of the time dependent height gain interpolation table

3.3 Drift

The implementation of thermal drift is almost identical to the process described in the previous section. Figure 3.3 and Eqs. (3.5) show the implementation of thermal drift effect. The main difference here is that drift distance is added to the original thermal center location, rather than multiplied, as was the case for the time dependent gains.

Figure 3.3 Implementation of drift

$$\begin{aligned} x_{th} &= x_{th_{original}} + \Delta x_{th} \\ y_{th} &= y_{th_{original}} + \Delta y_{th} \end{aligned} \quad (3.5)$$

Figure 3.4 shows an example of the thermal drift distance in the y direction. Notice that drift values were only specified up to 60 seconds. If a simulation ran for 300 seconds, the remaining drift values would be extrapolated to the nearest set of values at 60 seconds; implying that the updraft would stop moving in the y-direction after 60 seconds.

<yDrift>		Extrapolate to nearest value	
Time, s	0.0	60.0	240.0 300.0
Thermal #1	0.0	18.0	18.0 18.0
Thermal #2	0.0	28.0	28.0 28.0
Thermal #3	0.0	0.0	0.0 0.0
</yDrift>			

Figure 3.4 Example of time dependent drift values in y direction

3.4 Improved Thermal Model

The additional features of the previous two sections are demonstrate through a 300 second long simulation with three thermals varying their shape and strength, and drifting away in different directions; time lapse images of these thermals are shown in Figure 3.5. The thermal parameters used for this example simulation are given in Appendix E .

Figure 3.5 Time lapse images of three thermals

Thermal #1 and Thermal #3 start forming at time = 0 seconds. Thermal #2 was already formed at the start of the simulation. As simulation time moved forward, thermal #1 dissipated at the end; thermal #2 slowly decreased in size and lost its strength; whereas thermal #3 remained relatively constant throughout the simulation. The drift path of thermal #2 was outlined in Figure 3.5 (b).

This chapter presented the summarized version of the basic thermal model from Allen's research and demonstrated the improvements made to the thermal model of this simulation.

Chapter 4 Autopilot Design

The goal of this chapter is to present the design of an autopilot system that would stabilize the glider longitudinal mode as well as that of a turn rate and a heading command controller. Longitudinal and lateral flying qualities from Nelson were used to evaluate the glider's natural dynamics. The load-factor-sensitivity handling quality from Foster was also used to evaluate the longitudinal mode.²⁵ Some self-imposing guidelines were employed to help design the autopilot systems. The modeled glider should reach and hold the commanded pitch angle in about 2 seconds and have less than 20% overshoot, given that the commanded input is not too excessive. Coordinated turns are desired, as it implies that there is no lateral acceleration during turning maneuvers.

4.1 Linear Control Law Design

Initial analysis and controller design are based on the linear model of the glider. Controller designs are then verified and tuned in nonlinear simulation. In linear control analysis, the main assumption is that the glider flight maneuvers are small, and the glider motions are linear. Small perturbation theory was used to linearize the glider model.

4.1.1 Small Perturbation Theory

Small perturbation theory replaced all the variables in the EOM with reference values and a small perturbation. With the assumption that the glider is in a steady state flight, many of the reference values can be neglected, yielding a set of equations in state space form, given by Eq. (4.1) for longitudinal mode, and Eq. (4.2) for lateral-directional mode.³²

$$\begin{bmatrix} \Delta \dot{u} \\ \Delta \dot{w} \\ \Delta \dot{q} \\ \Delta \dot{\theta} \end{bmatrix} = \begin{bmatrix} X_u & X_w & 0 & -g \\ Z_u & Z_w & u_0 & 0 \\ M_u + M_{\dot{w}}Z_u & M_w + M_{\dot{w}}Z_w & M_q + M_{\dot{w}}u_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta w \\ \Delta q \\ \Delta \theta \end{bmatrix} + \begin{bmatrix} X_{\delta_e} \\ Z_{\delta_e} \\ M_{\delta_e} + M_{\dot{w}}Z_{\delta_e} \\ 0 \end{bmatrix} [\delta_e] \quad (4.1)$$

$$\begin{bmatrix} \Delta \dot{\beta} \\ \Delta \dot{p} \\ \Delta \dot{r} \\ \Delta \dot{\phi} \end{bmatrix} = \begin{bmatrix} Y_{\beta}/u_0 & Y_p/u_0 & Y_r/u_0 - 1 & g \cos \theta_0 / u_0 \\ L_{\beta} & L_p & L_r & 0 \\ N_{\beta} & N_p & N_r & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \beta \\ \Delta p \\ \Delta r \\ \Delta \phi \end{bmatrix} + \begin{bmatrix} 0 & Y_{\delta_r}/u_0 \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \delta_a \\ \Delta \delta_r \end{bmatrix} \quad (4.2)$$

Definitions of the dimensional stability derivatives used above are given in Appendix A .

4.1.2 Longitudinal Analysis

The natural longitudinal dynamics of the modeled glider are given in Table 4.1, which also includes the criteria for achieving level 1 and 2 handling qualities.³² With no control laws implemented, the glider has level 1 Short-period mode handling qualities, whereas the Phugoid mode only satisfied level 2 flying qualities³²; this would be remedied in the subsequent analysis.

Table 4.1 Glider natural longitudinal response and handling qualities

Mode	Eigenvalue	ζ	ω_n (rad/s)	Handling Qualities
Phugoid	$-0.0261 + 0.858i$	0.03	0.86	Level 1: $\zeta > 0.04$
	$-0.0261 - 0.858i$	0.03	0.86	Level 2: $\zeta > 0$
Short Period	$-14.2 + 5.31i$	0.94	15.20	Level 1: $0.35 < \zeta < 1.30$
	$-14.2 - 5.31i$	0.94	15.20	Level 2: $0.25 < \zeta < 2.00$

Where:

ζ Damping ratio
 ω_n Natural frequency, radian/second

The load-sensitivity factor n/α for the modeled glider is calculated to be 10.6 g's/rad . According to the flying qualities criterion from Foster, the HYPER-DL satisfied Level-1 load-sensitivity factor flying qualities, as shown in Figure 4.1.

$$\frac{n}{\alpha} = \frac{QC_{L\alpha}}{W/S} \quad (4.3)$$

Figure 4.1 Load-sensitivity factor flying qualities

A pitch attitude angle controller is considered for the longitudinal mode. The design criteria are to achieve less than 20% overshoot and settling time of less than 2 seconds. Figure 4.2 shows the block

diagram of the longitudinal mode. The inner pitch rate feedback loop acted as a stability augmentation system (SAS) and was intended to stabilize the glider and improve damping on the longitudinal modes. The outer feedback loop was used to reduce pitch angle steady state error and improve transient response.

Figure 4.2 Longitudinal control loop

Where:

θ_{com}	Commanded pitch attitude angle
q	Pitch rate
θ	Pitch attitude angle
K_q	Inner loop rate feedback gain

All actuators used in the linear and nonlinear analysis are modeled as first order transfer functions with a time constant of 0.1 seconds, given in Eq. (4.4).

$$actuator = 10/(s + 10) \quad (4.4)$$

Both frequency and time domain analysis were used to explore the effects of the inner loop rate feedback gains (k_q) on the outer loop. In the frequency domain analysis, root-contour plot of the outer loop in Figure 4.3 showed that the Phugoid mode's natural frequency decreased with increased k_q ; for a second order system, a small natural frequency would lead to long settling time. The Short-period mode's natural frequency was significantly higher than the Phugoid mode's, thus the Short-period mode would settle much faster and was not included in Figure 4.3. The Bode-contour in Figure 4.4 showed that with increased k_q , the phase margin increased, but at the cost of reduced crossover frequency that indicates slow response. Thus, both contour plots suggested that increasing k_q would produce sluggish response. The frequency domain analysis indicates that a small k_q is desired.

Figure 4.3 Root-contour of different pitch rate feedback gains, Phugoid mode

Figure 4.4 Bode-contour plot of different pitch rate feedback gains

In the time domain analysis, the residue is attributed to the amplitude contribution of each mode to the output response, whereby a smaller residue is desired. Table 4.2 shows that as k_q increased, the residues of both Phugoid and Short-period modes became smaller, in contrast to the frequency domain analysis.

Table 4.2 Pitch rate feedback gain effects on time domain analysis

k_q	θ/δ_{com}					Residue and Phase				
	Phugoid	Short-period	ζ_{ph}	ζ_{sp}	$\sigma_{actuator}$	$R_{actuator}$	R_{ph}	Φ_{ph}	R_{sp}	Φ_{sp}
0.00	-0.026 + 0.86i	-14.23 + 5.31i	0.028	0.94	-10	-7.75	1.89	-15	2.80	-41
0.01	-0.030 + 0.84i	-23.83 + 5.50i	0.032	0.93	-10.8	-7.39	1.83	-14	3.17	-50
0.10	-0.048 + 0.73i	-11.88 + 8.64i	0.062	0.81	-14.4	-1.05	1.39	-11	2.22	111
1.00	-0.065 + 0.39i	-10.77 + 23.15i	0.160	0.42	-16.5	-0.024	0.41	-10	0.42	153
3.00	-0.064 + 0.23i	-10.62 + 39.27i	0.253	0.26	-16.8	-0.003	0.16	-15	0.16	165

Where:

k_q	Pitch rate feedback gain
σ	Attenuation
R	Residue
ϕ	Phase shift
ph	Phugoid
sp	Short-period

From the above analysis, pitch rate feedback gains from 0.1 to 1 have a balanced crossover frequencies and residue values. These feedback gains were further experimented with different outer loop compensators to determine a better controller design.

Proportional-integral-derivative (PID) controllers were used to design the outer loop compensator for the different pitch rate feedback gains. Time domain and frequency domain performance results of these different controllers are summarized in Table 4.3. All PID controllers were tuned manually; hence, the results might not be the optimum solutions. However all controller designs satisfied the self imposed performance criteria. The longitudinal system without inner loop feedback ($k_q=0$) was also considered because it produced an unexpected fast settling time. In the nonlinear analysis, results between the PID controllers with $k_q = 0.0$ and $k_q = 0.9$ were compared, as they have shown the best performance so far. Step responses of these pitch angle controllers are shown in Figure 4.5.

PID controllers were used in this study to help achieve better performance for the autopilot. In actual hardware implementation, PID controllers are more difficult to work with. An alternative would be the

lead-lag compensators, which are easier to implement. The lead-lag compensators can be configured to mimic the performance of the PID controllers at the desired bandwidth.

Table 4.3 PID compensator designs for different rate feedback gains

Compensator	Transfer Function	%OS	Tr	Ts	e _{ss}	GM	PM	ω_c
PID ($k_q=0.0$)	$(1.01s^2 + 4.07s + 3.48)/s$	17	0.07	0.36	0	∞	47	20
PID ($k_q=0.1$)	$(0.63s^2 + 5.54s + 7.94)/s$	20	0.09	0.93	0	∞	49	16
PID ($k_q=0.3$)	$(0.46s^2 + 5.96s + 15.89)/s$	16	0.10	0.93	0	∞	62	12
PID ($k_q=0.5$)	$(0.81s^2 + 9.91s + 27.03)/s$	14	0.07	0.93	0	∞	53	20
PID ($k_q=0.7$)	$(1.02s^2 + 14.99s + 10.00)/s$	13	0.06	0.86	0	∞	42	25
PID ($k_q=0.9$)	$(0.88s^2 + 17.25s + 10.03)/s$	11	0.06	0.84	0	∞	46	25

Figure 4.5 Step response of pitch angle controllers

4.1.3 Lateral Analysis

The glider's bare-frame lateral-directional flying qualities were summarized in Table 4.4, which shows that it satisfied all level-1 flying qualities. However, the spiral mode was unstable; this is acceptable if a pilot is controlling the aircraft, but not in the case of UAV. Two methods were investigated to stabilize the spiral mode: feedback of the roll angle to the aileron control, and feedback of the body yaw rate to rudder control. Root loci of both methods are shown in Figure 4.6. Both methods could stabilize the spiral mode, but when their close loop step responses are compared in Figure 4.7, the roll angle to aileron control exhibited faster settling time and less oscillation. Thus, the roll angle feedback design would be used to stabilize the spiral mode.

Table 4.4 Glider natural lateral response and handling qualities

Mode	Eigenvalue	t_{double}	τ	ζ	ω_n (rad/s)	Handling Qualities - Level 1
Spiral	0.05	13.75		1	0.05	$t_{double} > 12$ sec
Dutch Roll	$-1.71 \pm 4.47i$			0.36	4.79	$\zeta > 0.19$; $\zeta\omega_n > 0.35$; $\omega_n > 1.0$
Roll	-22.6		0.044	1	22.6	$\tau < 1.0$

t_{double} is the time it takes the spiral mode to double its amplitude.

Figure 4.6 Roll angle to aileron and body yaw rate to rudder root loci

Figure 4.7 Roll angle to aileron and body yaw rate to rudder unity feedback step responses

Proportional, proportional-Integral (PI), and PID controllers were tested in place of the roll angle to aileron unity feedback gain to yield better results. Time and frequency domain performance results are summarized in Table 4.5, and the step responses of the three controllers were shown in Figure 4.8. The simple proportional controller was used because it had no overshoot, had acceptable settling time and steady state error, and would be less demanding on the actuator.

Table 4.5 Compensator designs for roll angle to aileron loop

Compensator	Transfer Function	%OS	Tr	Ts	e _{ss}	GM	PM	ω_c
P	0.4	0	1.30	2.4	0.04	27	78	1.4
PI	$(1.5s + 0.03)/s$	14	0.24	1.63	0	15	56	5.0
PID	$(0.63s^2 + 3.88s + 0.75)/s$	2	0.11	0.4	0	∞	63	17.4

Figure 4.8 Roll attitude angle controllers comparison

The goal of the lateral mode analysis was to design a yaw rate controller and a heading controller. In autonomous soaring mode, the yaw rate controller would receive input from the autonomous soaring guidance algorithm, whereas in normal flying mode, the yaw rate controller would receive input from the heading controller to hold the commanded heading. The final lateral-directional control law block diagram is shown in Figure 4.9 to aid navigating through this section. Five feedback loops were used in this multiple-input-multiple-output system. The order of loop closure is also labeled in Figure 4.9.

Figure 4.9 Lateral-directional controller block diagram

After the spiral mode was stabilized with the roll angle feedback loop, the rudder control loop was closed to ensure coordinate turn. Rate of change of the sideslip angle $\dot{\beta}$ was used to increase damping of the inner loop. Different rate feedback gains were tested for this control loop; step responses of these designs were shown in Figure 4.10. Feedback gain of $K_{\dot{\beta}} = 0.2$ was used due to its short settling time and small oscillation.

Figure 4.10 Step response of various rate feedback gains for $\dot{\beta}$ to rudder

Further simulation testing showed that even though the inner loop controller drove $\dot{\beta}$ to zero, the sideslip angle generated by the initial $\dot{\beta}$ was not cancelled. Sideslip angle was feedback to drive its steady state error to zero. Different compensator designs are compared in Table 4.6. The integral-lag compensator was chosen because it has no overshoot, and suffers from less oscillatory response.

Table 4.6 Compensator design for coordinate turn

Compensator	Transfer Function	%OS	Tr	Ts	e _{ss}	GM	PM	ω_c
P	2	61	0.18	5	0.39	7	25	6.6
ILC	$(0.24s + 1)/s$	0	2.78	5.21	0	15	91	0.8
PID	$(1.51s^2 + 5.48s + 4.98)/s$	9	0.09	2.8	0	∞	55	16.8

Next, a yaw rate controller was needed to receive command input from the normal and autonomous soaring mode. Three manually tuned PID controller step responses are shown in Figure 4.11 and their performances are summarized in Table 4.7. In Figure 4.11, the yaw rates first started moving in the opposite direction to that commanded, then self-corrected shortly. This is caused by the right-hand plane zero shown in the r/r_{com} transfer function in Eq. (4.5), indicative of a non-minimum phase system. The actual physical phenomenon that caused this behavior is adverse yaw. PID-3 will be used because of its superior performance.

Figure 4.11 Yaw rate controllers

Table 4.7 Yaw rate PID controllers performance

Compensator	Transfer Function	%OS	Tr	Ts	e _{ss}	GM	PM	ω_c
PID-1	$(0.01878 + 0.178s + 0.50)/s$	4	1.42	4.3	0	13	64	0.86
PID-2	$(0.0085s^2 + 0.015s + 0.348)/s$	2	2.09	3.4	0	17	67	0.61
PID-3	$(0.015 + 0.173s + 0.45)/s$	2	1.64	2.7	0	13	67	0.79

$$\frac{r}{r_{com}} = \frac{-1.59s^3 - 205.6s^2 - 283.5s + 12790}{s^6 + 45.94s^5 + 717.5s^4 + 5424s^3 + 24690s^2 + 69610s + 70020} \quad (4.5)$$

Zeros	7.05	-8.95	-127.49	
Poles	-1.79	-6.97	-23.6	-10 -1.77 ± 4.54i

In normal flying mode, the glider is given a heading command to search for thermals. A simple proportional controller is sufficient for the glider to track the heading direction. Different forward loop proportional gains were compared; with time and frequency domain performance results shown in Table 4.8. K_{ψ} -2 was chosen because it had no overshoot, and had fast settling time. At first glance the settling time of 8.7 seconds seems long, but if a large heading command is given to the aircraft, it is reasonable to expect the glider to take a few seconds to reach the desired heading.

Table 4.8 Heading controllers comparison of forward path proportional gain

Compensator	Forward Path Gain	%OS	Tr	Ts	e_{ss}	GM	PM	ω_c
K_{ψ} -1	0.2	0	8.3	15.9	0	19.5	77.5	0.20
K_{ψ} -2	0.3	0	4.4	8.7	0	16.0	71.1	0.30
K_{ψ} -3	0.4	3	2.9	6.9	0	13.5	64.6	0.40
K_{ψ} -4	0.6	17	1.8	9.4	0	10.0	51.9	0.60
K_{ψ} -5	0.8	31	1.4	11.3	0	7.5	40.2	0.76

4.2 Nonlinear Control Law Design

Controller designs in the linear analysis are verified in this section. Some modifications were made to the compensators to account for nonlinearity effects.

4.2.1 Longitudinal Analysis

The open inner loop PID controller and the PID ($k_q = 0.9$) controller were tested in the nonlinear simulation and the results summarized in Table 4.9. The PID ($k_q = 0.9$) controller had to be modified to account for oscillatory motion in the nonlinear simulation. The new compensator is reflected in the step response comparison of the two controller designs in Figure 4.12. A PID ($k_q = 0.9$) controller was chosen for the final design because it had a faster settling time and performed better in turbulence, as shown in Figure 4.12 to Figure 4.15.

Table 4.9 Nonlinear simulation controller performance comparison

Compensator	Transfer Function	%OS	Tr	Ts	e_{ss}
PID ($k_q=0.0$)	$(1.01s^2 + 4.07s + 3.48)/s$	6	0.42	10.8	0
PID ($k_q=0.9$)	$(0.88s^2 + 17.25s + 10.03)/s$	12	0.26	1.69	0

Figure 4.12 Pitch angle in nonlinear simulation

Figure 4.13 Elevator control activity in nonlinear simulation

Figure 4.14 Step response with Dryden Turbulence Model in nonlinear simulation

Figure 4.15 Elevator control activity with Dryden Turbulence Model

4.2.2 Lateral Analysis

Figure 4.16 shows that the transient responses of both linear and nonlinear simulations were similar, but the nonlinear simulation had a greater steady state error. This was acceptable because the inner loop was meant to stabilize the spiral mode and the error tracking was handled by the yaw rate controller.

Figure 4.16 Linear and nonlinear roll angle controller step responses

Figure 4.17 shows that the yaw rate autopilot yielded similar result for both linear and nonlinear simulations. The corresponding aileron and rudder deflections activities for the step yaw rate command are shown in Figure 4.18, indicating that both control deflection activities were within reasonable limits.

Figure 4.17 Linear and nonlinear yaw rate step response comparison

Figure 4.18 Nonlinear simulation lateral-directional control activity

For normal flight mode, the linear and nonlinear heading controllers were compared in Figure 4.19, the step responses show exact fit. Figure 4.20 and Figure 4.21 show that the heading controller can tolerate small amount of turbulence. Nevertheless, as it is difficult to keep track of a small heading angle if turbulence is present; hence, a heading angle command of 45 degrees was used in the turbulence analysis.

Figure 4.19 Heading angle controllers comparison

Figure 4.20 Heading controller in turbulence

Figure 4.21 Turbulence used in heading controller analysis

This chapter presented the detailed controller design process to stabilize the glider. First, the controllers were designed with a simplified linear model. Then, they were verified in nonlinear simulation and the controller designs were modified in order to yield the optimal solution. Turbulence was added to the nonlinear simulation to test the robustness of the controller designs.

Chapter 5 Thermal Soaring Guidance

The autonomous soaring guidance used in this simulation is performed in four consecutive steps: 1) The mode logic engages the glider into soaring mode using estimated thermal vertical velocity. 2) The turn logic determines the turn direction of the glider, based on the direction it was pushed away in by a thermal. 3) The thermal identification algorithm estimates the thermal location, radius, and vertical velocity by using centroid calculation and iterative-fit algorithm. 4) Finally, the soaring controller calculates the yaw rate command based on glider pilots' rules-of-thumb.

5.1 Mode Logic

The thermal vertical velocity that an aircraft experienced can be estimated by the aircraft's specific energy. Specific energy is the sum of the kinetic energy and potential energy of an aircraft, normalized by the aircraft weight, as given by Eq. (5.1).

$$e = h_{est} + \frac{V^2}{2g} \quad (5.1)$$

When an aircraft flies through a thermal, the vertical wind velocity of a thermal would directly increase the velocity terms of the specific energy. The thermal's vertical velocity directly contributes to the rate of change of the specific energy $\dot{e}$; hence $\dot{e}$ is a good estimation for the thermal vertical velocity. According to glider pilots' experience, a glider should change from normal flying mode to soaring mode when the glider is experiencing the strongest lift, near the center of a thermal. The mode logic is based on this guideline. The pseudo code of the mode logic is given below

*if $\dot{e} > 1.0$ && $\ddot{e} < 0$
engage soaring mode
if $\dot{e} < 0.1$ for more than 5 seconds
disengage soaring mode*

The specific energy acceleration $\ddot{e}$ is used to determine the inflection point of the assumed normal vertical velocity profile; $\ddot{e}$ would change from positive before passing the thermal center, to negative after it passed the thermal center. The values used in the mode logic can be changed depending on the type of vehicle used, and when the designer wants the aircraft to enter and leave a thermal.

5.2 Turn Logic

When a glider enters a thermal, it would be rolled away by nonuniform lift distribution within a thermal. The glider turning logic can use this phenomenon to determine which way the glider should turn. If a glider rolled to the left when it encountered a thermal, then it should turn to the right toward the center of the thermal and vice versa. Simulation showed that the change in heading/yaw angle is more prominent than the roll angle when a glider is being pushed away by a thermal; hence, the change in heading angle is used as the determining factor for the turn logic algorithm.

In order to observe the effect of nonuniform thermal lift distribution, the glider was sent flying straight through a thermal without interacting with the thermal; this is done to measure the thermal vertical velocities at three different locations on the glider. Figure 5.1 shows that the difference of the updraft velocity at the left and right wing is small; however, this continuous effect does induce a roll moment on the aircraft. The glider was sent through a thermal once again without the autopilot system, but with the interaction between the thermal and glider enabled. Results in Figure 5.2 show that the thermal is capable of changing the glider's heading angle by more than 90 degrees; this large maneuver is partially attributed to the unstable spiral mode. In Figure 5.2, the magnitude of the updraft velocity is represented by the color of the flight path; the red portion of the flight path indicates that the glider was flying close to the thermal.

Figure 5.1 Updraft velocities felt by glider at CG, left wing, and right wing locations

Figure 5.2 is a bird's-eye view of the glider flight path with the updraft velocity magnitude represented by color. The vertical axis on this graph represents the local frame x axis, which points north, where as the horizontal axis represents the local frame y axis, which points east. This type of graphical representation of the simulation results would be continuously used throughout the rest of this paper.

Figure 5.2 Glider flying through a thermal without autopilot

Next, the glider was sent through a thermal with the autopilot system designed in Chapter 4 and was commanded to go straight ahead. The glider's response was shown in Figure 5.3. The thermal is located at $x = 200$ and $y = 200$. The changes in roll and yaw angle were shown in Figure 5.3(a), with the change in yaw angle more noticeable. The scale on the horizontal axis in Figure 5.3(b) was zoomed in to show the change in the flight path. The K_{ψ} gain was changed from 0.3 to 0.1 to deteriorate the performance of the heading controller, in order to observe more deviation in the heading angle.

Figure 5.3 Glider flying through a thermal with autopilot on

The glider's default turn direction is clockwise. Thus, if a thermal is on the port side of a glider, the turn logic should use the initial detected heading angle error caused by the thermal induced roll moment to bank the glider into a counterclockwise turn. The turn logic algorithm is shown in the following pseudo code.

```

if (mode == normal) && (heading error > 2 degree)
    turn counterclockwise
else
    turn clockwise

```

5.3 Thermal Identification

Allen's thermal identification algorithm is used here and a brief description is given. A set of first-in-first-out data containers was used to store the past 45 seconds' positions and rates of change of specific energy $\dot{e}$ information. The data were first normalized and corrected for thermal drift and the thermal center's location was calculated by using centroid method on the data set. An iterative fit method was used to estimate the thermal's radius and vertical velocity.

5.4 Soaring Controller

Soaring controller used here is based on Allen's research, with modifications made to account for the different glider platform used. Allen suggested that the glider should soar at 65% of the thermal's radius, Eq. (5.2).

$$r_{cmd} = 0.65r_{th} \quad (5.2)$$

Position error and steady state turn rate were used in the soaring controller. The position error is given by

$$e_p = r_{cmd} - S \quad (5.3)$$

Where:

r_{th}	Estimated radius, m
S	Estimated distance between thermal and glider, m

In a coordinated turn, the steady state turn rate is calculated as

$$\psi_{ss} = \left(\frac{V}{r_{cmd}} \right) \left(\frac{180}{\pi} \right) \quad (5.4)$$

Figure 5.4 shows the soaring controller used in this simulation. Reichmann's rules given in Chapter 3 were applied as design guideline for this controller. The rate of change of $\dot{e}$ corresponds to the improvement and deterioration of the glider's climb performance; hence $\ddot{e}$ contribution to the soaring controller was used to satisfy rule #1 and #2. For rule #3, if the climb remains constant, the steady state turn rate from Eq. (5.4) would keep the bank angle constant. The position error was included to help reduce steady state error.

Figure 5.4 Soaring controller

This chapter covered the thermal soaring guidance, which is separate into four main components: 1) mode logic, 2) turn logic, 3) thermal identification, and 4) soaring controller; emphasis was on the turn logic.

Chapter 6 Simulation Implementation

Additional methods important to the simulation implementation in HSF are included here. Interpolation methods used to interpolate values from stability and control derivatives and the updraft model parameters are discussed first. The controller designed for the autopilot and the soaring guidance were converted to state space form for HSF, an overview of this process is given next.

6.1 Interpolation

In flight simulation, aircraft aerodynamic stability and control (S&C) derivatives are stored in lookup tables, as a function of one or two parameters. Table 6.1 and Table 6.2 are examples of S&C derivative lookup tables from DATCOM+. During simulation, the drag coefficient C_D can be interpolated from the table with the given aircraft angle of attack value. Most of the S&C derivatives used in this simulation only depend on one input variable, as shown in Table 6.1 that has only two columns, one for the S&C derivative, and the other for the variable; such table would be referred to as one-dimensional table. Some of the derivatives do depend on two variables, an example is given by Table 6.2; such table would be referred to as two-dimensional table.

Table 6.1 Drag coefficient as function of angle of attack

α (deg)	C_D
-16	0.093
-12	0.057
-8	0.031
-4	0.018
0	0.017
4	0.029
8	0.054
12	0.088
16	0.121
20	0.080

Four interpolation methods were added to the HSF library. Linear interpolation and natural cubic spline interpolation methods were used to interpolate one-dimensional tables, whereas bilinear interpolation and point fitting interpolation methods were used to interpolate two-dimensional tables. Linear and bilinear interpolations were based on the algorithms outlined in *Numerical Recipes in C*³³; natural cubic spline method is based on the algorithms outlined in *Numerical Mathematics and Computing*³⁴; and point fitting method is developed by Dr. Jimenez from California Polytechnic State University's Mathematics

department. Pseudo codes of these methods are presented in Appendix F and the verification and performance comparisons are discussed next.

Note that in Table 6.2, data across rows and columns follow an increasing, decreasing, or parabolic trend except at the row of -8 angle of attack. The values from 0 to 20 degrees of elevator deflection decreased from 0 to -0.001, and then increased to 0 again. This deviation can be attributed to the numerical error inherent in the computational calculation. Another possible explanation is that the program DATCOM calculates the result base on an elevator deflection and varies the angle of attack each time; hence, not necessary capturing all the changes across different elevator deflections.

Table 6.2 Change in drag coefficient due to angle of attack and elevator deflection

α (deg)	C_D						
-16	0.010	0.008	0.005	0.000	-0.003	-0.003	-0.003
-12	0.008	0.006	0.004	0.000	-0.002	-0.002	-0.001
-8	0.006	0.005	0.003	0.000	-0.001	0.000	0.001
-4	0.004	0.003	0.001	0.000	0.001	0.002	0.002
0	0.002	0.001	0.000	0.000	0.002	0.003	0.004
4	0.000	0.000	-0.001	0.000	0.003	0.005	0.006
8	-0.002	-0.002	-0.002	0.000	0.004	0.007	0.008
12	-0.004	-0.004	-0.003	0.000	0.005	0.009	0.010
16	-0.006	-0.006	-0.004	0.000	0.006	0.011	0.013
20	-0.011	-0.009	-0.007	0.000	0.009	0.014	0.017
δ_e (deg)	-30	-20	-10	0	10	20	30

Linear interpolation method assumes the region within the given data points is piece-wise continuous and values between every set of data points are linearly connected. Natural cubic spline method not only assumes the data points are continuous, it also assumes that the first and second derivatives are continuous, which enhances the smoothness of the interpolated curve. Both methods were used to interpolate the drag coefficient in Table 6.1, and the results are shown in Figure 6.1.

Figure 6.1 Interpolation of the one-dimension drag coefficient table

Bilinear interpolation method is basically an extension of the linear interpolation method for two-dimensional tables; the interpolated values of this method lie on a flat surface in between grid points. The point fitting method uses a best estimation method to calculate the coefficients of a surface that fits the given data points with the least amount of error. Results from both methods are the same; however their implementation and computational performance differ. Figure 6.2 is used to graphically verify the two-dimensional interpolation methods; point fitting interpolation method result was not shown because it was the same as the bilinear method.

Figure 6.2 Interpolation of the two-dimensional drag coefficient table

The implementation of the linear, natural cubic spline, and point fitting methods can be divided into two steps. The first step calculates a set of coefficients that are used to construct the interpolating function and the second step evaluates the interpolating function with the input variable and the pre-calculated coefficients. Note that the pre-calculated coefficients are only calculated once and stored in the computer memory at the beginning of the simulation. Thus, during simulation, only the second step is performed. On the other hand, the bilinear method performs the calculations of both steps during the simulation. From a computational stand point, the linear, natural cubic spline, and point fitting methods use less processing power and can run faster because they do not repeatedly calculate the coefficients during a simulation. Performance comparison of these four methods is presented in Table 6.3; note that these results are computer dependent. The natural cubic spline method takes 50% more time to run than linear method, but provides a smoother curve. Bilinear and point fitting methods yield the same result, but point fitting method is faster because it pre-calculates its coefficients.

Table 6.3 Computational performance of the interpolation methods

Method	# of Run	Total Time (second)	Time/Run (second)
Linear	50000	4.0	0.00008
Cubic Spline	50000	6.0	0.00012
Bilinear	266000	31.0	0.00012
Point Fitting	266000	22.9	0.00009

The decision of whether to use linear or natural cubic spline method depends on the given data set. In general, linear interpolation method is preferred because of its faster computational performance compare to the spline method. However, if the resulting interpolated curve has an abrupt sharp corner such as the one in Figure 6.1, natural cubic spline method should be used to yield more realistic result. An alternative is to add more data points around the sharp corner region so that the linear method can capture the curvature around that point. As mention earlier, the results from bilinear and point fitting method are the same. If the computer memory can hold all the pre-calculated coefficients, point fitting method should always be used. In situation where there are more coefficients than the allowed memory space, bilinear method should be used instead.

6.2 Frequency to state space domain

Horizon Simulation Framework models its system in time domain state space form, whereby the RK45 integrator is used to solve the continuous state values of the modeled system as time progresses during the simulation. The autopilot and soaring controller described in Chapter 4 and Chapter 5 were modeled using frequency domain techniques. In this section, all previous controller designs were then systematically converted from frequency domain to state space domain by using the transfer function routine. The routine used here is based on the algorithm developed by NASA³⁵ and MATLAB.³⁶ The pseudo code of this routine is given in Appendix G . In this chapter, the conversion method was demonstrated using a generic transfer function; the actual algorithm can be deduced from the following example. MATLAB/Simulink was used to verify this algorithm in the second section of this chapter.

6.2.1 Example

A simple generic transfer function is used, where zeros z_1, z_2, z_3 and poles p_1, p_2, p_3 are known values.

$$\frac{y(s)}{u(s)} = \frac{z_1 s^2 + z_2 s + z_3}{p_1 s^2 + p_2 s + p_3} \quad (6.1)$$

The transfer function is treated as two separate functions.

$$u \rightarrow \boxed{\frac{1}{s^2 + (p_2/p_1)s + (p_3/p_1)}} \xrightarrow{w} \boxed{(z_1/p_1)s^2 + (z_2/p_1)s + (z_3/p_1)} \rightarrow y \quad (6.2)$$

$$\frac{w}{u} = \frac{1}{s^2 + (p_2/p_1)s + (p_3/p_1)}$$

$$\frac{y}{w} = (z_1/p_1)s^2 + (z_2/p_1)s + (z_3/p_1)$$

The transfer function $\frac{w}{u}$ is converted to time domain by cross multiplying its left and right sides and performing inverse-Laplace transform.

$$\ddot{w}(t) + (p_2/p_1)\dot{w}(t) + (p_3/p_1)w(t) = u(t) \quad (6.3)$$

The time domain equation is then rearranged into two ordinary-differential-equations in state space format.

$$\dot{x} = Ax + Bu$$

$$\begin{bmatrix} \ddot{w} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} -p_2/p_1 & -p_3/p_1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{w} \\ w \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u \quad (6.4)$$

The $\frac{y}{u}$ transfer function was also converted to time domain

$$y(t) = (z_1/p_1)\ddot{w}(t) + (z_2/p_1)\dot{w}(t) + (z_3/p_1)w(t) \quad (6.5)$$

Finally, it was also rearranged into state space format

$$y = Cx + Du$$

$$y = \left[-\left(\frac{p_2}{p_1}\right)\left(\frac{z_1}{p_1}\right) + \frac{z_2}{p_1} \quad -\left(\frac{p_3}{p_1}\right)\left(\frac{z_1}{p_1}\right) \right] \begin{bmatrix} \dot{w} \\ w \end{bmatrix} + \left[\frac{z_1}{p_1} \right] u \quad (6.6)$$

6.2.2 Derivative approximation

Derivative terms used in the PID controller and the rate of change of the specific energy were approximated by the transfer function given by Eq.(6.7). For large constant term N, the s term in the denominator would be negligible, approximating the entire expression by s. $N = 50$ was used in this simulation.

$$\boxed{\frac{du}{dt}} \cong \frac{Ns}{s + N} \quad (6.7)$$

6.2.3 Verification

A sine wave signal was used as the input for a transfer function Eq.(6.8) and the state space form of that transfer function. The state space form was generated by the previously described algorithm. The outputs of the transfer function and the state space matrix were identical, as shown in Figure 6.3.

$$tf = \frac{2.3s^4 + 1.17s^3 + 4.1s^2 - 11s}{1.1s^4 + 2.2s^3 + 4.4s + 5.5} \quad (6.8)$$

Figure 6.3 Transfer function routine verification

This chapter presented two important processes, interpolation and frequency to state space domain transformation, used by this study's simulation.

Chapter 7 Simulation Result

This chapter presents, a series of soaring flights simulated in Horizon Simulation Framework, used to demonstrate the improved flight simulation features. In the first two flights, thermals are placed on the starboard and port sides of the glider. The resulting flights demonstrated the utility of the turn logic. This was followed by two gliders sent flying through a set of three thermals with the thermals' locations randomly generated for 20 simulation flights; one of the gliders had the turn logic and the other did not. Performance difference between the two gliders was interpreted from the simulated race result. In the last part of this chapter, three thermals were created with time varying shape, strength, and drift characteristics; the resulted glider flight path demonstrated the interaction between the improved thermal and the glider.

7.1 Turn Logic

A glider was sent flying straight ahead with: (a) a thermal on its starboard side, and (b) a thermal on its port side. These two flights, shown in Figure 7.1 and Figure 7.2, demonstrate that the glider turned in the correct direction for both flights, validating that the turn logic was performing as designed. Note that in both flights initially the gliders were slightly pushed away by the thermals. These two flights also show that the gliders can center and stay soaring within a thermal.

Figure 7.1 Clockwise turn demonstration flight

Figure 7.3(a) was included to show the three-dimensional view of the first flight. Figure 7.3(b) showed the thermal model that the glider encountered in its flight. The glider flight path and the thermal model were separated for the sake of clarity.

Figure 7.2 Counterclockwise turn demonstration flight

Figure 7.3 Three-dimensional view of the clockwise turn demonstration flight

7.2 Thermal Soaring Race

The goal of this race was to observe the benefits, if any exist, of including the thermal induced roll effect and the turn logic algorithm. Hence, other time dependent properties of a thermal were not activated in this simulated race.

Two gliders, one with the turn logic – referred to as glider #1, and one without the turn logic – referred to as glider #2, were placed in a thermal soaring race. For every simulation flight, a set of three thermals were randomly positioned in the rectangular race course measuring: x from 50 to 250 meters, and y from 100 to 1100 meters; with each thermal positioned at least 200 meters away from each other. The positions of these twenty sets of thermals are shown in Figure 7.4. Both gliders started at the same position at $x = 100$ meters and $y = 150$ meters, and the same set of thermals were presented for each flight.

Figure 7.4 Twenty sets of thermals randomly positioned in the race course

The final results of the twenty flights are given in Table 7.1. In nine flights, the gliders were too far away from the thermals and failed to detect them. In seven flights, the detected thermals were positioned on the starboard side of the gliders. Both gliders successfully circled the thermals; however, these flights did not trigger the turn logic. In four flights, the detected thermals were positioned on the port side of the

gliders. The final x location, altitude, and specific energy of the two gliders in Table 7.1 showed that the glider #1 traveled farther down the race course but was at a lower altitude. Flight #10 in Figure 7.5 was used to demonstrate this result. Since glider #1 made the initial correct turn and entered the thermal sooner than glider #2, as a result, glider #1 also reached the top of the thermal and exited earlier than glider #2. Thus glider #1 used more of the altitude gain inside the thermal to travel farther. This finding can be useful to shorten the cross country flight time. Note that even though flight #15's turn direction was on the port side, a closer look at the flight paths and the thermals in Figure 7.6 revealed that the thermals were almost right in front of the gliders. Hence, the gliders had very similar results.

Table 7.1 Summary of the soaring flight race

Flight #	Thermal Encountered	Turn Direction	With Turn Logic			Without Turn Logic		
			x (m)	Altitude (m)	e (m)	x (m)	Altitude (m)	e (m)
1	#2	Starboard	933	682	683	682	683	933
2	#1	Port	912	662	664	751	707	709
3	#1	Port	1028	658	660	766	703	704
4	None	N/A	1684	495	496	1684	495	496
5	#2	Starboard	942	689	690	942	689	690
6	None	N/A	1684	504	506	1684	504	506
7	None	N/A	1684	515	517	1684	515	517
8	None	N/A	1684	493	494	1684	493	494
9	None	N/A	1683	487	489	1683	487	489
10	#1	Port	921	685	686	750	709	711
11	None	N/A	1683	477	478	1683	477	478
12	#2	Starboard	869	702	703	869	702	703
13	#1	Starboard	939	662	664	939	662	664
14	None	N/A	1684	496	498	1684	496	498
15	#1	Port	984	670	671	975	658	659
16	None	N/A	1683	489	490	1683	489	490
17	None	N/A	1684	492	493	1684	492	493
18	#1	Starboard	986	680	681	986	680	681
19	#2	Starboard	841	693	694	841	693	694
20	#1 & #3	Starboard	918	692	693	918	692	693

Figure 7.5 Flight #10 of thermal soaring race

Figure 7.6 Flight #15 of thermal soaring race

In summary, only four of the twenty simulation flights engaged the first thermal on the port side of the glider and utilized the turn logic algorithm to travel farther down the race course. On average, glider #1 (with turn logic) traveled 150 meters farther and was at 26 meters lower in altitude than glider #2. This finding can be useful to help shorten the time for cross country flight.

7.3 Shape, Strength, and Drift

In this simulated flight, three thermals were created with temporal variation of height, radius, and vertical velocities. All three thermals were drifting in elliptical patterns for the entire simulation period of 500 seconds. Thermal #1 faded away in 60 seconds; thermal #2 faded in 350 seconds; and thermal #3 lasted the entire duration of the simulation. The size and strength of the three thermals were set to decrease from thermal#1 to thermal#3; these effects were not as apparent as the dissipation effect. Figure 7.7(b) shows the drift paths of the three thermals. Figure 7.7(a) shows that the glider disengaged from thermal#1 in only two

circles; due to the short duration of thermal#1 of only 60 seconds. The updraft velocity, as indicated by the color of the flight path in Figure 7.7(a), shows that thermal#1's vertical velocity is stronger than thermal#2, whereas thermal #3 had the weakest lift.

Figure 7.7 Demonstration flight to show thermal's time dependent properties

This chapter showed that the turn logic is working as designed by placing thermals on the left and right side of the glider's flight path. The twenty simulation flights also demonstrated that the glider with the turn logic can be useful to shorten the cross country flight time. The last simulation showed that the glider react to the thermal models as expected.

Chapter 8 Conclusion

In this study, the changes made in the aircraft model, the thermal model, and the guidance algorithm improved the thermal soaring flight simulation. The aircraft model includes the nonuniform thermal vertical velocity distribution in the glider's aerodynamic moment calculation. This models the effect of an aircraft being pushed away when it encounters a thermal. The thermal model enables the time dependent parameters height, radius, vertical velocity and location to be varied. This feature allows researchers to model the constantly morphing thermals occurring in nature more closely. The guidance algorithm is adjusted to utilize the thermal induced roll effect to direct the glider to turn toward a thermal. An aircraft with this algorithm would be able to enter a thermal sooner than possible otherwise.

For the Horizon Simulation Framework, the successful implementation and simulation results of the autonomous thermal soaring flight had validated the Dynamic-ECI position type feature of the framework. Some of the mathematical models and algorithms used in this aircraft simulation were added to the HSF library and can be reused by other projects.

A few suggestions for future work are given here. The thermal model can be further improved by modeling the merging effects of two or more thermals. Currently, the architecture of varying a thermal's height, radius, vertical velocity, and locations is integrated in the thermal model. However, the variation of these parameters in an actual thermal as time progresses throughout a day was not utilized. Actual measurement of these thermal properties, which can be found in other reported studies, should be used in future simulations. Only one UAV was simulated in the thermal environment, for future projects multiple UAVs can be simulated at the same time with them interacting with each other.

Bibliography

- ¹Langelaan, J. W., Long Distance/Duration Trajectory Optimization for Small UAVs, AIAA Guidance, Navigation and Control Conference and Exhibit, 20-23 August 2007, Hilton Head, South Carolina.
- ²Patel, C. K., and Kroo, I., *Control Law Design for Improving UAV Performance Using Wind Turbulence*, 44th AIAA Aerospace Sciences Meeting and Exhibit, 9-12 January 2006, Reno, Nevada.
- ³Boslough, M. B. E., *Autonomous Dynamic Soaring Platform for Distributed Mobile Sensor Arrays*, Scandia National Laboratories, 2002.
- ⁴Kerlinger, P., *Flight Strategies of Migrating Hawks*, University of Chicago Press, Chicago, 1989.
- ⁵Kahveci, N. E., Ioannou, P. A., and Mirmirani, M. D., *A Stochastic Approach to Optimal Soaring Problem and Robust Adaptive LQG Control*, AIAA 2007 Conference and Exhibit, 7-10 May 2007, Rohnert Park, California.
- ⁶Sachs, G., and Costa, O. D., *Optimization of Dynamic Soaring at Ridges*, AIAA Atmospheric Flight Mechanic Conference and Exhibit 11-14 August 2003, Austin, Texas.
- ⁷FAA, *Glider Flying Handbook*, Aviation Supplies & Academics, New Castle, WA, 2004.
- ⁸Allen, M. J., *Autonomous Soaring for Improved Endurance of a Small Uninhabited Air Vehicle*, 43rd AIAA Aerospace Sciences Meeting and Exhibit, 10-13 January 2005, Reno, Nevada.
- ⁹Allen, M. J., *Updraft Model for Development of Autonomous Soaring Uninhabited Aerial Vehicles*, AIAA-2006-1510.
- ¹⁰Allen, M. J., and Lin, V., *Guidance and Control of an Autonomous Soaring Vehicle with Flight Test Results*, 45th AIAA Aerospace Science Meeting and Exhibit, 8-11 January 2007, Reno, Nevada.
- ¹¹Edward, D. J., *Implementation Details and Flight Test Result of an Autonomous Soaring Controller*, 2008
- ¹²Kahveci, N. E., and Ioannou, P. A., *Optimal Flight Trajectories Through Recursive Thermal Streets with Uncertain Climb Rates*, 49th AIAA conference, 7-10, April 2008, Schaumburg, Illinois
- ¹³Andersson, K., Kaminer, I., Jones, K. D., Dobrokhodov, V., and Lee, D., *Cooperating UAVs Using Thermal Lift to Extend Endurance*, AIAA conference, 6-9 April 2009, Seattle, Washington.
- ¹⁴Lee, D., Andersson, K., Jones, K. D., *Hybrid Control of Long Endurance Unmanned Aerial Vehicles for Robust Wireless Communication Networking*, AIAA-2009-6099.
- ¹⁵Andersson, K., and Kaminer, I., *On Stability of a Thermal Centering Controller*, AIAA Guidance, Navigation, and Control Conference, 10-13 August 2009, Chicago, Illinois
- ¹⁶Hazard, M., W., *Unscented Kalman Filter for Thermal Parameter Identification*, 48th AIAA Aerospace Sciences Meeting, 4-7 January 2010, Orlando, Florida
- ¹⁷Adan, C., *Earth Center Earth Fixed Vehicle Simulation*, senior project, California Polytechnic State University, San Luis Obispo, California, 2006
- ¹⁸Sims, A. T., *Design and Control Considerations for a Skid-to-Turn Unmanned Aerial Vehicle*, thesis, Aerospace Engineering department, California Polytechnic State University, San Luis Obispo, CA, 2009
- ¹⁹Anderson, J. D. Jr., *Introduction to Flight*, 5th Ed., McGraw-Hill, Boston, 2005
- ²⁰Diston, D. J., *Computational Modeling and Simulation of Aircraft and the Environment, Vol 1*, John Wiley & Sons, Atrium, United Kingdom, 2009.
- ²¹Turevskiy, A., Gage, S., and Buhr, C., *Model-Based Design of a New Light-weight Aircraft*, AIAA Modeling and Simulation Technologies Conference and Exhibit, 20-23 August 2007, Hilton Head, South Carolina
- ²²Leong, H. I., Jager, R., Keshmiri, S., and Colgren, R., *Development of a Pilot Training Platform for UAVs Using a 6DOF Nonlinear Model with Flight Test Validation*, AIAA Modeling and Simulation Technologies Conference and Exhibit 18-21 August 2008, Honolulu, Hawaii
- ²³Rolfe, J. M., and Staples, K. J., *Flight Simulation*, Cambridge University Press, Cambridge, United Kingdom, 1986.

- ²⁴Ghoreyshi, M., Da Ronch, A., Badcock, K. J., Dees, J., Berard, A., and Rizzi, A., *Aerodynamic Modeling for Flight Dynamics Analysis of Conceptual Aircraft Designs*, 27th AIAA Applied Aerodynamic Conference, 22-25 June 2009, San Antonio, Texas
- ²⁵Foster, T. M., and Bowman, W. J., *Dynamic Stability and Handling Qualities of Small Unmanned-Aerial-Vehicles*, 43rd AIAA Aerospace Sciences Meeting and Exhibit 10-13 January 2005, Reno, Nevada
- ²⁶O'Connor, C., M., and Mehiel, E., A., Butler, B., *HORIZON 2.1: A Space System Simulation Framework*, AIAA Modeling and Simulation Technologies conference and Exhibit 18-21 August 2008, Honolulu, Hawaii
- ²⁷Mehiel, E., A., *HORIZON: A Systems-of-systems Simulation Framework Users Guide*, Aerospace Engineering Department, Cal Poly, San Luis Obispo
- ²⁸Art Hobby, HYPER-DL 1.5M DLG, <http://www.arthobby.com> [cited 30 June 2010]
- ²⁹Drela, M., SuperGee Plan, <http://www.charlesriverrc.org/articles/supergee/supergee.pdf> [cited 30 June, 2010]
- ³⁰Galbraith, B., DATCOM+, <http://www.holycows.net/> [cited 30 June 2010]
- ³¹Anderson, J. D. Jr., *Fundamentals of Aerodynamics*, 4th Ed., McGraw-Hill, Boston, 2005.
- ³²Nelson, R. C., *Flight Stability and Automatic Control*, 2nd ed., McGraw-Hill, Boston, 1998.
- ³³Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T., *Numerical Recipes in C*, Cambridge University Press, New York, 1988
- ³⁴Cheney, W., and Kincaid, D., *Numerical Mathematics and Computing*, Thomson Higher Education, Belmont, 2004
- ³⁵McFarland, R.E., *Transfer Function Routine XFRRUN*, 1990
- ³⁶MATLAB, *MATLAB USER's Guide*, The MathWorks, Inc, Natick, MA, March 2007.
- ³⁷Biezad, D. J., *Integrated Navigation and Guidance Systems*, AIAA Education Series, Reston, VA, 1999.
- ³⁸Drela, M., and Youngren, H., *Athena Vortex Lattice User Guide*, <http://web.mit.edu/drela/Public/web/avl/> [cited 30 June 2010]

Appendix A Stability and Control Derivatives Definition

Nondimensional Stability and Control Derivatives

$C_{L_{wbh}}\{\alpha\}$	Lift coefficient of wing-body-horizontal tail as a function of angle of attack
$C_{L_{\dot{\alpha}}}$	Derivative of lift coefficient with respect to rate change of angle of attack
C_{L_q}	Derivative of lift coefficient with respect to pitch rate
$C_L\{\delta_e\}$	Lift coefficient as a function of elevator deflection
$C_{D_{wbh}}\{\alpha\}$	Drag coefficient of wing-body-horizontal tail as a function of angle of attack
$C_D\{\alpha, \delta_e\}$	Drag coefficient as a function of angle of attack and elevator deflection
C_{y_β}	Derivative of side force coefficient with respect to sideslip angle
C_{y_p}	Derivative of side force coefficient with respect to roll rate
C_{y_r}	Derivative of side force coefficient with respect to yaw rate
$C_{y_{\delta_r}}$	Derivative of side force coefficient with respect to rudder deflection
C_{l_β}	Derivative of roll moment coefficient with respect to sideslip angle
C_{l_p}	Derivative of roll moment coefficient with respect to roll rate
C_{l_r}	Derivative of roll moment coefficient with respect to yaw rate
$C_l\{\delta_a\}$	Roll moment coefficient as a function of aileron deflection
$C_{l_{\delta_r}}$	Derivative of roll moment coefficient with respect to rudder deflection
$C_{m_{wbh}}\{\alpha\}$	Pitch moment coefficient of wing-body-horizontal tail as a function of angle of attack
$C_{m_{\dot{\alpha}}}$	Derivative of pitch moment coefficient with respect to rate change of angle of attack
C_{m_q}	Pitch moment coefficient due to pitch rate
$C_m\{\delta_e\}$	Pitch moment coefficient as a function of elevator deflection
C_{n_β}	Derivative of yaw moment coefficient with respect to sideslip angle
C_{n_p}	Derivative of yaw moment coefficient with respect to roll rate
C_{n_r}	Derivative of yaw moment coefficient with respect to yaw rate
$C_n\{\alpha, \delta_a\}$	Yaw moment coefficient as a function of angle of attack and aileron deflection
$C_{n_{\delta_r}}$	Derivative of yaw moment coefficient with respect to rudder deflection

Dimensional Stability and Control Derivatives

$$\begin{aligned}
X_u &= -(C_{D_u} + 2C_{D_0})QS/(mu_0) \\
X_w &= -(C_{D_\alpha} - C_{L_0})QS/(mu_0) \\
Z_u &= -(C_{L_u} + 2C_{L_0})QS/(mu_0) \\
Z_w &= -(C_{L_\alpha} + C_{D_0})QS/(mu_0) \\
Z_{\dot{w}} &= (C_{z_{\dot{w}}})QS\bar{c}/(2mu_0^2) \\
Z_\alpha &= u_0Z_w \\
Z_{\dot{\alpha}} &= u_0Z_{\dot{w}} \\
Z_q &= (C_{z_q})QS\bar{c}/(2mu_0) \\
Z_{\delta_e} &= C_{z_{\delta_e}}QS/m \\
M_u &= (C_{m_u})QS\bar{c}/(u_0I_y) \\
M_w &= (C_{m_\alpha})QS\bar{c}/(u_0I_y) \\
M_{\dot{w}} &= (C_{m_{\dot{w}}})QS\bar{c}^2/(2I_yu_0^2) \\
M_\alpha &= u_0M_w \\
M_{\dot{\alpha}} &= u_0M_{\dot{w}} \\
M_q &= (C_{m_q})QS\bar{c}^2/(2I_yu_0) \\
M_{\delta_e} &= (C_{m_{\delta_e}})QS\bar{c}/(I_y)
\end{aligned}$$

$$\begin{aligned}
Y_\beta &= (C_{y_\beta})QS/(m) \\
N_\beta &= (C_{n_\beta})QSb/(I_z) \\
L_\beta &= (C_{l_\beta})QSb/(I_x) \\
Y_p &= (C_{y_p})QSb/(2mu_0) \\
N_p &= (C_{n_p})QSb^2/(2I_zu_0) \\
L_p &= (C_{l_p})QSb^2/(2I_xu_0) \\
Y_r &= (C_{y_r})QSb/(2mu_0) \\
N_r &= (C_{n_r})QSb^2/(2I_zu_0) \\
L_r &= (C_{l_r})QSb^2/(2I_xu_0) \\
Y_{\delta_a} &= (C_{y_{\delta_a}})QS/(m) \\
Y_{\delta_r} &= (C_{y_{\delta_r}})QS/(m) \\
N_{\delta_a} &= (C_{n_{\delta_a}})QSb/(I_z) \\
N_{\delta_r} &= (C_{n_{\delta_r}})QSb/(I_z) \\
L_{\delta_a} &= (C_{l_{\delta_a}})QSb/(I_x) \\
L_{\delta_r} &= (C_{l_{\delta_r}})QSb/(I_x)
\end{aligned}$$

Appendix B Glider Geometry Measurements

Geometry coordinates of glider		
Axis	Direction	Origin
x	positive from nose to tail	nose of glider
y	positive from left to right wing tip	symmetrical plane of glider
z	positive from bottom to top of glider body	bottom of glider

Body		Moment of Inertia	
Overall length[m]	0.895	lxx	0.0160
Body nose [m]	0.455	lzz	0.0114
Width_max[m]	0.03	lyy	0.0220
Width_min[m]	0.015	lxz	-0.0003
Hight_max[m]	0.04		
Hight_min[m]	0.015		
Center Line[m]	0.017		
Body aft[m]	0.44		
Aft Diameter[m]	0.012		

Body Configuration Parameter											
Station #	0	1	2	3	4	5	6	7	8	9	10
X	0.000	0.010	0.050	0.100	0.150	0.200	0.300	0.400	0.455	0.675	0.895
R	0.000	0.010	0.013	0.014	0.016	0.015	0.012	0.009	0.007	0.006	0.006
ZU	0.017	0.025	0.032	0.037	0.040	0.038	0.032	0.025	0.024	0.023	0.023
ZL	0.017	0.010	0.005	0.002	0.000	0.003	0.008	0.009	0.010	0.011	0.011

Wing	
Airfoil	S7012
Area[m^2]	0.22
x_leading edge[m]	0.22
z_leading edge[m]	0.04
Span[m]	1.50
Chord-root[m]	0.16
Chord-tip[m]	0.10
Sweep_LE[deg]	4.57
Incedence[deg]	0.00
Dihedral[deg]	2.00
m.g.c[m]	0.13
x_m.g.c[m]	0.25
y_m.g.c[m]	0.35

Aileron	
y_inboard[m]	0.38
y_outboard[m]	0.70
Inboard_chord[m]	0.03
Outboard_chord[m]	0.03

Horizontal Stabilizer	
Airfoil	NACA 0009
x_h_LE[m]	0.79
z_h_LE[m]	0.02
Span[m]	0.33
Chord-root[m]	0.10
Chord-tip[m]	0.07
Sweep_LE[deg]	11.58
Incedence[deg]	0.00
Area[m^2]	0.03

Vertical Stabilizer	
Airfoil	NACA 0009
x_h_LE[m]	0.88
z_h_LE[m]	-0.05
Span[m]	0.19
Chord-root[m]	0.10
Chord-tip[m]	0.08
Sweep_LE[deg]	6.01
Incedence[deg]	0.00
Area[m^2]	0.02

Elevator	
Chord[m]	0.03
Span[m]	Full

Rudder	
Chord[m]	0.04
Span[m]	Full

S7012	
x	y
1.000	0.000
0.985	0.002
0.941	0.008
0.872	0.017
0.783	0.028
0.680	0.039
0.568	0.050
0.454	0.059
0.345	0.063
0.245	0.062
0.159	0.055
0.089	0.043
0.038	0.027
0.008	0.011
0.000	0.001
0.012	-0.009
0.051	-0.017
0.114	-0.023
0.199	-0.025
0.301	-0.024
0.415	-0.021
0.533	-0.016
0.650	-0.010
0.762	-0.003
0.859	0.001
0.935	0.002
0.984	0.001
1.000	0.000

Appendix C DATCOM+ Input and Output

Input File

```
DIM M
DERIV RAD
DAMP
BUILD
*****
* Flight Conditions *
*****
$FLTCN LOOP=3.0,NMACH=1.0, MACH(1)=0.030,
      NALT=1.0, ALT(1)=500.0,NALPHA=10.0,
      ALSCHD(1)=-16.0,-12.0,-8.0,-4.0,0.0,4.0,8.0,12.0,16.0,20.0,
      STMACH=0.6, TSMACH=1.4, TR=1.0$
*****
* Reference Parameters *
*****
$OPTINS SREF=0.22, CBARR=0.132, BLREF=1.50, ROUGFC=0.02E-3$
*****
* Group II Synthesis Parameters *
*****
$SYNTHS XCG=0.285,ZCG=0.012,
      XW=0.223,ZW=0.038,ALIW=0.0,
      XH=0.791,ZH=0.023,ALIH=0.0,
      XV=0.875,ZV=-0.048,
      SCALE=1.0$
*****
* Body Configuration Parameters *
*****
$BODY NX=11.0,
      X(1)=0.000,0.010,0.050,0.100,0.150,0.200,0.300,0.400,0.455,0.675,0.895,
      R(1)=0.000,0.010,0.013,0.014,0.016,0.015,0.012,0.009,0.007,0.006,0.006,
      ZU(1)=0.017,0.025,0.032,0.037,0.040,0.038,0.032,0.025,0.024,0.023,0.023,
      ZL(1)=0.017,0.010,0.005,0.002,0.000,0.003,0.008,0.009,0.010,0.011,0.011,
      ITYPE=2.0, METHOD=1.0$
*****
* Wing planform variables*
*****
$WGPLNF CHRDR=0.16,CHRDTP=0.10,
      SSPN=0.75,SSPNE=0.735,
      SAVSI=4.57,CHSTAT=0.0,
      DHDADI=2.0,TWISTA=0.0,
      TYPE=1.0$
*****
* Wing Sectional Characteristics Parameters *
*****
$WGSCHR TOVC=0.087, DELTAY=2.67, XOVC=31.0, CLI=0.75, ALPHAI=4.5,
      CLALPA=0.0992, CLMAXL=0.909,CMO=-0.0465, LERI=0.05,
      CAMBER=.TRUE., YCM=2.0, DWASH=1.0, CLD=0.0, TYPEIN=1.0, NPTS=40.0,
      XCORD(1)=0.0000,0.0007,0.0049,0.0121,0.0222,0.0352,0.0510,0.0695,
      0.0907,0.1143,0.1403,0.1686,0.1990,0.2313,0.2654,0.3009,
      0.3378,0.3757,0.4145,0.4538,0.4935,0.5332,0.5726,0.6117,
      0.6504,0.6886,0.7258,0.7618,0.7963,0.8289,0.8594,0.8874,
      0.9128,0.9353,0.9547,0.9708,0.9835,0.9927,0.9982,1.0000,
      YUPPER(1)=0.0000,0.0054,0.0106,0.0161,0.0217,0.0273,0.0327,0.0380,
      0.0429,0.0474,0.0550,0.0580,0.0603,0.0621,0.0632,0.0636,
```

```

0.0634,0.0624,0.0608,0.0587,0.0562,0.0532,0.0500,0.0465,
0.0391,0.0352,0.0314,0.0276,0.0239,0.0203,0.0169,0.0137,
0.0108,0.0082,0.0059,0.0039,0.0023,0.0010,0.0002,0.0000,
YLOWER(1)=0.0000,-0.0028,-0.0060,-0.0092,-0.0122,-0.0150,-0.0174,
-0.0195,-0.0213,-0.0227,-0.0237,-0.0244,-0.0247,-0.0247,
-0.0245,-0.0240,-0.0232,-0.0222,-0.0210,-0.0197,-0.0181,
-0.0164,-0.0145,-0.0123,-0.0098,-0.0073,-0.0050,-0.0030,
-0.0013,0.0001,0.0011,0.0018,0.0021,0.0021,0.0019,
0.0015,0.0011,0.0006,0.0002,0.0000$
*****
*   Asymmetrical Control Deflection parameters : Ailerons   *
*****
$ASYFLP STYPE=4.0, NDELTA=9.0,
SPANFI=0.38, SPANFC=0.7,CHRDFI=0.032, CHRDFO=0.027,
DELTAL(1)=-20.0,-15.0,-10.0,-5.0,0.0,5.0,10.0,15.0,20.0,
DELTAR(1)=20.0,15.0,10.0,5.0,0.0,-5.0,-10.0,-15.0,-20.0$
CASEID AILERONS HYPER
SAVE
NEXT CASE
*****
*   Horizontal Tail Sectional Characteristics*
*****
NACA H 4 0009 ! Popular with glider
*****
*   Vertical Tail Sectional Characteristics *
*****
NACA V 4 0009
*****
*   Horizontal Tail planform variables *
*****
$HTPLNF CHRDR=0.104,CHRDTP=0.07,
SSPN=0.166,SSPNE=0.16,
SAVSI=11.58,CHSTAT=0.0,
TWISTA=0.0,DHDADI=0.0,
TYPE=1.0$
*****
*   Vertical Tail planform variables*
*****
$VTPLNF CHRDR=0.1,CHRDTP=0.08,
SSPN=0.19,SSPNE=0.19,
SAVSI=6.0,CHSTAT=0.0,
TYPE=1.0$
*****
*   Elevator Deflection parameters*
*****
$SYMFLP FTYPE=1.0, NDELTA=7.0,
DELTA(1)=-30.0,-20.0,-10.0,0.0,10.0,20.0,30.0,
PHETE=0.05,PHETEP=0.05,
CHRDFI=0.03,CHRDFO=0.03,
SPANFI=0.006,SPANFC=0.166,
CB=0.0075,TC=0.0025,
NTYPE=1.0$
CASEID HYPER-DL 1.5M

```


Output File

AUTOMATED STABILITY AND CONTROL METHODS PER APRIL 1976 VERSION OF DATCOM
CHARACTERISTICS AT ANGLE OF ATTACK AND IN SIDESLIP
WING-BODY-VERTICAL TAIL-HORIZONTAL TAIL CONFIGURATION
TOTAL: HYPER-DL 1.5M

FLIGHT CONDITIONS						REFERENCE DIMENSIONS					
MACH NUMBER	ALTITUDE	VELOCITY	PRESSURE	TEMPERATURE	REYNOLDS NUMBER	REF. AREA	REFERENCE LONG.	LENGTH	MOMENT LAT.	REF. CENTER HORIZ	REF. CENTER VERT
	M	M/SEC	N/ M**2	DEG K	1/ M	M**2	M	M	M	M	M
.030	500.00	10.15	9.5461E+04	284.900	6.6500E+05	.220	.132	1.500	.285	.012	
DERIVATIVE (PER RADIANT)											
ALPHA	CD	CL	CM	CN	CA	XCP	CLA	CMA	CYB	CNB	CLB
-16.0	.093	-1.207	.3738	-1.186	-.243	-.315	4.381E+00	-4.049E-01	-1.508E-01	4.220E-02	-5.043E-03
-12.0	.057	-.874	.2793	-.867	-.126	-.322	5.029E+00	-1.321E+00			-1.328E-02
-8.0	.031	-.505	.1893	-.505	-.040	-.375	5.175E+00	-1.325E+00			-2.303E-02
-4.0	.018	-.151	.0943	-.152	.007	-.619	4.951E+00	-1.360E+00			-3.215E-02
.0	.017	.186	-.0006	.186	.017	-.003	5.005E+00	-1.405E+00			-4.063E-02
4.0	.029	.547	-.1019	.548	-.009	-.186	5.310E+00	-1.519E+00			-4.993E-02
8.0	.054	.927	-.2126	.926	-.076	-.230	5.045E+00	-1.750E+00			-5.998E-02
12.0	.088	1.252	-.3463	1.243	-.175	-.279	3.836E+00	-2.079E+00			-6.807E-02
16.0	.121	1.463	NA	1.440	-.287	NA	-3.325E+00	NA			-7.177E-02
20.0	.080	.788	NA	.767	-.194	NA	-1.603E+01	NA			-4.405E-02

ALPHA	Q/QINF	EPSLON	D(EPSLON)/D(ALPHA)
-16.0	1.000	-4.255	.297
-12.0	1.000	-3.067	.310
-8.0	1.000	-1.772	.322
-4.0	.990	-.491	.318
.0	.976	.770	.316
4.0	1.000	2.034	.318
8.0	1.000	3.312	.296
12.0	1.000	4.405	.205
16.0	1.000	4.954	-.153
20.0	1.000	3.180	-.444

*** NA PRINTED WHEN METHOD NOT APPLICABLE

AUTOMATED STABILITY AND CONTROL METHODS PER APRIL 1976 VERSION OF DATCOM
DYNAMIC DERIVATIVES
WING-BODY-VERTICAL TAIL-HORIZONTAL TAIL CONFIGURATION
TOTAL: HYPER-DL 1.5M

FLIGHT CONDITIONS						REFERENCE DIMENSIONS					
MACH NUMBER	ALTITUDE	VELOCITY	PRESSURE	TEMPERATURE	REYNOLDS NUMBER	REF. AREA	REFERENCE LONG.	LENGTH	MOMENT LAT.	REF. CENTER HORIZ	REF. CENTER VERT
	M	M/SEC	N/ M**2	DEG K	1/ M	M**2	M	M	M	M	M
.030	500.00	10.15	9.5461E+04	284.900	6.6500E+05	.220	.132	1.500	.285	.012	
DYNAMIC DERIVATIVES (PER RADIANT)											
PITCHING			ACCELERATION			ROLLING			YAWING		
ALPHA	CLQ	CMQ	CLAD	CMAD	CLP	CYP	CNP	CNR	CLR		
-16.00	5.170E+00	-1.303E+01	9.027E-01	-3.727E+00	-4.131E-01	-4.077E-02	1.251E-01	-6.135E-02	-2.725E-01		
-12.00			9.439E-01	-3.897E+00	-4.596E-01	-4.317E-02	8.986E-02	-5.239E-02	-1.952E-01		
-8.00			9.796E-01	-4.045E+00	-4.709E-01	-4.646E-02	5.136E-02	-4.588E-02	-1.085E-01		
-4.00			9.569E-01	-3.951E+00	-4.481E-01	-4.914E-02	1.516E-02	-4.336E-02	-2.603E-02		
.00			9.366E-01	-3.867E+00	-4.540E-01	-5.135E-02	-1.973E-02	-4.425E-02	5.255E-02		
4.00			9.666E-01	-3.991E+00	-4.838E-01	-5.417E-02	-5.672E-02	-4.877E-02	1.372E-01		
8.00			9.015E-01	-3.722E+00	-4.569E-01	-5.765E-02	-9.703E-02	-5.755E-02	2.264E-01		
12.00			6.242E-01	-2.577E+00	-3.351E-01	-5.994E-02	-1.357E-01	-6.785E-02	2.998E-01		
16.00			-4.660E-01	1.924E+00	3.675E-01	-2.215E-02	-1.379E-01	-7.470E-02	3.420E-01		
20.00			-1.349E+00	5.571E+00	1.618E+00	-2.893E-02	-1.671E-01	-4.729E-02	1.516E-01		

AUTOMATED STABILITY AND CONTROL METHODS PER APRIL 1976 VERSION OF DATCOM
CHARACTERISTICS OF HIGH LIFT AND CONTROL DEVICES
TAIL PLAIN TRAILING-EDGE FLAP CONFIGURATION

TOTAL: HYPER-DL 1.5M

FLIGHT CONDITIONS						REFERENCE DIMENSIONS				
MACH NUMBER	ALTITUDE	VELOCITY	PRESSURE	TEMPERATURE	REYNOLDS NUMBER	REF. AREA	REFERENCE LONG.	LENGTH LAT.	MOMENT HORIZ	REF. CENTER VERT
	M	M/SEC	N/ M**2	DEG K	1/ M	M**2	M	M	M	M
.030	500.00	10.15	4.1640E+01	923.077	6.1780E+04	.220	.132	1.500	.285	.012

-----INCREMENTS DUE TO DEFLECTION-----

---DERIVATIVES (PER DEGREE)---

DELTA	D (CL)	D (CM)	D (CL MAX)	D (CD MIN)	(CLA) D	(CH) A	(CH) D
-30.0	-.055	.2922	.077	.01206	NDM	-1.463E-03	-6.291E-03
-20.0	-.048	.2517	.058	.00530	NDM		-5.982E-03
-10.0	-.031	.1615	.033	.00172	NDM		-5.590E-03
.0	.000	-.0002	.000	.00000	NDM		-5.590E-03
10.0	.031	-.1615	.033	.00172	NDM		-5.590E-03
20.0	.048	-.2517	.058	.00530	NDM		-5.982E-03
30.0	.055	-.2922	.077	.01206	NDM		-6.291E-03

----- INDUCED DRAG COEFFICIENT INCREMENT , D(CDI) , DUE TO DEFLECTION -----

DELTA =	-30.0	-20.0	-10.0	.0	10.0	20.0	30.0
ALPHA							

-16.0	9.79E-03	8.14E-03	4.72E-03	-3.71E-06	-2.71E-03	-3.33E-03	-3.42E-03
-12.0	7.80E-03	6.41E-03	3.60E-03	-2.59E-06	-1.58E-03	-1.59E-03	-1.42E-03
-8.0	5.87E-03	4.74E-03	2.52E-03	-1.51E-06	-5.01E-04	7.44E-05	5.05E-04
-4.0	3.94E-03	3.06E-03	1.43E-03	-4.22E-07	5.84E-04	1.75E-03	2.44E-03
.0	2.00E-03	1.37E-03	3.37E-04	6.72E-07	1.68E-03	3.44E-03	4.38E-03
4.0	5.27E-05	-3.18E-04	-7.55E-04	1.76E-06	2.77E-03	5.13E-03	6.33E-03
8.0	-1.88E-03	-2.00E-03	-1.84E-03	2.85E-06	3.86E-03	6.81E-03	8.26E-03
12.0	-3.95E-03	-3.79E-03	-3.00E-03	4.01E-06	5.02E-03	8.61E-03	1.03E-02
16.0	-6.40E-03	-5.92E-03	-4.38E-03	5.39E-06	6.40E-03	1.07E-02	1.28E-02
20.0	-1.05E-02	-9.49E-03	-6.69E-03	7.69E-06	8.70E-03	1.43E-02	1.69E-02

-----YAWING MOMENT COEFFICIENT,CN,DUE TO CONTROL DEFLECTION-----

(DELTAL-DELTAR) =	-40.0	-30.0	-20.0	-10.0	.0	10.0	20.0	30.0	40.0
ALPHA									

-16.0	-1.008E-02	-8.714E-03	-5.946E-03	-2.973E-03	0.000E+00	2.973E-03	5.946E-03	8.714E-03	1.008E-02
-12.0	-7.264E-03	-6.281E-03	-4.286E-03	-2.143E-03	0.000E+00	2.143E-03	4.286E-03	6.281E-03	7.264E-03
-8.0	-4.113E-03	-3.557E-03	-2.427E-03	-1.214E-03	0.000E+00	1.214E-03	2.427E-03	3.557E-03	4.113E-03
-4.0	-1.105E-03	-9.552E-04	-6.518E-04	-3.259E-04	0.000E+00	3.259E-04	6.518E-04	9.552E-04	1.105E-03
.0	1.769E-03	1.529E-03	1.044E-03	5.218E-04	0.000E+00	-5.218E-04	-1.044E-03	-1.529E-03	-1.769E-03
4.0	4.859E-03	4.202E-03	2.867E-03	1.434E-03	0.000E+00	-1.434E-03	-2.867E-03	-4.202E-03	-4.859E-03
8.0	8.106E-03	7.009E-03	4.783E-03	2.392E-03	0.000E+00	-2.392E-03	-4.783E-03	-7.009E-03	-8.106E-03
12.0	1.080E-02	9.336E-03	6.371E-03	3.185E-03	0.000E+00	-3.185E-03	-6.371E-03	-9.336E-03	-1.080E-02
16.0	1.238E-02	1.070E-02	7.304E-03	3.652E-03	0.000E+00	-3.652E-03	-7.304E-03	-1.070E-02	-1.238E-02
20.0	5.745E-03	4.967E-03	3.390E-03	1.695E-03	0.000E+00	-1.695E-03	-3.390E-03	-4.967E-03	-5.745E-03

DELTAL	DELTAR	(CL) ROLL
-20.0	20.0	-6.9727E-02
-15.0	15.0	-6.0311E-02
-10.0	10.0	-4.1156E-02
-5.0	5.0	-2.0578E-02
.0	.0	0.0000E+00
5.0	-5.0	2.0578E-02
10.0	-10.0	4.1156E-02
15.0	-15.0	6.0311E-02
20.0	-20.0	6.9727E-02

Appendix D AVL Input and Output

Input File

```

HYPER
0.030                      Mach
0      0      0      iYsym  iZsym  Zsym
0.220  0.132  1.500  Sref   Cref   Bref
0.285  0.000  0.012  Xref   Yref   Zref
#=====
BODY
Fuse pod
16  1.0
TRANSLATE
0.000  0.000  0.017
BFIL
fuselage.dat
#=====
SURFACE
WING
#Nchord  Cspace  Nspan  Sspace
8        1.0    32     -2.0
YDUPLICATE
0.0
ANGLE
0.0
SCALE
1.0  1.0  1.0
TRANSLATE
0.223  0.000  0.038      | dX  dY  dZ
#=====
SECTION
#      Xle      Yle      Zle      chord      angle  Nspan  Sspace
      0.000      0.000      0.000      0.160      0.000   19    -2.0
AFIL
S7012.dat
#=====
SECTION
#      Xle      Yle      Zle      chord      angle  Nspan  Sspace
      0.030      0.380      0.013      0.130      0.000   10    -2.0
AFIL
S7012.dat
CONTROL
#Cname  Cgain  Xhinge  HingeVec  SgnDup
aileron  1.0   0.755   0 -1 0    -1.0
#=====
SECTION
#      Xle      Yle      Zle      chord      angle  Nspan  Sspace
      0.060      0.750      0.026      0.100      0.000   1     -2.0
AFIL
S7012.dat
CONTROL
#Cname  Cgain  Xhinge  HingeVec  SgnDup
aileron  1.000  0.730   0 -1 0    -1.0
#=====
SURFACE
Horizontal tail

```

```

#Nchord  Cspace  Nspan  Sspace
8         1.0     16     -2.0
ANGLE
0.000
YDUPLICATE
0.000
TRANSLATE
0.791     0.0     0.023
=====
SECTION
#  Xle      Yle      Zle      chord    angle    Nspan  Sspace
0.000      0.000    0.000    0.104    0.000    15     -2.0
NACA
0009
CONTROL
#Cname      Cgain    Xhinge      HingeVec    SgnDup
elevator    1.000    0.711      0 1 0      1.0
=====
SECTION
#  Xle      Yle      Zle      chord    angle    Nspan  Sspace
0.033      0.166    0.000    0.070    0.000    1      -2.0
NACA
0009
CONTROL
#Cname      Cgain    Xhinge      HingeVec    SgnDup
elevator    1.000    0.571      0 1 0      1.0
=====
SURFACE
Vertical tail
#Nchord  Cspace  Nspan  Sspace
8         1.0     20     1.0
TRANSLATE
0.875     0.000    -0.048
=====
SECTION
#  Xle      Yle      Zle      chord    angle    Nspan  Sspace
0.000      0.000    0.000    0.100    0.000    19     1.0
NACA
0009
CONTROL
#Cname      Cgain    Xhinge      HingeVec    SgnDup
rudder      1.0     0.60      0 0 -1      1.0
=====
SECTION
#  Xle      Yle      Zle      chord    angle    Nspan  Sspace
0.020      0.000    0.190    0.080    0.000    1      1.0
NACA
0009
CONTROL
#Cname      Cgain    Xhinge      HingeVec    SgnDup
rudder      1.0     0.50      0 0 -1      1.0
=====

```

```

-----
Run case 1: 10 meter per sec
alpha      -> alpha      = 0.00000
beta       -> beta       = 0.00000
pb/2V      -> pb/2V      = 0.00000
qc/2V      -> qc/2V      = 0.00000
rb/2V      -> rb/2V      = 0.00000
aileron    -> aileron    = 0.00000
elevators  -> elevators  = 0.00000
rudder     -> rudder     = 0.00000
alpha      = 0.00000      deg
beta       = 0.00000      deg
pb/2V      = 0.00000
qc/2V      = 0.00000
rb/2V      = 0.00000
CL         = 0.00000
CDo        = 0.01700      from DATCOM
bank       = 0.00000      deg
elevation  = 0.00000      deg
heading    = 0.00000      deg
Mach       = 0.03000
velocity   = 8.0000      Lunit/Tunit
density    = 1.16730      Munit/Lunit^3
grav.acc.  = 9.81000      Lunit/Tunit^2
turn_rad.  = 0.00000      Lunit
load_fac.  = 1.00000
X_cg       = 0.28500      Lunit
Y_cg       = 0.00000      Lunit
Z_cg       = 0.01200      Lunit
mass       = 0.23000      kg
Ixx        = 0.01600      kg-m^2
Iyy        = 0.01140      kg-m^2
Izz        = 0.02200      kg-m^2
Ixy        = 0.00000      kg-m^2
Iyz        = 0.00000      kg-m^2
Izx        = -0.00030      kg-m^2
visc CL_a  = 0.00000
visc CL_u  = 0.00000
visc CM_a  = 0.00000
visc CM_u  = 0.00000

```

Output File

```
<aerodynamics>
  <Coefficient name = CYb>
    -2.1057E-001
    -2.1490E-001
    -2.1826E-001
    -2.2064E-001
    -2.2203E-001
    -2.2242E-001
    -2.2181E-001
    -2.2020E-001
    -2.1760E-001
    -2.1403E-001
  </Coefficient>
  <Coefficient name = CYp>
    -2.5659E-001
    -2.1016E-001
    -1.5931E-001
    -1.0502E-001
    -4.8363E-002
    9.5700E-003
    6.7648E-002
    1.2474E-001
    1.7974E-001
    2.3157E-001
  </Coefficient>
  <Coefficient name = CYr>
    8.2848E-002
    1.1787E-001
    1.4609E-001
    1.6695E-001
    1.8006E-001
    1.8515E-001
    1.8213E-001
    1.7106E-001
    1.5215E-001
    1.2577E-001
  </Coefficient>
  <Coefficient name = CYdr>
    1.4479E-001
    1.5057E-001
    1.5493E-001
    1.5785E-001
    1.5928E-001
    1.5911E-001
    1.5739E-001
    1.5418E-001
    1.4954E-001
    1.4347E-001
  </Coefficient>
  <Coefficient name = Cldr>
    4.4691E-003
    4.2972E-003
    4.0680E-003
```

```

3.8388E-003
3.4950E-003
3.1513E-003
2.8075E-003
2.4064E-003
2.0054E-003
1.6043E-003
</Coefficient>
<Coefficient name = Cnb>
9.7954E-002
9.0041E-002
8.4185E-002
8.0500E-002
7.9057E-002
7.9885E-002
8.2967E-002
8.8244E-002
9.5613E-002
1.0493E-001
</Coefficient>
<Coefficient name = Cnr>
1.1681E-001
8.4064E-002
5.1529E-002
1.9035E-002
-1.3597E-002
-4.6541E-002
-7.9955E-002
-1.1396E-001
-1.4864E-001
-1.8402E-001
</Coefficient>
<Coefficient name = Cnr>
-7.4855E-002
-7.3125E-002
-7.2529E-002
-7.3165E-002
-7.5057E-002
-7.8150E-002
-8.2318E-002
-8.7362E-002
-9.3026E-002
-9.9005E-002
</Coefficient>
<Coefficient name = Cndr>
-6.1937E-002
-6.4400E-002
-6.6291E-002
-6.7552E-002
-6.8125E-002
-6.8067E-002
-6.7323E-002
-6.5947E-002
-6.3942E-002
-6.1364E-002
</Coefficient>

```

Appendix E Updraft Model Parameter Inputs

```

<Updraft>
  <reference wstar = 2.56
            zi = 1400.0
            A = 1080000.0
            N = 3
            zGain = 4
            wGain = 4
            rGain = 4
            xDrift = 4
            yDrift = 2>
  </reference>
  <xt>
    177
    833
    1033
  </xt>
  <yt>
    127
    189
    711
  </yt>
  <zGain>
    0.0    60.0    240.0    300.0
    0.0     1.0     1.0     1.0
    1.5     1.5     1.2     1.0
    0.0     0.8     1.0     1.2
  </zGain>
  <rGain>
    0.0    60.0    240.0    300.0
    0.0     1.0     1.0     0.0
    2.0     1.4     1.5     0.5
    0.0     0.7     1.0     1.2
  </rGain>
  <wGain>
    0.0    60.0    240.0    300.0
    0.0     1.0     1.0     0.0
    1.5     1.3     1.1     0.6
    0.0     0.7     1.1     1.2
  </wGain>
  <xDrift>
    0.0    60.0    240.0    300.0
    0.0    12.0    25.0    35.0
    0.0    13.0   105.0   100.0
    0.0     0.0     0.0     0.0
  </xDrift>
  <yDrift>
    0.0    60.0
    0.0    18.0
    0.0   -108.0
    0.0     0.0
  </yDrift>
</Updraft>

```


Appendix F Pseudo Codes for Interpolation Methods

n is the number of rows in the table; t is the first column of the table; y is the second column of the table; z is the coefficients for the linear function; x is where the values want to be interpolated.

[illegible]

Natural Cubic Spline

n is the number of rows in the table; t is the first column of the table; y is the second column of the table; z is the coefficients for the spline function; x is where the values want to be interpolated.

```

#####S_#####( t, y, z)
#####
#####
#####t, z, z
#####h1:t-1, z1:t-1, z1:t-1
#####z= 1 z1:t-1- 1 z1:t-1
    ht← zt-1 - zt
    zt← (zt-1 - zt)/ht
#####
z2 ← 2(h1 + h2)
z2 ← 6(z2 - z1)
#####z= 3 z1:t-1- 1 z1:t-1
    zt← 2(ht + ht-1) - ht-12/zt-1
    zt← 6(zt - zt-1) - ht-1 zt-1/zt-1
#####
zt ← 0.0
#####z= zt- 1 z1:t-1
    zt← (zt - ht-1zt-1)/zt
#####
zt ← 0.0
#####h, z, z, z
#####S_#####
#####S_#####( t, y, z, z)
#####
#####h, z, z
#####t, z, z, z
#####z= zt- 1 z1:t-1
    zt - zt ≥ 0 #####
#####
h ← zt-1 - zt
zt← zt/2 + (zt - zt)(zt-1 - zt)/(6h)
zt← -(h/6)(zt-1 + 2zt) + (zt-1 - zt)/h + (zt - zt)(zt)
#####S_#####← zt+ (zt- zt)(zt)
#####S_#####

```

Bilinear Method

m is the number of rows of the two dimensional table; n is the number of columns; $t1$ is a vector for the dependent variable for different row; $t2$ is a vector for the dependent variable for the different columns; $x1$ is the value used to interpolate the table in $t1$; $x2$ is the value used to interpolate the table in $t2$.

```

#####
##### ( x2, x1, t1, t2, x1, x2 )
#####
##### x1, x2
##### t1_1: x1 t2_1: x2
##### t1, t2, x1 x2
##### = x2 - 1 t1_1 x2
      t1_1 - t1_m >= 0 #####
#####
##### = x1 - 1 t2_1 x1
      t2_1 - t2_n >= 0 #####
#####
x1 <- (t1 - t1_m) / (t1_{m+1} - t1_m)
x2 <- (t2 - t2_n) / (t2_{n+1} - t2_n)
##### <- (1 - x1)(1 - x2)(t1_{m+1}) + (x1)(1 - x2)(t1_{m+1,m})
      + (x1)(x2)(t1_{m+1,m+1}) + (1 - x1)(x2)(t1_{m+1})
#####

```

Point Fitting Method

m is the number of rows of the two dimensional table; n is the number of columns; $t1$ is a vector for the dependent variable for different row; $t2$ is a vector for the dependent variable for the different columns; $x1$ is the value used to interpolate the table in $t1$; $x2$ is the value used to interpolate the table in $t2$.

a_1, a_2, a_3 , and a_4 are the coefficients being calculated.

```

#####(??, ??, ??, ??, ??, ??, ??, ??)
#####(??, ??, ??)
#####1:??, ??1:??, ??2:??, ??3:??-1:??-1, ??4:??-1:??-1, ??5:??-1:??-1, ??6:??-1:??-1
#####4:4, ??1, ??1
#####= 1 ???- 1 ???
#####= 1 ???- 1 ???
?? ← [ 1 ?? ?? ?? ]
      [ 1 ??-1 ??-1 ??-1 ??-1 ]
      [ 1 ?? ??-1 ??-1 ??-1 ]
      [ 1 ??-1 ??-1 ??-1 ??-1 ]
?? ← [ ?? ]
      [ ??-1, ?? ]
      [ ??-1 ]
      [ ??-1, ??-1 ]
?? ← (??-1) ??
??_?? ← ??
??_?? ← ??
??_?? ← ??
??_?? ← ??

#####
#####
#####(??, ??, ??, ??, ??, ??, ??, ??)
#####(??, ??, ??)
#####1:??, ??1:??, ??2:??-1:??-1, ??3:??-1:??-1, ??4:??-1:??-1, ??5:??-1:??-1
#####??, ??
??? < ??_1 ??? < ??_1
??? < ??_1 ??? < ??_1
??? > ??_?? ??? < ??_??
??? > ??_?? ??? < ??_??
#####= ?? - 1 ???
??? - ??_?? ≥ 0
#####
#####= ?? - 1 ???
??? - ??_?? ≥ 0
#####
#####← ??_?? + ??_?? + ??_?? + ??_??
#####

```

Appendix G Transfer Function Routine

num is the numerator of the transfer function stored in a cell array; *den* is the denominator of the transfer function stored in a cell array; *m* is the number of elements in *num*; *n* is the number of element in *den*; *A*, *B*, *C*, and *D* are the state space matrices that are being calculated

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%( ??, ??, ??????, ?????, ??, ??, ??, ??)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%??, ??, ??
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%1:??, ?????:??,
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%????-1:??-1, ??-1:??-1, ??-1:??-1, ??1:1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%??= 2 *????
                ??1,??-1 ← -????/????
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% ??= 2 *????- 1
                ??1,??-1 ← 1.0
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                ??1,1 ← 1.0
                ???< 3*????
                ??← ??- (??+ 1)
                ?????= 1 *???
                ??,?? ← ?????/????
                ?????
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%??== 3*????
                ?????= 1 *????- 1
                ??,?? ← (????1,??-1 + ??,??) *????1/????
                ?????
                ??1,1 ← ??1,1 *????1/????
                ?????
                ?????
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```