DYNAMIC INVERSE RESILIANT CONTROL FOR DAMAGED ASYMMETRIC AIRCRAFT: MODELING AND SIMULATION

A Thesis by

Michael Arruda

Bachelor's of Science, Wichita State University, 2007

Submitted to the Department of Aerospace Engineering
and the faculty of the Graduate School of
Wichita State University
in partial fulfillment of
the requirements for the degree of
Master of Science

July 2009

©Copyright 2009 by Michael Arruda

All Rights Reserved

DYNAMIC INVERSE RESILIANT CONTROL FOR DAMAGED ASYMMETRIC AIRCRAFT: MODELING AND SIMULATION

The following faculty members have examined the final copy of this thesis for form and content, and recommended that it be accepted in partial fulfillment of the requirement for the degree of Master of Science with a major in Aerospace Engineering.

James Steck, Committee Chair

Gawad Nagati, Committee Member

John Watkins, Committee Member

DEDICATION

To my wife, Emily, for her love and support

ACKNOWLEDGEMENTS

I would like to thank my advisor, James Steck, for his guidance, support, and understanding. To the members of my committee, Gawad Nagati, Kamran Rohksaz, and John Watkins, you have my gratitude for your comments and suggestions. Thanks also to Nhan Nguyen for allowing me to assist in their research. I also thank NASA and the National Space Grant College & Fellowship Program NNX07A391H for providing financial assistance for this research. I am also grateful for the support and instruction from Fuaad at the beginning of my graduate studies.

ABSTRACT

This paper presents the development and simulation of a dynamic inverse resilient control (DIRC) system that can recover a general aviation aircraft that has sustained damage to a lifting surface in flight, and is now asymmetrical. A new set of differential equations of motion for the aircraft are derived that account for the asymmetries of the aircraft and for the dynamic effects of the shift in the center of gravity away from the original body centered axis. The typical symmetric force and moment coefficient buildup is altered to include the influences of asymmetric lift, drag, and control deflection. For the simulation, a Raytheon Bonanaza F-33C fly-by-wire testbed is selected since it has been used in both flight and simulation testing for prior Wichita State University research. Geometric, mass, and aerodynamic properties for the damaged aircraft are determined with conceptual design techniques.

The DIRC consists of an inverse controller with decoupled flight controls and an adaptive system to correct the command signals for a damaged aircraft. Modeling errors between the inverse control and the actual aircraft are large. Adaptive Bias Correctors, a simplified neural network, are applied to adapt the controller to the modeling errors between the undamaged symmetric based inverse controller and damaged asymmetric aircraft. Adaptation happens fast enough to limit the bank angle excursion and bring the aircraft back to wings level flight.

Simulations are performed with a MATLAB/Simulink®and a full 6DOF model of the aircraft. Tests show that for losses to the left wing, the DIRC can recover the aircraft until the available control authority is unable to counteract the asymmetric moments.

TABLE OF CONTENTS

Chapter Page

1 INTRODUCTION 1

1.1 Asymmetric Flight Situations 1
1.2 Dynamic Inverse Resilient Control 2
1.3 Flight Testing & Simulation 3
1.4 Goals & Procedures 3

2 BACKGROUND 4

2.1 Aircraft Flight Damage 4
2.2 Adaptive Control Systems 7
2.3 Local Research Efforts 9

3 ASYMMETRIC AIRCRAFT EQUATIONS OF MOTION 12

3.1 Force Equations 13
3.2 Moment Equations 15
3.2.1 Inertial to Body Center 15
3.2.2 Body Center to Mass Element 16
3.2.3 Center of Gravity to Mass Element Substitution 17
3.3 Perturbed Equations of Motion 19
3.4 Equation Discussion 22
3.4.1 Force and Moment Equations 22
3.4.2 Perturbed Equations 23
3.5 Asymmetric Stability Derivatives 24
3.5.1 Asymmetric Coefficients 24
3.5.2 Control Derivatives 26
3.6 Standard Derivatives 27
3.7 Stability Relationships and Limits of Control 28

4 MODELING AND SIMULATION 32

4.1 Simulation Structure 32
4.1.1 Aerodynamic Model 35
4.1.2 Six Degrees of Freedom Model 36
4.1.3 Damage Initiation 39
4.2 Control Architecture 39
4.2.1 Aircraft Response and Model Follower 40
4.2.2 Inverse Controller 45
4.2.3 Adaptive Bias Correctors 46
4.3 Asymmetric Aircraft Data 48
4.3.1 Geometric and Mass Properties 49
4.3.2 Coefficient Estimations 51
4.3.3 Test Aircraft Data 53

5 RESULTS 58

5.1 Test Aircraft Simulation 58

TABLE OF CONTENTS (continued)

Chapter Page

6 CONCLUSIONS 73

6.1 Equations of Motion 73
6.2 DIRC and Compensator 74
6.3 Further Work / Recommendations 75

REFERENCES 77

APPENDIX 80

A. Non-Critical Condition Results 81
B. Matlab Source Code 87

LIST OF TABLES

Table Page

4.1 Top Level Simulation Blocks . . . 33

4.2 Aircraft Coefficient Switching S-Functions . . . 33

4.3 Relationship Between Compensator Gains and 2nd Order Characteristics . . . 43

4.4 Bank Angle Control - 2nd Order System Characteristics . . . 44

4.5 Sideslip Angle Control - 2nd Order System Characteristics . . . 44

4.6 Compensator Design Bank Angle Results . . . 45

4.7 Bias Coefficients . . . 47

4.8 Damaged Bonanza Geometric Data . . . 51

4.9 Damaged Bonanza Mass Data . . . 51

4.10 Bonanza Geometric Data . . . 52

4.11 Bonanza Mass Properties . . . 52

4.12 Estimated Aircraft Coefficients . . . 52

4.13 Undamaged Aircraft Coefficients . . . 54

6.1 Predicted vs. Actual Aileron Deflections . . . 74

LIST OF FIGURES

Figure Page

2.1 Damage to DHL A300B4 . . . 5

2.2 Bonanza F-33C Three-View . . . 10

3.1 Aircraft and Coordinate Systems . . . 12

3.2 Root Locus Plot for Increasing Left Wing Damage . . . 24

3.3 Wing Roll/Yaw "0" Coefficient . . . 25

3.4 Wing Sideforce "0" Coefficient . . . 25

3.5 CG Shift from Wing Damage . . . 29

3.6 CG Rolling Moment from Wing Damage . . . 30

4.1 Simulation Top Level Diagram . . . 34

4.2 Example Simulink Coefficient Buildup . . . 35

4.3 Euler Angle Simulation Structures . . . 38

4.4 Variable Switching Process . . . 39

4.5 Generalized Aircraft Control Diagram . . . 40

4.6 Simplified Aircraft Control Diagram . . . 41

4.7 Adaptive Bias Corrector Flowchart . . . 47

4.8 Asymmetric Coefficient Determination and Selection . . . 49

4.9 Aerodynamic Chord and Ybar Determination . . . 50

4.10 Shift in Aerodynamic Chord and Ybar . . . 50

4.11 Strip Method for Aileron Control Power . . . 53

4.12 Asymmetric Control Coefficient Damage Values . . . 54

4.13 Asymmetric "0" Coefficient Damage Values . . . 55

4.14 Longitudinal Coefficient Damage Values . . . 56

4.15 Lateral-Directional Coefficient Damage Values . . . 57

LIST OF FIGURES (continued)

Figure Page

5.1 Simulation Results at 20% Damage at 80 Knots 61
5.2 Simulation Results at 25% Damage at 80 Knots 62
5.3 Simulation Results at 30% Damage at 80 Knots 63
5.4 Simulation Results at 25% Damage at 100 Knots 64
5.5 Simulation Results at 30% Damage at 100 Knots 65
5.6 Simulation Results at 35% Damage at 100 Knots 66
5.7 Simulation Results at 30% Damage at 120 Knots 67
5.8 Simulation Results at 35% Damage at 120 Knots 68
5.9 Simulation Results at 40% Damage at 120 Knots 69
5.10 Simulation Results at 35% Damage at 140 Knots 70
5.11 Simulation Results at 40% Damage at 140 Knots 71
5.12 Simulation Results at 45% Damage at 140 Knots 72
A.1 Simulation Results at 20% Damage at 100 Knots 81
A.2 Simulation Results at 20% Damage at 120 Knots 82
A.3 Simulation Results at 25% Damage at 120 Knots 83
A.4 Simulation Results at 20% Damage at 140 Knots 84
A.5 Simulation Results at 25% Damage at 140 Knots 85
A.6 Simulation Results at 30% Damage at 140 Knots 86

LIST OF ABBREVIATIONS / NOMENCLATURE

PV	Position Vector
NASA	National Aeronautical and Space Administration
IS	Inertial System
BRS	Body Reference System
WSU	Wichita State University
ANN	Artificial Neural Network(s)
PD	Proportional-Derivative
PID	Proportional-Integral-Derivative
DOF	Degrees Of Freedom
CG	Center of Gravity
OEI	One Engine Inoperative
DIRC	Dynamic Inverse Resilient Control(ler)
IC	Inverse Controller
FCS	Flight Control System
ABC	Adaptive Bias Corrector

LIST OF SYBMOLS

α	Angle of Attack
β	Sideslip Angle
γ	Flight Path Angle
Θ	Pitch Euler Angle
Φ	Bank Euler Angle
Ψ	Yaw Euler Angle
p	X axis angular velocity (body fixed roll rate)
q	Y axis angular velocity (body fixed pitch rate)
r	Z axis angular velocity (body fixed yaw rate)
u	X axis linear velocity (body fixed)
v	Y axis linear velocity (body fixed)
w	Z axis linear velocity (body fixed)
L	Rolling moment (stability axis)
M	Pitching moment (stability axis)
N	Yawing Moment (stability axis)
$\vec{\omega}$	Angular velocity vector $[p\ q\ r]^T$
$\vec{g}$	Gravitational acceleration vector
δm	Mass element in the aircraft
Δx	X axis Center of Gravity shift
Δy	Y axis Center of Gravity shift
Δz	Z axis Center of Gravity shift
$\vec{R_b}$	PV from IS to aircraft BRS origin
$\vec{R'}$	PV from IS to δm
$\vec{r_b}$	PV from BRS to δm
$\vec{r}$	PV from CG to δm
$\vec{\Delta r}$	PV from BRS to CG
$\vec{\mathbf{F}}$	External forces vector
$\vec{\mathbf{F_a}}$	External aerodynamic forces vector
$\vec{\mathbf{F_t}}$	External propulsive forces vector
$\vec{\mathbf{M_a}}$	External aerodynamic moments vector
$\vec{\mathbf{M_t}}$	External propulsive moments vector

Chapter 1

1 INTRODUCTION

The control of aircraft has been of paramount concern since the beginning of flight. From the first work on gliders by Otto Lilienthal where control was given by shifting the location of the pilot's body, to the wing warping of the Wright Brother's Model 1903 Wright Flyer, it is apparent that regardless of how flight is (or is not) powered, the ability and requirements to control the vehicle in flight change very little. In the Wright Flyer, distinct and separate controls were given to the banking and pitching behavior of the aircraft. In today's aircraft, this fundamental concept stays the same. The pilot still manipulates the separate pitch, roll, and yaw controls, but the job of flying the aircraft has become easier as mechanical systems were introduced to help move the controls with less force. Digital flight control systems were then introduced to further assist control over the aircraft and add further stability to the aircraft so that a pilot's workload for reacting to a flight disturbance is reduced. Over the range of normal flying conditions there exist possibilities of system failures from internal failures (e.g. mechanical issues) and external influences (e.g. bird strike). For these unintended situations, the workload of the pilot increases dramatically. Depending on the situation, human reaction times and capabilities, along with available knowledge of the failure, may render the aircraft unflyable.

1.1 Asymmetric Flight Situations

A majority of aircraft failures involve the loss of engines or the ability to use the actuated control surfaces. In these situations, the coupling between longitudinal and lateral-directional modes of the aircraft are of minimal concern compared to the larger affect of the failure. For instance, an engine inoperative situation will require more rudder and aileron control to counteract the yawing moment created by differential thrust about the aircraft centerline. Depending on the engine location relative to the center of gravity, pitch inputs may be necessary to maintain a flight path angle. However, the control necessary in pitch will be of less concern than control necessary in roll and yaw. It should also be known that engine failures are enunciated to the pilots, and adequate procedures are developed in the design of the aircraft, as per FAA regulations. Control failures within the aircraft may be enunciated only as a hydraulic or electronics failure. Discovering the actual failure is left up to pilots as they test control inputs and the aircraft reaction.

While an engine inoperative situation is an example of an asymmetric flight situation, it is not one of the more severe instances as all the flying surfaces may be intact and operational. In situations where a bird strike, collision with another aircraft, or structural fatigue and fracture causes the loss of flying surfaces, the

coupling of control inputs to aircraft behavior is more extreme. A loss of flying surfaces means decreased lift, which requires more airspeed or a higher angle of attack. If part of a horizontal stabilizer is lost, then pitch inputs will couple with rolling motion. For damaged wings and asymmetric aileron behavior, roll inputs couple with pitching motion. For these failures, the ability of a pilot to adapt quickly is severely limited due to minimal knowledge of the failure and the actual coupled results of control inputs.

1.2 Dynamic Inverse Resilient Control

In the event of any failure, the best outcome would be when the pilot has a prior knowledge of the failure and of the necessary corrective measures to take. Even such precautions for engine failure and cabin depressurization require some safety margin for pilot reaction times. It is then advantageous to develop some method of automatic control that has less delay for reaction time and which can quickly discover the actual result of every controller input. The latter capability can be achieved by the use of adaptive inverse control.

For an undamaged symmetric aircraft, the motion of the aircraft can be described by nine differential equations of motion. Decoupling the longitudinal and lateral-directional equations and linearizing them gives a simple system of linear equations. These equations produce aircraft linear and angular accelerations based on thrust and control inputs. Inverting these equations allows a controller to input the desired aircraft accelerations and be given the required control inputs. This is the basis of Dynamic Inverse Control. Such an inverse, however, will not be exact. The linearization and simplification of the original differential equations of motion leaves some error between the desired and actual aircraft accelerations. Such error can be corrected by including an adaptive system such as an Artificial Neural Network (ANN) as a feedback from the actual aircraft behavior to the inputs of the inverse controller. The adaptive system views the difference between the commanded and actual behavior of the aircraft and adjusts the inverse controller inputs till no error exists in the system.

An adaptive system that can completely eliminate the error will in essence reduce the combination of the inverse controller, aircraft behavior, and adaptive system to unity, where the inputs equal the outputs. An undamaged symmetric aircraft will have only minimal error with respect to its inverse controller. A damaged asymmetric aircraft will have a much larger error with respect to the inverse controller, as the aerodynamic and physical properties changed by the damage are greater in magnitude than errors resulting from linearization. It is then up to the adaptive system to correct the error. If such error correction is possible, than that controller can react as fast as the system sampling rate and adaptation allow and will produce the necessary control surface or thrust input. It is this adaptive system that makes the dynamic inverse controller become resilient to changes in aircraft properties.

1.3 Flight Testing & Simulation

Testing such an adaptive system is difficult to do physically. Test aircraft would have to be sacrificed and pilots put in danger. The ability to flight test an aircraft with realistic simulation is a safer and faster method to test a multitude of flight conditions and obtain pilot feedback. Assembling a realistic simulation of the aircraft requires solving the differential equations of motion with minimal or no simplifications and use a sampling time that is at minimum the same as the sampling time of the aircraft sensors and systems. Physical flight testing and wind tunnel results may be necessary to obtain the real aircraft behavior as opposed to the calculated behavior.

Asymmetric aircraft flight poses another issue for simulation. The standard Euler Angle Six-Degree-of-Freedom (6DOF) mathematics that apply to a symmetric aircraft must be altered for a damaged asymmetric aircraft. Standard simulation is performed about the undamaged aircraft center of gravity (CG), which is also the origin of the body centered axis system. A damaged asymmetric aircraft has lost some section of a flying surface. That loss of mass means a shift in the CG which requires a dynamic evaluation of the affect of the new CG about the original body centered axis system. The consequences of the new dynamics, new aerodynamics, and new control behavior must be accounted for within the simulation of the aircraft before adequate testing of any adaptive system can take place.

1.4 Goals & Procedures

The goal of this thesis will be to develop a Dynamic Inverse Resilient Controller (DIRC) which can save a damaged asymmetric aircraft in the event of the loss of some portion of a lifting surface during flight, with a focus on a General Aviation application. First, the simulation of the aircraft will be developed with the derivation of new differential equations of motion to account for the CG shift off of the original body centered axis. Second, the new aerodynamic and control derivatives of the aircraft will be estimated to give an accurate depiction of a damaged aircraft in flight. Finally, the DIRC will be developed and tested on the newly modeled damaged asymmetric aircraft over a range of flight velocities and failure conditions. These first two procedures, while necessary to adequately test any DIRC, will also provide insight into damaged asymmetric flight behavior. This insight is necessary in understanding how to build a DIRC, and what errors or parameters must be prioritized to control various lifting surface damage. A final evaluation of the DIRC results correlated with the fundamental aircraft behavior will show the capabilities and drawbacks of the control system, and give direction as to where future work should take place. No discussion will occur on the boundedness or certification of the DIRC.

Chapter 2

2 BACKGROUND

Motivation for this research has been briefly mentioned in the Introduction. This Background chapter serves to illustrate specific examples of aircraft flight situations where an control architecture that can adapt to aircraft damage would be a beneficial and life saving addition to a flight control system (FCS). It also discusses past and current research endeavors and their applicability and affect on this research. Finally, research and projects originating at WSU are presented, along with how this research draws from and expands upon that work.

2.1 Aircraft Flight Damage

Numerous failures that can limit the control of an aircraft exist. These can be anything from frozen controls, floating trim tabs, hydraulic failures, engine fan bursts, and bird strikes. For the majority of such failures, the lifting surfaces of the aircraft remain intact, only controls become affected. One of the most famous examples of this scenario is the Sioux City DC-10 crash [1]. United Airline Flight 232 was en route from Denver to Chicago when the No. 2 engine, mounted in the tail, suffered a catastrophic failure and severed all (redundant included) hydraulic controls. The aircraft was then controlled by just the two remaining engines and led to an emergency crash landing in Sioux City, Iowa. 111 people were killed, while 185 people survived. Despite the catastrophic nature of this crash, it did illustrate the ability of an aircraft to be controlled with a lack of standard control surfaces.

This led NASA to work on engine-only flight control systems with an F-15 and MD-11. More information can be found in NASA Technical Memorandum 4798 by Burcham et al. [2] and in its references. It was found that purposely putting some CG shift laterally toward the operative engines on one wing gave the ability to better pitch and roll the aircraft with just throttle manipulations. Such CG shifts were performed by manually operating the fuel management system. The benefit of the lateral offset is discussed further in Section 3.7.

Further use of propulsion only control occurred in 2003 on the shipping company DHL's Airbus A300B4 over Baghdad, as a surface-to-air missile struck the aircraft's left wing. All the hydraulics were disabled, freezing and eliminating any pilot influence over the control surfaces. The pilots were able to crash land the aircraft after adjusting independent throttle commands and shifting fuel to avoid a complete drain of the left wing's fuel stores [3]. A significant portion of the left wing surface was lost from about 10% to 50% span for the aft half of the chord. Photographs of the damage are given in Figure 2.1. These pictures also serve

as an example to the type of damage a wing sustains after an engine separation.

(a) Left Wing, from wing root

(b) Left Wing, aft of wing

Figure 2.1: Damage to DHL A300B4

In 1993, a Boeing 747-121 operated by Japan Airlines Flight 46E (on wet-lease from Evergreen International) experienced severe turbulence upon takeoff and went through some extreme bank, yaw, and air speed fluctuations. The No. 2 engine then separated from the aircraft, removing a significant portion of the wing around the engine pylon. Pilots landed the aircraft back on the Anchorage airport from which they took off, using nearly full aileron and full rudder authority to control the aircraft. Left wing damage consisted

of missing leading edge panels and dented surfaces on the trailing edge flaps and spoilers. Some rudder and stabilizer damage was also seen in the form of dents and scrapes. Information on this flight was provided by the National Transportation Safety Board (NTSB) Aircraft Accident Report [4].

In 2001, an American Airlines Flight 5874, an Airbus A300-605R, suffered an in-flight separation of the vertical stabilizer after departing from John F. Kennedy International Airport in New York. Excessive rudder pedal inputs, to counteract wake turbulence, caused the vertical to exceed its ultimate load capabilities. Other large travels in the control column were recorded as well. The aircraft crashed, and there were no survivors [5]. Other notable aircraft losses of lifting surface are as follows:

- Braniff Airways - Lockheed L-188 - 1968: The right wing and empennage separated in flight after structural limits were exceeded during an attempted recovery from an unusual attitude induced by turbulence. The crash was fatal to all crew and passengers [6].
- Kalitta Air Flight 828 - Boeing 747-132 - 2004: An engine separation removed part of the pylon and wing structure, similar to Flight 46E. The aircraft was able to land safely [7].
- Flying Boat Inc. Flight 101 - Grumman Turbo Mallard - 2005: The right wing separated in flight shortly after takeoff under normal flight loads due to existing fatigue and fracture damage within the right wing. The aircraft crashed into a shipping channel, killing all on board [8].
- Piper PA-60 Aerostar - 2006: Upon flight into thunderstorms, the right wing separated 9 feet and 2 inches outboard of the wing root. The aircraft crashed, killing the pilot [9].
- Beech T-34A - 1999: This aircraft suffered a right wing separation from the wing root during a simulated combat flight and crashed. The cause was most likely fatigue cracking from continued wing loading above design loads. Both pilots were killed [10].
- Piper PA-32 - 2006: The aircraft was piloted into the flight path a Boeing 737 had taken a few minutes prior. Wake turbulence overstressed the aircraft and caused the separation of a section of the left wing, left and right side of the stabilator, and portions of the right wing's flap and aileron. The subsequent crash killed the pilot [11].

While more examples can be found, this list serves to illustrate numerous instances of wing, engine, and control surface separations. Most causes are related to material fatigue or loading beyond design limits by turbulence or wakes. Despite continued efforts for better maintenance and inspection, as well as FAA Advisory Circulars on flight paths of aircraft around wake turbulence, these accidents still occur. It is of critical importance to aircraft safety to develop adaptive methods to recover aircraft from situations where pilots are unable to act fast or accurately enough.

2.2 Adaptive Control Systems

Numerous efforts have taken place over the last few years to begin to solve the problems of controlling an aircraft during large and nearly unrecoverable failures. Control efforts began with parameter identification and adaptive control and with inverse control techniques to predict the required aircraft control surface and thrust settings to achieve certain linear and angular accelerations. Nonlinear effects, unmodeled dynamics, and linearization errors necessitated the use of adaptation schemes to enhance the performance of any inverse control and feedback. These original studies focus more on the ability to use an adaptation technique to avoid the rigorous design procedures that come with gain scheduling. The ability of an adaptive system to be robust to uncertainties either in dynamics, turbulence, or servos has been of major concern (Brinker, Calise w/ missile, etc). Errors can either be adapted for, or using an indirect adaptive approach, unknown system parameters, dynamics and state values are estimated. This thesis is concerned with direct adaptive techniques, which avoids estimating the actual system parameters and instead will adjust system controls or even control laws until feedback errors are removed [12, 13]. Prior work in this focus is reviewed below.

Work by Kim and Calise [14] performed direct adaptive control with an ANN being used for the inverse model of the aircraft. The ANN for the inverse was modeled off-line to map the aircraft dynamics, based on coefficient buildups, to control deflections using radial basis functions and sigma-pi neurons. An on-line ANN was also included to adapt for any inversion errors, and was adapted based on the input to the modeled inverse and the actual aircraft output. Further research on the XV-15 Tilt Rotor [15] established an online training scheme and demonstrated the boundedness of ANN weights. The boundedness of the adaptive weights over a certain range of errors begins to demonstrate the convergence of an adaptive system to a stable solution. Further discussion on bounded weight analysis with various adaptive methods are found in NASA's Integrated Resilient Aircraft Control (IRAC) by Nguyen, Bakhtiari-Nejad, and Huang [13], as well as being found that a Lyapunov based adaptive law has high frequency oscillations when high gain learning occurs, and that other methods of adaptive law may be necessary to avoid high gain learning.

These Model-Reference Adaptive Control system (MRAC) can be extended to highly non-linear flight regimes and use Nonlinear Dynamic Inverse techniques to model the system. Shin et al. [16] developed an adaptive output feedback based on dynamic inversion which was simulated on a modified NASA F-15. Of more importance in this work is the acknowledgment of control saturation in both the unadapted and adapted system, and the ability of such an adaptive system, with control hedging, to exit the saturation in a "graceful manner". For damage rejection and control, the worst case limits of control will be based around deflections near saturation, so the logic around adaptation will be critical in this research.

While most of the research done to date involves simulation in 6DOF environments, NASA's Integrated

Flight Control System (IFCS) work at Dryden has led to the development of the NF-15B, or the F-15 ACTIVE, which is a modified F-15 with F-18 horizontal stabilators placed as canards and with added thrust vectoring. The system is fully fly-by-wire and the canards are used to change the inherent stability of the aircraft. Failures of stabilators and other surfaces allow flight damage management to be tested by various control logic and adaptive techniques. Bosworth et al. [17] have performed flight testing on the F-15 testbed and have demonstrated the effects of pilots in the loop on adaptive systems. For NF-15B stabilator freezing, it was found that as pilots reacted correctly to the failure, ANN weights were never increased to expected adaptation levels since sufficient error did not exist in the adaptive system. Besides large scale flight testing, smaller scale testing is being performed at Georgia Tech by Johnson et al. [18] on a small UAV which will become a testbed for adaptive flight control methods.

Prior to the recent NF-15B tests, NASA and Boeing performed simulation and flight testing on numerous aircraft from transport (B-747 and MD-11) to military aircraft (F-18, F15, and T-38). These tests were performed to demonstrate the flying capabilities of aircraft using propulsion-only control, and began their Propulsion Controlled Aircraft (PCA) system tests on the F-15, and were able to recover from flight control failures, gusts, and upset conditions and subsequently land [19]. Following these tests, PCA flight testing of the MD-11 led to similar results, with the aircraft being able to be both manually controlled and make automatic landings [20]. PCA flight tests on the Boeing-747 again showed the success of the system in landing the aircraft and recovering from gusts and unusual attitudes, such as bank angles of over 90 degrees [21].

Both the Georgia Tech (GT) and NASA Dryden IRAC work is developing toward the control of more than just control surface or sensor failures, but the continued safe flight of aircraft with large losses of lifting or control surfaces. Nguyen et al. [22] takes earlier developed [13] Lyapunov and Recursive Least Squares adaptive laws and tests them on a damaged aircraft. The damaged aircraft is modeled in a vortex lattice code and is referred to as a generic transport model (GTM). Both aerodynamic coefficient changes and the new flight dynamics (equations of motions) are discussed and developed. The new equations of motion are also derived in the UAV test bed development at Georgia Tech [18]. Some discussion of the change in aerodynamic coefficients takes place in the NASA and GT papers, but both do not include some CG shift moment effects. One of the major contributions of this thesis is a further examination of the equations of motion and aerodynamic coefficients of a damaged aircraft.

The before-mentioned research also indicates that aircraft with physical losses of lifting surfaces have much larger parametric uncertainties in the system modeling. A discrepancy of some angle or rate sensor is a minor issue compared to a 25% change or more in critical aircraft inverse parameters such as wing span or lift curve slope. For this reason Nguyen et al. [22] develop a direct-indirect hybrid method to perform both adaptation based on an error with some understanding of the damaged aircraft parameters. Once high-gain

learning was reduced by use of parameter estimation via Recursive Least Squares (similar to results in [13]), this hybrid method provided a better tracking estimate than just a direct adaptive method. For this research, simpler neural networks will be employed to avoid the parameter estimation in order to keep any adaptive system less complex and more practical.

Besides for estimating the damages aircraft parameters by an adaptive method, there has been some focus on an overall FCS that can direct flight planning for damaged aircraft. Tang et al. [23] developed an optimal control LQR and PID based system to plot a trajectory for a damaged aircraft to take toward the safest landing site, accounting for a new estimated flight envelope and capabilities of the damaged aircraft. Idan et al. [24] investigated and simulated secondary channels of flight controls that activate upon a sensed failure. This secondary system is based off the adaptive methods mentioned previously in this section.

2.3 Local Research Efforts

Efforts to apply these adaptive control methods to G.A. have occurred at WSU and with the local aerospace industry, and are discussed separately here, so that the foundations to this research are better illustrated.

Before ANNs became a segment of the research at WSU, Duerksen developed a Fuzzy Logic flight control system that decoupled the flight controls for a General Aviation aircraft [25]. The decoupling separated the flight speed and the flight path angle parameters of longitudinal flight, and reduces pilot workloads and errors [26]. Later, in cooperation with the NASA Small Aircraft Transportation System (SATS) and the Hawker Beechcraft Corporation (previously Raytheon Aircraft Company), an adaptive inverse controller along with a decoupled longitudinal control system were tested both in simulation, and on Bonanza AGATE fly-by-wire test bed [27]. Results of both testing methods showed promising results in the ability for the decoupled controls to reduce workload, as well as for the adaptive system to correct for modeling errors and inaccuracy of gyro or sensor information.

The fly-by-wire testbed Bonanza F-33C is the test aircraft for this research. A 3-View of the aircraft is given in Figure 2.2. Further information on the aircraft is given in Section 4.3. Besides having the capabilities to test adaptive systems safely, with one pilot operating adapted fly-by-wire controls and the other pilot operating traditional mechanical controls, the Beech Bonanza provides operational capabilities that are typical for higher performance general aviation aircraft [28].

The adaptive inverse controller developed by Pesonen et al. [29] for the Bonanza uses the decoupled flight control theory, where the pilot's control inputs distinctly command a flight path angle and a velocity, not throttle or elevator position. The commanded velocity and flight path angle are separate control loops with individual feedback, and the inverse controller takes the commanded accelerations and solves the coupled

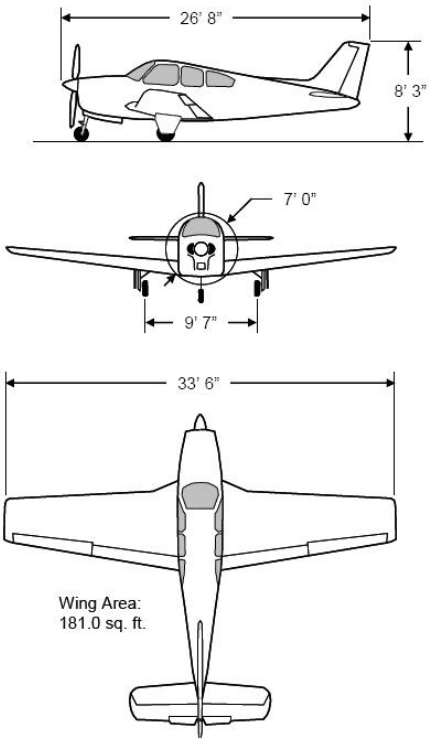


Figure 2.2: Bonanza F-33C Three-View

inverse trim solution of the aircraft to achieve the distinct velocity and flight path angle.

The inverse controller has gone through some maturation since the original development by Steck et al. [27] and Pesonen et al. [29]. Discussion of the current inverse controller is contained in Section 4.2.2. No documented efforts into the mathematical robustness or boundedness of the adaptive methods have been made. However, a turbulence model was tested along with the 6DOF simulation and adaptive inverse controller, yielding good results when a reasonable stall model and limits on aircraft accelerations were implemented [30]. This 6DOF model, as well as continued graduate student work on developing a propulsion assist control system is the simulation foundation for this research.

Of final interest, both for WSU and external research, is the use of the Model Follower to produce desired flight handling qualities and aircraft response to a disturbance or tracking command. A Model Following control structure is discussed in many of the references already cited [13,17] [22], and composes the segment of the FCS that determines the actual flight characteristics of the aircraft. A Proportional-Integral-Derivative filter is usually employed (or only components of one), along with other linear model filters, and combined in the final closed loop transfer function to give explicit relationships between the gains of the compensators, and the 2nd Order System response of the aircraft. Each controlled aircraft parameter, such as flight path angle or sideslip angle, can be compensated separately by their own Model Following structure to give the

best aircraft response.

Another common, and necessary, thread in all of the Model Following usage is the addition of the error corrections from an adaptive system to the output of the model following structure. Only the true aircraft parameters are in the feedback to the prefilter or Model Follower, keeping the adaptive inverse system contained within the open loop of the total control architecture. This is discussed and pictured in more depth in Section 4.2.1.

This thesis will apply the Direct Inverse Adaptive Control concepts developed by Calise and Nhan (many of these prior concepts) and make the following contributions. First, the damaged asymmetric aircraft equations of motion are derived and the new stability and control relationships are discussed. Second, the 6DOF simulation structure at WSU will be updated with the new equations of motion and aerodynamic coefficient buildup changes to allow for simulation flight testing . Third, a Model Following method is applied to define the aircraft response with a decoupled flight control system. Fourth, the Adaptive Bias Correctors learning rates for controllable flight over a range of damage are found. Finally, the results of simulation will show that the DIRC parameters can recover the damaged aircraft for all controllable conditions.

Chapter 3

3 ASYMMETRIC AIRCRAFT EQUATIONS OF MOTION

For a conventional symmetric aircraft, the nonlinear differential equations of motion are defined around the center of gravity of the aircraft with respect to earth fixed coordinates and ignore product of inertia terms I_{xy} and I_{yz}. A damaged asymmetric aircraft requires a re-examination of these equations of motion. First, all product of inertia terms must be included, as the aircraft is no longer symmetric. Second, the center of gravity shifts due to a loss of mass from damage. The standard assumption that the center of gravity is the origin of the body reference axis system no longer applies. This derivation will be solved around the body centered axis which is the original CG location before the onset of damage. Figure 3.1 depicts the coordinate systems for the damaged asymmetric aircraft.

Point **O** is the body centered axis system origin and the CG location before the aircraft is damaged. $\mathbf{R_b}$ is the vector from the inertial reference frame to the body centered axis system. The vector $\mathbf{r_b}$ gives the location of some mass element in the aircraft relative to the body centered axis system. $\mathbf{R}'$ gives the location of the mass element in the aircraft relative to the inertial reference frame. The CG shift is given by the vector $\mathbf{\Delta r}$, and the distance of the mass element to the CG is given by $\mathbf{r}$.

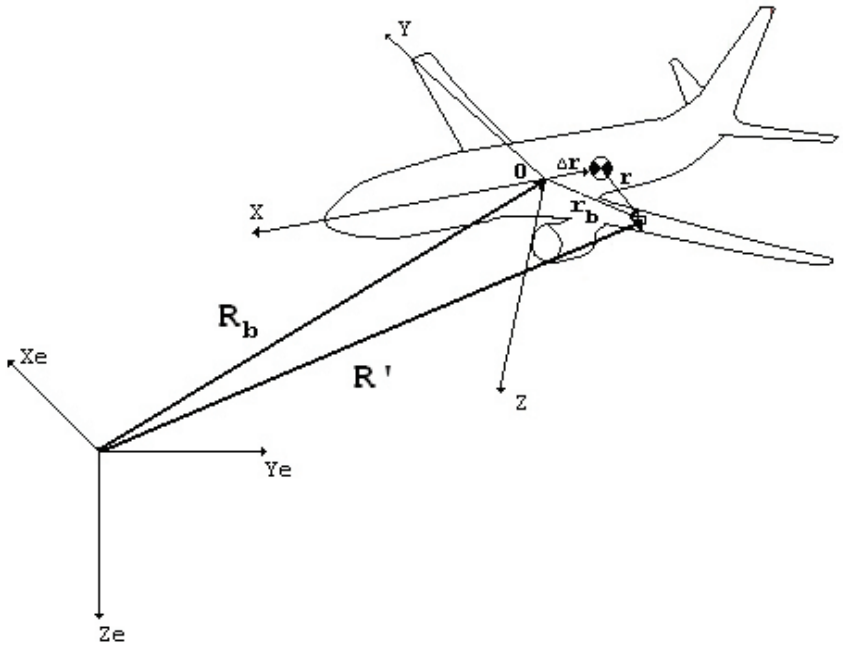


Figure 3.1: Aircraft and Coordinate Systems

3.1 Force Equations

The derivation for the three differential equations describing the forces acting the aircraft will follow a similar method as Roskam [31] uses for a symmetrical aircraft. First, Newton's Second Law for translational momentum is applied to the aircraft. For the derivation, ρ_a is the density of some mass element in the aircraft, and $\vec{g}$ is the acceleration due to gravity vector transformation from the inertial to body axis. $\frac{D}{Dt}$ is defined as an inertial derivative, with $\int_v$ being the integral over the volume of the aircraft and $\int_s$ as the integral about the surface of the aircraft.

$$\frac{D}{Dt}\int_v \rho_a \left(\frac{D\mathbf{R}^{'}}{Dt}\right) dv = \int_v \rho_a \vec{g} dv + \int_s \vec{F} ds \tag{3.1}$$

The vectors $\mathbf{r_b}$ and $\mathbf{R}^{'}$ are defined in Equations (3.2) and (3.3).

$$\mathbf{r_b} = \mathbf{\Delta r} + \mathbf{r} \tag{3.2}$$

$$\mathbf{R}^{'} = \mathbf{R_b} + \mathbf{r_b} \tag{3.3}$$

$$\mathbf{R}^{'} = \mathbf{R_b} + \mathbf{\Delta r} + \mathbf{r} \tag{3.4}$$

Substitution of equation (3.2) into equation (3.3) yields equation (3.4) which describes the location of some mass element on the aircraft in terms of the inertial reference frame. Replacing $\mathbf{R}^{'}$ in equation (3.1) with equation (3.4) gives Newtons's Second Law for the aircraft in terms of the body centered axis and the CG shift.

$$\frac{D}{Dt}\int_v \rho_a \frac{D}{Dt}\left(\mathbf{R_b} + \mathbf{\Delta r} + \bar{\mathbf{r}}\right) dv = \int_v \rho_a \vec{\mathbf{g}} dv + \int_s \vec{\mathbf{F}} ds \tag{3.5}$$

Note that since $\bar{\mathbf{r}}$ is the vector from the CG to mass element δm, the volume integral is equal to zero, as shown in equation (3.6). All other position vectors, $\mathbf{R_b}$ and $\mathbf{\Delta r}$, can be removed from the volume integral since they have no relationship to any mass element. Equation (3.7) shows the result of the volume integral of some mass element density.

$$\int_v \rho_a \ \bar{\mathbf{r}} \ dv = 0 \tag{3.6}$$

$$m = \int_v \rho_a \ dv \tag{3.7}$$

Lastly, ρ_a is assumed to not be time dependent. Once these relationships are applied to equation (3.5),

it simplifies to equation (3.8).

$$m\frac{D}{Dt}\left[\frac{D}{Dt}\left(\mathbf{R_b}+\mathbf{\Delta r}\right)\right]=m\vec{g}+\vec{F_A}+\vec{F_T} \tag{3.8}$$

Total derivatives for $\mathbf{R_b}$ and $\mathbf{\Delta r}$ are given in equations (3.9) and (3.10), respectively. The angular velocity term within the total derivative is $\vec{\omega}$, which is the rotational rate of the body centered axis with respect to the inertial system. For the position vector $\mathbf{R_b}$, there is no $\vec{\omega}\times\mathbf{R_b}$ because it is already in terms of the inertial reference system.

$$\frac{D}{Dt}\left(\mathbf{R_b}\right)=\begin{bmatrix}u & v & w\end{bmatrix}^T=\mathbf{V_B} \tag{3.9}$$

$$\frac{D}{Dt}\left(\mathbf{\Delta r}\right)=\frac{d}{dt}\left(\mathbf{\Delta r}\right)+\vec{\omega}\times\mathbf{\Delta r} \tag{3.10}$$

It is important to note that the term $\frac{d}{dt}\left(\mathbf{\Delta r}\right)$ will be zero. This is because the length $\mathbf{\Delta r}$ does not change over time, as any damage to the aircraft is assumed to occur instantaneously and without subsequent losses of flying surfaces, along with the aircraft being assumed to be a rigid body. Plugging the vector derivatives back into equation (3.8) yields equation (3.11), which describes the forces acting on the aircraft in terms of the body centered axis velocity $\mathbf{V_B}$ and the rotational effects of the CG offset. The surface integral of external forces will be simplified to the sum of the aerodynamic and thrust forces on the aircraft.

$$m\frac{D}{Dt}\left[\mathbf{V_B}+\vec{\omega}\times\mathbf{\Delta r}\right]=m\vec{g}+\vec{F_A}+\vec{F_T} \tag{3.11}$$

The derivative $\frac{D}{Dt}\left[\mathbf{V_B}\right]$ contains both a linear and angular change, as does $\frac{D}{Dt}\left[\vec{\omega}\times\mathbf{\Delta r}\right]$. These are given in equation (3.12) and equation (3.13). Again, $\frac{d}{dt}\left(\mathbf{\Delta r}\right)$ will be zero.

$$\begin{aligned}\frac{D}{Dt}\left[\mathbf{V_B}\right]&=\frac{d}{dt}(\mathbf{V_B})+\vec{\omega}\times\mathbf{V_B}\\&=\dot{\mathbf{V}}_\mathbf{B}+\vec{\omega}\times\mathbf{V_B}\end{aligned} \tag{3.12}$$

$$\begin{aligned}\frac{D}{Dt}\left[\vec{\omega}\times\mathbf{\Delta r}\right]&=\frac{d}{dt}(\vec{\omega}\times\mathbf{\Delta r})+\vec{\omega}\times(\vec{\omega}\times\mathbf{\Delta r})\\&=\dot{\vec{\omega}}\times\mathbf{\Delta r}+\vec{\omega}\times\frac{d\mathbf{\Delta r}}{dt}+\vec{\omega}\times(\vec{\omega}\times\mathbf{\Delta r})\\&=\dot{\vec{\omega}}\times\mathbf{\Delta r}+\vec{\omega}\times(\vec{\omega}\times\mathbf{\Delta r})\end{aligned} \tag{3.13}$$

These results are substituted back into equation (3.11).

$$m\left[\dot{\mathbf{V}}_{\mathbf{B}} + \vec{\omega} \times \mathbf{V_B} + \vec{\dot{\omega}} \times \mathbf{\Delta r} + \vec{\omega} \times (\vec{\omega} \times \mathbf{\Delta r})\right] = m\vec{g} + \vec{F_A} + \vec{F_T} \tag{3.14}$$

Here, $\left[\dot{\mathbf{V}}_{\mathbf{B}} + \vec{\omega} \times \mathbf{V_B}\right]$ is the normal aircraft acceleration term. The gravity vector along the body centered axis is found from the Euler Angle transformation of g from the inertial Z axis (z_e). Performing the linear derivatives and cross products from inside the brackets yields Newton's Second Law for a damaged asymmetric aircraft.

$$m\begin{bmatrix} \dot{u} + \dot{q}\Delta z - \dot{r}\Delta y + q\ w - r\ v - (q^2 + r^2)\Delta x + qp\Delta y + rp\Delta z + g\sin\theta \\ \dot{v} - \dot{p}\Delta z + \dot{r}\Delta x + r\ u - p\ w + pq\Delta x - (r^2 + p^2)\Delta y + rq\Delta z - g\cos\theta\sin\phi \\ \dot{w} + \dot{p}\Delta y - \dot{q}\Delta x - q\ u + p\ v + pr\Delta x + qr\Delta y - (p^2 + q^2)\Delta z - g\cos\theta\cos\phi \end{bmatrix} = \vec{F_A} + \vec{F_T} \tag{3.15}$$

3.2 Moment Equations

The derivation for the three differential equations describing the moments acting the aircraft will follow a similar method as Roskam [31] uses for a symmetrical aircraft. There are numerous vector substitutions, and as such, the majority of the efforts for this derivation are concerned with appropriate simplifications to reduce as many terms as possible. First, Newton's Second Law for angular momentum is applied to the aircraft.

$$\frac{D}{Dt}\int_v \mathbf{R}' \times \rho_a \frac{D\mathbf{R}'}{Dt} dv = \int_v \mathbf{R}' \times \rho_a \vec{g} dv + \int_s \mathbf{R}' \times \vec{\mathbf{F}} ds \tag{3.16}$$

Position vectors $\mathbf{R}'$ and $\mathbf{r_b}$ are defined in Section 3.1 with equation (3.3) and equation (3.2). $\mathbf{R}'$ is substituted into equation (3.16), except for its derivative term.

$$\frac{D}{Dt}\int_v (\mathbf{R_b} + \mathbf{r_b}) \times \rho_a \frac{D\mathbf{R}'}{Dt} dv = \int_v (\mathbf{R_b} + \mathbf{r_b}) \times \rho_a \vec{g} dv + \int_s (\mathbf{R_b} + \mathbf{r_b}) \times \vec{\mathbf{F}} ds \tag{3.17}$$

There are two parts of equation (3.17) that will be examined separately in order to limit confusion and better reduce terms. The $\mathbf{R_b}$ terms will be covered in Section 3.2.1. Section 3.2.2 will cover $\mathbf{r_b}$ terms.

3.2.1 Inertial to Body Center

Evaluating only the $\mathbf{R_b}$ terms from equation (3.17) gives equation (3.18).

$$\left[\frac{D}{Dt}\int_v \mathbf{R_b} \times \rho_a \frac{D\mathbf{R}'}{Dt} dv\right] = \int_v \mathbf{R_b} \times \rho_a \vec{g} dv + \int_s \mathbf{R_b} \times \vec{\mathbf{F}} ds \tag{3.18}$$

Performing the derivative on the first term of equation (3.18) gives the product rule expansion in equation (3.19).

$$\left[\frac{D\mathbf{R_b}}{Dt} \times \int_v \rho_a \frac{D\mathbf{R}^{'}}{Dt} dv\right] + \left[\mathbf{R_b} \times \frac{D}{Dt}\int_v \rho_a \frac{D\mathbf{R}^{'}}{Dt} dv\right] = \int_v \mathbf{R_b} \times \rho_a \vec{g} dv + \int_s \mathbf{R_b} \times \vec{\mathbf{F}} ds \tag{3.19}$$

The last three terms of equation (3.19) forms the cross product of $\mathbf{R_b}$ and equation (3.1).

$$\left[\mathbf{R_b} \times \frac{D}{Dt}\int_v \rho_a \frac{D\mathbf{R}^{'}}{Dt} dv\right] = \mathbf{R_b} \times \int_v \rho_a \vec{g} dv + \mathbf{R_b} \times \int_s \vec{\mathbf{F}} ds \tag{3.20}$$

$$\mathbf{R_b} \times \left(\frac{D}{Dt}\int_v \rho_a \frac{D\mathbf{R}^{'}}{Dt} dv = \int_v \rho_a \vec{g} dv + \int_s \vec{\mathbf{F}} ds\right) \tag{3.21}$$

Equation (3.21) reduces to zero since both sides of the equality are the same. The only remaining terms from equation (3.19) are given in equation (3.22).

$$\frac{D\mathbf{R_b}}{Dt} \times \int_v \rho_a \frac{D\mathbf{R}^{'}}{Dt} dv \tag{3.22}$$

Substituting in for $\mathbf{R}^{'} = \mathbf{R_b} + \mathbf{r_b}$ and using the cross product distributive property gives

$$\frac{D\mathbf{R_b}}{Dt} \times \int_v \rho_a \frac{D\mathbf{R_b}}{Dt} dv + \frac{D\mathbf{R_b}}{Dt} \times \int_v \rho_a \frac{D\mathbf{r_b}}{Dt} dv \tag{3.23}$$

Since a vector cross itself is zero, the first term is zero. Note again that the integral $\int_v \rho_a dv$ only applies to the position vectors referenced to the CG.

$$\frac{D\mathbf{R_b}}{Dt} \times \left(\int_v \rho_a \frac{D\mathbf{r_b}}{Dt} dv\right) \tag{3.24}$$

Equation (3.24) is the final simplification and will be substituted back into equation (3.17) in Section 3.2.3.

3.2.2 Body Center to Mass Element

Evaluating only the $\mathbf{r_b}$ terms from equation (3.17) gives equation (3.25).

$$\left[\frac{D}{Dt}\int_v \mathbf{r_b} \times \rho_a \frac{D\mathbf{R}^{'}}{Dt} dv\right] = \int_v \mathbf{r_b} \times \rho_a \vec{g} dv + \int_s \mathbf{r_b} \times \vec{\mathbf{F}} ds \tag{3.25}$$

$\mathbf{R}^{'}$ is substituted with equation (3.4). The expansion yields

$$\left[\frac{D}{Dt}\int_v \mathbf{r_b}\times\rho_a\frac{D\mathbf{R_b}}{Dt}dv\right]+\left[\frac{D}{Dt}\int_v \mathbf{r_b}\times\rho_a\frac{D\mathbf{r_b}}{Dt}dv\right]=\int_v \mathbf{r_b}\times\rho_a\vec{g}dv+\int_s \mathbf{r_b}\times\vec{\mathbf{F}}ds \tag{3.26}$$

When the product rule is applied to the first term the entire equation becomes

$$\left[\left(\frac{D}{Dt}\int_v \rho_a\mathbf{r_b}dv\right)\times\frac{D\mathbf{R_b}}{Dt}\right]+\left[\left(\int \rho_a\mathbf{r_b}dv\right)\times\frac{D^2\mathbf{R_b}}{Dt^2}\right]+\frac{D}{Dt}\int_v \mathbf{r_b}\times\rho_a\frac{D\mathbf{r_b}}{Dt}dv=\int_v \mathbf{r_b}\times\rho_a\vec{g}dv+\int_s \mathbf{r_b}\times\vec{\mathbf{F}}ds \tag{3.27}$$

Notice that the third and the last term in equation (3.27) are the Moment equations for an aircraft with the CG as the origin of the body centered coordinate system, as all other terms would equal zero if $\mathbf{r_b}$ was the vector from the CG to a mass element.

3.2.3 Center of Gravity to Mass Element Substitution

Equation (3.24) and equation (3.27) are substituted back into equation (3.17).

$$\left[\frac{D\mathbf{R_b}}{Dt}\times\left(\int_v \rho_a\frac{D\mathbf{r_b}}{Dt}dv\right)\right]+\left[\left(\int_v \rho_a\frac{D\mathbf{r_b}}{Dt}dv\right)\times\frac{D\mathbf{R_b}}{Dt}\right]+\left[\left(\int_v \rho_a\mathbf{r_b}dv\right)\times\frac{D^2\mathbf{R_b}}{Dt^2}\right]+$$

$$\frac{D}{Dt}\int_v \mathbf{r_b}\times\rho_a\frac{D\mathbf{r_b}}{Dt}dv=\int_v \mathbf{r_b}\times\rho_a\vec{g}dv+\int_s \mathbf{r_b}\times\vec{\mathbf{F}}ds \tag{3.28}$$

Since cross products are anti-commutative, that is $A\times B=-B\times A$, the first term is simply the opposite of the second term, and they cancel out. Equation (3.28) becomes

$$\left[\left(\int \rho_a\mathbf{r_b}dv\right)\times\frac{D^2\mathbf{R_b}}{Dt^2}\right]+\frac{D}{Dt}\int_v \mathbf{r_b}\times\rho_a\frac{D\mathbf{r_b}}{Dt}dv=\int_v \mathbf{r_b}\times\rho_a\vec{g}dv+\int_s \mathbf{r_b}\times\vec{\mathbf{F}}ds \tag{3.29}$$

Performing the derivative in the second term will allow further simplification, and is shown in equation (3.30).

$$\int_v \frac{D\mathbf{r_b}}{Dt}\times\rho_a\frac{D\mathbf{r_b}}{Dt}dv+\int_v \mathbf{r_b}\times\rho_a\frac{D^2\mathbf{r_b}}{Dt^2}dv \tag{3.30}$$

Within equation (3.30), the first term is a vector cross itself, so it disappears from the equation. Plugging this result into equation (3.29) gives the furthest reduction possible, as shown in equation (3.31), before substituting the components of $\mathbf{r_b}$.

$$\left[\left(\int_v \rho_a\mathbf{r_b}dv\right)\times\frac{D^2\mathbf{R_b}}{Dt^2}\right]+\int_v \mathbf{r_b}\times\rho_a\frac{D^2\mathbf{r_b}}{Dt^2}dv=\int_v \mathbf{r_b}\times\rho_a\vec{g}dv+\int_s \mathbf{r_b}\times\vec{\mathbf{F}}ds \tag{3.31}$$

From here the position vector $\mathbf{r_b}$ is replaced by equation (3.2) so that the change of angular momentum is

expressed in terms of the CG shift off of the body centered axis. Performing this substitution yields equation (3.32).

$$\left[\left(\int_v \rho_a(\bar{\mathbf{r}}+\mathbf{\Delta r})dv\right)\times\frac{D^2\mathbf{R_b}}{Dt^2}\right]+\int_v \mathbf{r_b}\times\rho_a\frac{D^2\mathbf{r_b}}{Dt^2}dv=\int_v(\bar{\mathbf{r}}+\mathbf{\Delta r})\times\rho_a\vec{g}dv+\int_s \mathbf{r_b}\times\vec{\mathbf{F}}ds \tag{3.32}$$

No substitution is made into the last term since $\int_s \mathbf{r_b}\times\vec{\mathbf{F}}ds = M_a + M_t$, that is, equal to the sum of the aerodynamic and thrust moments about the body centered axis. Neither has a substitution been made into the second term, which will be explained shortly.

For clarity, simplifications will be made term by term in equation (3.32). The first term simplifies to equation (3.33).

$$\left(\int_v \rho_a(\mathbf{\Delta r})dv\right)\times\frac{D^2\mathbf{R_b}}{Dt^2}=m\mathbf{\Delta r}\times\frac{D^2\mathbf{R_b}}{Dt^2} \tag{3.33}$$

The second term is simply the rate of change in angular momentum about the origin where $\mathbf{r}$ is centered [32]. Performing the vector substitution would have given the change in angular momentum about the center of gravity plus the change in inertia from the center of gravity to the body centered axis. This gives the same answer, which is expected.

$$I\dot{\vec{\omega}}+\vec{\omega}\times I\vec{\omega} \tag{3.34}$$

The third term in equation (3.32) can be separated into two separate integrals and reduced by equation (3.6).

$$\left[\int_v \rho_a\bar{\mathbf{r}}dv\times\vec{g}\right]+\left[\mathbf{\Delta r}\times\int\rho_a\vec{g}dv\right]$$
$$=[\mathbf{\Delta r}\times m\vec{g}] \tag{3.35}$$

Combining all the terms in equations (3.33) to (3.35) back into equation (3.32) gives the furthest simplification possible to describe the moments about the body centered axis of the aircraft.

$$\left[(m\mathbf{\Delta r})\times\frac{D^2\mathbf{R_b}}{Dt^2}\right]+I\dot{\vec{\omega}}+\vec{\omega}\times I\vec{\omega}=[\mathbf{\Delta r}\times m\vec{g}]+M_a+M_t \tag{3.36}$$

Equation (3.36) is expanded, and the three differential equations that describe the change in angular momentum about the body centered axis are given in equations (3.37) to (3.39).

$$I_{xx}\dot{p}-I_{xy}\dot{q}-I_{xz}\dot{r}+I_{xy}pr-I_{xz}pq+(I_{zz}-I_{yy})qr+I_{yz}(r^2-q^2)+m\Delta y(\dot{w}-qu+pv)-m\Delta z(\dot{v}+ru-pw)$$
$$=mg(\Delta y\cos\theta\cos\phi-\Delta z\cos\theta\sin\phi)+M_{a_x}+M_{t_x} \tag{3.37}$$

$$I_{yy}\dot{q} - I_{xy}\dot{p} - I_{yz}\dot{r} - I_{xy}qr + I_{yz}pq + (I_{xx} - I_{zz})pr + I_{xz}(p^2 - r^2) + m\Delta z(\dot{u} + qw - rv) - m\Delta x(\dot{w} - qu + pv)$$
$$= -mg(\Delta x \cos\theta \cos\phi + \Delta z \sin\theta) + M_{a_y} + M_{t_y} \tag{3.38}$$

$$I_{zz}\dot{r} - I_{xz}\dot{p} - I_{yz}\dot{q} + I_{xz}qr - I_{yz}pr + (I_{yy} - I_{xx})pq + I_{xy}(q^2 - p^2) + m\Delta x(\dot{v} + ru - pw) - m\Delta y(\dot{u} + qw - rv)$$
$$= mg(\Delta x \cos\theta \sin\phi + \Delta y \sin\theta) + M_{a_z} + M_{t_z} \tag{3.39}$$

3.3 Perturbed Equations of Motion

To analyze the dynamic behavior of the aircraft, the Perturbed Equations of motion must be assembled. Any results from this section will not be included in the simulation, but will only illustrate the effect of some loss of lifting surface to the disturbance response of the aircraft. The current six differential equations of motion, along with the three kinematic equations, have nine variables: $u,\ v,\ w,\ p,\ q,\ r,\ \psi,\ \phi,\ \theta$. Each variable can be defined relative to a steady state value plus some small perturbation. For example, the X velocity component u is substituted for $U_1 + \Delta u$, where U_1 is the beginning steady state forward velocity and Δu is some forward velocity perturbation.

The steady state aircraft is assumed to have some nonzero forward velocity and pitch angle, but zero linear velocity, angular velocity, sideslip, and bank angle. Therefore $U_1 = U_1,\ \Theta_1 = \Theta_1$, and $V_1 = W_1 = P_1 = Q_1 = R_1 = \Psi_1 = \Phi_1 = 0$. It is also assumed that products of perturbed terms, such as $\Delta u \Delta r$, are small enough to be considered negligible.

$$m_1(\dot{u} + \dot{q}\Delta z - \dot{r}\Delta y) = -m_1 g \cos\Theta_1 \Delta\theta + f_{a_x} + f_{t_x} \tag{3.40}$$

$$m_1(\dot{v} - \dot{p}\Delta z + \dot{r}\Delta x + U_1\Delta r) = m_1 g \cos\Theta_1 \Delta\phi + f_{a_y} + f_{t_y} \tag{3.41}$$

$$m_1(\dot{w} + \dot{p}\Delta y - \dot{q}\Delta x - U_1\Delta q) = -m_1 g \sin\Theta_1 \Delta\theta + f_{a_z} + f_{t_z} \tag{3.42}$$

$$I_{xx}\dot{p} - I_{xy}\dot{q} - I_{xz}\dot{r} + m_1\Delta y(\dot{w} - U_1\Delta q) - m_1\Delta z(\dot{v} + U_1\Delta r) = -m_1 g(\sin\Theta_1 \Delta y \Delta\theta + \cos\Theta_1 \Delta z \Delta\phi) + m_{a_x} + m_{t_x} \tag{3.43}$$

$$-I_{xy}\dot{p} + I_{yy}\dot{q} - I_{yz}\dot{r} - m_1\Delta x(\dot{w} - U_1\Delta q) + m_1\Delta z\dot{u} = m_1 g(\sin\Theta_1\Delta x - \cos\Theta_1\Delta z)\Delta\theta + m_{a_y} + m_{t_y} \tag{3.44}$$

$$-I_{yz}\dot{p} - I_{yz}\dot{q} + I_{zz}\dot{r} + m_1\Delta x(\dot{v} - U_1\Delta r) - m_1\Delta y\dot{u} = m_1 g\cos\Theta_1(\Delta x\Delta\phi + \Delta y\Delta\theta) + m_{a_y} + m_{t_y} \tag{3.45}$$

$$\Delta p = \dot{\phi} - \dot{\psi}\sin\Theta_1 \tag{3.46}$$

$$\Delta q = \dot{\theta} \tag{3.47}$$

$$\Delta r = \dot{\psi} \cos \Theta_1 \tag{3.48}$$

The terms f_a, f_t, m_a, and m_t describe the perturbed aerodynamic and thrust forces and moments. These are defined in numerous texts [33] [31], and for the purposes of this study, are given in equation (3.49) to (3.54). Thrust perturbations for the forces and moments are assumed to be negligible.

$$f_{a_x} = [X_u \Delta u + X_\alpha \Delta\alpha + X_{\dot{\alpha}} \dot{\alpha} + X_q \Delta q + X_{\delta e} \Delta\delta e + X_{\delta a} \Delta\delta a + X_{\delta r} \Delta\delta r] \tag{3.49}$$

$$f_{a_y} = [Y_\beta \Delta\beta + Y_{\dot{\beta}} \dot{\beta} + Y_p \Delta p + Y_r \Delta r + Y_{\delta a} \Delta\delta a + Y_{\delta r} \Delta\delta r] \tag{3.50}$$

$$f_{a_z} = [Z_u \Delta u + Z_\alpha \Delta\alpha + Z_{\dot{\alpha}} \dot{\alpha} + Z_q \Delta q + Z_{\delta e} \Delta\delta e + Z_{\delta a} \Delta\delta a + Z_{\delta r} \Delta\delta r] \tag{3.51}$$

$$m_{a_x} = [L_\beta \Delta\beta + L_{\dot{\beta}} \dot{\beta} + L_p \Delta p + L_r \Delta r + L_{\delta a} \Delta\delta a + L_{\delta r} \Delta\delta r] \tag{3.52}$$

$$m_{a_y} = [M_u \Delta u + M_\alpha \Delta\alpha + M_{\dot{\alpha}} \dot{\alpha} + M_q \Delta q + M_{\delta e} \Delta\delta e + M_{\delta a} \Delta\delta a + M_{\delta r} \Delta\delta r] \tag{3.53}$$

$$m_{a_z} = [N_\beta \Delta\beta + N_{\dot{\beta}} \dot{\beta} + N_p \Delta p + N_r \Delta r + N_{\delta a} \Delta\delta a + N_{\delta r} \Delta\delta r] \tag{3.54}$$

It is advantageous to assemble the nine differential equations ((3.40) to (3.48)) into a state-space format $\dot{X} = AX + BU$. X is the vector of state variables and U is the vector of inputs. The state vector X is chosen to be $X = [\Delta u\ \Delta\alpha\ \Delta q\ \Delta\theta\ \Delta\beta\ \Delta p\ \Delta r\ \Delta\phi\ \Delta\psi]^T$. However, the current set of equations are linearly dependent and not suitable for state-space in their present form. The Kinematic Equations may be solved simply on their own to find $\dot{\theta}$, $\dot{\phi}$, and $\dot{\psi}$.

$$\dot{\theta} = \Delta q \tag{3.55}$$

$$\dot{\phi} = \Delta p + \Delta r \tan \Theta_1 \tag{3.56}$$

$$\dot{\psi} = \Delta r \sec \Theta_1 \tag{3.57}$$

The remaining equations can be solved a number of ways, however, the analytical result of the determinant of a sixth order matrix gives little useful information about the system. For this reason, a modified state space system will be shown that can be solved analytically. Some further substitutions are necessary to put the remaining equations in terms of the state variables. For small angles, $\Delta w = U_1 \alpha$ so $\dot{w} = U_1 \dot{\alpha}$. Similarly, $\dot{v} = U_1 \dot{\beta}$. For reasons of space, the trigonometric functions $sin\Theta_1$ and $cos\Theta_1$ will be written as S_{Θ_1} and C_{Θ_1}. Equation (3.58)shows the modified state space system $A_1 \dot{X} = A'X + B'U$ matrices. Longitudinal equations (X axis force, pitching moment, Z axis force) are listed first, followed by the lateral-directional equations (Y axis force, rolling moment, yawing moment).

$$\begin{bmatrix} m_1 & -X_{\dot{\alpha}} & m_1\Delta z & 0 & 0 & 0 & -m_1\Delta y & 0 \\ m_1\Delta z & -(m_1\Delta x U_1 + M_{\dot{\alpha}}) & I_{yy} & 0 & 0 & -I_{xy} & -I_{yz} & 0 \\ 0 & (m_1 U_1 - Z_{\dot{\alpha}}) & -m_1\Delta x & 0 & 0 & m_1\Delta y & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & (m_1 U_1 - Y_{\dot{\beta}}) & -m_1\Delta z & m_1\Delta x & 0 \\ 0 & m_1\Delta y U_1 & -I_{xy} & 0 & -(m_1\Delta z U_1 + L_{\dot{\beta}}) & I_{xx} & -I_{xz} & 0 \\ -m_1\Delta y & 0 & -I_{yz} & 0 & (m_1\Delta x U_1 - N_{\dot{\beta}}) & -I_{xz} & I_{zz} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \\ \dot{\beta} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \end{bmatrix} =$$

$$\begin{bmatrix} X_u & X_\alpha & X_q & -m_1 g C_{\Theta_1} & 0 & 0 & 0 & 0 \\ M_u & M_\alpha & (M_q - m_1\Delta x U_1) & m_1 g(S_{\Theta_1}\Delta x - C_{\Theta_1}\Delta z) & 0 & 0 & 0 & 0 \\ Z_u & Z_\alpha & (Z_q + m_1 U_1) & -m_1 g S_{\Theta_1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & Y_\beta & Y_p & (Y_r - m_1 U_1) & m_1 g C_{\Theta_1} \\ 0 & 0 & m_1\Delta y U_1 & -m_1 g S_{\Theta_1}\Delta y & L_\beta & L_p & (L_r + m_1\Delta z U_1) & -m_1 g C_{\Theta_1}\Delta z \\ 0 & 0 & 0 & m_1 g C_{\Theta_1}\Delta y & N_\beta & N_p & (N_r - m_1\Delta x U_1) & m_1 g C_{\Theta_1}\Delta x \\ 0 & 0 & 0 & 0 & 0 & 1 & \tan(\Theta_1) & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta\alpha \\ \Delta q \\ \Delta\theta \\ \Delta\beta \\ \Delta p \\ \Delta r \\ \Delta\phi \end{bmatrix} +$$

$$\begin{bmatrix} X_{\delta e} & M_{\delta e} & Z_{\delta e} & 0 & Y_{\delta e} & L_{\delta e} & N_{\delta e} & 0 \\ X_{\delta a} & M_{\delta a} & Z_{\delta a} & 0 & Y_{\delta a} & L_{\delta a} & N_{\delta a} & 0 \\ X_{\delta r} & M_{\delta r} & Z_{\delta r} & 0 & Y_{\delta r} & L_{\delta r} & N_{\delta r} & 0 \end{bmatrix}^T \begin{bmatrix} \delta e \\ \delta a \\ \delta r \end{bmatrix} \tag{3.58}$$

3.4 Equation Discussion

The set of six differential equations that describe the motion of the aircraft are given in equations (3.15), (3.37), (3.38), and (3.39). In order to verify these equations' validity, the effect of CG position changes will be analyzed, as well as the addition of Products of Inertia that are normally not relevant to a symmetric aircraft.

3.4.1 Force and Moment Equations

The differential equations of force in equation (3.15) contain the CG shift dimensions in the vector $\mathbf{\Delta r}$. Setting the CG shift to zero satisfies the symmetric aircraft assumption that the body centered axis system is located at the CG. In this instance, the three differential equations for force return to the original system of equations. All of the units within brackets in equation (3.15) match the units for acceleration (distance per unit of time squared). Here $\mathbf{\Delta r}$ is a vector of displacements (feet or meters), and rotational rates from $\vec{\omega}$ are in radians per second and their derivatives are in radians per second squared. Finally, the CG offset should not alter the gravity vector's affect on the body centered axis since the original axis system about the CG is only shifted, not rotated. This is confirmed by equation (3.15).

The differential equations of moments in equations (3.37) to (3.39) also contain the CG shift dimensions. The acceleration of the CG along each of the body axes is multiplied by the mass and distance to the body centered axis to find the resulting moments caused by the CG acceleration. A more physically obvious term that appears in the equations is $\mathbf{\Delta r} \times m\vec{g}$, and is simply the moments caused by the CG's offset from the body centered axis. In other words, the component of aircraft weight in the body centered Z axis will cause a moment about the body centered X axis over some distance Δy and a moment about the body centered Y axis over some distance Δx. If no CG shift is present, then these extra moment terms disappear and the differential equations of motion return to the standard symmetric aircraft equations, except for a few terms.

These remaining terms are the inclusion of the Products of Inertia I_{xy} and I_{yz}. In the event of some loss of a flying surface, the symmetry of the aircraft is no longer maintained as these Products of Inertia about the body centered axis system change as the mass distribution of the aircraft changes. Removal of these extra Products of Inertia returns the moment equations to the standard symmetric aircraft equations. It is also noted that these extra inertia terms can create coupling independent of a center of gravity shift. For example, in the event of a loss of a section of the horizontal tail, the pilots may shift fuel. For an extreme scenario, the passengers in the aircraft may attempt to reset the CG to its original location by moving aft. However, this does not mean that the inertia matrix of the aircraft will return to its original value as some existing mass on the aircraft is being shifted over a small distance. For example, in the case of the loss of

some mass in the tail, the value of I_{yy} will be decreased by a magnitude of the tail mass lost multiplied by the longitudinal distance squared to the original body centered Y axis. This longitudinal distance is a few orders of magnitude greater than the longitudinal shift of any passengers necessary to place the CG at the original location.

Both the force and moment differential equations contain aerodynamic and thrust influences. Since the final equations are cast about the original body centered axis system with respect to the inertial system, all of those influences will be taken about the original CG location. This makes their determination less tedious, which will be discussed in more depth in Section 3.5 and Section 4.3.

3.4.2 Perturbed Equations

It is important to first check the effect of the inclusion of the CG shift and Products of Inertia into the Perturbed Equations of Motion. Since the effects on the forces and moments have been validated in Section 3.4.1, it is of little surprise that a similar examination into equation (3.58) yields the same conclusions. In matrix form, the coupling between the longitudinal and lateral-directional modes can be better observed. Removing the CG shift and the asymmetric Products of Inertia takes the six equations and decouples them into the independent longitudinal and lateral-directional modes.

Of further interest in the removal of the coupling terms is the lack of such terms in the A' matrix in equation (3.58) in the pitching moment equation (2nd row). The reason for this is that while the pitching moment (equation (3.38)) does include coupling terms with the X and Z axis velocities, these are assumed to be zero at steady state. So while U_1 appears in the rolling and yawing moment equations, no similar terms appears in the pitching moment. Since the control of the aircraft in this research is concerned with a level cruising condition that satisfies the given initial conditions, the system is adequately defined. If the Y and Z axis velocities or any of the angular velocities are nonzero before some disturbance is encountered (e.g. in a steady state pitching or turning maneuver) then they must be included.

The B' matrix in equation (3.58) allows for the possibility of any control coupling situation and will be discussed in Section 3.5.2.

Analytically solving for the state space system of equations for the symmetric aircraft is not a difficult task. However, analytically solving this coupled system of six equations for the asymmetric aircraft becomes tedious and the result is confusing. Using MATLAB software, the state space matrices can be determined numerically by multiplying the modified state space system ($A_1\dot{X} = A'X + B'U$) by A_1^{-1}. Figure 3.2 plots the shift in the eigenvalues of the test aircraft for various percentage losses of the wing. The poles are labeled for the equivalent modes in a decoupled longitudinal and lateral-directional system. It will be noted that for the test aircraft damage, the $\mathbf{\Delta r}$ terms are small and keeps the influence of coupling to a minimum. The

asymmetric damaged aircraft stability derivative values are responsible for most of the shift in the poles.

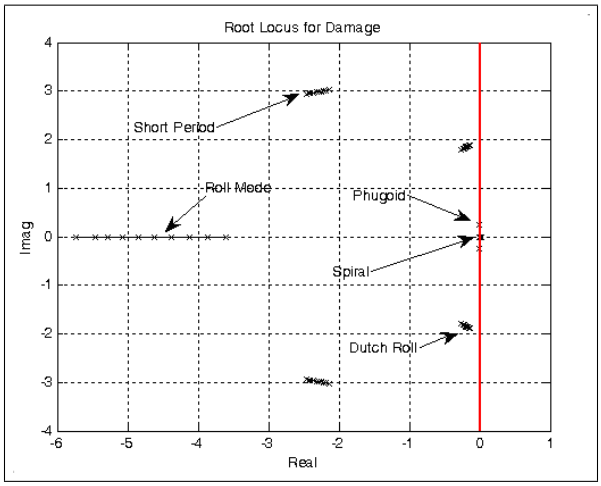


Figure 3.2: Root Locus Plot for Increasing Left Wing Damage

Note: Plot shows Root Locus at 80 knots airspeed.

From Figure 3.2 it can be seen that increasing wing damage begins shifting poles toward the right hand plane. The Spiral mode pole moves to the positive real axis slowly. Since wing segments are lost, the Roll mode is affected the most and the damping in roll capabilities of the aircraft are greatly reduced. Longitudinal responses are slightly less stable, but since no portion of the horizontal tail is lost, this is to be expected.

3.5 Asymmetric Stability Derivatives

While evaluation of the Perturbed Equations of Motion and the subsequent development of the damaged asymmetric aircraft state space model provides an analysis of the disturbance response of the damaged aircraft and the stability of these responses, the aerodynamic and thrust force and moment models of the system have not been defined. The force and moment buildup seen in numerous references [31] [34] [33] all make symmetric assumptions about the zero sideslip rolling moment, yawing moment, and sideforce stability derivatives. Control coupling may occur, requiring the determination of new control derivatives. It will be the job of Section 4.3 to discuss the details of finding the numeric values for these derivatives. This section will discuss what new derivatives must be determined and some conceptual methods for their estimation.

3.5.1 Asymmetric Coefficients

For an undamaged symmetrical aircraft, the zero sideslip rolling moment, yawing moment, and sideforce coefficients (C_{l_0}, C_{n_0}, C_{y_0}) will be zero. As some section of a lifting surface is lost asymmetrically, the distribution of lift and drag on the wings and tail is no longer equal. The undamaged half of the aircraft

will produce more lift than the damaged half, and therefore more drag. These differences in lift about the body centered X axis define C_{l_0}. Drag differences act parallel to the X axis, but they will create a moment about the body centered Z axis that will define C_{n_0}. Both of these differences are pictured in Figure 3.3.

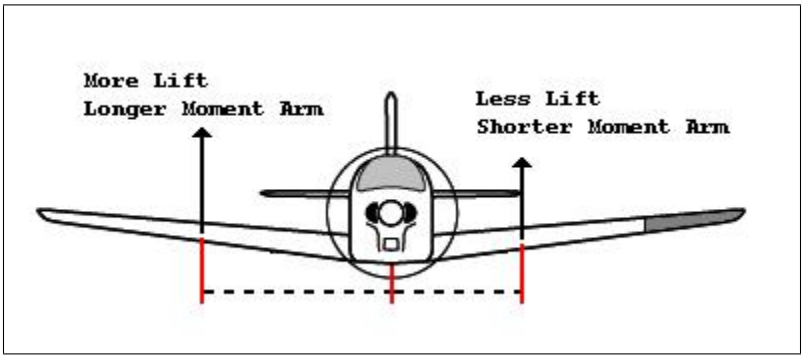


Figure 3.3: Wing Roll/Yaw "0" Coefficient

Note: Only the vertical component of lift is depicted.

For a straight wing with no dihedral, there will be little side force change for loss of some lifting surface. With dihedral, lift shares components along the Z and Y body centered axis, as is depicted in Figure 3.4. As one wing produces more lift than the damaged section, the Y axis projection of the lift vector becomes unbalanced on either side. This imbalance defines C_{y_0}.

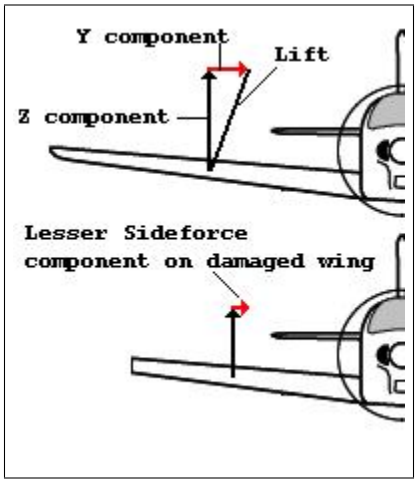


Figure 3.4: Wing Sideforce "0" Coefficient

Values for these "0" Coefficients are based upon the flight speed and the angle of attack. Given some loss of lifting surface, the damaged surface will have a shift in the aerodynamic center to a new value of ybar, as seen in Figure 4.9 using a simple root and tip chord estimation method. The damaged surface's lift acting over the new ybar produces a smaller magnitude and opposite moment than the undamaged surface's lift. Dividing this difference by qSb gives C_{l_0}, which means that C_{l_0} is essentially a nondimensionalized lift times

a moment arm.

$$C_{l_0} = \frac{(C_{L_\alpha}\alpha\ qS)\ \Delta y_{bar}}{qSb} \tag{3.59}$$

Equation (3.59) shows a simplified C_{l_0} to illustrate some of the expected results. For a change in airspeed, the numerical value of C_{l_0} will not change, as q is present in both the numerator and denominator of equation (3.59). However, the asymmetric rolling moment will be greater as airspeed increases since the lift will be greater. For a change in angle of attack, the numerical value will change accordingly. This same trend will be evident with C_{n_0} and C_{y_0}.

The determination of C_{n_0} is the same as that for C_{l_0}, except the wing drag will be substituted for wing lift. C_{y_0} may be found by a similar determination of lift from flight speed and angle of attack, except that instead of nondimensionalizing a moment, only the difference between the lift vectors Y axis component is nondimensionalized. The result of this simplified analysis of the "0" Derivatives is that numerical values do not need to be recalculated if the flight velocity changes, but new values do need to found for changes in the aircraft angle of attack.

3.5.2 Control Derivatives

There are numerous possible control derivatives that must be added for a damaged asymmetric aircraft. These have already been briefly described in Section (3.4.2) through the development of the modified state space B' matrix. The decoupled behavior of damaged control surfaces will be discussed first. For a decoupled system, the associated control derivatives are described in the standard aircraft state space's control power matrix.

$$\begin{bmatrix} X_{\delta e} & M_{\delta e} & Z_{\delta e} & 0 & 0 & 0 \\ 0 & 0 & 0 & Y_{\delta a} & L_{\delta a} & N_{\delta a} \\ 0 & 0 & 0 & Y_{\delta r} & L_{\delta r} & N_{\delta r} \end{bmatrix}^T \begin{bmatrix} \delta e \\ \delta a \\ \delta r \end{bmatrix} \tag{3.60}$$

For horizontal or vertical tail damage that decreases the size of the elevator or rudder, the new decoupled behavior is not difficult to find. The control effectiveness factor [31] will remain the same for any amount of damage, since this property is based on the chord ratio of the control surface to its lifting surface. The aspect ratio change of the horizontal or vertical will decrease the lift curve slope, and therefore reduce the final effectiveness of the elevator or rudder. The standard aileron behavior can be found by performing a strip build up method on both the ailerons.

The coupled control derivative terms are those that affect a mode that is not standard for the control surface. The elevator, for instance, normally has no affect on the lateral-directional modes of the aircraft. An asymmetric elevator about the body centered X axis, however, has a difference in lift increment on either

side of the X axis that will cause a rolling moment. This effect is described by the derivative $L_{\delta e}$ and can be found by a similar method as finding C_{l_0}, where the lift over the horizontal tail after some elevator deflection is split into component on either side of the body centered X axis. Similarly, $N_{\delta e}$ can be calculated as the difference between drag increments on either side of the Z axis. $Y_{\delta e}$ may be found as some changed interference affect between the horizontal and the fuselage and vertical tail, but could be assumed to be negligible.

An asymmetric loss of part of the ailerons will change the effects of the aileron pitching moments, which causes the coupled aileron terms. It will be assumed that the ailerons will still deflect to equal and opposite degrees. The trailing edge up deflection of one aileron versus the trailing edge down deflection of the other can no longer be assumed to have a negligible pitching moment change. $M_{\delta a}$ can then be determined by finding the difference between the pitching moment increments for some aileron deflection. It is also originally assumed that the equal and opposite aileron deflections will cause a decrease of lift on one wing and an increase on the other wing that canceled each other out. Since the overall lift will not change, the loss of altitude caused by aileron deflection, without any other control inputs, is then just a result of tilting the airplane's lift vector off of the inertial Z axis. In the damaged asymmetric scenario, the unmatched aileron sizes will cause either an increase or decrease in the aircraft lift. Again, flap properties can be used to develop the term $Z_{\delta a}$ and similarly with the drag differences, $X_{\delta a}$.

Coupled rudder behavior may be treated if desired. Rudder deflection drag may already be considered on the aircraft, and if not, there is little reason to consider adding it for rudder damage, as the increased vertical surface drag from a lowered aspect ratio will be of larger magnitude and more readily calculated. For a conventional aircraft with the vertical perpendicular to the Y axis, no pitching moment can be created, save for the bending of the vertical or wash effects over the horizontal. Those aircraft with V-tail configurations will have such pitching considerations, but this will not be explored in this research. Nor will the issues from mixed controls be discussed.

3.6 Standard Derivatives

The standard derivatives for the aircraft may be considered as those found in a symmetric aircraft's force and moment coefficient build up [33] [31]. A buildup for lift is given as an example in equation (3.61), ignoring the control terms as these have already been discussed separately. For these derivatives, the primary inputs are based on the lifting surfaces' airfoil characteristics, geometric parameters (such as aspect ratios and moment arms about the CG), and flight Mach number.

$$C_L = C_{L_0} + C_{L_\alpha}\alpha + C_{L_q}\frac{qc}{2U_1} + C_{L_{\dot\alpha}}\frac{\dot\alpha c}{2U_1} \tag{3.61}$$

Numerous methods may be utilized to find these standard derivatives, from vortex lattice codes to linear DATCOM [35] estimations. Regardless of which method provides the new values, the same underlying principles are affecting these derivatives. The first of these principles is the aspect ratio change of the damaged lifting surface. Any damage to a lifting surface will be assumed to occur in spanwise segments, so span is the largest dimension influenced. There will be some change in the mean aerodynamic chord of the lifting surface that must be calculated, and this change along with a smaller span will give the lifting surface a decreased aspect ratio. The finite lift curve slope for the surface will then decrease from the original value and some performance capabilities of the original aircraft will be lost.

The second underlying principle is the volume ratios of the tail surfaces. If wing damage is the only incurred failure, then no extensive work into the tail effectiveness is necessary, unless the wing failure becomes large enough that downwash will significantly change over one half the tail surfaces. However, the values of terms the stability derivatives such as C_{L_α}, C_{M_α}, and C_{N_β} include tail lift curve slopes that are ratioed by the the tail surface area over the wing reference area. As the wing reference area decreases from damage, the tail contributions to the before mentioned terms will increase, *but in coefficient value only.* If these terms are examined apart from the wing contributions, the force that the tail actually creates can be found from the following derivation. The horizontal tail will be used as an example, but the same result will apply for any tail surface.

$$\begin{aligned} L_{tail} &= [C_{L_{tail}}]\ q_{bar}\ S_{ref} \\ L_{tail} &= \left[C_{L_{\alpha tail}}\ \alpha_{tail}\ \frac{S_{tail}}{S_{ref}}\right] q_{bar} S_{ref} \\ L_{tail} &= C_{L_{\alpha tail}}\ \alpha_{tail}\ S_{tail}\ q_{bar} \end{aligned} \tag{3.62}$$

From equation (3.62) it can be seen that the actual tail force doesn't depend upon the reference area, only the area of the tail surface. The ratio of areas is applied to the tail lift coefficient to allow that coefficient value to be included in a combined aircraft C_{L_α}. Another consequence of this ratio is that other tail based derivatives (such as estimations for $C_{M_{\dot\alpha}}$ or C_{n_r}) must be updated for wing damage.

3.7 Stability Relationships and Limits of Control

Finding the new force and moment build up help reveal some physical limits and behaviors to the recovery and control of the aircraft after the onset of damage. These limits can be insurmountable, even assuming

the implementation of a perfect inverse resilient controller. What follows in this section are some overviews of the stability relationships between failures and the limits of control. This will help set success criteria for the controller.

For most losses of lifting surfaces, the CG will shift forward along the body centered X axis since for most aircraft as the original CG location is located close to and slightly forward of the center of mass of the wings [34]. It is apparent that any mass lost in the tail will shift the CG forward. The neutral point for the aircraft, where the change in pitching moment per change in angle of attack is zero, will move aft some small percentage of chord for some wing loss. Any loss of the tail will shift the neutral point forward but will not necessarily destabilize the aircraft in pitch, as the forward CG shift may maintain stable pitch stiffness. An advantage to some loss of pitch stability is the increased control power capability of the remaining tail control surfaces since there is less natural stability to overcome.

The asymmetric loss of wing area may only minutely shift the CG along the X axis except in aircraft with a large wing sweep angle, but the CG shift along the Y axis will be of significant consequence. Figure 3.5 shows a representative CG shift along the X and Y axis for a percentage of left wing damage in a minimally swept wing, like that of the test aircraft. Estimation of these CG shifts are described in Section 4.3.

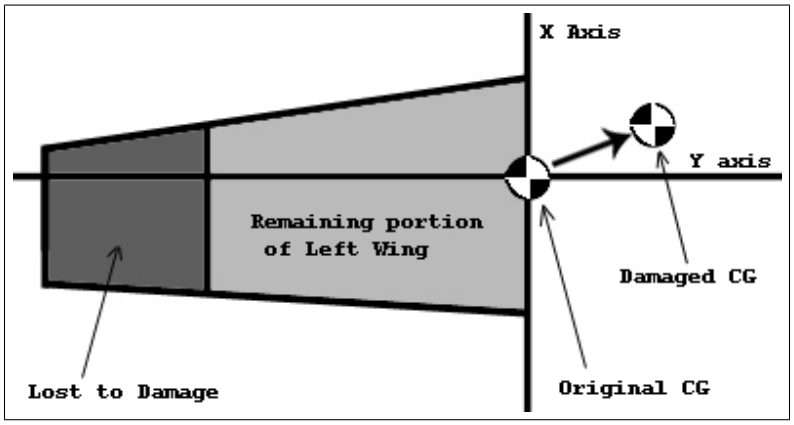


Figure 3.5: CG Shift from Wing Damage

In this failure mode, the CG shift will actually aid in reducing the asymmetric roll coefficient defined as C_{l_0}. Figure 3.6 illustrates this point in the body centered Y axis. The right half side of equation (3.37) presents the same conclusion. Any loss to the left wing will shift the CG some positive Δy and the resultant lift distribution about the body centered x axis will create a negative C_{l_0}. Similarly, losses to the right wing will give a negative Δy and a positive C_{l_0}.

Despite the aid in reducing the effects of C_{l_0}, the loss of wing weight and the small Δy will counteract the aerodynamic rolling moment whose moment arm is much longer. At some percentage of wing damage and loss of aileron control power, maximum deflection of the ailerons will no longer be able to hold the aircraft

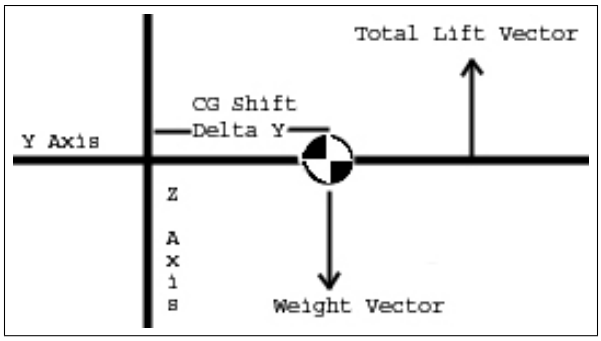


Figure 3.6: CG Rolling Moment from Wing Damage

at a level bank angle. Using Figure 3.3, a simple one degree of freedom analysis of the roll equation can provide an estimate of the necessary aileron deflection. Since the final steady state aileron deflection is of concern and not the transient recovery capabilities, $\vec{\omega}$, $\dot{\vec{\omega}}$, $\dot{V}_b$, and V_b (except for U_1) will be set to zero, along with the Φ Euler Angle.

$$\frac{-\Delta y \; m \; g}{q_{bar} \; S_{ref} \; b} = C_{l_0} + C_{l_{\delta a}} \delta a \tag{3.63}$$

For damage to the left wing, C_{l_0} will be negative and Δy will be positive. The left hand side of the equation will then be smaller in magnitude than C_{l_0}, and both will have the same sign. Only positive aileron deflection, to bring the right wing down, will reduce the influence of C_{l_0}, until it equals the moment due to weight.

In a similar manner, the elevator will have some physical limits on controlling the aircraft. If wing damage is present, the tail still remains very effective and can continue to trim the aircraft at higher angles of attack to achieve the necessary lift for level flight. As was discussed in Section 2, aircraft that experience a loss in wing area must increase their velocity to keep angles of attack within stall limits. Simulator results will later show that for wing damage, the aileron control power becomes the limiting factor. Standard aircraft will lack the benefit of roll control with full moving elevons that allows velocity to be the limiting factor in asymmetric flight.

Tail damage does present boundaries for control. Losing a whole side of the horizontal tail may not decrease the capability of the aircraft to develop sufficient lift. This is especially the case if the tail must provide downforce to balance the aircraft moments. If the full ailerons are available, then $L_{\delta e}$ can be controlled and loss of pitch control (the requirement of more than full elevator deflection) is the only failure criteria. Such an elevator deflection can be assesed by assembling a two degree of freedom pitch assesment with the same rotation and velocity criteria as the aileron analysis. Euler Angle Θ can be nonzero and a flight path angle γ will be set to zero, so that Θ equals α. Bank angle Φ will also be set to zero. Working

from equations (3.15) and (3.38) the following system of equations are developed to find α and δe to control some CG shift Δx at some thrust determined flight dynamic pressure q_{bar}.

$$\begin{bmatrix} C_{L_\alpha} & C_{L_{\delta e}} \\ C_{M_\alpha} & C_{M_{\delta e}} \end{bmatrix} \begin{bmatrix} \alpha \\ \delta e \end{bmatrix} = \begin{bmatrix} \frac{m\ g\ \sin\alpha}{q_{bar}\ S_{ref}} - C_{L_0} \\ \frac{m\ g\ \Delta x \cos\alpha}{q_{bar}\ S_{ref}\ c_{bar}} - C_{M_0} \end{bmatrix} \tag{3.64}$$

The solution to equation (3.64) will be iterative to converge to an angle of attack that is the same in both the trigonometric functions and in the solution to the system of equations. A similar analysis can be performed for the rudder and its ability to hold headings or even reject an OEI. Equation (3.63) and (3.64) and the concepts behind them will become useful in not only analyzing the success of the controller design, but on how to determine what parameters and feedbacks deserve more attention than others.

Chapter 4

4 MODELING AND SIMULATION

This chapter will describe the steps taken to implement the asymmetric aircraft equations of motion in a simulation environment and to determine the values of the stability derivatives for the Beech Bonanaza F33C. The current simulator is a Six Degree of Freedom model built in MATLAB/Simulink, and the changes made to it are described in Section 4.1. Methods discussed previously in Section 3 are used to determine the stability derivatives of damaged aircraft. Details are presented in Section 4.3.

4.1 Simulation Structure

Simulation of the damaged asymmetric aircraft is performed in the MATLAB/Simulink environment. In this environment, calculations are run by combining "blocks" that perform various mathematical operations or output signals. There is no capability to "fly" the aircraft in the current setup as the simulation runs for a given time based on a set of timed command inputs such as bank angle, sideslip angle, flight path angle, and velocity. Once the simulation runs, the results are essentially a pilot not in the loop, or "hands off", response of the aircraft to the inputs and the controller logic. The block diagram of the highest level of the simulation structure is given in Figure 4.1. Table 4.1 lists these top level subsystems and their function in the simulator. Table 4.2 describes the added "s-functions", which will be referred to in Section 4.1.3.

Table 4.1

Top Level Simulation Blocks

Block	Subsystems	Function
Beta Command Bank Command Velocity Command Gamma Command	none	Outputs ramp commands for trimmed values that vary from the initial conditions.
Lateral Outputs Gamma Outputs Velocity Outputs	none	Takes aerodynamic property signals (bank angle, angular rates)and plots them or stores the data.
Control Subsystem	Bank Control Beta Control Gamma Control	Control input signals from the command blocks are fed through various compensators, producing commanded angular rates.
Inverse Controller	none	Estimates control surface values from commanded rates.
Adapative Bias Correctors	none	Observes error b/w commands and aircraft behavior and adjusts Control Subsystem outputs to eliminate the error.
Airplane	Forces Moments 6DOF	Control surface deflections and current motion data are input to solve for the new velocities and accelerations.

Table 4.2

Aircraft Coefficient Switching S-Functions

S-Function	Location	Function
Var_Change	Top Level	Changes undamaged aircraft coefficients to damaged values when the simulation time reaches a predetermined failure time.
Lift_Var	Airplane/Forces/Lift	Outputs Lift Coefficient buildup and wing area.
Drag_Var	Airplane/Forces/Drag	Outputs Drag Coefficient buildup and wing area.
Side_Var	Airplane/Forces/Side Force	Outputs Side Force Coefficient buildup and wing area.
Roll_Var	Airplane/Moments/Roll	Outputs Roll Coefficient buildup and wing area.
Yaw_Var	Airplane/Moments/Yaw	Outputs Yaw Coefficient buildup and wing area.
Pitch_Var	Airplane/Moments/Pitch	Outputs Pitch Coefficient buildup and wing area.
MassChange	Airplane/Moments (all) Airplane/Euler Angles (6DOF) Airplane/Non-Dim Rates Airplane/Aero Angles	Outputs the current mass for the gravity vector. Chord and span values are output to find moments and to non-dimensionalize the aerodynamic angles and rates

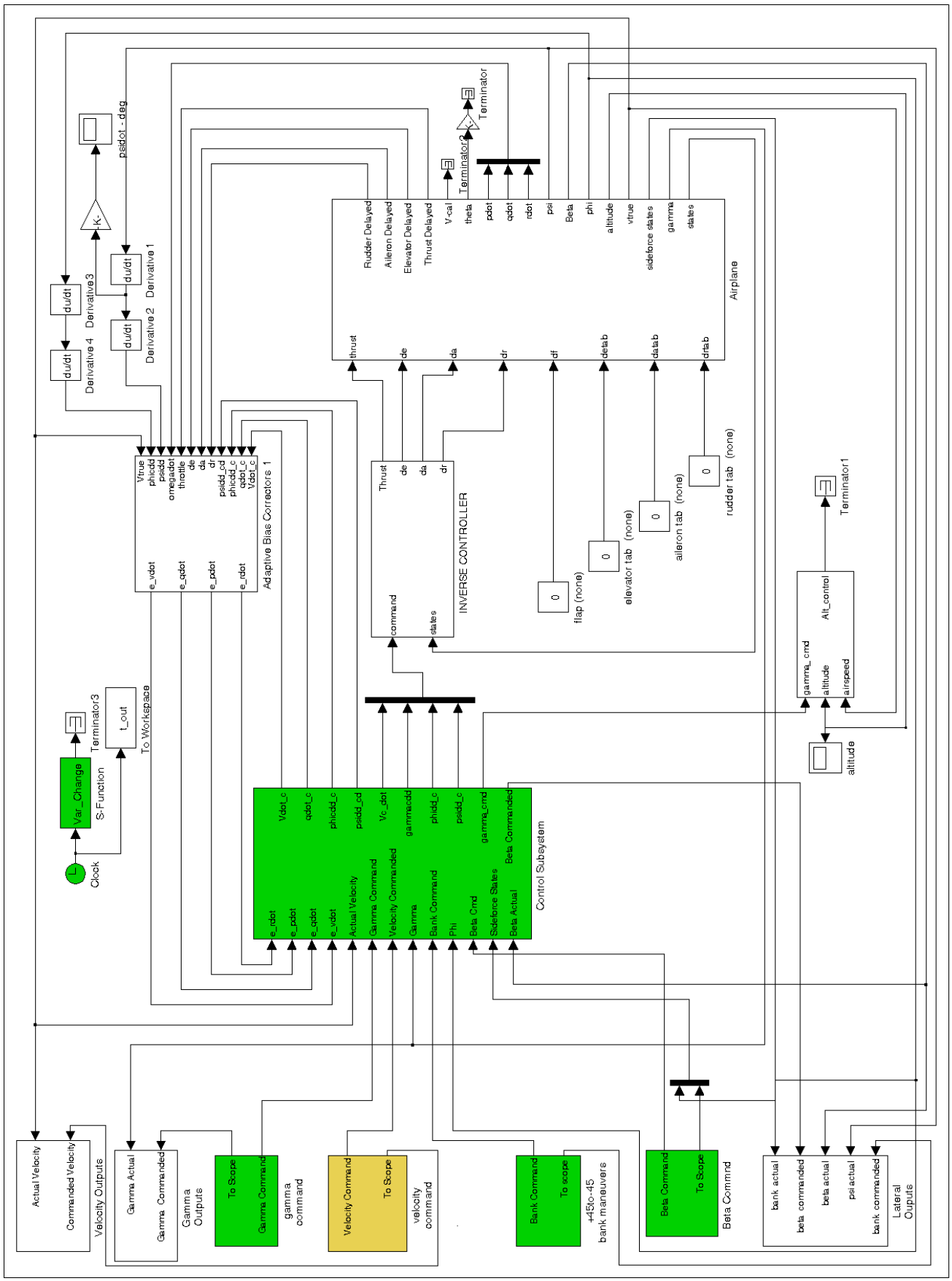


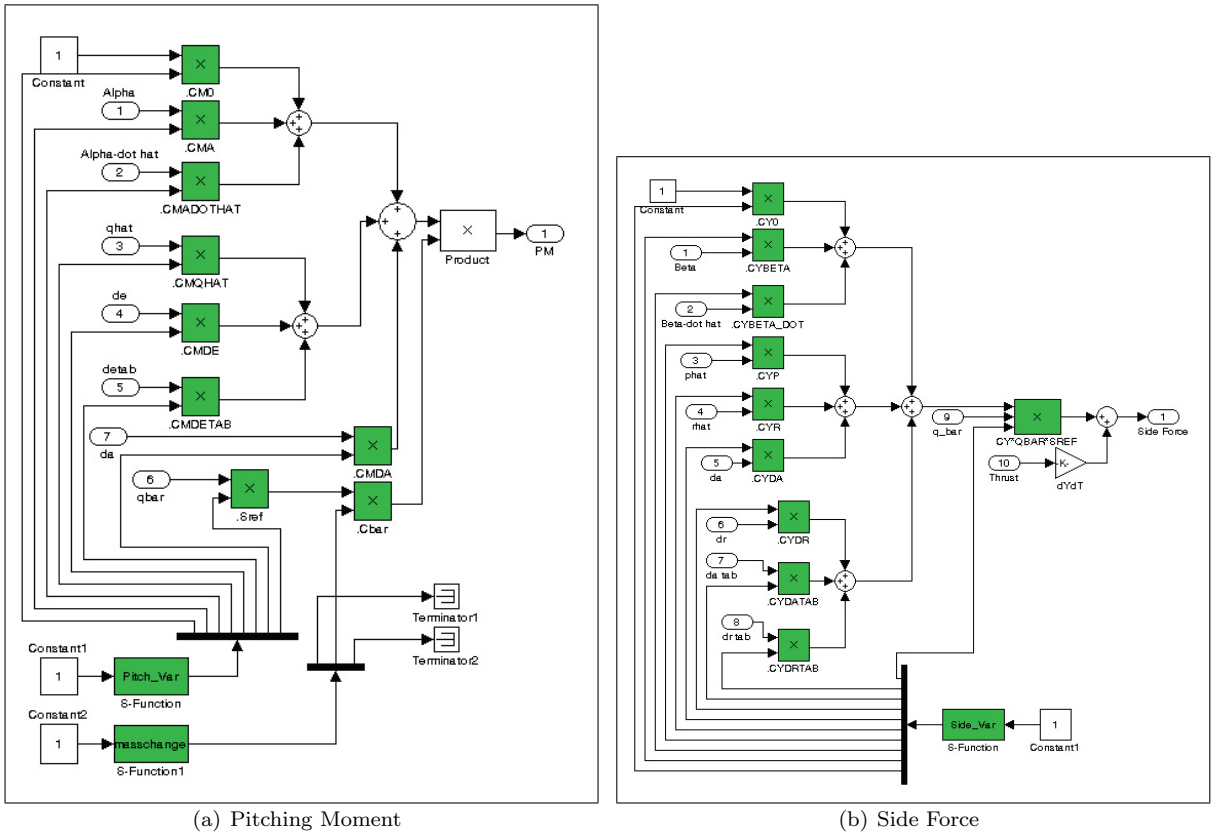
Figure 4.1: Simulation Top Level Diagram

The Adaptive Bias Correctors and the Control Subsystem will be discussed in Section 4.2, and elements of the Airplane block will be covered in the proceeding sections.

4.1.1 Aerodynamic Model

The aerodynamics of the aircraft are built into the simulator in the form of a coefficient buildup. These coefficients are loaded into the MATLAB workspace by the use of m-files, and can be changed to model any aircraft for which these values are known. Since the coefficient buildup produces forces and moments in the aircraft's stability axis, these values are converted to the body centered axis system. Some examples of the coefficient buildup can be seen in Figure 4.2. For a damaged asymmetric aircraft, the simulator must be able to accept any new coupled terms, such as $C_{m_{\delta a}}$, so these blocks have been added onto the existing simulation structure. The coefficient buildup is provided for the Pitching Moment Block in Figure 4.2(a) to demonstrate how the block works.

$$M = \bar{q}S_{ref}\bar{c}\left[C_{m_0} + C_{m_\alpha}\alpha + C_{m_{\dot{\hat{\alpha}}}}\dot{\hat{\alpha}} + C_{m_{\hat{q}}}\hat{q} + C_{m_{\delta e}}\delta e + C_{m_{\delta etab}}\delta_{etab} + C_{m_{\delta a}}\delta a\right] \tag{4.65}$$



(a) Pitching Moment (b) Side Force

Figure 4.2: Example Simulink Coefficient Buildup

4.1.2 Six Degrees of Freedom Model

MALTAB/Simulink, as part of the Aerospace toolbox, comes with a subsystem to compute the 6DOF Euler Angles, accelerations, and velocities (both linear and angular) of an aircraft. The inputs to the system are the body axis forces and moments. For an undamaged aircraft, the weight vector's influence on force, as seen in equation (3.15), must be computed before entering the Euler Angle block. Figure 4.3(a) shows the original Euler Angle system, and Figure 4.3(b) shows the added terms from equations (3.15) and (3.37) through (3.39). Modifications to the angular velocity determination require the addition of the moment caused by the weight vector, the acceleration of the mass over $\mathbf{\Delta r}$, and addition of the normally excluded asymmetric Products of Inertia. Modifications to the linear velocity determination require the addition of the rotational effects of the CG shift. Additions to the model are shaded for clarity in the figures.

The Simulink structure pictured in Figure 4.3 solves the vector form of the force and moment equations. For comparison purposes, the symmetric aircraft equations of motion are presented alongside the damaged asymmetric equations. The inertia matrix $\mathbf{I}$ is constant for the undamaged aircraft, but in the updated Euler Angle Block, a switching function changes the inertia matrix from symmetric to asymmetric. A similar switching function changes $\mathbf{\Delta r}$ from zero to damaged aircraft values. A few aircraft acceleration terms with cross products are computed using the anti-commutative property, so the order and the sign of the cross product is switched. Besides solving for the aircraft accelerations, the Euler Angle block computes the Earth relative velocity and displacement, but no changes are necessary for this portion of the simulation.

In a vector equation form, the undamaged symmetric and damaged asymmetric equations for the Euler Angle blocks are solved for the acceleration vectors $\dot{\mathbf{V}}_\mathbf{b}$ and $\dot{\vec{\omega}}$. For the undamaged symmetric aircraft linear acceleration, this solution is as follows

$$m\left[\dot{\mathbf{V}}_\mathbf{B} + \vec{\omega} \times \mathbf{V_B}\right] = m\vec{g} + \vec{F_A} + \vec{F_T} \tag{4.66}$$

$$\dot{\mathbf{V}}_\mathbf{B} = -\vec{\omega} \times \mathbf{V_B} + \left[m\vec{g} + \vec{F_A} + \vec{F_T}\right]/m \tag{4.67}$$

$$\dot{\mathbf{V}}_\mathbf{B} = \mathbf{V_B} \times \vec{\omega} + \left[m\vec{g} + \vec{F_A} + \vec{F_T}\right]/m \tag{4.68}$$

The bracketed term is the Force vector that is input into the Euler Angle Block. A similar solution occurs for the damaged asymmetric aircraft linear acceleration.

$$m\left[\dot{\mathbf{V}}_\mathbf{B} + \vec{\omega} \times \mathbf{V_B} + \dot{\vec{\omega}} \times \mathbf{\Delta r} + \vec{\omega} \times (\vec{\omega} \times \mathbf{\Delta r})\right] = m\vec{g} + \vec{F_A} + \vec{F_T} \tag{4.69}$$

$$\dot{\mathbf{V}}_\mathbf{B} = \mathbf{V_B} \times \vec{\omega} - \dot{\vec{\omega}} \times \mathbf{\Delta r} - \vec{\omega} \times (\vec{\omega} \times \mathbf{\Delta r}) + \left[m\vec{g} + \vec{F_A} + \vec{F_T}\right]/m \tag{4.70}$$

Again, the final bracketed term is the input to the Euler Angle Block. Angular acceleration for the undamaged symmetric aircraft can then be found in vector form.

$$I\dot{\vec{\omega}} + \vec{\omega} \times I\vec{\omega} = M_a + M_t \tag{4.71}$$

$$I\dot{\vec{\omega}} = -\vec{\omega} \times I\vec{\omega} + [M_a + M_t] \tag{4.72}$$

$$\dot{\vec{\omega}} = I^{(-1)} \left[-\vec{\omega} \times I\vec{\omega} + [M_a + M_t]\right] \tag{4.73}$$

The larger bracketed term here is the output from the summing block (the circle) in Figure 4.3(a), and is multiplied by the inverse of the Inertia matrix to obtain the angular accelerations. For the damaged aircraft, the angular acceleration of the CG is added and results in the following solution for angular acceleration.

$$\left[(m\mathbf{\Delta r}) \times \frac{D^2\mathbf{R_b}}{Dt^2}\right] + I\dot{\vec{\omega}} + \vec{\omega} \times I\vec{\omega} = [\mathbf{\Delta r} \times m\vec{g}] + M_a + M_t \tag{4.74}$$

$$I\dot{\vec{\omega}} = -\left[(m\mathbf{\Delta r}) \times \frac{D^2\mathbf{R_b}}{Dt^2}\right] - \vec{\omega} \times I\vec{\omega} + [\mathbf{\Delta r} \times m\vec{g}] + M_a + M_t \tag{4.75}$$

$$\dot{\vec{\omega}} = I^{(-1)} \left[-\left[(m\mathbf{\Delta r}) \times \frac{D^2\mathbf{R_b}}{Dt^2}\right] - \vec{\omega} \times I\vec{\omega} + [\mathbf{\Delta r} \times m\vec{g}] + M_a + M_t\right] \tag{4.76}$$

Linear acceleration of the aircraft appears in the term $\left[(m\mathbf{\Delta r}) \times \frac{D^2\mathbf{R_b}}{Dt^2}\right]$. Solving for $\frac{D^2\mathbf{R_b}}{Dt^2}$ provides the terms necessary to pull from the Forces segment of the Euler Angle Block.

$$\frac{D^2\mathbf{R_b}}{Dt^2} = \dot{\mathbf{V}}_\mathbf{B} + \vec{\omega} \times \mathbf{V_B} \tag{4.77}$$

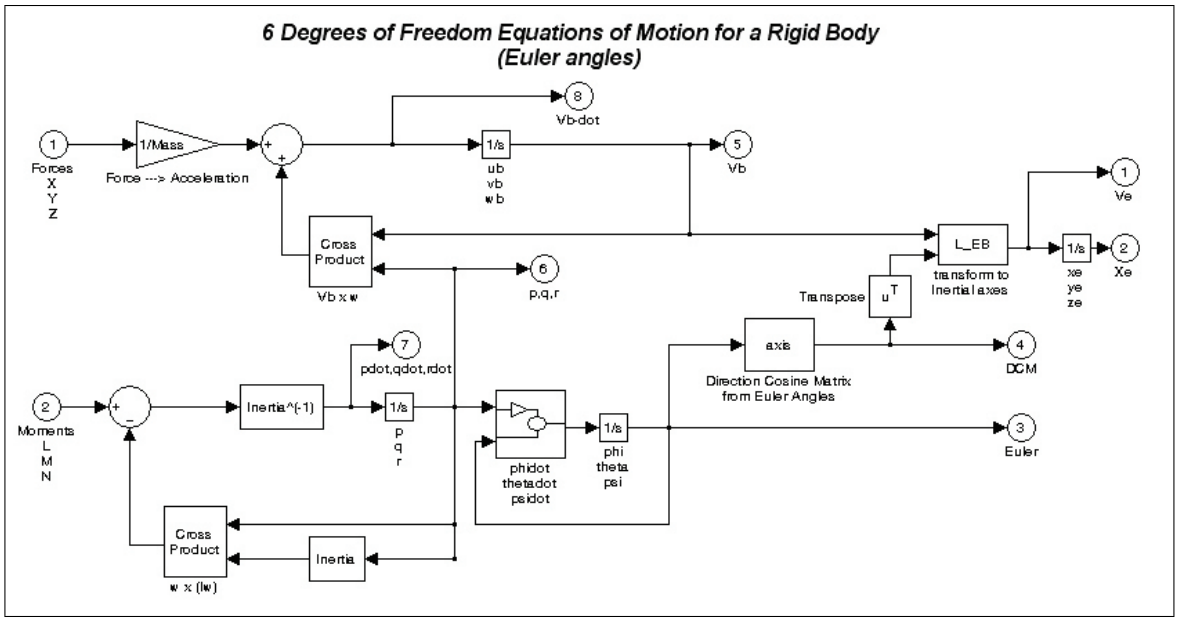
Rearranging equation (4.70) and substituting into equation (4.77) provides the new terms to include the angular acceleration determination.

$$\dot{\mathbf{V}}_\mathbf{B} + \vec{\omega} \times \mathbf{V_B} = \left[m\vec{g} + \vec{F_A} + \vec{F_T}\right]/m - \left[\dot{\vec{\omega}} \times \mathbf{\Delta r} + \vec{\omega} \times (\vec{\omega} \times \mathbf{\Delta r})\right] \tag{4.78}$$

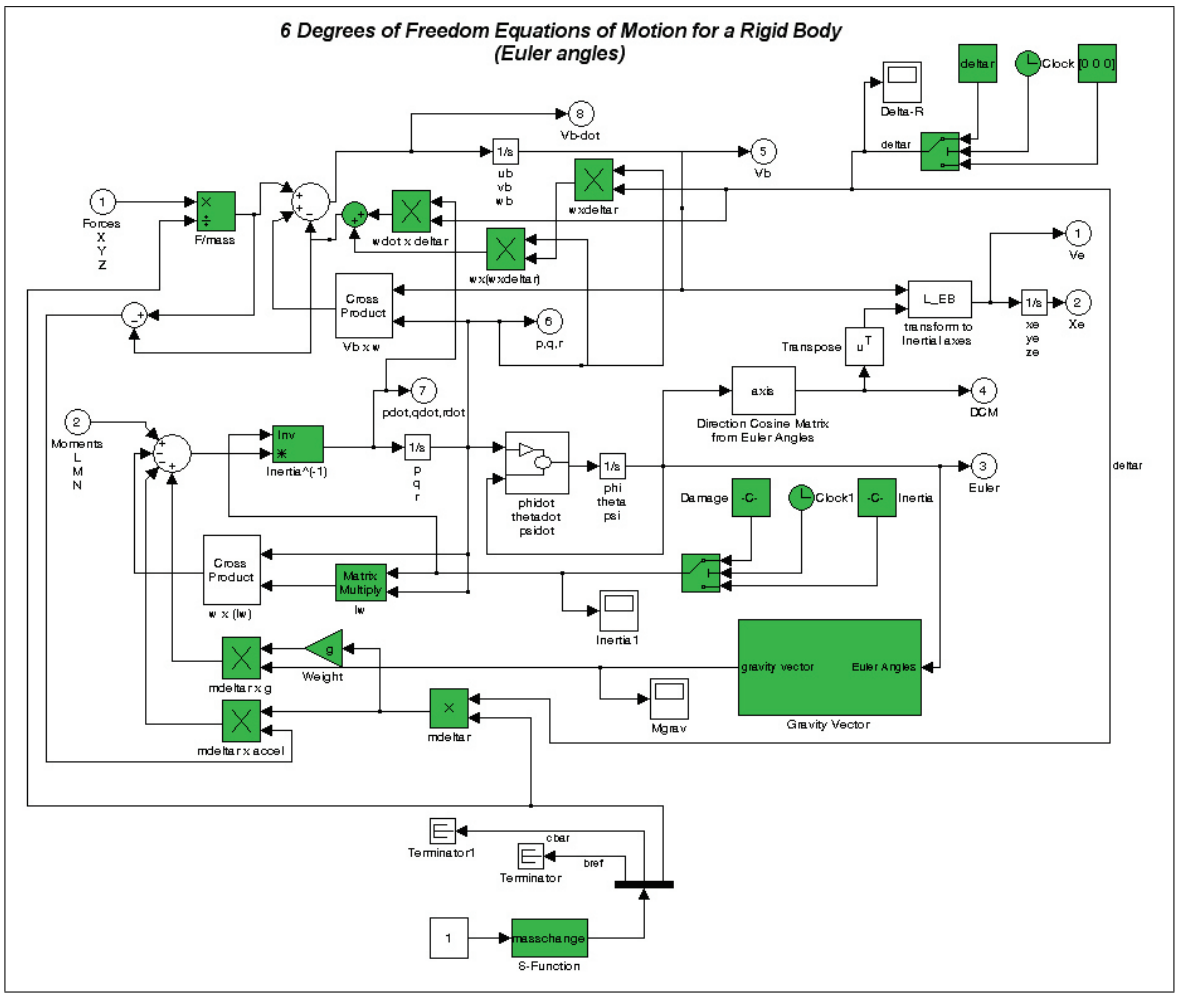
$$\frac{D^2\mathbf{R_b}}{Dt^2} = \left[m\vec{g} + \vec{F_A} + \vec{F_T}\right]/m - \left[\dot{\vec{\omega}} \times \mathbf{\Delta r} + \vec{\omega} \times (\vec{\omega} \times \mathbf{\Delta r})\right] \tag{4.79}$$

The final equation to solve for $\dot{\vec{\omega}}$ is now

$$\dot{\vec{\omega}} = I^{(-1)} \left[-\left[\frac{m\vec{g} + \vec{F_A} + \vec{F_T}}{m} - \left[\dot{\vec{\omega}} \times \mathbf{\Delta r} + \vec{\omega} \times (\vec{\omega} \times \mathbf{\Delta r})\right]\right] - \vec{\omega} \times I\vec{\omega} + [\mathbf{\Delta r} \times m\vec{g}] + M_a + M_t\right] \tag{4.80}$$



(a) Original Euler Angle Block



(b) Updated Euler Angle Block

Figure 4.3: Euler Angle Simulation Structures

4.1.3 Damage Initiation

Simulations are run for an undamaged symmetric aircraft by loading, from a MATLAB m-file, the force and moment coefficients, geometric and mass properties, and trim values for a given altitude and flight speed. Parameters for the compensators in the Control Subsystem are initialized via m-files as well. The calculations for the simulation for any amount of time do not require any update of the coefficients, geometry, or mass parameters. To simulate an aircraft that sustains damage in the middle of cruise, the simulator has been adjusted from using the coefficients as simple gains. Since static global variable values do not update in Simulink Gain blocks, another method had to be used to multiply the aerodynamic angles or control deflection by their corresponding coefficients. Since user defined "s-functions" will update global variable values, these functions were written to continually output the coefficients, wing span, wing area, chord, mass, and inertia as detailed in Table 4.2. A single function "Var_Change" switches the aircraft coefficients and geometric properties to the damaged asymmetric values at the predetermined moment of aircraft damage. The remaining functions in Table 4.2 output the current values for every simulation variable. Figure 4.4 provides the code structure for this process. Source code for these functions are given in Appendix B.

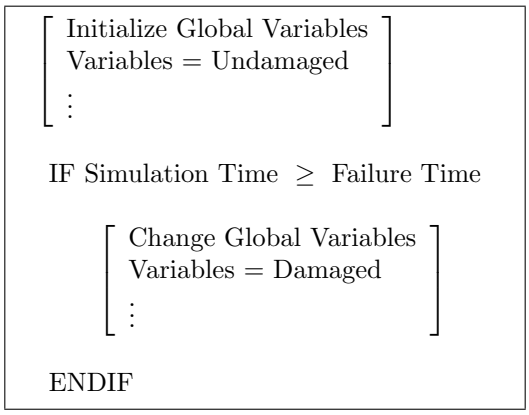


Figure 4.4: Variable Switching Process

4.2 Control Architecture

Figure 4.1 shows the specific setup for the control of a damaged asymmetric aircraft. In order to develop each subsystem properly, a simplified look at the control architecture gives a clearer depiction of the control problem. A block diagram of the general control architecture is given in Figure 4.5. This control architecture is general for each command input (velocity, flight path angle, bank angle, sideslip angle) into the simulator. Every reference input runs through a Model Follower and then the tracking error is calculated and goes through a Proportional-Integral-Derivative filter (PID). The resultant signal is the commanded parameter's

second derivative. The Inverse Controller (IC) then receives the second derivative signal and outputs control surface deflections or throttle settings. Finally, the Aircraft block takes these inputs and the state values for the aircraft are output and then become part of a unity feedback.

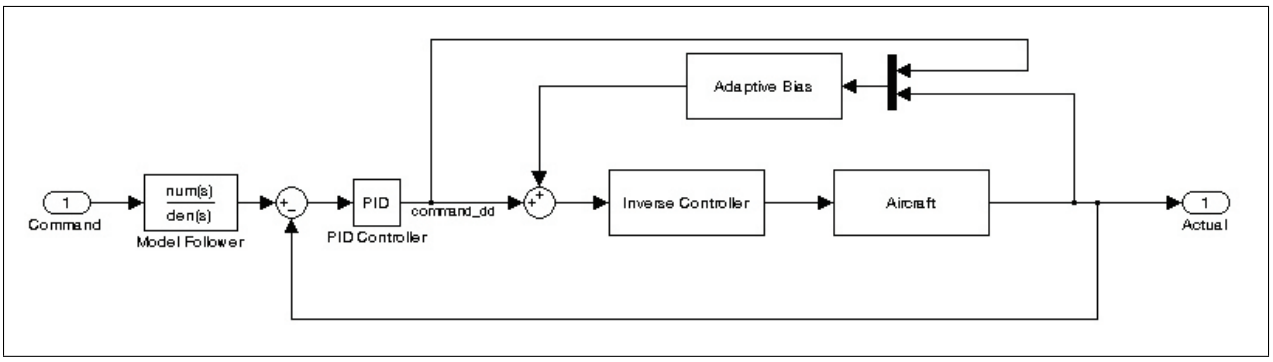


Figure 4.5: Generalized Aircraft Control Diagram

The upper loop in Figure 4.5 is the Adapative Bias Correctors. This system provides a correction to the PID output based on the pre-correction commands and the actual aircraft outputs. Each of these control structures will be discussed separately in the following sections. The Model Follower and PID are designed to have the entire closed loop system behave as a model second order system, represented by equation (4.81).

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{4.81}$$

4.2.1 Aircraft Response and Model Follower

This section will discuss the determination of the Model Follower and PID parameters. These parameters will directly influence the aircraft response to a command, or response to a disturbance or damage. The latter is of more importance in this research, as the commands to the aircraft will normally be to hold some flight path angle and velocity at a given altitude, with zero sideslip and zero bank angle. Once damage is initiated on the aircraft, the aircraft's outputs will no longer be equal to the commands. It will be the Model Follower's and the PID's job to specify how the aircraft returns to the commanded zero sideslip and zero bank angle state.

The selection of the Model Follower and PID come from the following theory, illustrated in Figure 4.6. If the IC is a perfect inverse to the symmetrical aircraft, or if the adaptive bias correctors can immediately adapt for any error, then that loop reduces to unity. Since a commanded state's second derivative is fed into the IC, and the actual state is output from the Aircraft block, this loop will reduce to a double integrator.

Achieving a closed loop transfer function equal to that of equation (4.81) is not a difficult task, but neither is it a unique solution. The selection of a PID eliminates enough variables in the closed loop system

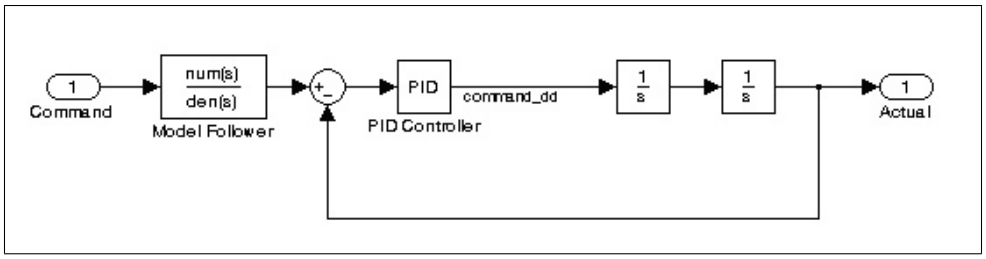


Figure 4.6: Simplified Aircraft Control Diagram

to select a Model Follower that gives the desired second order closed loop system in equation (4.81). For the derivation of the Model Follower, the Laplace domain transfer functions of the PID and Model Follower will be referred to by $PID(s)$ and $MF(s)$, respectively.

For the inner loop in Figure 4.6, the open loop transfer function is $PID(s) * \frac{1}{s^2}$. $PID(s)$ can be rewritten as $K_P + \frac{K_I}{s} + K_D s$, where the K's are the gains on the proportional, integral, and derivative terms. The closed loop transfer function for the entire ideal system from Figure 4.6 is now given by equation (4.82). $Com(s)$ is the command input, and $Act(s)$ is the actual aircraft output.

$$Act(s) = \frac{K_P + \frac{K_I}{s} + K_D s}{s^2}\big[MF(s)Comm(s) - Act(s)\big] \tag{4.82}$$

Rearranging to find $Act(s)/Com(s)$ yields

$$\frac{Act(s)}{Com(s)} = \frac{\big[K_D s^2 + K_P s + K_I\big]}{\big[s^3 + K_D s^2 + K_P s + K_I\big]} MF(s) \tag{4.83}$$

The necessary Model Follower to give a closed loop transfer function like that of equation (4.81) is a 4^{th} order denominator with a 3^{rd} order numerator, and is given in equation (4.84).

$$MF(s) = \frac{\big[s^3 + K_D s^2 + K_P s + K_I\big]\frac{K_I}{K_D}}{\big[K_D s^2 + K_P s + K_I\big]^2 \frac{1}{K_D}} \tag{4.84}$$

While the system is proper and therefore physically realizable, the possibility of pole-zero cancellations do exist. Obtaining the final closed loop transfer function provides the relationship between ζ and ω_n and the PID gains.

$$\frac{Act(s)}{Com(s)} = \frac{\big[K_D s^2 + K_P s + K_I\big]}{\big[s^3 + K_D s^2 + K_P s + K_I\big]} \frac{\big[s^3 + K_D s^2 + K_P s + K_I\big]\frac{K_I}{K_D}}{\big[K_D s^2 + K_P s + K_I\big]^2 \frac{1}{K_D}} \tag{4.85}$$

$$\frac{Act(s)}{Com(s)} = \frac{\frac{K_I}{K_D}}{\big[K_D s^2 + K_P s + K_I\big]\frac{1}{K_D}} \tag{4.86}$$

$$\frac{Act(s)}{Com(s)} = \frac{\frac{K_I}{K_D}}{\left[s^2 + \frac{K_P}{K_D}s + \frac{K_I}{K_D}\right]} \tag{4.87}$$

Comparing equation (4.87) to equation (4.81), it is seen that $\omega_n^2 = \frac{K_I}{K_D}$ and $2\zeta\omega_n = \frac{K_P}{K_D}$. Since there are two equations and three unknowns, K_D is set to one, and the PID gains can be found based on the 2nd Order characteristics. In a similar manner, a much simpler Model Follower can be derived for the use of proportional-derivative compensator (PD). Equation (4.88) gives the Model Follower transfer function to achieve a 2nd order closed loop system with a PD in the inner loop.

$$MF(s) = \frac{K_P}{K_D s + K_P} \tag{4.88}$$

Solving the entire closed loop to achieve the PD gains in terms of ζ and ω_n

$$\frac{Act(s)}{Com(s)} = \frac{K_P + K_D s}{s^2 + K_D s + K_P} MF(s) \tag{4.89}$$

$$\frac{Act(s)}{Com(s)} = \frac{K_P + K_D s}{s^2 + K_D s + K_P} \frac{K_P}{K_D s + K_P} \tag{4.90}$$

$$\frac{Act(s)}{Com(s)} = \frac{K_P}{s^2 + K_D s + K_P} \tag{4.91}$$

From equation (4.91), it is seen that $\omega_n^2 = K_P$ and $2\zeta\omega_n = K_D$. Another possible replacement for the model closed loop transfer function G(s) from equation (4.81) is a third order system. Any third or higher order system can be expressed as a partial fraction expansion of multiple smaller order systems [36]. This model system will be defined to have a partial fraction expansion, that in the time domain, gives a second order response plus a stable exponential decay. The Laplace Domain of this is

$$G(s) = \frac{1}{s + a} + \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{4.92}$$

Resulting from a third order closed loop model transfer function

$$G(s) = \frac{s^2 + (\omega_n^2 + 2\zeta\omega_n)s + \omega_n^2(1 + a)}{s^3 + (a + 2\zeta\omega_n)s^2 + (\omega_n^2 + a2\zeta\omega_n)s + a\omega_n^2} \tag{4.93}$$

Equation (4.94) gives the necessary Model Follower to put equation (4.83) in the form of equation (4.93).

$$MF(s) = \frac{s^2 + (\omega_n^2 + 2\zeta\omega_n)s + \omega_n^2(1 + a)}{K_D s^2 + K_P s + K_I} \tag{4.94}$$

Obtaining the final closed loop transfer function provides the relationship between ζ and ω_n and the PID

gains.

$$\frac{Act(s)}{Com(s)} = \frac{\left[K_D s^2 + K_P s + K_I\right]}{\left[s^3 + K_D s^2 + K_P s + K_I\right]} \frac{s^2 + (\omega_n^2 + 2\zeta\omega_n)s + \omega_n^2(1+a)}{K_D s^2 + K_P s + K_I} \tag{4.95}$$

$$\frac{Act(s)}{Com(s)} = \frac{s^2 + (\omega_n^2 + 2\zeta\omega_n)s + \omega_n^2(1+a)}{\left[s^3 + K_D s^2 + K_P s + K_I\right]} \tag{4.96}$$

Comparing the denominator of equation (4.96) to that of equation (4.93) gives the relationship between the PID gains and ζ, ω_n, and a: $K_D = a + 2\zeta\omega_n$, $K_P = \omega_n^2 + a2\zeta\omega_n$, and $K_I = a\omega_n^2$. For the inclusion of a numerator value other than 1 for the exponential time response in equation (4.92) (e.g. $d/(s+a)$), the Model Follower necessary is given by equation (4.97), and the final closed loop transfer function is given by equation (4.98). Notice there is no change to relationship between the PID gains and the system response characteristics.

$$MF(s) = \frac{ds^2 + (\omega_n^2 + 2d\zeta\omega_n)s + \omega_n^2(d+a)}{K_D s^2 + K_P s + K_I} \tag{4.97}$$

$$\frac{Act(s)}{Com(s)} = \frac{ds^2 + (\omega_n^2 + 2d\zeta\omega_n)s + \omega_n^2(d+a)}{\left[s^3 + K_D s^2 + K_P s + K_I\right]} \tag{4.98}$$

For each Model Follower, the gains K_P, K_D, and K_I are selected by choosing the damping ratio (ζ) and rise time (t_r), and obtaining the undamped natural frequency (ω_n) from ζ and t_r. Table 4.3 shows the algebraic relationship between the compensator gains and the 2nd order system characteristics for both the PD, PID 2nd order, and PID 3rd order based Model Follower. As detailed earlier, the solution for the PID based 2nd Order Model Follower requires setting the gain K_D equal to 1 so there are only two equations and two unknowns.

Table 4.3

Relationship Between Compensator Gains and 2nd Order Characteristics

Compensator	**PID Gains**		
	K_P	K_I	K_D
PD 2O	ω_n^2	-	$2\zeta\omega_n$
PID 2O	$2\zeta\omega_n$	ω_n^2	1
PID 3O	$\omega_n^2 + a2\zeta\omega_n$	$a\omega_n^2$	$a + 2\zeta\omega_n$

Note: a is equal to the time constant of the 3rd Order exponential response.

Despite the closed loop transfer functions and their time domain response being similar for each compensator and their Model Follower, the response of the aircraft will differ. This is caused by the IC-Airplane-Bias loop not being unity for the initial phase of damage, since the adaptive bias correctors will take some time

to completely adjust the commanded signal. It is difficult to determine what the non-unity version of the upper loop system in Figure 4.5 will be. Choosing compensator gains then becomes a trial and error process for each combination of compensator and Model Follower. From gain experimentation, a few notable trends were observed. First, increasing t_r allows control of higher percent damage, but reduces the performance of smaller percent damages. Second, decreasing ζ allows control of higher percent damage, but increases the overshoot of the response and can even lead to controller induced oscillations.

Testing on the three compensator designs shows that the third order closed loop transfer function in the bank angle loop allowed for a better recovery from damage. Section 5 demonstrates these results with the bank angle time history for a range of wing damages and flight speeds. Time response characteristics for each compensator in the Bank Angle Control subsystem are given in Table 4.4. While the characteristics of the second order response are not the same in simulation, they each provide a "best" response for the individual compensator over a range of flight speeds and amounts of damage. The Beta Control compensator design is a second order closed loop PID and Model Follower, with the same rise time, damping ratio, and natural frequency as the 2nd Order PID in Table 4.4.

Table 4.4

Bank Angle Control - 2nd Order System Characteristics

	PD 2O	**PID 2O**	**PID 3O**
t_r (sec)	6	8.0	9.0
ζ	0.9	0.85	0.925
ω_n	0.49	0.34	0.346

Note: a is set to 0.5 and d is set to 0.05 for the 3rd Order System.

Table 4.5

Sideslip Angle Control - 2nd Order System Characteristics

	PID 2O
t_r (sec)	8.0
ζ	0.85
ω_n	0.34

For all the bank angle controllers in Table 4.4, a long rise time with some barely underdamped or critically damped damping ratios are selected. Faster values for the rise time, or more damping, will quicken the response but will try and force the aircraft to achieve roll rates faster than the roll mode of the aircraft will allow. This leads to commanded aileron deflection saturation which induces overcorrection when the bank angle goes from negative to positive, or vice versa.

Despite the 2nd Order PID compensator behaving close to the 3rd Order compensator, the 3rd Order compensator design provides the best transient response to the loss of a percentage of a wing. While flight conditions with low percentages of damage may be controlled better by the 2nd Order PID or PD, the largest controllable percentage of damage is compensated better by the 3rd Order PID.

Results for the Compensator Design along with specific test conditions and notable results are given in Table 4.6.

Table 4.6

Compensator Design Bank Angle Results

Flight Speed	**% Damage**	**2nd Order PD**	**2nd Order PID**	**3rd Order PID**
80 knots	20	(-14.0 : 0.0)	(-11.0 : 2.5)	(-10.5 : 2.0)
80 knots	25	FAIL	(-21.0 : 15.0)	(-21.0 : 11.5)
100 knots	25	(-17.5 : 0.0)	(-13.0 : 3.5)	(-12.0 : 3.8)
100 knots	30	(-20.0 : 0.5)	(-20.0 : 6.5)	(-20.0 : 4.5)
120 knots	30	(-20.0 : 1.5)	(-16.0 : 4.0)	(-15.0 : 5.0)
120 knots	35	(-22.5 : 2.5)	(-21.0 : 6.0)	(-21.0 : 4.0)
140 knots	35	(-24.0 : 3.0)	(-18.0 : 4.0)	(-17.5 : 6.5)
140 knots	40	(-27.5 : 5.0)	(-22.5 : 5.0)	(-22.5 : 4.0)

Angles are in degrees and are given as (Max Bank Angle After Damage : Max Bank Angle Overshoot)

All final steady state aileron deflections are the same for each compensator architecture.

4.2.2 Inverse Controller

The foundations of the Dynamic Inverse Resilient Controller are discussed in this section and Section 4.2.3. For this research, the DIRC will consist of two main components. First is the dynamic inverse of the aircraft, and second is a simplified neural network known as an Adaptive Bias Corrector.

The Inverse Controller (IC) can be see in Figure 4.1 between the Control Subsystem and the Aircraft blocks. The development of the IC is discussed in previous research at WSU in Section 2.3. Inputs to the IC are commanded accelerations, such as $\dot{\vec{\omega}}$. Instead of solving the separate linear thrust and elevator deflection equations developed by Pesonen et al. [29], separate matrix systems of equations for trim in both the longitudinal and lateral-directional modes [31] are solved. Kinematic equations solve the relationships between the Euler Angles and aerodynamic angles, and their derivatives, at nonzero bank angles.

Using this decoupled inverse for a standard symmetric aircraft, the IC determines the aileron, rudder, elevator, and thrust setting necessary to achieve the commanded accelerations. In the simulator environment, the IC is given the undamaged aircraft force and moment coefficients. When damage is initiated, only the

Aircraft block receives the updated values, and the IC continues to make control deflection decisions based on the inverse of the undamaged aircraft model.

Inverse Controller commanded control surface deflections are then going to be different than what is required to recover from the damage. For example, the loss of an entire aileron will require the remaining aileron to be deflected twice as much for a similar roll requirement. Without a system in place to adapt the commanded accelerations input into the IC, the aircraft will never be able to recover.

4.2.3 Adaptive Bias Correctors

Numerous methods exist to adapt to changes in a control system. Artificial Neural Networks (ANN) can be trained off-line to deal with mathematically nonlinear problems. However, training an ANN to cover the entire range of command inputs and error corrections is a monumental task. The training of an ANN on-line is a tested solution. Both Nguyen [22] and Bosworth [17] use a modified single-layer sigma-pi neural network in the direct adaptive feedback, which is the signal that corrects the output of the model follower or the linear prefilter before it enters the IC. Calise et al. [12, 16] have used Single Hidden Layer Neural Networks that are updated online based on nonlinear update laws. As discussed in Section 2, an issue with on-line ANN training is that for high-gain learning in flight regimes of large errors (wing damage for example) and unmodeled dynamics there can be high frequency oscillations in the adaptive signals.

Another option is to have a simplified ANN that performs a simple bias weight updating. Referring to Figure 4.5, the output of the Model Follower is a pre-corrected command which is compared to the actual aircraft output. For the Bank Angle Channel, the pre-corrected command is a roll acceleration, so the Bank Angle Channel's error is the difference between this signal and the actual aircraft roll acceleration. Commanded signal corrections are decoupled from one another to preserve the entire FCS decoupled higher level flight parameters.

An Adaptive Bias Corrector (ABC) is a simplified one-node network that performs such a task. The inputs to the node are the pre-error corrected rate commands coming from the control system, the actual aircraft rates, and the prior value of the error correction. A new error correction is output from the network. Figure 4.7 shows the Simulink structure of the ABC. Equations (4.100) and (4.99) give the Adaptive Bias Correction function.

$$E_{(t)} = Command_{(t-1)} - Actual_{(t-1)} \tag{4.99}$$

$$Correct_{(t)} = Correct_{(t-1)} + C_1 E_{(t)} + C_2 E_{(t)} |E_{(t)}| \tag{4.100}$$

The bias coefficients C_1 and C_2 are essentially the "learning rates", and help adjust the amount of error

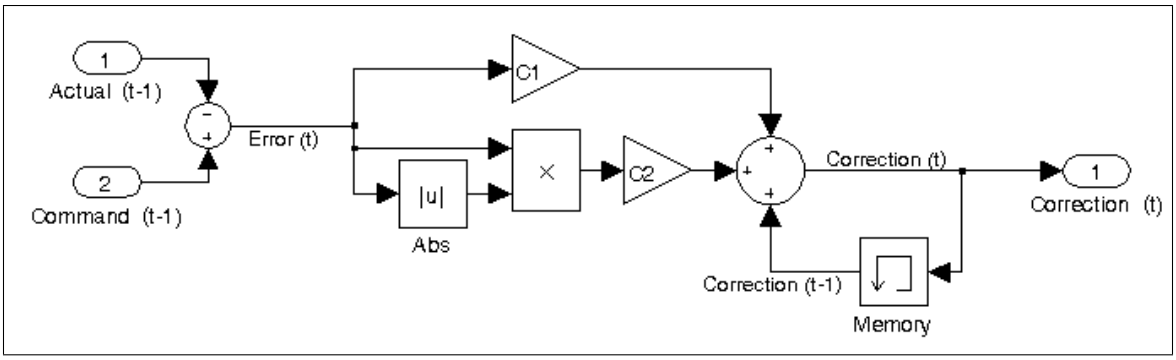


Figure 4.7: Adaptive Bias Corrector Flowchart

that shifts the prior time step's correction. If the error is zero, then no error is adjusted into the new correction value. For a nonzero error the correction is adjusted in the direction of the error, both by the magnitude and the square of the magnitude. C_1 and C_2 are normally on the order of 10^{-1} or smaller since values near 1 will drastically overshoot the proper correction and saturate the control surface deflections.

Determination of these bias coefficients is done iteratively by running the simulator for increasing damage values and adjusting the bias so that all controllable conditions can be met. Starting values came from previous research at WSU on the E-Z Fly System. There are four channels of Adaptive Bias Correctors: Roll rate, Yaw rate, Pitch rate, and forward acceleration. Table 4.7 shows the final values for the bias coefficients.

Table 4.7

Bias Coefficients

Bias	Pitch Rate	Roll Rate	Yaw Rate	$\dot{u}$
C_1	0.2	0.003	0.09	0.0
C_2	0.0	0.03	0.0	0.0

The forward acceleration has no bias correction applied to it. As discussed in Section 2, any aircraft suffering from the loss of a lifting surface must increase flight speed to achieve the same lift with less wing area. This also keeps the angle of attack below stall ranges. During the simulator tests for the bias coefficients, it was found that having an adaptation on the velocity loop would greatly increase the angle of attack. Since the flight path angle is also being commanded, the only solution to hold the pre-damage trim values was to increase the angle of attack. Removing the velocity adaptation allowed the IC to increase thrust to maintain lift and flight path angle. A large amount of pitch adaptation is added to maintain the commanded flight path angle and altitude control.

Adaptation in roll rate is a different problem than for pitch control. Any loss of lifting surface that changes the pitching moment of the aircraft can be estimated to create an aircraft response similar to a unit

step input of elevator. The longitudinal forced response to an elevator step input is a stable oscillation that settles on a new angle of attack, but a pitch rate of zero. Similarly, the response to damage that alters the longitudinal parameters will exhibit the same response. However, the pure roll response to a forced step aileron input is reaching a steady state roll rate, and a linearly increasing bank angle [31]. Asymmetric wing damage creates the same behavior as aileron deflection, that is, a constant rolling moment.

What this translates to in setting adaptive learning rates, is that for pitch rate control, the aircraft will return to zero pitch rate for damage that does not remove pitch stability. A larger learning rate can be assigned to pitch since the aircraft will provide a natural stable assistance to the controller. For roll control, any aileron deflection except for the precise value necessary to hold a steady state bank angle will cause a roll rate. Smaller learning rates for the roll channel are then necessary, since a higher inclusion of error in the correction will cause large oscillations as the ABC continually over corrects.

Yaw rate is only slightly adapted since the sideslip of the aircraft is of minimal concern when compared to a failure that induces large roll and pitch disturbances. Some sideslip is desired as any extra rudder deflection allowed can assist the ailerons in roll control. C_{N_β} and C_{Y_β} provide some assistance in a natural, stable return to a steady state sideslip and zero yaw rate. However, the differential drag from wing loss acts similarly to a forced response from a rudder input. Therefore, the learning rate for yaw is somewhat smaller than for pitch, and larger than for roll.

4.3 Asymmetric Aircraft Data

Theoretical methods for the estimation of the asymmetric stability and control derivatives have been given in Section 3.5. Specific formulas and methods, as well as the asymmetric coefficients for the percentage loss of a wing, are presented in this section. The procedure for determining the asymmetric coefficients is detailed in Figure 4.8. The goal in this section will be to obtain approximate values that are characteristic of the type of damage. Precise results are not necessary to test the effectiveness of the DIRC, as the physical limits of control exist regardless of the accuracy of the aircraft model.

First, the actual production aircraft derivatives must be known. Knowledge of these values comes from years of engineering development, wind tunnel tests, and flight testing. Second, a sufficient baseline estimation of the undamaged aircraft derivatives are found with methods from the USAF DATCOM [35] and other design texts. The aircraft planform, geometry, airfoil, and control surface specification of the test aircraft are used to create the estimations. These derivative estimations will vary from the actual aircraft values, but the methods are less complex and quicker at obtaining results. Third, the damaged aircraft planform data, combined with methods outlined in Section 3.5, provide the asymmetric aircraft derivatives.

Since the baseline estimation will not match the production aircraft derivatives, the damaged derivative

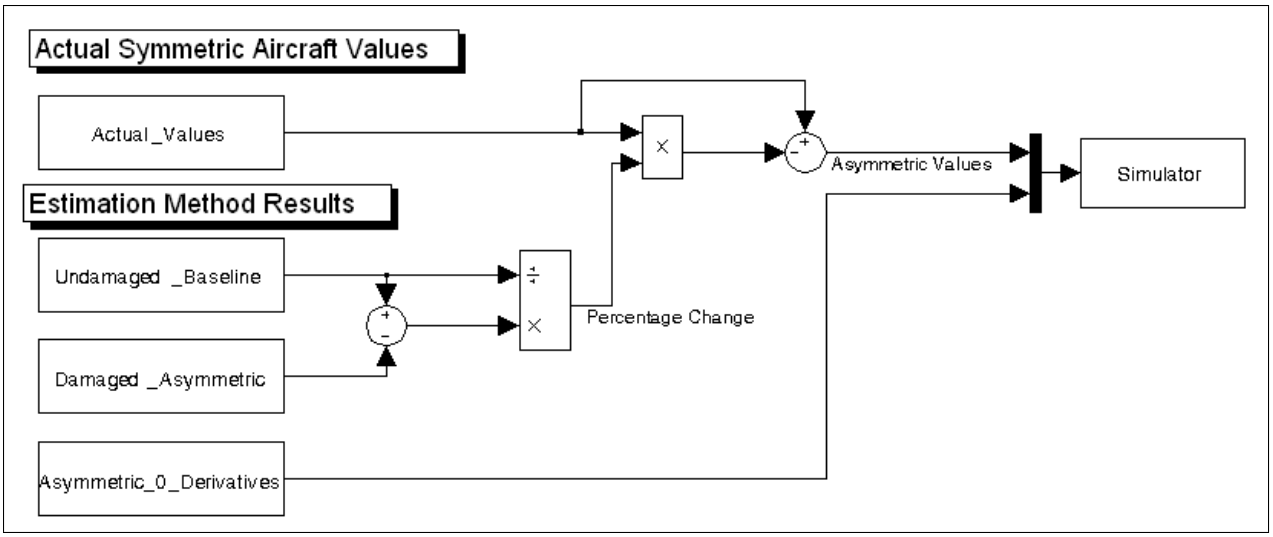


Figure 4.8: Asymmetric Coefficient Determination and Selection

estimates cannot be used accurately. Instead, the percentage error from the baseline estimation to the damaged estimation is applied to the production aircraft derivatives, and is demonstrated by equation (4.101). For the asymmetric derivatives, the estimated values will be used directly since there can be no percentage difference found from the undamaged aircraft baseline values of zero.

$$C_{a_{damaged}} = C_{a_{symmetric}} * \left(1 - \frac{C_{b_{symmetric}} - C_{b_{damaged}}}{C_{b_{symmetric}}}\right) \tag{4.101}$$

Subscript "a" refers to the actual aircraft while "b" refers to the baseline estimations

4.3.1 Geometric and Mass Properties

Given some percentage of loss of the main wing, the span will decrease and the mean aerodynamic chord will change. For no taper, the chord will remain the same, but for some amount of taper the chord will increase. Since the mean aerodynamic chord will be different on each half of wing, the total mean aerodynamic chord is calculated as the average of each wing. The wing aerodynamic center y_{bar} is also calculated for use in the estimation of the "0" derivatives. A visual method [34] is used to determine the new location of the wing aerodynamic center and mean aerodynamic chord. This method is depicted in Figure 4.9. Figure 4.10 shows the shift in the aerodynamic center and increase in chord for some percent of damage.

The new mass of the aircraft is determined based on empirical mass buildups for similar aircraft and the wing CG is assumed to be at 40% of the mean aerodynamic chord [34] and is set at half span. A mass buildup of the undamaged aircraft is adjusted to produce the same CG and Inertia as the production aircraft. As tip portions of the left wing are removed for damage, the wing mass lowers, and its CG shifts inboard. The aircraft CG then shifts positive along body centered Y axis and positive along the body centered X axis (starboard and forward). Any shift along the Z axis is considered negligible.

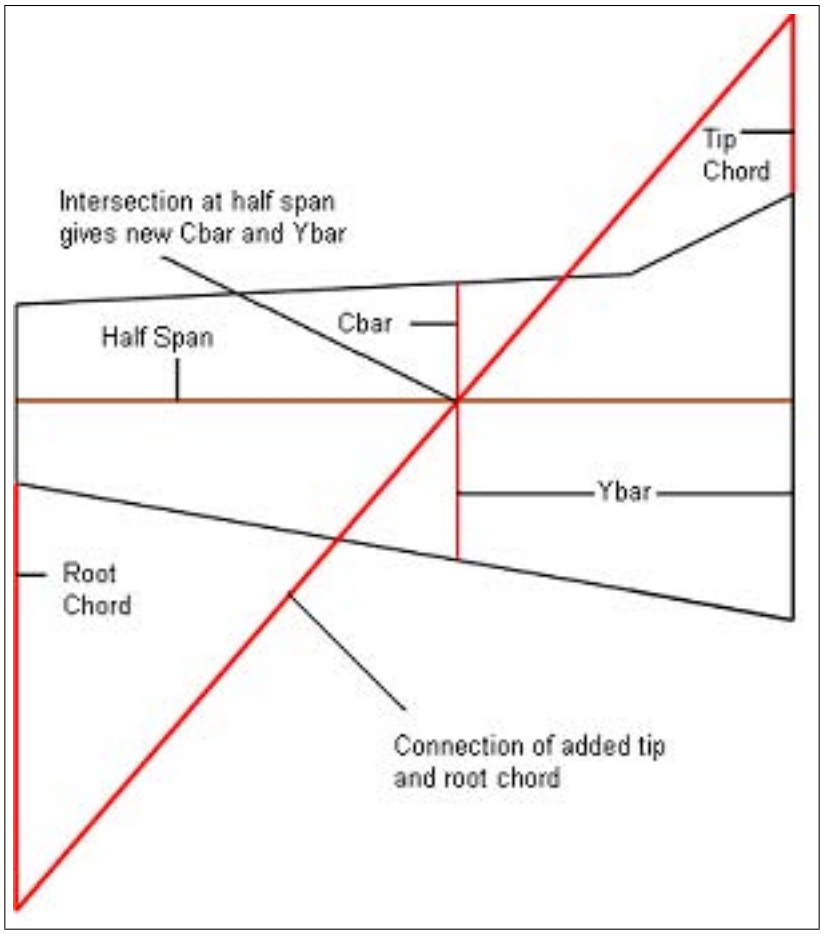


Figure 4.9: Aerodynamic Chord and Ybar Determination

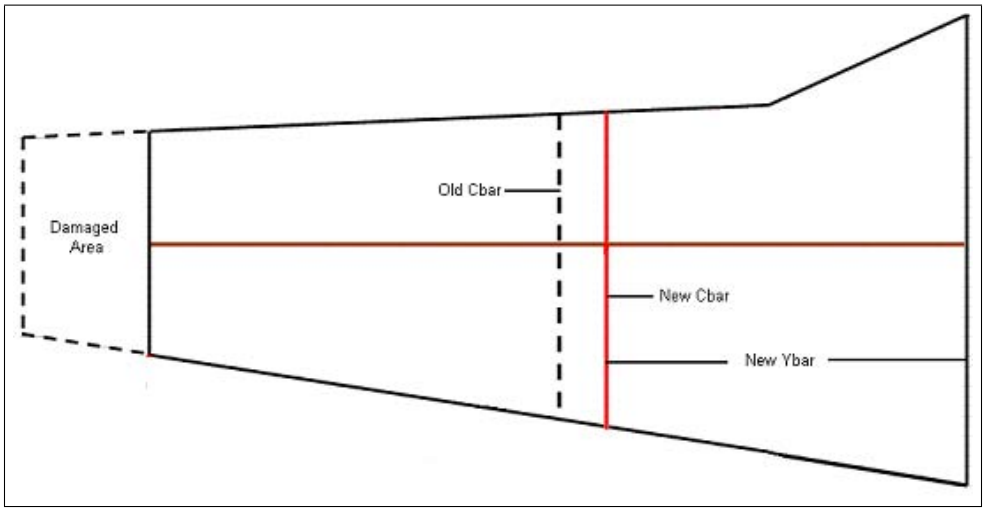


Figure 4.10: Shift in Aerodynamic Chord and Ybar

Table 4.8 gives the percentage damage to half of a wing, the resulting span, new chord, and CG Shift, while Table 4.9 shows the mass and Inertia versus percentage damage.

Table 4.8

Damaged Bonanza Geometric Data

	Geometry (feet)		CG Shift (feet)	
Damage	Span	Chord	Δx	Δy
10%	31.18	5.49	0.0001	0.09
15%	30.36	5.54	0.0022	0.14
20%	29.54	5.57	0.0043	0.18
25%	28.72	5.59	0.0064	0.22
30%	27.90	5.63	0.0085	0.26
35%	27.07	5.66	0.0107	0.29
40%	26.25	5.70	0.0128	0.33
45%	25.43	5.74	0.0150	0.36
50%	24.61	5.77	0.0171	0.38

Δz is negligible and not shown.

Table 4.9

Damaged Bonanza Mass Data

		Moments & Products of Inertia (slugs/ft^2)					
Damage	Mass (slugs)	Ixx	Iyy	Izz	Ixz	Ixy	Iyz
10%	101.86	1264.55	1919.40	3183.95	-48.48	-6.56	0.00
15%	101.55	1216.52	1919.25	3135.77	-48.48	-9.60	0.00
20%	101.24	1173.83	1919.11	3092.94	-48.48	-12.45	0.00
25%	100.93	1136.16	1918.97	3055.13	-48.48	-15.14	0.00
30%	100.62	1103.14	1918.82	3021.96	-48.48	-17.65	0.00
35%	100.31	1074.56	1918.68	2993.24	-48.48	-19.99	0.00
40%	100.00	1050.06	1918.54	2968.59	-48.48	-22.16	0.00
45%	99.69	1029.32	1918.39	2947.71	-48.48	-24.15	0.00
50%	99.38	1011.99	1918.25	2930.24	-48.48	-25.97	0.00

Ixz and Iyz exhibit no change since Δz is negligible.

4.3.2 Coefficient Estimations

Baseline undamaged derivative estimates are found with methods in the previous section and from the geometric and lifting surface data presented in Table 4.10 for the Beechcraft Bonanza. Table 4.11 gives the mass properties for the Bonanza.

From this data, the aircraft coefficients in Table 4.12 are selected for the baseline and damage estimation. The damping derivatives (C_{l_p}, C_{n_r}, C_{m_q}, etc) are not included, but this keeps the simulator more conservative, as discussed in Section 3.6. Drag coefficient changes are not performed, only the aspect ratio change is

Table 4.10

Bonanza Geometric Data

Main Wing		**Horizontal Tail**		**Vertical Tail**	
Span (ft)	5.4	Span (ft)	11.16	Span (ft)	4.8
MAC (ft)	5.4	Chord (ft)	3	Chord (ft)	3.4
Root Airfoil	23015	Ce/Ct	0.35	Cr/Ct	0.4
Tip Airfoil	23012	Airfoil	0009	Airfoil	0009

Note: All airfoils are NACA types. Wing dihedral is 6 degrees.

Table 4.11

Bonanza Mass Properties

Mass (slugs)	Weight (lbs)	Moments & Products of Inertia (slugs/ft^2)					
		Ixx	Iyy	Izz	Ixy	Iyz	Ixz
102.5	3300	1377	1919	3235	0	0	-48.5

included for induced drag, which will increase its affect.

Table 4.12

Estimated Aircraft Coefficients

Longitudinal				**Lateral-Directional**			
C_{L_0}	C_{L_α}	$C_{L_{\delta e}}$	-	C_{Y_0}	C_{Y_β}	-	$C_{Y_{\delta r}}$
C_{M_0}	C_{M_α}	$C_{M_{\delta e}}$	$C_{M_{\delta a}}$	C_{l_0}	C_{l_β}	$C_{l_{\delta a}}$	$C_{l_{\delta r}}$
-	-	-	-	C_{N_0}	C_{N_β}	$C_{N_{\delta a}}$	$C_{N_{\delta r}}$

Damaged aircraft estimations for the standard coefficients use the same methods that are used to produce the baseline undamaged values. For the loss of a lifting surface, the aspect ratio change alters the finite wing lift curve slope of the lifting surface. No Mach effects are considered for the test aircraft since the flight speeds are minimal enough to assume little compressibility effects. The transformation from 2-D to a finite wing lift curve slope assuming no compressibility is given by equation 4.102 [34]. Airfoil data is taken from NACA Report 824 [37].

$$A_{3D} = \frac{A_{2D}}{1 + \frac{A_{2D}}{\pi e AR}} \tag{4.102}$$

Asymmetric coefficients are found for a range of percent damages to half of the main wing from the new aerodynamic centers and chords in Section 4.3.1. Given some trim velocity, the lift on either wing is determined from an area ratio of the total lift, which is set equal to the damaged aircraft weight. Drag is separated on each wing the same way. Since the damaged aircraft is asymmetric, C_{l_0} and C_{N_0} are non zero

and calculated from equations (4.103) and (4.104). For these definitions, the subscripts "l" and "r" refers to values on the left and right wing, respectively. Similarly, the asymmetric sideforce C_{Y_0} is nonzero for the damaged aircraft and is calculated from equation (4.105), and is the difference between the body axis Y components of lift.

$$C_{l_0} = \frac{Lift\ (S_l/S\ y_{bar-l} - S_r/S\ y_{bar-r})}{qSb} \tag{4.103}$$

$$C_{N_0} = \frac{Drag\ (S_l/S\ y_{bar-l} - S_r/S\ y_{bar-r})}{qSb} \tag{4.104}$$

$$C_{Y_0} = \frac{Lift\ (S_l/S\ sin(\Gamma) - S_r/S\ sin(\Gamma))}{qSb} \tag{4.105}$$

Control coefficients are obtained using flap lift and pitching moment data and control effectiveness estimations from empirical formulas and charts [31] [33]. Aileron control power comes from a strip method buildup pictured in Figure 4.11. As the loss of the wing removes sections of the aileron, the control power must be recalculated. The difference in left and right aileron size requires the calculation of the difference in pitching moment and lift change.

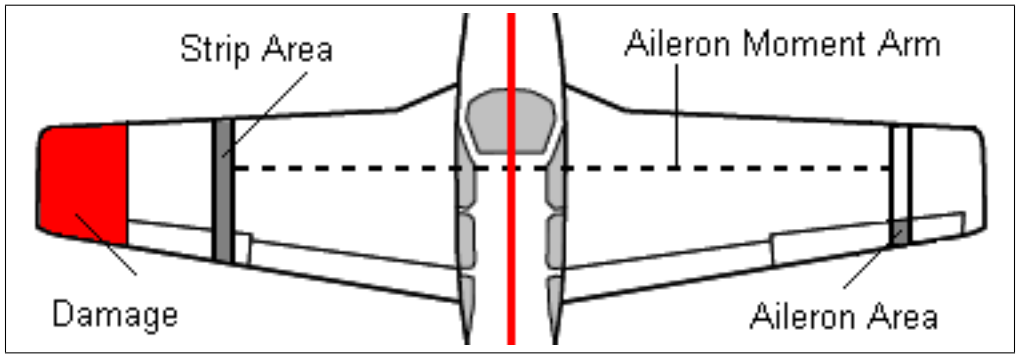


Figure 4.11: Strip Method for Aileron Control Power

4.3.3 Test Aircraft Data

Data for the test aircraft is presented in Table 4.13. These values are the baselines that the percent change in coefficients are applied to. Following the coefficient calculation from Figure 4.8, coefficient values that are zero for an undamaged aircraft are given in Figures 4.12 and 4.13. Percentage changes for coefficients that are non zero on an undamaged aircraft are given in Figures 4.14 to 4.15. Positive percent changes reflect an increase of the actual coefficient, and negative percent changes reflect a decrease of the actual coefficient. The new asymmetric coefficients (C_{l_0}, etc...) are presented at two different flight speeds of 100 and 140 knots to illustrate the magnitude decrease as the required angle of attack to achieve level flight lift decreases.

Any coefficients that are affected by the aileron see no change until after 10% damage to left wing, since the F-33C's ailerons do not reach the wing tips.

As discussed previously, the change in the standard aircraft coefficients is primarily an effect from the decrease in the wing's aspect ratio. For non-aileron control coefficients, the forces and moments that surface exerts do not change. The difference in numerical value is caused by the normalization of the force with respect to a smaller wing area and span. Pitching moment control coefficients increase to a lesser degree since the mean aerodynamic chord increases with the amount of left wing damage. This puts a larger term in the normalization of the pitching moment control force, which counteracts the decrease in wing area. The lateral-directional coefficients see a large percentage shift because both the wing area and span are decreasing significantly.

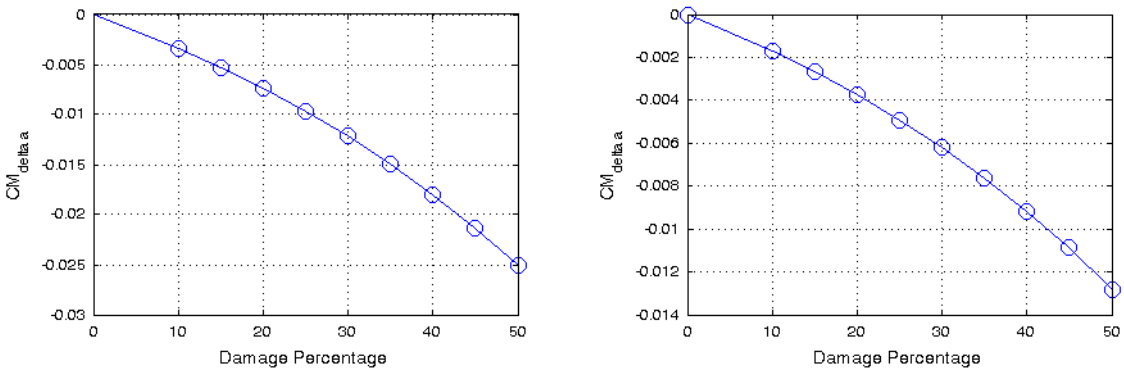
Table 4.13

Undamaged Aircraft Coefficients

Longitudinal						
	0	α	$\dot{\alpha}$	q	δe	δa
C_L	0.188	5.261	2.139	4.579	0.359	0.0
C_M	0.050	-0.805	-5.822	-13.71	-0.978	0.0

Lateral-Directional							
	0	β	$\dot{\beta}$	r	p	δa	δr
C_Y	0.0	-0.393	0.0	0.214	-0.075	0.0	0.187
C_l	0.0	-0.092	0.0	0.079	-0.484	0.229	0.015
C_N	0.0	0.059	0.0	-0.094	-0.028	-0.022	-0.065

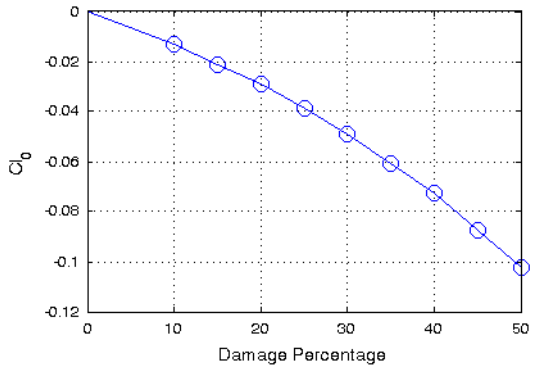
Drag Values are determined from $C_{D_0} = 0.0204$ and $C_D = C_{D_0} + C_L^2/\pi eAR$.



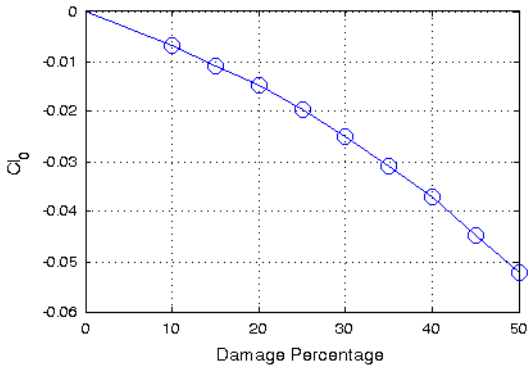
(a) $C_{M_{\delta a}}$ (b) $C_{M_{\delta a}}$

Figure 4.12: Asymmetric Control Coefficient Damage Values

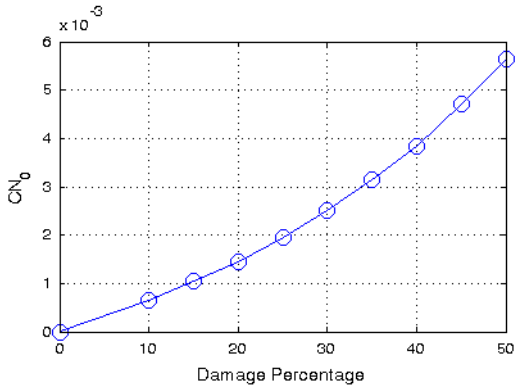
Left column values are for 100 knots. Right column values are for 140 knots.



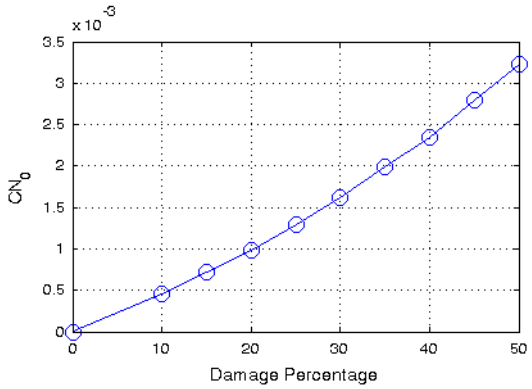
(a) C_{l_0}



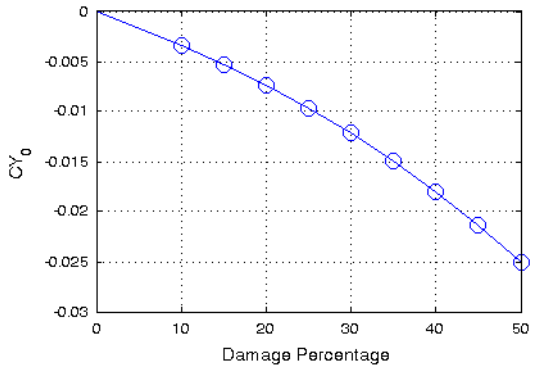
(b) C_{l_0}



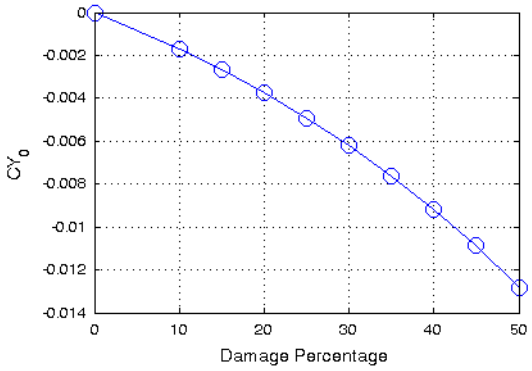
(c) C_{N_0}



(d) C_{N_0}



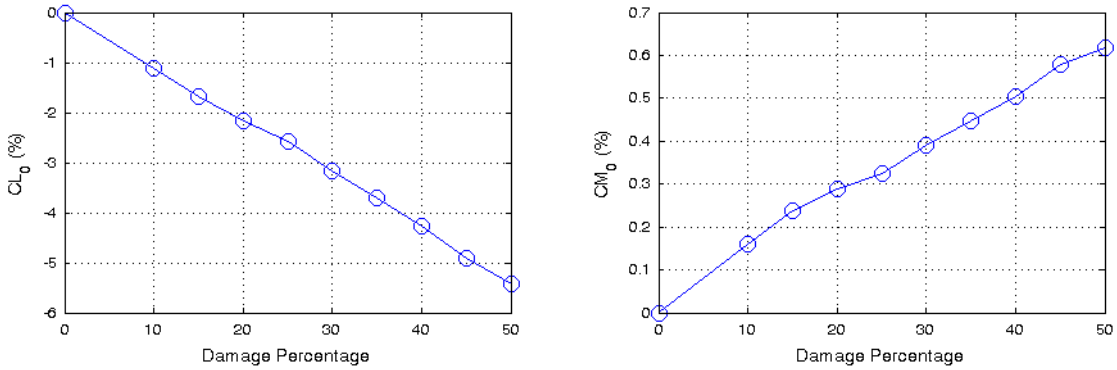
(e) C_{Y_0}



(f) C_{Y_0}

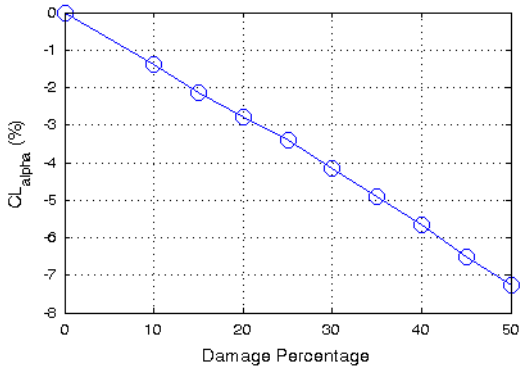
Figure 4.13: Asymmetric "0" Coefficient Damage Values

Left column values are for 100 knots. Right column values are for 140 knots.

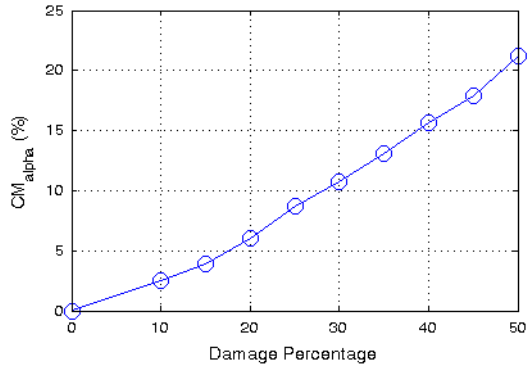


(a)

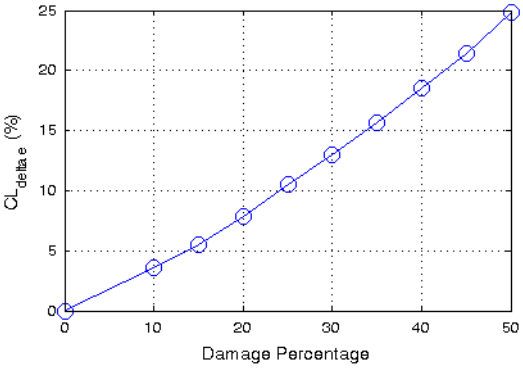
(b)



(c)



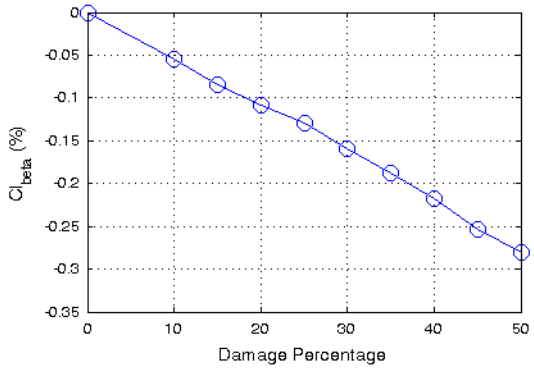
(d)



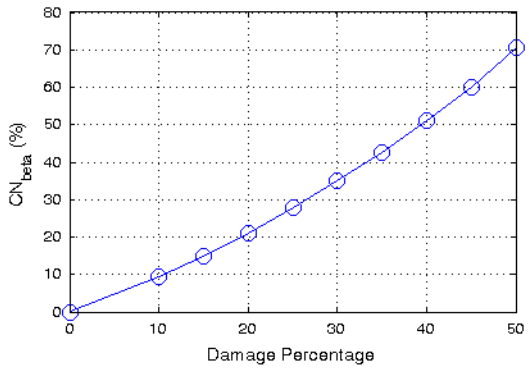
(e)

(f)

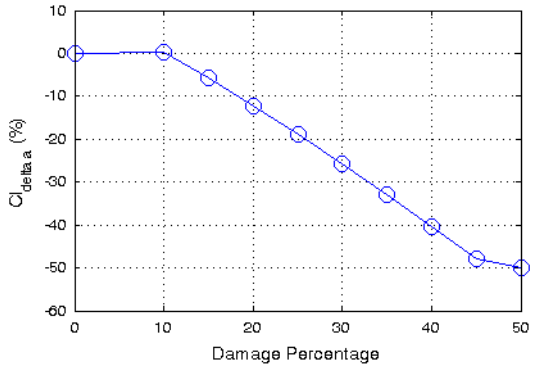
Figure 4.14: Longitudinal Coefficient Damage Values



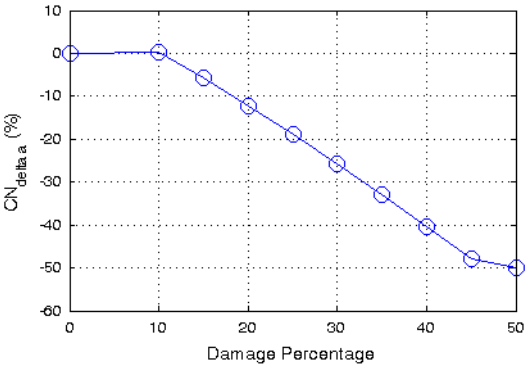
(a)



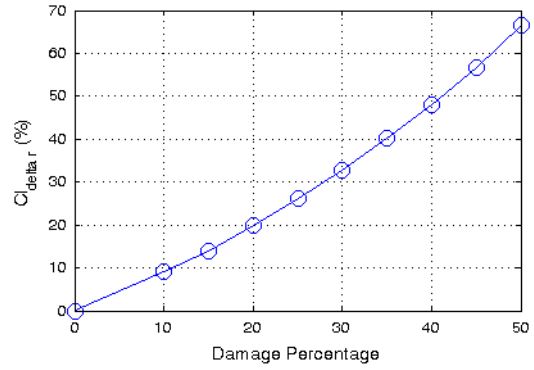
(b)



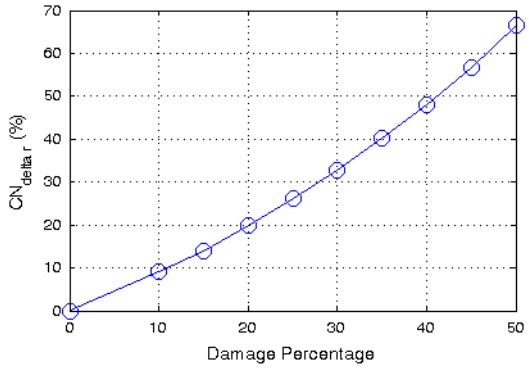
(c)



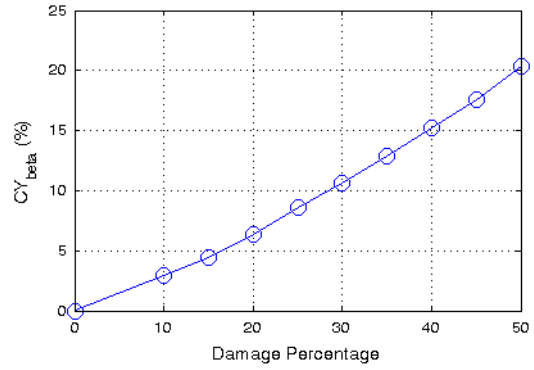
(d)



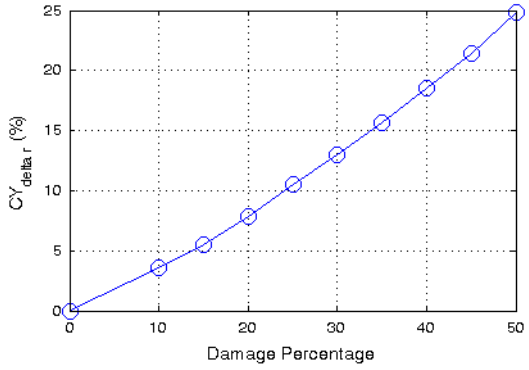
(e)



(f)



(g)



(h)

Figure 4.15: Lateral-Directional Coefficient Damage Values

Chapter 5

5 RESULTS

5.1 Test Aircraft Simulation

Testing on the DIRC and Aircraft Response Compensator is performed over a range of flight speeds from 80 to 140 knots and damages from 10% to 50% loss of the **left** wing. The slowest speed provides a trim angle of attack below 10 degrees. The maximum structural speed of the F-33C is 167 knots indicated airspeed (KIAS) [28], so the range of flight speeds tested provides a boundary of reasonable flight speeds. Figures for bank angle, aileron deflection, velocity, adaptive bias Correction (ABC) signals, and angle of attack are given at each flight speed. Bank angle, sideslip, and flight path are commanded to 0 degrees with velocity being commanded for the trim solution at that altitude and airspeed, and inverse controller outputs of control surfaces are seen in red, with the actual values presented in blue. Time history of the Roll ABC provides three separate signals. Blue shows the actual aircraft roll rate, green is the pre-corrected commanded roll rate, and the red line is the correction signal. For all ABC plots, the correction signal will be some positive number, where the actual and commanded roll rate will settle at 0 rad/sec for successful conditions. Percent damages shown cover 5% above and below the final passing condition. Damage occurs at 30 seconds. Non-critical condition results are presented in Appendix A.

Figure 5.1 plots the time response of the Bonanza at 80 knots flight speed with 20% damage to the left wing. The bank angle response is initially negative as the damaged left wing drops. The remaining response behaves similarly to a lightly damped second order system while the adaptation occurs, as opposed to the near critically damped Model Follower setup. It takes around one second for the adaptation to deflect the aileron enough to begin bringing the aircraft to a normal bank attitude. Aileron deflection does not initially settle on the steady state value to hold zero bank angle. A larger deflection is commanded as the controller must provide a larger rolling moment to roll the aircraft back to zero bank angle. From the ABC plot, it is seen that the correction converges and remains steady even as the aircraft is still not at zero bank angle. Sideslip and angle of attack both remain within linear ranges, with the angle of attack settling on a larger than the pre-damage trim value to counteract the loss of lifting surface. Velocity increases slightly which allows the angle of attack to increase smaller than without any velocity increase.

Figure 5.2 plots the time response of the Bonanza at 80 knots flight speed with 25% damage to the left wing. A similar bank angle, sideslip, velocity, and angle of attack response appears in this test condition. Angle of attack does become large at 11.5 degrees, but is acceptable for the short excursion to this angle. Roll rate correction overshoots the steady state value by double as the controller commands a faster deflection

than the lag in the aileron can provide. This discrepancy leads the ABC to want more aileron deflection until the maximum deflection is reached and the roll rate begins to return the aircraft to zero bank angle.

Figure 5.3 plots the time response of the Bonanza at 80 knots flight speed with 30% damage to the left wing. This condition requires aileron deflections beyond the maximum to hold a bank angle of zero. Full aileron deflection is insufficient to recover the aircraft, and it loses airspeed and enters stall regions of angle of attack. If adequate aileron did exist, the aircraft could be recovered, as the loss of lifting surface does not require an angle of attack in a stall reqion to maintain level flight.

Figure 5.4 plots the time response of the Bonanza at 100 knots flight speed with 25% damage to the left wing. This condition is easily controlled, and the ABC adapts fast and the aircraft banks at angles smaller than for some standard manuevers. The remaining parameters behave similarly to the previous passing conditions, with an increasing velocity and aerodynamic angles well within linear regions.

Figure 5.5 plots the time response of the Bonanza at 100 knots flight speed with 30% damage to the left wing. Bank angle reaches a maximum of -20 degrees (left wing down) in around one second and settles very close to zero degrees within 10 to 15 seconds. After an initial overestimation of the required aileron deflection, the ABC settles quickly on the correct solution. Sideslip and angle of attack present no issues here. There is a noticeable difference in settling time between this condition and the one in Figure 5.4. Less damage took longer to settle to zero degrees bank angle, but had smaller bank angles overall. This difference comes from the ABC itself, where the magnitude of the error determines the amount of correction added. Less damage, and therefore less error, adds a smaller correction and gives a slower response.

Figure 5.6 plots the time response of the Bonanza at 100 knots flight speed with 35% damage to the left wing. 19.9 degrees of aileron deflection is the calculated value to hold zero degrees bank angle. Within this flight speed, the actual aileron deflection is a tenth or more of a degree than the calculated deflection. Full aileron deflection is commanded by the ABC, and the aircraft does begin to return to zero bank angle.

Figure 5.7 plots the time response of the Bonanza at 120 knots flight speed with 30% damage to the left wing. Only 10 degrees of aileron deflection is necessary to recover the aircraft from the damage. Again, the sideslip, angle of attack, and velocity all behave as in prior controlled conditions. The maximum bank angle seen is only 15 degrees left wing down with an overshoot of 5 degrees left wing up. A better performance is possible, but will come at the cost of controlling less damage.

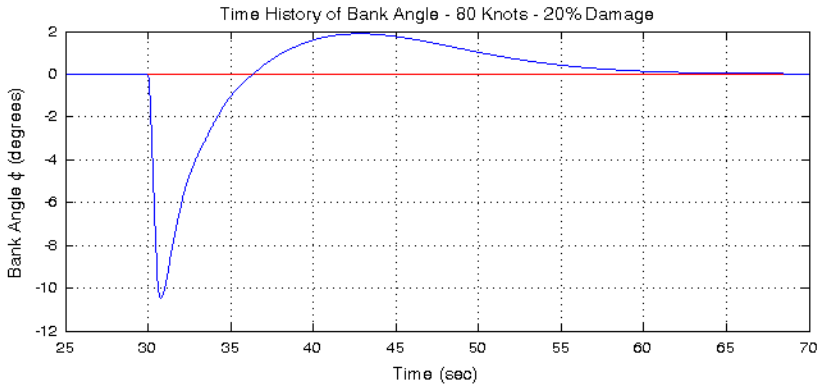
Figure 5.8 plots the time response of the Bonanza at 120 knots flight speed with 35% damage to the left wing. The sideslip and angle of attack respond as previously, with the angle of attack increasing slightly to account for the loss of lifting surface. The ABC adapts quickly to the required aileron deflection, and the bank angle only drops to 21 degrees left wing down with an overshoot smaller than the prior condition. Velocity increases by nearly one knot.

Figure 5.9 plots the time response of the Bonanza at 120 knots flight speed with 40% damage to the left wing. Aileron deflections near maximum allowable are necessary to recover the aircraft under this flight condition. Since there is little margin above the required steady state deflection, the recovery of the aircraft will be slower than what the control architecture is commanding for the 3^{rd} Order response. The ABC then tries to command more aileron deflection to increase the speed of the response, which delays the ABC from settling on an appropriate correction. This delay causes the odd response in bank angle, but it is still a response that returns the aircraft to zero bank angle, with sideslip, velocity, and angle of attack all within limits.

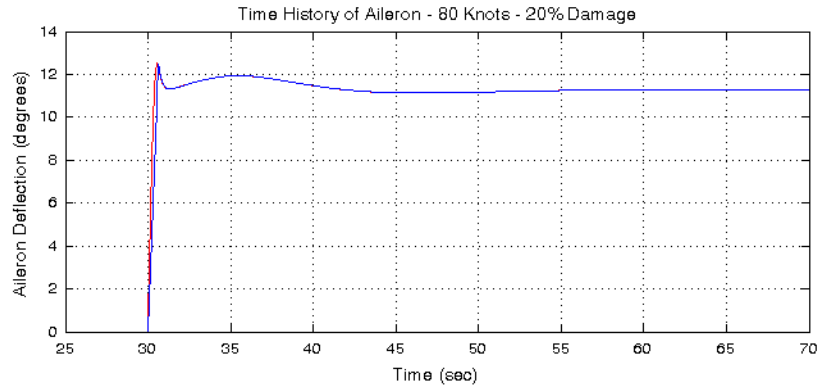
Figure 5.10 plots the time response of the Bonanza at 140 knots flight speed with 35% damage to the left wing. Sideslip, velocity, and angle of attack behave similarly to other controllable flight conditions. The bank angle response is different in this instance, and has a larger overshoot upon recovery. From the ABC plot, it is observed that unlike prior conditions, the correction signal increases from the initial estimation. Instead of having to slow down the aircraft response, the ABC must quicken it which causes the larger overshoot.

Figure 5.11 plots the time response of the Bonanza at 140 knots flight speed with 40% damage to the left wing. This condition continues to demonstrate the oscillatory settling of the bank angle response. Here the ABC follows the more prevalent pattern of overcorrecting with a larger aileron deflection to recover the aircraft, and then a settling to the steady state correction. The sideslip response varies from the majority of other flight conditions, and hangs at 1 degree instead of returning smoothly to zero degrees.

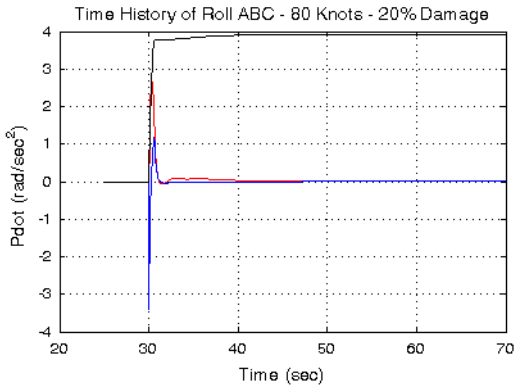
Figure 5.12 plots the time response of the Bonanza at 140 knots flight speed with 45% damage to the left wing. For this percentage of damage, all the left aileron was missing, but the 1-D estimation for aileron deflection requires only 19 degrees of deflection for the remaining control surface. Initially, the response of all parameters is similar to that of the controlled conditions. The velocity increases slightly, as does angle of attack before it drops off. Sideslip becomes negative and aids in the recovery. However, the large drop in bank angle along with the large loss of lift from losing nearly half of the wing causes the aircraft to lose altitude fast. The mixing of altitude loss, roll rate, and sideslip make this a much more difficult dynamic case. Also, the ABC correction peaks much higher than any previous flight condition, which illustrates that the ABC learning rates may be set too high for such a condition.



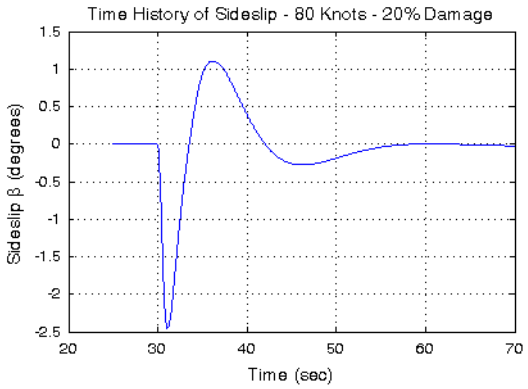
(a)



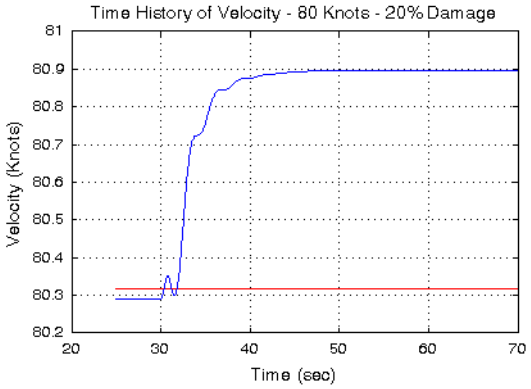
(b)



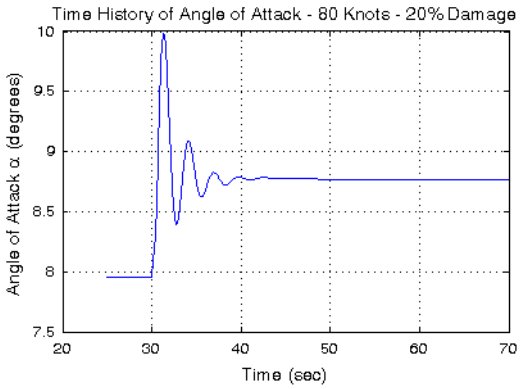
(c)



(d)

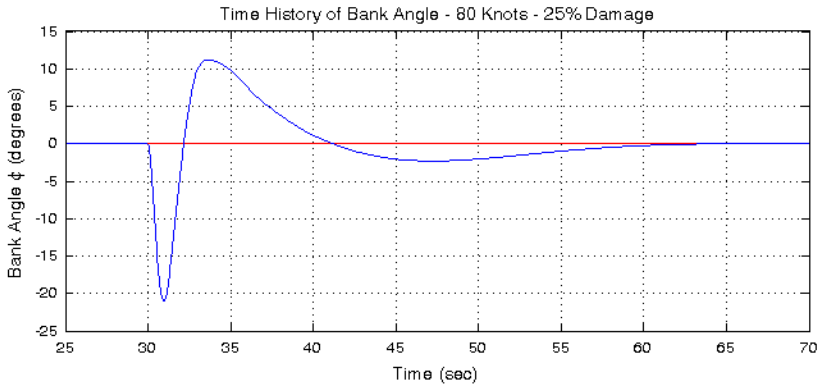


(e)

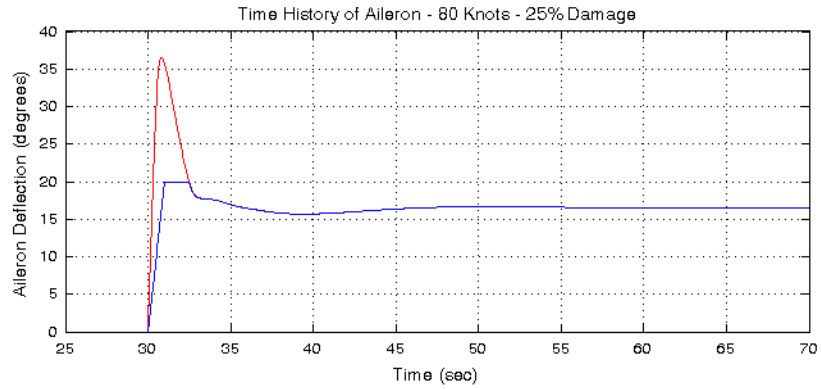


(f)

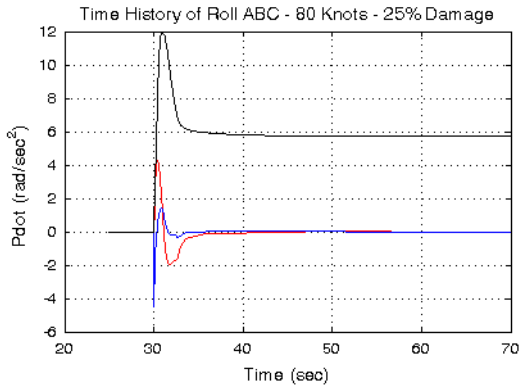
Figure 5.1: Simulation Results at 20% Damage at 80 Knots



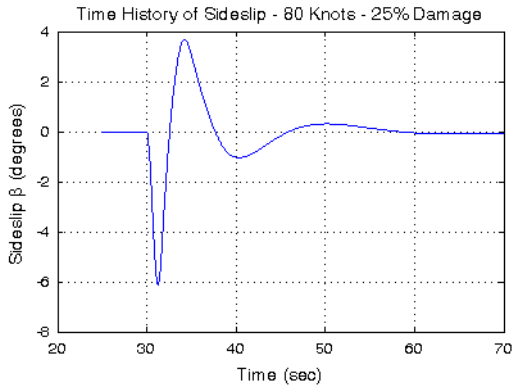
(a)



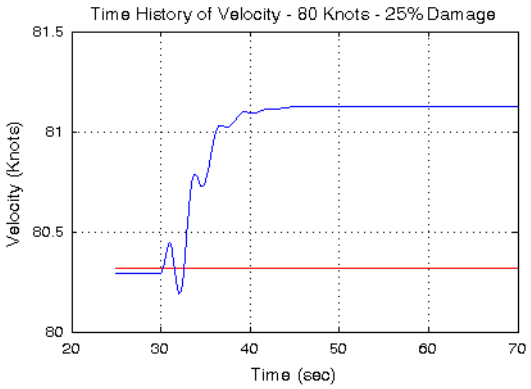
(b)



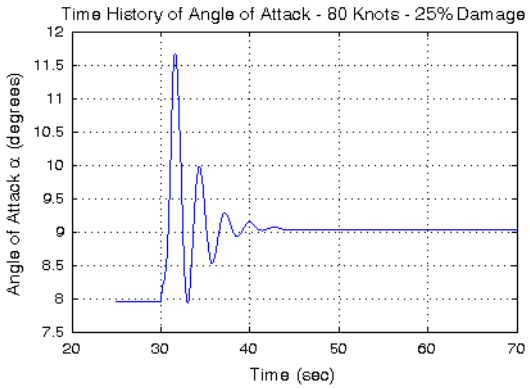
(c)



(d)

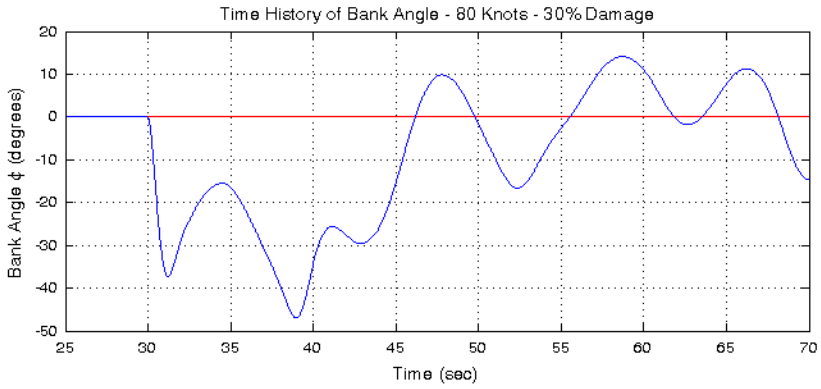


(e)

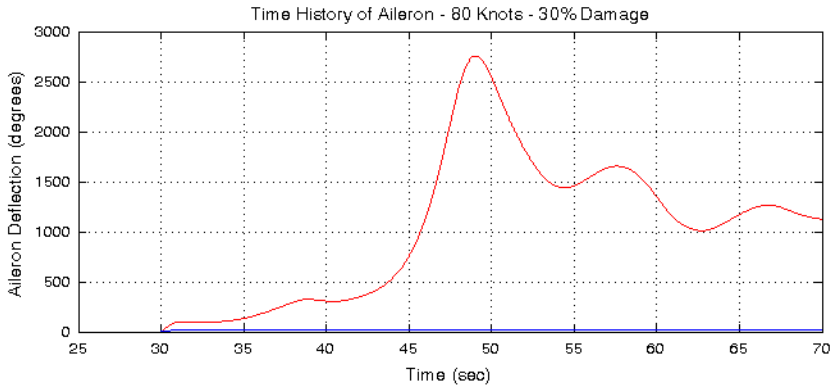


(f)

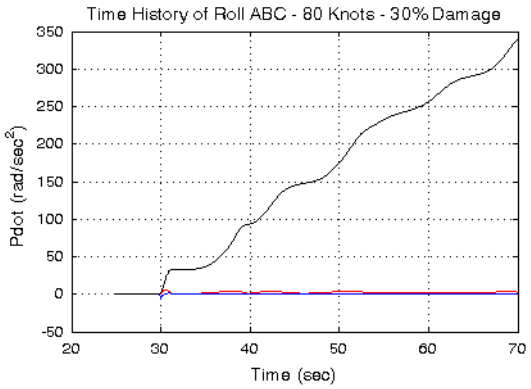
Figure 5.2: Simulation Results at 25% Damage at 80 Knots



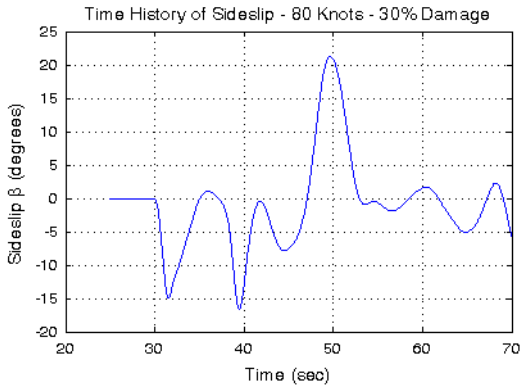
(a)



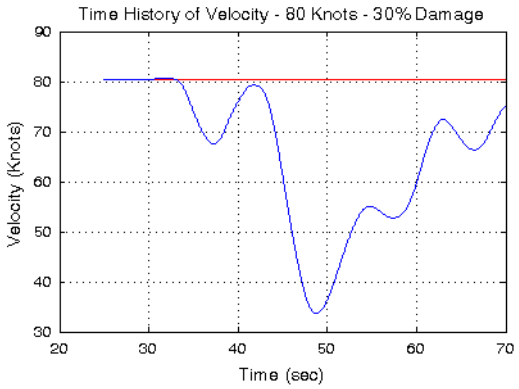
(b)



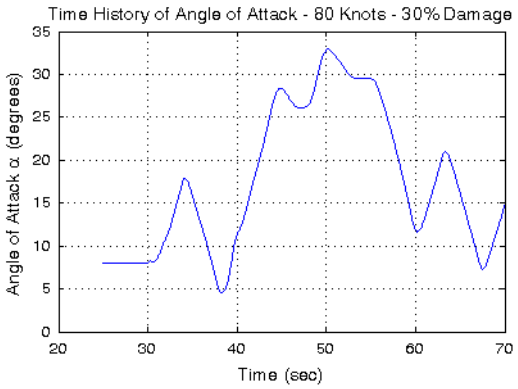
(c)



(d)

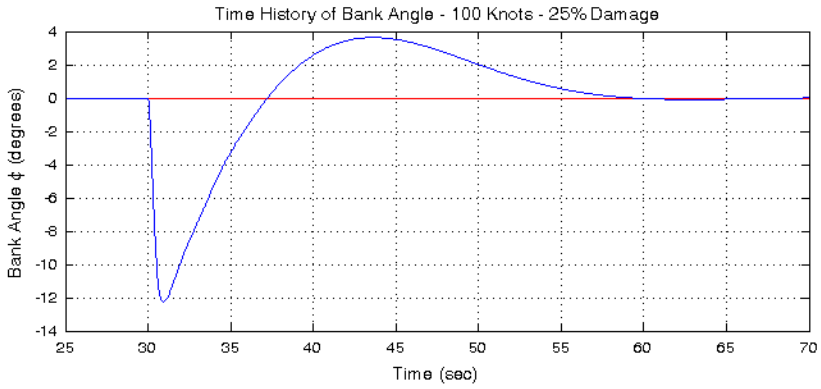


(e)

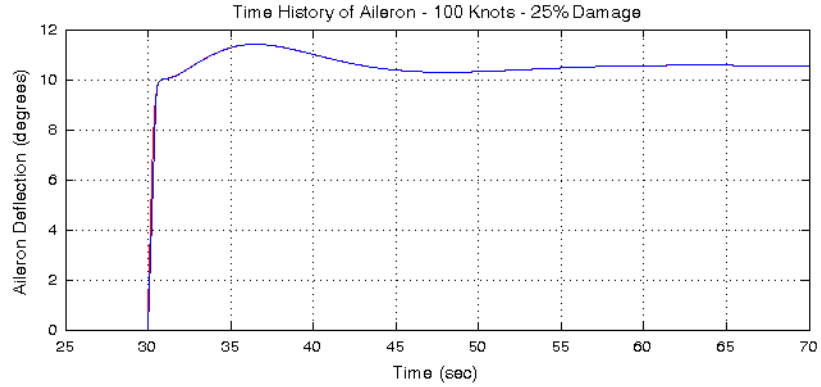


(f)

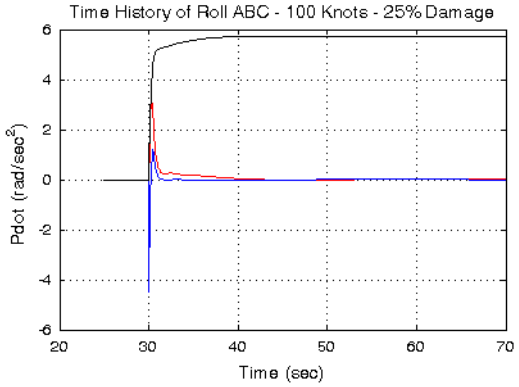
Figure 5.3: Simulation Results at 30% Damage at 80 Knots



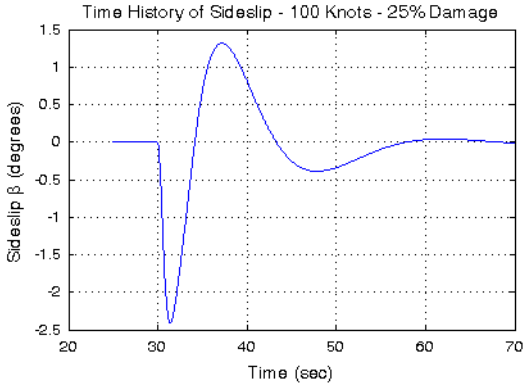
(a)



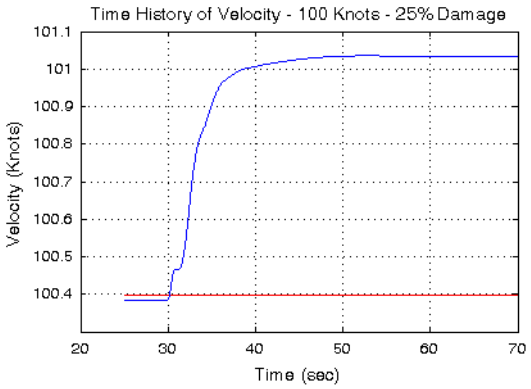
(b)



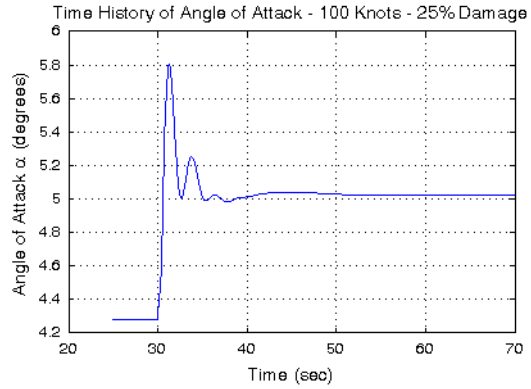
(c)



(d)

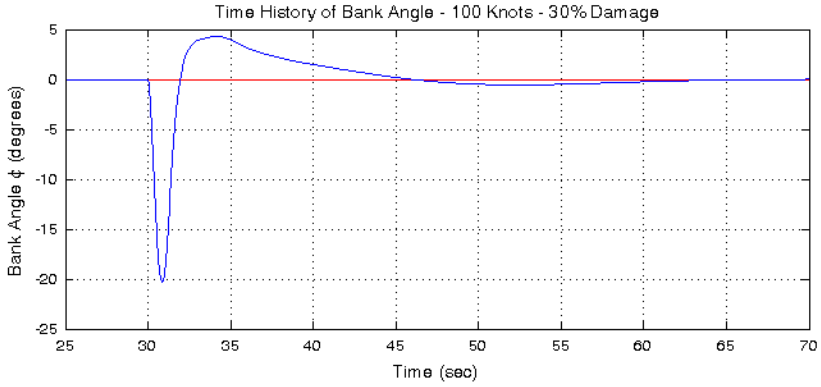


(e)

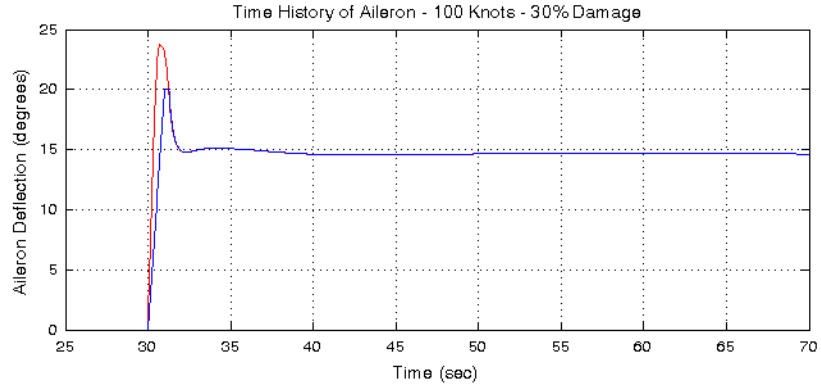


(f)

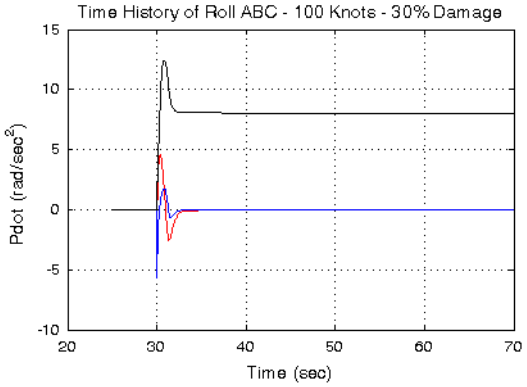
Figure 5.4: Simulation Results at 25% Damage at 100 Knots



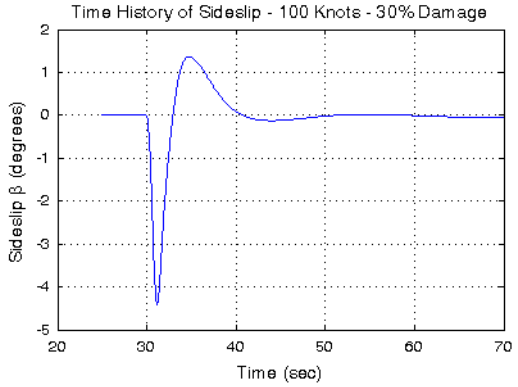
(a)



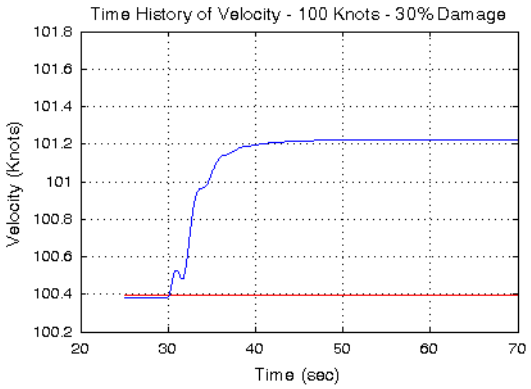
(b)



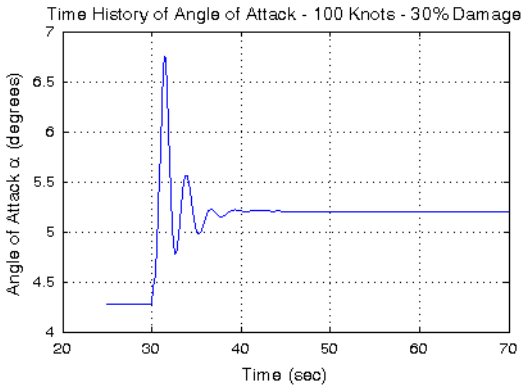
(c)



(d)

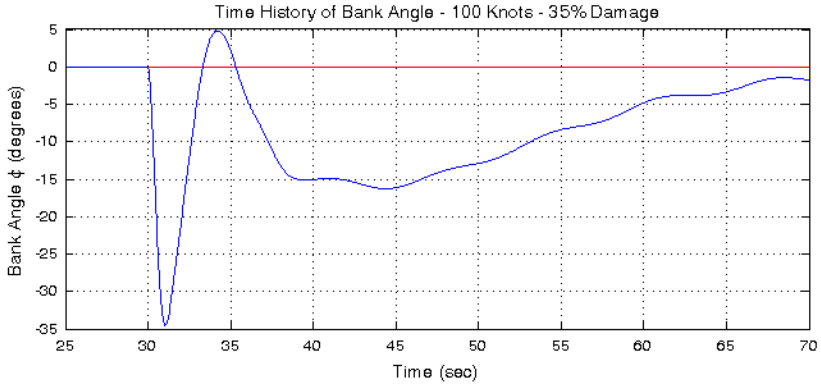


(e)

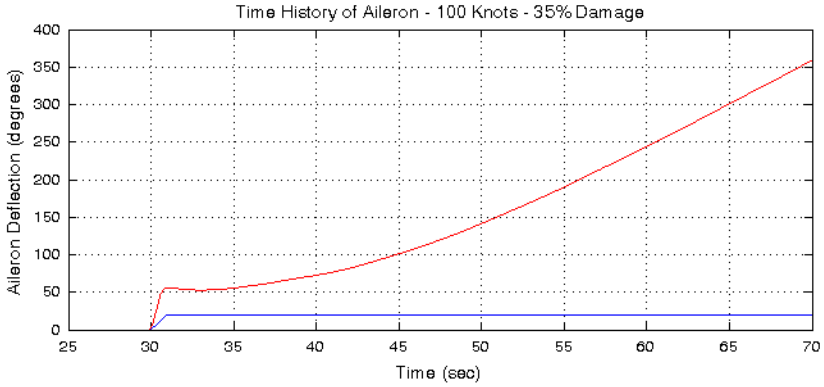


(f)

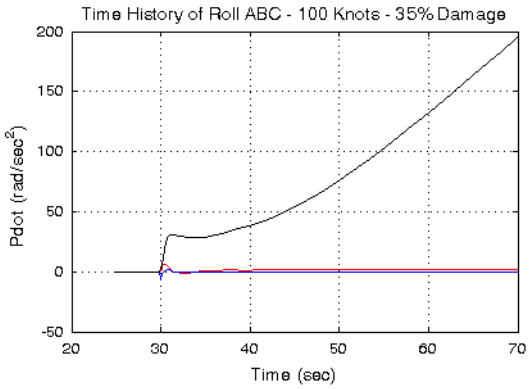
Figure 5.5: Simulation Results at 30% Damage at 100 Knots



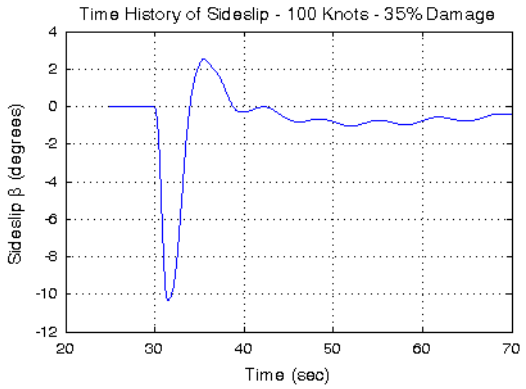
(a)



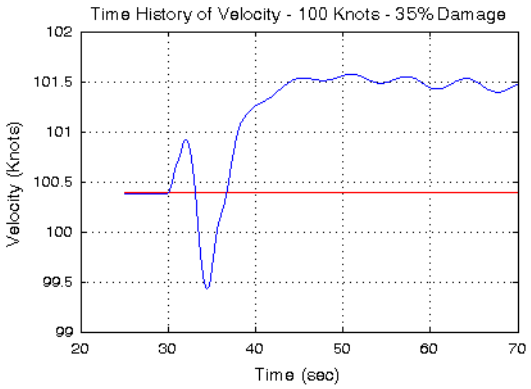
(b)



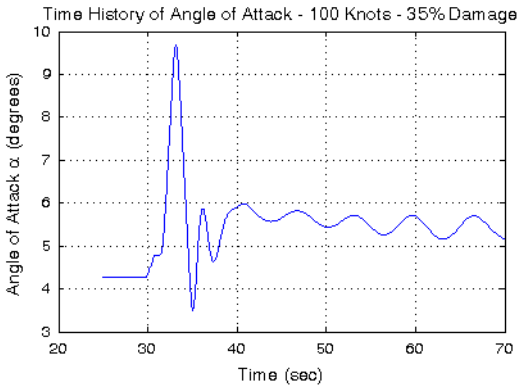
(c)



(d)

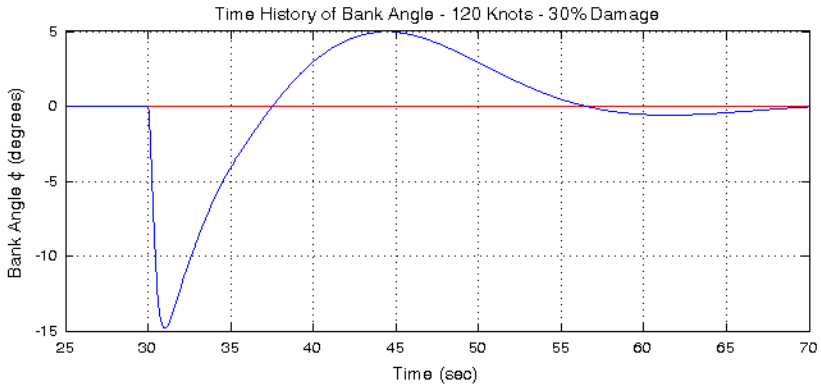


(e)

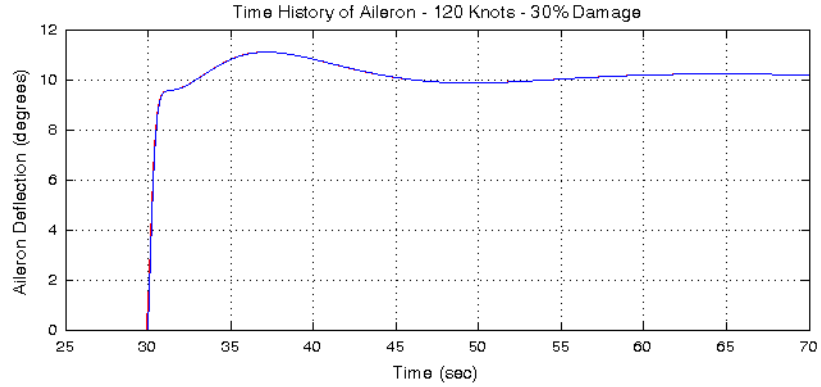


(f)

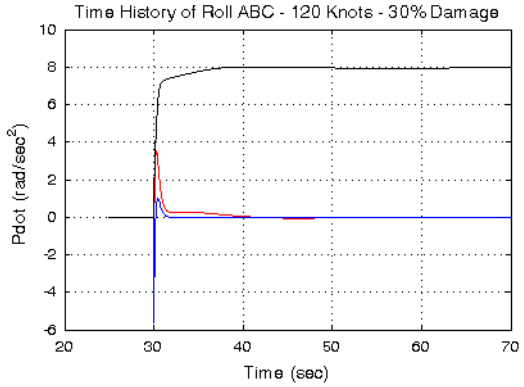
Figure 5.6: Simulation Results at 35% Damage at 100 Knots



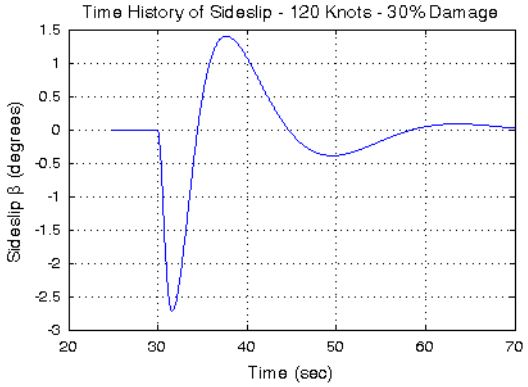
(a)



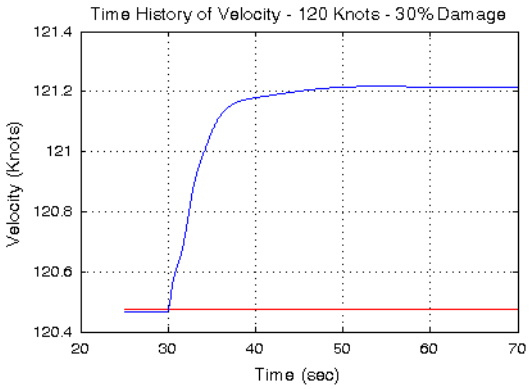
(b)



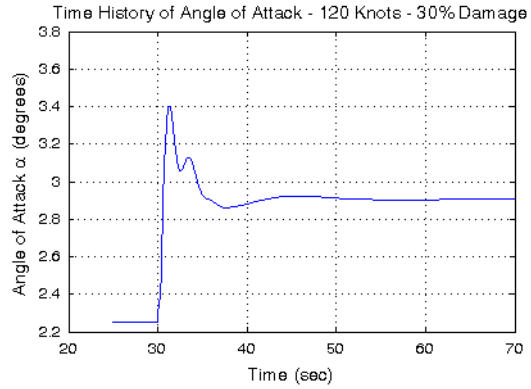
(c)



(d)

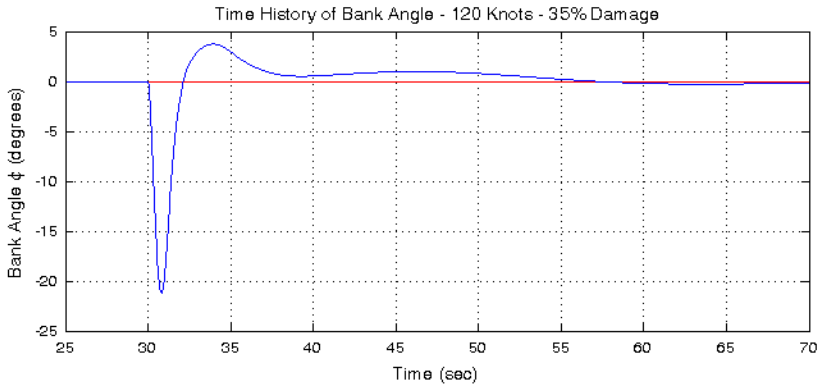


(e)

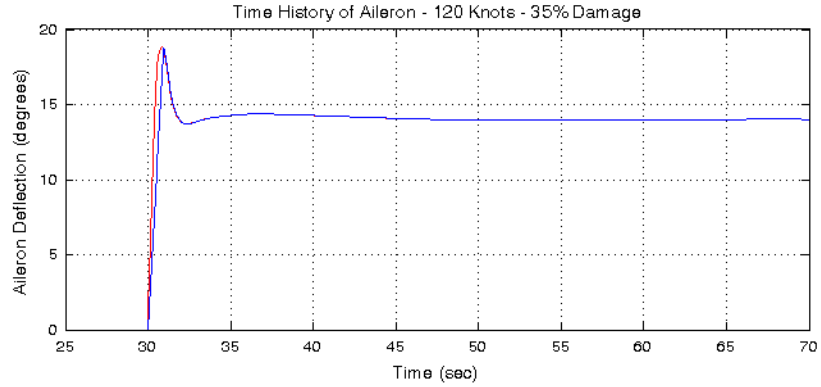


(f)

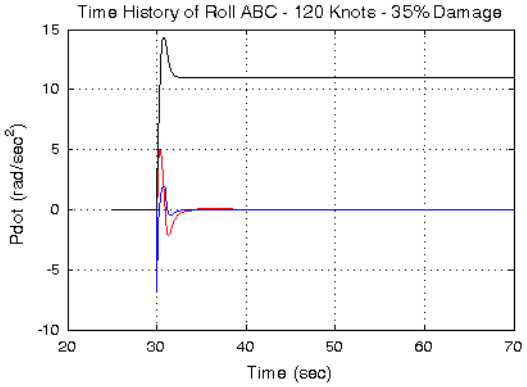
Figure 5.7: Simulation Results at 30% Damage at 120 Knots



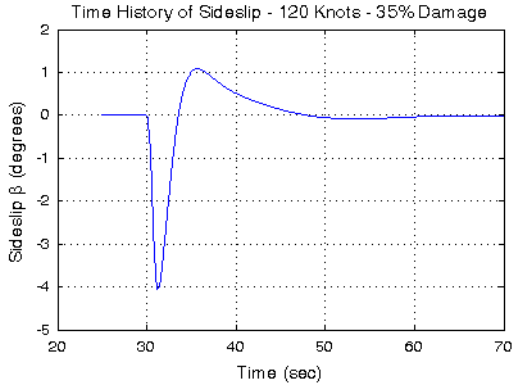
(a)



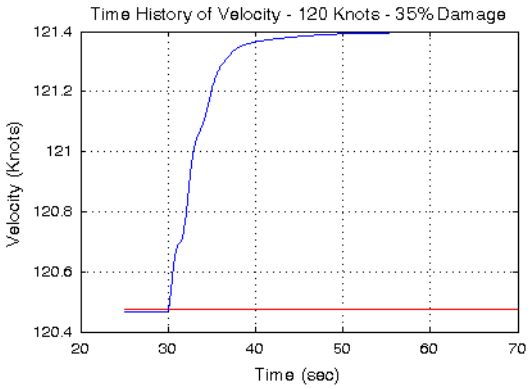
(b)



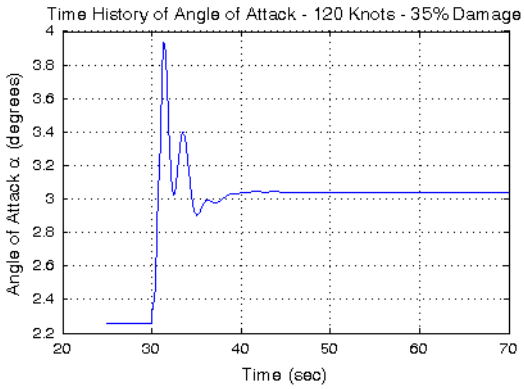
(c)



(d)

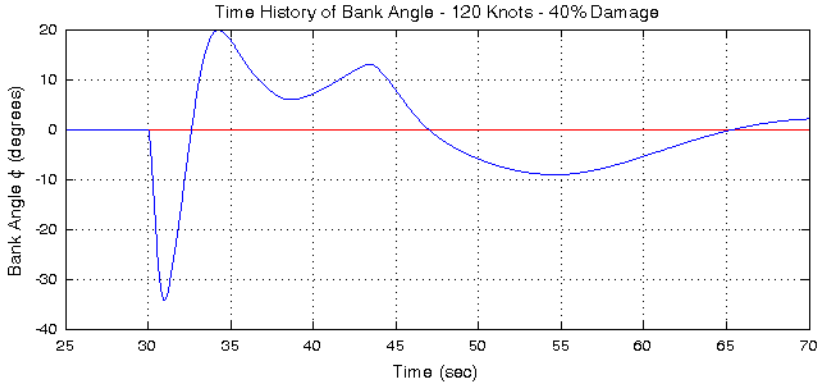


(e)

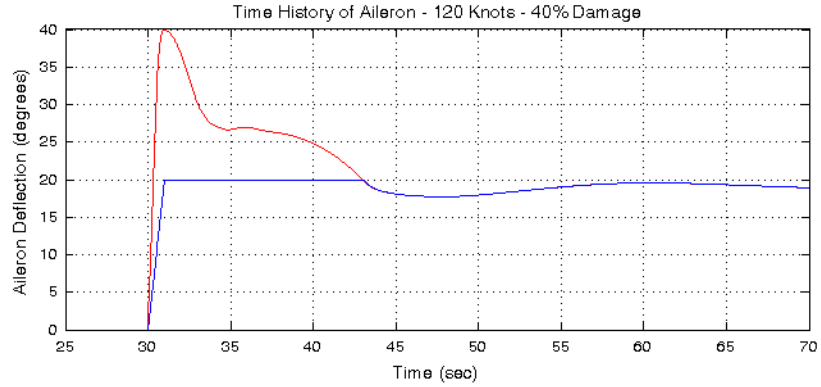


(f)

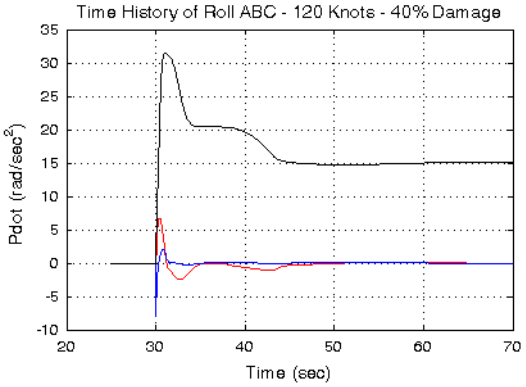
Figure 5.8: Simulation Results at 35% Damage at 120 Knots



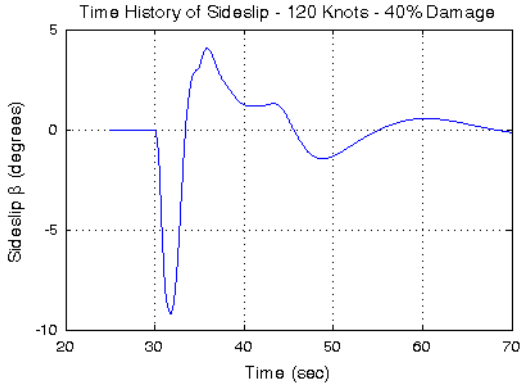
(a)



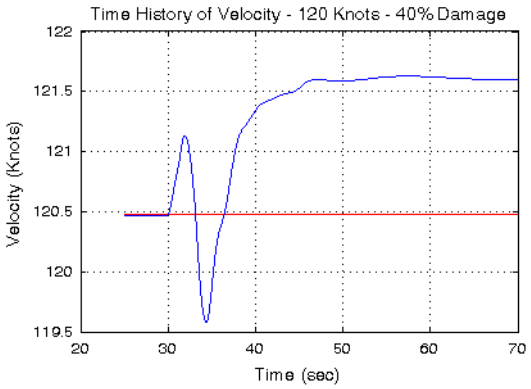
(b)



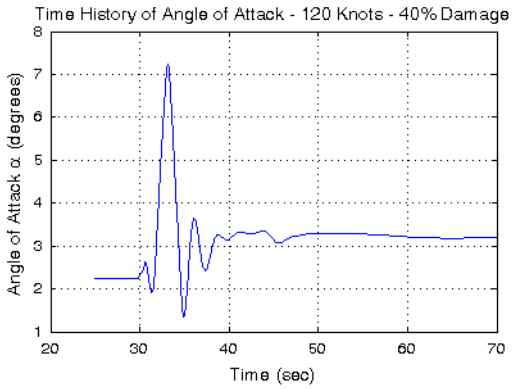
(c)



(d)

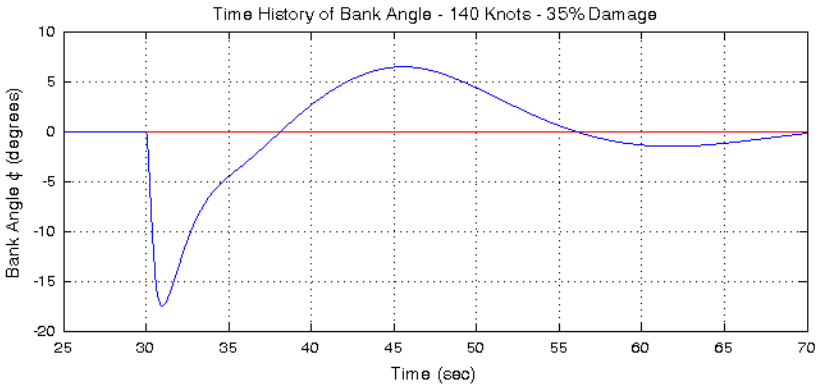


(e)

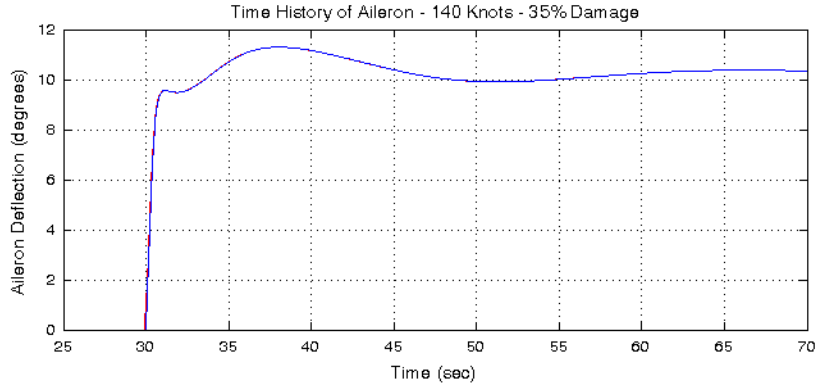


(f)

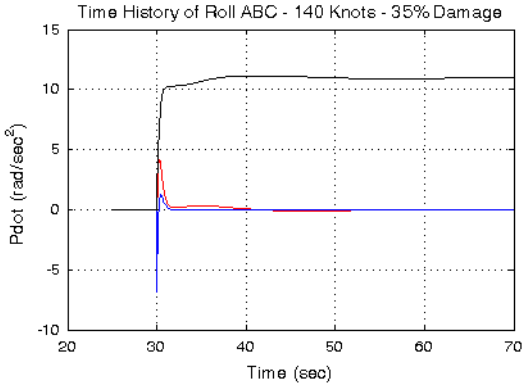
Figure 5.9: Simulation Results at 40% Damage at 120 Knots



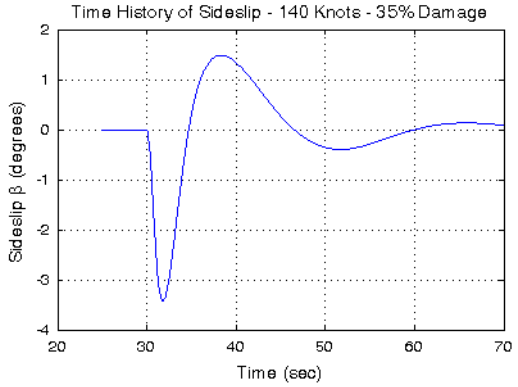
(a)



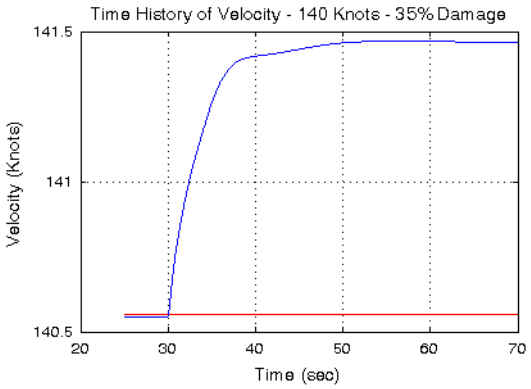
(b)



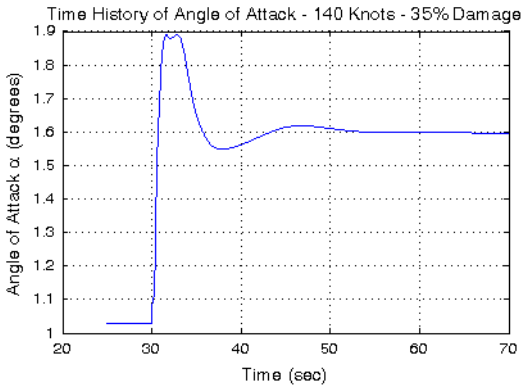
(c)



(d)

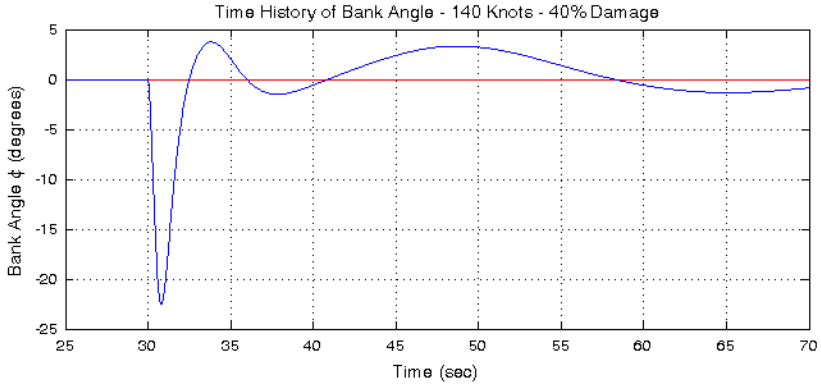


(e)

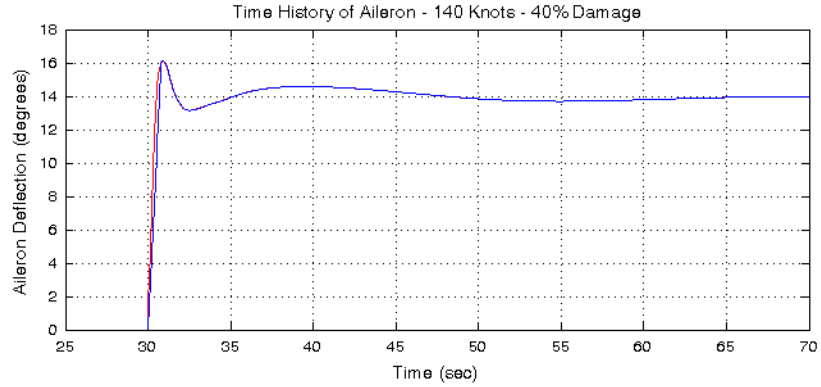


(f)

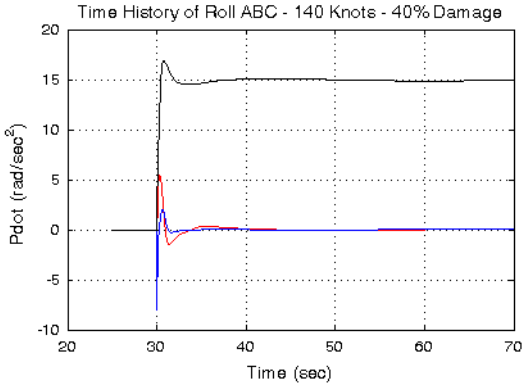
Figure 5.10: Simulation Results at 35% Damage at 140 Knots



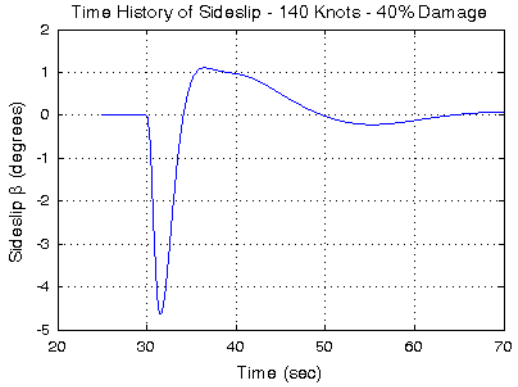
(a)



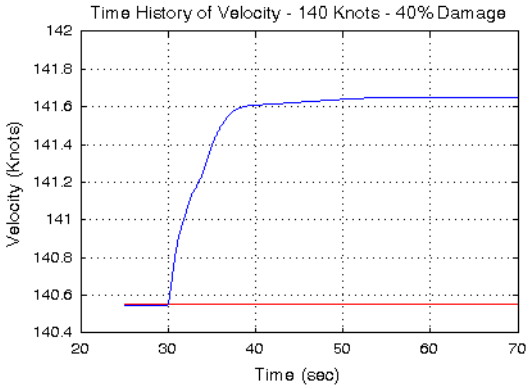
(b)



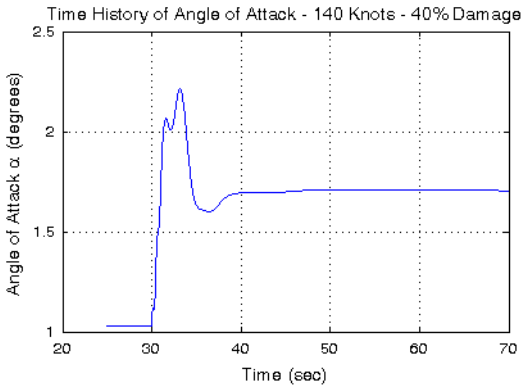
(c)



(d)

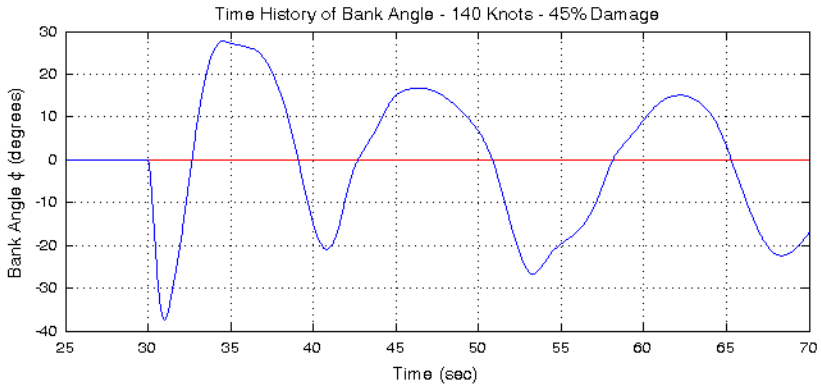


(e)

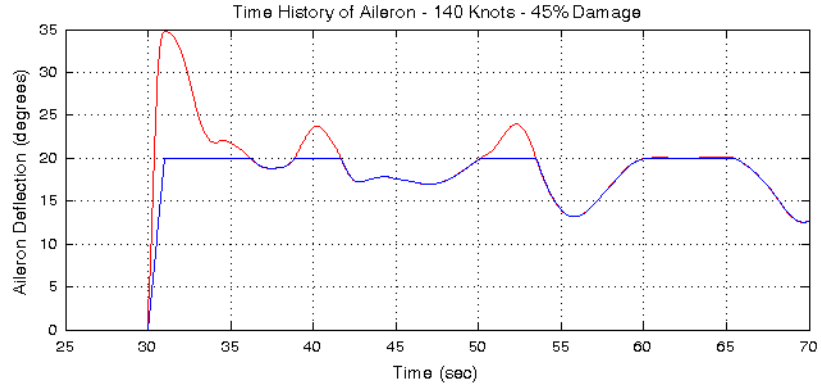


(f)

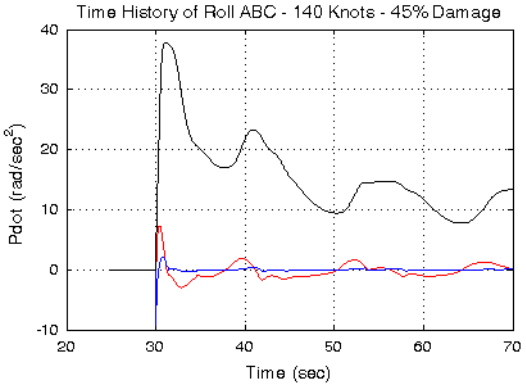
Figure 5.11: Simulation Results at 40% Damage at 140 Knots



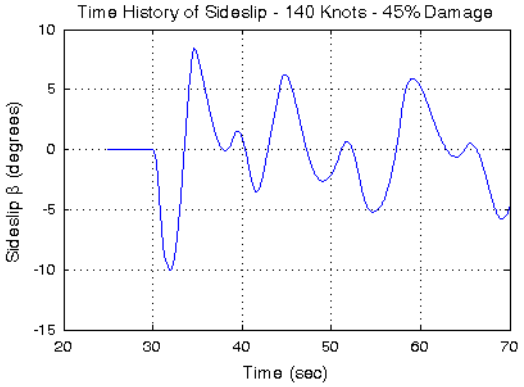
(a)



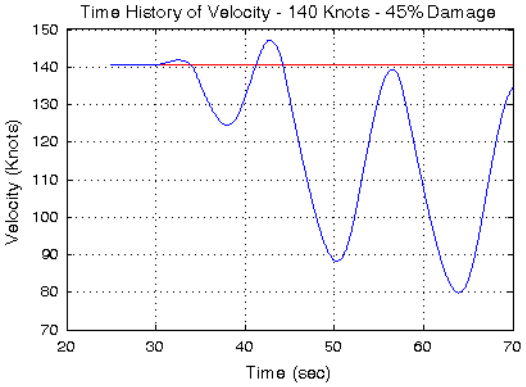
(b)



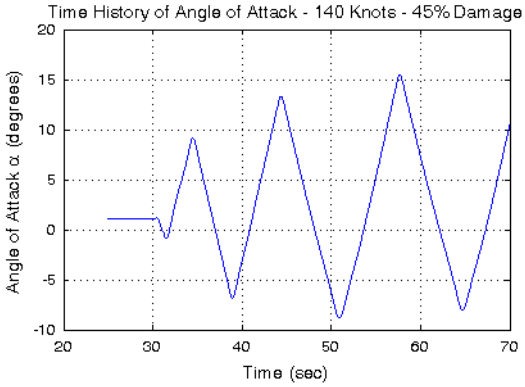
(c)



(d)



(e)



(f)

Figure 5.12: Simulation Results at 45% Damage at 140 Knots

Chapter 6

6 CONCLUSIONS

In this research, the new differential equations of motion were derived for a damaged asymmetric aircraft with the CG offset from the body centered axis system. Asymmetry in lifting surface areas was shown to lead to nonzero roll and yaw coefficients at zero sideslip angles, and the physical limitations of the control surfaces for various failures were discussed. These physical limitations establish the maximum controllable damage, even with a perfect adaptive system or *a priori* knowledge of the damaged aircraft parameters.

A Six Degree-Of-Freedom simulator developed at Wichita State University in the MATLAB/Simulink environment was modified to match the new equations of motion. Force and moment aerodynamic models were also modified to facilitate the change of aircraft coefficients for damage initiation. The Dynamic Inverse Resilient Control was accomplished through the use of Adaptive Bias Correctors for error adaptation and a dynamic aircraft inverse. Outer loop aircraft response compensation was performed with a Model Follower and PID that is decoupled for each aircraft command (Velocity, γ, β, Φ).

A Bonanza F-33C was used as the test aircraft. Damaged aircraft coefficient percentage changes were determined by simple conceptual design techniques. These percentage changes were applied to the F-33C baseline data to provide specific test aircraft values. Calculated values of asymmetric coefficients were applied to the test aircraft since no baseline for a percentage change existed. Mass properties of the F-33C were estimated by a mass buildup that provided the same CG and inertia as the test aircraft. An equal distribution of wing mass provided the basis for the CG shift and inertia change.

6.1 Equations of Motion

The new differential equations of motion describe the behavior of the aircraft about the body centered axis, at which the origin is the CG location before damage occurs. From the final equations, it is seen that this result is advantageous over keeping the standard equations of motion, but observing them about the new CG location. For an analysis about the CG, all the aerodynamic coefficients would have to be determined about a new location. While some shift along the body centered X axis adds no difficulty to this task, a shift off the centerline of the aircraft removes all benefits of symmetry in the simplification of the analysis. For computer models, such as NASA's GTM vortex-lattice code [22], altering the model to calculate for CG locations off the aircraft centerline may be too tedious or unnecessary of a task.

Another benefit of keeping the original axis system is the inclusion of transient effects of a CG shift. For some "instantaneous" loss of lifting surface with some time variant amount of fuel loss or shifting, as

was tested in Burcham et al. [2], continual estimation of the aerodynamic coefficients need not occur. Only the new mass values along with CG shift dimenions must be calculated, greatly simplifying the simulation process. In the event of transient analysis, the current equations of motion do not address the coriolis acceleration of the CG about the body centered axis system.

6.2 DIRC and Compensator

Table 6.1 shows the comparison of the 1-D aileron deflection calculation discussed in Section 3.7 compared to the deflection given by the Simulation and the 3rd Order PID compensator architecture. The aileron becomes saturated at deflections beyond 20 degrees. Any further deflection is not possible.

Table 6.1

Predicted vs. Actual Aileron Deflections

	80 knots		**100 knots**		**120 knots**		**140 knots**	
Damage	Predicted	Actual	Predicted	Actual	Predicted	Actual	Predicted	Actual
20%	11.1	11.2	7.1	7.25	5.0	5.0	3.6	3.7
25%	16.3	16.5	10.4	10.5	7.2	7.3	5.3	5.4
30%	22.6	N/A	14.5	14.5	10.0	10.3	7.4	7.5
35%	31.1	N/A	19.9	20.0	13.8	14.2	10.2	10.4
40%	42.1	N/A	27.0	N/A	18.7	19	13.8	14
45%	58.1	N/A	37.2	N/A	25.8	N/A	19.0	*

*No steady state deflection found. Bank angle has a stable decaying bank angle oscillation around ± 20 deg

The final compensation and DIRC parameters have succesfully controlled simulated wing damage for all cases where the control surfaces have adequate aerodynamic authority to hold the aircraft at zero bank angle. All conditions that failed showed a bank angle that decreases and shows multiple revolutions of the aircraft, which further proves that even with full aileron deflection, no capabilities to recover the aircraft existed.

Within Table 6.1, the differences between the 1-D estimation and simulator aileron deflections comes from the inclusion of relaxed sideslip and velocity requirements. Faster flight velocities allow the control of larger amounts of damage, which has been observed on actual damaged aircraft, such as the Israeli Air Force F-15 that suffered total left wing loss [38]. This is due to the increased dynamic pressure over the control surfaces, which affords them more control. At any flight speed, the lift necessary for level flight remains the same for a set amount of damage, and therefore the same asymmetric rolling moment exists at each flight speed. Thus, at higher flight speeds, the aircraft will have more control surface power available to correct the same asymmetric rolling moment that was uncontrollable at lower speeds.

The performance of the compensator design follows the same trends seen in other robust control systems. For small errors off the nominal aircraft (i.e. small amounts of damage) the response could have a smaller overshoot and faster settling time for some disturbance. However, the compensator gains are set to give a specific 2nd Order response over all ranges of damage and plant error, since it is assumed that the DIRC can reduce the inner loop inverse control to zero error signal. Increasing the smaller damage performance causes any high damage recovery to occur with a large loss of altitude, numerous positive to negative bank angle oscillations, and entrance into nonlinear ranges of angle of attack. Relaxing the compensator standards help to stabilize the aircraft over a larger range of damage, but performance is lost overall.

A solution to this problem would be compensator gain scheduling to achieve the best performance for each percent of lifting surface damage. Such scheduling is time consuming, but possible given that the mass properties and aerodynamic coefficients are known, as they are in this research. Even then, accurate *in-flight* knowledge of the aircraft dynamics and damage type will not be known to effectively schedule any compensator gains. Any estimate of the current plant dynamics with a state observer are far more useful in the inverse control, since reducing the error between the inverse controller and the plant will reduce the magnitude of the learning rate any adaptive network requires.

No mathematical boundedness or convergence of the ABC signal was solved in this research, and will be left for later research efforts. While the correction signal from the ABC does converge on a solution for all controllable conditions, any certification of such a system will require such a proof. While the correction signal does not appear to suffer high-gain issues, it has only been tested on one aircraft. Other aircraft may require a different set of learning rates for each command channel's ABC. The determination of such rates requires the simulation of a large range of failures to find a solution, and is not as dynamic as an on-line training mechanism.

6.3 Further Work / Recommendations

There are numerous avenues for future work on the DIRC. Foremost, since only wing damage has been sufficiently demonstrated, subsequent tests should be to simulate losses of the horizontal tail. This issue was not examined in much depth in this research as changes in pitch stability and flight envelope stall and speed protection have been successfully tested before [29, 30]. However, the validation of the current compensator architecture and the convergence of DIRC on a solution are necessary steps before overall FCS work can be considered.

The final lifting surface to be examined would be the vertical tail. A large portion of general aviation aircraft are single engine and the only kind of aircraft that beginning pilots would be able to fly. Research at WSU has included focus on the certification of such adaptive methods with the FAA [39] and for the

purpose of making flying safer to attract new pilots. Such single engine aircraft lack a distinct yaw control method other than the rudder, and with segments of the vertical tail missing the largest provider of yaw stability is removed. Standard aileron deflections (one side trailing edge up, other side trailing edge down) does produce a yawing moment, but one that is a few magnitudes smaller than the rudder. The remaining wing-body contribution to yawing moment due to sideslip is normally minor. Sideslip and minor heading changes can be accomplished by banking and returning to wings level. Twin engine work is occurring at WSU, and the ability to perform differential thrust is almost a necessary requirement to control the loss of a vertical tail.

Regardless of the location of damage, the ability of the aircraft to perform maneuvers is necessary to provide a safe return to an airport, and not just a safe recovery in flight. Currently, the recovery of the aircraft ends with a return to zero bank angle and sideslip, but the heading angle has rotated. The desired heading change may be decision that the pilot must make, and as such, the next step is to put pilots in the loop via simulation and test the current capabilities to command velocity, flight path, and coordinated turns that require values different than those in trimmed damaged flight. Here the decoupling of the flight controls and the pilot command of higher order flight parameters is beneficial as the pilot no longer needs to be concerned with what mixing of controls must occur to perform the task. Hopefully in the future, this decoupled control system with the recovery capabilities of the Dynamic Inverse Resilient Controller will be able to provide an extended safe flight envelope and encourage the participation in general aviation flying.

REFERENCES

References

[1] National Transportation Safety Board. United Airlines Flight 232. *Aircraft Accident Report NTSB/AAR-90/06*, 1989.

[2] Frank W. Burcham Jr., John Burken, Trindel A. Maine, and John Bull. "Emergency Flight Control Using Only Engine Thrust and Lateral Center-of-Gravity Offset: A First Look". *NASA Technical Memorandum 4798*, 1997.

[3] David Hughes and Michael A. Dornheim. "No Flight Controls". *Aviation Week & Space Technology*, 159(23), December 2003.

[4] National Transportation Safety Board. In-Flight Engine Separation - Japan Airlines, Inc. *Aircraft Accident Report NTSB/AAR-93/06*, 1993.

[5] National Transportation Safety Board. In-Flight Separation of Vertical Stabilizer. *Aircraft Accident Report NTSB/AAR-04/04*, 2004.

[6] National Transportation Safety Board. Braniff Airways, Inc. *Aircraft Accident Report NTSB/AAR-69/03*, 1969.

[7] National Transportation Safety Board. NTSB ID CHI05MA011. *Factual Report - Aviation*, 2007.

[8] National Transportation Safety Board. In-Flight Separation of Right Wing - Flying Boat, Inc. *Aircraft Accident Report NTSB/AAR-07/04*, 2007.

[9] National Transportation Safety Board. NTSB ID ATL06FA076. *Factual Report - Aviation*, 2006.

[10] National Transportation Safety Board. NTSB ID ATL99FA072. *Factual Report - Aviation*, 2001.

[11] National Transportation Safety Board. NTSB ID CHI06FA154. *Factual Report - Aviation*, 2007.

[12] A.J. Calise, S. Lee, and M. Sharma. "Direct Adaptive Reconfigurable Control of a Tailless Fighter Aircraft". *AIAA-98-4108*, 1998.

[13] Nhan T. Nguyen, Maryam Bakhtiari-Nejad, and Yong Huang. "Hybrid Adaptive Flight Control with Bounded Linear Stability Analysis". *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2007.

[14] Byoung S. Kim and Anthony J. Calise. "Nonlinear Flight Control Using Neural Networks". *Journal of Guidance, Control, and Dynamics*, 20(1), January-February 1997.

[15] Rolf T. Rysdyk and Anthony J. Calise. "Adaptive Model Inversion Flight Control for Tilt-Rotor Aircraft". *Journal of Guidance, Control, and Dynamics*, 22(3), May-June 1999.

[16] Yoonghyun Shin, Anthony J. Calise, and Matthew D. Johnson. "Adaptive Control of Advanced Fighter Aircraft in Nonlinear Flight Regimes". *Journal of Guidance, Control, and Dynamics*, 31(5), September-October 2008.

[17] John T. Bosworth and Peggy S. Williams-Hayes. "Flight Test Results from the NF-15B Intelligent Flight Control System (IFCS) Project with Adapation to a Simulated Stabilator Failure". *AIAA 2007 Conference and Exhibit*, May 2007.

[18] Eric N. Johnson, Anthony J. Calise, and Hugo De Blauwe. "In Flight Validation of Adaptive Flight Control Methods". *AIAA Guidance, Navigation and Control Conference and Exhibit*, August 2008.

[19] Frank W. Burcham Jr., Trindel A. Maine, C. Gordon Fullerton, and Lannie Dean Webb. "Development and Flight Evaluation of an Emergency Digital Flight Control System Using Only Engine Thrust on an F-15 Airplane". *NASA Technical Paper 3627*, 1996.

[20] Frank W. Burcham Jr., John Burken, Trindel A. Maine, and Drew Pappas. "Development and Flight Test of an Augmented Thrust-Only Flight Control System on an MD-11 Transport Aircraft". *NASA Technical Memorandum 4745*, 1996.

References (continued)

[21] John Bull, Robert Mah, Gordon Hardy, Barry Sullivan, Jerry Jones, Diane Williams, Paul Soukup, and Jose Winters. "Piloted Simulation Tests of Propulsion Control as Backup to Loss of Primary Flight Controls for a B747-400 Jet Transport". *NASA Technical Memorandum 112191*, 1997.

[22] Nhan Nguyen, Kalmanje Krishnakumar, John Kaneshige, and Pascal Nespeca. "Flight Dynamics and Hybrid Adaptive Control of Damaged Aircraft". *Journal of Guidance, Control, and Dynamics*, 31(3), May-June 2008.

[23] Yunshen Tang, Ella M. Atkins, and Robert M. Sanner. "Emergency Flight Planning for a Generalized Transport Aircraft with Left Wing Damage". *AIAA Guidance, Navigation, and Control Conference and Exhibit*, August 2007.

[24] Moshe Idan, Matthew Johnson, and Anthony J. Calise. "Hierarchical Approach to Adaptive Control for Improved Flight Safety". *Journal of Guidance, Control, and Dynamics*, 25(6), November-December 2002.

[25] N. Duerksen. *"Fuzzy Logic Adaptive Decoupled Flight Control for General Aviation Airplane"*. Doctoral Dissertation, Department of Aerospace Engineering, Wichita State University, Spring 1996.

[26] D.B. Beringer. "Applying Performance-Controlled Systems, Fuzzy Logic, and Fly-by-Wire Controls to General Aviation". *DOT/FAA/AM-02/7*, 2002.

[27] James E. Steck, Kamran Rokhsaz, Urpo Pesonen, Stuart Mochrie, and Mike Maxfield. "Pilot Evaluation of an Adaptive Controller on a General Aviation SATS Testbed Aircraft". *Journal of Guidance, Control, and Dynamics*, 22(3), May-June 1999.

[28] H. Samuel Bruner and James E. Steck. "A Model Following Inverse Controller with Adaptive Compensation for General Aviation Aircraft". *AIAA Guidance, Navigation and Control Conference and Exhibit*, August 2008.

[29] Urpo J. Pesonen, James E. Steck, Kamran Rokhsaz, Hugh Samuel Bruner, and Noel Duerksen. "Adaptive Neural Network Inverse Controller for General Aviation Safety". *Journal of Guidance, Control, and Dynamics*, 27(3), 2004.

[30] K. Rokhsaz, J. Steck, R. Chandramohan, and B. Singh. "Response of an Advanced Flight Control System to Microburst Encounters". *2005-01-03420*, 2003.

[31] Jan Roskam. *Airplane Flight Dynamics and Automatic Flight Controls Part I*. DARcorporation, 2003.

[32] Donald T. Greenwood. *Principles of Dynamics*. Prentice Hall, second edition, 1988.

[33] Bandu N. Pamadi. *Performance, Stability, Dynamics, and Control of Airplanes*. AIAA Education Series, second edition, 2004.

[34] Daniel P. Raymer. *Aircraft Design: A Conceptual Approach*. AIAA Education Series, fourth edition, 2006.

[35] D.E. Hoak. *USAF Stability and Control Datcom*. Global Engineering Documents, California, 1978.

[36] Katsuhiko Ogata. *Modern Control Engineering*. Prentice Hall, fourth edition, 2002.

[37] Ira H. Abbott, Albert E. Von Doenhoff, and Louis S. Stivers Jr. "Summary of Airfoil Data". *NACA Report No. 824*.

[38] James E. Tomayko. *The Story of Self-Repairing Flight Control Systems*. Dryden Historical Study No. 1, 2003.

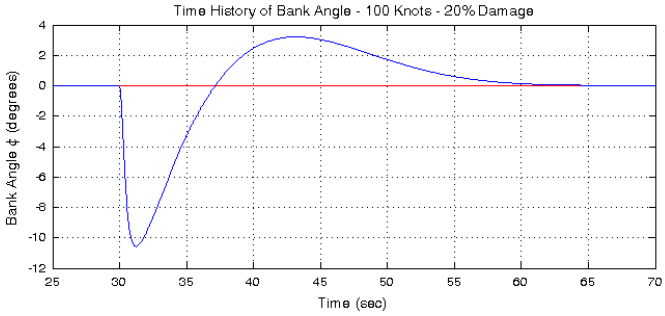
[39] Steck, J. et al. "Simulation and Flight Test Assesment of Safety Benefits and Certification Aspects of Advanced Flight control Systems". *U.S. Department of Transportation Report DOT/FAA/AR-00-C-WSU-00-17*, 2002.

APPENDIX

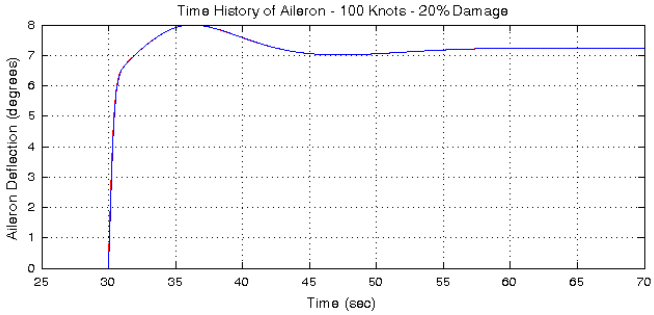
APPENDIX A

Non-Critical Condition Results

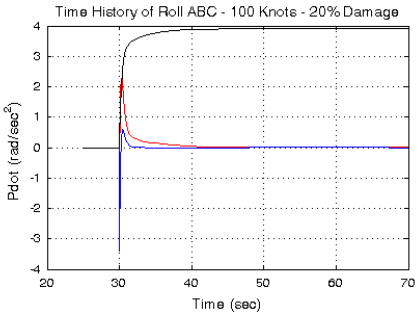
100 Knots - 20% damage



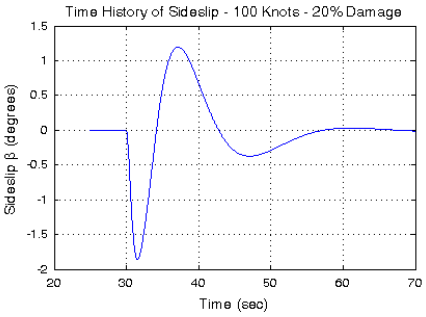
(a)



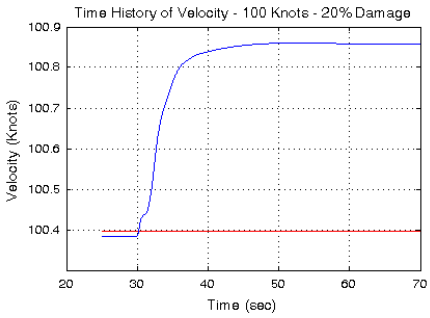
(b)



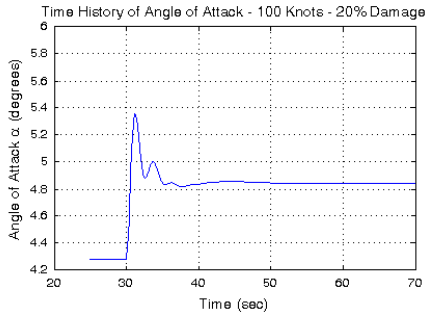
(c)



(d)



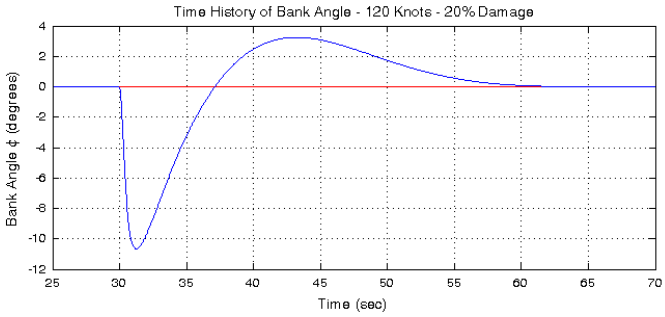
(e)



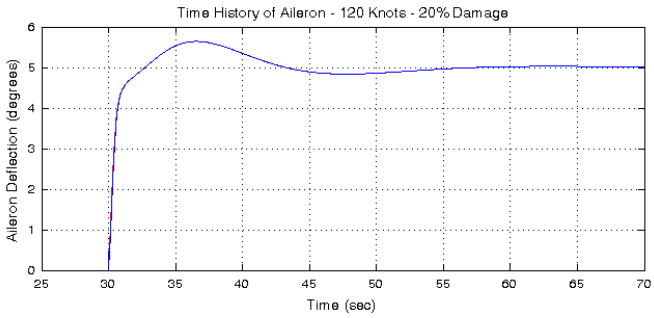
(f)

Figure A.1: Simulation Results at 20% Damage at 100 Knots

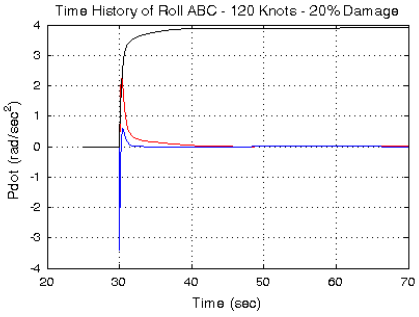
120 Knots - 30-25% damage



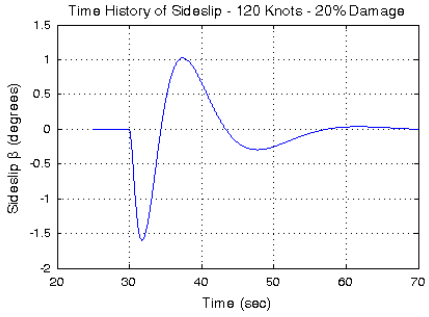
(a)



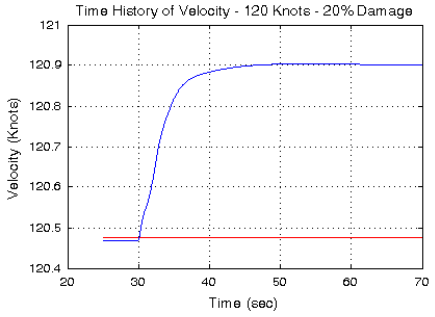
(b)



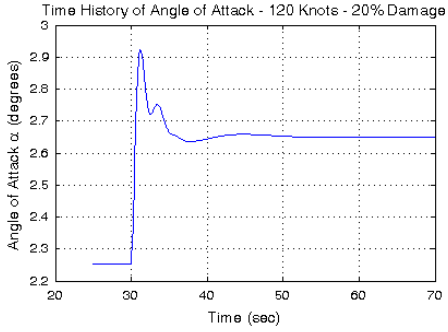
(c)



(d)

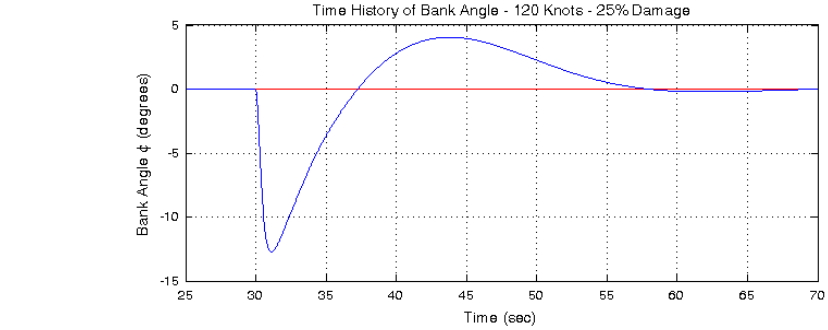


(e)

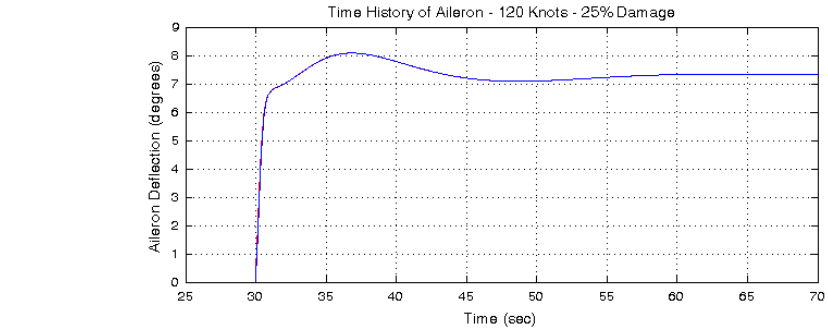


(f)

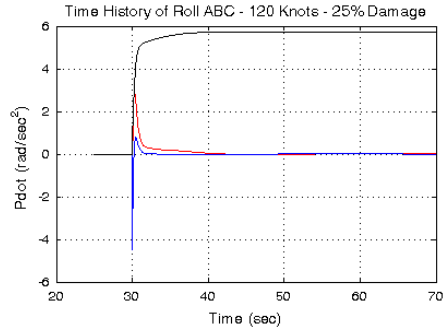
Figure A.2: Simulation Results at 20% Damage at 120 Knots



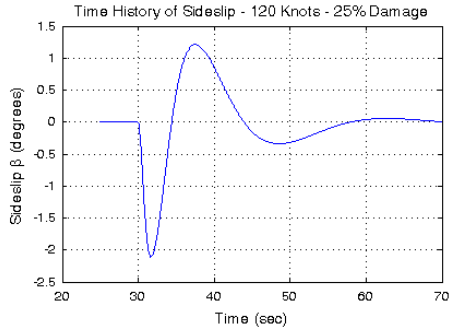
(a)



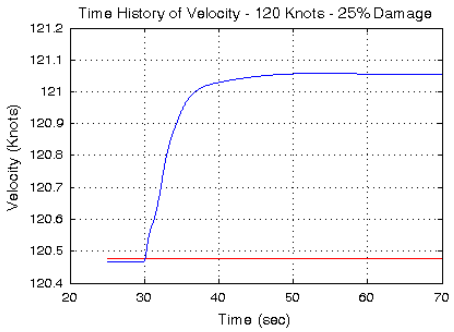
(b)



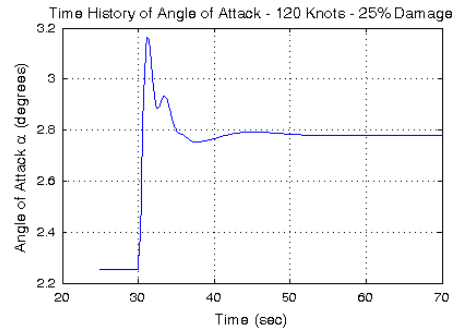
(c)



(d)



(e)



(f)

Figure A.3: Simulation Results at 25% Damage at 120 Knots

140 Knots - 20-30% damage

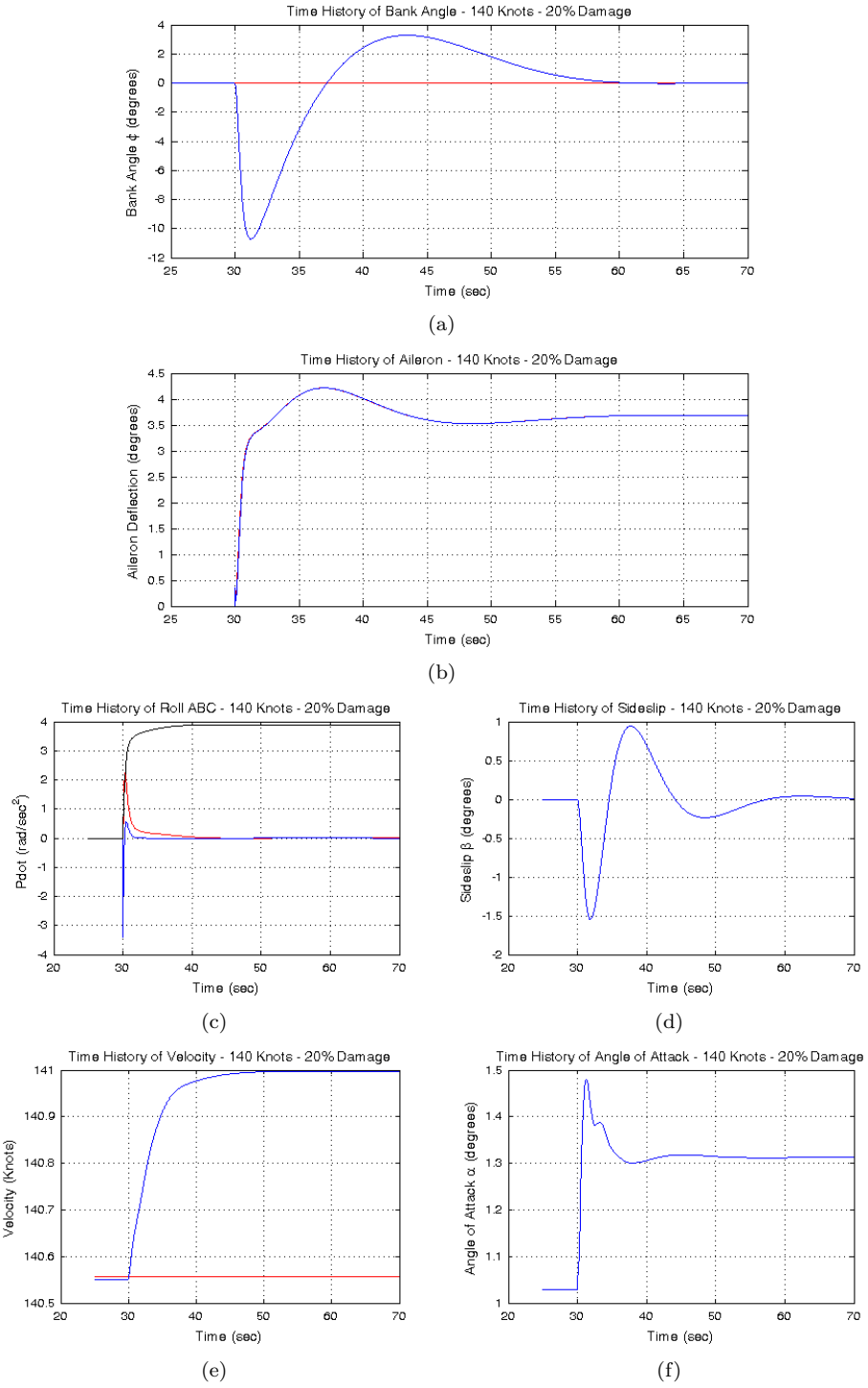


Figure A.4: Simulation Results at 20% Damage at 140 Knots

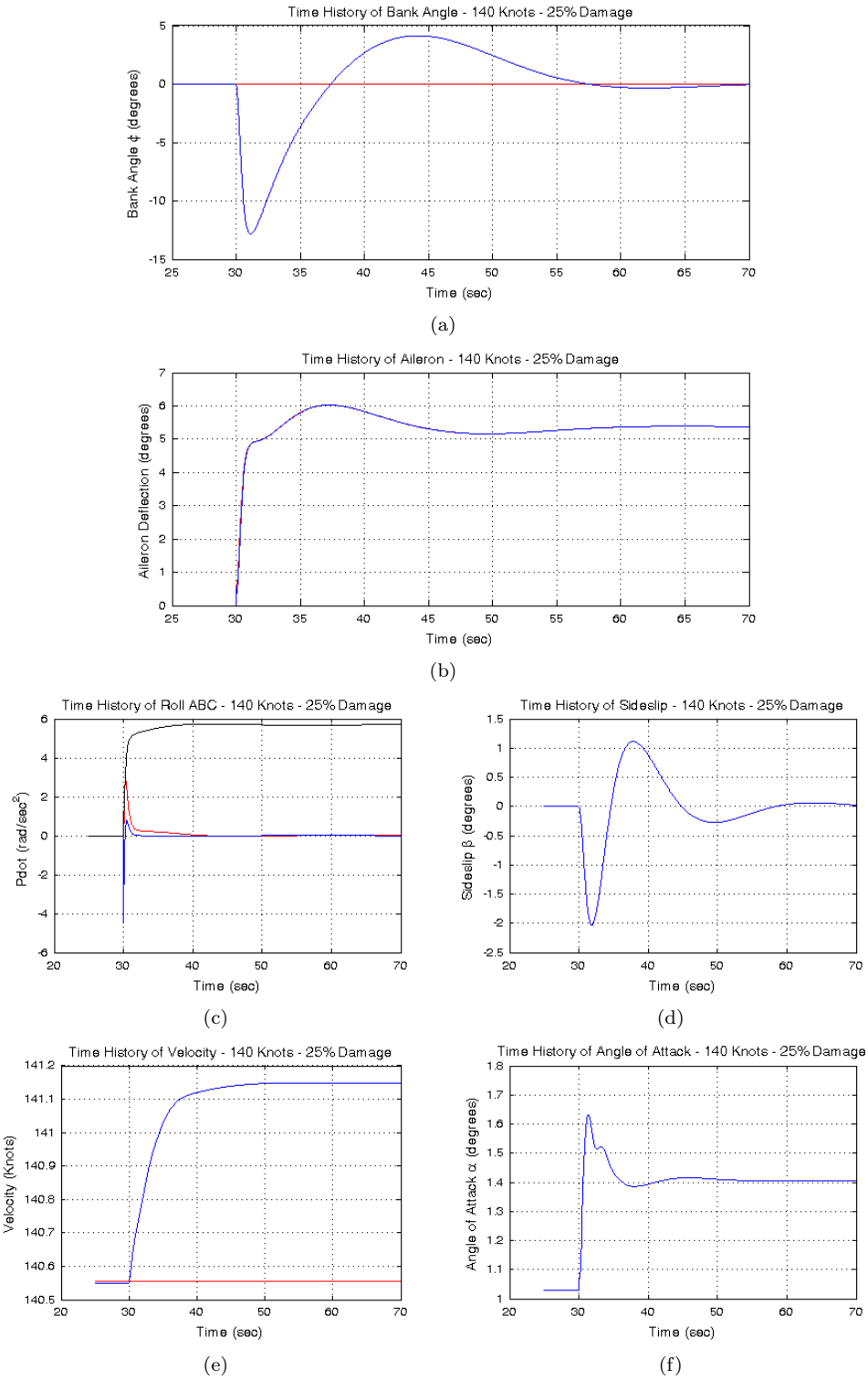


Figure A.5: Simulation Results at 25% Damage at 140 Knots

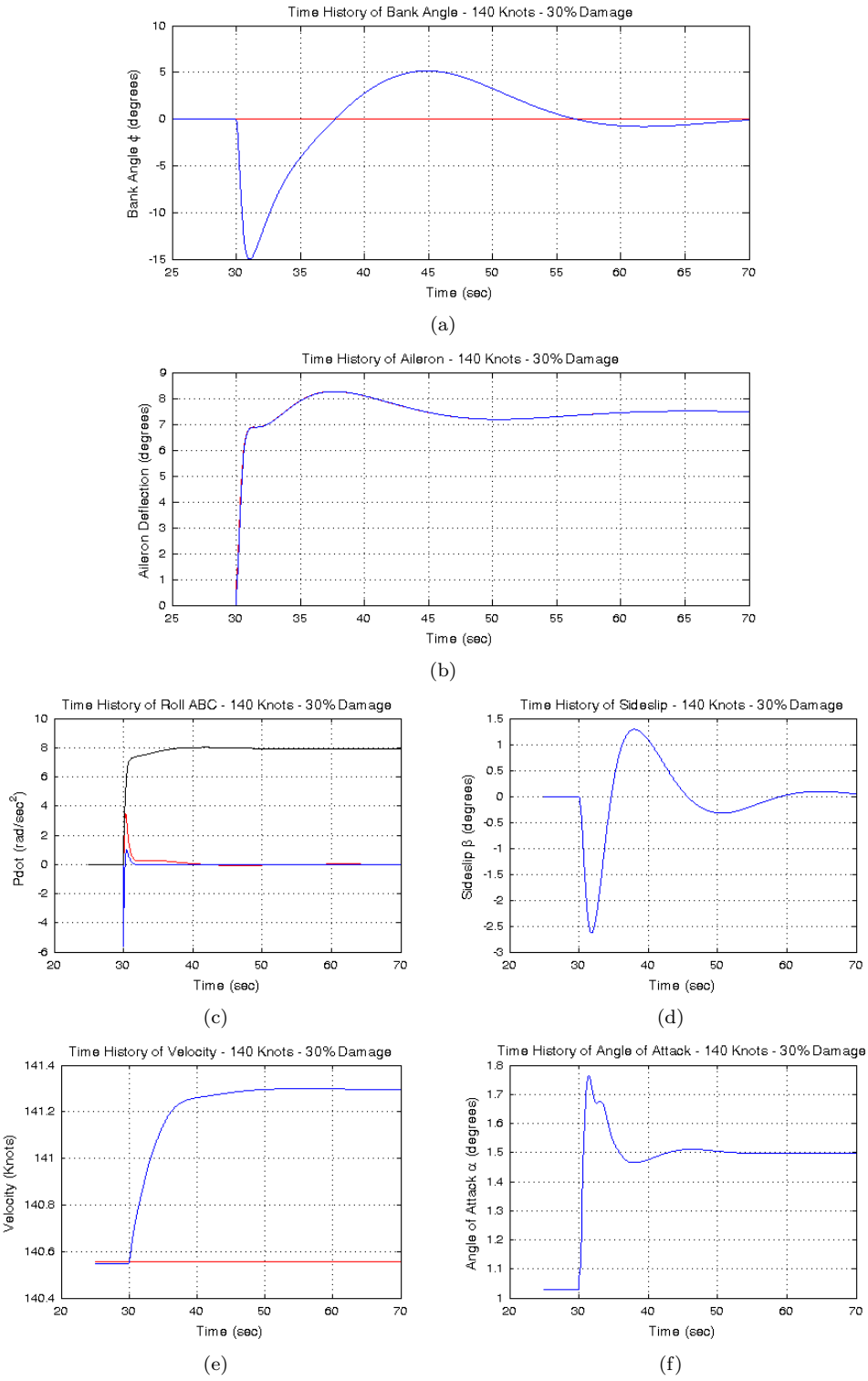


Figure A.6: Simulation Results at 30% Damage at 140 Knots

APPENDIX B

MATLAB SOURCE CODE

- **Variable Loading and Initialization**

```
%========================================================================
% Thesis Simulink Model Master File
% Author: Michael Arruda
% Date: August 26th, 2008
% ThesisMaster.m
%
%========================================================================
% This file does the following:
%   - User schedules when the failure occurs by setting the FAIL_TIME
%       variable to the desired seconds.
%   - User sets the asymmetric aircraft coefficients to the desired value.
%       These damaged asymmetric coefficients are noted with an "a" in
%       front of the typical name for the coefficients, e.g. CLA and aCLA.
%   - User sets the new mass of the aircraft.
%   - User sets the c.g. shift vector in feet.
%       ["x axis shift" "y axis shift" "z axis shift"]
%   - Runs bonalat.m to initialize all standard aircraft parameters before
%       the model can be run.
%   - Loads data from Excel Sheet that defines all the damaged parameters
%   - Runs bonlat1.mdl
%========================================================================
clear all
clc
%========================================================================
global g Sref Ixxb Iyyb Izzb Ixzb cbar mass phiT dT bref
global CD0 CDK CDDE CL1 CLA CL0 CM0 CMA CMQHAT CMADOTHAT CMDE CLDE0
global CR0 CRBETA CRBETA_DOT CRP CRR CRDA CRDATAB CRDR CRDRTAB dRMdT
global CN0 CNBETA CNBETA_DOT CNR CNP CNDA CNDATAB CNDR CNDRTAB dYMdT
global CY0 CYBETA CYBETA_DOT CYP CYR CYDA CYDATAB CYDR CYDRTAB dYdT LATAMATINV
global CLADOTHAT CLQHAT CLDE CLDETAB CMDA
global CDDETAB CMDETAB inertia_matrix

global aSref aIxxb aIyyb aIzzb aIxzb amass acbar abref
global aCL0 aCLA aCLADOTHAT aCLQHAT aCLDE aCLDETAB
global aCDDE aCDDETAB aCDK
global aCY0 aCYBETA aCYBETA_DOT aCYP aCYR aCYDA aCYDR aCYDATAB aCYDRTAB
global aCM0 aCMA aCMQHAT aCMADOTHAT aCMDE aCMDETAB aCMDA
global aCR0 aCRBETA aCRBETA_DOT aCRP aCRR aCRDA aCRDR aCRDATAB aCRDRTAB
global aCN0 aCNBETA aCNBETA_DOT aCNP aCNR aCNDA aCNDR aCNDATAB aCNDRTAB
global FAIL_TIME deltar damage_matrix
%========================================================================
% SEPARATE VARIABLES FOR INVERSE_LINEAR AND SIDE_FORCE
global Sref_inv Ixxb_inv Iyyb_inv Izzb_inv Ixzb_inv cbar_inv mass_inv bref_inv
global CLA_inv CL0_inv CM0_inv CMA_inv CMDE_inv
global CR0_inv CRBETA_inv CRDA_inv CRDR_inv
global CN0_inv CNBETA_inv CNDA_inv CNDR_inv
global CY0_inv CYBETA_inv CYDA_inv CYDR_inv
%========================================================================
% Learning Rates for Adaptive Bias Correctors
global PLEARN1 PLEARN2 QLEARN1 QLEARN2 RLEARN1 RLEARN2 VLEARN1 VLEARN2
PLEARN1 = 0.003;, PLEARN2 = 0.03;
QLEARN1 = 0.2;, QLEARN2 = 0.00;
RLEARN1 = 0.09;, RLEARN2 = 0.00;
VLEARN1 = 0.0;, VLEARN2 = 0.0;
%========================================================================
% Enter Velocity in Knots: Select RUN = 1 to pull new data for a new flight speed
U = i
U1 = U*1.15*5280/3600;
RUN = 1;
% Update coefficient database with flight velocity
```

```
if RUN == 1
    hXls = actxserver('Excel.Application');
    wkbk = hXls.Workbooks;
    file = wkbk.Open('F:\Thesis Backup\Sizings.xls');
    sheets = file.Sheets;
    OutputSheet = sheets.Item('Plots');
    OutputRange = OutputSheet.get('Range','C3','C4');
    OutputRange.value{1} = U1
    hXls.ExecuteExcel4Macro('!Asymmetric_Update()');
    file.Save;
    wkbk.Close;
    hXls.Quit;
end
% Pull undamaged coefficients
bonalat
% Percent damage for one half of the wing. Use 0,10,15,20,25,30,35,40,45,50
damage = ;
% Loads damaged variables from Excel
varloading
% Sets Gamma, Beta, and Bank Angle in DEGREES
GAMMa = 0;
BETa = 0;
BANk = 0;
%=========================================================================
% Damaged Asymmetric Aircraft Parameter & Coefficients
%   General Aircraft
aSref = Sref*(1-Srefc);
abref = bref*(1-brefc);
acbar = cbar*(1-cbarc);
aIxxb = Ixxc;
aIyyb = Iyyc;
aIzzb = Izzc;
aIxzb = Ixzc;
amass = massc;
%   Lift
aCL0 = CL0*(1-CL0c);
aCLA = CLA*(1-CLAc);
aCLADOTHAT = CLADOTHAT;
aCLQHAT = CLQHAT;
aCLDE = CLDE*(1-CLDEc);
aCLDETAB = CLDETAB;
%   Drag
aCDDE = CDDE;
aCDDETAB = CDDETAB;
e = 0.74;
aCDK = 1/(pi*e*abref^2/aSref);
%   Side Force
aCY0 = CY0c
aCYBETA = CYBETA*(1-CYBETAc);
aCYBETA_DOT = CYBETA_DOT;
aCYP = CYP;
aCYR = CYR;
aCYDA = CYDA;
aCYDR = CYDR*(1-CYDRc);
aCYDATAB = CYDATAB;
aCYDRTAB = CYDRTAB;
%   Pitching
aCM0 = CM0*(1-CM0c);
aCMA = CMA*(1-CMAc);
aCMQHAT = CMQHAT;
aCMADOTHAT = CMADOTHAT;
aCMDE = CMDE*(1-CMDEc);
aCMDETAB = CMDETAB;
aCMDA = CMDAc;
CMDA = 0.0;
%   Roll
```

```
aCR0 = CR0c;
aCRBETA = CRBETA*(1-CRBETAc);
aCRBETA_DOT = CRBETA_DOT;
aCRP = CRP;
aCRR = CRR;
aCRDA = CRDA*(1-CRDAc);
aCRDATAB = CRDATAB;
aCRDR = CRDR*(1-CRDRc);
aCRDRTAB = CRDRTAB;
%   Yaw
aCN0 = CN0c;
aCNBETA = CNBETA*(1-CNBETAc);
aCNBETA_DOT = CNBETA_DOT;
aCNP = CNP;
aCNR = CNR;
aCNDA = CNDA*(1-CNDAc);
aCNDR = CNDR*(1-CNDRc);
aCNDATAB = CNDATAB;
aCNDRTAB = CNDRTAB;
% INVERSE_LINEAR & SIDE_FORCE SAVED VARIABLES
Sref_inv = Sref;
Ixxb_inv = Ixxb;
Iyyb_inv = Iyyb;
Izzb_inv = Izzb;
Ixzb_inv = Ixzb;
cbar_inv = cbar;
mass_inv = mass;
bref_inv = bref;
CLA_inv = CLA;
CL0_inv = CL0;
CM0_inv = CM0;
CMA_inv = CMA;
CMDE_inv = CMDE;
CR0_inv = CR0;
CRBETA_inv = CRBETA;
CRDA_inv = CRDA;
CRDR_inv = CRDR;
CN0_inv = CN0;
CNBETA_inv = CNBETA;
CNDA_inv = CNDA;
CNDR_inv = CNDR;
CY0_inv = CY0;
CYBETA_inv = CYBETA;
CYDA_inv = CYDA;
CYDR_inv = CYDR;
%=============================================================================
% Damage Initiation Variables
FAIL_TIME = 30;
deltar = [x y z]
damage_matrix = [Ixxc Ixyc Ixzc; Ixyc Iyyc 0;Ixzc 0 Izzc]
%=============================================================================
sim('bonlat1',70)
%=============================================================================
```

```
%==========================================================================
%   M-File to set up parameters on general6 simulation
%   Filename: bonlat.m
%==========================================================================
global maxelev minelev
global maxthrust minthrust
global maxaileron minaileron
global maxrudder minrudder
global maxaileron minaileron

% Aircraft constants initialized below
%Longitudinal
global g Sref Ixxb Iyyb Izzb Ixzb cbar mass phiT dT bref
global CD0 CDK CDDE CL1 CLA CL0 CM0 CMA CMQHAT CMADOTHAT CMDE CLDE0 CMDA
%Lateral-Directional
global CR0 CRBETA CRBETA_DOT CRP CRR CRDA CRDATAB CRDR CRDRTAB dRMdT
global CN0 CNBETA CNBETA_DOT CNR CNP CNDA CNDATAB CNDR CNDRTAB dYMdT
global CY0 CYBETA CYBETA_DOT CYP CYR CYDA CYDATAB CYDR CYDRTAB dYdT LATAMATINV
%   Lift
global CLADOTHAT CLQHAT CLDE CLDETAB
%   Drag
global CDDETAB

g = 32.2;

% Initialize standard atmosphere
stdatm;
% Geometric Data
Sref = 177.6;
cbar = 5.4;
bref = 32.819;
h_trim=2000;
gammatrim = 0.0; % level flight trim
rho = interp1(hp_data,rho_data,h_trim);
qbar = 0.5*rho*U1*U1;
it = 2.0*pi/180;
cosit = cos(it);
sinit = sin(it);

% Mass Properties

Wt = 3300.0;  % all data for S-35 Bonanza about 25% MAC
mass = Wt/g;
CL1 = Wt/(qbar*Sref); %level flight, L=W
Ixxb = 1377.27;
Iyyb = 1919.01;
Izzb = 3235.0;
Ixzb = 48.48;
inertia_matrix = [Ixxb 0 -Ixzb; 0 Iyyb 0;-Ixzb 0 Izzb];

% Longitudinal Aero Data Force and Moment Model
% Note all the zeros are zeros not Ohs  ie CL0
CL0 = 0.187985;
CLA = 5.2611;
CLADOTHAT = 2.1386;
CLQHAT = 4.5794;
CLDE = .35946;
CLDETAB = CLDE/6.;

CM0 = 0.05;
CMA = -.8053;
CMAMOD = CMA;
TMOD = 1.0;
CMADOTHAT = -5.8222;
CMQHAT = -13.7139;
CMDE = -0.97842;
```

```
CMDEMOD= 1.0;
CLDEMOD= 1.0;
CMDETAB = CMDE/6;
dPMdT = 0;
CMDA = 0.0;
CLDE=CLDEMOD*CLDE;

CD0 = 0.0204;
CDK = 0.071;
CDDE = 0;
CDDETAB = 0;
%  Elevator and throttle limits
maxelev = +15.0;   %deg
minelev = -25.0;   %deg
maxthrust = 500.0;  %lbf
minthrust = 0.0;     %lbf
%  Aileron and Rudder limits
maxaileron = +20.0;   %deg
minaileron = -20.0;   %deg
maxrudder = 25.0;  %lbf
minrudder = -25.0;     %lbf

% Additional parameters required by the inverse control block:

phiT = 0.0;    % Thrust vector angle (positive up)
dT = 0.0;      % Thrust vector distance from center of mass
CDQ = 0.0;
CDADOT = 0.0;

% Lateral Directional Aero Data Force and Moment Model

CR0 = 0.0;
CRBETA = -.0923;
CRBETA_DOT = 0;
CRP = -.484;
CRR = .0798;
CRDA = .229;
CRDATAB = 0;
CRDR = .0147;
CRDRTAB = 0;
dRMdT = 0;

CN0 = 0.0;
CNBETA = .0587;
CNBETA_DOT = 0;
CNR = -.0937;
CNP = -.0278;
CNDA = -.0216;
CNDATAB = 0;
CNDR = -.0645;
CNDRTAB = 0;
dYMdT = 0;

CY0 = 0.0;
CYBETA = -.393;
CYBETA_DOT = 0;
CYP = -.075;
CYR = .214;
CYDA = 0;
CYDATAB = 0;
CYDR = .187;
CYDRTAB = 0;
dYdT = 0;
```

```
%  Precalculate lateral matrix inverse for use in inverse controller
LATAMAT = [ CNBETA  CNDA  CNDR
         CRBETA  CRDA  CRDR
         CYBETA  CYDA  CYDR  ];
LATAMATINV=inv(LATAMAT);

%   Call the M-File that calculates the trim solution.
trim_ac_A;
Vtrue_ic=sqrt(Uic^2+Vic^2+Wic^2);
%%%%%%
%  Properties of the linear controllers and input commands
%  velocity controller
slope=1.0;        % slope in the sigmoid command functions
ttime=10.0;       % time of thrust command
% delta_u=-0.0 % commanded change in airpseed (ft/sec) relative to intital airpspeed of U1
delta_u=1.68;  %conversion factor for velocity from knots to ft/sec
T1_u =3.0;

% flight path angle controller
atime=0.0; % time of first flight path angle command
delta_gamma = 3.14159/180;   %conversion factor for gamma for change from degrees to radians
atime2=0.0;        % time of second flight path angle command

% Gamma PID controller
rise_time_gamma=4.0;
zeta_gamma =1.5;
omegan_gamma = (1.0-0.4167*zeta_gamma+2.917*zeta_gamma^2)/rise_time_gamma;
Kp_gamma = omegan_gamma^2;
Ki_gamma =0.025;
Kd_gamma = 2*zeta_gamma*omegan_gamma;
integral_altitude = 0*0.0000025;  %  gain of integral altitude feedback ontroller

% Beta Angle controller
rise_time_beta = 10.0;
zeta_beta = 0.85;
omegan_beta = (1.0-0.4167*zeta_beta+2.917*zeta_beta^2)/rise_time_beta;
Kd_beta = 1.0;
Ki_beta = omegan_beta^2*Kd_beta;
Kp_beta = 2*(zeta_beta)*omegan_beta*Kd_beta;

% Bank Angle controller
phitime=0.0;   % time of bank angle command
delta_phi= 3.14159/180.;
rise_time_phi= 9;
zeta_phi = 0.925;
%zeta_phi=0.69;
omegan_phi = (1.0-0.4167*zeta_phi+2.917*zeta_phi^2)/rise_time_phi;
a_phi = 1/(rise_time_phi-7);
d_phi = a_phi/10; % a_phi/2; /10
b_phi = 2*zeta_phi*omegan_phi;
c_phi = omegan_phi^2;
Kd_phi = a_phi + b_phi;
Kp_phi = c_phi + a_phi*b_phi;
Ki_phi = a_phi*c_phi;
```

- **Simulation Variable Changing S-Functions**
  - **Var_Change Function**

```
function [sys,x0,str,ts] = Var_Change(t,x,u,flag)
% S-function for workspace variable change.
global g Sref Ixxb Iyyb Izzb Ixzb cbar mass phiT dT bref
global CD0 CDK CDDE CL1 CLA CL0 CM0 CMA CMQHAT CMADOTHAT CMDE CLDE0
global CR0 CRBETA CRBETA_DOT CRP CRR CRDA CRDATAB CRDR CRDRTAB dRMdT
global CN0 CNBETA CNBETA_DOT CNR CNP CNDA CNDATAB CNDR CNDRTAB dYMdT
global CY0 CYBETA CYBETA_DOT CYP CYR CYDA CYDATAB CYDR CYDRTAB dYdT LATAMATINV
global CLADOTHAT CLQHAT CLDE CLDETAB
global CDDETAB CMDETAB inertia_matrix

global aSref aIxxb aIyyb aIzzb aIxzb amass
global aCL0 aCLA aCLADOTHAT aCLQHAT aCLDE aCLDETAB
global aCDDE aCDDETAB
global aCY0 aCYBETA aCYBETA_DOT aCYP aCYR aCYDA aCYDR aCYDATAB aCYDRTAB
global aCM0 aCMA aCMQHAT aCMADOTHAT aCMDE aCMDETAB
global aCR0 aCRBETA aCRBETA_DOT aCRP aCRR aCRDA aCRDR aCRDATAB aCRDRTAB
global aCN0 aCNBETA aCNBETA_DOT aCNP aCNR aCNDA aCNDR aCNDATAB aCNDRTAB
global FAIL_TIME deltar damage_matrix
switch flag,
  %%%%%%%%%%%%%%%%%%
  % Initialization %
  %%%%%%%%%%%%%%%%%%
  % Initialize the states, sample times, and state ordering strings.
  case 0
    [sys,x0,str,ts]=mdlInitializeSizes;

  %%%%%%%%%%
  % Outputs %
  %%%%%%%%%%
  % Return the outputs of the S-function block.
  case 3
    sys=mdlOutputs(t,x,u);

  %%%%%%%%%%%%%%%%%%%
  % Unhandled flags %
  %%%%%%%%%%%%%%%%%%%
  % There are no termination tasks (flag=9) to be handled.
  % Also, there are no continuous or discrete states,
  % so flags 1,2, and 4 are not used, so return an emptyu
  % matrix
  case { 1, 2, 4, 9 }
    sys=[];

  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  % Unexpected flags (error handling)%
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  % Return an error message for unhandled flag values.
  otherwise
    error(['Unhandled flag = ',num2str(flag)]);

end

%
%=============================================================================
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function.
%=============================================================================
%
function [sys,x0,str,ts] = mdlInitializeSizes()

global g Sref Ixxb Iyyb Izzb Ixzb cbar mass phiT dT bref
global CD0 CDK CDDE CL1 CLA CL0 CM0 CMA CMQHAT CMADOTHAT CMDE CLDE0
```

```
global CR0 CRBETA CRBETA_DOT CRP CRR CRDA CRDATAB CRDR CRDRTAB dRMdT
global CN0 CNBETA CNBETA_DOT CNR CNP CNDA CNDATAB CNDR CNDRTAB dYMdT
global CY0 CYBETA CYBETA_DOT CYP CYR CYDA CYDATAB CYDR CYDRTAB dYdT LATAMATINV
global CLADOTHAT CLQHAT CLDE CLDETAB
global CDDETAB CMDETAB inertia_matrix

global aSref aIxxb aIyyb aIzzb aIxzb amass
global aCL0 aCLA aCLADOTHAT aCLQHAT aCLDE aCLDETAB
global aCDDE aCDDETAB
global aCY0 aCYBETA aCYBETA_DOT aCYP aCYR aCYDA aCYDR aCYDATAB aCYDRTAB
global aCM0 aCMA aCMQHAT aCMADOTHAT aCMDE aCMDETAB
global aCR0 aCRBETA aCRBETA_DOT aCRP aCRR aCRDA aCRDR aCRDATAB aCRDRTAB
global aCN0 aCNBETA aCNBETA_DOT aCNP aCNR aCNDA aCNDR aCNDATAB aCNDRTAB
global FAIL_TIME deltar damage_matrix

sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 1;  % dynamically sized with -1
sizes.NumInputs      = 1;  % dynamically sized with -1
sizes.DirFeedthrough = 1;  % has direct feedthrough
sizes.NumSampleTimes = 1;

sys = simsizes(sizes);
str = [];
x0  = [];
ts  = [-1 0];   % inherited sample time

% end mdlInitializeSizes

%
%=============================================================================
% mdlOutputs
% Return the output vector for the S-function
%=============================================================================
%
function [sys] = mdlOutputs(t,x,u)
global g Sref Ixxb Iyyb Izzb Ixzb cbar mass phiT dT bref
global CD0 CDK CDDE CL1 CLA CL0 CM0 CMA CMQHAT CMADOTHAT CMDE CLDE0
global CR0 CRBETA CRBETA_DOT CRP CRR CRDA CRDATAB CRDR CRDRTAB dRMdT
global CN0 CNBETA CNBETA_DOT CNR CNP CNDA CNDATAB CNDR CNDRTAB dYMdT
global CY0 CYBETA CYBETA_DOT CYP CYR CYDA CYDATAB CYDR CYDRTAB dYdT LATAMATINV
global CLADOTHAT CLQHAT CLDE CLDETAB
global CDDETAB CMDETAB inertia_matrix

global aSref aIxxb aIyyb aIzzb aIxzb amass abref acbar
global aCL0 aCLA aCLADOTHAT aCLQHAT aCLDE aCLDETAB
global aCDDE aCDDETAB aCDK
global aCY0 aCYBETA aCYBETA_DOT aCYP aCYR aCYDA aCYDR aCYDATAB aCYDRTAB
global aCM0 aCMA aCMQHAT aCMADOTHAT aCMDE aCMDETAB
global aCR0 aCRBETA aCRBETA_DOT aCRP aCRR aCRDA aCRDR aCRDATAB aCRDRTAB
global aCN0 aCNBETA aCNBETA_DOT aCNP aCNR aCNDA aCNDR aCNDATAB aCNDRTAB
global FAIL_TIME deltar damage_matrix

testtime = u(1);

if testtime == FAIL_TIME

    Sref = aSref;
    bref = abref;
    cbar = acbar;
    Ixxb = aIxxb;
    Iyyb = aIyyb;
    Izzb = aIzzb;
    Ixzb = aIxzb;
    mass = amass;
```

```
    %   Lift
    CL0         = aCL0;
    CLA         = aCLA;
    CLADOTHAT   = aCLADOTHAT;
    CLQHAT      = aCLQHAT;
    CLDE        = aCLDE;
    CLDETAB     = aCLDETAB;
    %   Drag
    CDDE        = aCDDE;
    CDDETAB     = aCDDETAB;
    CDK         = aCDK;
    %   Side Force
    CY0         = aCY0;
    CYBETA      = aCYBETA;
    CYBETA_DOT  = aCYBETA_DOT;
    CYP         = aCYP;
    CYR         = aCYR;
    CYDA        = aCYDA;
    CYDR        = aCYDR;
    CYDATAB     = aCYDATAB;
    CYDRTAB     = aCYDRTAB;
    %   Pitching
    CM0         = aCM0;
    CMA         = aCMA;
    CMQHAT      = aCMQHAT;
    CMADOTHAT   = aCMADOTHAT;
    CMDE        = aCMDE;
    CMDETAB     = aCMDETAB;
    %   Roll
    CR0         = aCR0;
    CRBETA      = aCRBETA;
    CRBETA_DOT  = aCRBETA_DOT;
    CRP         = aCRP;
    CRR         = aCRR;
    CRDA        = aCRDA;
    CRDATAB     = aCRDATAB;
    CRDR        = aCRDR;
    CRDRTAB     = aCRDRTAB;
    %   Yaw
    CN0         = aCN0;
    CNBETA      = aCNBETA;
    CNBETA_DOT  = aCNBETA_DOT;
    CNP         = aCNP;
    CNR         = aCNR;
    CNDA        = aCNDA;
    CNDR        = aCNDR;
    CNDATAB     = aCNDATAB;
    CNDRTAB     = aCNDRTAB;

end

out(1) = 1;
sys = out;

% end mdlOutputs
```

– **Lift_Var Function**

```
function [sys,x0,str,ts] = Lift_Var(t,x,u,flag)

global CL0 CLA CLADOTHAT CLQHAT CLDE CLDETAB Sref
switch flag,
  %%%%%%%%%%%%%%%%%%
  % Initialization %
  %%%%%%%%%%%%%%%%%%
  % Initialize the states, sample times, and state ordering strings.
  case 0
    [sys,x0,str,ts]=mdlInitializeSizes;

  %%%%%%%%%%
  % Outputs %
  %%%%%%%%%%
  % Return the outputs of the S-function block.
  case 3
    sys=mdlOutputs(t,x,u);

  %%%%%%%%%%%%%%%%%%
  % Unhandled flags %
  %%%%%%%%%%%%%%%%%%
  % There are no termination tasks (flag=9) to be handled.
  % Also, there are no continuous or discrete states,
  % so flags 1,2, and 4 are not used, so return an emptyu
  % matrix
  case { 1, 2, 4, 9 }
    sys=[];

  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  % Unexpected flags (error handling)%
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  % Return an error message for unhandled flag values.
  otherwise
    error(['Unhandled flag = ',num2str(flag)]);

end

%
%=============================================================================
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function.
%=============================================================================
%
function [sys,x0,str,ts] = mdlInitializeSizes()

global CL0 CLA CLADOTHAT CLQHAT CLDE CLDETAB Sref

sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 7;  % dynamically sized with -1
sizes.NumInputs      = 1;  % dynamically sized with -1
sizes.DirFeedthrough = 1;  % has direct feedthrough
sizes.NumSampleTimes = 1;

sys = simsizes(sizes);
str = [];
x0  = [];
ts  = [-1 0];   % inherited sample time

% end mdlInitializeSizes
```

```
%
%=============================================================================
% mdlOutputs
% Return the output vector for the S-function
%=============================================================================
%
function [sys] = mdlOutputs(t,x,u)
global CL0 CLA CLADOTHAT CLQHAT CLDE CLDETAB Sref

testtime = u(1);

out(1) = CL0;
out(2) = CLA;
out(3) = CLADOTHAT;
out(4) = CLQHAT;
out(5) = CLDE;
out(6) = CLDETAB;
out(7) = Sref;

sys = out;

% end mdlOutputs
```

– **Pitch_Var Function**

```
function [sys,x0,str,ts] = Drag_Var(t,x,u,flag)

global CL0 CLA CDDE CDDETAB Sref
switch flag,
  %%%%%%%%%%%%%%%%%%
  % Initialization %
  %%%%%%%%%%%%%%%%%%
  % Initialize the states, sample times, and state ordering strings.
  case 0
    [sys,x0,str,ts]=mdlInitializeSizes;

  %%%%%%%%%%
  % Outputs %
  %%%%%%%%%%
  % Return the outputs of the S-function block.
  case 3
    sys=mdlOutputs(t,x,u);

  %%%%%%%%%%%%%%%%%%
  % Unhandled flags %
  %%%%%%%%%%%%%%%%%%
  % There are no termination tasks (flag=9) to be handled.
  % Also, there are no continuous or discrete states,
  % so flags 1,2, and 4 are not used, so return an emptyu
  % matrix
  case { 1, 2, 4, 9 }
    sys=[];

  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  % Unexpected flags (error handling)%
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  % Return an error message for unhandled flag values.
  otherwise
    error(['Unhandled flag = ',num2str(flag)]);

end

%
%=============================================================================
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function.
%=============================================================================
%
function [sys,x0,str,ts] = mdlInitializeSizes()
%saved between calls for internal derivative calculation
global CL0 CLA CDDE CDDETAB Sref
%load('neural_storage.mat');
sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 6;  % dynamically sized with -1
sizes.NumInputs      = 1;  % dynamically sized with -1
sizes.DirFeedthrough = 1;  % has direct feedthrough
sizes.NumSampleTimes = 1;

sys = simsizes(sizes);
str = [];
x0  = [];
ts  = [-1 0];   % inherited sample time

% end mdlInitializeSizes
```

```
%
%==========================================================================
% mdlOutputs
% Return the output vector for the S-function
%==========================================================================
%
function [sys] = mdlOutputs(t,x,u)
global CL0 CLA CDDE CDDETAB Sref CDK

testtime = u(1);

out(1) = CL0;
out(2) = CLA;
out(3) = CDDE;
out(4) = CDDETAB;
out(5) = Sref;
out(6) = CDK;

sys = out;

% end mdlOutputs
```

– **Pitch_Var Function**

```
function [sys,x0,str,ts] = Pitch_Var(t,x,u,flag)

global CM0 CMA CMADOTHAT CMQHAT CMDE CMDETAB CMDA Sref
switch flag,
  %%%%%%%%%%%%%%%%%%
  % Initialization %
  %%%%%%%%%%%%%%%%%%
  % Initialize the states, sample times, and state ordering strings.
  case 0
    [sys,x0,str,ts]=mdlInitializeSizes;

  %%%%%%%%%%
  % Outputs %
  %%%%%%%%%%
  % Return the outputs of the S-function block.
  case 3
    sys=mdlOutputs(t,x,u);

  %%%%%%%%%%%%%%%%%%
  % Unhandled flags %
  %%%%%%%%%%%%%%%%%%
  % There are no termination tasks (flag=9) to be handled.
  % Also, there are no continuous or discrete states,
  % so flags 1,2, and 4 are not used, so return an emptyu
  % matrix
  case { 1, 2, 4, 9 }
    sys=[];

  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  % Unexpected flags (error handling)%
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  % Return an error message for unhandled flag values.
  otherwise
    error(['Unhandled flag = ',num2str(flag)]);

end

%
%=============================================================================
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function.
%=============================================================================
%
function [sys,x0,str,ts] = mdlInitializeSizes()

global CM0 CMA CMADOTHAT CMQHAT CMDE CMDETAB Sref CMDA

sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 8;  % dynamically sized with -1
sizes.NumInputs      = 1;  % dynamically sized with -1
sizes.DirFeedthrough = 1;  % has direct feedthrough
sizes.NumSampleTimes = 1;

sys = simsizes(sizes);
str = [];
x0  = [];
ts  = [-1 0];   % inherited sample time

% end mdlInitializeSizes
```

```
%
%==========================================================================
% mdlOutputs
% Return the output vector for the S-function
%==========================================================================
%
function [sys] = mdlOutputs(t,x,u)
global CM0 CMA CMADOTHAT CMQHAT CMDE CMDETAB Sref CMDA

testtime = u(1);

out(1) = CM0;
out(2) = CMA;
out(3) = CMADOTHAT;
out(4) = CMQHAT;
out(5) = CMDE;
out(6) = CMDETAB;
out(7) = CMDA;
out(8) = Sref;

sys = out;

% end mdlOutputs
```

– **Side_Var Function**

```
function [sys,x0,str,ts] = Side_Var(t,x,u,flag)

global Sref CYDRTAB CYDATAB CYDR CYDA CYR CYP CYBETA_DOT CYBETA CY0
switch flag,
  %%%%%%%%%%%%%%%%%%
  % Initialization %
  %%%%%%%%%%%%%%%%%%
  % Initialize the states, sample times, and state ordering strings.
  case 0
    [sys,x0,str,ts]=mdlInitializeSizes;

  %%%%%%%%%%
  % Outputs %
  %%%%%%%%%%
  % Return the outputs of the S-function block.
  case 3
    sys=mdlOutputs(t,x,u);

  %%%%%%%%%%%%%%%%%%
  % Unhandled flags %
  %%%%%%%%%%%%%%%%%%
  % There are no termination tasks (flag=9) to be handled.
  % Also, there are no continuous or discrete states,
  % so flags 1,2, and 4 are not used, so return an emptyu
  % matrix
  case { 1, 2, 4, 9 }
    sys=[];

  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  % Unexpected flags (error handling)%
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  % Return an error message for unhandled flag values.
  otherwise
    error(['Unhandled flag = ',num2str(flag)]);

end

%
%=============================================================================
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function.
%=============================================================================
%
function [sys,x0,str,ts] = mdlInitializeSizes()

global Sref CYDRTAB CYDATAB CYDR CYDA CYR CYP CYBETA_DOT CYBETA CY0

sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 10;  % dynamically sized with -1
sizes.NumInputs      = 1;  % dynamically sized with -1
sizes.DirFeedthrough = 1;  % has direct feedthrough
sizes.NumSampleTimes = 1;

sys = simsizes(sizes);
str = [];
x0  = [];
ts  = [-1 0];   % inherited sample time

% end mdlInitializeSizes
```

```
%
%==============================================================================
% mdlOutputs
% Return the output vector for the S-function
%==============================================================================
%
function [sys] = mdlOutputs(t,x,u)
global Sref CYDRTAB CYDATAB CYDR CYDA CYR CYP CYBETA_DOT CYBETA CY0

testtime = u(1);

out(1) = Sref;
out(2) = CYDRTAB;
out(3) = CYDATAB;
out(4) = CYDR;
out(5) = CYDA;
out(6) = CYR;
out(7) = CYP;
out(8) = CYBETA_DOT;
out(9) = CYBETA;
out(10) = CY0;

sys = out;

% end mdlOutputs
```

– **Yaw_Var Function**

```
function [sys,x0,str,ts] = Yaw_Var(t,x,u,flag)

global CN0 CNBETA CNBETA_DOT CNP CNR CNDA CNDR CNDATAB CNDRTAB Sref
switch flag,
  %%%%%%%%%%%%%%%%%%
  % Initialization %
  %%%%%%%%%%%%%%%%%%
  % Initialize the states, sample times, and state ordering strings.
  case 0
    [sys,x0,str,ts]=mdlInitializeSizes;

  %%%%%%%%%%
  % Outputs %
  %%%%%%%%%%
  % Return the outputs of the S-function block.
  case 3
    sys=mdlOutputs(t,x,u);

  %%%%%%%%%%%%%%%%%%
  % Unhandled flags %
  %%%%%%%%%%%%%%%%%%
  % There are no termination tasks (flag=9) to be handled.
  % Also, there are no continuous or discrete states,
  % so flags 1,2, and 4 are not used, so return an emptyu
  % matrix
  case { 1, 2, 4, 9 }
    sys=[];

  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  % Unexpected flags (error handling)%
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  % Return an error message for unhandled flag values.
  otherwise
    error(['Unhandled flag = ',num2str(flag)]);

end

%
%=============================================================================
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function.
%=============================================================================
%
function [sys,x0,str,ts] = mdlInitializeSizes()

global CN0 CNBETA CNBETA_DOT CNP CNR CNDA CNDR CNDATAB CNDRTAB Sref

sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 10;  % dynamically sized with -1
sizes.NumInputs      = 1;  % dynamically sized with -1
sizes.DirFeedthrough = 1;  % has direct feedthrough
sizes.NumSampleTimes = 1;

sys = simsizes(sizes);
str = [];
x0  = [];
ts  = [-1 0];   % inherited sample time

% end mdlInitializeSizes
```

```
%
%=============================================================================
% mdlOutputs
% Return the output vector for the S-function
%=============================================================================
%
function [sys] = mdlOutputs(t,x,u)
global CN0 CNBETA CNBETA_DOT CNP CNR CNDA CNDR CNDATAB CNDRTAB Sref
testtime = u(1);

out(1) = CN0;
out(2) = CNBETA;
out(3) = CNBETA_DOT;
out(4) = CNP;
out(5) = CNR;
out(6) = CNDA;
out(7) = CNDR;
out(8) = CNDATAB;
out(9) = CNDRTAB;
out(10)= Sref;

sys = out;

% end mdlOutputs
```

– **Roll_Var Function**

```
function [sys,x0,str,ts] = Roll_Var(t,x,u,flag)

global CR0 CRBETA CRBETA_DOT CRP CRR CRDA CRDR CRDATAB CRDRTAB Sref
switch flag,
  %%%%%%%%%%%%%%%%%%
  % Initialization %
  %%%%%%%%%%%%%%%%%%
  % Initialize the states, sample times, and state ordering strings.
  case 0
    [sys,x0,str,ts]=mdlInitializeSizes;

  %%%%%%%%%%
  % Outputs %
  %%%%%%%%%%
  % Return the outputs of the S-function block.
  case 3
    sys=mdlOutputs(t,x,u);

  %%%%%%%%%%%%%%%%%%
  % Unhandled flags %
  %%%%%%%%%%%%%%%%%%
  % There are no termination tasks (flag=9) to be handled.
  % Also, there are no continuous or discrete states,
  % so flags 1,2, and 4 are not used, so return an emptyu
  % matrix
  case { 1, 2, 4, 9 }
    sys=[];

  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  % Unexpected flags (error handling)%
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  % Return an error message for unhandled flag values.
  otherwise
    error(['Unhandled flag = ',num2str(flag)]);

end

%
%=============================================================================
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function.
%=============================================================================
%
function [sys,x0,str,ts] = mdlInitializeSizes()

global CR0 CRBETA CRBETA_DOT CRP CRR CRDA CRDR CRDATAB CRDRTAB Sref

sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 10;  % dynamically sized with -1
sizes.NumInputs      = 1;  % dynamically sized with -1
sizes.DirFeedthrough = 1;  % has direct feedthrough
sizes.NumSampleTimes = 1;

sys = simsizes(sizes);
str = [];
x0  = [];
ts  = [-1 0];   % inherited sample time

% end mdlInitializeSizes
```

```
%
%=============================================================================
% mdlOutputs
% Return the output vector for the S-function
%=============================================================================
%
function [sys] = mdlOutputs(t,x,u)
global CR0 CRBETA CRBETA_DOT CRP CRR CRDA CRDR CRDATAB CRDRTAB Sref

testtime = u(1);

out(1) = CR0;
out(2) = CRBETA;
out(3) = CRBETA_DOT;
out(4) = CRP;
out(5) = CRR;
out(6) = CRDA;
out(7) = CRDR;
out(8) = CRDATAB;
out(9) = CRDRTAB;
out(10)= Sref;

sys = out;

% end mdlOutputs
```