

37999424

NONLINEAR SIX DEGREE OF FREEDOM SIMULATION
OF A TWIN JET ENGINE TRANSPORT AIRCRAFT

A Thesis Presented to

The Faculty of the

Fritz J. and Dolores H. Russ
College of Engineering and Technology

Ohio University

In Partial Fulfillment

of the Requirement for the Degree

Master of Science

by

Jason G. Wozniak

August, 1997

Thesis
M
1997
WOZN

OHIO UNIVERSITY
LIBRARIES

Acknowledgements

I would like to thank the people who helped make this project and thesis possible. I would like to thank Dr. Douglas Lawrence for his support during my studies at Ohio University. I am grateful to Dr. Frank Van Graas for his encouragement and for introducing me to the Joint University Program for Air Transportation Research, through which this project was sponsored. Special thanks go to Dick Hueschen at NASA Langley for his feedback and support of this project.

Finally, I would like to thank my parents, Tom and Nancy, as well as my siblings, Wendy, Chad, and Heather, for their love and support throughout my life.

Table of Contents

Acknowledgements	i
Table of Contents	ii
List of Figures	iv
List of Tables	v
Chapter 1: INTRODUCTION	1
1.1 Motivation	1
1.2 SIMULINK Implementation	2
1.3 Thesis Organization	4
Chapter 2: AIRFRAME RIGID BODY DYNAMICS	5
2.1 Aircraft Coordinate Systems	5
2.2 Earth Centered Coordinate Systems	9
2.3 Aircraft Equations Of Motion	11
Chapter 3: OVERVIEW OF AIRCRAFT	15
3.1 Interfacing Between SIMULINK and the Aircraft Dynamics Model	15
3.2 Program Flow of the Aircraft Dynamics Simulation	17
3.3 Aircraft Dynamics Model States, Inputs, and Outputs	18
3.4 Description of Top Level Subroutines	22
3.4.1 Subroutine AC_A	22
3.4.2 Subroutine MISCEL	23
3.4.3 Subroutine INERTIA	24
3.4.4 Subroutine ACCEL	26
3.4.5 Subroutine AC_B	26
3.4.6 Subroutine DEFLEC	27
3.4.7 Subroutine ENGFM	27
3.4.8 Subroutine LNGFM	28
3.4.9 Subroutine AEROFM	28
3.4.10 Subroutine SUMFM	29
3.4.11 Subroutine DERIV	30
3.5 Algebraic Loop in the Aircraft Simulation	32
3.5.1 Description of the Algebraic Loop	32
3.5.2 Alpha-Dot and Beta-Dot Effects on the Aircraft Dynamics	34

Chapter 4: OVERVIEW OF SIMULINK BLOCK DIAGRAMS	39
4.1 Elevator Actuator Block Diagram Description	40
4.2 Stabilitor Actuator Block Diagram Description	41
4.3 Aileron Actuator Block Diagram Description	42
4.4 Rudder Actuator Block Diagram Description	43
4.5 Flap Actuator Block Diagram Description	43
4.6 Throttle Actuator Block Diagram Description	44
4.7 Wind Model Description	45
Chapter 5: SIMULATION RESULTS	57
5.1 Autoland Flight Path	57
5.2 Time Histories of the Aircraft States	59
Chapter 6: CONCLUSIONS	74
References	76
Appendix A: RUNNING THE AIRCRAFT SIMULATION	77
A.1 System Requirements	77
A.2 List of Files and Directory Structure	77
A.3 Executing the Simulation	79

Abstract

List of Figures

Figure 1.1 Block Diagram of Six Degree Of Freedom Aircraft Simulation	3
Figure 2.1 Relationship Between the Body, Stability, and Wind Axes	8
Figure 3.1 Aircraft Dynamics Simulation High-Level Subroutine Listing	18
Figure 3.2 Lift Coefficient Due to Alpha-Dot at 100 ft Altitude	36
Figure 3.3 Pitching Moment Coefficient Due to Alpha-Dot at 100 ft Altitude	37
Figure 3.4 Yawing Moment Coefficient Due to Beta-Dot at 100 ft Altitude	38
Figure 4.1 Elevator Actuator Model Block Diagram	47
Figure 4.2 Modified Lag Sub-System for the Elevator Model	48
Figure 4.3 Stabilizer Actuator Model Block Diagram	49
Figure 4.4 Aileron Actuator Model Block Diagram	50
Figure 4.5 Fast Dynamics Sub-System for the Aileron Block Diagram	51
Figure 4.6 Rudder Actuator Model	52
Figure 4.7 Modified Lag Sub-System for the Rudder Actuator Model	53
Figure 4.8 Flap Actuator Model Block Diagram	54
Figure 4.9 Throttle Actuator Model Block Diagram	55
Figure 4.10 Wind Model Block Diagram	56
Figure 5.1 Time History of Longitude	60
Figure 5.2 Time History of Latitude	61
Figure 5.3 Time History of Altitude	62
Figure 5.4 Time History of North Velocity	63
Figure 5.5 Time History of East Velocity	64
Figure 5.6 Time History of Down Velocity	65
Figure 5.7 Time History of Yaw Rate (Body Axes)	66
Figure 5.8 Time History of Roll Rate (Body Axes)	67
Figure 5.9 Time History of Pitch Rate (Body Axes)	68
Figure 5.10 Time History of Heading	69
Figure 5.11 Time History of Pitch	70
Figure 5.12 Time History of Roll	71
Figure 5.13 Time History of Engine Pressure Ratio	72
Figure 5.14 Time History of Weight	73

List of Tables

Table 3.1 Definition of the Vector Returned by Subroutine SIZES	16
Table 3.2 State Variables of the Aircraft Dynamics Model	20
Table 3.3 Input Variables of the Aircraft Dynamics Model	21
Table 3.4 Output Variables of the Aircraft Dynamics Model (States and Feedback Variables Excluded)	22
Table 4.1 Inputs to the Elevator Actuator Model	40
Table 4.2 Inputs to the Stabilitor Actuator Model	42
Table 4.3 Inputs to the Aileron Actuator Model	43
Table 4.4 Inputs to the Rudder Actuator Model	44
Table 4.5 Outputs of the Wind Model	45

Chapter 1: INTRODUCTION

1.1 Motivation

Due to the stringent requirements placed on current autoland systems for commercial aircraft, simulation has become an essential tool in the certification of these autoland systems. Current requirements established by the Federal Aviation Administration (FAA) and Joint Aviation Administrations (JAA) mandate that the probability of an undetected failure which could cause a catastrophic event must be less than 10^{-9} per approach and the probability of an undetected failure which could cause a hazardous but not catastrophic event must be less than 10^{-7} per approach (Miller, 1996). Clearly, using flight testing to evaluate a system to these specifications would take a large amount of time and money. Therefore, aircraft manufacturers use software simulations validated against flight test data to demonstrate compliance with the requirements. These software simulations emulate the performance of any signal used as guidance, the airborne sensor, the autoland system, the aircraft, and the physical environment. These simulations are run in monte-carlo iterations where the probability of certain types of failures and environmental variables are assumed. The overall performance of the system can then be evaluated to the required statistical confidence based on the simulation results. During these simulations, it is assumed that the weather is always bad, that is, winds are set at the standard models used by the FAA or JAA. In this way,

a conservative measure of the system's operational performance is obtained.

An additional advantage of utilizing software simulations to evaluate an aircraft system's performance is the ability to evaluate the system under extreme conditions such as wind shear or loss of guidance. It would be unsafe to test under these conditions operationally, but the simulation should provide an indication of how the system would perform under these adverse conditions. Appropriate design changes may then be evaluated to increase the performance of the system in these rare cases without endangering a flight test crew.

The nonlinear simulation that is detailed in this thesis was provided by the National Aeronautics and Space Administration (NASA). This simulation was written completely in FORTRAN and was undocumented. The purpose of this work was to implement the simulation in the MATLAB SIMULINK environment and document the code so that it will be more flexible and more easily modified. A primary advantage of the SIMULINK environment is that a variety of tools are available to aid control system design and evaluate system performance. These tools enable a user to trim the model about some operating point, find linearized models of systems, and conveniently display results or store them in variables for further analysis in MATLAB.

1.2 SIMULINK Implementation

The implementation of the aircraft dynamics model in SIMULINK was designed to be modular. Each actuator was implemented as a block diagram

with the same functionality as the code. The wind model was also implemented as a block diagram to enable the user to easily adjust the wind parameters. The aircraft dynamics model has been implemented as a FORTRAN mex-file. The figure below shows a high-level block diagram of the inputs and outputs of the components of the model.

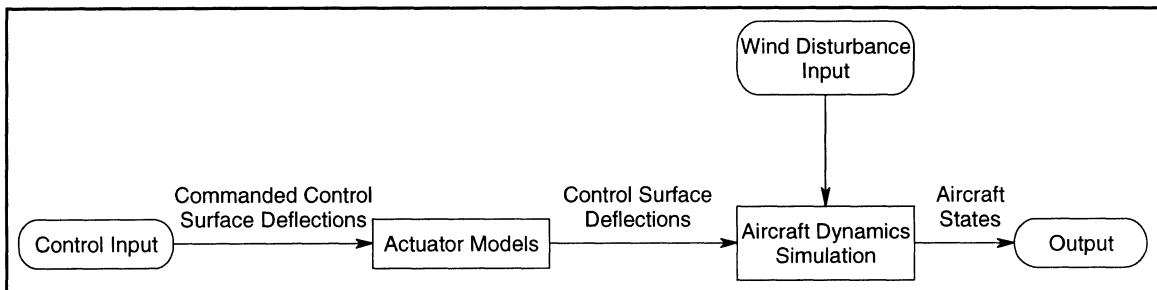


Figure 1.1 Block Diagram of Six Degree Of Freedom Aircraft Simulation

The purpose of the aircraft SIMULINK model is to serve as a tool for the development and evaluation of autoland control systems and guidance signals. The model takes as inputs commanded control surface deflections and returns the aircraft states as outputs. This structure provides the user with the option of implementing a guidance sensor, such as the Global Positioning System (GPS), or assume an ideal position sensor. A controller may then be designed that would use the guidance signals to produce the appropriate control surface deflections so that the desired en-route or autoland performance goals are met.

1.3 Thesis Organization

This thesis is intended to serve as a user manual for the implementation of the twin-jet engine aircraft model. Chapter 2 lists software requirements and provides instructions for executing the model. Chapter 3 describes the FORTRAN mex-file that simulates the aircraft dynamics. The FORTRAN code is not presented as an appendix in this document because it has limited distribution controlled by NASA. The actuator models, implemented as SIMULINK block diagrams, are described in Chapter 4. Results validating the model performance against the NASA FORTRAN baseline are presented in Chapter 5. Conclusions and suggestions for future work are presented in Chapter 6.

Chapter 2: AIRFRAME RIGID BODY DYNAMICS

This chapter presents an introduction to airframe rigid body dynamics.

The aircraft coordinate systems used in the aircraft simulation are presented in Section 2.1. Earth centered coordinate systems are presented in Section 2.2.

The aircraft equations of motion are described in Section 2.3.

2.1 Aircraft Coordinate Systems

This section will present the coordinate systems used in the aircraft dynamics simulation. These are coordinate systems that are used in the analysis of airframes and are employed throughout the simulation.

There are four right-handed coordinate systems that are commonly used to characterize the forces and moments applied to the airframe and the resulting aircraft motion. The two primary coordinate systems used are the North-East-Down, or NED, axes and the body axes. The NED axes are defined with the origin at the aircraft center of gravity, x-axis positive north, y-axis positive east, and z-axis positive down. For purposes of this discussion, "down" represents the direction from the aircraft center of gravity downward along the direction of the local gravity vector. In this simulation, the Earth is considered spherical with uniform mass distribution. Therefore, the local gravity vector always points to the center of the earth. The body axes are defined with the center at the aircraft's center of gravity, the x-axis positive through the nose

of the aircraft, the y-axis positive through the starboard wing, and the z-axis positive through the bottom of the aircraft. The plane on which the body x-axis and body z-axis lie is called the plane of symmetry.

The relationship between these two coordinate systems define important parameters. The plane formed by the North and East axes of the NED coordinate system is the horizontal plane. The angle between the body x-axis and the horizontal plane is called the elevation angle, θ , and its sense is positive nose up. The angle between the body y-axis and the horizontal plane is called the bank angle, ϕ , and its sense is positive starboard wing down. The angle between the projection of the body x-axis on the horizontal plane and the north axis of the NED coordinate system is called the azimuth angle, ψ , which ranges from 0 to 360° with 0 being magnetic north and increasing from north to east.

The NED axes are a useful frame for calculating the aircraft motion with respect to the earth and for developing wind velocity profiles. The body axes are useful as an aircraft-fixed reference system for thrust and forces acting upon the control surfaces. An orthonormal transformation from the NED coordinate system to the body coordinate system, often called the Euler Matrix, is

$$E = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix} \quad (2.1)$$

so that

$$\mathbf{p}_{BODY} = E \mathbf{p}_{NED} \quad (2.2)$$

where \mathbf{p}_{frame} is a vector in the respective coordinate frame.

The final two coordinate systems of interest are defined by the stability axes and the wind axes. These coordinate systems define the relationship between the velocity of the aircraft with respect to the air mass, \mathbf{V} (feet/second), and the body axes. Both of these coordinate systems have the same origin as the body axes, but they are not body fixed since their orientation varies as the velocity vector of the aircraft changes.

The stability axes rotate the body axes about the body y-axis to align the body x-axis with the projection of the aircraft's velocity vector in the plane of symmetry. The angle between the body x-axis and the projection of \mathbf{V} in the plane of symmetry is called the angle-of-attack, denoted α . The relationship between the stability axes and body axes is then

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{STAB} = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{BODY} \quad (2.3)$$

The wind axes are formed by rotating the stability axes about the stability z-

axis to align the stability x-axis with the aircraft's velocity vector, \mathbf{V} . The angle between the stability x-axis and \mathbf{V} is called the sideslip angle, β . The transformation between the stability and wind axes is

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{WIND} = \begin{bmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{STAB} . \quad (2.4)$$

Figure 2.1 shows a graphical depiction of the relationship between the body, stability, and wind axes.

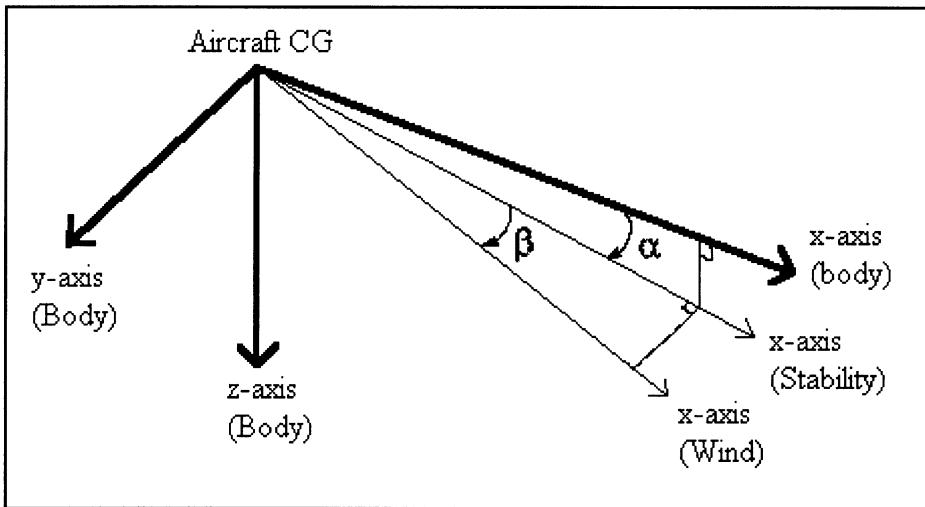


Figure 2.1 Relationship Between the Body, Stability, and Wind Axes

The stability and wind axes define coordinate systems that are useful for the calculation of lift, drag, and side force coefficients as well as the calculation of pitching, rolling, and yawing moments. These coordinate

systems are suitable for these calculations because the forces are normal to the airflow across the airframe. Additionally, the coefficients used in the calculation of the forces and moments are derived from wind tunnel experiments and flight tests and are expressed in either the stability or wind axes since these tests define parameters with respect to the aircraft's atmospheric velocity, \mathbf{V} .

2.2 Earth Centered Coordinate Systems

The final coordinate systems of interest are those that are Earth centered. The development of these coordinate systems begins

A necessary reference frame in the analysis of dynamics is an inertial reference frame. The ideal inertial reference frame is fixed or in uniform rectilinear motion with respect to distant stars. A nearly perfect inertial reference frame is centered at the Sun with the axes fixed with respect to distant stars. This reference frame is troublesome though, because the Earth's orbit around the Sun is present in the equations of motion despite having negligible effect on the aircraft.

One reasonable, though imperfect, inertial reference frame is the Earth Centered Inertial (ECI) frame (McRuer, 1973). In the ECI reference frame, the origin of the coordinate axes is at the center of the earth, the z-axis points towards the north pole, and the x-axis and y-axis are aligned along the equator. The ECI axes translate with the Earth, but are fixed with respect to

distant stars. Another important coordinate system is the Earth Centered Earth Fixed (ECEF) frame. The ECEF frame has the origin at the center of the earth, the z-axis points to the north pole, the x-axis points to the intersection of the equator and the prime meridian, and the y-axis completes the right handed coordinate system. Therefore, the ECEF coordinate system rotates with the earth so that a point on the earth's surface retains the same position in this coordinate system as the earth rotates.

Applying Newton's second law, the relationship between the acceleration of a point mass in the ECI and ECEF frame is

$$\mathbf{a}_{ECEF} = \mathbf{a}_{ECI} - 2\omega \times \mathbf{v}_{ECI} - \omega \times (\omega \times \mathbf{p}_{ECI}), \quad (2.5)$$

where \mathbf{a}_{ECEF} is the acceleration in the ECEF coordinate frame, \mathbf{a}_{ECI} is the acceleration in the ECI coordinate frame, \mathbf{p}_{ECI} is the position of the point mass in the ECI coordinate frame, and ω is the angular velocity of the ECEF coordinate system with respect to the ECI coordinate system. The term $2\omega \times \mathbf{v}_{ECI}$ is called the Coriolis acceleration and the term $\omega \times (\omega \times \mathbf{p}_{ECI})$ is called the centripetal acceleration. For the aircraft simulation, the centripetal acceleration can be neglected since its effect on the gravitational vector is on the order of 1/100 g variance from the north pole to the equator at sea level (Etkin, 1972). The Coriolis acceleration can be significant if the flight path is long or the velocity of the aircraft is high. Coriolis acceleration is taken into account in the state derivative calculations of the aircraft dynamics simulation.

Finally, the familiar latitude, longitude, height (LLH) coordinate system is also used in the simulation. This coordinate system is useful for around the earth navigation because an object at a constant altitude will rotate around the earth as it changes latitude and/or longitude.

2.3 Aircraft Equations Of Motion

The equations of motion of an aircraft may be separated into four categories: navigation equations, force equations, moment equations, and kinematic equations. Each category of equations will be described in this section. The reference frames and state derivatives used to develop the following equations are the same as those in the aircraft dynamics simulation.

The navigation equations solve for the aircraft's position state derivatives. Let the aircraft velocity and aircraft position be components of the state vector. In this case, the aircraft velocity vector is in NED coordinates and the aircraft's position is in LLH coordinates. The navigation equations of motion are then

$$\dot{\mathbf{p}}_{LLH} = \begin{bmatrix} \frac{1}{R_e} * \frac{180}{\pi} & 0 & 0 \\ 0 & \frac{1}{R_e} * \cos(LAT) & 0 \\ 0 & 0 & -1 \end{bmatrix} * \mathbf{V}_{NED} \quad (2.6)$$

where \mathbf{p}_{LLH} is the aircraft's position in LLH coordinates, \mathbf{V}_{NED} is the aircraft's

velocity in NED coordinates, R_e is the radius of the earth, and LAT is the latitude of the aircraft.

The force equations relate the derivative of the aircraft velocity to the forces and moments on the aircraft. Let \mathbf{F}_{NED} represent the sum of all forces on the aircraft, ω_{NED} represent the angular velocity of the aircraft in NED coordinates, and \mathbf{V}_{NED} represent the aircraft's velocity in NED coordinates. Then the force equations in vector form are

$$\dot{\mathbf{V}}_{NED} = \frac{\mathbf{F}_{NED}}{MASS} - \omega_{NED} \times \mathbf{V}_{NED} . \quad (2.7)$$

The moment equations relate the angular acceleration of the aircraft to the aircraft's inertia and the torques on the aircraft. Let the angular velocity of the aircraft be a component of the state vector. First, the moment of inertia of the aircraft must be defined. Let I_{xx} , I_{yy} , and I_{zz} be the moments of inertia about each of the body axes and I_{xz} be the cross-product of inertia. Since the aircraft described in this thesis has a plane of symmetry, the inertia matrix of the aircraft and its inverse are, respectively, (Stevens, 1992)

$$\mathbf{J} = \begin{bmatrix} I_{xx} & 0 & -I_{xz} \\ 0 & I_{yy} & 0 \\ -I_{xz} & 0 & I_{zz} \end{bmatrix} \quad (2.8)$$

and

$$J^{-1} = \frac{1}{\Gamma} \begin{bmatrix} I_{zz} & 0 & I_{xz} \\ 0 & \frac{\Gamma}{I_{yy}} & 0 \\ I_{xz} & 0 & I_{xx} \end{bmatrix} \quad (2.9)$$

where

$$\Gamma = I_{xx}I_{zz} - I_{xz}^2 \quad (2.10)$$

Notice that the plane of symmetry eliminates many of the cross-product of inertia terms from the aircraft's inertia matrix. Let ω_{BODY} be the angular velocity of the aircraft in body axis coordinates. Then the moment equations in matrix form are

$$\dot{\omega}_{BODY} = -J^{-1}(\omega_{BODY} \times (J\omega_{BODY})) + J^{-1}T_{BODY}. \quad (2.11)$$

The kinematic equations of motion relate the derivatives of the aircraft Euler angles to the aircraft angular rates. Let the aircraft Euler angles be the final components of the aircraft's state vector. The vector of aircraft attitudes is composed of the aircraft attitudes defined in Section 2.1,

$$\Phi = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}. \quad (2.12)$$

The kinematic equations in matrix form are

$$\dot{\Phi} = B(\Phi) \omega_{BODY} \quad (2.13)$$

where

$$B(\Phi) = \begin{bmatrix} 1 & \tan(\theta)\sin(\phi) & \tan(\theta)\cos(\phi) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \frac{\sin(\phi)}{\cos(\theta)} & \frac{\cos(\phi)}{\cos(\theta)} \end{bmatrix}. \quad (2.14)$$

The aircraft simulation uses these equations of motion to propagate the aircraft's state. The calculation of the forces and moments on the aircraft and the specific implementation of these equations of motion is discussed in Chapter 3.

Chapter 3: OVERVIEW OF AIRCRAFT MEX FILE

The aircraft dynamic model is described in this chapter. The subroutines that interface between SIMULINK and the aircraft dynamics mex file are presented in Section 3.1. A top-level diagram of the mex file is given in Section 3.2. Section 3.3 presents the states, inputs, and outputs for the model. Section 3.4 contains a description of each top-level subroutine and the calculations that produce the state derivatives. Section 3.5 details an algebraic loop contained in the aircraft simulation.

3.1 Interfacing Between SIMULINK and the Aircraft Dynamics Model

The file 'aircraft.f' contains the subroutines that interface between the FORTRAN mex file implementation of the aircraft dynamics and SIMULINK. The subroutines contained in this file supply the appropriate information to the SIMULINK environment while executing the model. There are four subroutines of interest in this file, SIZES, INITCOND, DERIVS, and OUTPUT. A description of the functionality of each subroutine follows.

Subroutine SIZES returns a vector containing the information listed in Table 3.1. This information allows SIMULINK to set up the proper vector sizes for the input and output of the aircraft dynamics model. Note that the simulation has 14 states, 24 outputs, and 21 inputs.

Subroutine INITCOND returns the initial condition of the states.

Simulink calls this routine at the start of the simulation to initialize the states. The user may change the values in this subroutine to execute flights with different initial trim conditions.

Vector	Description	Value Returned
1	number of continuous states	14
2	number of discrete states	0
3	number of outputs	25
4	number of inputs	21
5	number of discontinuous roots in the system	0
6	direct feedthrough flag	0

Table 3.1 Definition of the Vector Returned by Subroutine SIZES

The subroutines DERIVS and OUTPUT serve as subroutines to implement the commonly seen form for nonlinear state equations. Subroutine DERIVS implements the state derivative equation

$$\dot{x} = f(x, u) \quad (3.1)$$

and subroutine OUTPUT implements the output mapping

$$y = h(x, u), \quad (3.2)$$

where **x** is the state vector, **u** is the input vector, and **y** is the output vector.

Notice that the aircraft simulation is implemented as a time-invariant nonlinear model.

SIMULINK takes care of passing the parameters and integrating the states through its own architecture. Calls are made to these subroutines by SIMULINK to facilitate the execution of the model. The subroutine DERIVS calls the subroutine AC_DYN, which is the top subroutine of the aircraft dynamics simulation. The subroutine AC_DYN in the file 'main_twin_jet.f' implements the dynamics of the aircraft by processing the current state and input vectors as input and returning the state derivatives to subroutine DERIVS.

3.2 Program Flow of the Aircraft Dynamics Simulation

The aircraft dynamics simulation is arranged into subroutines that perform different functions. A listing of the high-level subroutines is shown in Figure 3.1.

These high-level routines were arranged to calculate parameters associated with the equations of motion that yield the state derivatives. These subroutines are described in detail in Section 3.4.

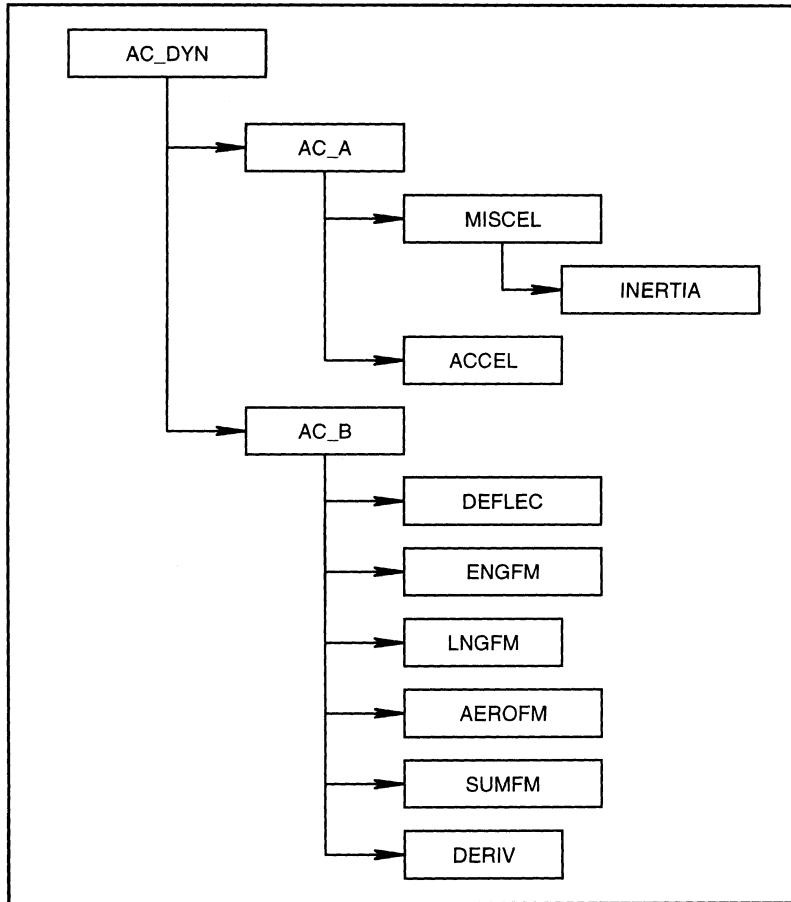


Figure 3.1 Aircraft Dynamics Simulation High-Level Subroutine Listing

3.3 Aircraft Dynamics Model States, Inputs, and Outputs

The state vector of the aircraft consists of the parameters defined in Table 3.2 at the end of this section. The first 12 states characterize the position, velocity, orientation, and rotation of the airframe. The engine model state is EPRARR, the engine pressure ratio of the engines. Finally, the change in the aircraft weight due to fuel consumption is reflected in WEIGHT.

The input vector of the aircraft dynamics model consists of the parameters defined in Table 3.3 at the end of this section. The first nine inputs characterize the positions of the control surfaces and the landing gear. The following six inputs define the wind velocity and acceleration; these inputs serve as disturbance or exogenous inputs to the aircraft dynamics model. The final six inputs are present due to an algebraic loop used to calculate the first derivative of the angle of attack and sideslip. This algebraic loop is discussed in Section 3.5.

The first 14 outputs are the states, which have been described above. The following five outputs are used by the actuator models in the calculation of the control surface deflections. These five are the only outputs that have not been described above, so only these outputs are detailed in Table 3.4 at the end of this section. The variables associated with the algebraic loop are the last six outputs.

State Variable	Description
LAT	latitude (degrees)
LON	longitude (degrees)
ALT	altitude (feet)
V_n	velocity in the North direction (feet/second)
V_e	velocity in the East direction (feet/second)
V_d	velocity in the down direction (feet/second)
P_B	rate of roll (positive right wing down, radians/sec)
Q_B	rate of pitch (positive nose up, radians/second)
R_B	rate of yaw (positive nose right, radians/second)
ϕ	bank angle (positive right wing down, radians)
θ	elevation angle (positive nose up, radians)
ψ	azimuth angle (with respect to magnetic north, radians)
EPRARR	engine pressure ratio
WEIGHT	weight of the aircraft (pounds)

Table 3.2 State Variables of the Aircraft Dynamics Model

Input Variable	Description
FLAPS	flap position (degrees)
THROTLER	throttle position (pilot units)
GEAR	logical; 0-gear up, 1-gear down
DELE	elevator deflection (positive nose up, degrees)
SP	stabilizer position (positive nose up, degrees)
DELA	aileron deflection (positive right wing down, degrees)
SPL	left spoiler deflection (degrees)
SPR	right spoiler deflection (degrees)
DELR	rudder deflection (positive nose right, degrees)
VW _N , VW _E , VW _D	wind velocities, NED frame (feet/second)
AW _N , AW _E , AW _D	wind accelerations, NED frame (feet/second ²)
U _B DOT	acceleration in the body x-axis (feet/second ²)
V _B DOT	acceleration in the body y-axis (feet/second ²)
W _B DOT	acceleration in the body z-axis (feet/second ²)
TOT _{XB}	sum of forces along the body x-axis (slug-foot/second)
TOT _{YB}	sum of forces along the body y-axis (slug-foot/second)
TOT _{ZB}	sum of forces along the body z-axis (slug-foot/second)

Table 3.3 Input Variables of the Aircraft Dynamics Model

Output Variable	Description
MACH	mach number
EAS	indicated airspeed (knots)
CAS	calibrated airspeed (knots)
QBAR	dynamic pressure (pounds/square inch)
CG	center of gravity of the aircraft (feet from nominal along body x-axis)

Table 3.4 Output Variables of the Aircraft Dynamics Model
(States and Feedback Variables Excluded)

3.4 Description of Top Level Subroutines

This section will describe the functions performed by each of the top-level subroutines in the aircraft dynamics simulation. The flow chart of the program was presented in Figure 3.1. The significant parameters that are derived in these subroutines are detailed in this section.

3.4.1 Subroutine AC_A

This subroutine does not perform calculations, but it calls the subroutines MISCEL and ACCEL to compute the equations of motion.

3.4.2 Subroutine MISCEL

This subroutine computes the equations of motion. This includes the calculation of the air density and speed of sound from lookup tables in subroutine ATMAT62. The inertia of the aircraft is computed in the subroutine INERTIA, which is described in the next subsection. The angle of attack, α , and the sideslip, β , are calculated from the body rate velocities as shown in the following equations:

$$\alpha = \tan^{-1} \left(\frac{W_B}{U_B} \right) \quad (3.3)$$

and

$$\beta = \tan^{-1} \left(\frac{V_B \cos \alpha}{U_B} \right), \quad (3.4)$$

where U_B , V_B , and W_B are the body rate velocities along the x, y, and z axes respectively. Also, the first derivatives of α and β are calculated. This calculation involves an algebraic loop that is described in detail in Section 3.7.

The aircraft velocity is also translated into mach number based on the atmospheric parameters calculated above. The instantaneous rates of the aircraft due to the translation of the aircraft over the earth's surface are calculated as follows:

$$\begin{bmatrix} P_L \\ Q_L \\ R_{L,NED} \end{bmatrix} = \frac{1}{R_e + ALT} * \begin{bmatrix} V_E \\ -V_N \\ -V_E * \tan(LAT) \end{bmatrix}, \quad (3.5)$$

the parameter R_e is the radius of the Earth. The variables P , Q , and R refer to angular velocities about the x , y , and z axis of a given coordinate system, respectively. These instantaneous rates are then transformed into the body axes by

$$\begin{bmatrix} P_L \\ Q_L \\ R_{L,BODY} \end{bmatrix} = E \begin{bmatrix} P_L \\ Q_L \\ R_{L,NED} \end{bmatrix}, \quad (3.6)$$

where E is the Euler matrix. The rates calculated in Equation 3.6 are subtracted from the angular velocity states P_B , Q_B , and R_B to form the relative angular velocities:

$$\begin{bmatrix} P_{REL} \\ Q_{REL} \\ R_{REL} \end{bmatrix} = \begin{bmatrix} P_B \\ Q_B \\ R_B \end{bmatrix} - \begin{bmatrix} P_L \\ Q_L \\ R_{L,BODY} \end{bmatrix}. \quad (3.7)$$

These angular velocities are then used in DERIV to calculate the derivatives of the Euler angles.

3.4.3 Subroutine INERTIA

This subroutine returns the inertia of the airframe in body axis coordinates. The return values are the moments of inertia about the body axes, I_{BODY} , and coefficients used in the state derivative equations. The inertias

are computed by a one-dimensional table lookup as a function of the weight of the aircraft. The inertias are then used to calculate coefficients that are used in the computation of the state derivative of angular velocity. The coefficient calculations are

$$\Gamma = I_{XX}I_{YY} - I_{XZ}^2 , \quad (3.8)$$

$$CF_1 = \frac{1}{\Gamma} \left\{ (I_{YY} - I_{ZZ})I_{ZZ} - I_{XZ}^2 \right\} , \quad (3.9)$$

$$CF_2 = \frac{1}{\Gamma} \left\{ (I_{XX} - I_{YY} + I_{ZZ}) * I_{XZ} \right\} , \quad (3.10)$$

$$CF_3 = \frac{I_{ZZ}}{\Gamma} , \quad (3.11)$$

$$CF_4 = \frac{I_{XZ}}{\Gamma} , \quad (3.12)$$

$$CF_5 = \frac{I_{ZZ} - I_{XX}}{I_{YY}} , \quad (3.13)$$

$$CF_6 = \frac{I_{XZ}}{I_{YY}} , \quad (3.14)$$

$$CF_7 = \frac{1}{I_{YY}} , \quad (3.15)$$

$$CF_8 = \frac{1}{\Gamma} \left\{ (I_{xx} - I_{yy}) * I_{xx} + I_{xz}^2 \right\}, \quad (3.16)$$

$$CF_9 = \frac{I_{xx}}{\Gamma}, \quad (3.17)$$

and

$$CF_{10} = \Gamma * I_{xx}. \quad (3.18)$$

3.4.4 Subroutine ACCEL

This subroutine computes the aircraft acceleration at the aircraft's center of gravity, or CG, in body axes coordinates. The inputs consist of the sum of forces and the angular accelerations of the body frame.

3.4.5 Subroutine AC_B

This subroutine, like AC_A, does not perform many calculations. The purpose of this subroutine is to call other routines that compute the forces and moments. This subroutine does, however, compute the derivative of the aircraft weight as follows

$$\frac{d}{dt} WEIGHT = -FUELCON * (FUELF(1) + FUELF(2)). \quad (3.19)$$

The parameter FUELCON is a constant and FUELF is a two-element vector that contains the fuel usage of each engine on the aircraft.

3.4.6 Subroutine DEFLEC

This subroutine is the point at which the control surface deflections are interfaced into the code. Recall that the states and inputs to the aircraft dynamics model were passed into the subroutine AC_DYN by the interfacing code between simulink and the model. The subroutine AC_DYN then passes the control surface deflections to DEFLEC, where they are stored for use by other subroutines.

3.4.7 Subroutine ENGFM

This subroutine calculates the thrust of each engine (THRSTR and THRSTL) and fuel consumption (FUELF) of each engine. In the current operating mode, the two engines are treated as if they were the same. This is because the throttle is constrained to provide the same command to each engine. The single state of the engine dynamics is the engine pressure ratio (EPR).

The thrust is used to calculate the forces and moments on the airframe due to the engine. The forces are calculated as follows

$$\begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}_{BODY} = \begin{bmatrix} THRSTR + THRSTL \\ 0 \\ 0 \end{bmatrix}. \quad (3.20)$$

The moment calculations is

$$\begin{bmatrix} T_L \\ T_M \\ T_N \end{bmatrix}_{BODY} = \begin{bmatrix} D_Y * (-THRSTR + THRSTL) * \frac{\Pi}{180} \\ T_{X_{BODY}} * D_Z \\ D_Y * (-THRSTR + THRSTL) \end{bmatrix}, \quad (3.21)$$

where D_Y and D_Z are the offset of the starboard engine from the CG in body coordinates.

3.4.8 Subroutine LNGFM

This subroutine calculates the forces and moments associated with the landing gear during and after touchdown. The forces and moments are returned in body axis coordinates.

3.4.9 Subroutine AEROFM

The aerodynamic forces and moments are calculated in this subroutine. Aerodynamic coefficients are calculated by calling subroutines AEROC1 and AEROC2 which contain lookup tables that are functions of the aircraft altitude, velocity, orientation with respect to the air mass, and the positions of the control surfaces.

The coefficients for drag (C_D), side force (C_Y), and lift (C_L) are then calculated. The rolling (C_{LS}), pitching (C_M), and yawing moments (C_N) are also computed. The forces and moments are first computed in the stability axis and then transformed into body axis coordinates. The force computations are

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}_{STAB} = \bar{Q}_s * \begin{bmatrix} -C_D \\ C_Y \\ -C_L \end{bmatrix} \quad (3.22)$$

and

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}_{BODY} = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}_{STAB}, \quad (3.23)$$

where F_x , F_y , and F_z are the forces along the respective axis and \bar{Q}_s is dynamic pressure times the wing area. The moment computations are

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix}_{STAB} = \begin{bmatrix} \bar{Q}_{SB} * C_{LS} \\ \bar{Q}_{SC} * C_M \\ \bar{Q}_{SB} * C_N \end{bmatrix} \quad (3.24)$$

and

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix}_{BODY} = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix} \begin{bmatrix} L \\ M \\ N \end{bmatrix}_{STAB}, \quad (3.25)$$

where L , M , and N are the moments about the x , y , and z axis respectively. Also, \bar{Q}_{SB} is \bar{Q}_s times the wing span and \bar{Q}_{SC} is \bar{Q}_s times the wing chord.

3.4.10 Subroutine SUMFM

This subroutine calculates the sum of the forces and moments acting on the airframe. The equation for the sum of forces is

$$\begin{bmatrix} TOT_X \\ TOT_Y \\ TOT_Z \end{bmatrix}_{BODY} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{BODY} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix}_{BODY} + \begin{bmatrix} LG_X \\ LG_Y \\ LG_Z \end{bmatrix}_{BODY}. \quad (3.26)$$

The sum of moments equation is

$$\begin{bmatrix} TOT_L \\ TOT_M \\ TOT_N \end{bmatrix}_{BODY} = \begin{bmatrix} L \\ M \\ N \end{bmatrix}_{BODY} + \begin{bmatrix} T_L \\ T_M \\ T_N \end{bmatrix}_{BODY} + \begin{bmatrix} LG_L \\ LG_M \\ LG_N \end{bmatrix}_{BODY}. \quad (3.27)$$

The forces and moments produced by the engines are reflected in the **T** vectors and the landing gear forces and moments are reflected in the **LG** vectors. The sum of forces is then transformed into NED coordinates as follows

$$\begin{bmatrix} F_N \\ F_E \\ F_D \end{bmatrix}_{NED} = E' \begin{bmatrix} TOT_X \\ TOT_Y \\ TOT_Z \end{bmatrix}_{BODY}, \quad (3.28)$$

where **E** is the Euler matrix and ' is the transpose operator. The forces are transformed into the NED coordinate system because the velocity states are also in the NED coordinate system.

3.4.11 Subroutine DERIV

This subroutine calculates the state derivatives. These equations are arranged into the same categories as the derivations in Section 2.3. The navigation equations are

$$\frac{d}{dt} LAT = V_N * \left(\frac{1}{R_e} \right) * \left(\frac{180}{\Pi} \right), \quad (3.29)$$

$$\frac{d}{dt} LON = \left(V_{ee} * \frac{1}{R_e} * \cos(LAT) \right) * \left(\frac{180}{\Pi} \right), \quad (3.30)$$

and

$$\dot{H} = -V_D. \quad (3.31)$$

The force equations are

$$\dot{V}_N = \frac{F_N}{M_{AC}} + \frac{V_N * V_D - V_E^2 * \tan(LAT)}{R_e + ALT}, \quad (3.32)$$

$$\dot{V}_E = \frac{F_E}{M_{AC}} + \frac{V_E * V_D + V_N * V_E * \tan(LAT)}{R_e + ALT}, \quad (3.33)$$

and

$$\dot{V}_D = \frac{F_D + F_{GRAV}}{M_{AC}} + \frac{-(V_N^2 + V_E^2)}{R_e + ALT}. \quad (3.34)$$

The moment equations are

$$\dot{P}_B = (CF_1 * R_B + CF_2 * P_B) * Q_B + CF_3 * TOT_{LB} + CF_4 * TOT_{NB}, \quad (3.35)$$

$$\dot{Q}_B = (CF_5 * R_B * P_B + CF_6 * (R_B^2 - P_B^2)) + CF_7 * TOT_{MB}, \quad (3.36)$$

and

$$\dot{R}_B = (CF_8 * P_B + CF_9 * R_B) * Q_B + CF_4 * TOT_{LB} + CF_{10} * TOT_{NB} . \quad (3.37)$$

The kinematic equations are

$$\Psi = \frac{Q_{REL} * \sin(\phi) + R_{REL} * \cos(\phi)}{\cos(\theta)} , \quad (3.38)$$

$$\dot{\Theta} = Q_{REL} * \cos(\phi) - R_{REL} * \sin(\phi) , \quad (3.39)$$

and

$$\dot{\phi} = P_{REL} + \Psi \sin(\theta) . \quad (3.40)$$

3.5 Algebraic Loop in the Aircraft Simulation

3.5.1 Description of the Algebraic Loop

There are two algebraic loops in the aircraft simulation due to the calculation of the first derivatives of α and β in the subroutine MISCEL. The equations used to calculate the first derivatives of α and β are formed by differentiating Equation 3.3 and Equation 3.4 to yield

$$\dot{\alpha} = \frac{U_B \dot{W}_B - W_B \dot{U}_B}{U_B^2 + W_B^2} \quad (3.41)$$

and

$$\dot{\beta} = \frac{\dot{V}_B(U_B^2 + W_B^2) - V_B(U_B \dot{U}_B + W_B \dot{W}_B)}{V_t^2(U_B^2 + W_B^2)}, \quad (3.42)$$

where V_t is the total velocity of the aircraft with respect to the air mass. The first derivatives of α and β are then used to calculate aerodynamic coefficients.

Notice that the first derivatives of α and β are functions of the first derivatives of the velocity of the aircraft with respect to the air mass in body axes coordinates. These are calculated later in subroutine ACCEL, therefore they are fed back between iterations and form an algebraic loop. The equation used to calculate the first derivatives of the body axes velocities is

$$\begin{pmatrix} \dot{U}_B \\ \dot{V}_B \\ \dot{W}_B \end{pmatrix} = \frac{1}{MASS} \begin{pmatrix} TOT_x \\ TOT_y \\ TOT_z \end{pmatrix}_{BODY} + \begin{pmatrix} G_x \\ G_y \\ G_z \end{pmatrix}_{BODY} + \begin{pmatrix} R_x \\ R_y \\ R_z \end{pmatrix}_{BODY}, \quad (3.43)$$

where the G_{BODY} vector is the acceleration due to gravity in body axes coordinates and the R_{BODY} vector is the acceleration due to the angular rates in body axes coordinates. TOT_{BODY} is the sum of forces in body axis coordinates which is calculated in the subroutine SUMFM called later in the execution of the code. Hence, there is a second algebraic loop that is formed.

These algebraic loops were preserved in the SIMULINK implementation by adding the terms that are fed back to the output and input vectors of the aircraft dynamics simulation. SIMULINK then has a delay block that will

break the algebraic loop and feedback the terms with a delay of one sample time. The low-level aircraft dynamics block and these delay feedbacks used to break the algebraic loops have been grouped together so that these extra inputs and outputs are not seen from the top-level block diagram.

3.5.2 Alpha-Dot and Beta-Dot Effects on the Aircraft Dynamics

Since the first derivative of angle-of-attack and sideslip are calculated by implementing algebraic loops in the aircraft dynamics simulation, it is useful to see how these terms are applied in the model. The rate of change of α and β are used to account for rate-dependent aerodynamic forces. These forces arise due to the fact that it takes the airflow time to pass from the front of the aircraft to the tail of the aircraft, creating time-dependent forces and moments.

In the aircraft dynamics model, these time-dependent forces and moments are modeled as terms in the static lift, total lift, and pitching moment coefficients. First, dimensionless alpha-dot and beta-dot are calculated in subroutine MISCEL where

$$\dot{\alpha}_h = \dot{\alpha} * \frac{\bar{C}}{2V_t} \quad (3.44)$$

$$\dot{\beta}_h = \dot{\beta} * \frac{B}{2V_t} . \quad (3.45)$$

The parameter \bar{C} is the mean aerodynamic chord and the parameter B is the wingspan. Then, these dimensionless parameters are multiplied by coefficients that are functions of mach number. Notice that the dimensionless alpha dot and beta dot are functions of the aircraft's velocity and the coefficients are a function of mach number. Since the speed of sound changes with respect to the altitude above sea level, the contribution of these terms is dependent on the aircraft's altitude.

Then, in subroutine AEROFM, these dimensionless parameters are multiplied by coefficients that are functions of mach number to determine their contribution to the appropriate forces and moments. Alpha-dot terms contribute to the lift force and pitching moment coefficients. Beta-dot terms contribute to the yawing moment coefficient. The next three pages contain figures that graphically illustrate the effect of these time-dependent forces and moments.

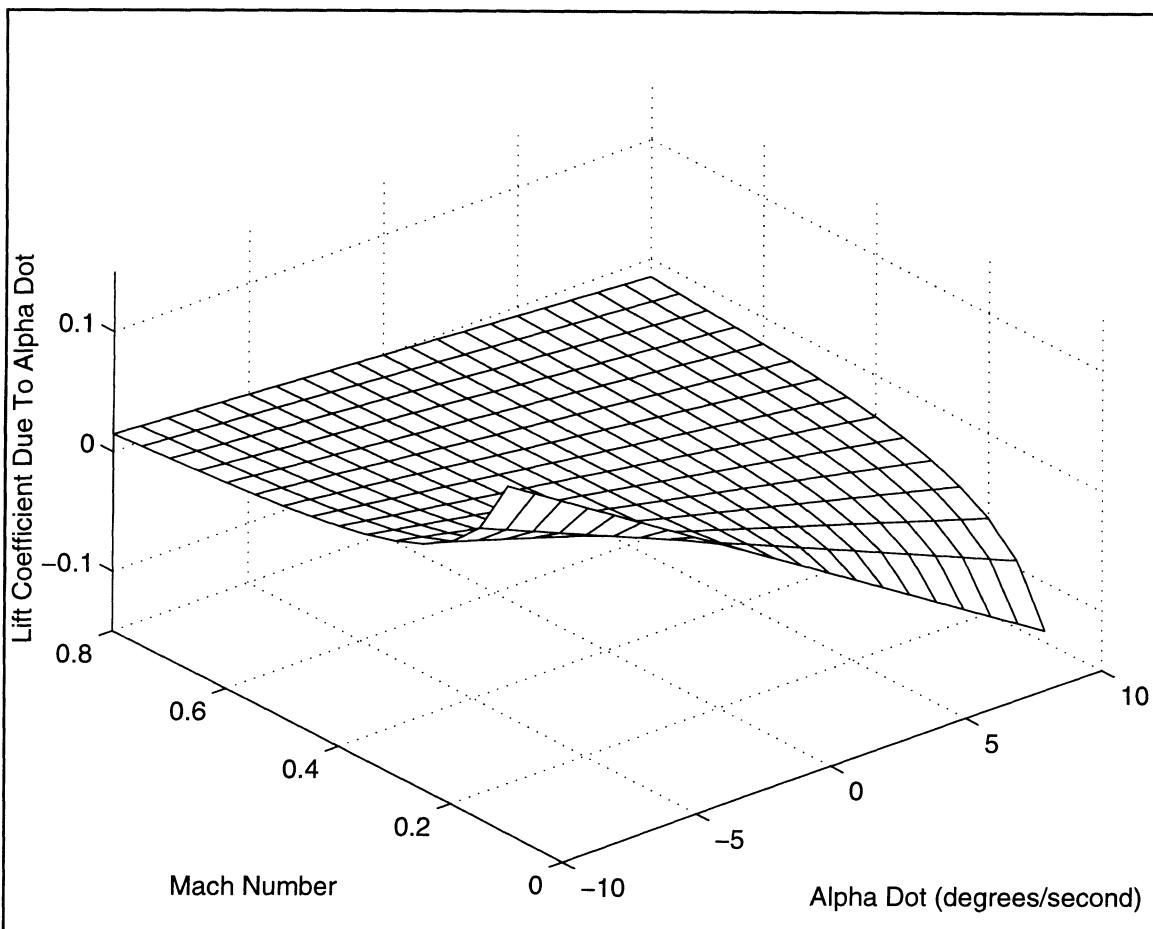


Figure 3.2 Lift Coefficient Due to Alpha-Dot at 100 ft Altitude

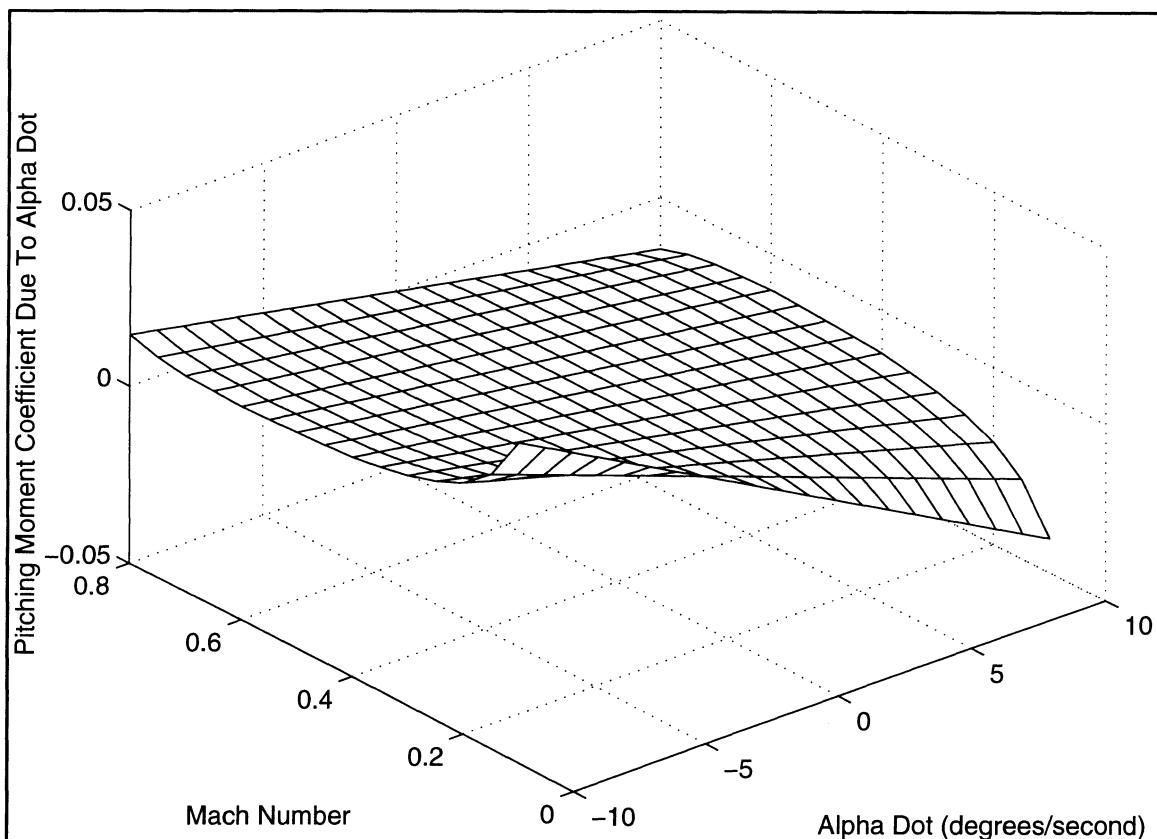


Figure 3.3 Pitching Moment Coefficient Due to Alpha-Dot at 100 ft Altitude

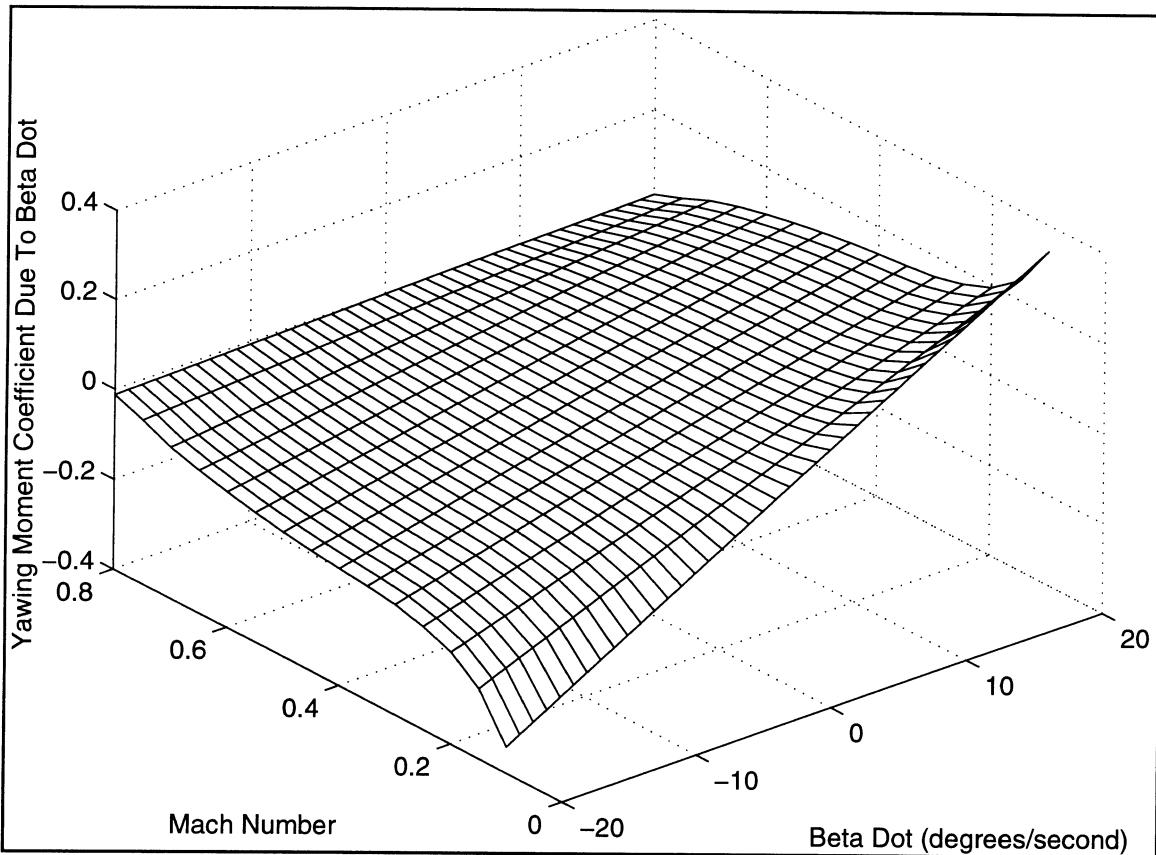


Figure 3.4 Yawing Moment Coefficient Due to Beta-Dot at 100 ft Altitude

Chapter 4: OVERVIEW OF SIMULINK BLOCK DIAGRAMS

The actuator models and the wind model have been implemented as SIMULINK block diagrams. This chapter describes these block diagram models. The figures referenced in this chapter are located at the end of the chapter.

The actuators of an aircraft are the link between the pilot or automatic control system and the control surfaces on the aircraft. Therefore, the actuator models take as input a commanded control surface deflection and return the actual control surface deflection. The basic model of an actuator is a lag function to reflect the delay between issuing a command and the motion of the desired control surface. Some actuator models also take into account nonlinear effects such as cable stretch, hysteresis, relay, and the force of the air mass on the control surface. Calculation of some of these additional forces that act on the control surface require aerodynamic parameters and the aircraft velocity through the air mass. The actuator models are described in Sections 4.1 through 4.6.

The wind model calculates the wind velocities and accelerations, which are disturbance inputs to the aircraft. The user may choose from two different wind model implementations. The wind model is described in Section 4.7.

4.1 Elevator Actuator Block Diagram Description

The elevators of an aircraft are located on the horizontal part of the tail. Deflection of these control surfaces produce a pitching moment. The stabilizer, described in Section 4.2 is used to trim the aircraft so that a constant commanded elevator deflection is not necessary to maintain the aircraft's altitude, pitch, and speed.

The block diagram of the elevator model is shown in Figure 4.1. The inputs of this model are shown in Table 4.1. The output is the elevator position in degrees, DELE.

Variable	Description
DELEC	commanded elevator deflection (degrees)
SP	stabilizer position (pilot units)
FLAPS	flap position (degrees)
QBAR	dynamic pressure (pounds/square inch)
CG	cg of the aircraft (feet from nominal along body x-axis)
EAS	indicated airspeed (knots)

Table 4.1 Inputs to the Elevator Actuator Model

This model contains a sub-system that implements a modified lag function. This subsystem is shown in Figure 4.2. This subsystem provides for

a system response that is dependent on both the commanded deflection and its derivative. Additionally, a nonlinear rate limit and saturation constrain the output. This modified lag function is also used in the rudder actuator simulation described in Section 4.4.

The elevator actuator model contains a cable stretch nonlinearity as well as a hinge moment nonlinearity. The hinge moment is a function of the lift created by the flaps, the mach number, and current elevator position. This moment serves to "blow back" the control surface from the commanded position.

4.2 Stabilizer Actuator Block Diagram Description

The stabilizer is used to trim the elevator so that the aircraft can maintain a constant altitude, velocity, and pitch angle without a constant, non-zero elevator command.

The inputs to the stabilizer are listed in Table 4.2 and the block diagram is shown in Figure 4.3. The subsystem output is the stabilizer position in pilot units, SP. The aircraft dynamics simulation uses an automatic stabilizer as well as inputs from the pilot. The pilot inputs consist of SPUP and SPDN, which indicate whether the pilot would like to trim up or down. A trim up raises the stabilizer to induce a nose-up pitching moment and a trim down lowers the stabilizer to induce a nose-down pitching moment. The automatic stabilizer uses the commanded aileron position, DELACM, and the flap

position, FLAPS, to set the automatic contribution to the stabilizer.

Variable	Description
SPUP	stabilizer trim up (logical)
SPDN	stabilizer trim down (logical)
FLAPS	flap position (degrees)
DELEC	commanded elevator position (degrees)

Table 4.2 Inputs to the Stabilizer Actuator Model

4.3 Aileron Actuator Block Diagram Description

The ailerons are located on the outboard aft portion of both wings. For a given deflection command, the ailerons on each wing will be deflected in opposite directions to achieve a rolling moment.

The inputs to the aileron actuator model are listed in Table 4.3 and the block diagram is shown in Figure 4.4. The sub-system that implements the fast dynamics of the aileron actuator is shown in Figure 4.5. The aileron model outputs the aileron deflection, DELA, and the right and left spoiler positions, SPR and SPL. The position of the spoilers is a function of the speed brake handle position and the aileron position.

The aileron model contains many nonlinear effects. The deployment of flaps or spoilers change the configuration of the wing, and therefore change the way that the air mass passes over the ailerons. Additionally, cable stretch and

Variable	Description
DELACM	commanded aileron deflection (degrees)
SPHP	speed brake handle position (pilot units)
FLAPS	flap position (degrees)
QBAR	dynamic pressure (pounds/square inch)

Table 4.3 Inputs to the Aileron Actuator Model

hysteresis are accounted for in the model.

4.4 Rudder Actuator Block Diagram Description

The rudder is located on the aft of the vertical tail. Rudder deflections will cause a yawing moment.

The inputs to the rudder actuator model are listed in Table 4.4 and the block diagram is shown in Figure 4.6. Notice that the rudder also uses the modified lag function block, which is shown in Figure 4.7. This is similar to the modified lag block used in the elevator actuator model, but in this case there is a nonlinear saturation that is applied to the output. A hysteresis nonlinearity is applied to the output of the modified lag function, the result yields the control surface deflection.

4.5 Flap Actuator Block Diagram Description

The flaps are located on the inboard aft side of each wing. The flaps are

Variable	Description
DELRC	commanded rudder deflection (degrees)
EAS	indicated airspeed (knots)

Table 4.4 Inputs to the Rudder Actuator Model

used to put the aircraft into a high lift configuration by extending the aft side of the wing surface downward. This wing configuration produces significantly more lift and is useful when flying at the slow speeds required to land the airplane.

A block diagram of the flap actuator model is shown in Figure 4.8. The input to this actuator model is the commanded flap position, FLCMP. The output is the actual flap position, FLAPS.

4.6 Throttle Actuator Block Diagram Description

The throttle command is passed to the engine in order to command the amount of thrust each engine produces. The block diagram of the throttle actuator is shown in Figure 4.9. The only input to this actuator is the commanded throttle position, THROTLC. The output is the actual throttle position, THROTLE. The throttle actuator is implemented as a discrete-time block. This is executed at 20 Hertz.

4.7 Wind Model Description

The wind model has one input, the aircraft altitude, ALT. The outputs consist of the wind velocities and accelerations in the NED coordinate system. Table 4.5 lists the output variables. The block diagram of the wind model is shown in Figure 4.10. In addition, there are six constant blocks that the user must set. These six blocks contain the runway altitude (ft), constant wind downward velocity (knots), constant wind side velocity (knots), constant wind direction (degrees), the wind shear logical, and the constant wind logical.

Variable	Description
VNW, VEW, VDW	wind velocities, NED frame (ft/s)
ANW, AEW, ADW	wind accelerations, NED frame (ft/s)

Table 4.5 Outputs of the Wind Model

The wind shear model consists of two lookup tables that schedule the wind shear downward and side velocity as a function of the aircraft's distance above the runway. These lookup tables have been initialized with the wind model provided by NASA. The user may select whether to use the wind shear model by setting the wind shear logical (0-wind shear OFF, 1-wind shear ON).

The constant wind model applies a constant wind independent of the aircraft's altitude. The user defined constant wind downward velocity, constant wind side velocity, and constant wind direction parameters define the

magnitude and direction of the constant wind. The user may select the constant wind model by setting the constant wind logical (0-constant wind OFF, 1-constant wind ON).

These models are not mutually exclusive. If the user selects both models, the wind velocities from the two models are added together. The user may, of course, add another wind model in this block to meet the needs of a particular application.

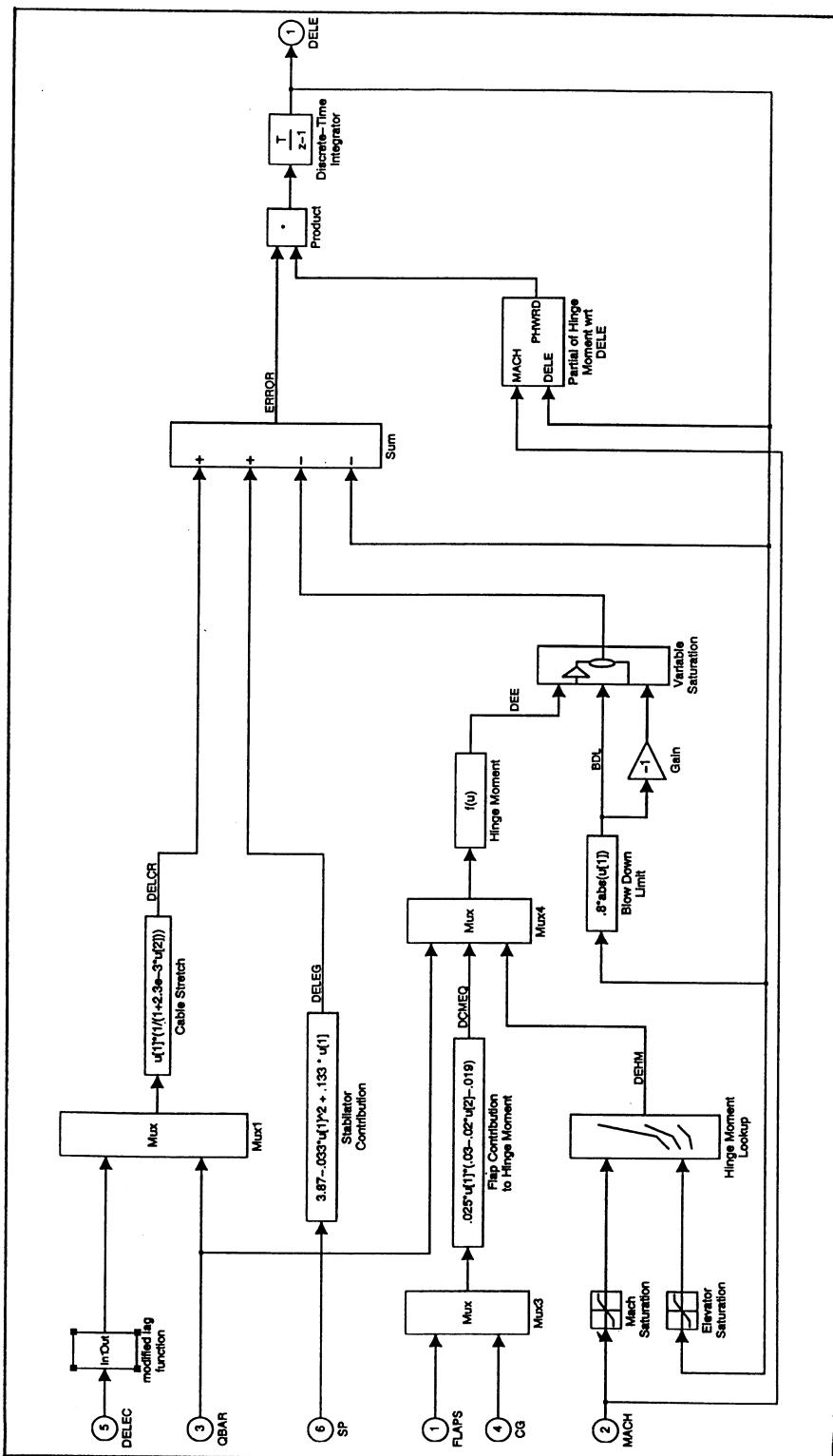


Figure 4.1 Elevator Actuator Model Block Diagram

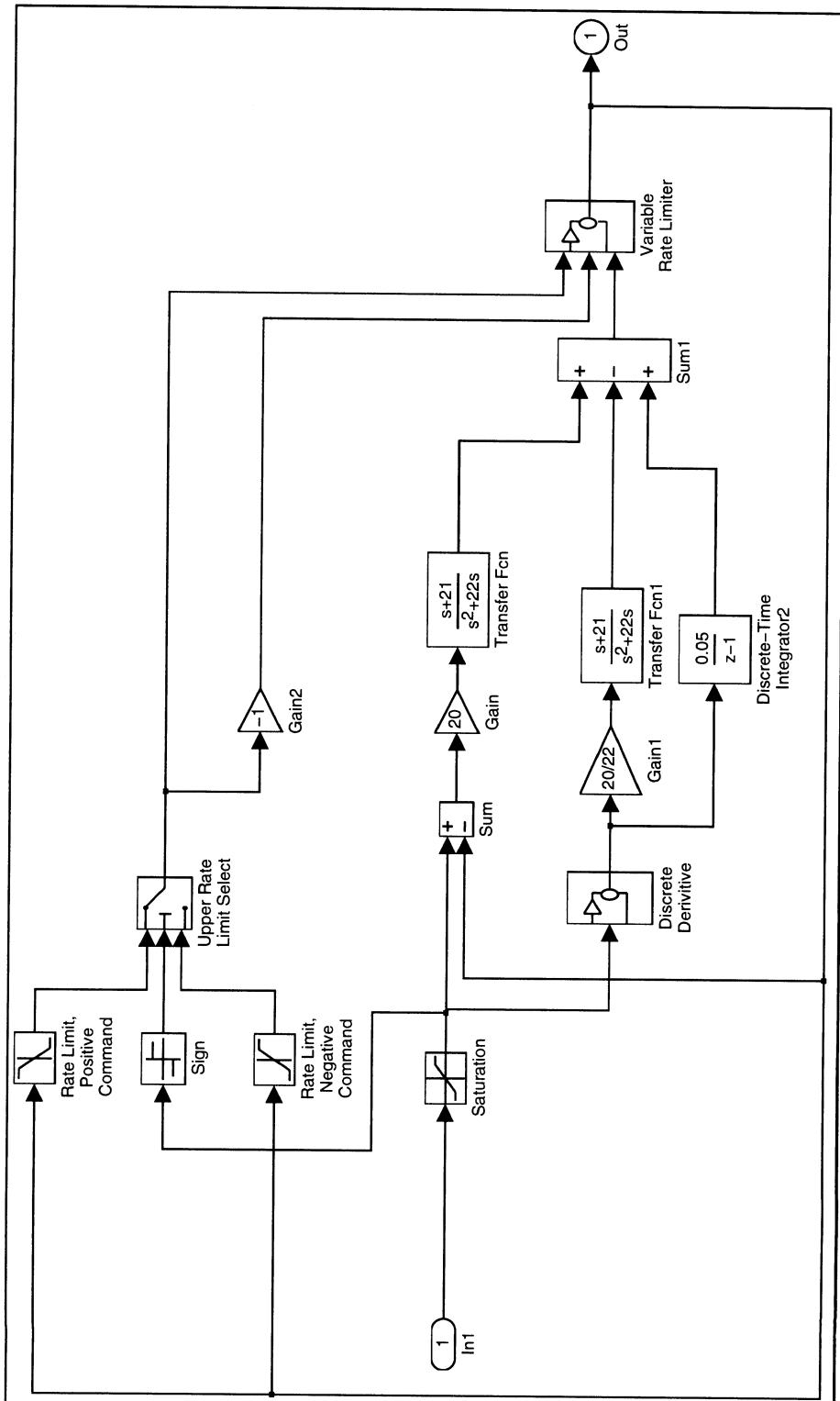


Figure 4.2 Modified Lag Sub-System for the Elevator Model

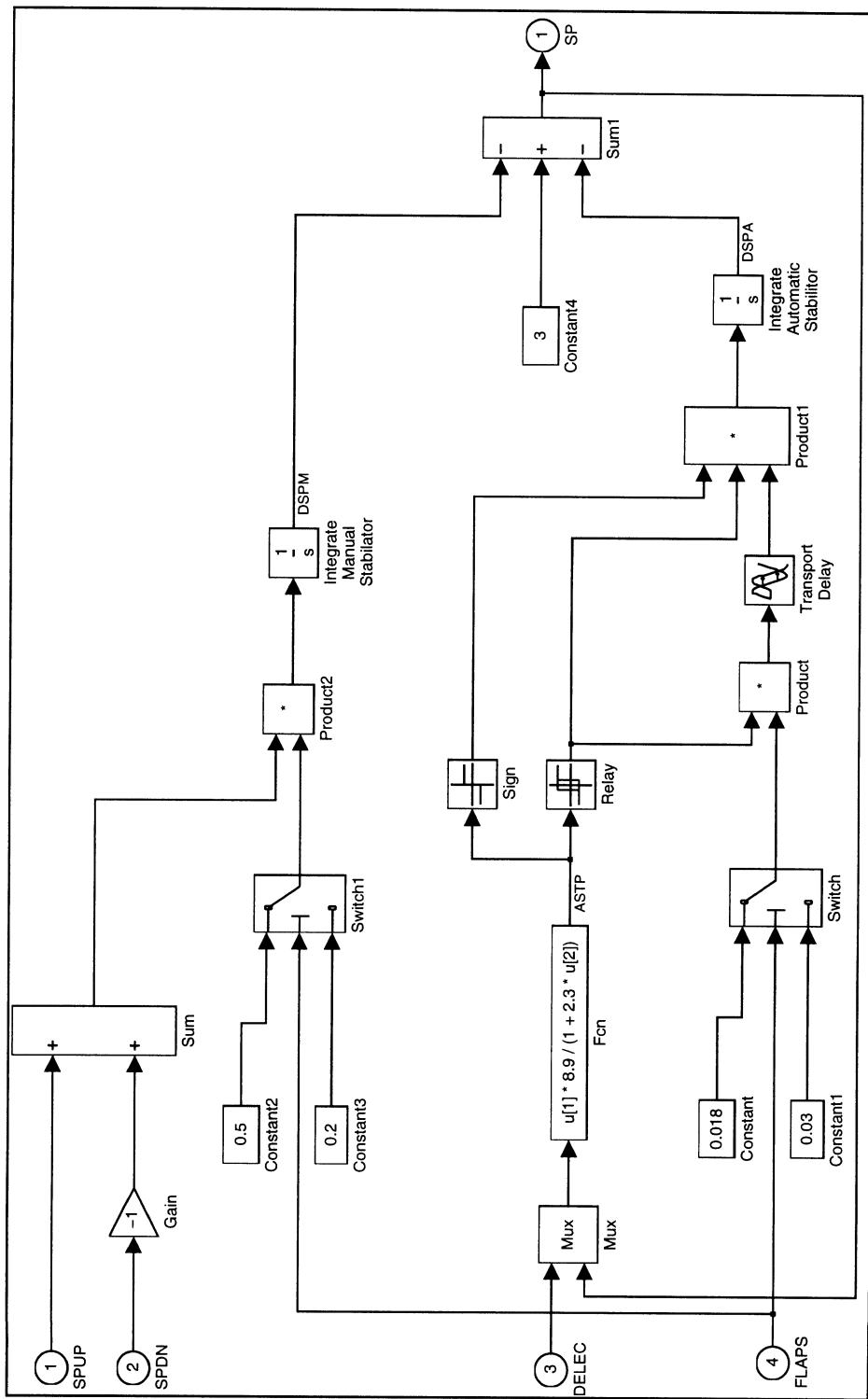


Figure 4.3 Stabilitor Actuator Model Block Diagram

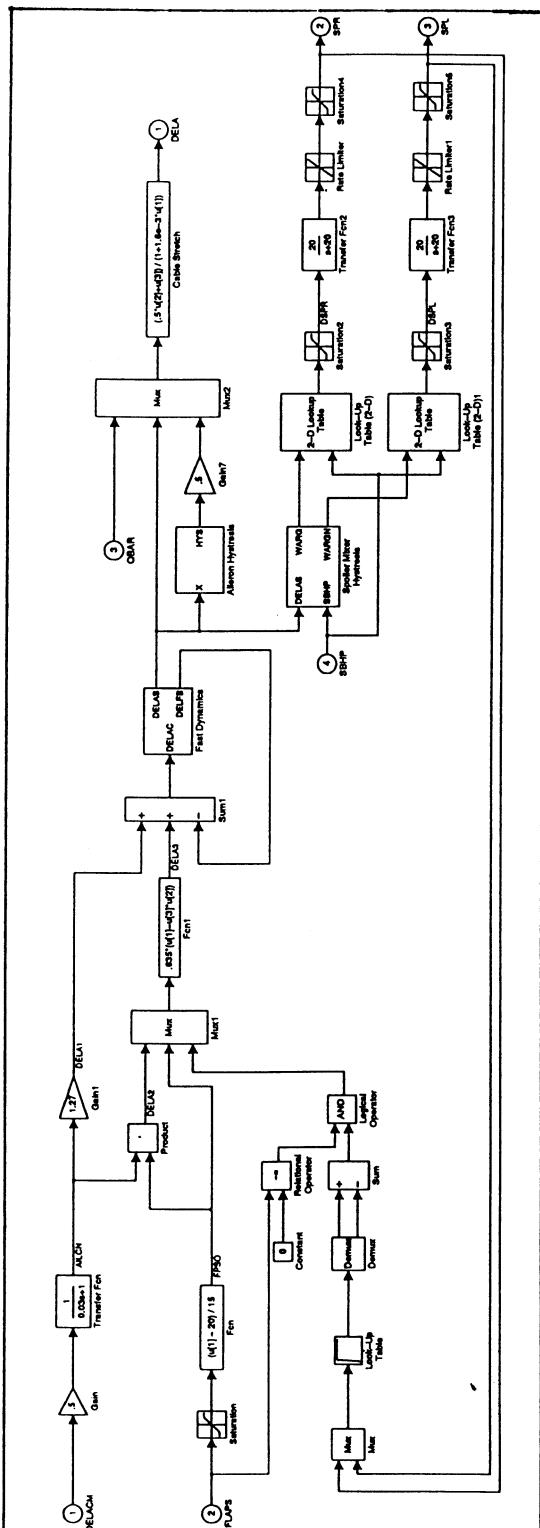


Figure 4.4 Aileron Actuator Model Block Diagram

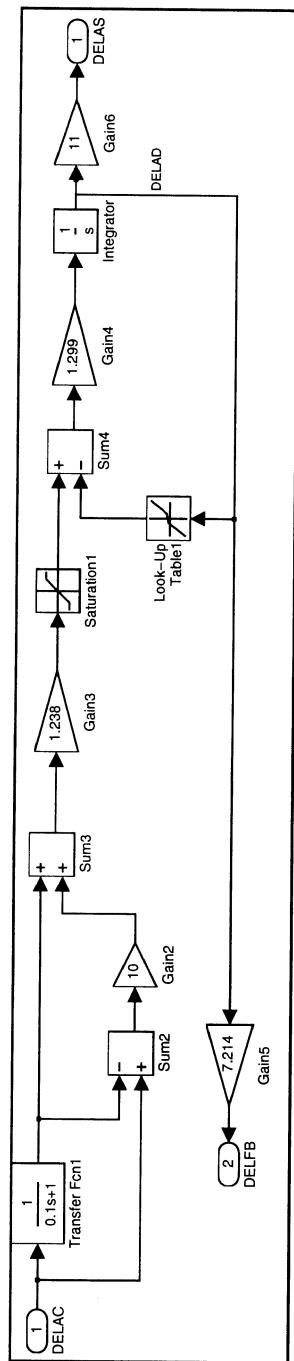


Figure 4.5 Fast Dynamics Sub-System for the Aileron Block Diagram

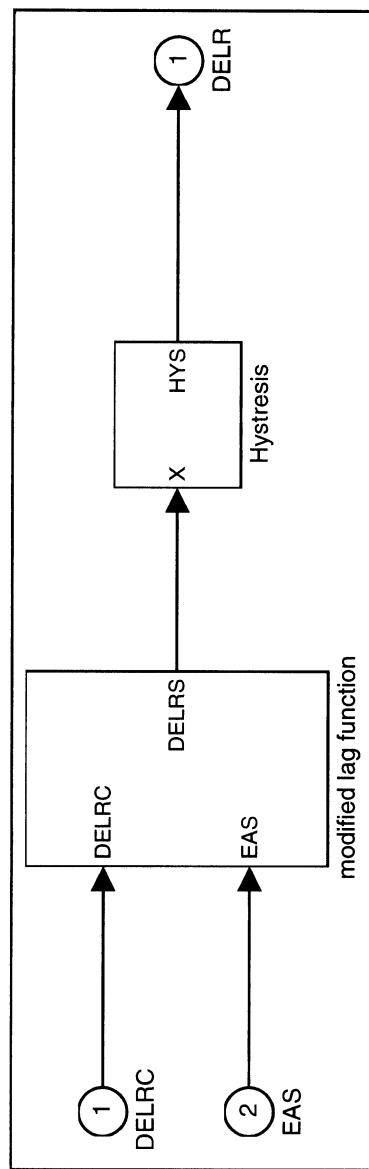


Figure 4.6 Rudder
Actuator Model

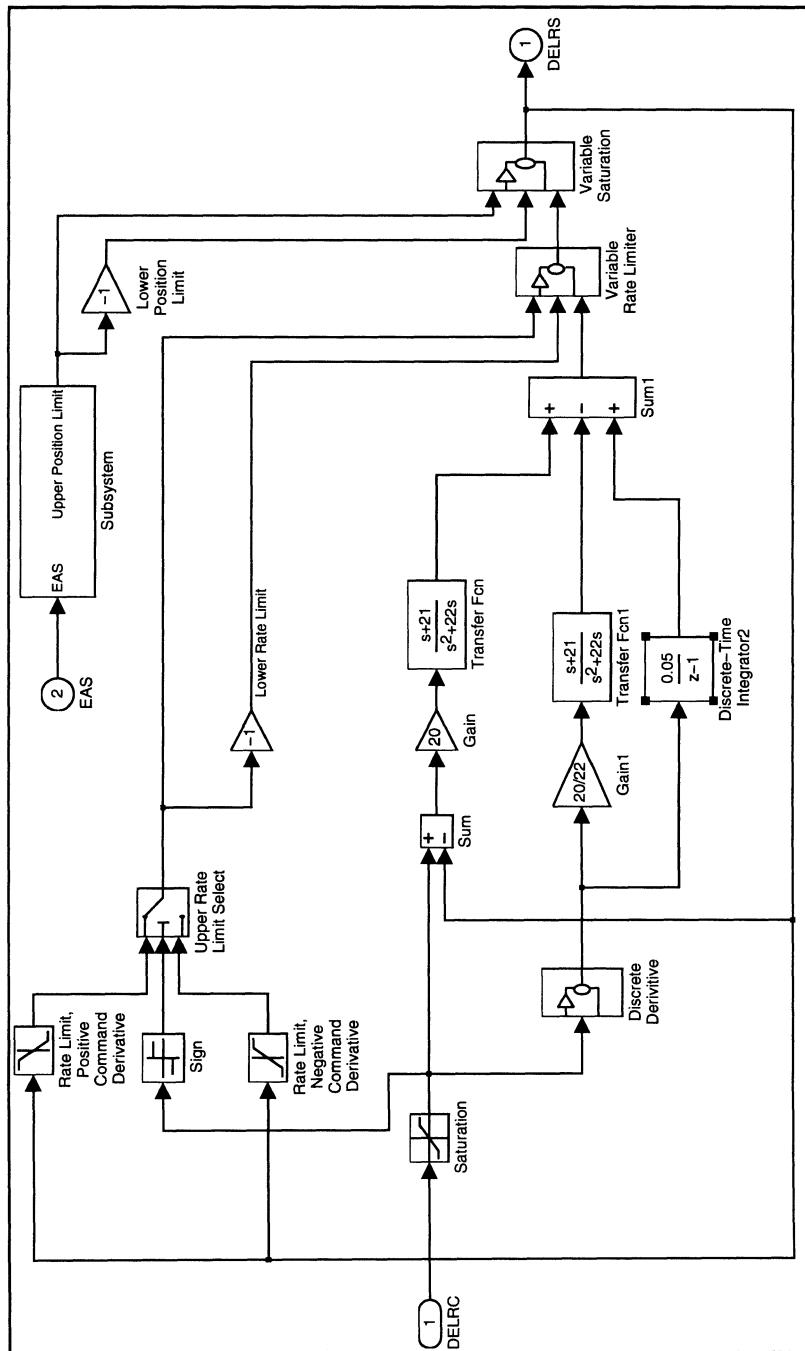


Figure 4.7 Modified Lag Sub-System for the Rudder Actuator Model

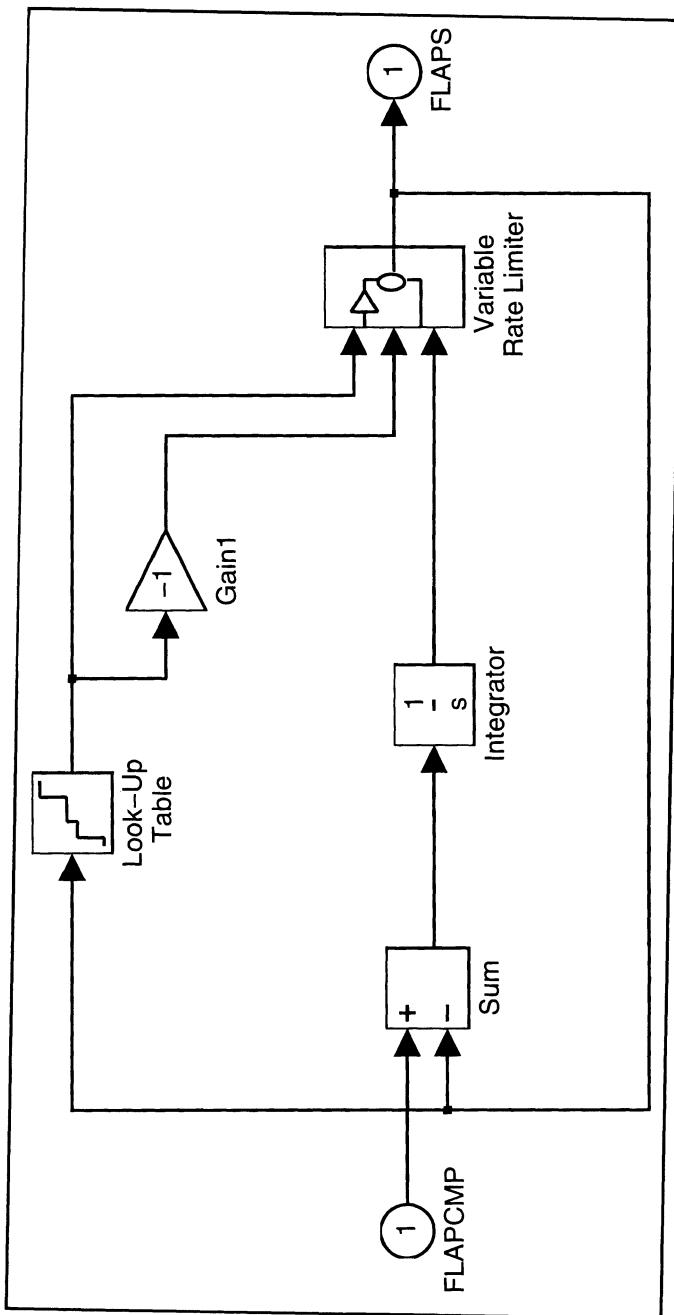


Figure 4.8 Flap Actuator Model Block Diagram

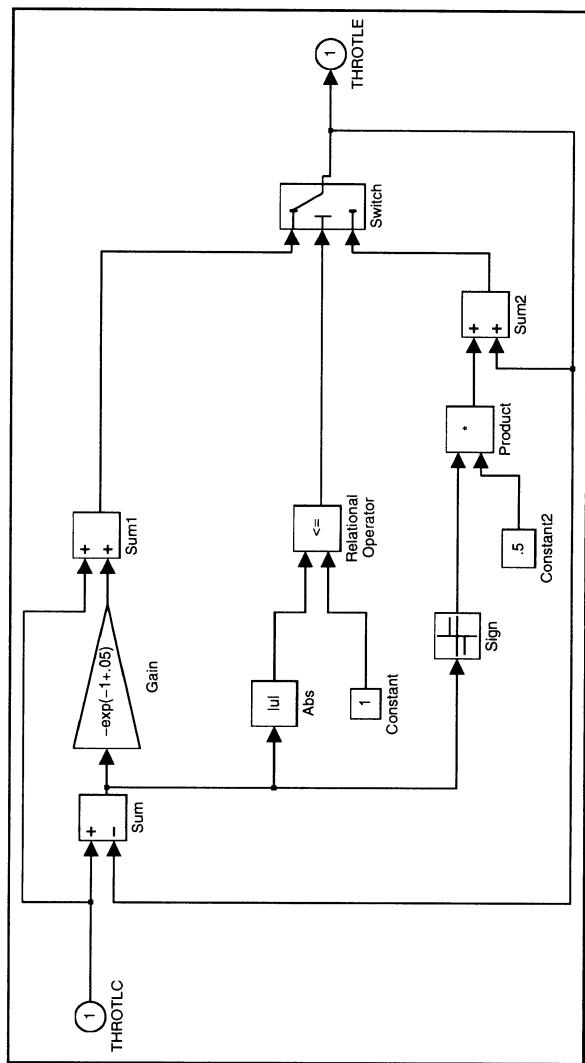


Figure 4.9 Throttle Actuator Model
Block Diagram

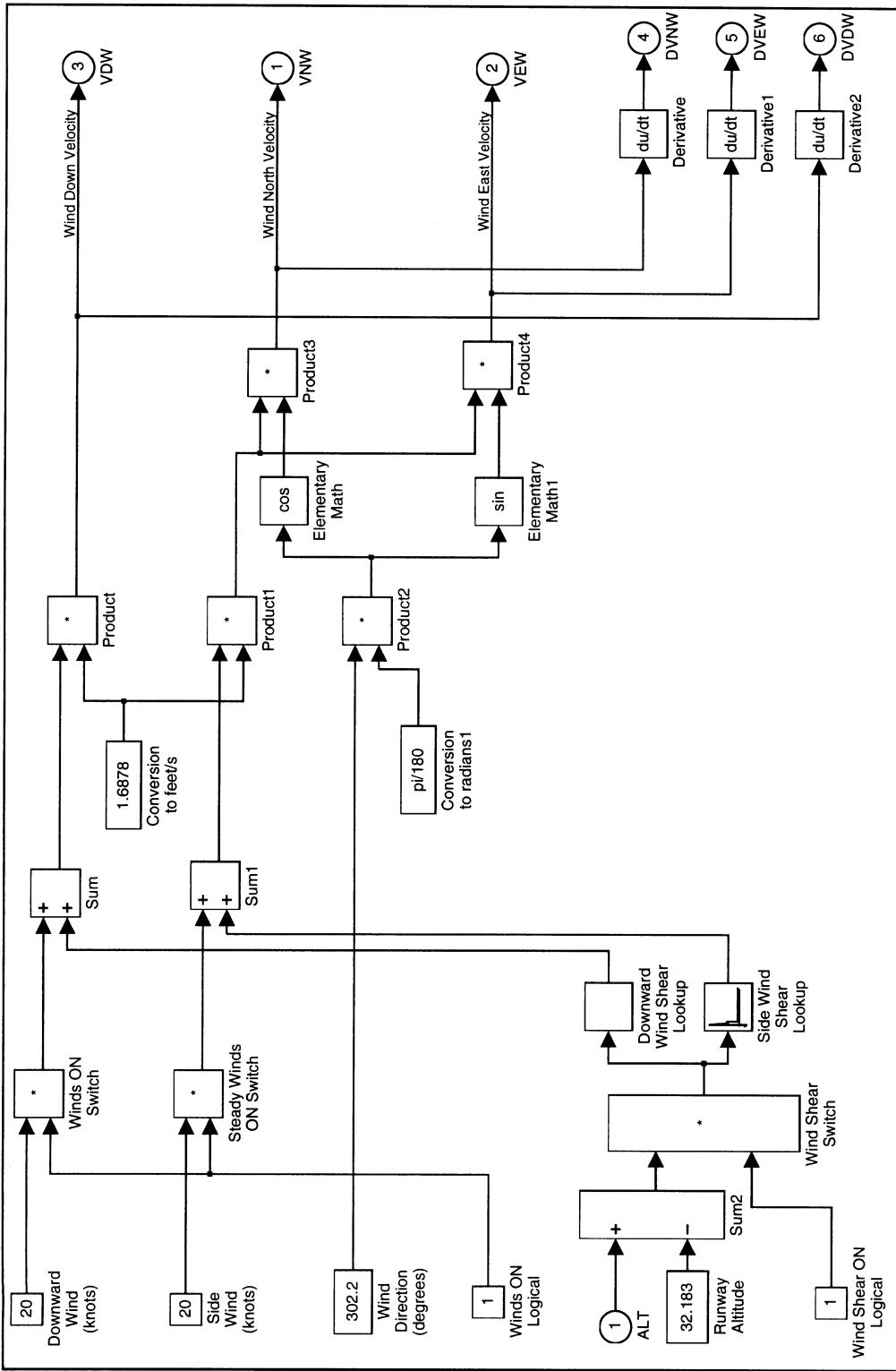


Figure 4.10 Wind Model Block Diagram

Chapter 5: SIMULATION RESULTS

This chapter presents the results of the comparison between the FORTRAN twin jet engine simulation and the SIMULINK implementation of the simulation. An autoland was simulated on the FORTRAN simulation using the autoland controller supplied by NASA. The commanded control surface deflections and states were recorded during the autoland. The SIMULINK model was then initialized to the trim settings of the FORTRAN simulation. Finally, the SIMULINK model was executed open loop using the recorded commanded control surface deflections and the aircraft states were recorded. A comparison of the aircraft state time histories for each implementation of the model shows that the SIMULINK simulation matches the FORTRAN simulation.

The first section of this chapter describes the autoland flight path and the wind model used. Section 2 presents graphs of the state time histories for the FORTRAN simulation and the SIMULINK model from the autoland.

5.1 Autoland Flight Path

The control system that was used to execute the autoland flight path used a simulation of an Instrument Landing System (ILS) for guidance. An ILS consists of two groundstations, the localizer and the glideslope, that produce signals for lateral and vertical guidance. The localizer is located at

the far end of an ILS equipped runway and sends out a beam that incoming aircraft can use for lateral guidance. The glideslope produces a beam that the aircraft uses for vertical guidance down to the glideslope intercept point (GPIP) on the runway. The glideslope is nominally three degrees in inclination and 52 feet over the runway at the runway threshold. In this simulation, the glideslope and localizer were ideal (no beam bends).

There are four significant events during an Instrument Landing System (ILS) autoland; localizer capture, glideslope capture, decrab, and flare. During localizer capture, the control system aligns aircraft heading with the runway heading using the localizer signal. Glideslope capture occurs when the aircraft receives the glideslope beam signal. The glideslope used in this autoland is three degrees in elevation. The aircraft then continues to use the localizer for lateral guidance and uses the glideslope for vertical guidance down to the threshold of the runway.

The autoland control system initiates a decrab maneuver when the aircraft's rear landing gear is 150 feet above the runway. This maneuver aligns the aircraft's plane of symmetry with the velocity vector. Therefore this maneuver reduces the sideslip angle, β , to zero and aligns the aircraft's plane of symmetry with the runway. The final maneuver that the control system initiates is the flare. The flare is initiated when the aircraft's rear landing gear is 42 feet over the runway. This maneuver slows the vertical decent rate so that the aircraft touches down on the runway with a vertical

velocity of approximately 3 ft/s.

The wind model was used to introduce a disturbance during the autoland. A constant wind of 20 knots with a heading of 302.2 degrees was on during the entire simulation. This wind is in excess of certification requirements on the autoland system, and the performance of the autoland controller is degraded in this scenario. The runway heading is 212.2 degrees.

5.2 Time Histories of the Aircraft States

The following figures contain the time histories of the aircraft dynamics model states through the autoland described above. Each plot contains the time history of the states and the error between the two simulations on a separate axis. The dotted line is the time history of the SIMULINK implementation and the solid line is the time history of the FORTRAN simulation. In this context, the term ‘error’ is defined to be the difference between the SIMULINK simulated aircraft states and the FORTRAN simulated aircraft states during the test condition.

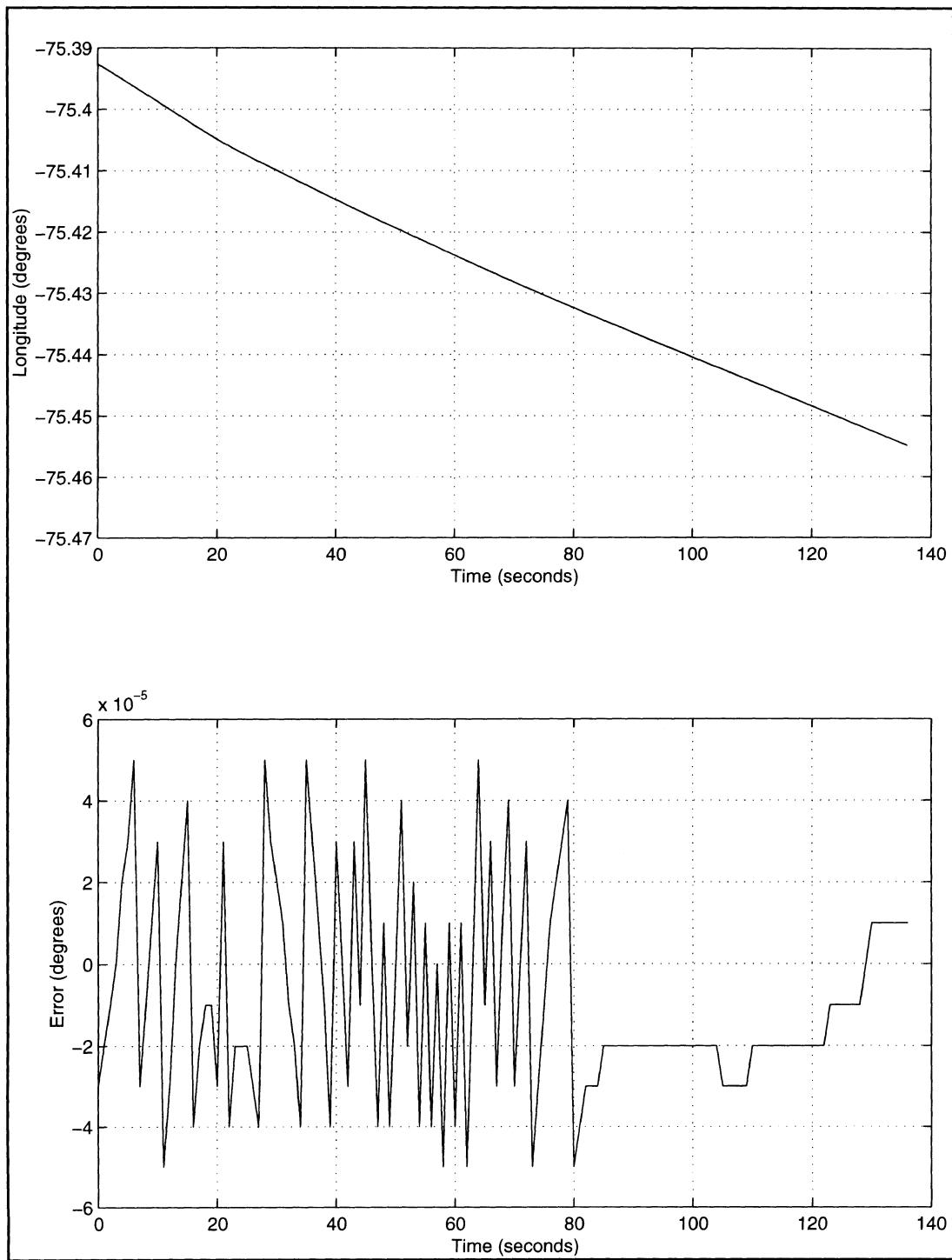


Figure 5.1 Time History of Longitude

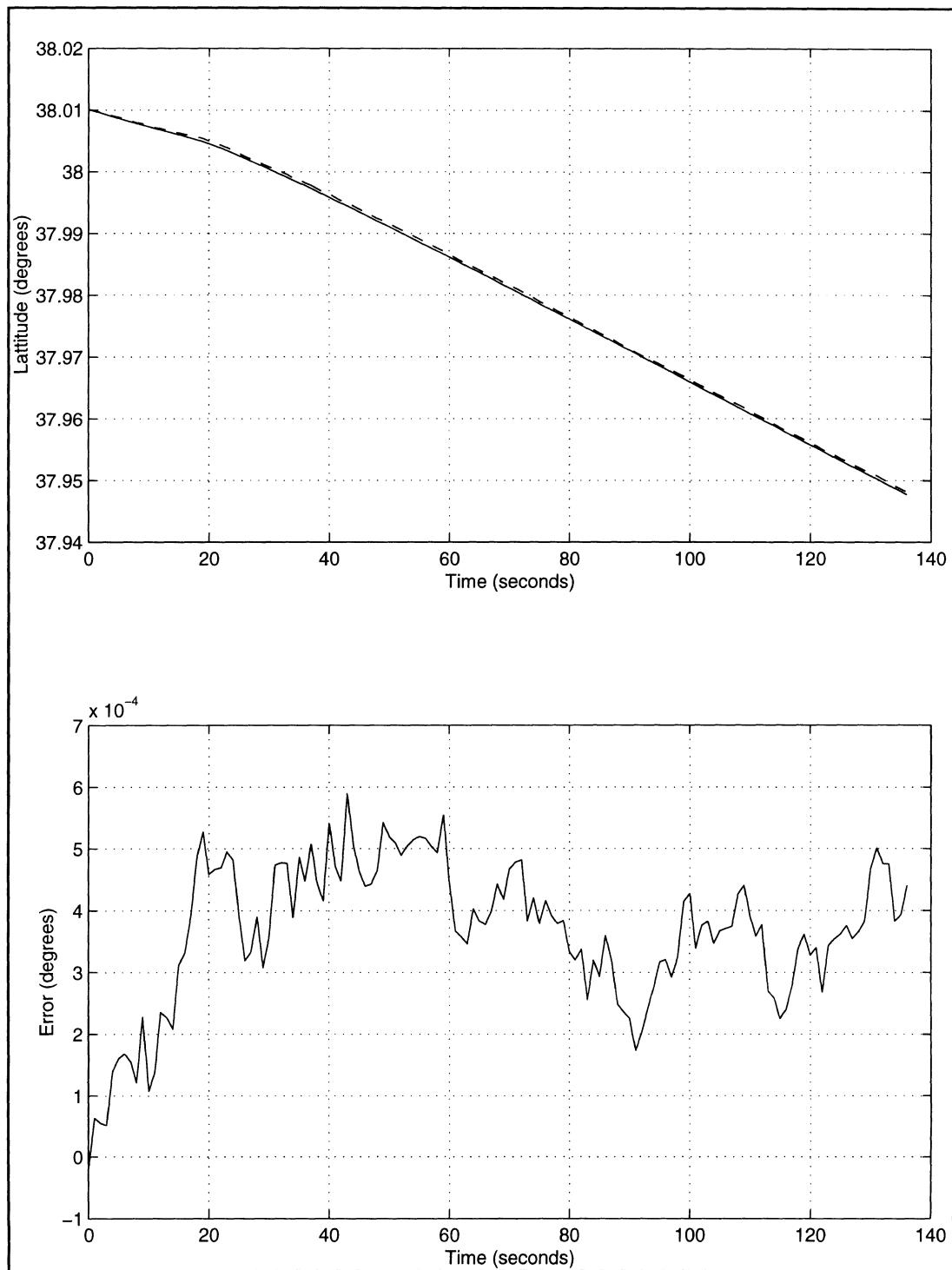


Figure 5.2 Time History of Latitude

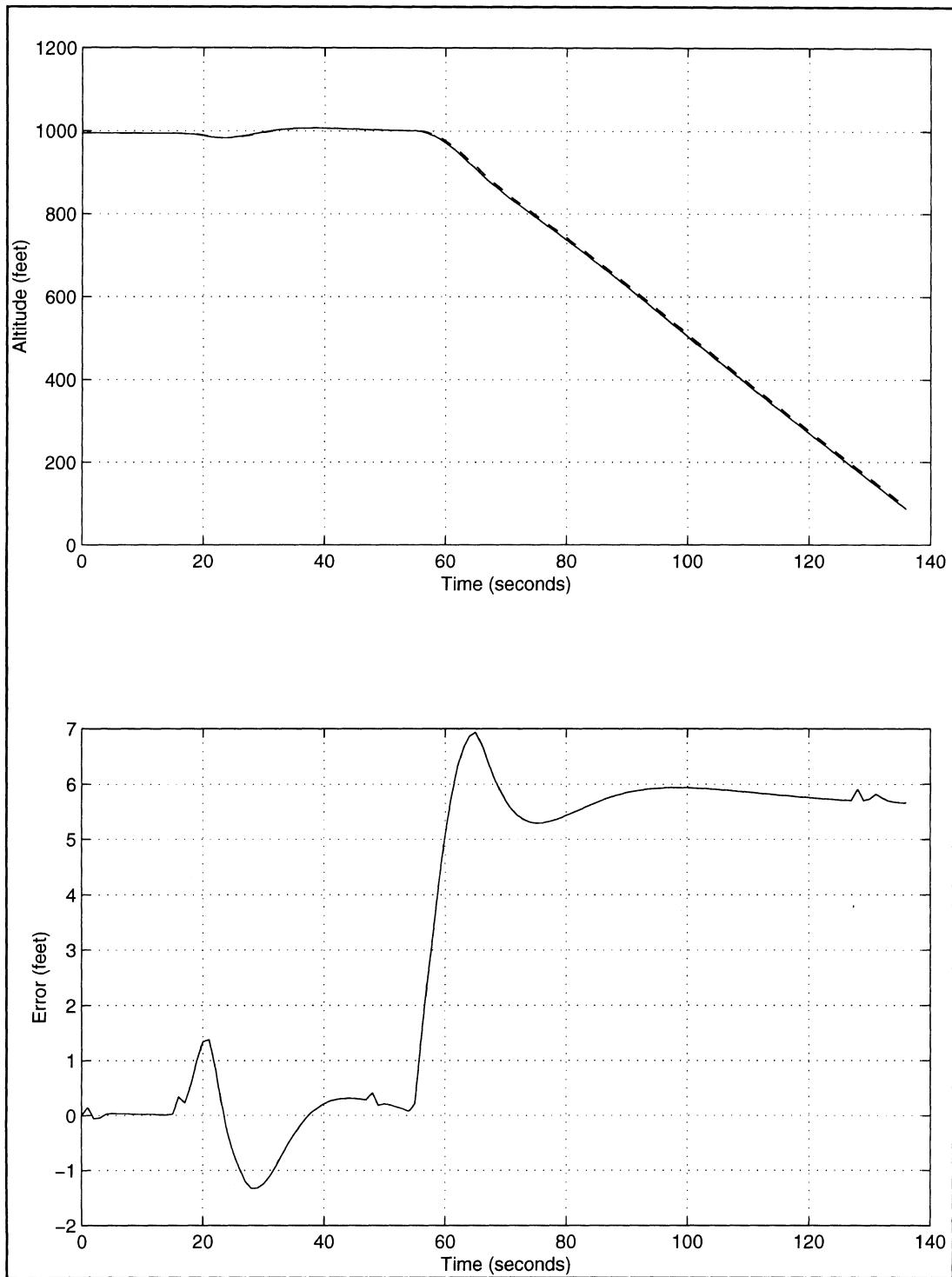


Figure 5.3 Time History of Altitude

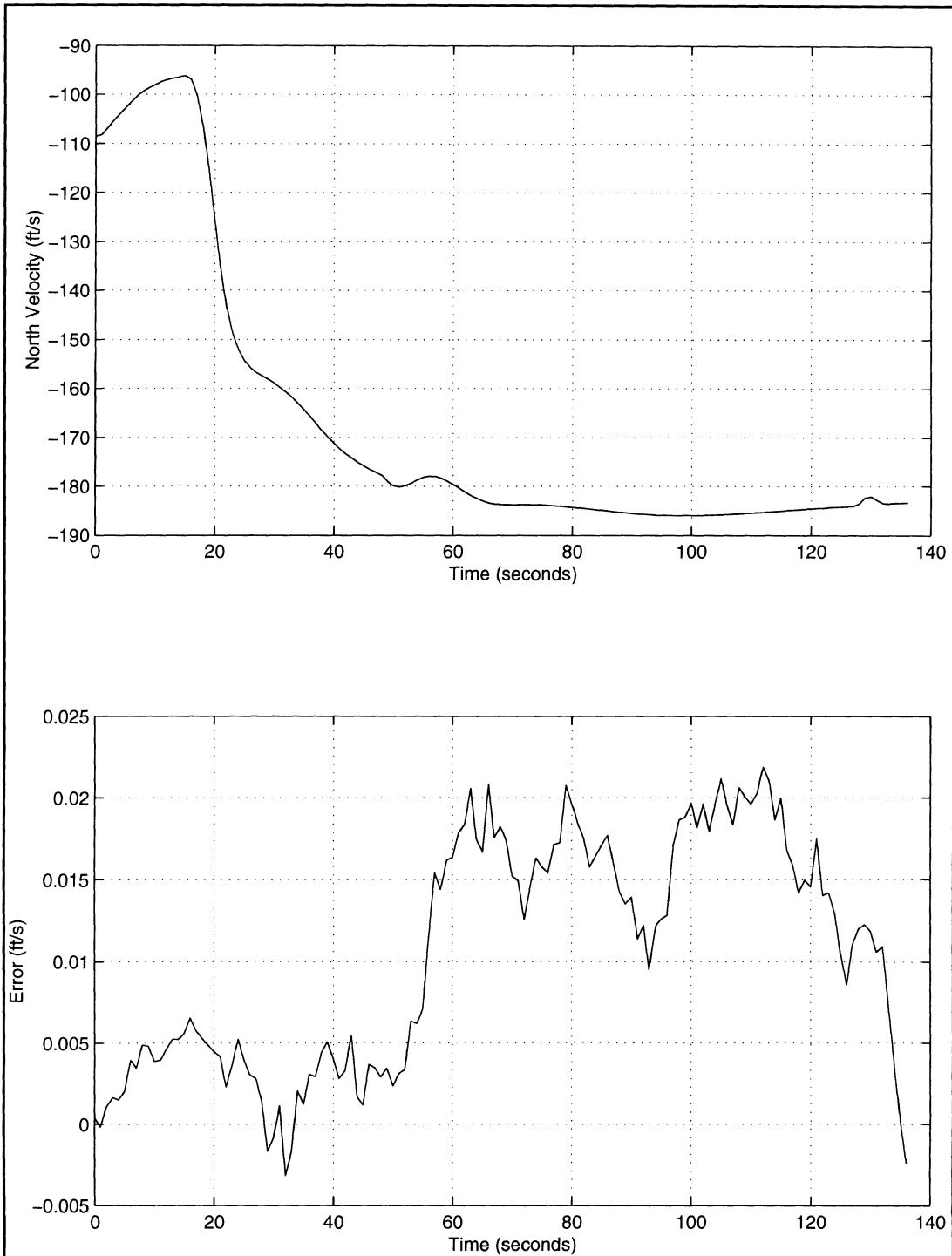


Figure 5.4 Time History of North Velocity

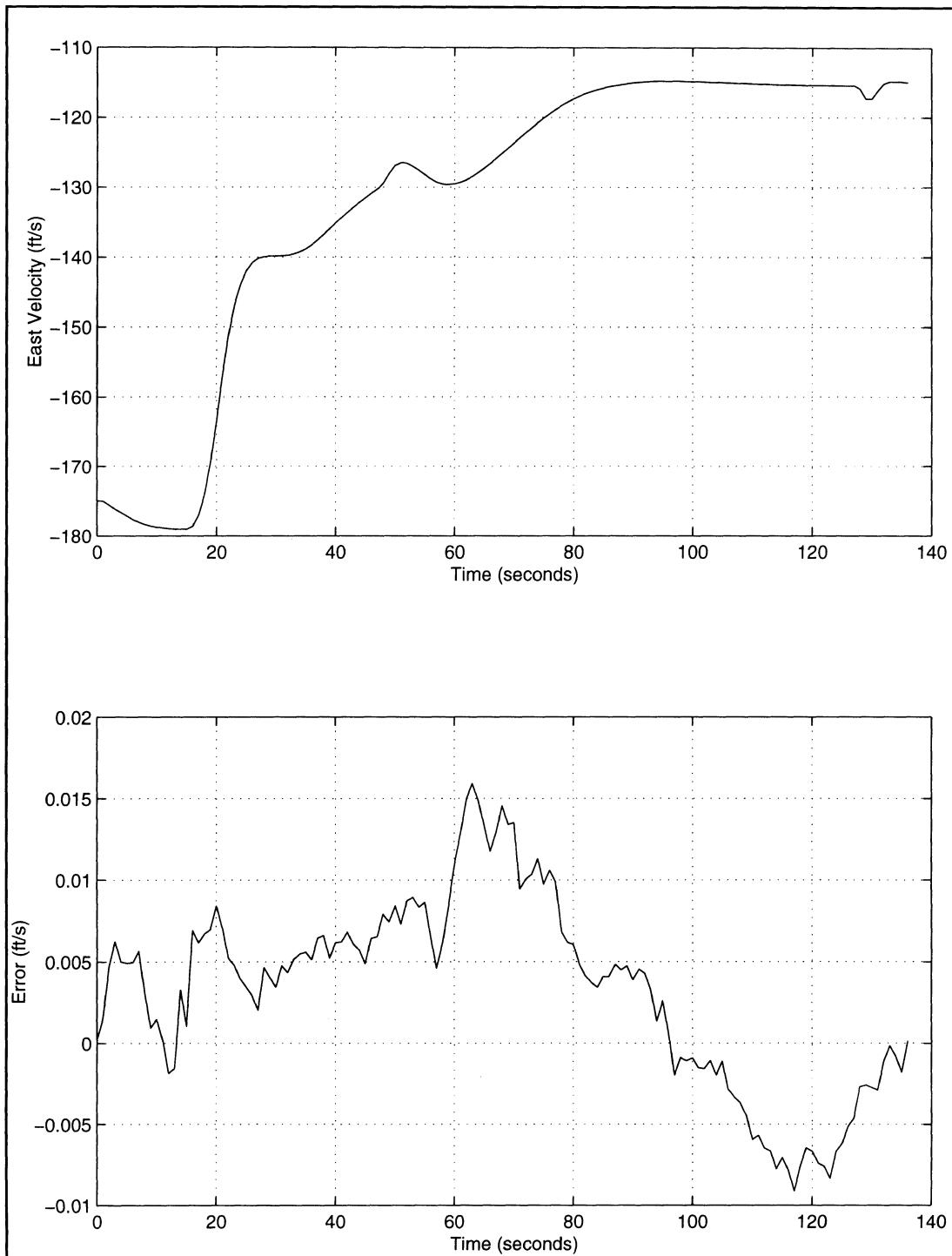


Figure 5.5 Time History of East Velocity

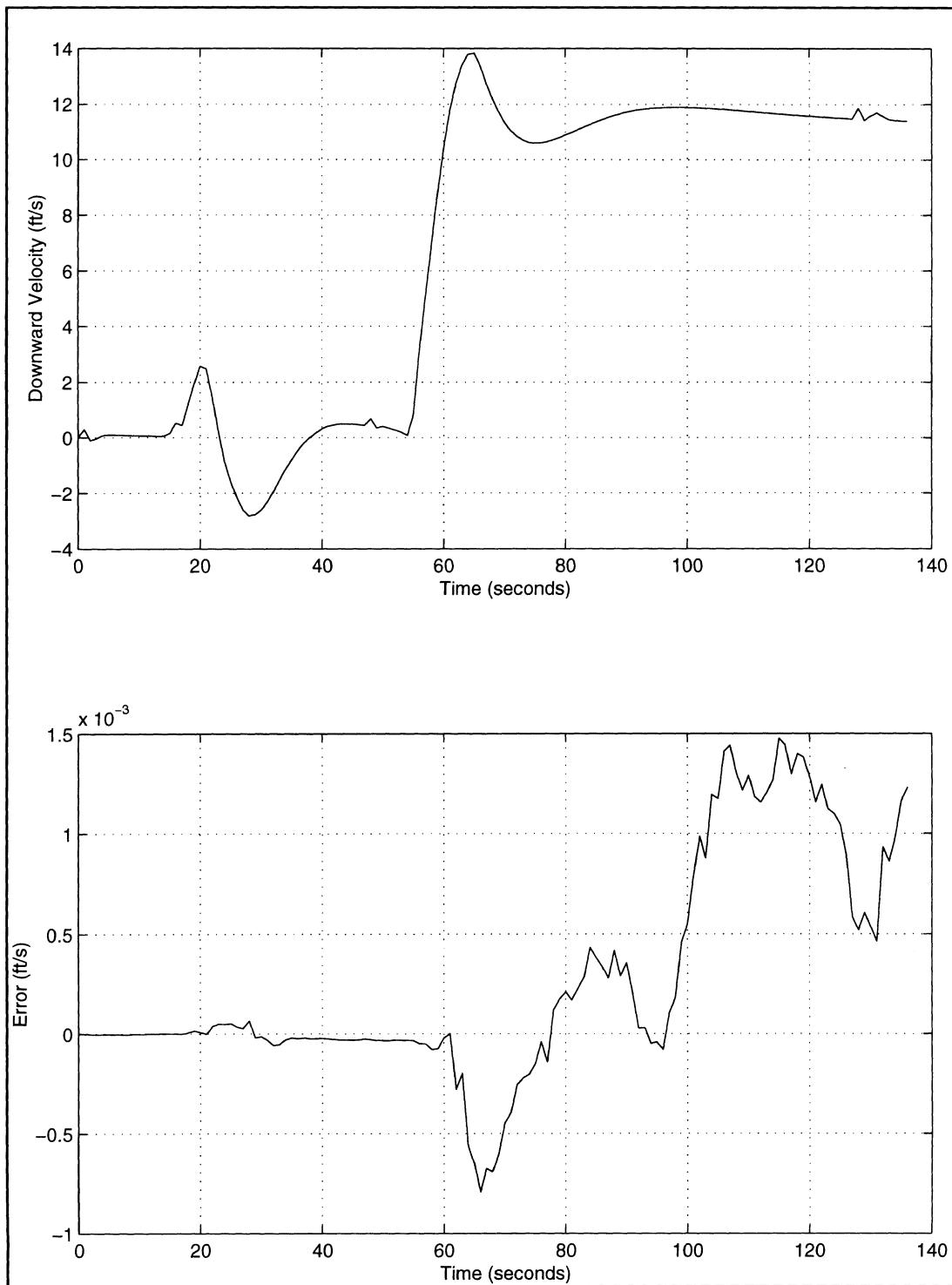


Figure 5.6 Time History of Down Velocity

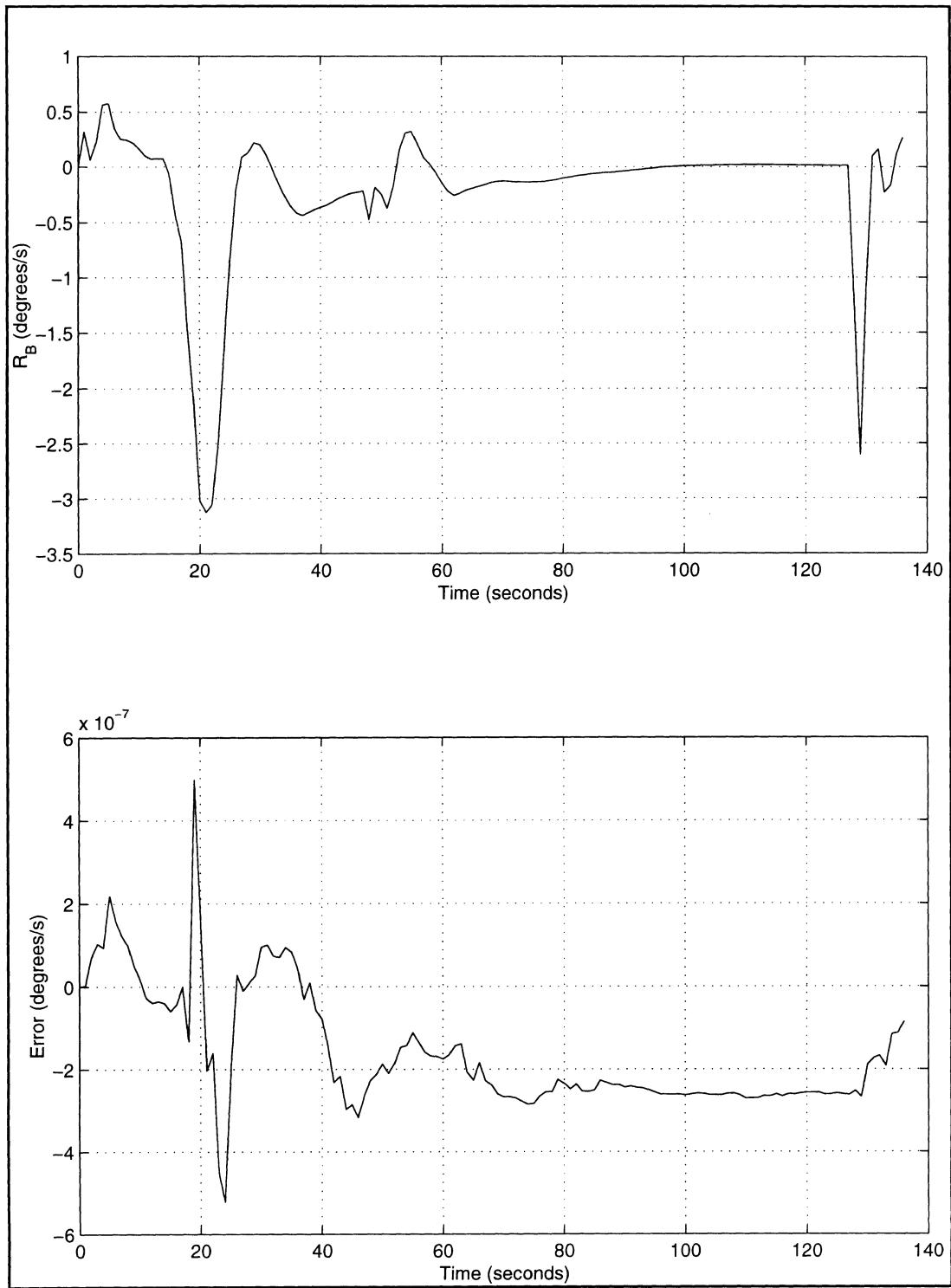


Figure 5.7 Time History of Yaw Rate (Body Axes)

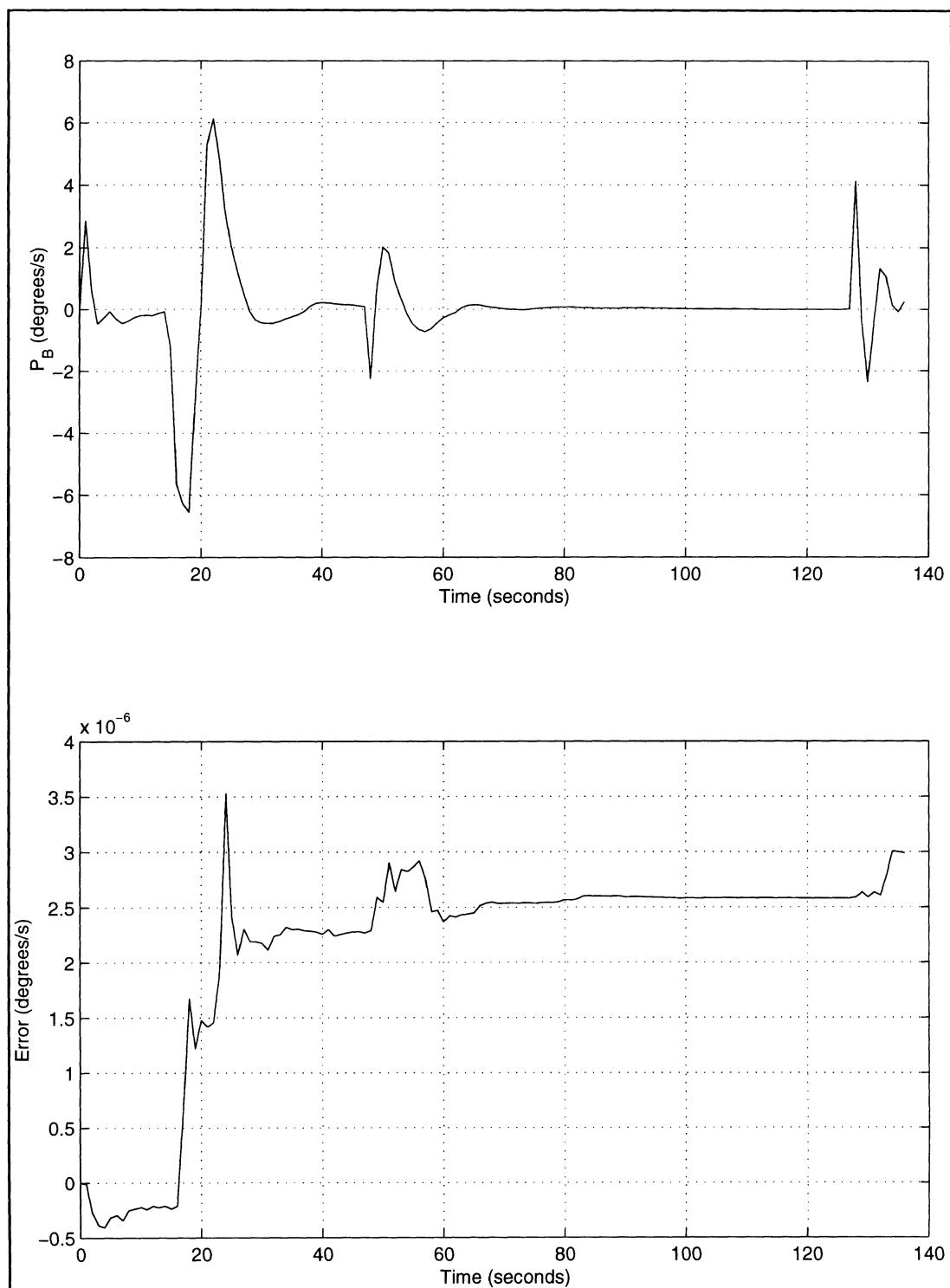


Figure 5.8 Time History of Roll Rate (Body Axes)

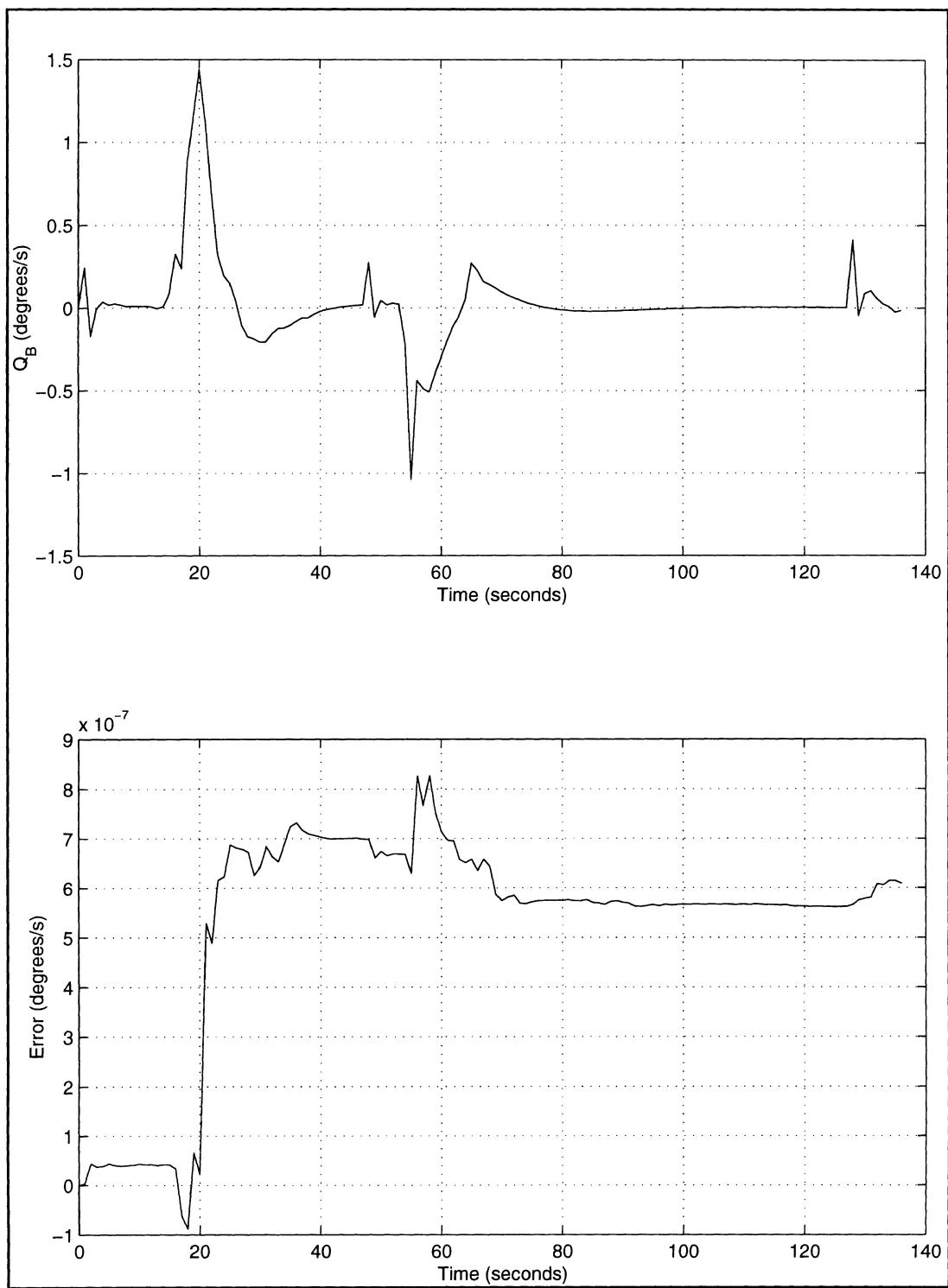


Figure 5.9 Time History of Pitch Rate (Body Axes)

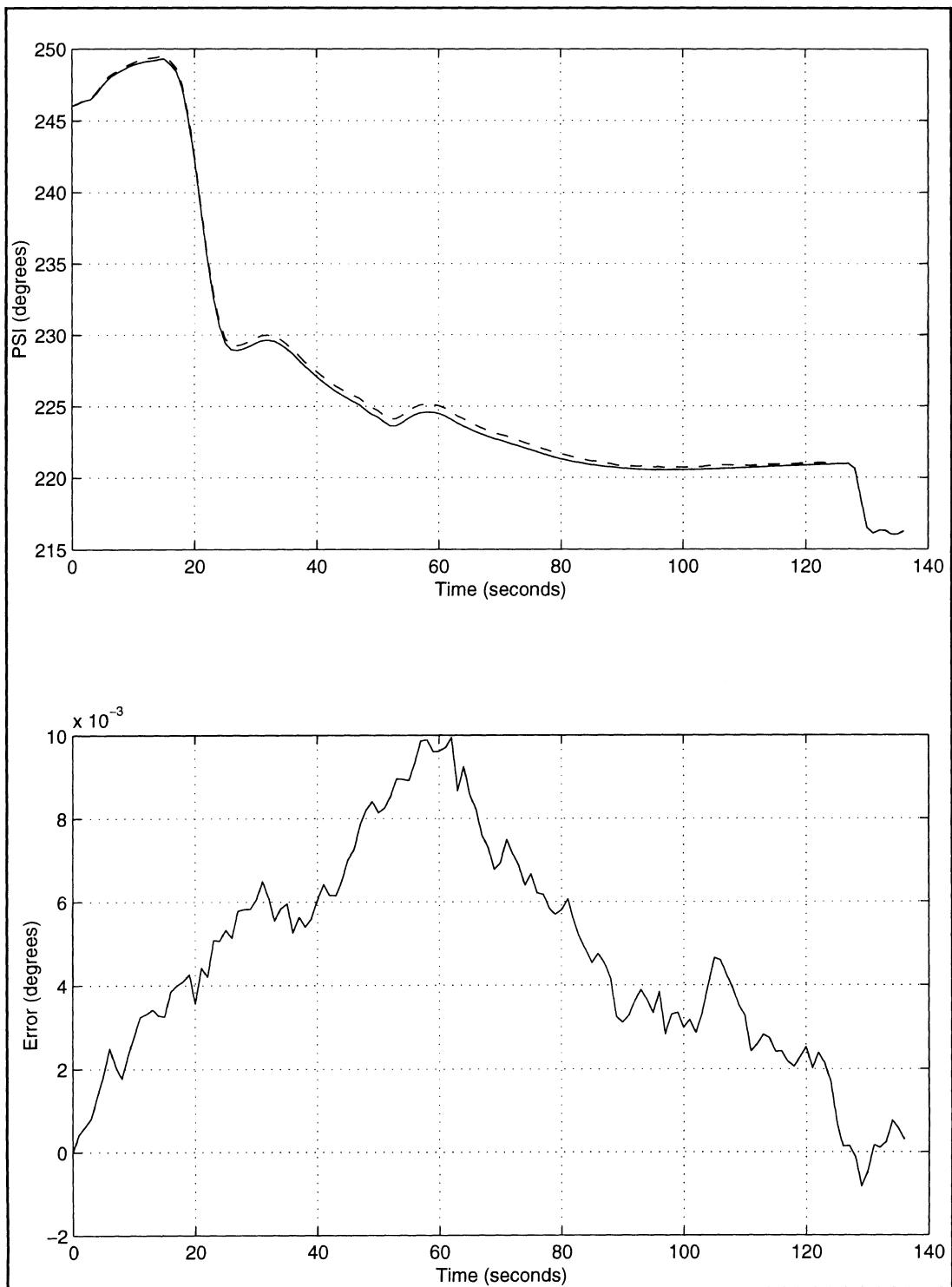


Figure 5.10 Time History of Heading

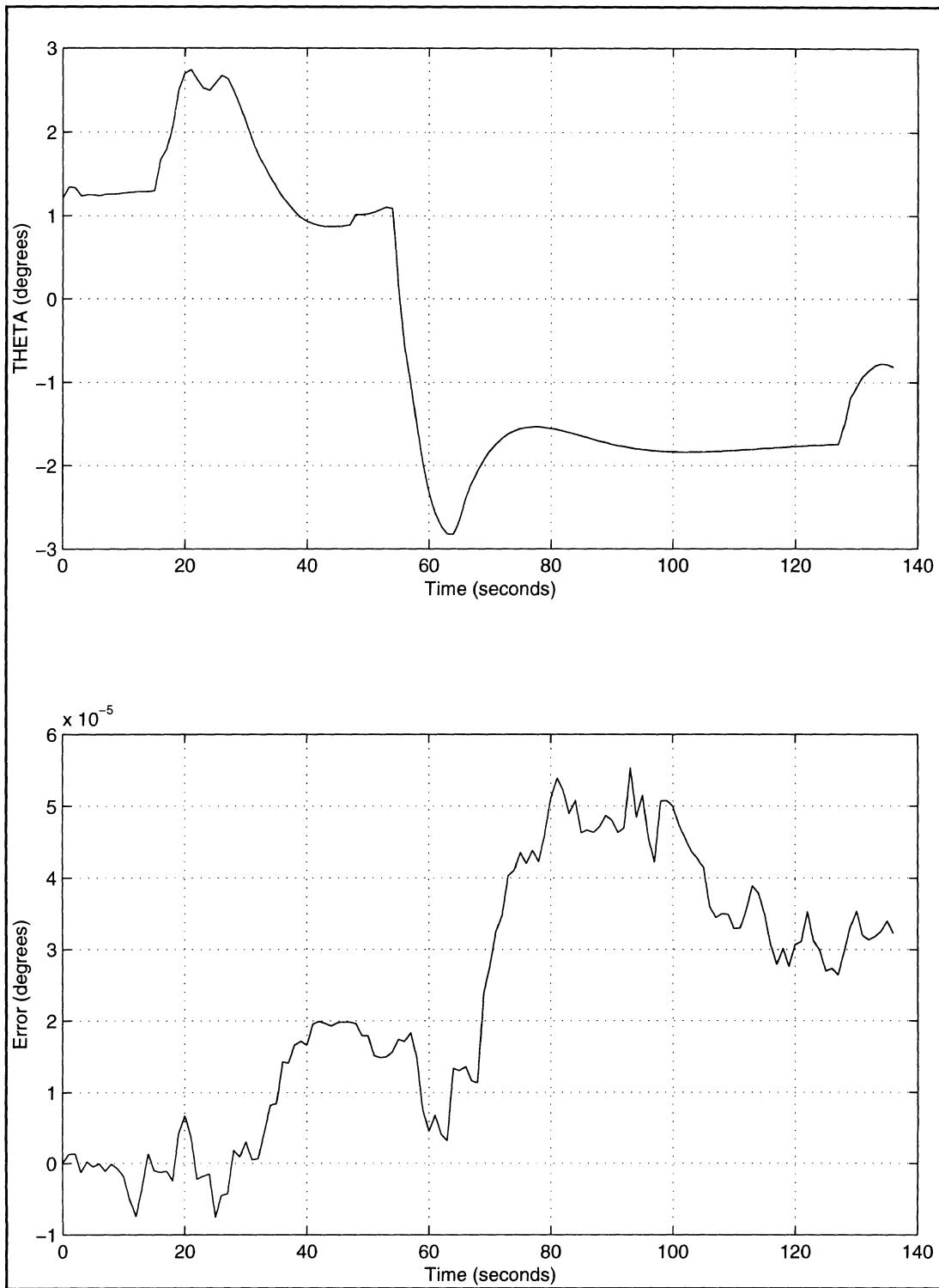


Figure 5.11 Time History of Pitch

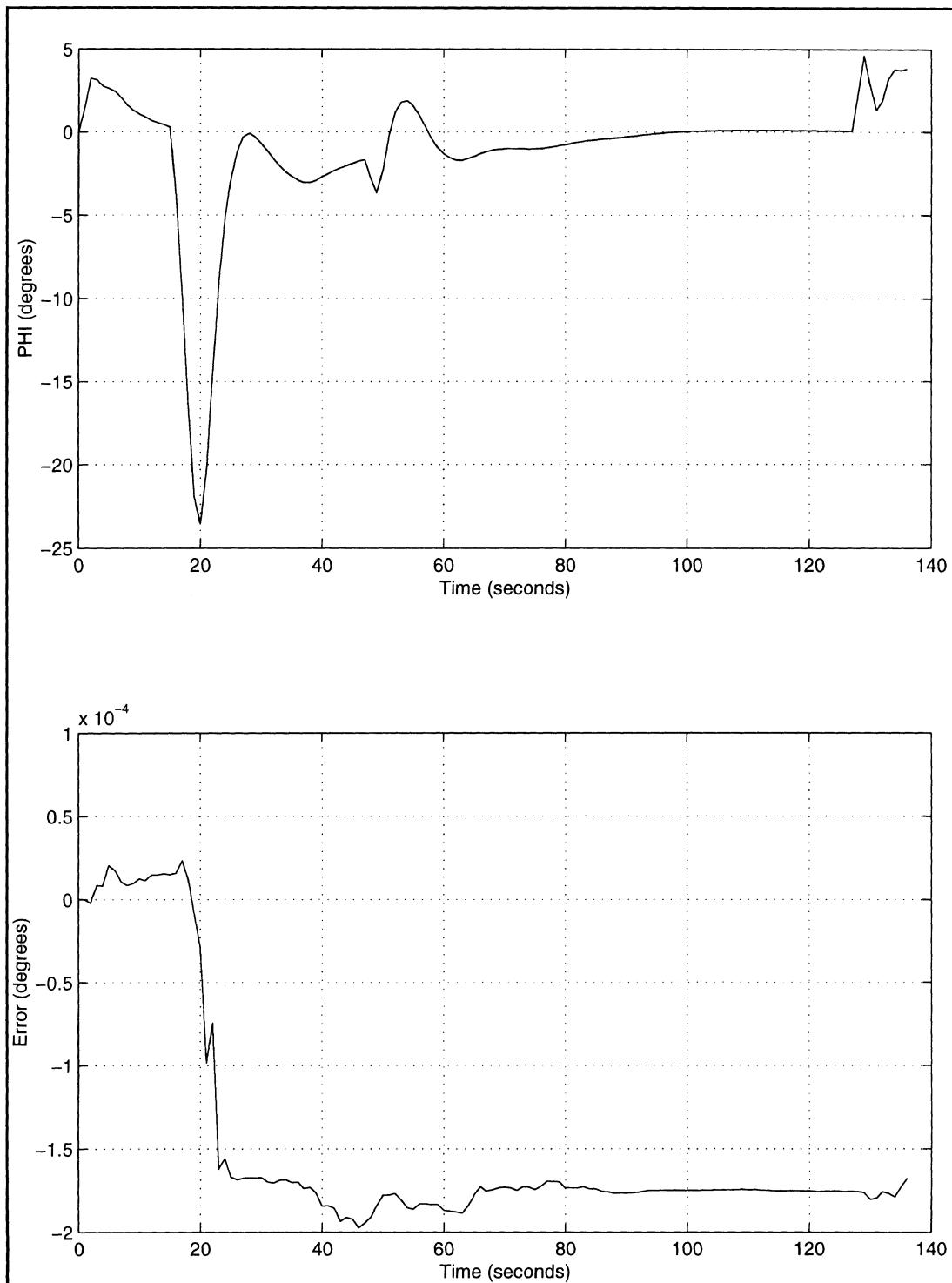


Figure 5.12 Time History of Roll

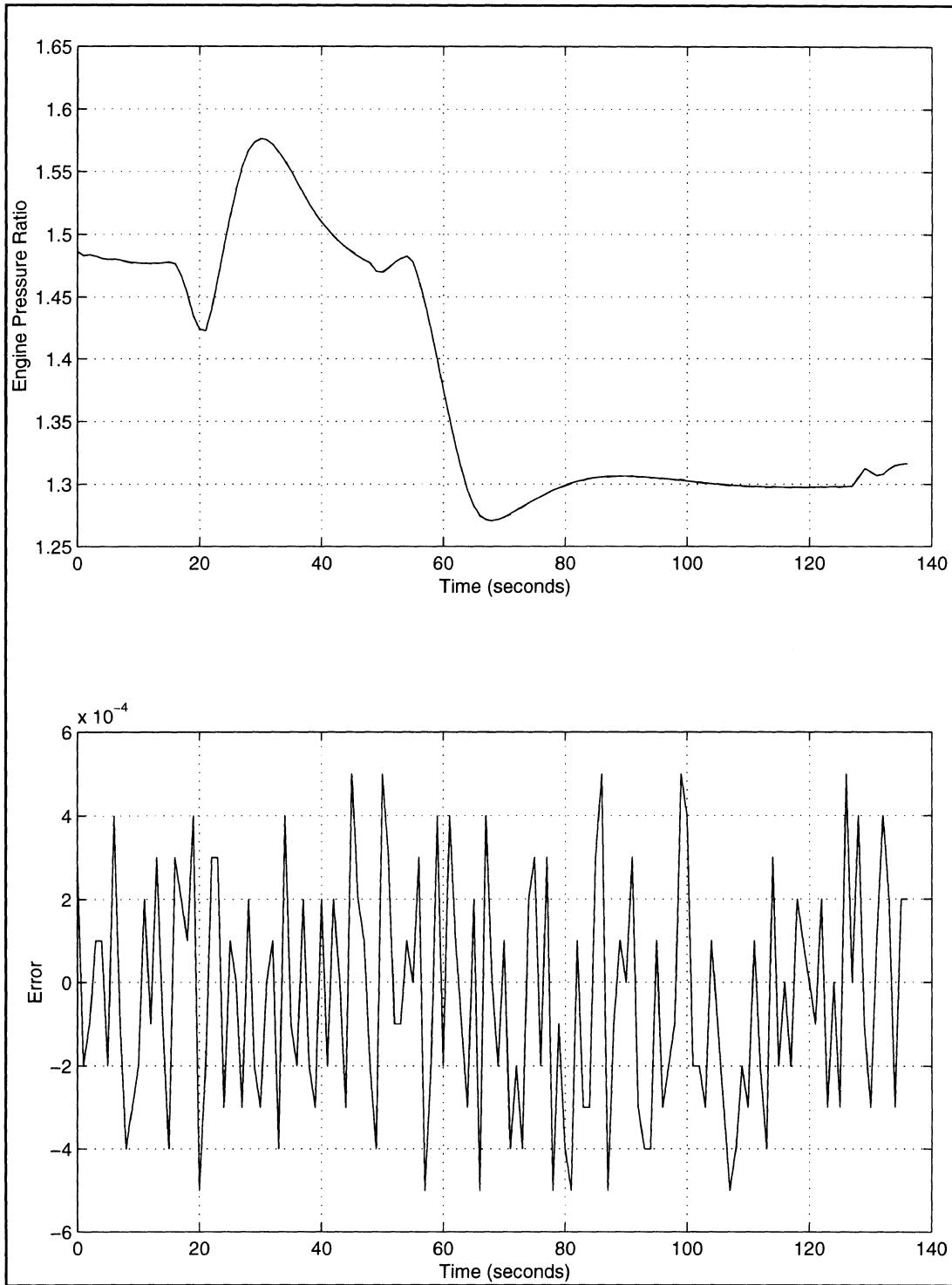


Figure 5.13 Time History of Engine Pressure Ratio

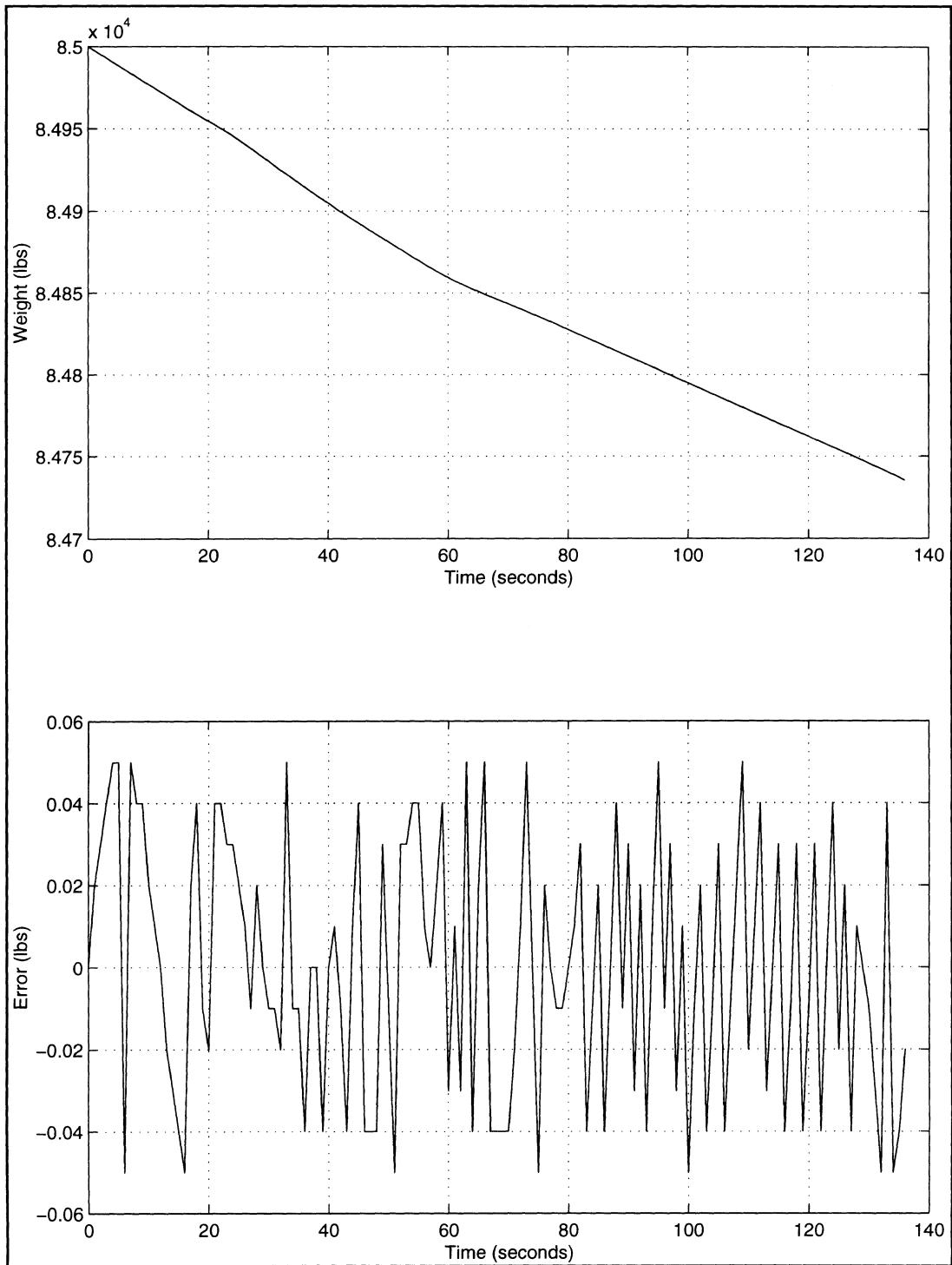


Figure 5.14 Time History of Weight

Chapter 6: CONCLUSIONS

This thesis presented a SIMULINK implementation of a verified FORTRAN twin jet engine transport aircraft simulation. In Chapter 1, the advantages of a software simulation environment for the design and testing of autoland control systems were presented. It was desired to implement the aircraft model in SIMULINK to take advantage of the tools that the environment provides for computing linearized models, trimming models, and interfacing to MATLAB. Also, since SIMULINK is a graphical environment, interfacing to the aircraft model from a control system and/or actuator model is straightforward. In Chapter 2, the coordinate systems and aircraft equations of motion were described. In Chapter 3, the aircraft dynamics model, which is implemented as a FORTRAN mex file, was described. In Chapter 4, the actuator models and the wind model, which are implemented as SIMULINK block diagrams, were presented.

In Chapter 5, results of a comparison between the FORTRAN and SIMULINK implementations of the aircraft simulation were presented. It is difficult to pinpoint the cause of the small differences between the simulations. It is my hypothesis that these minor differences were due to the different numeric environments in the respective simulations. The use of a single input set to verify the nonlinear simulation does have limitations. Since this is a nonlinear simulation, the verification process cannot hope to account for all

modes of operation of the aircraft. The particular set of data used did test the simulation in a high dynamic environment. This leads to the conclusion that the simulation is correctly implemented since the simulation matched the expected values in an environment where all equations of motion were stimulated. There may be simulation scenarios, however, that do not perform as well as the test case.

There are several areas for future research. The aircraft simulation presented in this thesis is intended to be a tool to aid in the design and evaluation of control systems and autoland sensors. Evaluating the benefits of using the global positioning system (GPS) as an autoland sensor is one area where the simulation could be useful. Implementing the sensor along with the aircraft simulation would provide the opportunity to evaluate the performance of the sensor during a wide variety of realistic landing scenarios. The GPS sensor would provide data that is not currently available to the autoland controller, including the distance to go on the glideslope and the velocity of the aircraft. This additional information could be used by a redesigned control system to improve on the current performance. The design of this new control system is another area for future research.

References

- Brown, Stuart C., Gordon H. Hardy and William S. Hindson. *A Flight-Test and Simulation Evaluation of the Longitudinal Final Approach and Landing Performance of an Automatic System for a Light Wing Loading STOL Aircraft.* Springfield, VA.: NASA Scientific and Technical Information Branch, 1983.
- Etkin, Bernard. *Dynamics of Atmospheric Flight.* New York: John Wiley & Sons, Inc., 1972.
- McRuer, Duane, Irving Ashkenas and Dustin Graham. *Aircraft Dynamics and Automatic Control.* Princeton, NJ: Princeton University Press, 1973.
- Miller, Dean, et al. "Development of Local Area Augmentation System Requirements." *Proceedings of the ION.* September 1996.
- Schiess, James R. and Christine G. Matthews. *Analysis and Monte-Carlo Simulation of Near-Terminal Aircraft Flight Paths.* Springfield, VA: NASA Scientific and Technical Information Branch, 1997.
- Stevens, Brian L. and Frank L. Lewis. *Aircraft Simulation and Control.* New York: John Wiley & Sons., Inc., 1992.

Appendix A: RUNNING THE AIRCRAFT SIMULATION

This chapter describes the hardware and software necessary to run the simulation. The system requirements are contained in Section 2.1. Section 2.2 provides a list of the files necessary to execute the simulation. Instructions for loading, executing, and verifying the simulation are presented in Section 3.3.

A.1 System Requirements

The following hardware and software was used to run the simulation.

HARDWARE: Sun Microsystems Sparc20

SOFTWARE: MATLAB by Mathworks version 4.2.1

SIMULINK by Mathworks version 1.4

F77 Sun FORTRAN compiler

A.2 List of Files and Directory Structure

The following is a list of the files in the 6 degree of freedom aircraft simulation as well as the directories where they should be located. Note that '~' may be any path that precedes the aircraft model directory structure.

~/aircraft

Makefile	acvalid.m
ac_sim.m	aerobrk.dat
aerofun.dat	atmosp_m.f
baseline_m.f	dynamics_m.f
filters_m.f	functional_m.f
getvars.m	getvars2.m
gnclaws_m.f	gout.dat

```
ionoparm.dat main_twin_jet.f  
namelist.dat output_m.f  
pathdef_m.f twin_j.out  
twin_j_autoland.out      twin_j_orig.out  
usercon.f     userintf_m.f
```

/CMN

```
aerobrk.cmn aerodata.cmn  
atmos_2.cmncaccel.cmn  
caeroc1.cmn caeroc2.cmn  
caerofm.cmn cengfm.cmn  
cfdata1.cmn cgust.cmn  
clngfm.cmn cmiscel.cmn  
cneng2.cmn cneweng.cmn  
cnlcmd.cmn cnlmeas.cmn  
cnlmode.cmncnstant.cmn  
cnstd2.cmn comfil.cmn  
csumfm.cmn ctrim.cmn  
cuseri.cmn cusero.cmn  
dgps.cmn dials.cmn  
dialslg.cmn dialslt.cmn  
estimat.cmn flpthc.cmn  
geardsc.cmn gsflare.cmn  
guiderr.cmn int_ac.cmn  
intcomm.cmn intintr.cmn  
magloc.cmn mlssysm.cmn  
nav aids.cmn navest.cmn  
numseg.cmn pathup.cmn  
pldial2.cmn plot1.cmn  
plotwpt.cmn prmcnl.cmn  
r2.cmn rodata.cmn  
seg3d.cmn shrdat.cmn  
tcable.cmn tevents.cmn  
waypts.cmn windsri.cmn  
wpt.cmn xform.cmn  
zaero fm.cmn
```

A.3 Executing the Simulation

First, copy all the files and subdirectories into a directory named "aircraft". This will be the directory from which all the following commands will be executed.

The first step in executing the simulation is to compile the FORTRAN-mex file. In order to do this, make sure that you are in the "aircraft" directory and type "make" on the command line. This will automatically link and compile the FORTRAN subroutines and generate the mex file.

Remaining in the "aircraft" directory, execute MATLAB by typing "matlab" at the command prompt. At the MATLAB prompt, type "ac_sim" to open the aircraft simulation. What you see is the top-level graphical diagram of the aircraft simulation. The actuators are on the left of the diagram and the aircraft dynamics model is to the right. Notice that the actuators require commanded control surface deflections as inputs and the aircraft model outputs the states of the aircraft dynamic model. Using these feedback signals and control inputs, the user may include a flight control system, landing control system, and/or sensor simulations for evaluation.

In order to evaluate whether the simulation is functioning properly, return to the MATLAB prompt. Type "acvalid" at the prompt to run the evaluation m-file. This m-file will load in the parameters of an autoland and compare the results with those of the original NASA FORTRAN simulation. If the time histories match within reasonable tolerances, then the simulation is

functioning correctly for this input set. These results may be compared to those in Chapter 5, where the results for the simulation on the platform detailed above are shown.

WOZNIAK, JASON, GERARD. MASTER OF SCIENCE. August, 1997
Electrical Engineering

Nonlinear Six Degree of Freedom Simulation of a Twin Jet Engine Transport Aircraft (80pp.)

Director of Thesis: Douglas A. Lawrence

(Keywords: aircraft, simulations, control systems, autoland)

Due to stringent requirements placed on current autoland systems for commercial aircraft, simulation has become an essential tool for the development and certification of autoland systems. This thesis presents a simulation designed to support autoland control system design for twin jet engine aircraft. The simulation runs in the SIMULINK graphical simulation environment in MATLAB. The SIMULINK environment enables the designer to implement sensor models and control system designs in block diagram form. This allows for the straightforward modification of feedback paths, gains, and functional blocks. The designer has access to aircraft states, actuator states, and environmental parameters for use in the development and testing of a control system or sensor design.

The simulation presented in this thesis was verified against an existing twin jet engine aircraft simulation maintained by the National Aeronautics and Space Administration (NASA). The test condition selected for the verification was an autoland scenario under high wind conditions that stimulated the aircraft dynamics. The guidance during the test condition was