Design of an Automatic Landing System for the Meridian UAV using Fuzzy Logic

BY

David Royer

Submitted to the graduate degree program in Aerospace Engineering and the Graduate Faculty of the University of Kansas in partial fulfillment of the requirements for the degree of Master of Science.

________________________________
Dr. Shahriar Keshmiri, Chairperson

Committee Members: ________________________________
Dr. David Downing

________________________________
Dr. Richard Hale

________________________________
Dr. Mark Ewing

Date Defended: April 20th, 2010

The Thesis Committee of David Royer certifies that this is
the approved version of the following thesis:

Design of an Automatic Landing System for the Meridian UAV using Fuzzy Logic

________________________________

Dr. Shahriar Keshmiri, Chairperson

Committee Members: ________________________________

Dr. David Downing

________________________________

Dr. Richard Hale

________________________________

Dr. Mark Ewing

Date Defended: April 20th, 2010

# Abstract

This document describes the design of an automatic landing system for the Meridian unmanned aerial vehicle (UAV) in development for glacial ice research. The Center for Remote Sensing of Ice Sheets (CReSIS), established by the National Science Foundation (NSF), will use the Meridian UAV to carry an ice penetrating radar system with eight flat plate antennas attached underneath the wings of the aircraft. The autolanding system designed in this thesis is meant to increase the autonomy of the Meridian to eventually reach a fully autonomous system.

A literature review of current research in automatic landing systems is presented. The Meridian UAV is modeled using *Advanced Aircraft Analysis* and *Athena Vortex Lattice* software; longitudinal and lateral state space models are developed and analyzed to evaluate the dynamic modes of the aircraft. A sensitivity analysis is performed evaluating the effect of changing $C_{l\beta}$, $C_{n\beta}$, and $C_{Y\beta}$ on the aircraft dynamics to investigate the effects of eight flat plate antennas attached below the wings. Fuzzy logic is determined the ideal application for an automatic landing controller based on its ability to control uncertain and nonlinear systems. Using fuzzy logic, a longitudinal automatic landing controller is designed which uses high level commands through the wePilot2000 for aircraft control. Simulation shows the controller is promising for further research and eventual implementation with the wePilot2000 flight control system.

## Acknowledgements

First and foremost, I would like to thank Dr. Shah Keshmiri for your support, motivation, and for constantly challenging me throughout the last two years. Your mentorship and friendship will not soon be forgotten. Thanks to Rylan Jager and Edmond Leong for getting this team started and paving the way for what we have done the last two years.

Thanks to Dr. Rick Hale for giving me the opportunity to work on such an incredible project. The experiences I have gained through this work will last a lifetime. Thank you to Dr. Downing for introducing me to the ‘magic’ of flight controls and inspiring me to do something like the work in this thesis.

Thanks to Bill Donovan, Jon Tom, Lance Holly, and Nick Brown for making such a stressful job enjoyable. Thank you to Andy Pritchard for brightening every day at the hangar, for your pearls of wisdom, and for your support over the last few years.

Thanks to all my friends in the KU aerospace department for making the last 6 years so memorable. I have thoroughly enjoyed working with a group of very bright and talented students/colleagues. I will sorely miss you all.

Finally, I want to express my deep love and appreciation for my family. Thanks to my parents for their constant support and encouragement. Thanks to my brother for your incredible friendship and for all your late-night/early-morning calls that kept me going through the hardest times. You guys mean the world to me.

# Table of Contents

## List of Figures

## List of Tables

## List of Symbols

| Symbol | Definition | Units |
| --- | --- | --- |
| $C_D$ | Drag coefficient | -- |
| $C_{D_\alpha}$ | Variation of airplane drag coefficient with angle of attack | 1/rad |
| $C_{D_u}$ | Variation of airplane drag coefficient with dimensionless speed | 1/rad |
| $C_L$ | Lift coefficient | -- |
| $C_{L_u}$ | Variation of airplane lift coefficient with dimensionless speed | 1/rad |
| $C_{L_\alpha}$ | Variation of airplane lift coefficient with angle of attack | 1/rad |
| $C_{L_{\dot{\alpha}}}$ | Variation of airplane lift coefficient with dimensionless rate of change of angle of attack | 1/rad |
| $C_{L_q}$ | Variation of airplane lift coefficient with dimensionless pitch rate | 1/rad |
| $C_l$ | Rolling moment coefficient | -- |
| $C_{l_\beta}$ | Variation of airplane rolling moment coefficient with angle of sideslip | 1/rad |
| $C_{l_p}$ | Variation of airplane rolling moment coefficient with dimensionless rate of change of roll rate | 1/rad |
| $C_{l_r}$ | Variation of airplane rolling moment coefficient with dimensionless rate of change of yaw rate | 1/rad |

## Symbols (cont.)

| | | |
|---|---|---|
| $C_m$ | Pitching moment coefficient | -- |
| $C_{m_u}$ | Variation of airplane pitching moment coefficient with dimensionless speed | 1/rad |
| $C_{m_\alpha}$ | Variation of airplane pitching moment coefficient with angle of attack | 1/rad |
| $C_{m_{\dot{\alpha}}}$ | Variation of airplane pitching moment coefficient with dimensionless rate of change of angle of attack | 1/rad |
| $C_{m_q}$ | Variation of airplane pitching moment coefficient with pitch rate | 1/rad |
| $C_{m_{T_u}}$ | Variation of airplane pitching moment coefficient due to thrust with dimensionless speed | 1/rad |
| $C_{m_{T_\alpha}}$ | Variation of airplane pitching moment coefficient due to thrust with angle of attack | 1/rad |
| $C_n$ | Yawing moment coefficient | -- |
| $C_{n_\beta}$ | Variation of airplane yawing moment coefficient with angle of sideslip | 1/rad |
| $C_{n_{T_\beta}}$ | Variation of airplane yawing moment coefficient due to thrust with angle of sideslip | 1/rad |
| $C_{n_p}$ | Variation of airplane yawing moment coefficient with dimensionless rate of change of roll rate | 1/rad |
| $C_{n_r}$ | Variation of airplane yawing moment coefficient with dimensionless rate of change of yaw rate | 1/rad |
| $C_T$ | Thrust coefficient | -- |

**Symbols (cont.)**

| | | |
|---|---|---|
| $C_{T_{X_1}}$ | Thrust coefficient in the X axis direction | -- |
| $C_{T_{X_u}}$ | Variation of airplane thrust coefficient in the X-axis direction w.r.t. dimensionless speed | -- |
| $C_y$ | Side force coefficient | -- |
| $C_{y_\beta}$ | Variation of airplane side force coefficient with angle of sideslip | 1/rad |
| $C_{y_p}$ | Variation of airplane side force coefficient with dimensionless rate of change of roll rate | 1/rad |
| $C_{y_r}$ | Variation of airplane side force coefficient with dimensionless rate of change of yaw rate | 1/rad |
| g | Acceleration due to gravity | $m/s^2$ |
| $i$ | Incidence angle | deg or rad |
| $I_{xx}$, $I_{yy}$, $I_{zz}$ | Airplane moments of inertia about X,Y,Z | slug $ft^2$ |
| $L_\beta$ | Roll angular acceleration per unit sideslip angle | $rad/sec^2/rad$ |
| $L_p$ | Roll angular acceleration per unit roll rate | 1/sec |
| $L_r$ | Roll angular acceleration per unit yaw rate | 1/sec |
| $L_{\delta_a}$ | Roll angular acceleration per unit aileron angle | $rad/sec^2/rad$ |
| $L_{\delta_r}$ | Roll angular acceleration per unit rudder angle | $rad/sec^2/rad$ |
| $L_{i_{vt}}$ | Roll angular acceleration per unit v-tail incidence angle | $rad/sec^2/rad$ |

**Symbols (cont.)**

| | | |
|---|---|---|
| $M_{\alpha}$ | Pitch angular acceleration per unit angle of attack | 1/sec$^2$ |
| $M_{\dot{\alpha}}$ | Pitch angular acceleration per unit rate of change of angle of attack | 1/sec |
| $M_{T_{\alpha}}$ | Pitch angular acceleration per unit angle of attack (due to thrust) | 1/sec$^2$ |
| $M_{u}$ | Pitch angular acceleration per unit change in speed | rad/sec/ft |
| $M_{T_{u}}$ | Pitch angular acceleration per unit change in speed (due to thrust) | rad/sec/ft |
| $M_{\delta_e}$ | Pitch angular acceleration per unit elevator angle | 1/sec$^2$ |
| $M_{i_{vt}}$ | Pitch angular acceleration per unit v-tail incidence angle | 1/sec$^2$ |
| $N_{\beta}$ | Yaw angular acceleration per unit sideslip angle | rad/sec$^2$/rad |
| $N_{T_{\beta}}$ | Yaw angular acceleration per unit sideslip angle (due to thrust) | rad/sec$^2$/rad |
| $N_{p}$ | Yaw angular acceleration per unit roll rate | 1/sec |
| $N_{r}$ | Yaw angular acceleration per unit yaw rate | 1/sec |
| $N_{\delta_a}$ | Yaw angular acceleration per unit aileron angle | rad/sec$^2$/rad |
| $N_{\delta_r}$ | Yaw angular acceleration per unit rudder angle | rad/sec$^2$/rad |
| $N_{i_{vt}}$ | Yaw angular acceleration per unit v-tail incidence angle | rad/sec$^2$/rad |
| $p,q,r$ | Perturbed values of P,Q and R | rad/s |

**Symbols (cont.)**

| | | |
|---|---|---|
| $P,Q,R$ | Airplane angular velocity components about XYZ | rad/s |
| $T_{2_s}$ | Time to double amplitude | s |
| $T_D$ | Disturbance transfer function | -- |
| $T_R$ | Tracking transfer function | -- |
| $T_{RL}$ | Lower limit tracking transfer function | -- |
| $T_{RU}$ | Upper limit tracking transfer function | -- |
| u,v,w | Perturbed states of U,V and W | m/s |
| U,V,W | Components of $V_\infty$ along X,Y,Z | m/s |
| $V_\infty$ | Airspeed | m/s |
| $V_{cruise}$ | Cruise Airspeed | kts |
| $V_{min/approach}$ | Minimum Airspeed/Approach Airspeed | kts |
| $V_x$ | Forward velocity | m/s |
| $V_z$ | Vertical velocity | m/s |
| $X_\alpha$ | Forward acceleration per unit angle of attack | $ft/sec^2/rad$ |
| $X_u$ | Forward acceleration per unit change in speed | 1/sec |
| $X_{T_u}$ | Forward acceleration per unit change in speed (due to thrust) | 1/sec |
| $X_{\delta_e}$ | Forward acceleration per unit elevator angle | $ft/sec^2/rad$ |
| $X_{i_{vt}}$ | Forward acceleration per unit v-tail incidence angle | $ft/sec^2/rad$ |
| $Y_\beta$ | Lateral acceleration per unit sideslip angle | $ft/sec^2/rad$ |

**Symbols (cont.)**

| | | |
|---|---|---|
| $Y_p$ | Lateral acceleration per unit roll rate | ft/sec/rad |
| $Y_r$ | Lateral acceleration per unit yaw rate | ft/sec/rad |
| $Y_{\delta_a}$ | Lateral acceleration per unit aileron angle | ft/sec$^2$/rad |
| $Y_{\delta_r}$ | Lateral acceleration per unit rudder angle | ft/sec$^2$/rad |
| $Y_{i_{vt}}$ | Lateral acceleration per unit v-tail incidence angle | ft/sec$^2$/rad |
| $Z_\alpha$ | Vertical acceleration per unit angle of attack | ft/sec$^2$/rad |
| $Z_{\dot{\alpha}}$ | Vertical acceleration per unit rate of change of angle of attack | ft/sec/rad |
| $Z_u$ | Vertical acceleration per unit change in speed | 1/sec |
| $Z_q$ | Vertical acceleration per unit pitch rate | ft/sec/rad |
| $Z_{\delta_e}$ | Vertical acceleration per unit angle of attack | ft/sec$^2$/rad |
| $Z_{i_{vt}}$ | Vertical acceleration per unit v-tail incidence angle | ft/sec$^2$/rad |

**Greek**

| | | |
|---|---|---|
| $\alpha$ | Angle of attack | deg or rad |
| $\beta$ | Angle of sideslip | deg or rad |
| $\delta$ | Control surface deflection | deg or rad |
| $\Delta$ | Increment of a parameter | -- |
| $\zeta$ | Damping ratio | -- |

**Greek (cont.)**

| | | |
|---|---|---|
| $\theta$ | Perturbed value of $\Theta$ | deg or rad |
| $\Theta$ | Airplane pitch angle | deg or rad |
| $\mu$ | Membership Function | -- |
| $\tau$ | Time constant | sec |
| $\phi$ | Perturbed value of $\Phi$ | deg or rad |
| $\Phi$ | Airplane bank angle | deg or rad |
| $\psi$ | Perturbed value of $\Psi$ | deg or rad |
| $\dot{\psi}$ | Course rate | deg or rad |
| $\Psi$ | Airplane heading angle | deg or rad |
| $\omega_n$ | Natural frequency | rad/s |

**Subscripts**

| | |
|---|---|
| a | aileron |
| agg | aggregate |
| cmd | command |
| d | desired |
| e | elevator, error |
| $\dot{e}$ | error rate |
| r | rudder |
| ss | steady state |
| TD | touchdown |
| vt | v-tail |

## Acronyms

| | |
|---|---|
| AAA | Advanced Aircraft Analysis |
| ACU | Auxiliary Control Unit |
| AVL | Athena Vortex Lattice |
| $BHP_r$ | Brake Horsepower required |
| $BHP_a$ | Brake Horsepower available |
| COTS | Commercial-Off-The-Shelf |
| CReSIS | Center for Remote Sensing of Ice Sheets |
| DATCOM | Data Compendium |
| DOF | Degrees of Freedom |
| FCS | Flight Control System |
| FIS | Fuzzy Inference System |
| FLC | Fuzzy Logic Controller |
| GPS | Global Positioning System |
| IAS | Indicated Airspeed |
| ILS | Instrument Landing System |
| IMU | Inertial Measurement Unit |
| KU | The University of Kansas |
| KUAE | University of Kansas Department of Aerospace Engineering |
| NED | North East Down |
| NSF | National Science Foundation |
| PC | Personal Computer |
| PCB | Printed Circuit Board |

**Acronyms (cont.)**

| | |
|---|---|
| PID | Proportional-Integral-Derivative |
| PWM | Pulse-Width Modulation |
| QFT | Quantitative Feedback Theory |
| RC | Remote Control |
| TF | Transfer Function |
| UAV | Unmanned Aerial Vehicle |

# 1 Introduction

The Center for Remote Sensing of Ice Sheets (CReSIS) at the University of Kansas was established by the National Science Foundation (NSF) in 2005, with the mission of developing advanced sensors to measure ice thickness, surface elevation, accumulation rate, and basal conditions in Greenland and Antarctica in support of global climate change research. The center has developed sensors and radar systems deployed on ground based vehicles and manned aircraft, and an initial goal of CReSIS was to develop an autonomous unmanned aerial vehicle (UAV) to carry ice penetrating radar. The University of Kansas Department of Aerospace Engineering (KUAE) has developed a UAV called the Meridian, shown in Figure 1.1, which will carry a CReSIS developed radar system which includes attaching eight flat plate antennas under the wings.


**Figure 1.1: The Meridian UAV**

The Meridian UAV is a 1,083 lb aircraft with a 26.4 ft wingspan. The Meridian will be controlled by the wePilot2000 flight control system, which is a semi-autonomous FCS for fixed wing UAVs which requires a pilot for takeoff and landing. This presents a major concern for Meridian flight operations. Though the pilots for the Meridian flight test operations have substantial experience with RC aircraft, the Meridian is much larger and heavier than a normal RC aircraft and will have significant differences in flight characteristics. In response to this concern, this thesis describes the design of an automatic landing system for the Meridian UAV the uses the wePilot flight control system.

Chapter 2 of this document summarizes a literature review of current research in automatic landing systems. Chapter 3 presents the modeling of the Meridian UAV and a sensitivity analysis of $C_{l\beta}$, $C_{n\beta}$, and $C_{Y\beta}$ which reveals potential for more difficult than expected flight qualities when landing the aircraft. This potential supports the need for an automatic landing system and particularly an autolanding controller that can handle uncertainty in the plant model. Chapter 4 describes the wePilot2000 Flight Control System (FCS) that was chosen for the Meridian UAV and a brief discussion of flight test of the FCS. Chapter 5 introduces fuzzy logic control theory and the fuzzy logic controller designed for the automatic landing system. Chapter 6 presents the design of a fuzzy logic automatic landing system for the Meridian UAV with simulation results. Chapter 7 presents conclusions and recommendations for future research.

## 2 Literature Review of Current Research in Autoland Systems

This section presents a literature review of current work in automatic landing systems for fixed wing aircraft. The main purpose for this literature review is to research different control theories that have been used for autolanding controllers and compare results. This comparison will aid the decision of what control theory will be used for the autoland system of the Meridian UAV.

The landing phase of flight is widely recognized as a difficult challenge for autonomous controller design. An autoland controller must be designed to reject external disturbances, such as wind and turbulence, as well as uncertainties in the plant model used for the design of the controller. Furthermore, the autoland system must consistently control an aircraft to an accurate touchdown point and a smooth touchdown to prevent damage to the aircraft.

The landing phase is divided into two phases: approach and flare to touchdown. The approach requires the controller to follow a constant glide slope and track the runway centerline. The flare requires the controller to follow an exponential curve which will guide the aircraft to a smooth touchdown. References [1] and [2] put particular emphasis on the necessity of modeling the ground effect during the flare maneuver for controller design as this introduces a large nonlinearity in the aircraft dynamic model between the approach and flare phases of landing. Both papers use [3] to calculate the change in zero-lift angle of attack and the airplane lift curve slope. Reference [4] takes this one step further and provides an overview of ground-effect identification as well as a process used by Airbus to validate an automatic landing system. Since the two phases of the landing process are present in every automatic landing system, the main focus of this literature review is the different controllers designed for autonomous landing. Four papers are discussed, each with a different control theory.

## 2.1 Automatic Landing Control Methods

### 2.1.1 Feedback Linearization Method

The goal of Reference [1] was to use the feedback linearization method for nonlinear control in the design of an automatic landing system. The paper writes, "To perform feedback linearization, the output should be differentiated until a component of the input, $\delta_e$ appears." Pages 10 through 12 present a derivation of this feedback linearization along with the steps taken in the control process. Unfortunately, the paper presents limited insight into the performance of simulations of this controller and no tests are performed outside of these simulations. In evaluating the controller performance in the presence of sensor noise, the paper uses three descriptions, excellent, good, and fair, without any mention of criteria for these ratings. Thus, the paper lacks good evidence for whether or not feedback linearization is a valuable method for control design.

On a separate note from the controller design, this paper states the following: "The longitudinal touchdown point is stipulated to be within a zone of 250m to 600m of the desired point and the lateral touchdown point should be such that the main landing gear is at a maximum distance of 19 m from the runway centre line." This means that the designers of this autoland system define an acceptable longitudinal error of 8%-20% (if the runway length is 10,000 ft).

### 2.1.2 Fuzzy Neural Networks and Genetic Algorithm

In Reference [5] a fuzzy neural network is designed to command pitch angle with inputs of altitude, altitude command, altitude rate, and altitude rate command. The paper assumes that the change in the throttle command is zero since the aircraft maintains a constant speed along the flight path. The results from simulations of this fuzzy neural network controller are presented as acceptable and the paper concludes that fuzzy controllers can successfully control a vehicle in severe wind disturbances. However, the touchdown conditions for this auto-landing controller include a pitch angle of approximately -10°. This landing condition is unacceptable considering in the case of the Meridian UAV, this pitch angle at touchdown would drive the nose into the ground and cause considerable damage to the aircraft including a broken propeller.

### 2.1.3 Fuzzy Logic Controller

This paper, Reference [6] uses a Fuzzy Logic Controller (FLC) to control the glide path error (via altitude control), for pitch stabilization, and to maintain constant speed. The fuzzy controller uses the error (from desired path) and the time-rate-of-change of the error (from desired path) to determine the control input, *u*. The FLC also uses scaling factors for input variables, the error and the error rate to affect the controller response.

For testing, the aircraft was placed in initial conditions meant to take the vehicle off of the desired path. The simulations show the FLC guides the vehicle back on course and holds the desired path quite well. The ultimate benefit of this controller is its simple design, as shown in block diagram implementation, and its ability to account for nonlinearity.

### 2.1.4 Quantitative Feedback Theory

Reference [2] uses Quantitative Feedback Theory (QFT) to design a digital autoland controller. QFT is a step-by-step process to create robust gains for a controller. The process, as described in the paper, is as follows:

1. Determine the set of plants ***P*** that cover the range of structured parameter uncertainty as well as plant templates for each frequency of interest.
2. Specify acceptable tracking models that the closed-loop response satisfies, $T_{RL} \leq T_R \leq T_{RU}$, and determine tracking bounds.
3. Determine disturbance rejection models $T_D$ based on disturbance rejection specifications, and determine disturbance bounds.
4. Specify stability margin and determine U contours.
5. Draw U contours, disturbance bounds, and tracking bounds on a Nichols chart.
6. Synthesize nominal loop transfer function $L_0(z)$
7. Synthesize prefilter F(z)
8. Simulate system to evaluate performance.
9. Iterate as required to meet specifications.

Based on Monte Carlo simulation, the performance of the QFT controller matched that of classic PID gains for all cases. However, the QFT controller performance far exceeds classic PID gains for the case with model uncertainty and in moderate turbulence. The paper concludes that the QFT controller offers better overall safety, performance, and robustness for the case of an airplane with model uncertainties flying in turbulent conditions.

## 2.2 Conclusion

The automatic landing system to be designed by the author will be designed specifically for control through the wePilot2000 flight control system. The wePilot limits the control inputs to outer-loop controls: $V_x$, $V_z$, and $\dot{\psi}$. Based upon the reviewed literature and this control limitation, the author concludes that a fuzzy logic controller will be the best for the approach and flare portion of the autoland system that will be designed. This conclusion is based on the simplicity that the fuzzy logic controller presents to the problem of not having direct control over aircraft attitude. The fuzzy logic controller design needs to be tested for robustness in the presence of both model uncertainty and external disturbances. However, considering the controller of Ref. [6] also performed well with a nonlinear aircraft model, the author hypothesizes that a fuzzy logic controller will prove robust and ultimately a good choice for operation with the wePilot.

# 3 Modeling and Sensitivity Analysis of the Meridian UAV

The design phase of the Meridian UAV did not involve wind tunnel testing or any high-fidelity CFD analysis. Based on preliminary research conducted by KUAE, the AAA and similar conventional parametric methods either underestimate or overestimate aerodynamic derivatives of UAVs due to the low Reynolds numbers at which UAVs operate [8]. Consequently, the eight flat plate antennas introduce a large uncertainty to the aircraft dynamic model. The antennas are assumed to have the largest affect $C_{l\beta}$ and $C_{n\beta}$, which are calculated with an estimated uncertainty of 20% and 15% respectively ([14]). By varying the value of these two stability derivatives, the change in flying qualities of the vehicle is studied to estimate whether or not the antennas harbor the possibility of drastically changing the vehicle's dynamics. Special attention is given to an already unstable spiral mode.

## 3.1 Modeling with Advanced Aircraft Analysis (AAA)

*Advanced Aircraft Analysis* (AAA) is a software program developed by *DAR Corporation* [9] primarily for preliminary aircraft design. *AAA* estimates stability and control derivatives of an airplane using geometric parameters and basic aerodynamic characteristics. For a given flight condition and airplane configuration, *AAA* uses the input geometric and aerodynamic characteristics to interpolate through an internal aerodynamic database of different size aircraft, mostly based on DATCOM[10] and figures from the Roskam Airplane Design series [11]. This process results in a low-fidelity estimation of stability and control derivatives produces a very quick, useful analysis of airplane stability dynamic flight qualities for preliminary design.

### 3.1.1 Physical Geometry

The geometric parameters of the Meridian are shown in Table 3.1. When modeling an aircraft with a V-tail, such as the Meridian UAV, *AAA* calculates the projected horizontal and vertical tails. This information is also given in Table 3.1, as a second *AAA* file was created with the projected horizontal and vertical tails to double check calculations in *AAA*. Figure 3.1 presents a three-view of the Meridian UAV.

**Table 3.1: Basic Geometry of the Meridian UAV**

| | Span (ft) | Reference Area ($ft^2$) | Aspect Ratio | Taper Ratio | Incidence Angle (deg) | Dihedral Angle (deg) | Quarter Chord Sweep Angle (deg) |
|---|---|---|---|---|---|---|---|
| **Wing** | 26.38 | 69.61 | 10.0 | 1.0 | 0.0 | 5.0 | 0.0 |
| **V-Tail** | 4.64 | 15.69 | 5.5 | 0.5 | 0.0 | 50.0 | 17.6 |
| **Projected Horizontal Tail** | 5.92 | 10.00 | 3.5 | 0.5 | 0.0 | 0.0 | 26.3 |
| **Projected Vertical Tail** | 4.99 | 11.92 | 2.1 | 0.5 | 0.0 | 0.0 | 22.5 |

### 3.1.2 Engine Model

*AAA* allows for engine data to be modeled with the airplane, which inherently improves longitudinal mode dynamics estimation. Previous work by the authors [12] presents a performance evaluation of the engine that will fly on the Meridian. Table 3.2 presents data for the required brake horse power to fly the Meridian at a number of airspeeds. The information in this table was used in the *AAA* model per flight condition to calculate a large set of derivatives with varying airspeeds.

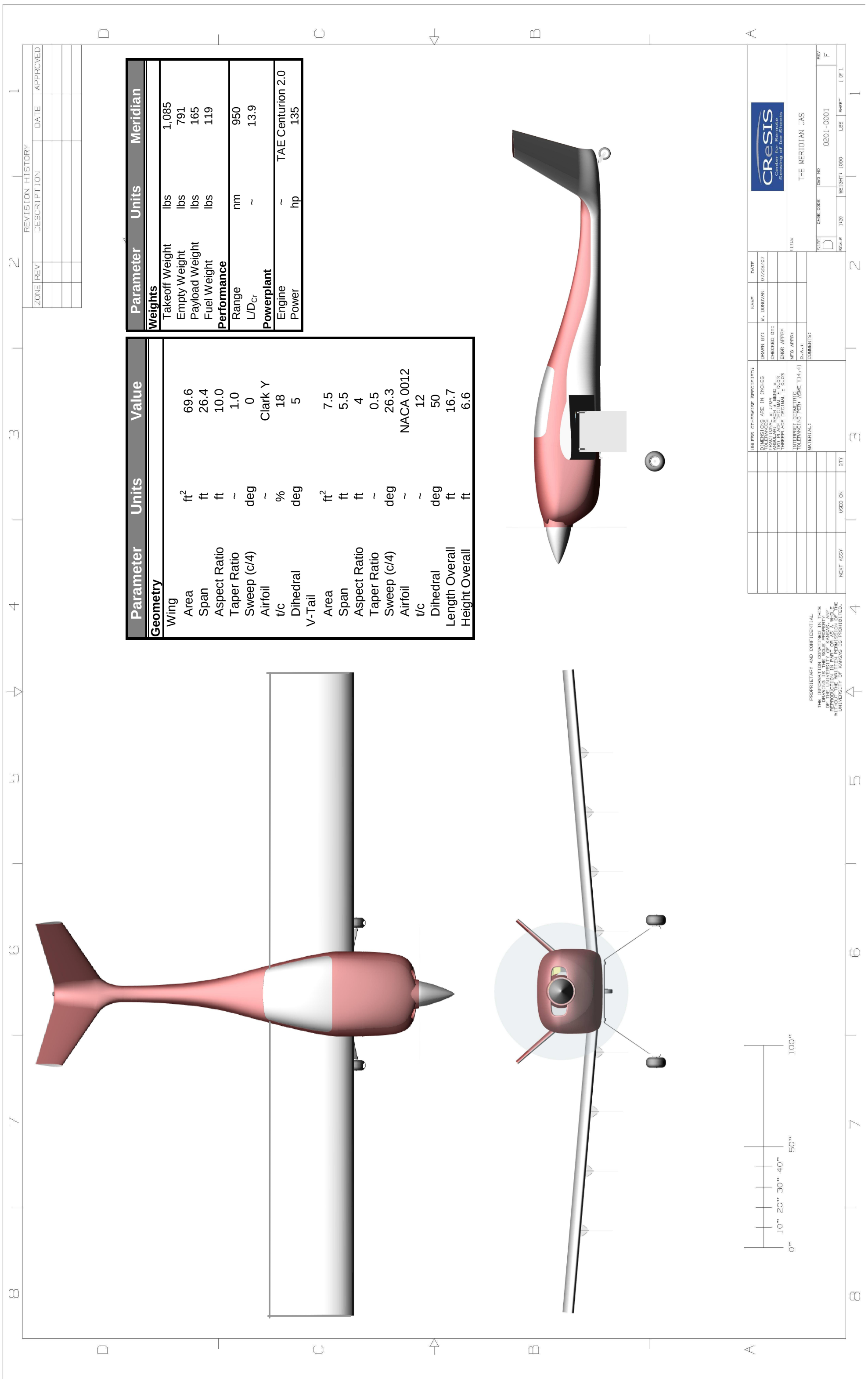



**Figure 3.1: Three-View of Meridian UAV**

**Table 3.2: Required BHP for Airspeed**

| $V_\infty$ (kts) | $BHP_r$ | $BHP_a$ | $\Delta BHP$ |
|---|---|---|---|
| 60 | 20 | 135 | 115 |
| 70 | 20 | 135 | 115 |
| 80 | 22 | 135 | 113 |
| 90 | 25 | 135 | 110 |
| 100 | 30 | 135 | 105 |
| 110 | 36 | 135 | 99 |
| 120 | 44 | 135 | 91 |
| 130 | 53 | 135 | 82 |
| 140 | 64 | 135 | 71 |
| 150 | 77 | 135 | 58 |
| 160 | 91 | 135 | 44 |

### 3.1.3 Stability and Control Derivatives

The geometry from Table 3.1 and the engine model from Table 3.2 are exported to *AAA* as required along with other minor geometric characteristics. The trim speed is set to 120 knots at Standard Sea Level conditions in steady level flight. The resulting derivatives that were calculated in *AAA* are presented in Table 3.3 and Table 3.4.

**Table 3.3: AAA Steady State Coefficients for Meridian UAV**

| Steady State Coefficients | |
|---|---|
| $C_{L_1}$ | 0.3214 |
| $C_{D_1}$ | 0.0232 |
| $C_{T_{X_1}}$ | 0.0238 |
| $C_{m_1}$ | 0.0097 |
| $C_{m_{T_1}}$ | $-0.0098$ |

**Table 3.4: AAA Dimensionless Stability & Control**

| Longitudinal Derivatives $(rad^{-1})$ | | Lateral-Directional Derivatives $(rad^{-1})$ | |
|---|---|---|---|
| $C_{D_u}$ | $0.0000$ | $C_{y_\beta}$ | $-0.4789$ |
| $C_{D_\alpha}$ | $0.1409$ | $C_{y_p}$ | $-0.1465$ |
| $C_{T_{X_u}}$ | $-0.0713$ | $C_{y_r}$ | $0.3217$ |
| $C_{L_u}$ | $0.0109$ | $C_{l_\beta}$ | $-0.0776$ |
| $C_{L_\alpha}$ | $5.1648$ | $C_{l_p}$ | $-0.5546$ |
| $C_{L_{\dot{\alpha}}}$ | $0.7407$ | $C_{l_r}$ | $0.1099$ |
| $C_{L_q}$ | $4.6179$ | $C_{n_\beta}$ | $0.1386$ |
| $C_{m_u}$ | $0.0028$ | $C_{n_{T_\beta}}$ | $-0.0007$ |
| $C_{m_\alpha}$ | $-0.6207$ | $C_{n_p}$ | $-0.0351$ |
| $C_{m_{\dot{\alpha}}}$ | $-2.9826$ | $C_{n_r}$ | $-0.1338$ |
| $C_{m_q}$ | $-13.973$ | $C_{l_{\delta_a}}$ | $0.2316$ |
| $C_{m_{T_u}}$ | $0.0294$ | $C_{y_{\delta_a}}$ | $0.0000$ |
| $C_{m_{T_\alpha}}$ | $-0.3419$ | $C_{n_{\delta_a}}$ | $-0.0134$ |
| $C_{D_{i_{vt}}}$ | $0.0117$ | $C_{y_{i_{vt}}}$ | $-0.3681$ |
| $C_{L_{i_{vt}}}$ | $0.4149$ | $C_{l_{i_{vt}}}$ | $-0.0253$ |
| $C_{m_{i_{vt}}}$ | $-1.6709$ | $C_{n_{i_{vt}}}$ | $0.1481$ |

These stability and control derivatives, along with derivatives corresponding to a wide range of airspeeds, were calculated for use as a look-up table in a 6-DOF nonlinear simulator [8] for pilot training and for use in preparing flight test plans.

## 3.2 Modeling with Athena Vortex Lattice (AVL)

The Athena Vortex Lattice (AVL) [13] software was also used to develop stability and control derivatives. AVL is a vortex lattice code developed at the Massachusetts Institute of Technology by Mark Drela and Harold Youngren. This code is a low-fidelity modeling program that estimates aerodynamic stability and control derivatives by solving for zero vorticity in the flow over the aircraft lifting surfaces. The code can also support fuselages by modeling them as slender bodies, but this was assumed unnecessary for modeling the Meridian. The AVL model for the Meridian uses the geometry from Table

3.1 and was used to compare AVL stability and control derivatives to AAA derivatives. AVL ignores the effects of the engine power. For this reason, AAA is expected to be more accurate in estimating longitudinal dynamics. Figure 3.2 shows the 3-D visualization of the AVL model of the Meridian.

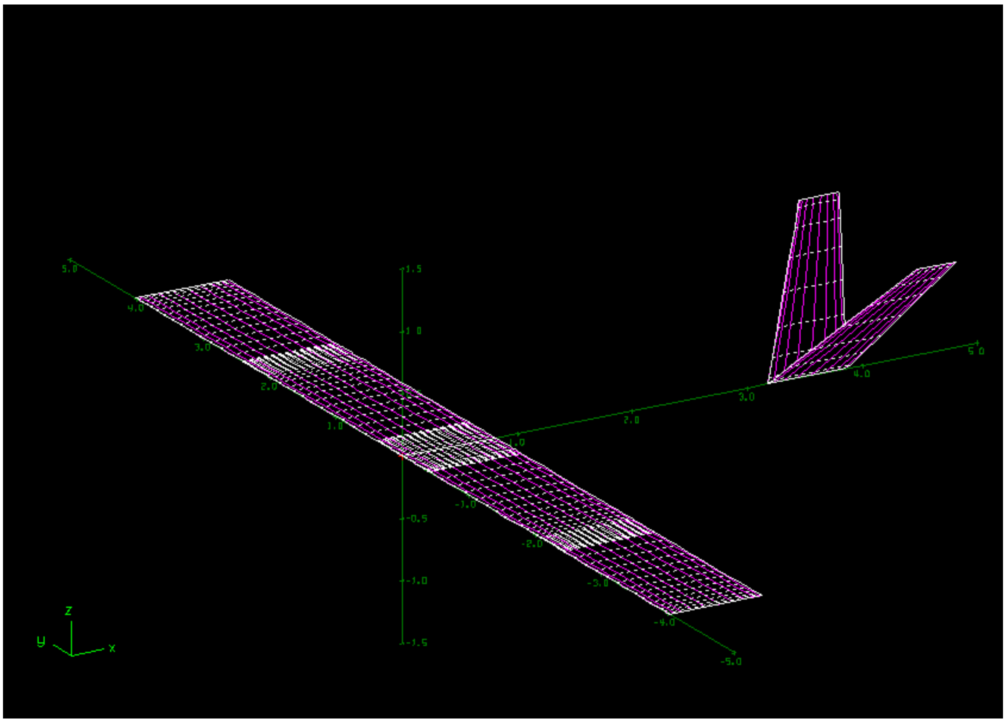


**Figure 3.2: AVL Model of the Meridian UAV**

Table 3.5and Table 3.6 summarize the results from the AVL model stability and control derivative estimation compared to the AAA results.

**Table 3.5: Comparison of AAA and AVL Longitudinal Stability Derivatives**

| Stability Derivative ($rad^{-1}$) | AAA | AVL | % Difference |
|---|---|---|---|
| $C_{L_\alpha}$ | 5.1648 | 4.596 | $-11.01$ |
| $C_{L_q}$ | 4.6179 | 6.2079 | 34.43 |
| $C_{m_\alpha}$ | $-0.6207$ | $-0.4858$ | $-21.73$ |
| $C_{m_q}$ | $-13.973$ | $-11.334$ | $-18.88$ |
| $C_{L_{i_{ht}}}$ | 0.4149 | 0.2392 | $-42.35$ |
| $C_{m_{i_{ht}}}$ | $-1.6709$ | $-0.9244$ | $-44.68$ |

Table 3.6: Comparison of AAA and AVL Lateral-Directional Stability Derivatives

| Stability Derivative $(rad^{-1})$ | AAA | AVL | % Difference |
|---|---|---|---|
| $C_{y_\beta}$ | $-0.4789$ | $-0.3431$ | 28.36 |
| $C_{y_p}$ | $-0.1465$ | $-0.01473$ | 90.24 |
| $C_{y_r}$ | 0.3217 | 0.2923 | 9.14 |
| $C_{l_\beta}$ | $-0.0776$ | $-0.0501$ | 35.44 |
| $C_{l_p}$ | $-0.5546$ | $-0.5826$ | $-5.05$ |
| $C_{l_r}$ | 0.1099 | 0.1109 | $-0.91$ |
| $C_{n_\beta}$ | 0.1386 | 0.1366 | 1.44 |
| $C_{n_p}$ | $-0.0351$ | $-0.0174$ | 50.43 |
| $C_{n_r}$ | $-0.1338$ | $-0.1228$ | 8.22 |
| $C_{l_{\delta_a}}$ | 0.2316 | 0.164 | 29.19 |
| $C_{y_{i_{vt}}}$ | 0.3681 | 0.2053 | 44.23 |
| $C_{l_{i_{vt}}}$ | 0.0253 | 0.0236 | 6.72 |
| $C_{n_{i_{vt}}}$ | $-0.1481$ | $-0.0833$ | 43.75 |

These results show that the AVL estimated stability derivatives are, in most cases, within the expected prediction error for AAA stability derivative estimation, as given in [14]. The difference in the control derivative estimation is most likely due to the difference of methods for modeling the v-tail between the two modeling methods, where AAA estimates derivatives based on the projected horizontal and vertical tails and AVL estimates derivatives based on the v-tail as modeled.

## 3.3 Meridian Dynamics

Using the AAA results and stability and control derivatives shown in Table 3.3 and Table 3.4, and the methods described in [14], the standard 6-DOF equations of motion for the longitudinal dynamics of a fixed wing aircraft are rewritten in state space form. The linearized, longitudinal and lateral-directional state space models are used for analysis of the dynamic modes of the Meridian UAV

### 3.3.1 Longitudinal State Space Model

The perturbed longitudinal equations of motion are written as [14]:

$$\begin{aligned}
\dot{u} &= -g\theta\cos\theta_1 + X_u u + X_{T_u} u + X_\alpha \alpha + X_{\delta_e}\delta_e \\
U_1\dot{\alpha} - U_1\dot{\theta} &= -g\theta\sin\theta_1 + Z_u u + Z_\alpha\alpha + Z_{\dot{\alpha}}\dot{\alpha} + Z_q\dot{\theta} + Z_{\delta_e}\delta_e \\
\ddot{\theta} &= M_u u + M_{T_u} u + M_\alpha\alpha + M_{\dot{\alpha}}\dot{\alpha} + M_q\dot{\theta} + M_{\delta_e}\delta_e
\end{aligned} \tag{3.1}$$

assuming steady level flight, $\dot{\theta} = q$ and $\ddot{\theta} = \dot{q}$, Eq(1.1) is rewritten in state space form:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & (U_1 - Z_{\dot{\alpha}}) & 0 & 0 \\ 0 & -M_{\dot{\alpha}} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} (X_u + X_{T_u}) & X_\alpha & 0 & -g\cos\theta_1 \\ Z_u & Z_\alpha & (U_1 + Z_q) & -g\sin\theta_1 \\ (M_u + M_{T_u}) & (M_\alpha + M_{T_\alpha}) & M_q & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ \alpha \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} X_{i_{vt}} \\ Z_{i_{vt}} \\ M_{i_{vt}} \\ 0 \end{bmatrix} [i_{vt}] \tag{3.2}$$

Equation (3.3) presents the longitudinal state space model for the Meridian, where the dimensional stability and control derivatives are calculated from the dimensionless derivatives of Table 3.4. In the standard LTI model, $\dot{x} = Ax + Bu$.

$$\begin{bmatrix} \dot{u} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -0.0347 & 18.0827 & 0 & -32.17 \\ -0.0016 & -2.5588 & 0.9828 & 0.0010 \\ 0.0057 & -27.2435 & -3.2861 & -0.0006 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ \alpha \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} -1.1738 \\ -0.2047 \\ -49.747 \\ 0 \end{bmatrix} [i_{vt}] \tag{3.3}$$

### 3.3.2 Lateral State Space Model

The perturbed lateral-directional equations of motion are written as [14]:

$$\begin{aligned}
&U_1\dot{\beta} - U_1\dot{\psi} = g\phi\cos\theta_1 + Y_\beta\beta + Y_p\dot{\phi} + Y_r\dot{\psi} + Y_{\delta_a}\delta_a + Y_{\delta_r}\delta_r \\
&\ddot{\phi} - \bar{A}_1\ddot{\psi} = L_\beta\beta + L_p\dot{\phi} + L_r\dot{\psi} + L_{\delta_a}\delta_a + L_{\delta_r}\delta_r \\
&\ddot{\psi} - \bar{B}_1\ddot{\phi} = N_\beta\beta + N_{T_\beta}\beta + N_p\dot{\phi} + N_r\dot{\psi} + N_{\delta_a}\delta_a + N_{\delta_r}\delta_r \\
&\text{where} \quad \bar{A}_1 = \frac{I_{xz}}{I_{xx}} \quad \text{and} \quad \bar{B}_1 = \frac{I_{xz}}{I_{zz}}
\end{aligned} \tag{3.4}$$

assuming steady level flight, $p = \dot{\phi}$ and $\dot{p} = \ddot{\phi}$, Eq(1.4) is rewritten in state space form:

$$\begin{bmatrix} 1 & 0 & 0 & -\bar{A}_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & U_1 & 0 \\ -\bar{B}_1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{\phi} \\ \dot{\beta} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} L_p & 0 & L_\beta & L_r \\ 1 & 0 & 0 & 0 \\ Y_p & g\cos\theta_1 & Y_\beta & (Y_r - U_1) \\ N_p & 0 & (N_\beta + N_{T_\beta}) & N_r \end{bmatrix} \begin{bmatrix} p \\ \phi \\ \beta \\ r \end{bmatrix} + \begin{bmatrix} L_{\delta_a} & L_{i_{vt}} \\ 0 & 0 \\ Y_{\delta_a} & Y_{i_{vt}} \\ N_{\delta_a} & N_{i_{vt}} \end{bmatrix} \begin{bmatrix} \delta_a \\ i_{vt} \end{bmatrix} \tag{3.5}$$

Equation (3.6) presents the lateral state space model for the Meridian. This is calculated using the same method for the longitudinal state space model. In the standard LTI model, $\dot{x} = Ax + Bu$.

$$\begin{bmatrix} \dot{p} \\ \dot{\phi} \\ \dot{\beta} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -6.3927 & 0 & -13.6841 & 1.2629 \\ 1 & 0 & 0 & 0 \\ -0.0047 & 0.1589 & -0.2366 & -0.9896 \\ -0.2846 & 0 & 16.2954 & -1.0295 \end{bmatrix} \begin{bmatrix} p \\ \phi \\ \beta \\ r \end{bmatrix} + \begin{bmatrix} 40.9715 & -4.4110 \\ 0 & 0 \\ 0 & -0.1820 \\ -1.4977 & 17.5244 \end{bmatrix} \begin{bmatrix} \delta_\alpha \\ i_{vt} \end{bmatrix} \tag{3.6}$$

### 3.3.3 Dynamic Analysis and Flying Qualities

An analysis of the eigenvalues of the dynamics matrices in Eq.(3.3) and (3.6) results in the dynamic characteristics of the Meridian shown in the $V_{cruise}$ column of Table 3.7. This table also shows the dynamic characteristics of the Meridian in the landing condition with the approach airspeed of 70 kts and a full 40 degree flap deflection.

**Table 3.7: Dynamic Mode Analysis from AAA Model**

| Mode | | $V_{cruise}$ | $V_{min/approach}$ |
|---|---|---|---|
| Phugoid | $\zeta$ | 0.048 | 0.084 |
| | $\omega_n$ (rad/s) | 0.162 | 0.386 |
| Short Period | $\zeta$ | 0.345 | 0.462 |
| | $\omega_n$ (rad/s) | 3.72 | 3.80 |
| Spiral | $\tau$ (s) | -110 | -54.6 |
| | $T_{2s}$ (s) | 76.3 | 54.6 |
| Dutch Roll | $\zeta$ | 0.083 | 0.132 |
| | $\omega_n$ (rad/s) | 2.76 | 2.6 |
| Roll | $\tau$ (s) | 0.34 | 0.26 |

The dynamic analysis of the Meridian reveals an unsteady spiral mode. The effect of the unstable spiral mode is summarized with the calculation of the *Time to Double Amplitude* shown in Eq(3.7) from [15]. This dynamic characteristic of the aircraft is very important as it describes how quickly the spiral mode will diverge, or more practically how quickly the bank angle will double in amplitude. In the case of the Meridian, this value will have a major impact on takeoff and landing the aircraft because it will be remotely piloted for these phases of flight. If the time to double amplitude is too small, the pilot will have to devote most of his focus to keeping the aircraft level and may have an incredibly difficult time landing the aircraft.

$$T_{2_s} = \frac{\ln 2}{\frac{1}{\tau_s}} \tag{3.7}$$

Requirements on handling qualities for the spiral mode, based on MIL-F-8785C standards, call for a value of $T_{2s}$ greater than 12 seconds for Level 1 (adequate handling qualities) and greater than 8 seconds for Level 2 (increased pilot workload, degradation in mission effectiveness). These handling qualities are meant for a piloted aircraft and are not necessarily adequate for a third-person RC pilot, specifically on an airplane as large as the Meridian.

## 3.4 Sensitivity Analysis of $C_{l\beta}$ and $C_{n\beta}$

### 3.4.1 Effect of Flat Plate Antennas on $C_{l\beta}$ and $C_{n\beta}$

The stability derivative $C_{l\beta}$ is often referred to as the *dihedral effect* [16], due to the large influence of wing geometric dihedral angle on this stability derivative, where a negative value is stabilizing. The eight flat plate antennas under the wing of the Meridian are below the aircraft center of gravity. Thus, a positive sideslip angle will produce a positive rolling moment, resulting in the relationship of Eq(3.8). This means that the antennas will have a de-stabilizing effect on the aircraft dynamics.

$$\left(C_{l_\beta}\right)_{Antennas} > 0 \tag{3.8}$$

The stability derivative $C_{n\beta}$ is referred to as the *directional stability effect*, where a positive value is stabilizing. The flat plate antennas on the Meridian will produce a positive yawing moment with a positive sideslip angle based on their position relative to the aircraft center of gravity. Equation (3.9) reflects this relationship, which means the antennas should have a stabilizing effect on the aircraft dynamics.

$$\left(C_{n_\beta}\right)_{Antennas} > 0 \tag{3.9}$$

The addition of the eight flat plate antennas to the aircraft will also affect the stability derivative $C_{Y\beta}$ which is an important derivative in dutch-roll dynamics and in flight path control. The flat plate antennas will produce a negative side-force due to sideslip, as defined in Eq(3.10).

$$\left(C_{Y_\beta}\right)_{Antennas} < 0 \tag{3.10}$$

### 3.4.2 Results

The values for $C_{n\beta}$ and $C_{l\beta}$ are varied up to 100 percent according to the relationships in Eq. (3.9) and (3.10) (increasing in a positive manner) to investigate the change in aircraft dynamics with the antennas installed. From this analysis, the Spiral mode time constant is the aircraft dynamic mode most affected by these two stability derivatives. The Phugoid, Short Period, and Roll modes were not changed in response to the derivative variation and the Dutch-Roll mode was only slightly affected with a decrease in Dutch-Roll damping. As mentioned earlier, the time to double amplitude, a characteristic of the spiral mode, will be very important for flight operations of the Meridian. It is therefore reasonable to make the time to double amplitude the principle aircraft flight quality of interest for a sensitivity analysis of $C_{n\beta}$ and $C_{l\beta}$.

Figure 3.3 shows the change of this variable versus the change of $C_{n\beta}$. As seen in this figure, a positive change in $C_{n\beta}$ actually exhibits a de-stabilizing effect on the aircraft dynamics, contrary to previous discussion. However, even at 100 percent positive change in this derivative, the spiral mode meets Level 1 standards (MIL-F-8785C).

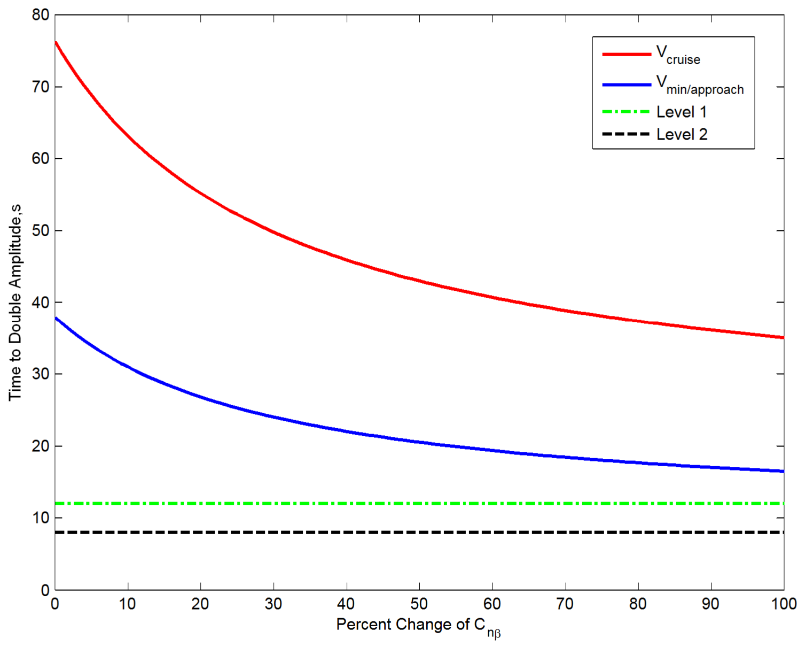



**Figure 3.3: Sensitivity Analysis of $C_{n\beta}$**

Figure 3.4 shows the change in time to double amplitude versus the change in $C_{l\beta}$. This analysis shows that the positive change in this stability derivative worsened instability in the spiral mode as expected. At 82 percent change in this derivative, the time to double amplitude no longer meets Level 1 standards but meets Level 2 standards up to 100 percent change and remains greater than 10 seconds.

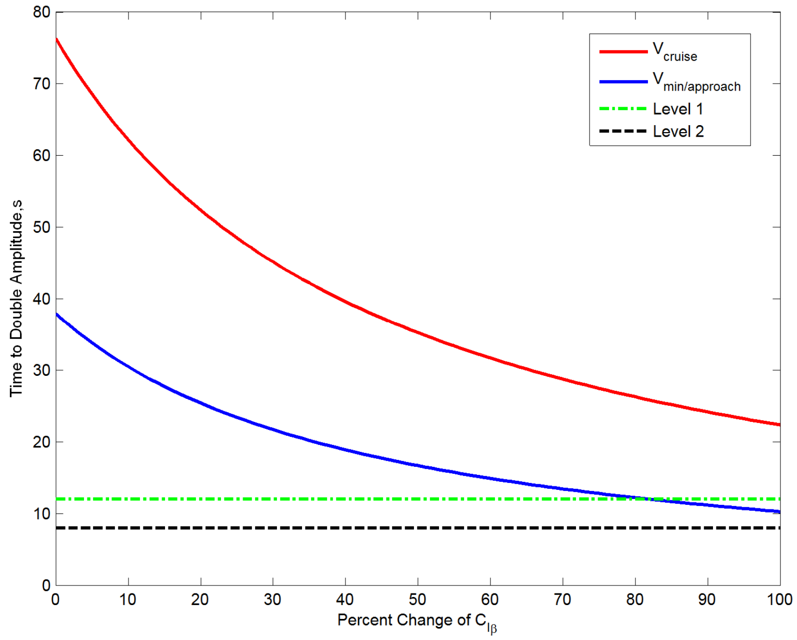



**Figure 3.4: Sensitivity Analysis of $C_{l\beta}$**

Figure 3.5 presents the results of changing both $C_{n\beta}$ and $C_{l\beta}$ simultaneously, which shows a change in time to double amplitude similar to a change in only $C_{l\beta}$. In this case, the time to double amplitude meets Level 1 standards up to a 68 percent change in both derivatives and meets Level 2 standards up to 100 percent change and remains greater than 10 seconds.

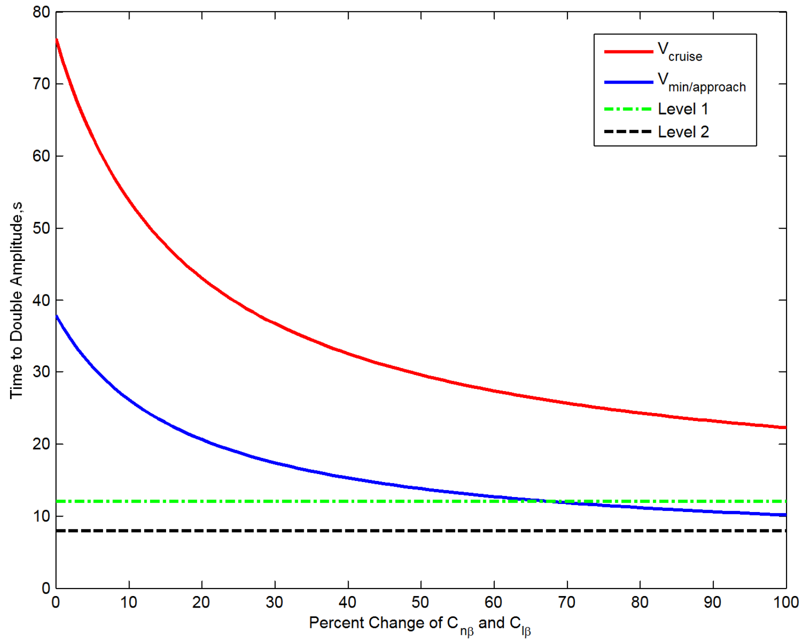



**Figure 3.5: Sensitivity Analysis of $C_{n\beta}$ and $C_{l\beta}$**

Figure 3.6 presents the results of an exhaustive Monte Carlo calculation of the time to double amplitude with varying $C_{n\beta}$ and $C_{l\beta}$. This calculation also included varying $C_{Y\beta}$, but this derivative's effect on $T_{2s}$ was negligible.

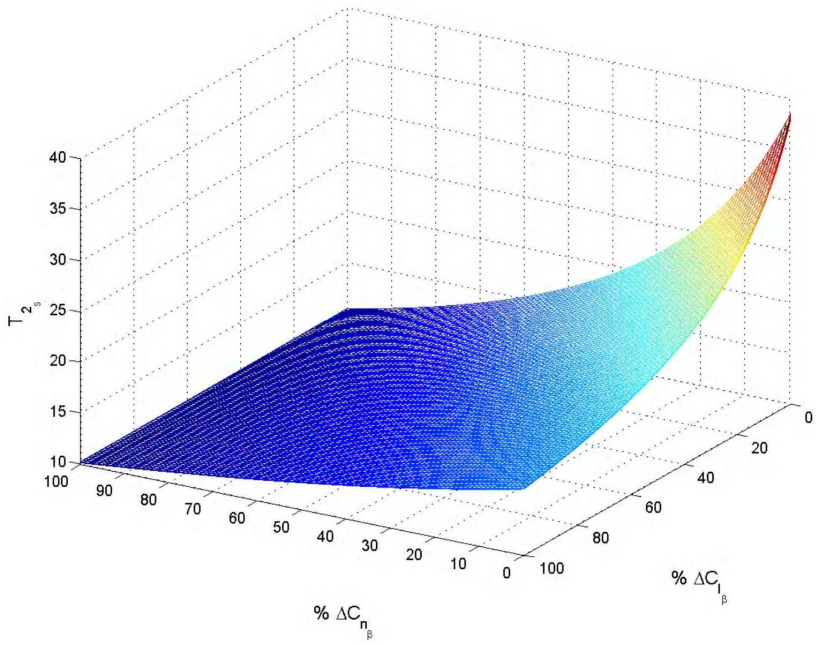



**Figure 3.6: Monte Carlo Calculation of $T_{2s}$**

Figure 3.7 shows the exhaustive Monte Carlo calculation of the dutch-roll damping ratio with varying $C_{n\beta}$, $C_{l\beta}$, and $C_{Y\beta}$. The combined effect of increasing $C_{n\beta}$ and $C_{l\beta}$ is shown to reduce Dutch Roll damping by up to ~40% whereas a 100% change in $C_{Y\beta}$ increases dutch-roll damping by ~20%.

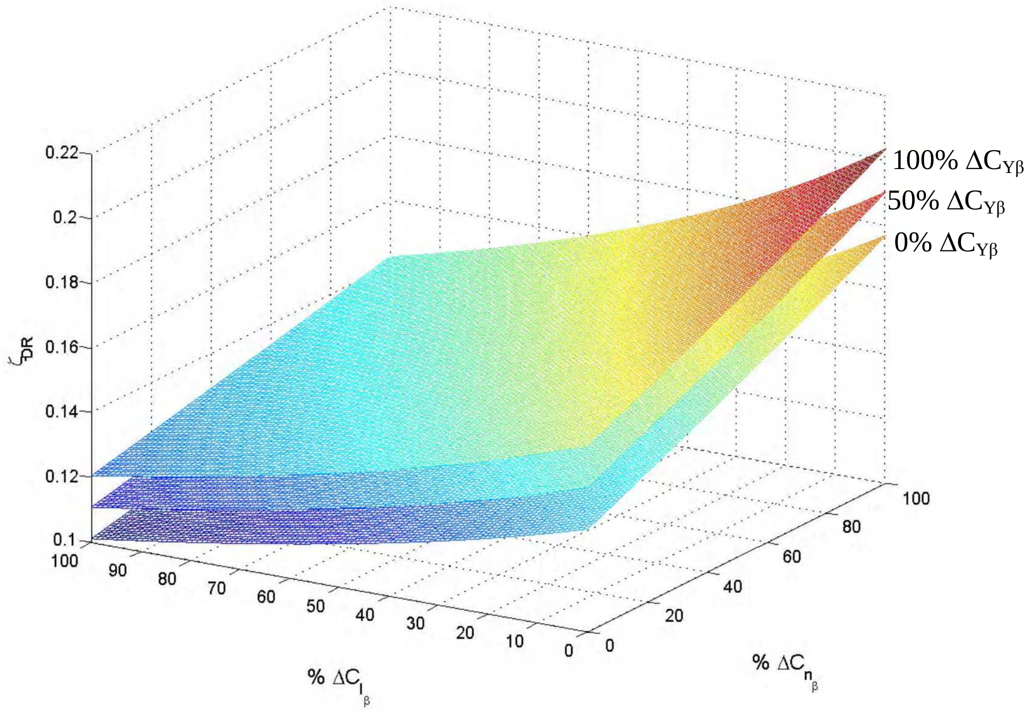



**Figure 3.7: Sensitivity of Dutch Roll Damping to $C_{l\beta}$ $C_{n\beta}$ and $C_{Y\beta}$**

This sensitivity analysis shows that the uncertainty in the aircraft model and the addition of eight flat plate antennas under the wings of the Meridian could have adverse effects on the handling qualities of the Meridian UAV. This sensitivity analysis shows that the Meridian pilot may experience difficult handling qualities on landing and supports the design of an automatic landing system. The sensitivity analysis also suggests an autolanding system for the Meridian should use a controller which can handle the plant model uncertainty and different aircraft dynamics with or without the antennas attached.

# 4 The wePilot2000 Fixed Wing Autonomous Flight Control System

The first task for the CReSIS/KU Flight Test Team was to secure a commercial-off-the-shelf (COTS) autopilot for future use on the Meridian UAV. After flight test of the Cloud Cap Piccolo II Autopilot System, [18], the team determined that this autopilot was not fit for use on an aircraft the size of the Meridian UAV. After this decision, testing began with the weControl wePilot2000 Flight Control System (FCS). The wePilot2000 is a semi-autonomous flight control system which includes onboard hardware, shown in Figure 4.1, and ground station software for mission control and aircraft monitoring. At the time of this report, the flight test team has completed 36 flights with the wePilot2000 on a one-third scale Yak-54 test bed aircraft and five successful flights onboard the Meridian UAV.


**Figure 4.1: wePilot2000 FCS, from weControl [17]**

The basic wePilot2000 system architecture is presented in Figure 4.2

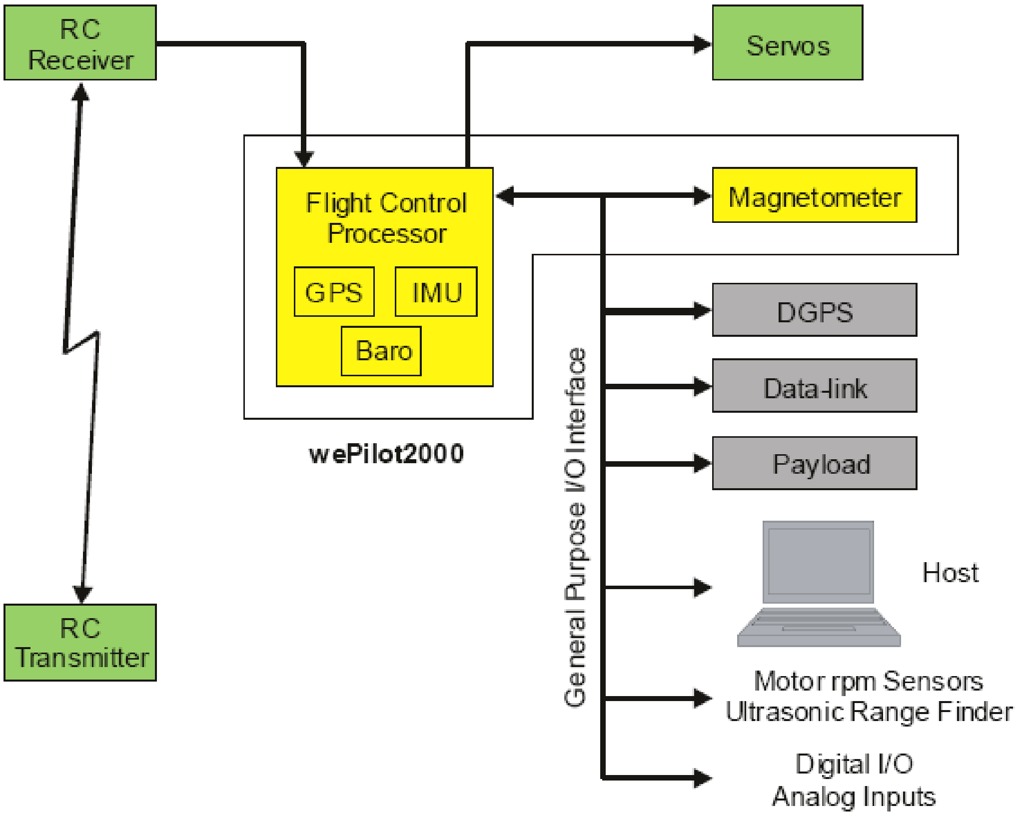



**Figure 4.2: wePilot2000 System Architecture [17]**

## 4.1 wePilot2000 Avionics Hardware

The wePilot2000 is a fixed wing FCS integrated onto one PCB which runs on an Intel XScale PXA255 32-bit RISC Processor. Position and groundspeed data is provided by a NovAtel GPS receiver, updated at 4 Hz. Barometric pressure is measured using a 10-1100mbar piezoresistive pressure sensor. The PCB has an integrated six-degree-of-freedom inertial measurement unit (IMU) with three gyroscopes (±100 deg/s) and 3 accelerometers (±10 g) which are used for state estimation with an extended Kalman filter. The wePilot has eight PWM input channels for control input from the RC transmitter and six PWM output channels for servo control. The FCS also has an interface for an external magnetometer. Reference [17] sums up the entirety of the avionics hardware included on the wePilot PCB. Figure 4.3 shows the wePilot PCB with integrated avionics hardware encased in the machined aluminum case as it was delivered by Viking Aerospace.

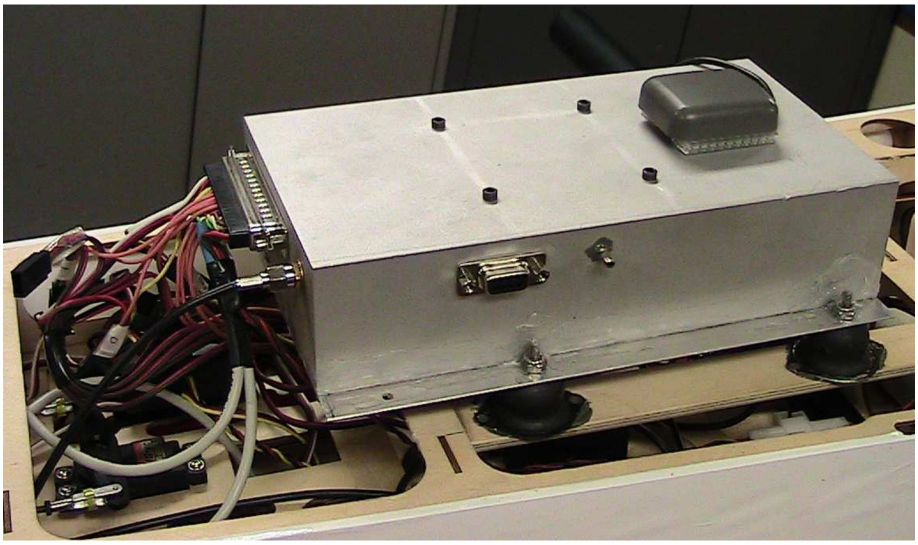


**Figure 4.3: wePilot in Aluminum Casing**

### 4.2 Communications

Communication with the wePilot involves both the ground station and the remote control (RC) transmitter. The RC controller can use either the standard 72 MHz or 2.4 GHz receiver to connect to the wePilot. For ground station communications, the wePilot uses an external RF modem at either 900 MHz or 2.4 GHz. The current flight test team uses the Digi XTend-PKG 900 MHz RF modem for the Meridian UAV and for the one-third scale Yak-54 test bed. This communication system transfers telemetry data and mission commands between the FCS and the ground station. Data can be updated at rates up to 8 Hz but has been set at 4 Hz for current flight test operations.

## 4.3 Ground Station

The wePilot2000 ground station software displays critical health data including airspeed, ground speed, altitude, attitudes, FCS battery health, distance from the ground station, GPS location. The software transcribes the GPS location onto a calibrated aerial map of the location of flight for navigation purposes. Through this program, mission plans are uploaded and high level mission commands are sent to the autopilot. Mission plans are created in the software and can be uploaded to the autopilot at any time while the system is in the 'Ready' state.

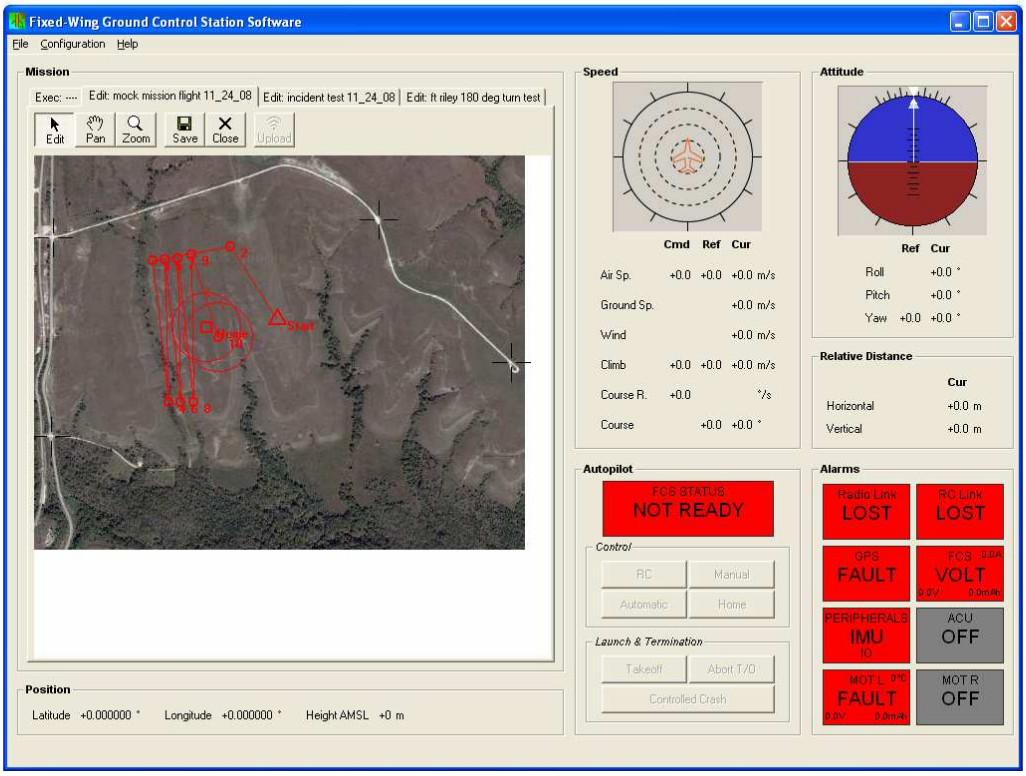



**Figure 4.4: wePilot Ground Station Software**

## 4.4 wePilot2000 Guidance, Navigation, and Control

The wePilot GNC is broken down into guidance and navigation, inner loop control, and outer loop control.

### 4.4.1 Guidance & Navigation

In autonomous mission flight, the wePilot tracks waypoints defined by latitude, longitude, altitude, and desired airspeed. During flight between waypoints, the wePilot tracks the flight path defined by a straight line between two waypoints and it uses an algorithm to reduce the error from the flight path exponentially. The autopilot is designed to fly over the next waypoint in a mission and then turn toward the next flight path. This means that the aircraft does not perform a pre-turn maneuver and mission plans need to be designed taking this into consideration. Mission plans can also include circle waypoints which has the aircraft fly an orbit for a defined number of times around the circle at a defined orbit radius.

### 4.4.2 Inner Loop Controls

Inner loop controls on the wePilot are designed using $H_\infty$ robust control methods. The controller is synthesized based on user-provided stability and control derivatives for the aircraft including the estimated uncertainties in the aircraft dynamics model. The $H_\infty$ controller maintains stability through a broad range of uncertainties in the plant model, which reduces the need for gain-tuning flights prior to autonomous flight. This also means the controller is robust enough to handle high wind conditions, as seen from experience gained by the CReSIS flight test team in flying the one-third scale Yak-54 in winds of up to 15-20 knots (see Ref. [20]).

### 4.4.3 Outer Loop Controls

The wePilot operates in one of three modes:

- **RC mode:** The pilot has direct control of the aircraft through the RC transmitter.

- **Assisted mode:** The pilot is limited to the high level commands for airspeed, climb rate, and course rate ($V_x$, $V_z$, $\dot{\psi}$). These high level commands can also be issued from the ground station using an auxiliary control unit (ACU) connected to the laptop.
- **Automatic mode:** Flight in this mode is fully autonomous in the programmed mission plan. This mode also includes three modes of flight interruption:
  - **Home mode:** commands the aircraft to fly to the designated Home waypoint and orbit until given a new command.
  - **Hold Figure mode:** commands the aircraft to enter an orbit about the point where the Hold Figure mode was triggered.
  - **Link Lost mode:** In the event that the FCS loses communication with the ground station for more than 30 seconds, the aircraft will return to the Home waypoint to attempt to regain the communication link. If communication with the ground station cannot be reestablished, the pilot can assume control of the aircraft and land.

At any time during flight, the pilot has the ability to revert to RC mode and assume full control of the aircraft; however, the pilot can only switch the autopilot into the Assisted mode but no other modes of operation. The ground station cannot switch the autopilot into RC mode at any point in wePilot operation, but the ground station can switch the autopilot between all other modes of operation. These mode-switching restrictions are setup to allow the pilot the chance to recover the aircraft if the autopilot malfunctions and to prevent the ground station operator from switching off the autopilot without warning to the pilot.

## 4.5 wePilot FCS Simulation

The wePilot2000 provides a system simulation environment which simulates the high level commands and programmed mission plans. The simulation uses the plant model used for the controller synthesis for a basic dynamic response and simulates sensor data. The ground station interacts with the wePilot in simulation as it does during normal operation, displaying all of the simulated data coming from the wePilot. This hardware-in-the-loop simulation environment allows the ground station operator a chance to test how the autopilot will behave throughout a planned mission and check that all waypoints are set with the correct attributes. It also gives the operator a good training environment for becoming familiar with all available information at the ground station during flight, how the autopilot responds to flight interruption, and how to edit the current mission plan and upload a revised mission.

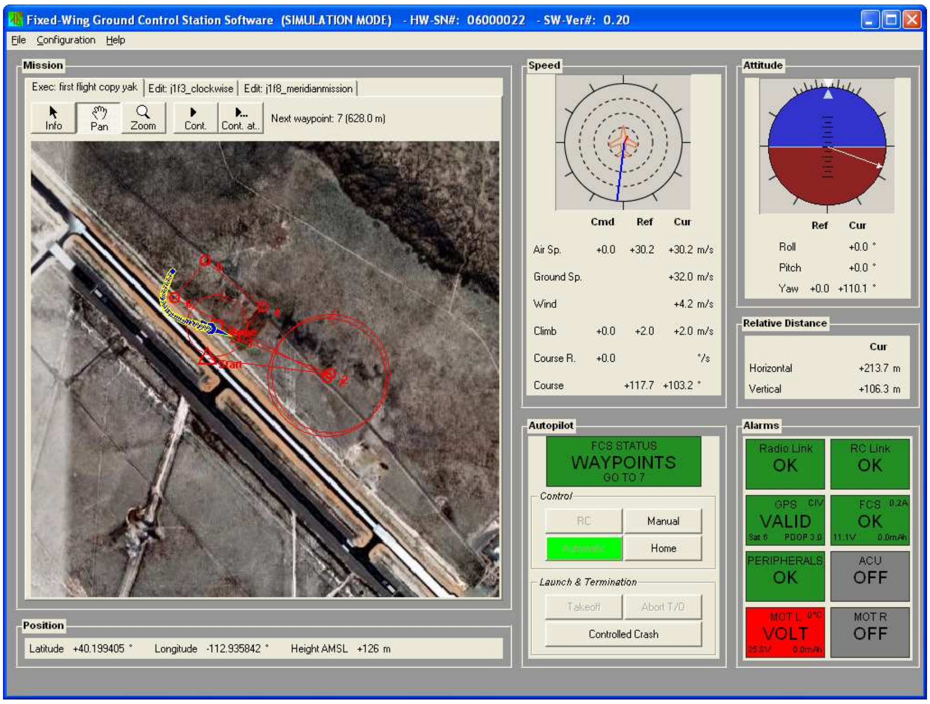



**Figure 4.5: wePilot2000 Ground Station in Simulation Mode**

## 4.6 Flight Test of wePilot2000 FCS

The wePilot2000 FCS was tested extensively by the CReSIS Flight Test Team on a one-third scale Yak-54 to validate the system for viability as the Meridian UAV flight control system and test all operational modes to be used in operation of the Meridian UAV. The experience with the wePilot2000 FCS gained through this testing, presented in Ref. [19] through [26], proves invaluable to the design of an automatic landing system using the wePilot.

Figure 4.6 shows the ground track for one particular flight in Antarctica which took the Yak-54 three miles away from the ground station. This mission was flown to test the flight path tracking capability of the wePilot.

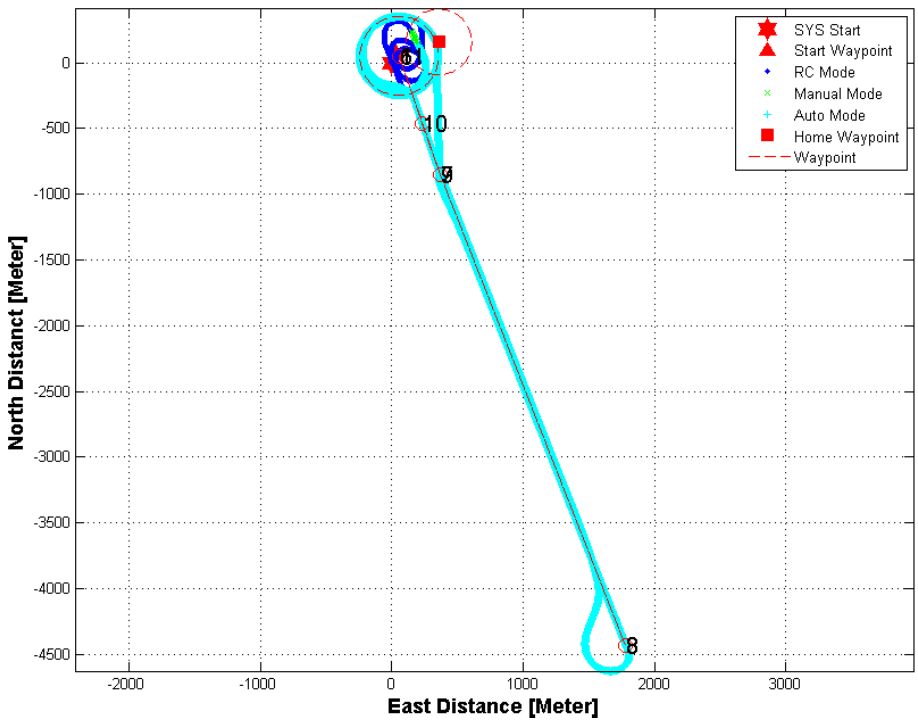



**Figure 4.6: Ground Track of Long Range Flight of Yak-54 in Antarctica**

Figure 4.7 shows a close-up of the ground track in the middle of the long straight-line flight path. This figure shows how far the aircraft deviates from the flight path when given time to converge to the flight path. When flying away from the ground station (the ground track to the left of the desired flight path), the aircraft maintained a position error of ~3 meters (~10 ft); flying back to the ground station (the ground track to the right of the desired flight path), the aircraft maintained a position error of ~6 meters (~20 ft). This implies that the flight path tracking of the wePilot is acceptable for the lateral tracking of the runway centerline for the automatic landing system.

It is important to note that flights in Antarctica were performed without the magnetometer and the wePilot software was updated to assume that the heading angle was equivalent to the course angle. This means that the wePilot was not able to compensate for crosswinds in this straight line flight. Although the ground track error is acceptable, it is also safe to assume that with an estimate for heading angle, this error could be reduced.

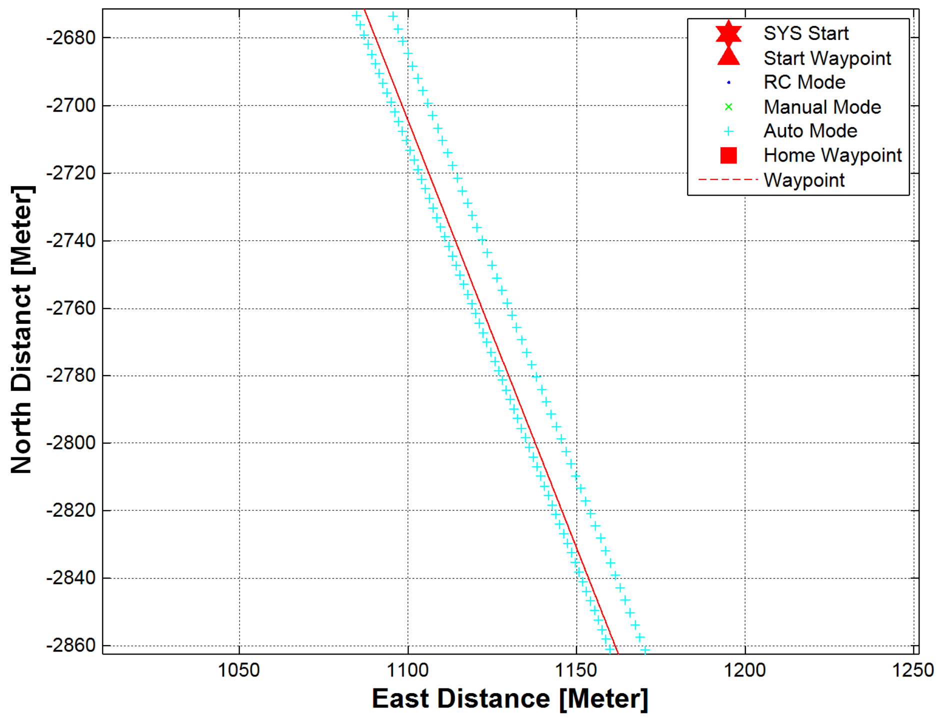



**Figure 4.7: Ground Track Compared to Desired Flight Path**

# 5 Fuzzy Logic Control Theory

Fuzzy Logic is a method for representing imprecise concepts. Where normally in crisp logic, an element is either a member of a particular set or it is not; set membership is defined as either 0 or 1. Fuzzy logic allows set membership to be partial, such that an element's membership to a set can be defined as any number between 0 and 1. [29]

A Fuzzy Logic Controller (FLC) is a control method based in linguistics in which the inputs to the controller are given a degree of membership in each applicable fuzzy set. A group of *If…and…then* rules are used to determine the fuzzy output of the FLC. This FLC output is then defuzzified with any number of varying methods to get a final crisp value for a control signal. Take for example a FLC with two inputs, *error* and *error rate*, and one control output. The inputs to this system can be called zero, negative, or positive, and if negative or positive, the inputs can be described as large or small or anything in between. This example FLC would be designed with rules such as: *If* error is negative and large *and* error rate is positive and small *then* control output is positive. This is the basic form of a Fuzzy Logic Controller.

This chapter presents the FLC design process and the MATLAB/SIMULINK construction for the FLC for simulation of the autolanding controller.

## 5.1 Fuzzy Logic Controller Synthesis

The first step in the design of the FLC is the construction of the membership functions for the error, the error rate, and the controller output $V_z$. These membership functions require predetermined fuzzy sets to describe the scale of a given parameter. As fuzzy logic is based on linguistic representation, this FLC uses five fuzzy sets named

Negative Big, Negative Small, Zero, Positive Small, and Positive Big, abbreviated as NB, NS, Z, PS, and PB respectively. Figure 5.1 through Figure 5.3 show the terms of the controller parameters using triangular membership functions for the five fuzzy sets. The degree of *error* ($e$=h-$h_d$) is measured on a scale from -10 m to 10 m, and the degree of *error rate* (*de/dt*) is measured on a scale from -4 m/s to 4 m/s. These scales are based on flight test experience with the wePilot FCS and the Meridian UAV.

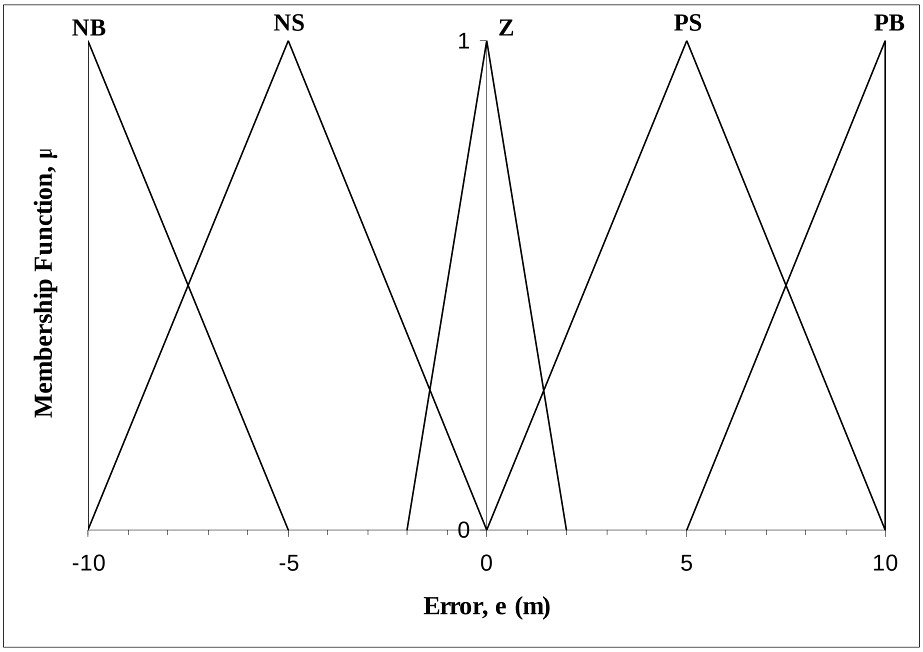



**Figure 5.1: Terms of the input parameter *error*.**

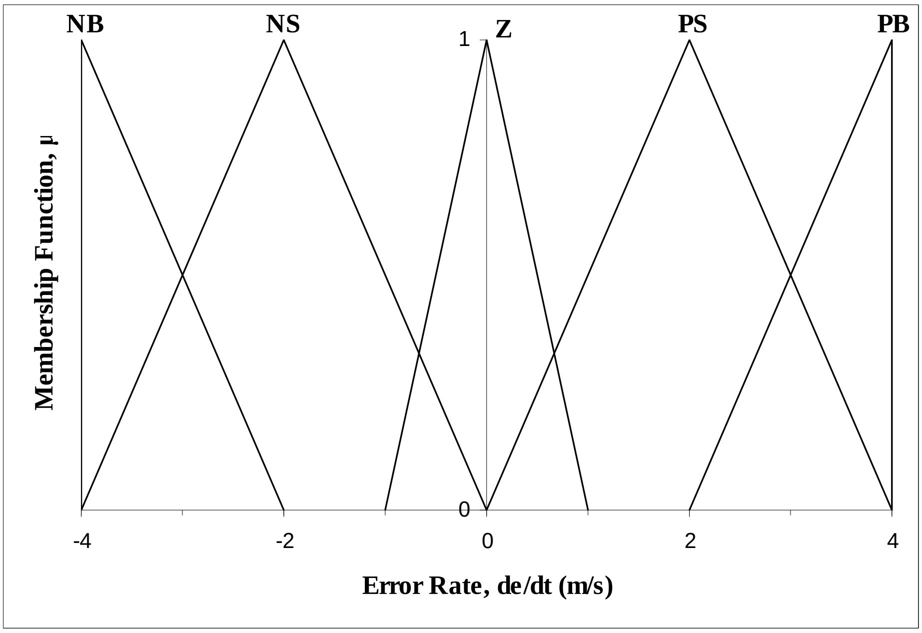



**Figure 5.2: Terms of the input parameter *error rate*.**

The degree of the control output, $V_z$, is measured from -2 m/s to 2 m/s based on the current $V_z$ control input limits for the wePilot FCS.

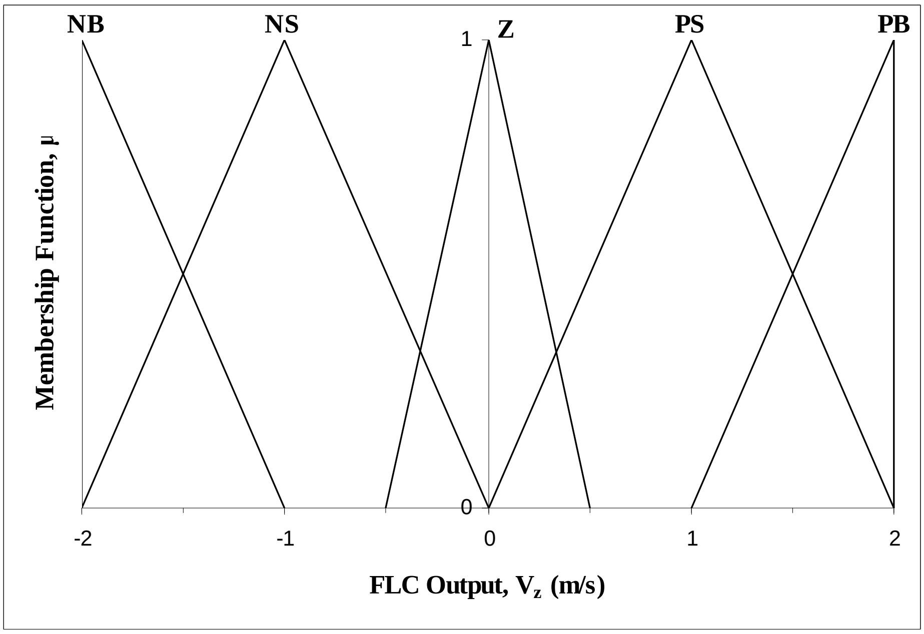



**Figure 5.3: Terms of the output parameter $V_z$.**

The terms of *error*, *error rate*, and *Vz* have membership functions presented analytically in Eqns(5.1) through (5.3).

$$\mu_e(x)=\begin{cases}\mu_{NB}(x)=-\left(\dfrac{x+5}{5}\right) & \text{for } -10\le x\le -5\\ \mu_{NS}(x)=\begin{cases}\dfrac{x+10}{5} & \text{for } -10\le x\le -5\\ -\dfrac{x}{5} & \text{for } -5\le x\le 0\end{cases}\\ \mu_{Z}(x)=\begin{cases}\dfrac{x+2}{2} & \text{for } -2\le x\le 0\\ \dfrac{2-x}{2} & \text{for } 0\le x\le 2\end{cases}\\ \mu_{PS}(x)=\begin{cases}\dfrac{x}{5} & \text{for } 0\le x\le 5\\ \dfrac{10-x}{5} & \text{for } 5\le x\le 10\end{cases}\\ \mu_{PB}(x)=\dfrac{x-5}{5} & \text{for } 5\le x\le 10\end{cases}$$

(5.1)

$$\mu_{\dot e}(y)=\begin{cases}\mu_{NB}(y)=\dfrac{2-y}{2} & \text{for } -4\le x\le -2\\ \mu_{NS}(y)=\begin{cases}\dfrac{y+4}{2} & \text{for } -4\le y\le -2\\ -\dfrac{y}{2} & \text{for } -2\le y\le 0\end{cases}\\ \mu_{Z}(y)=\begin{cases}y+1 & \text{for } -1\le y\le 0\\ 1-y & \text{for } 0\le y\le 1\end{cases}\\ \mu_{PS}(y)=\begin{cases}\dfrac{y}{2} & \text{for } 0\le y\le 2\\ \dfrac{4-y}{2} & \text{for } 2\le y\le 4\end{cases}\\ \mu_{PB}(y)=\dfrac{y-2}{2} & \text{for } 2\le y\le 4\end{cases}$$

(5.2)

$$\mu_{V_Z}(z)=\begin{cases}\mu_{NB}(z)=-1-z & \text{for} \quad -2\le z\le -1\\ \mu_{NS}(z)=\begin{cases}z+2 & \text{for} \quad -2\le z\le -1\\ -z & \text{for} \quad -1\le z\le 0\end{cases}\\ \mu_{Z}(z)=\begin{cases}\dfrac{z+0.5}{0.5} & \text{for} \quad -0.5\le z\le 0\\ \dfrac{0.5-z}{0.5} & \text{for} \quad 0\le z\le 0.5\end{cases}\\ \mu_{PS}(z)=\begin{cases}z & \text{for} \quad 0\le z\le z\\ 2-z & \text{for} \quad 1\le z\le 2\end{cases}\\ \mu_{PB}(z)=z-1 & \text{for} \quad 1\le z\le 2\end{cases} \tag{5.3}$$

With the membership functions designed, the next step is to formulate the fuzzy control rules for the FLC. These are linguistic rules for the FLC which take on the typical form

IF *error* is A AND *error rate* is B THEN $V_z$ is C

where A, B and C are the fuzzy sets for the error, error rate, and controller output. In this case, with five fuzzy sets for each of two input parameters, the FLC contains twenty-five rules which can be represented in table form as shown in Table 5.1.

**Table 5.1: Rule Base for FLC**

| | **Fuzzy Set, de/dt** | | | | |
|---|---|---|---|---|---|
| **Fuzzy Set, e** | **NB** | **NS** | **Z** | **PS** | **PB** |
| **NB** | PB | PB | PS | PS | NS |
| **NS** | PB | PS | PS | NS | NB |
| **Z** | PB | PS | Z | NS | NB |
| **PS** | PB | PS | NS | NS | NB |
| **PB** | PS | NS | NS | NB | NB |

The rule base for the FLC serves to determine the fuzzy output of the controller based on the fuzzified inputs. The common Mamdani-type min-max composition is used for this process of fuzzy output determination known as fuzzy inference or rule firing. In this process, the FLC rules are evaluated using the fuzzified inputs to the system and taking the minimum calculated degree of the output fuzzy set membership for each applicable rule. For example, assume the sensor readings are $e$=7 m and $de/dt$=0.75 m/s. To obtain the fuzzy inputs, these sensor readings are substituted into Eqns(5.1) and (5.2) which yields

$$\mu_{e,PS}(7)=3/5, \qquad \mu_{e,PB}(7)=2/5, \qquad \mu_{de/dt,Z}(0.75)=1/4, \qquad \mu_{de/dt,PS}(0.75)=3/8.$$

For this case, the rule base of Table 5.1 reduces to the induced decision Table 5.2.

**Table 5.2: Induced decision table for FLC example**

| | **0** | **0** | **$\mu_{de/dt,Z}(0.75)=1/4$** | **$\mu_{de/dt,PS}(0.75)=3/8$** | **0** |
|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 | 0 |
| **0** | 0 | 0 | 0 | 0 | 0 |
| **0** | 0 | 0 | 0 | 0 | 0 |
| **$\mu_{e,PS}(7)=3/5$** | 0 | 0 | $\mu_{NS}(z)$ | $\mu_{NS}(z)$ | 0 |
| **$\mu_{e,PB}(7)=2/5$** | 0 | 0 | $\mu_{NS}(z)$ | $\mu_{NB}(z)$ | 0 |

These four applicable rules can be expressed formally as:

Rule 1: If $e$ is PS and $de/dt$ is Z then $z$ is NS,
Rule 2: If $e$ is PS and $de/dt$ is PS then $z$ is NS,
Rule 3: If $e$ is PB and $de/dt$ is Z then $z$ is NS,
Rule 4: If $e$ is PB and $de/dt$ is PS then $z$ is NB.

With the Mamdani-type min-max composition employed, the strength of these four rules is calculated by taking the minimum of the fuzzy input values for each rule as shown in Table 5.3.

**Table 5.3: Rules strength table**

| | **0** | **0** | **$\mu_{de/dt,Z}(0.75)=1/4$** | **$\mu_{de/dt,PS}(0.75)=3/8$** | **0** |
|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 | 0 |
| **0** | 0 | 0 | 0 | 0 | 0 |
| **0** | 0 | 0 | 0 | 0 | 0 |
| **$\mu_{e,PS}(7)=3/5$** | 0 | 0 | min(3/5,1/4)=1/4 | min(3/5,3/8)=3/8 | 0 |
| **$\mu_{e,PB}(7)=2/5$** | 0 | 0 | min(2/5,1/4)=1/4 | min(2/5,3/8)=3/8 | 0 |

The control output for the four rules is obtained by taking the minimum of the rule strength and the corresponding fuzzy set in the output membership function, which results in Table 5.4. Figure 5.4 through Figure 5.6 show the control output calculated for the four applicable rules.

**Table 5.4: Control outputs**

| | **0** | **0** | $\mu_{de/dt,Z}(0.75)=1/4$ | $\mu_{de/dt,PS}(0.75)=3/8$ | **0** |
|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 | 0 |
| **0** | 0 | 0 | 0 | 0 | 0 |
| **0** | 0 | 0 | 0 | 0 | 0 |
| $\mu_{e,PS}(7)=3/5$ | 0 | 0 | $\min(1/4,\mu_{NS}(z))$ | $\min(3/8,\mu_{NS}(z))$ | 0 |
| $\mu_{e,PB}(7)=2/5$ | 0 | 0 | $\min(1/4,\mu_{NS}(z))$ | $\min(3/8,\mu_{NB}(z))$ | 0 |

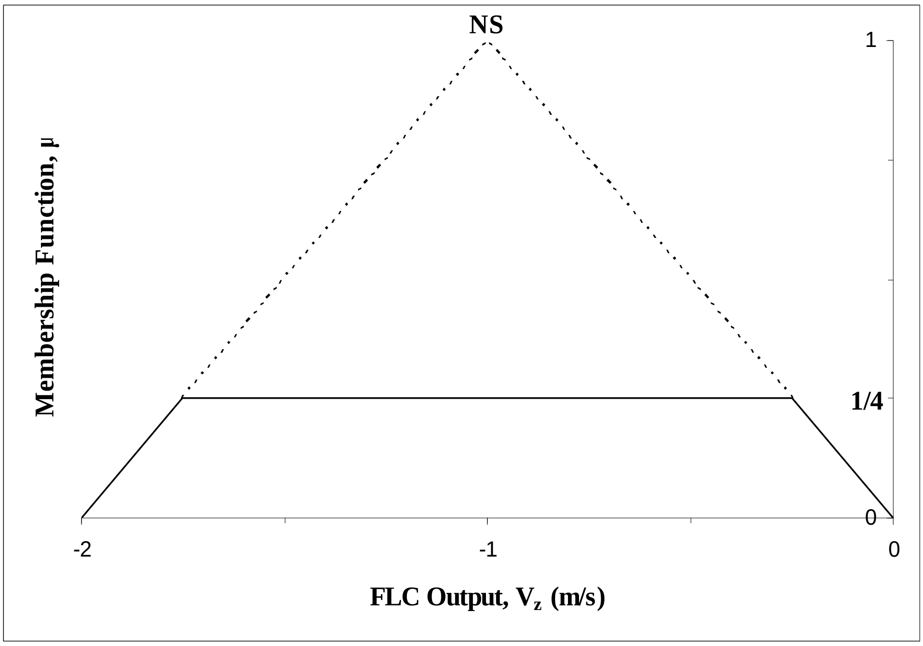



**Figure 5.4: Control output for Rules 1 & 3**

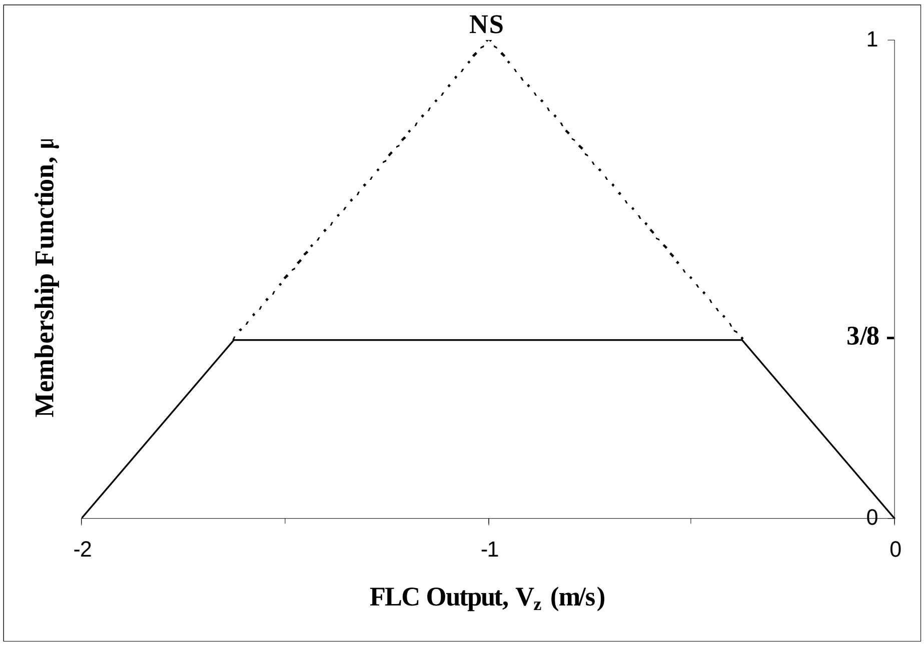



**Figure 5.5: Control Output for Rule 2**

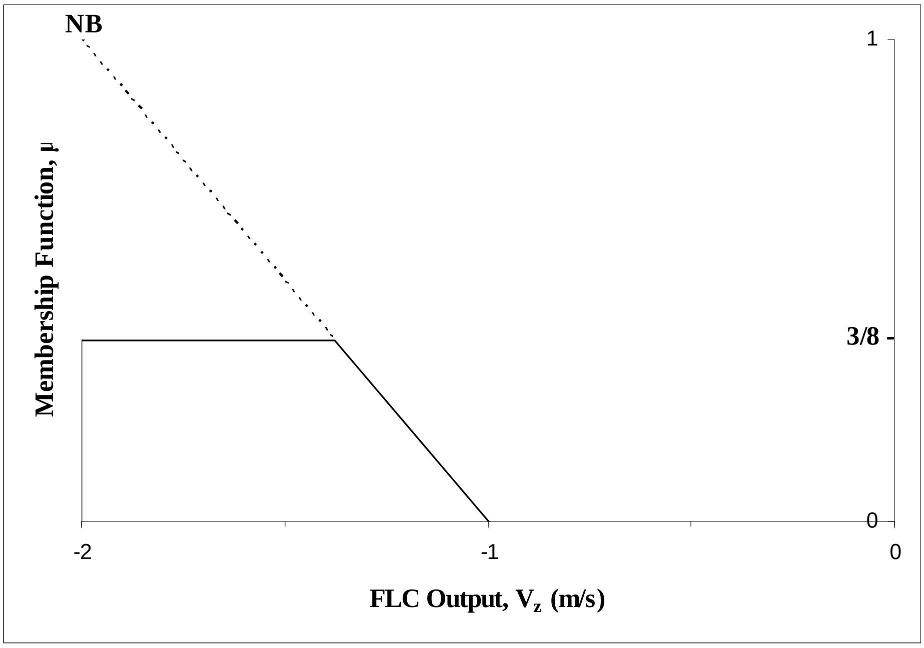



**Figure 5.6: Control Output for Rule 4**

The fuzzy output of the system is now calculated by aggregating the control outputs of the applicable rules. The Mamdani-type min-max FLC forms the aggregated control output by taking the maximum of the individual control outputs. In this example, the aggregate control output is

$$\mu_{agg}(z) = \max\{\min(1/4,\mu_{NS}(z)), \min(3/8,\mu_{NS}(z)), \min(3/8,\mu_{NB}(z))\},$$

which is shown in Figure 5.7.

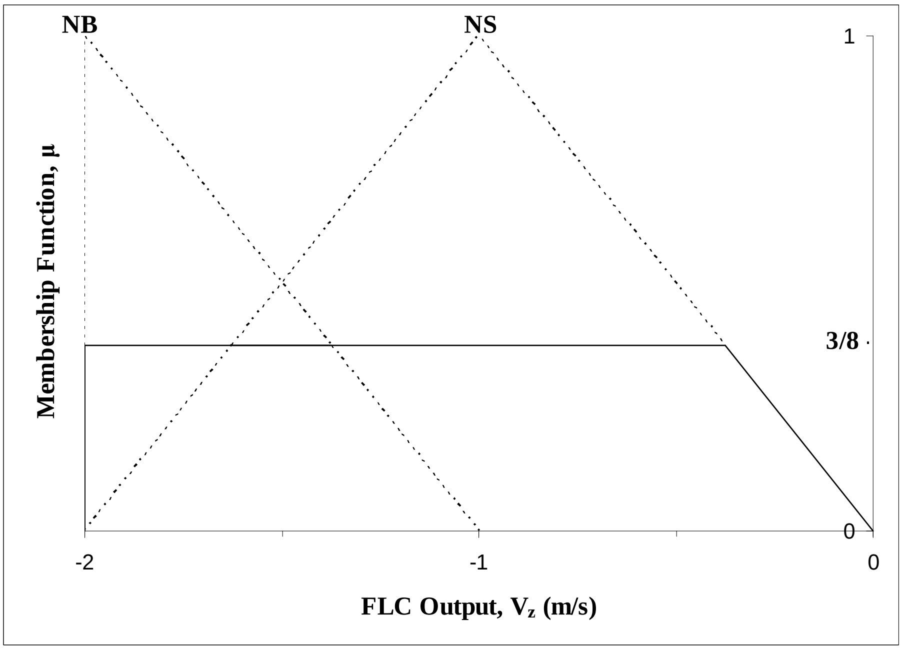



**Figure 5.7: Aggregate control output**

The final step in the FLC design is defuzzification of the aggregated control output into a crisp value for the controller output, in this case $V_z$. This FLC employs the center of area method which calculates the crisp value by finding the center of area of the aggregate control output using Eqn (5.4)

$$z^* = \frac{\sum_{j=1}^{k} z_j \mu_{agg}(z_j)}{\sum_{j=1}^{k} \mu_{agg}(z_j)} \tag{5.4}$$

where $z^*$ is the controller output, $V_z$ in this case, $k$ is the number of subintervals of the output, $z_j$ is the control output for a given subinterval, and $\mu_{agg}(z_j)$ is the membership value in the controller output fuzzy set. For the example at hand, the FLC output calculation yields $V_z$ = -1.1 m/s, as shown in Figure 5.8.

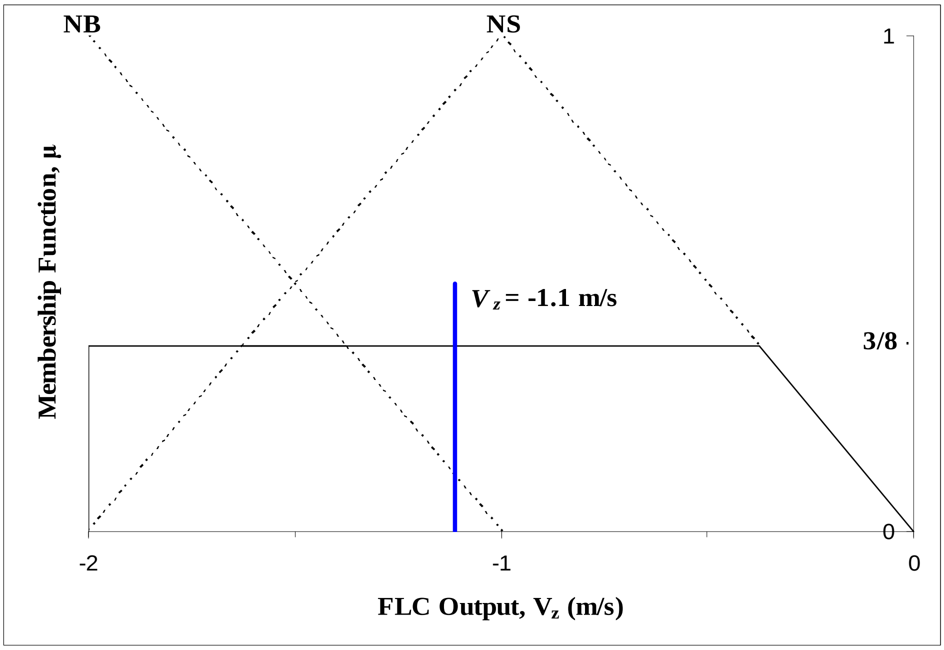



**Figure 5.8: Defuzzification of the FLC**

The FLC for $V_x$ control is designed in the same manner as the $V_z$ FLC. It does not, however, send a $V_x$ command directly to the wePilot. Rather, the output of the $V_x$ FLC is summed with the desired $V_x$ at the airplane's location in the approach phase. This FLC is also a one input – one output design, where the input is the error ($V_x$ – $V_{xd}$) and the output is the required change in the commanded $V_x$ to achieve the desired actual $V_x$.

## 5.2 MATLAB/SIMULINK FLC Model

The fuzzy logic controller described in Section 5.1 is built into the autolanding SIMULINK model using the MATLAB Fuzzy Logic Toolbox. This toolbox allows the user to design a fuzzy logic controller using a Graphical User Interface (GUI). Figure 5.9 shows the FIS (Fuzzy Inference System) editor window where the following basic attributes of the FLC are set: the input and output parameters, the type of inference (Mamdani-type in this case), the rule firing method, and the defuzzification method.

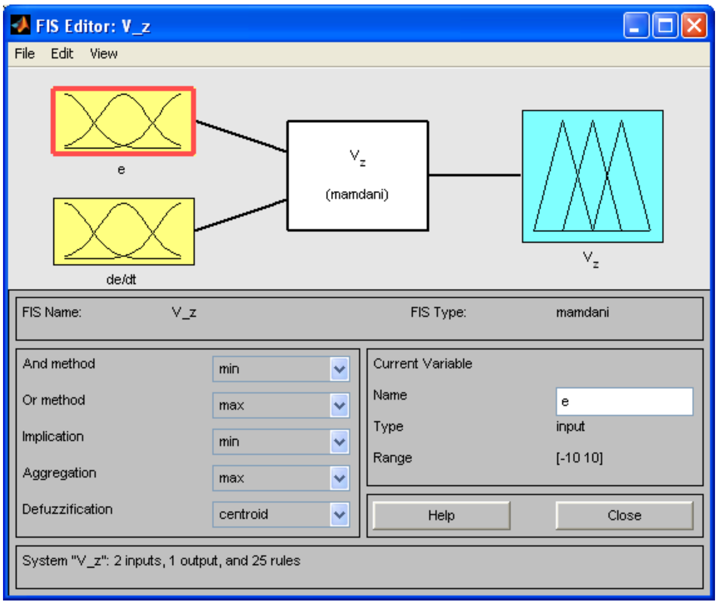



**Figure 5.9: FIS editor**

The Membership Function Editor, shown in Figure 5.10, is where the parameter terms of Figure 5.1 through Figure 5.3 are incorporated into the controller.

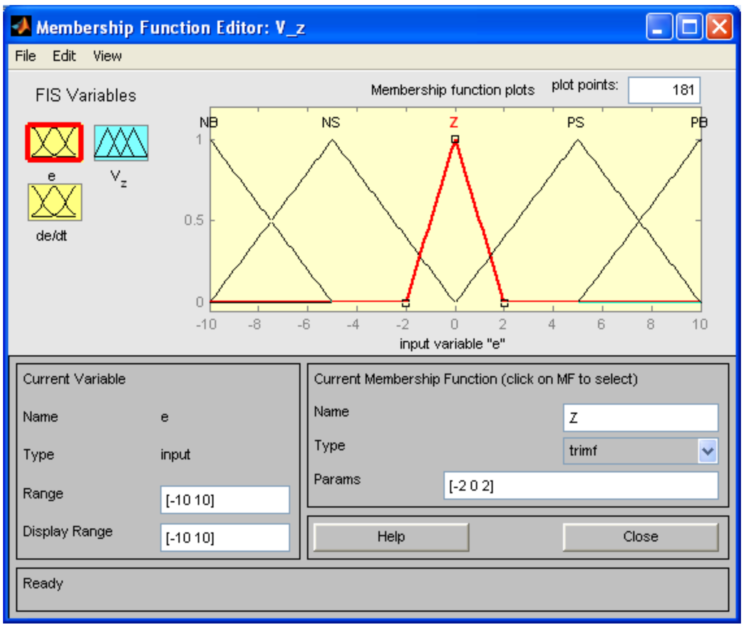



**Figure 5.10: Membership Function Editor**

Figure 5.11 shows the Rule Editor which is used to build the rule base for the FLC. In this case, all twenty-five rules of the FLC represented by Table 5.1 are entered in the linguistic form into the Rule Editor.

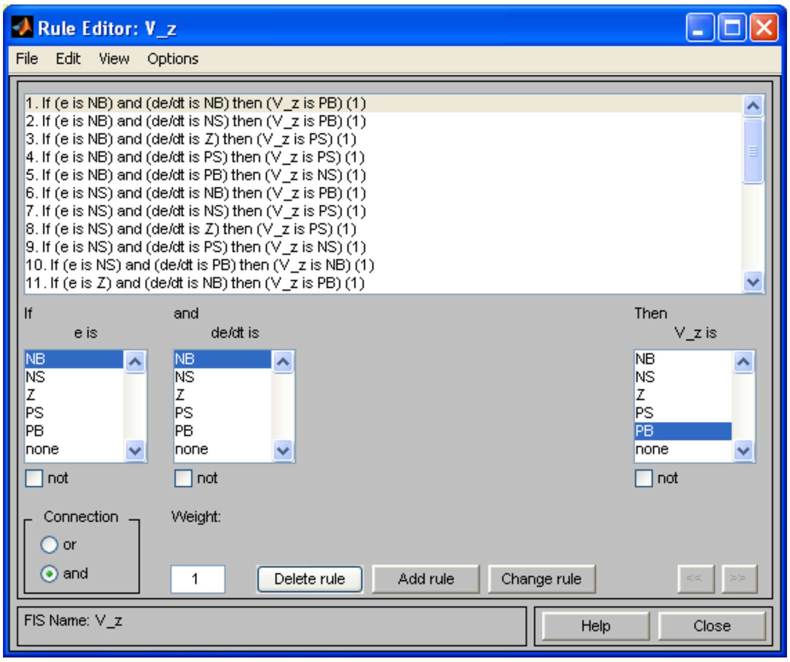



**Figure 5.11: Rule Editor**

With these three modules completed, the FLC design is complete. Figure 5.12 shows the Surface Viewer which allows the user to view how the inputs of the controller map to corresponding outputs. In this case, with only two inputs and one output, the surface viewer shows the entire mapping of the controller for $V_z$. The fuzzy logic toolbox buildup of the FLC design for $V_x$ is presented in Appendix B.

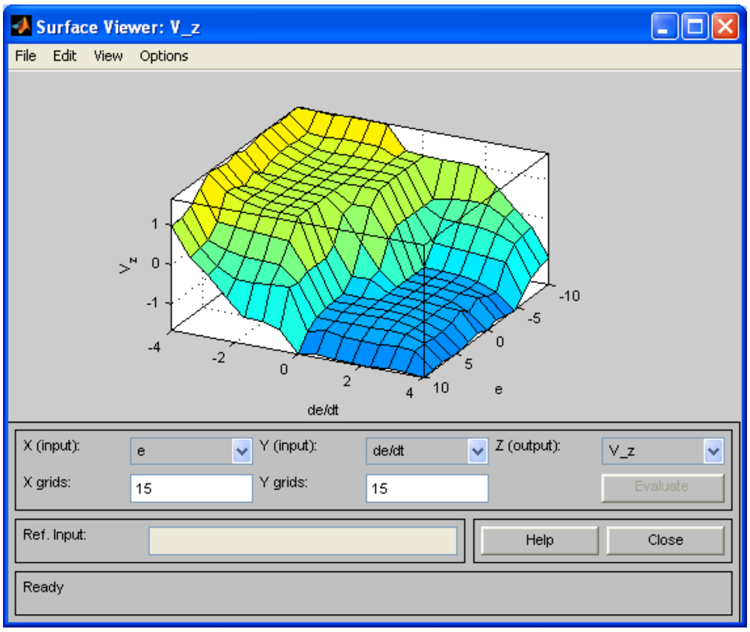



**Figure 5.12: Surface Viewer**

With the FLC design finished, the MATLAB Fuzzy Logic Toolbox incorporates the controller design into a FLC SIMULINK block for simulation. The block diagram implementation of the FLC for $V_x$ is shown in Figure 5.13.

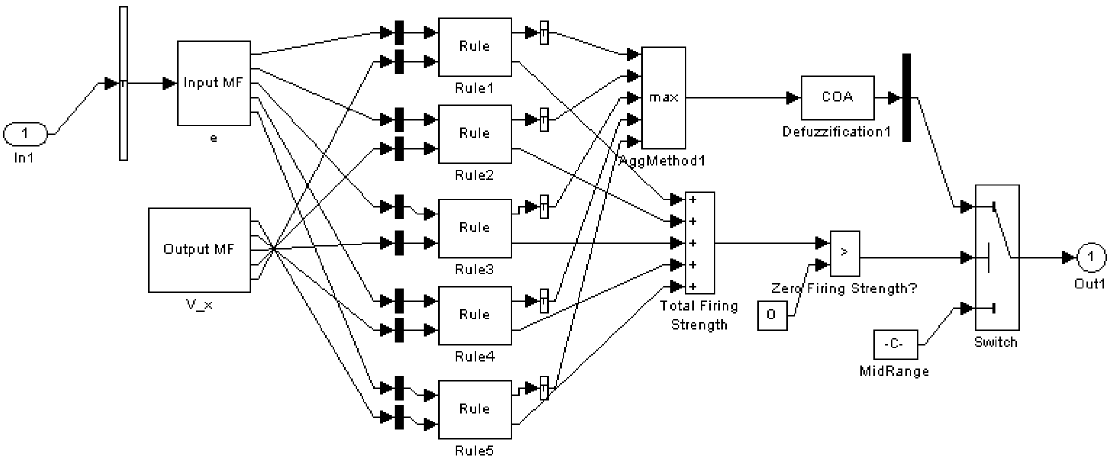



**Figure 5.13: SIMULINK block diagram of $V_x$ FLC**

# 6 Fuzzy Logic Autolanding Controller using the wePilot FCS

The fuzzy logic controller designed in Chapter 5 is sends outputs $V_{z\text{-cmd}}$ and $V_{x\text{-cmd}}$ to the wePilot while in Assisted Mode. The inputs to the FLC are altitude, distance relative to the start of the flare (h and x respectively), and the current $V_z$ and $V_x$. Figure 6.1 presents a general block diagram of the fuzzy logic autolanding controller using the wePilot, where $\bar{x} = [V_z \quad h \quad V_x \quad x]^T$. This controller is operational only in the vertical plane (x-z). As discussed in Section 4.6, the lateral tracking for the autolanding system will rely on the wePilot flight path tracking capability.

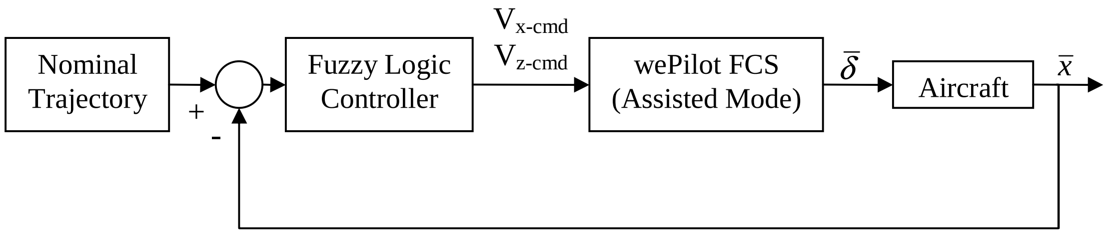



**Figure 6.1: General Block Diagram of Autolanding Controller**

This chapter presents the plant model for simulation, the desired flight path trajectory, and the results of the controller simulation. The SIMULINK model used for simulation is shown in Appendix C.

## 6.1 Plant Model for Autolanding Controller Simulation

The autolanding controller is designed to wrap a high level control algorithm around the wePilot FCS. As such, the controller is only given control over high level commands in the wePilot, $V_z$ and $V_x$ for longitudinal control. For simulation purposes, these high level commands were characterized in recent CReSIS work [27]. The MATLAB System I.D. toolbox was used to determine the transfer function for $V_z$ and $V_x$

control based on flight test data obtained over several autonomous flights with the wePilot FCS. The resulting average transfer functions are shown in Eqns(6.1) and (6.2).

$$\frac{V_z}{V_{z_{cmd}}} = \frac{2.3091}{s+1.6321} \tag{6.1}$$

$$\frac{V_x}{V_{x_{cmd}}} = \frac{0.2974}{s+0.2848} \tag{6.2}$$

The transfer functions are converted to state-space form and combined with the simple relationship that $V_z$ is the first derivative of altitude *h* and $V_x$ is the first derivative of position *x*. This state-space model is presented in Eqn (6.3).

$$\begin{bmatrix} \dot{V}_z \\ \dot{h} \\ \dot{V}_x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} -1.6321 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -0.2848 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} V_z \\ h \\ V_x \\ x \end{bmatrix} + \begin{bmatrix} 2.3091 & 0 \\ 0 & 0 \\ 0 & 0.2974 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_{z_{cmd}} \\ V_{x_{cmd}} \end{bmatrix} \tag{6.3}$$

This state-space model allows convenient calculation of the instantaneous altitude and distance from the runway for the simulation. This plant model also simulates the operation of the autolanding controller with the wePilot, as the controller has no direct control over the attitude of the airplane. The unfortunate side-effect of this is the lack of simulation of the attitude of the airplane at the touchdown point. For this reason, extensive flight test of the autolanding controller ‘landing’ at 100-500ft will be required to collect data on touchdown point attitude to facilitate controller tuning.

## 6.2 Desired Flight Path Trajectory

As discussed in [3], the landing process consists of two phases, the approach and the flare. The total landing trajectory is presented in Figure 6.2.

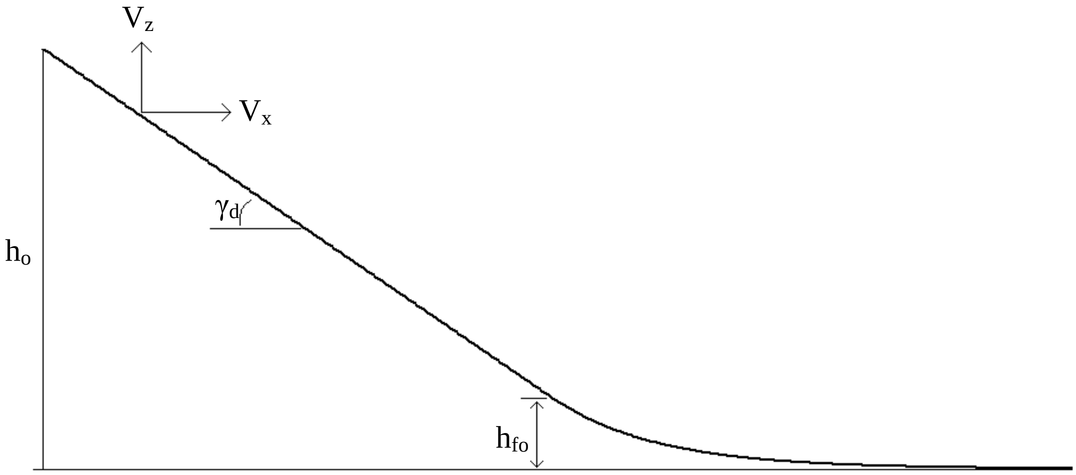



**Figure 6.2: Trajectory for Automatic Landing ($V_x$ and $V_z$ positive as shown)**

The approach is a descent to the flare height at a constant glide slope as described by

$$\dot{h}_d = V_T \sin(\gamma_d) \tag{6.4}$$

The standard range for the desired glide slope is 2.5°-3° degrees for commercial aircraft [3] or higher for different applications. For a UAV, this glide slope can be greater than that of commercial aircraft due to the lack of passenger comfort concerns; however, with the approach speed set at 41 m/s and a maximum commanded $V_z$ = 2 m/s, the maximum glide slope attainable is 2.8°. Furthermore, the desired glide slope should be less than this maximum so the controller can recapture the desired altitude in the case of wind disturbances. For this reason, the desired vertical velocity for the landing approach is set to -0.82 m/s which is a rather small glide slope of 1.15°.

For the autolanding controller, the instantaneous desired altitude during the approach is calculated based on the current distance from the start of the flare maneuver as shown in Eqn(6.5), where $h_{fo}$ is the altitude at the beginning of the flare.

$$h_d = \frac{V_{z_d}}{V_{x_d}} x + h_{f_0} \tag{6.5}$$

The desired flare trajectory is an exponential curve which starts at $h_{fo}$ and decays to h = 0 m. The general form of the equation for the desired flare trajectory is given by Eqn (6.6):

$$h_f = h_{f_0} e^{-t/\tau} \tag{6.6}$$

The time constant is set to $\tau$ = 5 s and the flare height is set to $h_{fo}$ = 10 m. The chosen time constant is relatively high, compared to other research in autolanding controllers (for example, [3] and [4]); but because of the lack of attitude control during this maneuver, a more gradual flare is desirable, i.e. an exponential curve with a larger time constant. The vertical velocity throughout the flare maneuver is the first derivative of Eqn(6.7):

$$\dot{h}_f = -(\frac{h_f}{\tau}) e^{-t/\tau} \tag{6.7}$$

For the controller to calculate the desired height and vertical velocity during the flare, Eqns(6.6) and (6.7) are adjusted to be in terms of the distance from the start of the flare as shown in Eqns(6.8) and (6.9).

$$h_f = h_{f_0} e^{-x/V_{x_d}\tau} \tag{6.8}$$

$$\dot{h}_f = -(\frac{h_f}{\tau}) e^{-x/V_{x_d}\tau} \tag{6.9}$$

The desired forward velocity control, $V_x$, is set to 41 m/s (80 kts) in the approach phase until the airplane is at an altitude of 50 meters. From there, and into the flare maneuver, the desired $V_x$ is set to 36 m/s (70 kts) which is ~$1.3V_{stall}$ as [3] states is the ideal approach speed.

### 6.3 Simulation Results

For simulation of the autolanding controller, the initial conditions are set as follows: $V_z(0) = 0$ m/s, $h_0(0) = 70$ m, $V_x(0) = 45$ m/s, and $x_0(0) = -3500$ m. At this initial position, $h_d = 80$ m (see Eqn(6.5)) which requires the controller to capture the flight path as it flies forward from the initial position. Figure 6.3 presents the overall results from the autolanding controller simulation. The flare trajectory described by Eqn(6.8) is adjusted such that the exponential curve converges to one meter below the runway, i.e. $h_{fo} = 11$ m and one meter is subtracted from the right hand side of the equation. Figure 6.4 presents the controller signal from both the $V_z$ and $V_x$ fuzzy logic controllers. The autolanding controller performance during the approach and the flare is presented in Figure 6.5 and Figure 6.6 with the deviation from the desired flight path.

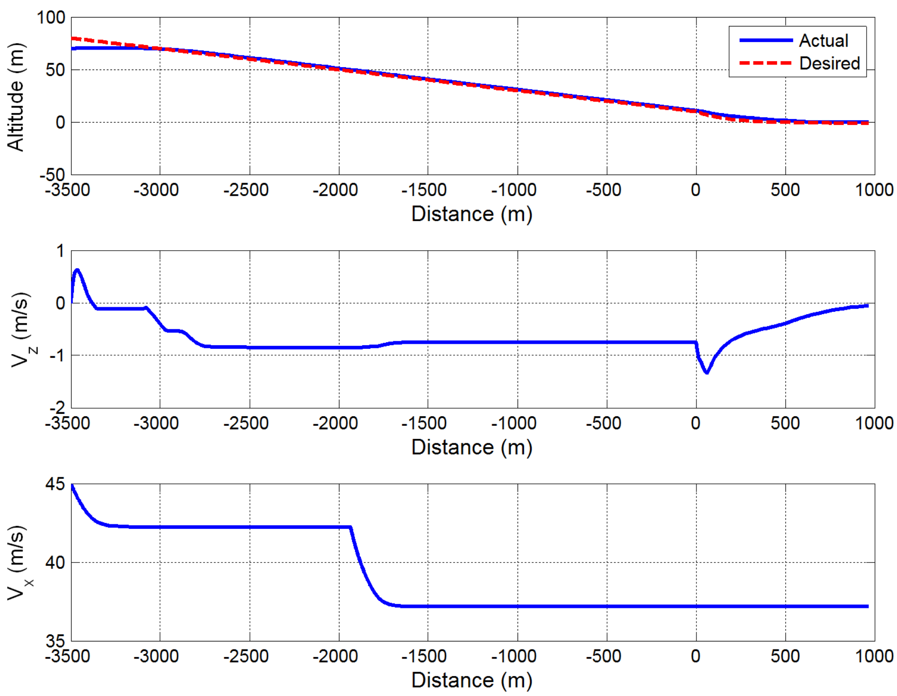



**Figure 6.3: Overall Autolanding Controller Simulation Results**

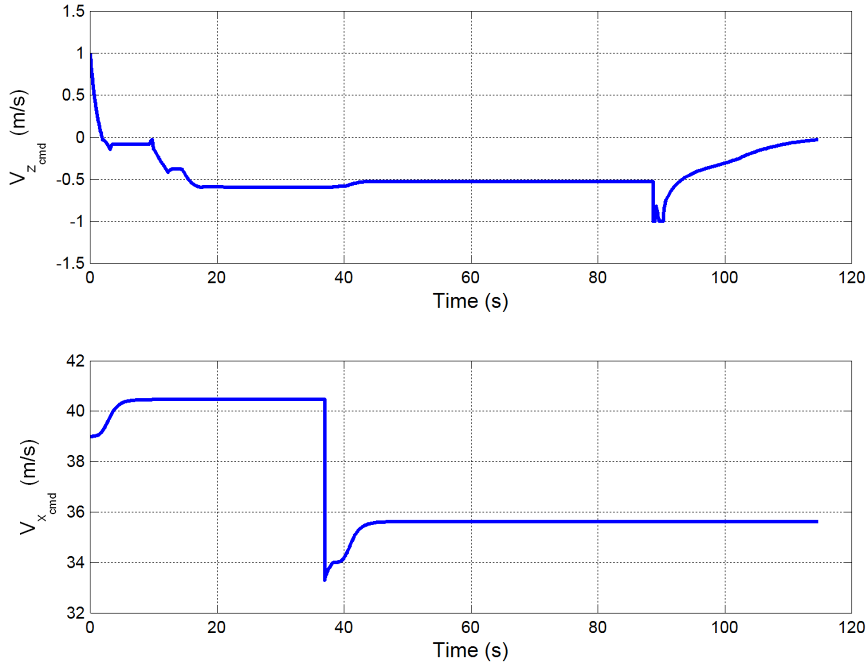



**Figure 6.4: FLC Output**

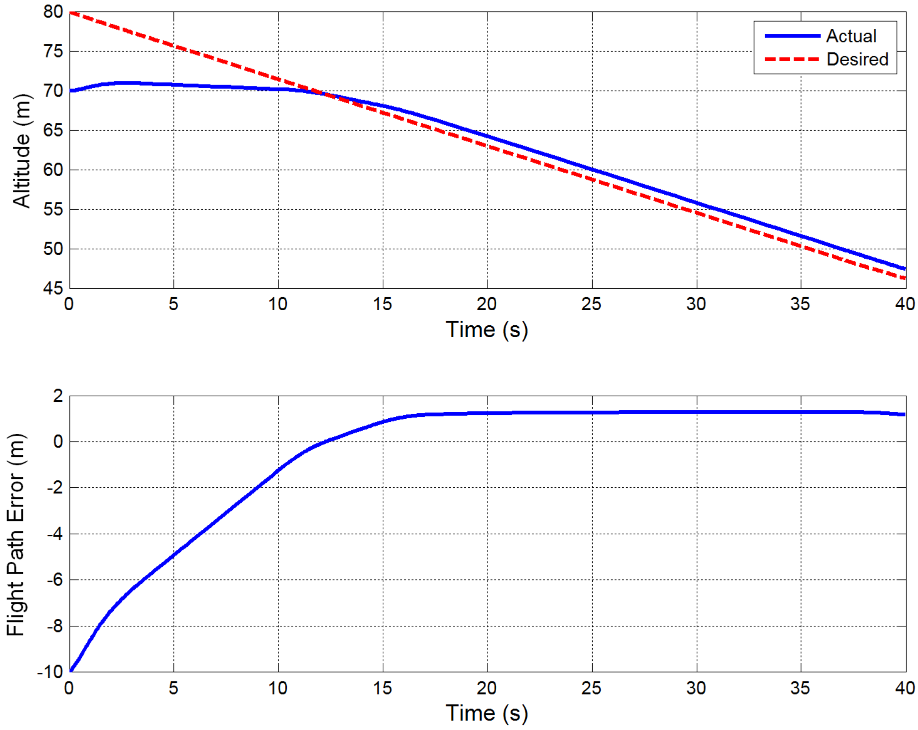



**Figure 6.5: Glide Path Trajectory**

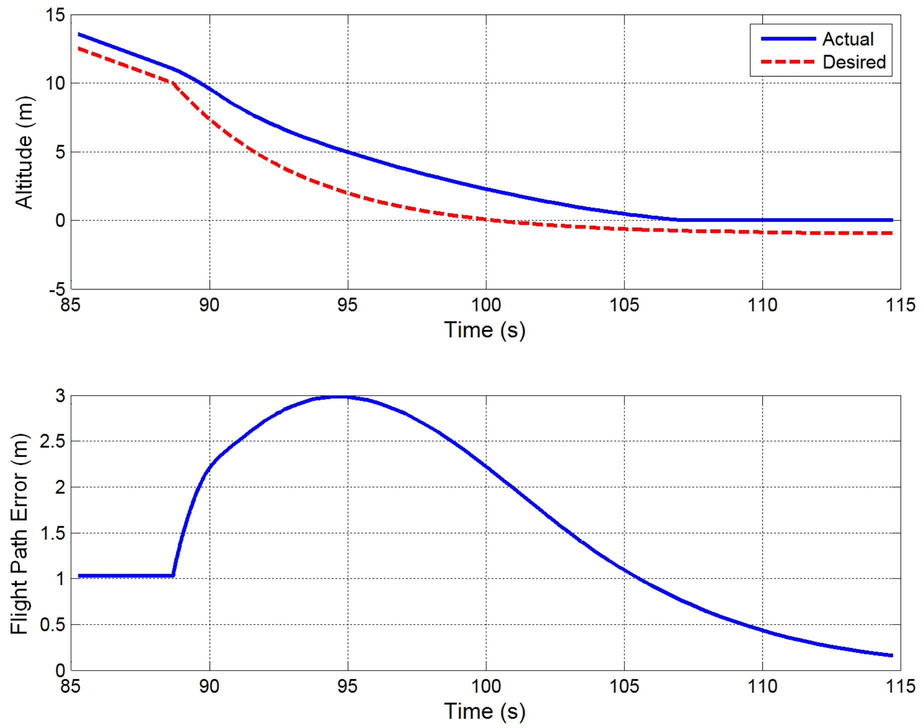



**Figure 6.6: Flare Trajectory**

The FLC settles at a steady state error of $e_{ss} \approx 1$ m (3.3 ft), which is seen in Figure 6.5, during the approach phase of the landing. During the flare maneuver, the error builds to almost 3 m above the desired exponential curve. This error is largely a result of the inherent priority that the FLC input of error rate, *de/dt*, holds within the controller. This relationship can be seen in Figure 6.7 where the initial spike in the error rate at the beginning of the flare incites the initial spike in the $V_{z\text{-}cmd}$ from the FLC. As the error rate reduces to zero, so follows the commanded vertical velocity.

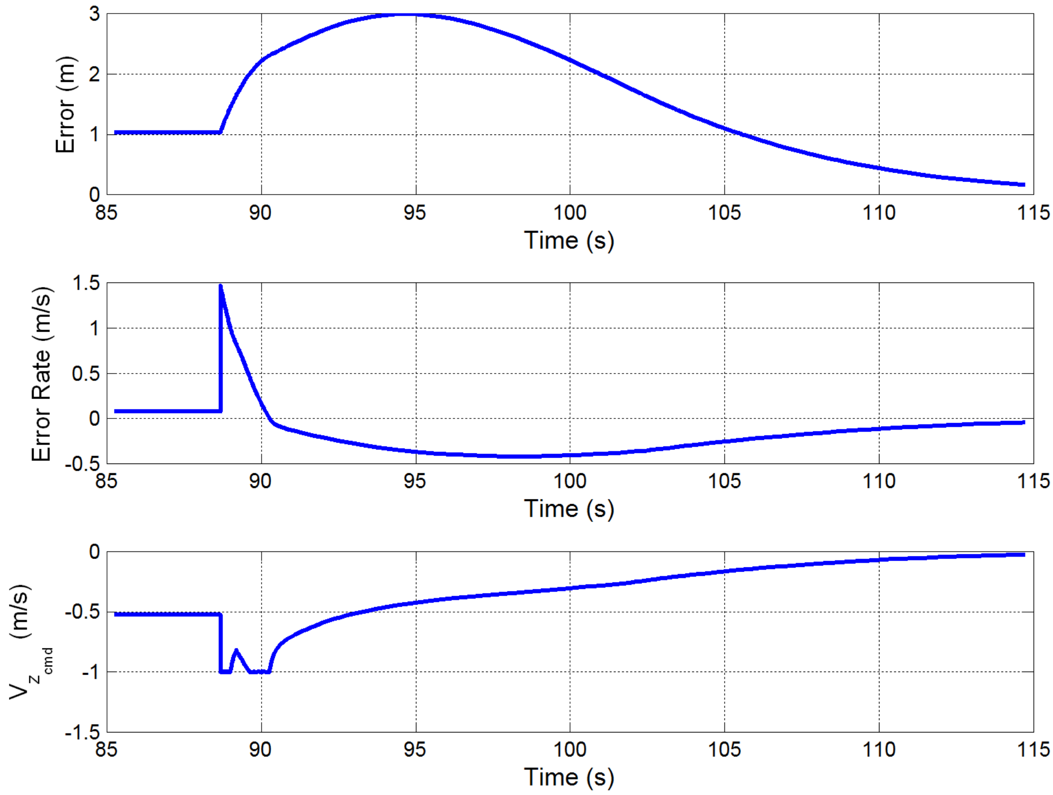



**Figure 6.7: FLC response to Error and Error Rate during flare**

Despite the error during the flare maneuver, the FLC guides the airplane to a smooth landing. Figure 6.8 shows the vertical velocity during the flare maneuver and highlights the vertical velocity at the touchdown point which is $V_{z\text{-}TD}$ = -0.19 m/s (-0.62 ft/s). The desirable touchdown speed is generally 2-3 ft/s (~0.6-0.9 m/s) as stated in [28]; this autolanding controller, however, is designed for a lower touchdown speed due to the lack of attitude control at the touchdown point. The vertical velocity at touchdown can be adjusted by simply reducing the altitude of the flare trajectory further into the ground. Before this is taken into consideration, flight test of the system landing at heights well above the ground should be performed to observe the attitude of the aircraft during the flare maneuver.

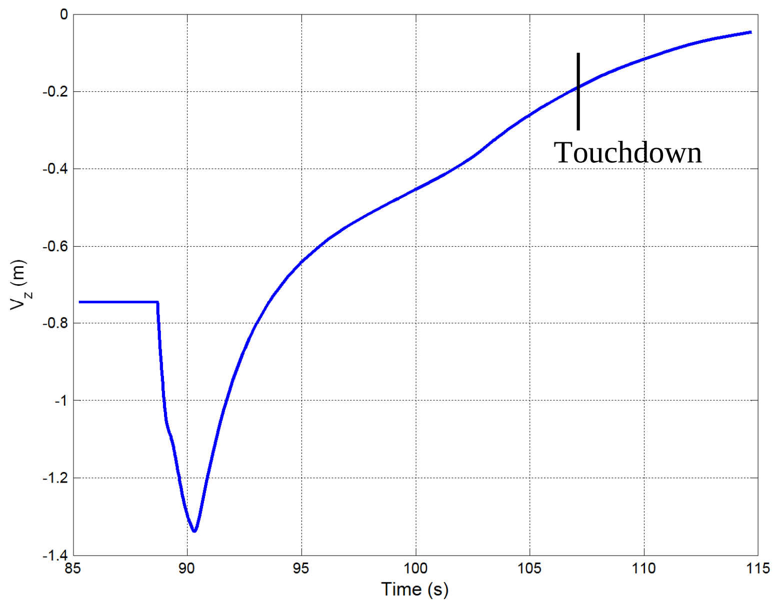



**Figure 6.8: Vz during the flare maneuver**

After initial simulation of the autolanding controller, the FLC is adjusted for the recommended case where the wePilot limit on $V_z$ is increased from 2 m/s to 4 m/s. In this case, the commanded glide slope in the approach phase of the landing is increased to $\gamma_d$=2.8° (2 m/s for the approach airspeed of 41 m/s). The overall results of this new simulation are presented in Figure 6.9. The steady state error in this case settles on $e_{ss} \approx$ 1.28 m which is similar to initial simulation.

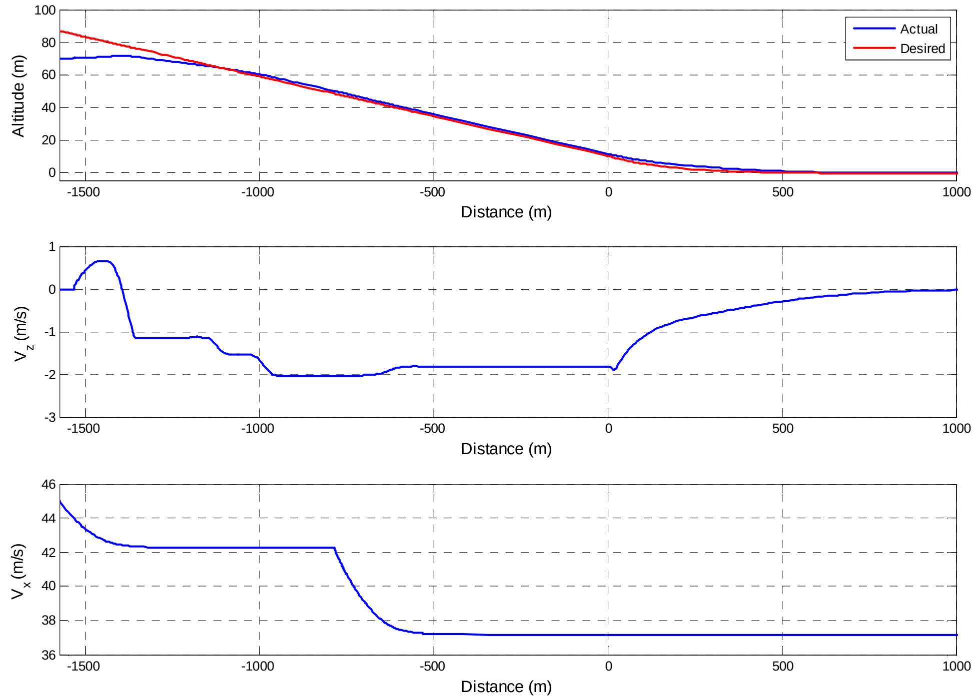



**Figure 6.9: Overall Simulation Results for $V_z$ limit of 4 m/s**

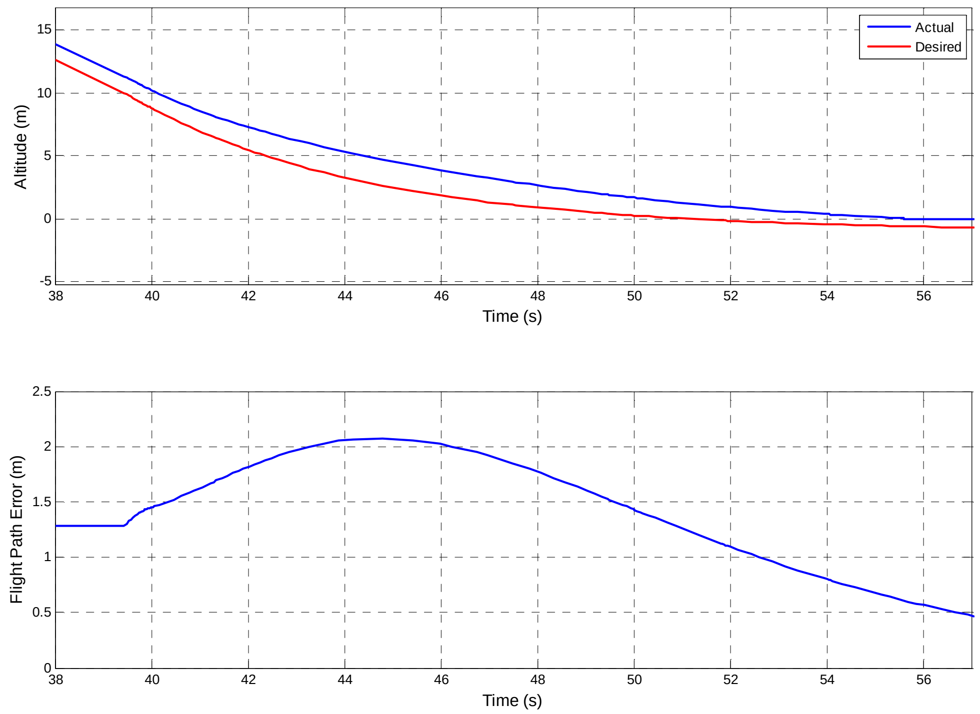



**Figure 6.10: Flare Trajectory for New $V_z$ Limit**

Figure 6.10 shows the flare trajectory and the flight path error for the new simulation. As seen in this figure, the error peaks at 2.1 m during the flare, a 0.9 m peak error improvement on the initial simulation. The change in the controller to accommodate the new $V_z$ control limit requires a change in the output membership function for the FLC.

One of the main reasons for using fuzzy logic in this autolanding controller design is this control method's ability to account for nonlinear and uncertain systems. This claim is validated by simulating the FLC autoland system for varying $V_z/V_{z\text{-cmd}}$ transfer functions. Without changing the FLC for any case, Figure 6.11 through Figure 6.13 show the results of the autolanding FLC for the following transfer functions, from Ref. [27], substituted into the plant model of Eq. (6.3):

#1. $$\frac{V_z}{V_{z_{cmd}}} = \frac{1.185}{s + 0.9159}$$

#2. $$\frac{V_z}{V_{z_{cmd}}} = \frac{4.992}{s + 2.394}$$

#3. $$\frac{V_z}{V_{z_{cmd}}} = \frac{2.099}{s + 1.808}$$

#4. $$\frac{V_z}{V_{z_{cmd}}} = \frac{2.657}{s + 2.281}$$

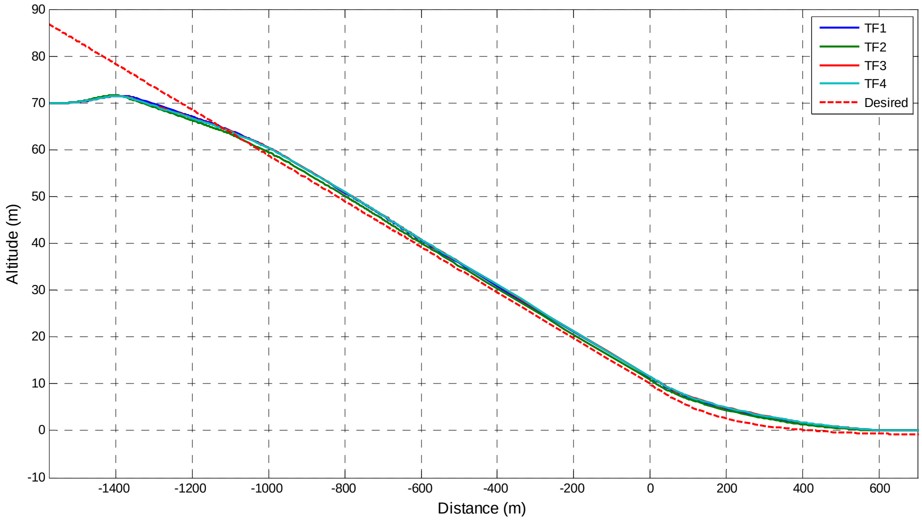



**Figure 6.11: Flight Path for Varying Transfer Functions**

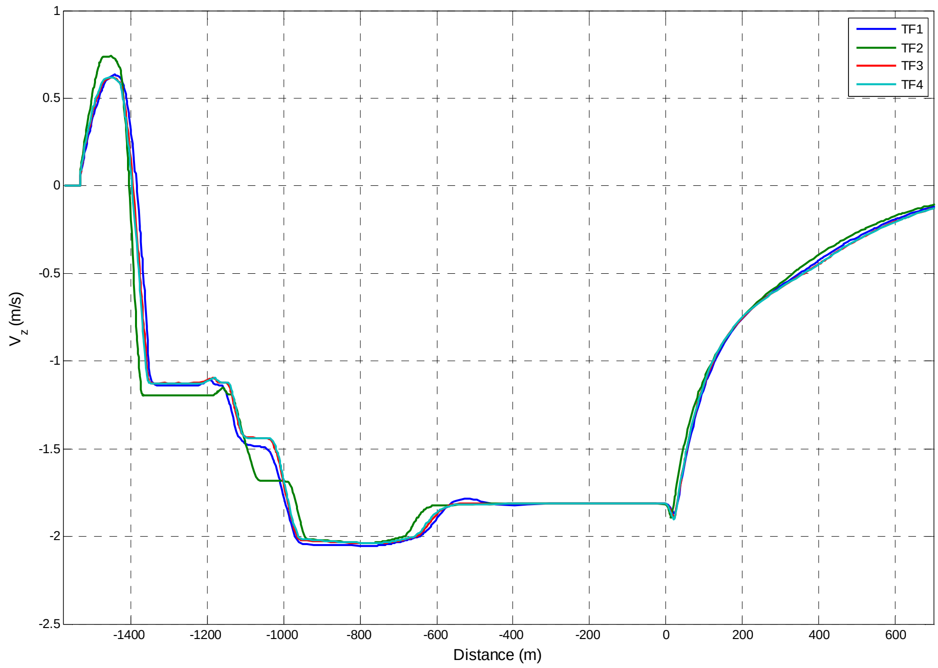



**Figure 6.12: $V_z$ for Varying Transfer Functions**

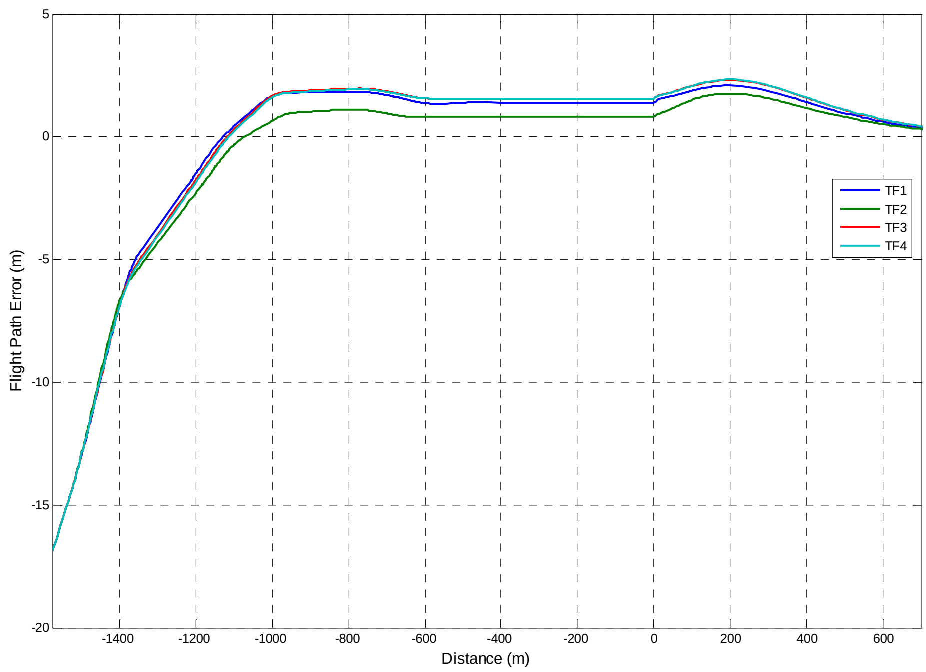



**Figure 6.13: Flight Path Error for Varying Transfer Functions**

These figures show that the FLC, with no modifications, can follow the desired flight path trajectory for varying plant models. These results show the robustness of the FLC design in the case of uncertainty in the plant model as is the case with using the wePilot outer loop control for the automatic landing system.

For realistic simulation of the autolanding FLC where the controller calculates control input for a discrete time interval, rather than continuous calculation as with all simulation thus far, the controller is simulated for sample rates of 50 Hz, 20 Hz, 10 Hz, and 5 Hz. The flight path and flight path error results of this simulation are presented in Figure 6.14 and Figure 6.15 respectively. These results show that the controller performance for the approach phase remains approximately the same at sample rates as low as 5 Hz, but the flare performance is degraded at 5 Hz.

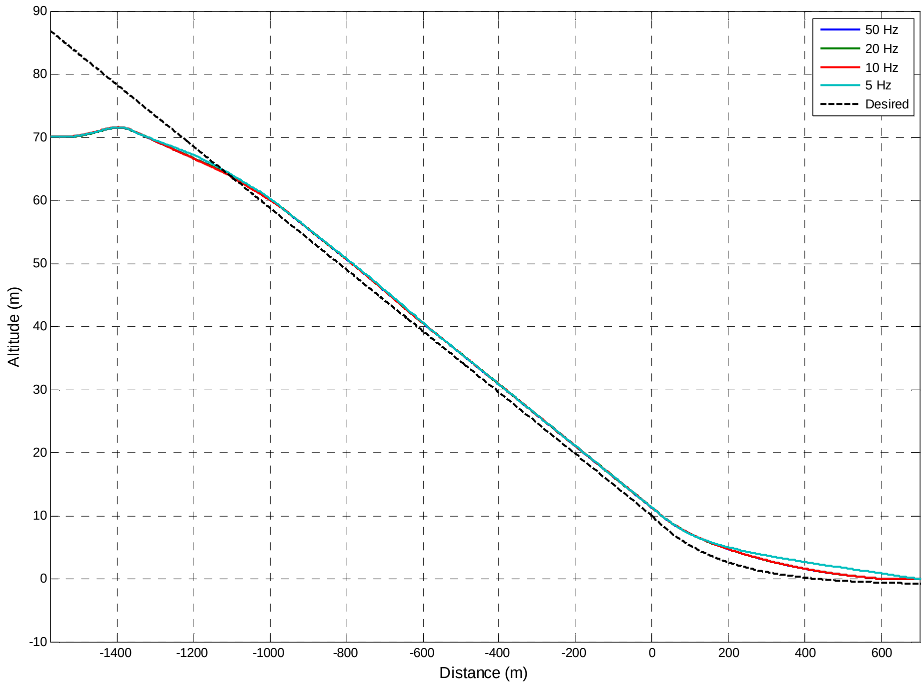



**Figure 6.14: Flight Path for Varying Sample Rates**

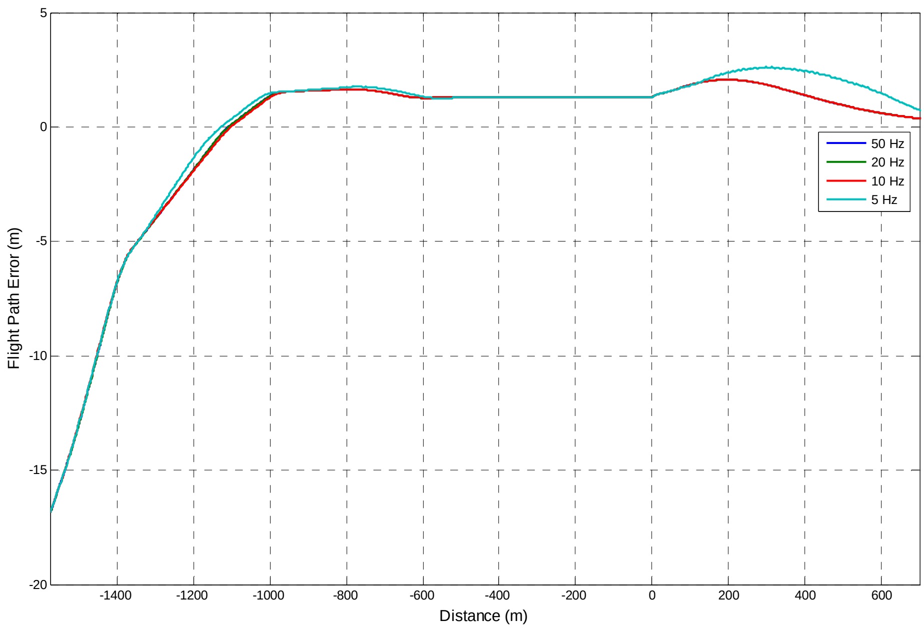



**Figure 6.15: Flight Path Error for Varying Sample Rates**

Figure 6.16 presents the vertical velocity for the autolanding simulation with varying sample rates. This figure shows that for sample rates of both 5 Hz and 10 Hz, the controller requires constant adjustment when capturing the flight path and during the flare maneuver. Based on these results, although the final performance is not degraded, it would not be ideal to run the FLC at a sample rate less than 10 Hz for the flight path capture and flare phases of the automatic landing sequence. However, a sample rate of 5 Hz would be valid for maintaining the constant glide slope in the landing approach after the aircraft converges on the approach trajectory.

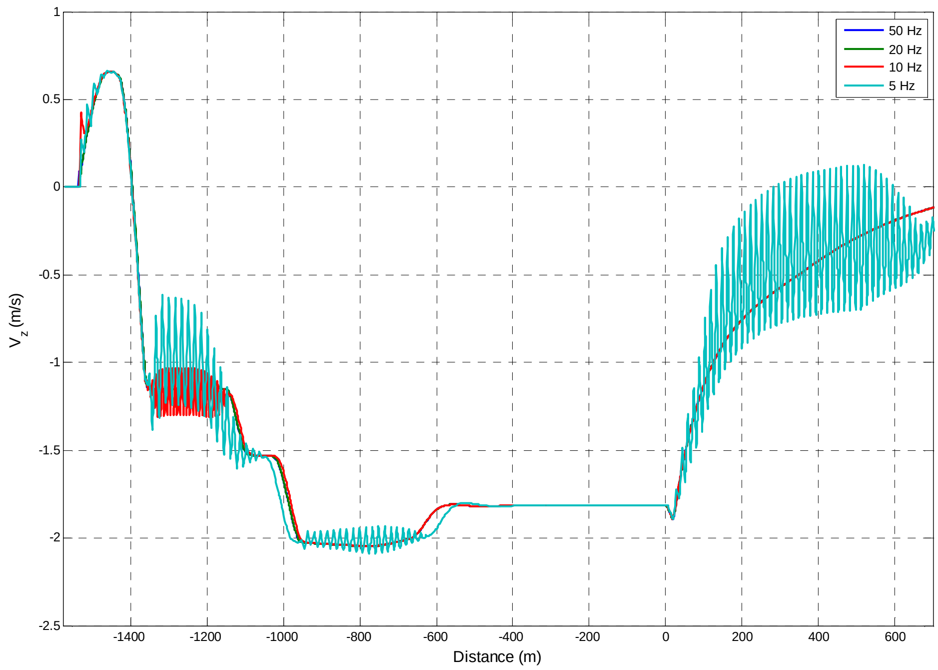



**Figure 6.16: Vertical Velocity for Varying Sample Rates**

Figure 6.17 presents the vertical velocity from the autolanding FLC simulation with sample rates of 50 Hz, 20 Hz, and 16 Hz. In this case, it is shown that a sample rate of at least 16 Hz eliminates the constant controller adjustment seen in the results of Figure 6.16. Based on these results, the FLC automatic landing system would ideally operate with a sampling rate of at least 16 Hz.

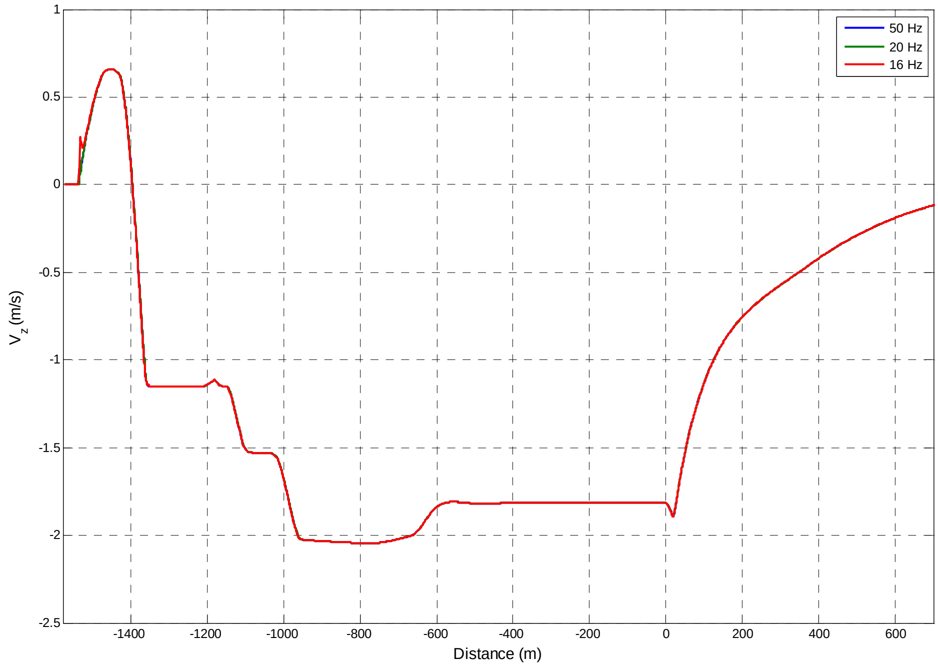



**Figure 6.17: Vertical Velocity Results for Viable Sample Rates**

# 7 Conclusions, Recommendations, and Future Work

## 7.1 Conclusions

Based on the design of an automatic landing system for the Meridian using the wePilot FCS, the following conclusions can be made:

- The phase of flight which poses the highest risk in operation of the Meridian UAV is the takeoff and landing due to the inherent lag introduced by an RC pilot flying in the third person perspective.
- An automatic landing system was designed using fuzzy logic using wePilot outer loop commands. Simulations show the controller adequately follows the desired landing trajectory with an acceptable steady state error which was expected. The controller is shown to perform well in the presence of model uncertainty.
- Fuzzy logic is considered the ideal control theory for this application based on the 'black box' nature of the combined aircraft model and wePilot inner loop controls as the plant model for which the automatic landing controller is designed.
- Due to the lack of direct input to control surfaces, the autolanding system lacks control over the attitude of the aircraft, which prevents the system from controlling the aircraft pitch angle at touchdown and prevents the controller from being useful for rollout after touchdown.

## 7.2 Recommendations

The following recommendations can be made regarding the design of an automatic landing system for the Meridian using fuzzy logic:

- The wePilot high level command limit for vertical velocity should be increased to 4 m/s to allow for the automatic landing to command an approach glide slope of 2.8 degrees as opposed to the current glide slope of 1.15 degrees.
- Flight tests should be planned to observe the attitude of the aircraft during autonomous flight with the wePilot while commanding different magnitudes of vertical velocity. This should give insight into the behavior of the aircraft when approaching the runway in the automatic landing scenario.
- Methods for reducing the steady state error in the fuzzy logic controller should be researched.
- Tests should be done on bringing the throttle to idle and commanding a positive vertical velocity and observing the attitude of the aircraft as the wePilot is given this command. This could potentially allow the autolanding system a method for rotating the tail on touchdown so it may retain control of the aircraft during rollout.
- Testing should be carried out on the behavior of the wePilot in manual mode after switching away from automatic mode tracking of a flight path. If the wePilot maintains the defined flight path despite disturbances, the autolanding system can use the flight path tracking ability of the wePilot for tracking the runway centerline.

### 7.3 Future Work

Based on this research, the following suggestions can be made for future work with the Meridian UAV:

- A precision altitude sensor (i.e. laser altimeter, radar altimeter, etc…) should be tested for use on the automatic landing system for the Meridian UAV.
- The fuzzy logic controller should be scaled down for and tested extensively on the one-third scale Yak-54 including the use of a precision altitude sensor.
- The automatic landing controller should be implemented to a hardware package incorporated onboard the aircraft to have direct communication with the wePilot2000 FCS and not rely on the ground station communication link.
- Flight test of the automatic landing system should first involve automatic landings at varying heights above the runway, starting at 150 meters and incrementally decreasing this test landing height. Close attention should be paid to the attitude of the aircraft at the virtual touchdown point. See Ref. [7] for proven method of flight testing an automatic landing system.
- An automatic takeoff controller should be designed and implemented with the automatic landing system to make the Meridian UAV a fully autonomous system.

# 8 References


1. Biju Prasad B., Dr. S. Pradeep, "Automatic Landing System Design using Feedback Linearization Method," AIAA 2007-2733, AIAA Infotech@Aerospace Conference and Exhibit, Rohnert Park, California, 7-10 May 2007.

2. Wagner, Thomas and Valasek, John, "Digital Autoland Control Laws Using Quantitative Feedback Theory and Direct Digital Design," Journal of Guidance, Control, and Dynamics, vol. 30, No.5, 2007, pp. 1399-1413.

3. Lan, C.E., and Roskam, J., *Airplane Aerodynamics and Performance*, DARcorporation, Lawrence, KS, 2003, pp. 480-486.

4. Devesa, B., Jourdan, C., and Marc, C., "Ground-Effect Identification and Autoland System Validation from Flight Data," Journal of Aircraft, vol. 41, No. 4, 2004, pp. 730-734.

5. Jih-Gau Juang, Kuo-Chih Chin, Jern-Zuin Chio, "Intelligent Automatic Landing System Using Fuzzy Neural Networks and Genetic Algorithm," Proceeding of the American Control Conference, 2004, pp. 5790-5795.

6. Kyungmoon Nho, Ramesh K. Agarwal, "Automatic Landing System Design Using Fuzzy Logic," Journal of Guidance, Control, and Dynamics, vol. 23, No.2, 2000, pp. 298-304.

7. Bauschat, J.M., Mönnich, W., Willemsen, D., and Looye, G., "Flight Testing Robust Autoland Control Laws," AIAA 2001-4208, AIAA Guidance, Navigation, and Control Conference and Exhibit, Montreal, Canada, 6-9 August 2001.

8. Leong, H., 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-2008-6368, AIAA Modeling and Simulation Technical Conference, Honolulu, Hawaii, 18 August, 2008.

9. "*DAR* Corporation," http://www.darcorp.com.

10. Williams, J. E. and Vukelich, S. R., "The USAF Stability and Control DATCOM," Tech. Rep., McDonnell Douglas Astronautics Company, St. Louis, MO, October 1979, AFFDL-TR-79-3032.

11. Roskam, Jan., *Airplane Design: Parts I-VIII*,. DARCorporation, Lawrence, KS, 1997.

12. Sweeten, B. C., Royer, D. and Keshmiri, S., "The Meridian UAV Flight Performance Analysis Using Analytical and Experimental Data," AIAA Infortech@Aerospace Conference and Exhibit, 6-9 April 2009.

13. "Athena Vortex Lattice," http://web.mit.edu/drela/Public/web/avl.

14. Roskam, Jan., *Airplane Flight Dynamics and Automatic Flight Control Part I*, DARCorporation,, Lawrence, KS, 2003.

15. Yechout, T. R., Morris, S. L., Bossert, D. E. and Hallgren, W. F., *Introduction to Aircraft Flight Mechanics*, AIAA Education Series, AIAA, New York, 2003, pp. 368,369.

16. Stengel, R. F., *Flight Dynamics*, Princeton University Press, New Jersey, 2004, pp. 98,99.

17. Anon., "wePilot2000 Datasheet," from Viking Aerospace Web Page, URL: http://www.vikingaero.com/uav_autopilot.html, [cited March 31st, 2010].

18. Jager, R., "Test and Evaluation of the Piccolo II Autopilot System on a One-Third Scale Yak-54," Master's Thesis, Department of Aerospace Engineering, The University of Kansas, Lawrence, KS, 30 April 2008.

19. Royer, D., "Acceptance Test Requirement/Procedure for wePilot2000," UAV Flight Test Safety Board Review, The University of Kansas, Lawrence, KS, 15 July 2008.

20. Leong, E., "Evaluation of Acceptance Flight Test Result for the wePilot Flight Control System on Yak-54 RC," Technical Report, Center for Remote Sensing of Ice Sheets, The University of Kansas, Lawrence, KS, 31 July 2008.

21. Royer, D., "CReSIS Flight Test Program Flight 08-22-08," Technical Report, Center for Remote Sensing of Ice Sheets, The University of Kansas, Lawrence, KS, 29 April 2009.

22. Royer, D., "CReSIS Flight Test Program Flight 08-29-08," Technical Report, Center for Remote Sensing of Ice Sheets, The University of Kansas, Lawrence, KS, 29 April 2009.

23. Royer, D., "CReSIS Flight Test Program Flight 09-17-08," Technical Report, Center for Remote Sensing of Ice Sheets, The University of Kansas, Lawrence, KS, 29 April 2009.

24. Royer, D., "Incident Report for Flight on 9-17-08," Technical Report, Center for Remote Sensing of Ice Sheets, The University of Kansas, Lawrence, KS, 3 October 2008.

25. Royer, D., “CReSIS Flight Test Program Flight 09-26-08,” Technical Report, Center for Remote Sensing of Ice Sheets, The University of Kansas, Lawrence, KS, 29 April 2009.

26. Royer, D., “CReSIS Flight Test Program Flight 11-24-08,” Technical Report, Center for Remote Sensing of Ice Sheets, The University of Kansas, Lawrence, KS, 29 April 2009.

27. Lykins, R., and Keshmiri, S., “System Identification of a COTS Autopilot System Using Flight Test Data,” CReSIS Meridian inner-team document.

28. Roskam, J., *Airplane Flight Dynamics and Automatic Flight Controls Part II*, DARCorporation, Lawrence, KS, 2003, pp. 843-855.

29. Klir, George J., St. Clair, Ute H., Yuan, Bo, *Fuzzy Set Theory: Foundations and Applications*, Prentice Hall, 1997, pp. 73-95.

# Appendix A: AVL Input Geometry File

## File: Meridian.avl

```
#**************************************
#AVL dataset for Meridian wing/tail
#**************************************
MERIDIAN
#Mach
 0.181
#IYsym   IZsym   Zsym
 0       0       0.0
#Sref     Cref    Bref
6.466981   0.804164  8.04164
#
#
#AVL Axes:
#  +X   downstream
#  +Y   out the right wing
#  +Z   up
#
#
#Xref, Yref, Zref is the c.g. location.
#c.g. is 0.157m in front of the wing trailing edge
#root chord is 0.254
#X value (from wing LE) is 0.242-0.157 = 0.085
#Z c.g. is 0.09m
#Xref      Yref   Zref
#0.29845     0.0    0.0
0.16080     0.0    0.0
#
#
#*********************************
SURFACE
RWing
#Nchordwise  Cspace  Nspanwise  Sspace
10           1.0     15      -2.0
#
#================Root section
SECTION
#Xle    Yle    Zle     Chord   AngleOfIncidence
0.0 0.0 0.0 0.804164 0.0  15 -2.0
#
AFILE
```

```
clarky.dat
#
#Set  dcl/da  =  2 pi CLaf, i.e. CLaf = CLalpha/(2*pi)
CLAF
0.9949
#
#Set parabolic drag polar
#CL1 CD1  CL2 CD2  CL3 CD3    |  CD(CL) function parameters
CDCL
0.1759 0.00637 0.4051 0.00615 1.4416 0.01392
#
# =================Define Section where Flap starts
SECTION
#Xle    Yle    Zle     Chord   AngleOfIncidence
0.0 0.402082 0.0 0.804164 0.0  15 -2.0
#
CONTROL
#label   gain xfrac  vx vy vz sgn
Rflap  1.0  0.75  0. 0. 0.  1.
#
AFILE
clarky.dat
#
#Set  dcl/da  =  2 pi CLaf, i.e. CLaf = CLalpha/(2*pi)
CLAF
0.9949
#
#Set parabolic drag polar
#CL1 CD1  CL2 CD2  CL3 CD3    |  CD(CL) function parameters
CDCL
0.1759 0.00637 0.4051 0.00615 1.4416 0.01392
#
#=================Define End of Flap Section
SECTION
#Xle    Yle    Zle     Chord   AngleOfIncidence
0. 2.088134 0. 0.804164 0.  15 -2.0
#
CONTROL
#label   gain xfrac  vx vy vz sgn
Rflap  1.0  0.75  0. 0. 0.  1.
#
AFILE
clarky.dat
#
#Set  dcl/da  =  2 pi CLaf, i.e. CLaf = CLalpha/(2*pi)
CLAF
```

```
0.9949
#
#Set parabolic drag polar
#CL1 CD1  CL2 CD2  CL3 CD3    |  CD(CL) function parameters
CDCL
0.1759 0.00637 0.4051 0.00615 1.4416 0.01392
#
#=================Define Start of Aileron Section
SECTION
#Xle    Yle    Zle     Chord   AngleOfIncidence
0. 2.412492 0. 0.804164 0.  15 -2.0
#
CONTROL
#label   gain xfrac  vx vy vz sgn
Raileron  1.0  0.75  0. 0. 0.  -1.
#
AFILE
clarky.dat
#
#Set  dcl/da  =  2 pi CLaf, i.e. CLaf = CLalpha/(2*pi)
CLAF
0.9949
#
#Set parabolic drag polar
#CL1 CD1  CL2 CD2  CL3 CD3    |  CD(CL) function parameters
CDCL
0.1759 0.00637 0.4051 0.00615 1.4416 0.01392
#
#=================Define wingtip section
SECTION
#Xle    Yle    Zle     Chord   AngleOfIncidence
0.0000  4.02082  0.0000  0.804164    0.  15 -2.0
#
CONTROL
#label   gain xfrac  vx vy vz sgn
Raileron  1.0  0.75  0. 0. 0.  -1.
#
AFILE
clarky.dat
#
#Set  dcl/da  =  2 pi CLaf, i.e. CLaf = CLalpha/(2*pi)
CLAF
0.9949
#
#Set parabolic drag polar
#CL1 CD1  CL2 CD2  CL3 CD3    |  CD(CL) function parameters
```

```
CDCL
0.1759 0.00637 0.4051 0.00615 1.4416 0.01392
#
#**********************************
SURFACE
LWing
#Nchordwise  Ccpace  Nspanwise  Sspace
10           1.0     15      2.0
#
#
#================Define wingtip section
SECTION
#Xle    Yle    Zle     Chord   AngleOfIncidence
0.00  -4.02082  0  0.804164    0.  15 2.0
#
CONTROL
#label   gain xfrac  vx vy vz sgn
Laileron  1.0  0.75  0. 0. 0.  -1.
#
AFILE
clarky.dat
#
#Set  dcl/da  =  2 pi CLaf, i.e. CLaf = CLalpha/(2*pi)
CLAF
0.9949
#
#Set parabolic drag polar
#CL1 CD1  CL2 CD2  CL3 CD3    |  CD(CL) function parameters
CDCL
0.1759 0.00637 0.4051 0.00615 1.4416 0.01392
#
#================Define Start of Aileron
SECTION
#Xle    Yle    Zle     Chord   AngleOfIncidence
0. -2.412492 0. 0.804164 0.  15 2.0
#
CONTROL
#label   gain xfrac  vx vy vz sgn
Laileron  1.0  0.75  0. 0. 0.  -1.
#
AFILE
clarky.dat
#
#Set  dcl/da  =  2 pi CLaf, i.e. CLaf = CLalpha/(2*pi)
CLAF
0.9949
```

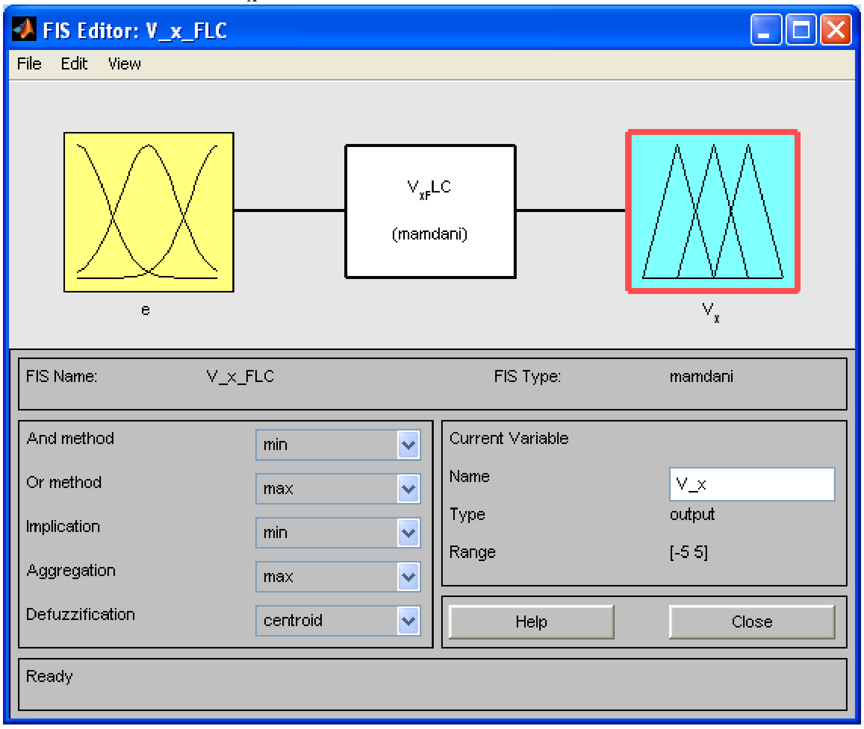
```
#
#Set parabolic drag polar
#CL1 CD1  CL2 CD2  CL3 CD3    |  CD(CL) function parameters
CDCL
0.1759 0.00637 0.4051 0.00615 1.4416 0.01392
#
#==================Define End of Flap
SECTION
#Xle    Yle    Zle     Chord   AngleOfIncidence
0. -2.088134 0. 0.804164 0.  15 2.0
#
CONTROL
#label   gain xfrac  vx vy vz sgn
Lflap  1.0  0.75  0. 0. 0.  1.
#
AFILE
clarky.dat
#
#Set  dcl/da  =  2 pi CLaf, i.e. CLaf = CLalpha/(2*pi)
CLAF
0.9949
#
#Set parabolic drag polar
#CL1 CD1  CL2 CD2  CL3 CD3    |  CD(CL) function parameters
CDCL
0.1759 0.00637 0.4051 0.00615 1.4416 0.01392
#
#=================Define Start of Flap
SECTION
#Xle    Yle    Zle     Chord   AngleOfIncidence
0.0 -0.402082 0.0 0.804164 0.0  15 2.0
#
CONTROL
#label   gain xfrac  vx vy vz sgn
Lflap  1.0  0.75  0. 0. 0.  1.
#
AFILE
clarky.dat
#
#Set  dcl/da  =  2 pi CLaf, i.e. CLaf = CLalpha/(2*pi)
CLAF
0.9949
#
#Set parabolic drag polar
#CL1 CD1  CL2 CD2  CL3 CD3    |  CD(CL) function parameters
CDCL
```

```
0.1759 0.00637 0.4051 0.00615 1.4416 0.01392
#
#=================Root section
SECTION
#Xle    Yle    Zle     Chord   AngleOfIncidence
0.      0.     0.      0.804164    0.  15 2.0
#
AFILE
clarky.dat
#
#Set  dcl/da  =  2 pi CLaf, i.e. CLaf = CLalpha/(2*pi)
CLAF
0.9949
#
#Set parabolic drag polar
#CL1 CD1  CL2 CD2  CL3 CD3    |  CD(CL) function parameters
CDCL
0.1759 0.00637 0.4051 0.00615 1.4416 0.01392
#
#
#***********************************
SURFACE
RTail
#Nchordwise  Cspace   Nspanwise   Sspace
8            1.0       8        1.0
TRANSLATE
3.175 0. 0.
#
#
#
#=================Tail Root Section
SECTION
#Xle    Yle    Zle     Chord   AngleOfIncidence
0.  0.     0    0.686816   0  8  1.0
#
NACA
0009
#
CONTROL
#label   gain xfrac  vx vy vz sgn
Rveetail  1  0.01  0. 0. 0.   1.
#
#
#
#=================Tail tip section
SECTION
```

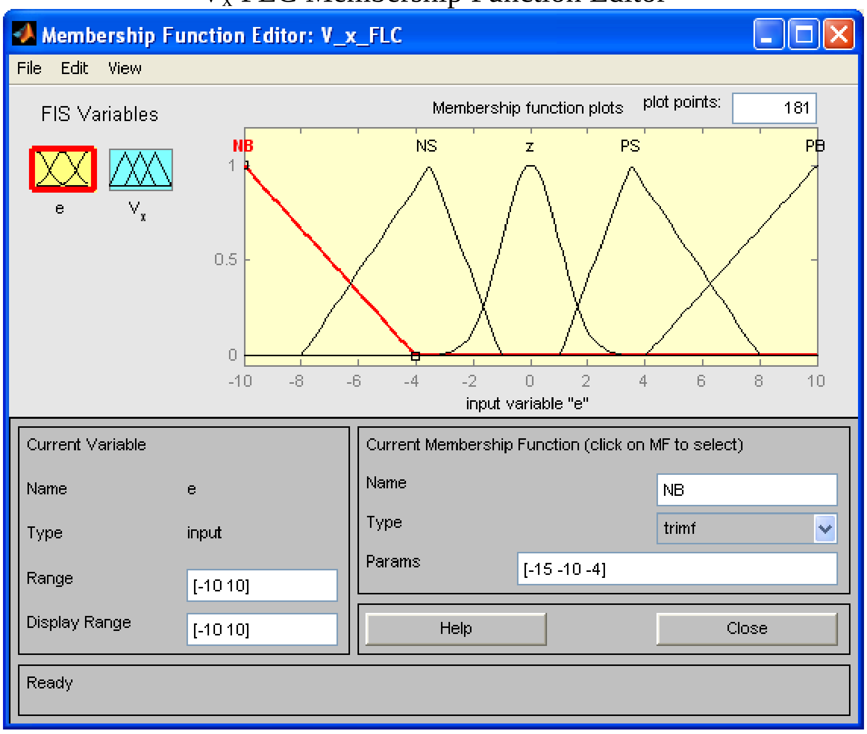
```
#Xle    Yle    Zle     Chord   AngleOfIncidence
0.531114  0.9017   1.074674    0.343408   0  8  1.0
#
NACA
0009
#
#
CONTROL
#label   gain xfrac  vx vy vz sgn
Rveetail  1  0.01  0. 0. 0.   1.
#
#
#**********************************
SURFACE
LTail
#Nchordwise  Cspace   Nspanwise   Sspace
8           1.0       8        1.0
TRANSLATE
3.175 0. 0.
#
#=================Tail tip section
SECTION
#Xle    Yle    Zle     Chord   AngleOfIncidence
0.531114   -0.9017   1.074674    0.343408   0  8  1.0
#
NACA
0009
#
#
CONTROL
#label   gain xfrac  vx vy vz sgn
Lveetail  1  0.01  0. 0. 0.   1.
#
#
#=================Tail Root section
SECTION
#Xle    Yle    Zle     Chord   AngleOfIncidence
0.   0.     0.    0.686816   0  8  1.0
#
NACA
0009
#
#
CONTROL
#label   gain xfrac  vx vy vz sgn
Lveetail  1  0.01  0. 0. 0.   1.
```

## Appendix B: $V_x$ FLC Design in Fuzzy Logic Toolbox

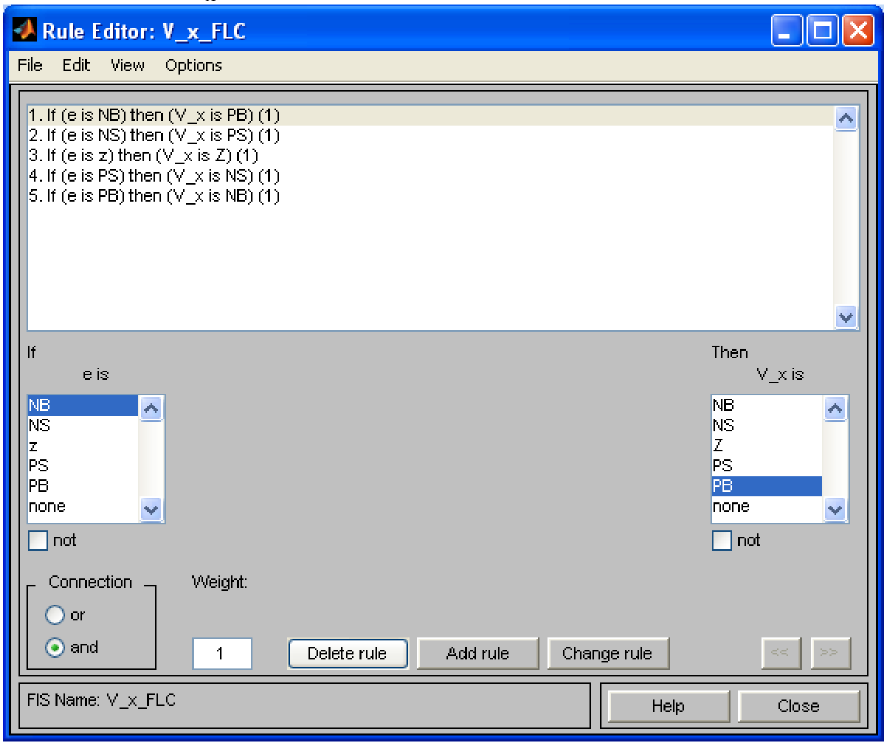
$V_x$ FLC defined in the FIS Editor



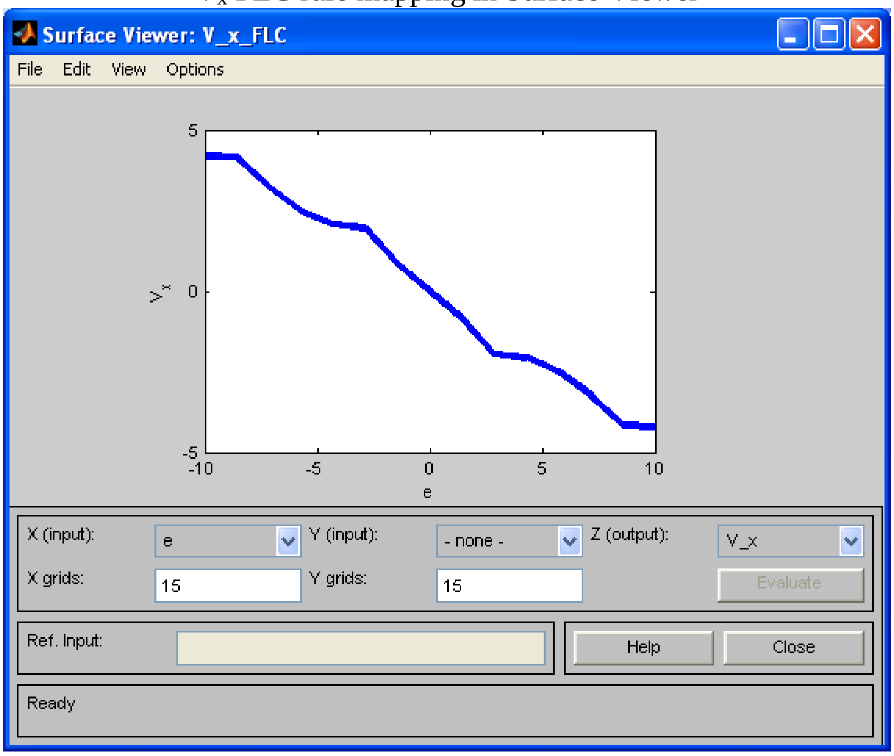

$V_x$ FLC Membership Function Editor



$V_x$ FLC rules entered into the Rule Editor

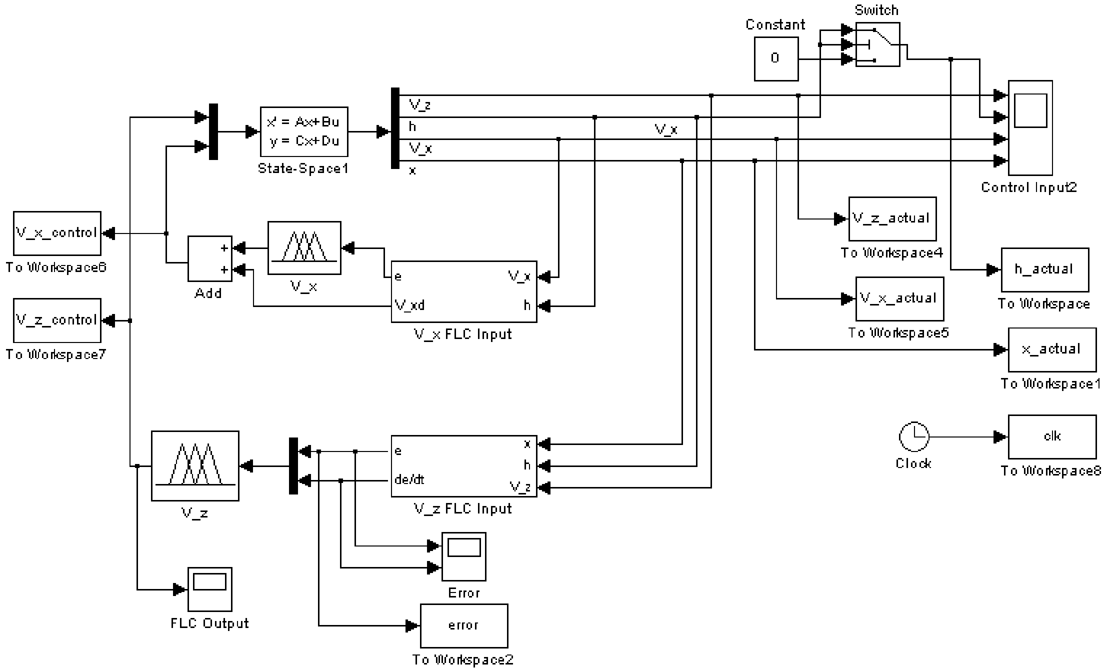



$V_x$ FLC rule mapping in Surface Viewer

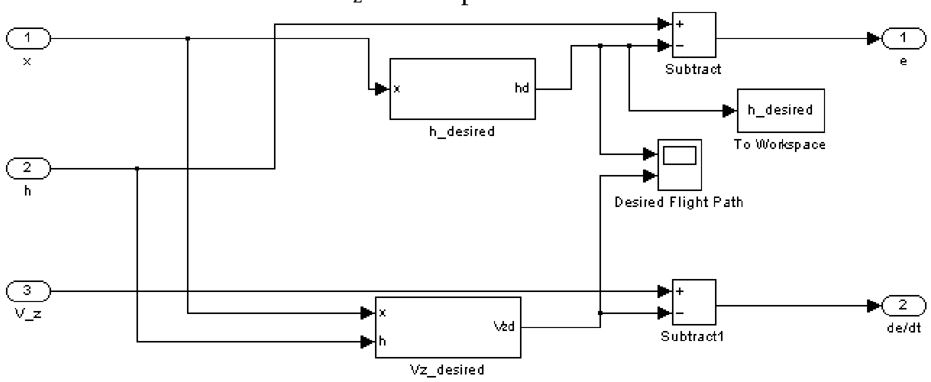


# Appendix C: SIMULINK Model of Autolanding Controller

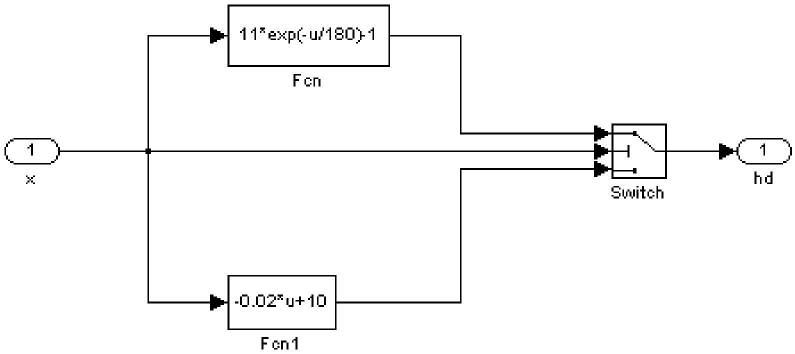
SIMULINK Model for Autolanding Controller Simulation



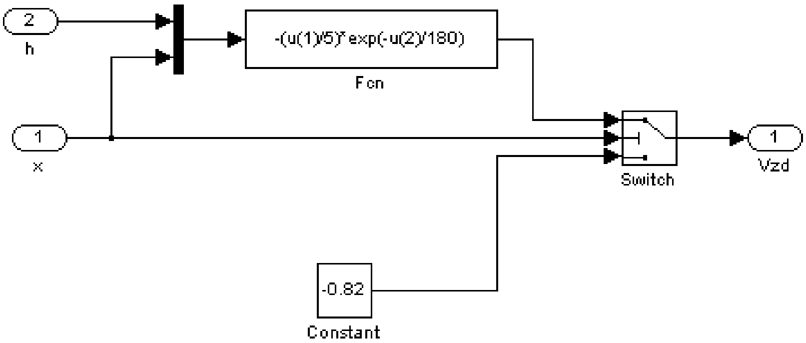

$V_z$ FLC Input Module



Desired Flight Path Module




Desired $V_z$ Module




Vx FLC Input Module