Variable Stability Transfer Function Simulation

Henrik B. Pettersson

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute & State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Aerospace Engineering

Dr. Wayne Durham - chair

Dr. Frederick Lutze

Dr. Craig Woolsey

June 2002

Blacksburg, Virginia

Keywords: Variable Stability, Linearization, Transfer Functions, Flight Simulation,

Model Following

Copyright 2002, Henrik B. Pettersson

Variable Stability Transfer Function Simulation

Henrik B. Pettersson

(ABSTRACT)

Simulation, whether in-flight or ground-based, is an invaluable tool for testing and evaluating aircraft. Classically, a simulation model is specific to a single particular airframe, only able to model those flying characteristics. Vast information can be gained from a simulation that is able to model a wide range of aircraft, through a comparison of the performance of these aircraft.

Such a variable stability simulation model was created based on 46 stability parameters, including natural frequencies, damping ratios, time constants, and gains. The simulation was obtained using transfer functions representing the aircraft state responses to control inputs. These transfer functions were converted into state space systems used to create the linear equations for the model.

The model was first developed as a desktop simulation and then converted for use with the Virginia Tech's 2F122A flight simulator. This conversion required a simple dynamic inversion of the body axis force and moment terms. To reduce the error in these terms, a model following scheme was incorporated.

A series of canned inputs and real-time pilot-in-the-loop tests were flown to evaluate the variable stability model. Results in this paper have demonstrated the successful creation of a variable stability simulation model.

Acknowledgments

I would first like to thank my family for their support in all of my endeavors. To my parents, thank you for the guidance you have given me in life. You have instilled upon me the discipline and perseverance to succeed and the freedom and encouragement to let me become myself and strive to be the best at whatever I may pursue. To Fredrik, your undying support and constant striving to stay out of my shadow have made me a stronger person. And to Lisa, thank you for being the most wonderful person you are, stay sweet and never change.

I would like to sincerely thank my advisor, Dr. Wayne Durham, without whom this would not have been possible. He has motivated me with his seemingly endless enthusiasm for and real-life experience from the aerospace field. Thank you for not only being persistent and patient with me academically as an advisor, but for being a friend too. To the remaining members of my committee, Dr. Frederick Lutze and Dr. Craig Woolsey, thank you for sharing your knowledge and advice on this subject with me.

Last but not least, I would like to thank my many friends who make life what it is.

A few in particular in reference to this thesis: To Roger Beck, for the patience to put up with my incessant questions; to Bill Oetjens, for keeping everything running in the Sim Lab; to Dan Hart, for his expert proof reading and test piloting; to Josh Durham, for fixing any computer problems I may have caused; and to Mark Nelson, for helping me with this thesis. More importantly, thanks to you and all my other friends for being just that, friends, you mean a lot to me.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms:

CASTLE	Control Analysis and Simulation Test Loop Environment
PIO	Pilot Induced Oscillation
VISTA	Variable stability In-flight Simulator Test Aircraft

Aircraft Variables:

V	Wind-axis velocity
α	Angle of attack
β	Sideslip angle
u	Body x-axis velocity
v	Body y-axis velocity
w	Body z-axis velocity
ϕ	Euler roll angle
θ	Euler pitch angle
ψ	Euler yaw angle

P	Body-axis roll rate
Q	Body-axis pitch rate
R	Body-axis yaw rate
X	Body x-axis force
Y	Body y-axis force
Z	Body z-axis force
L	Body-axis rolling moment
M	Body-axis pitching moment
N	Body-axis yawing moment
I_{xx}	x-axis moment of inertia
I_{yy}	y-axis moment of inertia
I_{zz}	z-axis moment of inertia
I_{xz}	x-z product of inertia
δ_{th}	Throttle setting
δ_e	Elevator deflection
δ_a	Aileron deflection
δ_r	Rudder deflection

Transfer Function Variables:

$K_{\alpha_{\delta e}}$	Gain for $\delta\alpha$ to δe transfer function
$K_{\alpha_{\delta th}}$	Gain for $\delta\alpha$ to δth transfer function

$K_{\beta_{\delta a}}$	Gain for $\delta\beta$ to δa transfer function
$K_{\beta_{\delta r}}$	Gain for $\delta\beta$ to δr transfer function
$K_{\phi_{\delta a}}$	Gain for $\delta\phi$ to δa transfer function
$K_{\phi_{\delta r}}$	Gain for $\delta\phi$ to δr transfer function
$K_{\psi_{\delta a}}$	Gain for $\delta\psi$ to δa transfer function
$K_{\psi_{\delta r}}$	Gain for $\delta\psi$ to δr transfer function
$K_{\theta_{\delta e}}$	Gain for $\delta\theta$ to δe transfer function
$K_{\theta_{\delta th}}$	Gain for $\delta\theta$ to δth transfer function
$K_{u_{\delta e}}$	Gain for δu to δe transfer function
$K_{u_{\delta th}}$	Gain for δu to δth transfer function
$\omega_{n_{sp}}$	Short period natural frequency
ω_{n_p}	Phugoid natural frequency
ω_{n_d}	Dutch roll natural frequency
ω_{n_α}	Natural frequency for $\delta\alpha$ to δe transfer function
$\omega_{n_{\phi_{\delta a}}}$	Natural frequency for $\delta\phi$ to δa transfer function
$\omega_{n_{\psi_{\delta a}}}$	Natural frequency for $\delta\psi$ to δa transfer function
$\omega_{n_{\psi_{\delta r}}}$	Natural frequency for $\delta\psi$ to δr transfer function
ω_{n_u}	Natural frequency for δu to δe transfer function
$\omega_{n_{u_{\delta th}}}$	Natural frequency for δu to δth transfer function
T_r	Roll mode time constant
T_s	Spiral mode time constant

T_{α}	Time constant for $\delta\alpha$ to δe transfer function
$T_{\alpha_{\delta th}}$	Time constant for $\delta\alpha$ to δth transfer function
$T_{\beta_{\delta a1}}$	1st time constant for $\delta\beta$ to δa transfer function
$T_{\beta_{\delta a2}}$	2nd time constant for $\delta\beta$ to δa transfer function
$T_{\beta_{\delta r1}}$	1st time constant for $\delta\beta$ to δr transfer function
$T_{\beta_{\delta r2}}$	2nd time constant for $\delta\beta$ to δr transfer function
$T_{\beta_{\delta r3}}$	3rd time constant for $\delta\beta$ to δr transfer function
$T_{\phi_{\delta r1}}$	1st time constant for $\delta\phi$ to δr transfer function
$T_{\phi_{\delta r2}}$	2nd time constant for $\delta\phi$ to δr transfer function
$T_{\psi_{\delta a}}$	Time constant for $\delta\psi$ to δa transfer function
$T_{\psi_{\delta r}}$	Time constant for $\delta\psi$ to δr transfer function
$T_{\theta 1}$	1st time constant for $\delta\theta$ to δe transfer function
$T_{\theta 2}$	2nd time constant for $\delta\theta$ to δe transfer function
T_{u}	Time constant for δu to δe transfer function
ζ_{d}	Dutch roll damping ratio
ζ_{p}	Phugoid damping ratio
ζ_{sp}	Short period damping ratio
$\zeta_{\phi_{\delta a}}$	Damping ratio for $\delta\phi$ to δa transfer function
$\zeta_{\psi_{\delta a}}$	Damping ratio for $\delta\psi$ to δa transfer function
$\zeta_{\psi_{\delta r}}$	Damping ratio for $\delta\psi$ to δr transfer function
ζ_{u}	Damping ratio for δu to δe transfer function

$\zeta_{u_{\delta th}}$ Damping ratio for δu to δth transfer function

Model Following Variables:

x_m Model variable

x_p Plant variable

λ_u Gain for $\dot{u}$ error term

λ_v Gain for $\dot{v}$ error term

λ_w Gain for $\dot{w}$ error term

λ_p Gain for $\dot{p}$ error term

λ_q Gain for $\dot{q}$ error term

λ_r Gain for $\dot{r}$ error term

Chapter 1

Introduction

1.1 Background

Simulation, whether in-flight or ground-based, is an invaluable tool for testing and evaluating aircraft. Classically, a simulation model is specific to a single particular airframe, only able to model those flying characteristics. Vast information can be gained from a simulation that is able to model a wide range of aircraft, through a comparison of the performance of these aircraft. One such in-flight simulator is the NF-16 VISTA (Variable stability In-flight Simulator Test Aircraft). It is capable of simulating the flying characteristics of aircraft designed for high-g, high altitude, and high speed missions. The scope of this thesis is to develop a ground based simulator for doing research in parallel with the VISTA.

A variable stability ground-based simulator model can be much more versatile, as it

is not dependent upon the aerodynamics of the base airframe. For instance, rather than only being able to model aircraft physically similar to the F-16, like the VISTA, a ground-based simulator has the ability to model a wide range of aircraft from small gliders, to highly maneuverable fighters like the F-16, to large airliners like the Boeing 747, by only changing certain parameters. In addition, such a simulator can be used in design by slightly altering the flying characteristics of any existing aircraft to test the resulting behavior.

While it may seem like a daunting task incorporating the aerodynamics of so many aircraft into a simulation, there are efficient ways to accomplish this, like eliminating the aerodynamic terms completely. One such method of eliminating the aerodynamics is to program the simulation using parameters such as time constants, damping ratios, and natural frequencies, from which changing the flying qualities of an aircraft would become trivial. The states of the aircraft are derived from a linear combination of these parameters and controls. The resulting aircraft forces and moments can be computed from these states through a simple dynamic inversion. These forces and moments had to be calculated for the integration of the simulation model with the *CASTLE* structure for use with a real-time simulator. This gives the ability to change such parameters as the short period frequency or the phugoid damping by directly changing that value in the aircraft model.

To test the fidelity of this linear simulation, an accurate method of simulation validation is needed. The linear transfer function model was compared with a full non-linear model exhibiting the desired flight characteristics through batch inputs. Real-time testing of the transfer function model for varying stability parameters was performed by a test pilot

for a given task. The handling qualities ratings of the model were assessed by pilot opinion ratings.

It must be noted that this linear variable stability model was obtained through linearizing the full system about some reference conditions. Therefore, the small angle assumption applies and the linear model is assumed to be valid for flight conditions close to the reference conditions. This lead to a choice of pilot tasks that did not entail any large excursions from the reference conditions.

This thesis is divided into four main sections as follows:

Chapter 2 briefly discusses the mathematics involved in creating the variable stability transfer function simulation.

Chapter 3 introduces the simulation environment used for this research. It then discusses the non-linear F-4 model used as a base comparison aircraft. Finally, the development and implementation of the variable stability transfer function computer simulation model is explained.

Chapter 4 discusses the methods that will be used to evaluate the fidelity of the variable stability simulation model. First, handling quality ratings and their application to this research are discussed. Then, ground-based versus in-flight simulation are compared. The test pilot is introduced in the following section. Finally, the representative maneuvers used for evaluation and their results are discussed.

Chapter 5 contains the summary of the work and the conclusions from the research.

1.2 Research Objectives

The objective of this thesis is to investigate the implementation of a linear variable stability transfer function simulation model. For the testing, the linear model will use such parameters as time constants, damping ratios, and natural frequencies obtained from linearizing a full non-linear model of the F-4 Phantom about reference conditions. This non-linear model will be used as a point of comparison for the transfer function model through various batch inputs.

To demonstrate the variable stability capabilities of the transfer function model, real-time pilot-in-the-loop simulation of the model will be completed for a given task with varying stability parameters. The performance of the model will be assessed through pilot opinion ratings and hard data gathered from the simulation. The validity of the linear variable stability model will be determined from these results.

1.3 NF-16D VISTA

The inspiration for the creation of such a ground-based variable stability simulator was the NF-16D Variable stability In-flight Simulator Test Aircraft (VISTA) pictured in Figure 1.1 [1]. It was developed jointly by the United States Air Force Research Laboratory (AFRL) and Veridian Engineering Flight Research Group (formerly known as Calspan) as a high performance in-flight simulator based on a highly modified F-16D.

Figure 1.1: NF-16D Variable stability In-flight Simulator Test Aircraft (VISTA)

The VISTA NF-16D aircraft provides flight research and in-flight simulation capabilities for advanced aircraft and flight control concepts. Using flight computers to incorporate a variable stability system into the fly-by-wire system of the F-16 and increasing controllability with control surfaces and an Axisymmetric Vectoring Exhaust Nozzle (AVEN), the VISTA is capable of emulating the flying characteristics of a wide range of aircraft that have high-g, high altitude, and high speed missions, like the F-14, F-15, F-22, or Joint Strike Fighter (JSF). A summary of the VISTA's flight envelope is shown in Table 1.1 [2]. While it is able to model a range of modern day fighter type aircraft, the flight envelope of the VISTA is fairly limited due to the aerodynamics of the base airframe. A variable stability ground-based simulator is much more versatile, as it has the ability to mimic the flying qualities of any aircraft.

Table 1.1: Flight envelope of the NF-16D VISTA.

Maximum airspeed:	440 KIAS/0.9 Mach
Angle-of-attack limits:	-10° to $+16^\circ$
Normal load factor limits:	-2.4 to $+6.8$ gs
Maximum sideslip:	$\pm 10^\circ$

Chapter 2

Linearization & Transfer Functions

2.1 Introduction

This chapter briefly discusses the mathematics involved in creating this variable stability transfer function simulation.

2.2 Linearization

The simulation created here is a linear transfer function simulation. There are twelve non-linear, ordinary differential equations that comprise the equations of motion for an aircraft. These twelve equations may be solved for the twelve states: the body axis force equations (u, v, w), the Euler angle kinematic equations (ϕ, θ, ψ), the body axis moment equations

(p, q, r), and the navigation states (x_E, y_E, h). They are generally as follows:

$$\dot{u} = rv - qw - g\sin\theta + \frac{X}{m} \tag{2.1}$$

$$\dot{v} = -ru + pw + g\sin\phi\cos\theta + \frac{Y}{m} \tag{2.2}$$

$$\dot{w} = qu - pv + g\cos\phi\cos\theta + \frac{Z}{m} \tag{2.3}$$

$$\dot{\phi} = p + (q\sin\phi + r\cos\phi)\tan\theta \tag{2.4}$$

$$\dot{\theta} = q\cos\phi - r\sin\phi \tag{2.5}$$

$$\dot{\psi} = \frac{q\sin\phi + r\cos\phi}{\cos\theta} \tag{2.6}$$

$$\begin{aligned}\dot{p} &= \frac{I_{zz}}{I_{xx}I_{zz} - I_{xz}^2}[L + I_{xz}pq - (I_{zz} - I_{yy})qr] \\ &\quad + \frac{I_{xz}}{I_{xx}I_{zz} - I_{xz}^2}[N - I_{xz}qr - (I_{yy} - I_{xx})pq]\end{aligned} \tag{2.7}$$

$$\dot{q} = \frac{1}{I_{yy}}[M + M_T - (I_{xx} - I_{zz})pr - I_{xz}(p^2 - r^2)] \tag{2.8}$$

$$\begin{aligned}\dot{r} &= \frac{I_{xz}}{I_{xx}I_{zz} - I_{xz}^2}[L + I_{xz}pq - (I_{zz} - I_{yy})qr] \\ &\quad + \frac{I_{xx}}{I_{xx}I_{zz} - I_{xz}^2}[N - I_{xz}qr - (I_{yy} - I_{xx})pq]\end{aligned} \tag{2.9}$$

$$\begin{aligned} \dot{x_E} &= u(\cos\theta\cos\psi) + v(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi) \\ &\quad + w(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi) \qquad (2.10) \\ \dot{y_E} &= u(\cos\theta\sin\psi) + v(\sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi) \\ &\quad + w(\cos\phi\sin\theta\sin\psi + \sin\phi\cos\psi) \qquad (2.11) \\ \dot{h} &= u\sin\theta - v\sin\phi\cos\theta - w\cos\phi\cos\theta. \qquad (2.12) \end{aligned}$$

Numerical integration can be used to solve this set of equations, where the differential equations are sequentially calculated in discrete time steps. The differential $\frac{dx}{dt}$ is approximated by $\frac{\Delta x}{\Delta t}$ and the change Δx is calculated over the time step Δt. This is a particular method of numerical integration known as a first order Euler integration. Numerical integration is the basis for real-time flight simulation. For given initial conditions and control inputs, the time history of the aircraft motion can be generated to analyze and characterize the response.

Another method for approximating the solution to these non-linear ordinary differential equations is to first linearize them by considering only small perturbations about some reference flight condition. A common way to achieve this linearization is with a Taylor series expansion about reference conditions through first order terms, as shown below:

$$f(x_1, x_2, ..., x_n) \approx f(x_1, x_2, ..., x_n)_{ref} + \frac{\partial f}{\partial x_1}|_{ref}\Delta x_1 + \frac{\partial f}{\partial x_2}|_{ref}\Delta x_2 + ... + \frac{\partial f}{\partial x_n}|_{ref}\Delta x_n \qquad (2.13)$$

This results in twelve linear ordinary differential equations, which can be solved an-

alytically. As stated above, these equations arise from linearizing equations about some reference flight conditions. Therefore, this linearized set of equations is not applicable for large control inputs or large excursions from this condition. [3]

The linear equations needed for this simulation are derived by numerical methods from the nonlinear state models. In the small perturbation linear equations, the nonlinear aerodynamic coefficients are replaced by terms involving the stability derivatives, as shown subsequently. Therefore, the linear equations provide great insight into the relative importance of the various stability derivatives under different flight conditions. And thus, the linearized system of ordinary differential equations used to model the dynamics of the aircraft for this study has the advantage of insight into the behavior of flight variables but the drawback of only being valid for small perturbations about the reference flight condition.

2.3 Transfer Functions

The model used for this simulation is based on transfer functions, and these transfer functions arise from the Laplace transform solutions to the ordinary differential equations that model the dynamics of the aircraft.

The Laplace transform method can be used in solving linear, ordinary differential equations. In the Laplace domain, differentiation of the time function corresponds to multiplication of the transform by a complex transform variable s, and thus the differential equations in the time domain become algebraic equations in the Laplace domain. Beginning

with the state equations:

$$\dot{X} = AX + BU \tag{2.14}$$

$$Y = CX + DU \tag{2.15}$$

with X representing the state vector, U the control vector, Y the output vector, and Jacobian coefficient matrices A, B, C, and D. Laplace transformations of equations 13 and 14 give:

$$sX(s) - x(0) = AX(s) + BU(s) \tag{2.16}$$

or

$$X(s) = (sI - A)^{-1}[x(0) + BU(s)] \tag{2.17}$$

for the state vector, and:

$$Y(s) = C(sI - A)^{-1}[x(0) + BU(s)] + DU(s) \tag{2.18}$$

for the output vector. The matrix that relates the system outputs, $Y(s)$, to the inputs, $U(s)$, with zero initial conditions is the transfer function matrix, $G(s)$. This matrix is a multi-input, multi-output (MIMO) transfer function:

$$G(s) = C(sI - A)^{-1}B + D. \tag{2.19}$$

Using the property that a matrix inverse can be expressed as the adjoint matrix divided by the determinant gives:

$$G(s) = \frac{Cadj(sI - A)B + D|sI - A|}{|sI - A|}. \tag{2.20}$$

The single-input, single-output (SISO) transfer function from the jth input to the ith output is given as:

$$g_{ij}(s) = \frac{c_i adj(sI - A)b_j + d_{ij}|sI - A|}{|sI - A|}. \tag{2.21}$$

The poles of a transfer function are positions at which the magnitude of the transfer function becomes infinite, and are found when the denominator of the transfer function equals zero. The roots or zeros occur when the numerator equals zero, and are where the magnitude of the transfer function is zero. [4]

If the transfer function of a system is known, the output can be studied for various inputs with a view toward understanding the nature of the system. [5]

A series of transfer functions can be obtained which relate the output of a system to the input solely based on parameters such as time constants, natural frequencies, and damping ratios. The transfer functions require linearized state equations. For these linearizations, the reference conditions are steady, straight, symmetric flight. It should be noted that linear dynamics decouple the longitudinal and lateral-directional terms. These linearizations result in the following longitudinal small perturbation equations in terms of the dimensional stability derivatives:

$$\dot{u} = -g(\cos\theta_{ref})\theta + X_u u + X_\alpha \alpha + X_{\delta e}\delta e \tag{2.22}$$

$$\dot{w} = V_{ref} q - g(\sin\theta_{ref})\theta + Z_u u + Z_\alpha \alpha + Z_{\dot{\alpha}}\dot{\alpha} + Z_q q + Z_{\delta e}\delta e \tag{2.23}$$

$$\dot{q} = M_u u + M_\alpha \alpha + M_{\dot{\alpha}}\dot{\alpha} + M_q q + M_{\delta e}\delta e \tag{2.24}$$

In the notation used here by Hewett [6], U represents the true airspeed V_T. Taking the Laplace transform of each of these equations, setting the initial conditions to zero, and solving for the transfer functions gives the matrix equation:

$$\begin{bmatrix} s - X_u & -X_\alpha & g\cos\theta_{ref} \\ -Z_u & s(V_{ref} - Z_{\dot{\alpha}}) - Z_\alpha & -(Z_q + V_{ref})s + g\sin\theta_{ref} \\ -M_u & -(M_{\dot{\alpha}}s + M_\alpha) & s^2 - M_q s \end{bmatrix} \begin{bmatrix} \frac{U(s)}{\delta e(s)} \\ \frac{\alpha(s)}{\delta e(s)} \\ \frac{\theta(s)}{\delta e(s)} \end{bmatrix} = \begin{bmatrix} X_{\delta e} \\ Z_{\delta e} \\ M_{\delta e} \end{bmatrix} \tag{2.25}$$

To solve for the SISO longitudinal transfer functions $\frac{U(s)}{\delta e(s)}$, $\frac{\alpha(s)}{\delta e(s)}$, and $\frac{\theta(s)}{\delta e(s)}$, Cramer's rule is applied to get:

$$\frac{U(s)}{\delta e(s)} = \frac{\begin{vmatrix} X_{\delta e} & -X_\alpha & g\cos\theta_{ref} \\ Z_{\delta e} & s(V_{ref} - Z_{\dot{\alpha}}) - Z_\alpha & -(Z_q + V_{ref})s + g\sin\theta_{ref} \\ M_{\delta e} & -(M_{\dot{\alpha}}s + M_\alpha) & s^2 - M_q s \end{vmatrix}}{\begin{vmatrix} s - X_u & -X_\alpha & g\cos\theta_{ref} \\ -Z_u & s(V_{ref} - Z_{\dot{\alpha}}) - Z_\alpha & -(Z_q + V_{ref})s + g\sin\theta_{ref} \\ -M_u & -(M_{\dot{\alpha}}s + M_\alpha) & s^2 - M_q s \end{vmatrix}} \tag{2.26}$$

$$\frac{\alpha(s)}{\delta e(s)} = \frac{\begin{vmatrix} s - X_u & X_{\delta e} & g\cos\theta_{ref} \\ -Z_u & Z_{\delta e} & -(Z_q + V_{ref})s + g\sin\theta_{ref} \\ -M_u & M_{\delta e} & s^2 - M_q s \end{vmatrix}}{\begin{vmatrix} s - X_u & -X_\alpha & g\cos\theta_{ref} \\ -Z_u & s(V_{ref} - Z_{\dot{\alpha}}) - Z_\alpha & -(Z_q + V_{ref})s + g\sin\theta_{ref} \\ -M_u & -(M_{\dot{\alpha}}s + M_\alpha) & s^2 - M_q s \end{vmatrix}} \tag{2.27}$$

$$\frac{\theta(s)}{\delta e(s)} = \frac{\begin{vmatrix} s - X_u & -X_\alpha & X_{\delta e} \\ -Z_u & s(V_{ref} - Z_{\dot{\alpha}}) - Z_\alpha & Z_{\delta e} \\ -M_u & -(M_{\dot{\alpha}}s + M_\alpha) & M_{\delta e} \end{vmatrix}}{\begin{vmatrix} s - X_u & -X_\alpha & g\cos\theta_{ref} \\ -Z_u & s(V_{ref} - Z_{\dot{\alpha}}) - Z_\alpha & -(Z_q + V_{ref})s + g\sin\theta_{ref} \\ -M_u & -(M_{\dot{\alpha}}s + M_\alpha) & s^2 - M_q s \end{vmatrix}} \tag{2.28}$$

Simplifying and rewriting these transfer functions in terms of time constants (T), damping ratios (ζ), natural frequencies (ω_n), and gains (K) results in the following for the longitudinal states in terms of elevator input (δe):

$$\frac{U(s)}{\delta e(s)} = \frac{K_{u_{\delta e}}(s + \frac{1}{T_u})(s^2 + 2\zeta_u\omega_{n_u}s + \omega_{n_u}^2)}{(s^2 + 2\zeta_{sp}\omega_{n_{sp}}s + \omega_{n_{sp}}^2)(s^2 + 2\zeta_p\omega_{n_p}s + \omega_{n_p}^2)} \tag{2.29}$$

$$\frac{\alpha(s)}{\delta e(s)} = \frac{K_{\alpha_{\delta e}}(s + \frac{1}{T_\alpha})(s^2 + 2\zeta_\alpha\omega_{n_\alpha}s + \omega_{n_\alpha}^2)}{(s^2 + 2\zeta_{sp}\omega_{n_{sp}}s + \omega_{n_{sp}}^2)(s^2 + 2\zeta_p\omega_{n_p}s + \omega_{n_p}^2)} \tag{2.30}$$

$$\frac{\theta(s)}{\delta e(s)} = \frac{K_{\theta_{\delta e}}(s+\frac{1}{T_{\theta_1}})(s+\frac{1}{T_{\theta_2}})}{(s^2+2\zeta_{sp}\omega_{n_{sp}}s+\omega_{n_{sp}}^2)(s^2+2\zeta_p\omega_{n_p}s+\omega_{n_p}^2)} \tag{2.31}$$

This gives equations for three of the four longitudinal states, U, α, and θ, leaving q unknown. An equation for q can be generated through the linear relation:

$$\dot{\theta} = q \tag{2.32}$$

Similarly, the following are the lateral-directional small perturbation equations in terms of the dimensional stability derivatives:

$$\dot{v} = -V_{ref}r + g(\cos\theta_{ref})\phi + Y_\beta\beta + Y_p p + Y_r r + Y_{\delta a}\delta a + Y_{\delta r}\delta r \tag{2.33}$$

$$\dot{p} - \frac{I_{xz}}{I_{xx}}\dot{r} = L_\beta\beta + L_p p + L_r r + L_{\delta a}\delta a + L_{\delta r}\delta r \tag{2.34}$$

$$\dot{r} - \frac{I_{xz}}{I_{zz}}\dot{p} = N_\beta\beta + N_{T_\beta}\beta + N_p p + N_r r + N_{\delta a}\delta a + N_{\delta r}\delta r \tag{2.35}$$

$$\tag{2.36}$$

This leads to the following lateral-directional transfer functions, for aileron (δa) and rudder input (δr), respectively:

$$\frac{\beta(s)}{\delta a(s)} = \frac{K_{\beta_{\delta a}}(s+\frac{1}{T_{\beta_{a_1}}})(s+\frac{1}{T_{\beta_{a_2}}})}{(s+\frac{1}{T_s})(s+\frac{1}{T_R})(s^2+2\zeta_D\omega_{n_D}s+\omega_{n_D}^2)} \tag{2.37}$$

$$\frac{\phi(s)}{\delta a(s)} = \frac{K_{\phi_{\delta a}}(s^2+2\zeta_{\phi_a}\omega_{n_{\phi_a}}s+\omega_{n_{\phi_a}}^2)}{(s+\frac{1}{T_s})(s+\frac{1}{T_R})(s^2+2\zeta_D\omega_{n_D}s+\omega_{n_D}^2)} \tag{2.38}$$

$$\frac{\psi(s)}{\delta a(s)} = \frac{K_{\psi_{\delta a}}(s+\frac{1}{T_{\psi_a}})(s^2+2\zeta_{\psi_a}\omega_{n_{\psi_a}}s+\omega_{n_{\psi_a}}^2)}{s(s+\frac{1}{T_s})(s+\frac{1}{T_R})(s^2+2\zeta_D\omega_{n_D}s+\omega_{n_D}^2)} \tag{2.39}$$

$$\frac{\beta(s)}{\delta r(s)} = \frac{K_{\beta_{\delta r}}(s+\frac{1}{T_{\beta_{r_1}}})(s+\frac{1}{T_{\beta_{r_2}}})(s+\frac{1}{T_{\beta_{r_3}}})}{(s+\frac{1}{T_s})(s+\frac{1}{T_R})(s^2+2\zeta_D\omega_{n_D}s+\omega_{n_D}^2)} \tag{2.40}$$

$$\frac{\phi(s)}{\delta r(s)} = \frac{K_{\phi_{\delta r}}(s+\frac{1}{T_{\phi_{r_1}}})(s+\frac{1}{T_{\phi_{r_2}}})}{(s+\frac{1}{T_s})(s+\frac{1}{T_R})(s^2+2\zeta_D\omega_{n_D}s+\omega_{n_D}^2)} \tag{2.41}$$

$$\frac{\psi(s)}{\delta r(s)} = \frac{K_{\psi_{\delta r}}(s+\frac{1}{T_{\psi_r}})(s^2+2\zeta_{\psi_r}\omega_{n_{\psi_r}}s+\omega_{n_{\psi_r}}^2)}{s(s+\frac{1}{T_s})(s+\frac{1}{T_R})(s^2+2\zeta_D\omega_{n_D}s+\omega_{n_D}^2)} \tag{2.42}$$

There are equations here for three of the five lateral-directional states, β, ϕ, and ψ, leaving p and r unknown. As in the longitudinal case, these can be solved through the linear relationships:

$$\dot{\phi} = p \tag{2.43}$$

and

$$\dot{\psi} = r \tag{2.44}$$

From these transfer function equations, the aircraft states can be related to the control inputs by the stability parameters alone. And thus, parameters such as the short period frequency or the phugoid damping can be altered by directly changing that value in the aircraft simulation. [6]

The above transfer functions assume a certain form, like one real root and a complex conjugate pair in the numerator of equation 2.29. If a different form is required, three real roots for example, then the appropriate changes can be made to the transfer function my changing stability parameters resulting in equation 2.43 below:

$$\frac{U(s)}{\delta e(s)} = \frac{K_{u_{\delta e}}(s+\frac{1}{T_{u1}})(s+\frac{1}{T_{u2}})(s+\frac{1}{T_{u3}})}{(s^2+2\zeta_{sp}\omega_{n_{sp}}s+\omega_{n_{sp}}^2)(s^2+2\zeta_{p}\omega_{n_{p}}s+\omega_{n_{p}}^2)} \tag{2.45}$$

In order to maintain the same steady state response for a system with modified stability parameters, the K also has to be changed. This can be seen through the final value theorem which states:

$$\lim_{t\to\infty} f(t) = \lim_{s\to 0} sF(s) \tag{2.46}$$

Using equation 2.41 as an example gives:

$$\begin{aligned}
\lim_{t\to\infty}\frac{\psi(t)}{\delta r(t)} &= \lim_{s\to 0} s\frac{\psi(s)}{\delta r(s)} \\
&= \lim_{s\to 0}\frac{sK_{\psi_{\delta r}}(s+\frac{1}{T_{\psi_r}})(s^2+2\zeta_{\psi_r}\omega_{n_{\psi_r}}s+\omega_{n_{\psi_r}}^2)}{s(s+\frac{1}{T_s})(s+\frac{1}{T_R})(s^2+2\zeta_D\omega_{n_D}s+\omega_{n_D}^2)} \\
&= \frac{K_{\psi_{\delta r}}\frac{\omega_{n_{\psi_r}}^2}{T_{\psi_r}}}{\frac{\omega_{n_D}^2}{T_sT_R}}
\end{aligned} \tag{2.47}$$

as the steady state response. To maintain the same steady state response with modified stability parameters, like T_s, T_R, or ω_{n_D}, $K_{\psi_{\delta r}}$ must be changed accordingly. More specifically, if ω_{n_D} is multiplied by a factor of ϵ, then $K_{\psi_{\delta r}}$ must be multiplied by ϵ^2 to maintain the steady state response. Likewise, if one of the time constants T_s or T_R is multiplied by ϵ, then $K_{\psi_{\delta r}}$ must be divided by ϵ.

Chapter 3

Aircraft Simulation and Implementation

3.1 Introduction

This chapter introduces the simulation environment used for this research. It then discusses the non-linear F-4 model used as a base comparison aircraft. Finally, the development and implementation of the variable stability transfer function computer simulation model is explained.

3.2 Simulation Environment

The F-4 and the variable stability linear transfer function Fortran simulation codes were implemented on a Silicon Graphics Origin 2000^{TM} Deskside System with dual 180 MHz CPUs with 256 MB of RAM and a 4 GB disk. Both simulations were then made entirely compatible with *CASTLE* (Control Analysis and Simulation Test Loop Environment) structure. *CASTLE* is a six degree of freedom non-linear aircraft simulation package developed by Naval Air Warfare Center's Manned Flight Simulator branch (MFS). [7] The *CASTLE* architecture is implemented in the Virginia Tech Flight Simulation Laboratory on the modified 2F122A cantilevered, motion-based, three degree of freedom flight simulator. [8]

Simulations are run at 100 Hz with data collection at 10 Hz in either a batch mode or a real-time piloted simulation. In batch mode, *CASTLE* has the ability to input canned maneuvers or a previously recorded time history of data. In a real-time piloted simulation, a pilot-in-the-loop "flies" a specific, predetermined task. A calligraphic visual scene depicts a dusk or nighttime setting for simulator flight. The visual database includes such realistic environments as Naval air stations, an aircraft carrier landing approach, and a formation or tanker aircraft. The stick feel is produced using an electronic control loader and can be programmed to produce a realistic feel of any aircraft stick. For this research, the stick characteristics of the F/A-18 E/F are used. The cockpit layout is that of an A-6E.

3.3 F-4 Model

The non-linear F-4 Fortran code was originally developed at the Naval Air Test Center (NATC) at Patuxent River, Maryland, and provided to Virginia Tech by the Naval Air Warfare Center Training Systems Division (NAWC-TSD) in Orlando, Florida. This F-4 model was used as the base comparison model for the linear model discussed below.

3.4 Development of Transfer Function Model

Transfer functions were established in equations 2.29 through 2.32 for the responses due to elevator and aileron, and in equations 2.36 through 2.43 for rudder inputs in terms of the stability parameters such as damping ratio (ζ), natural frequency (ω_n), time constants (T), and gain constants (K). For the responses due to throttle input, the following transfer functions were created:

$$\frac{U(s)}{\delta t(s)} = \frac{K_{u_{\delta t}} s(s^2 + 2\zeta_{u_{\delta t}}\omega_{n_{u_{\delta t}}} s + \omega^2_{n_{u_{\delta t}}})}{(s^2 + 2\zeta_{sp}\omega_{n_{sp}} s + \omega^2_{n_{sp}})(s^2 + 2\zeta_p\omega_{n_p} s + \omega^2_{n_p})} \tag{3.1}$$

$$\frac{\alpha(s)}{\delta t(s)} = \frac{K_{\alpha_{\delta t}} s(s + \frac{1}{T_{\alpha_{\delta t}}})}{(s^2 + 2\zeta_{sp}\omega_{n_{sp}} s + \omega^2_{n_{sp}})(s^2 + 2\zeta_p\omega_{n_p} s + \omega^2_{n_p})} \tag{3.2}$$

$$\frac{\theta(s)}{\delta t(s)} = \frac{K_{\theta_{\delta t}}}{(s^2 + 2\zeta_{sp}\omega_{n_{sp}} s + \omega^2_{n_{sp}})(s^2 + 2\zeta_p\omega_{n_p} s + \omega^2_{n_p})} \tag{3.3}$$

The eighteen transfer functions (two for each of the five lateral directional states and two for each of the four longitudinal states) were converted into state-space systems using the MATLAB command *TF2SS*. *TF2SS* is a transfer function to state-space conversion that takes the transfer function as an input and returns the A, B, C, and D matrices from equations 2.14 and 2.15 in controller canonical form. This lead to 36 linear equations, 18 for $\dot{x}$ and 18 for y, in terms of the 46 stability parameters.

In order to test this linearized model as accurately representing an aircraft, numerical values had to be found for each of the stability parameters. To find these, the F-4 model was linearized about some reference flight condition. This reference flight condition was straight, steady, symmetric, level flight at 500 ft/s at 10,000 ft altitude. This resulted in a state-space system which was converted into transfer-functions using the MATLAB command *SS2TF*. This conversion was performed in order to get the numerical values in a form similar to the symbolic transfer functions. From there, each of the 46 stability parameters (ζ, ω_n, T, & K) were solved numerically. This resulted in linear equations for the aircraft outputs in terms of the aircraft states. These outputs from the transfer function simulation were V_T, α, β, ϕ, θ, ψ, p, q, and r.

Rather than using outputs these as its inputs, *CASTLE* takes body axis forces and moments as inputs to calculate the aircraft outputs. To implement the transfer function simulation in *CASTLE*, equations for the body axis forces (X, Y, Z) and moments (L, M, N) were derived from equations 2.1, 2.2, and 2.3 for the forces and 2.7, 2.8, and 2.9 for the

moments. They were found to be:

$$X = m\dot{u} + mg\sin\theta - mrv + mqw \tag{3.4}$$

$$Y = m\dot{v} - mg\sin\phi\cos\theta - mpw + mru \tag{3.5}$$

$$Z = m\dot{w} - mg\cos\phi\cos\theta - mqu + mpv \tag{3.6}$$

and:

$$L = -I_{xz}pq + (I_{zz} - I_{yy})qr + I_{xx}\dot{p} - I_{xz}\dot{r} \tag{3.7}$$

$$M = I_{yy}\dot{q} + (I_{xx} - I_{zz})pr + I_{xz}(p^2 - r^2) \tag{3.8}$$

$$N = I_{zz}\dot{r} - I_{xz}\dot{p} + I_{xz}qr + (I_{yy} - I_{xx})pq \tag{3.9}$$

To get these equations in terms of the stability axis V, α, and β rather than the body axis u, v, and w, the following substitutions were used:

$$u = V\cos\alpha\cos\beta \tag{3.10}$$

$$v = V\sin\beta \tag{3.11}$$

$$w = V\sin\alpha\cos\beta \tag{3.12}$$

and thus:

$$\dot{u} = \dot{V}\cos\alpha\cos\beta - V\dot{\alpha}\cos\beta\sin\alpha - V\dot{\beta}\cos\alpha\sin\beta \tag{3.13}$$

$$\dot{v} = \dot{V}\sin\beta + V\dot{\beta}\cos\beta \tag{3.14}$$

$$\dot{w} = \dot{V}\sin\alpha\cos\beta - V\dot{\beta}\sin\alpha\sin\beta + V\dot{\alpha}\cos\beta\cos\alpha \tag{3.15}$$

These equations were used to create the variable stability transfer function simulation. The outputs from the transfer function simulation were used to derive the forces and moments. These forces and moments were used by *CASTLE* to compute its own outputs for the simulation. This is a simple dynamic inversion, from which ideally terms will cancel and the output from the transfer function simulation will equal that of *CASTLE*.

3.5 Model Following

Comparing the output from the transfer function simulation to that of *CASTLE* showed a growing discrepancy. As time progressed, so did the magnitude of the error between the outputs. This was accounted for by the difference between the transfer function outputs and *CASTLE* outputs at each time step. Therefore, terms in the force and moment equations (equations 3.4 to 3.9) did not cancel, resulting in greater and greater differences in the outputs. This problem was solved by implementing a simple model follower, as described below.

Model following control attempts to cause the states of a given physical system (the plant) to have responses to initial conditions and control inputs that are the same as those of some desired model (the model). It is a way to reduce the error between the desired and the actual aircraft dynamics. [9]

For example, take the body axis force equation (3.4). Consider the transfer function outputs to be the model, with subscript m, and the *CASTLE* outputs to be the plant, with

subscript p. Since there are six controls, six state rates can be modelled. These were $\dot{u}$, $\dot{v}$, $\dot{w}$, $\dot{p}$, $\dot{q}$, and $\dot{r}$, for equation 3.4, $\dot{u}$ applies. Equation 3.4 can be written as:

$$X = m\dot{u}_{cm} + mg\sin\theta - mrv + mqw \tag{3.16}$$

where

$$\dot{u}_{cm} = \dot{u}_m + \lambda_u(u_m - u_p) \tag{3.17}$$

Substituting equations 3.16 and 3.17 into equation 2.1 for the plant gives:

$$\begin{aligned} \dot{u}_p &= rv - qw - g\sin\theta + \frac{X}{m} \\ &= rv - qw - g\sin\theta + \dot{u}_{cm} - rv + qw + g\sin\theta \\ &= \dot{u}_m + \lambda_u(u_m - u_p) \end{aligned} \tag{3.18}$$

This results in an error between the model and the plant as follows:

$$e_u = u_m - u_p \tag{3.19}$$

and thus the error rate:

$$\begin{aligned} \dot{e}_u &= \dot{u}_m - \dot{u}_p \\ &= \dot{u}_m - \dot{u}_m - \lambda_u(u_m - u_p) \\ &= -\lambda_u(u_m - u_p) \\ &= -\lambda_u e_u \end{aligned} \tag{3.20}$$

The error between the plant and the model is reduced at a rate proportional to λ_u, and thus the plant state, $\dot{u}_p$, follows the model state, $\dot{u}_m$. This leads to an improved dynamic inversion, where the *CASTLE* outputs are nearly equal to the transfer function outputs.

A similar approach was taken for the remaining force and moment equations, resulting in the following six equations:

$$X = m\dot{u}_m + mg\sin\theta - mrv + mqw + m\lambda_u(u_m - u_p) \tag{3.21}$$

$$Y = m\dot{v}_m - mg\sin\phi\cos\theta - mpw + mru + m\lambda_v(v_m - v_p) \tag{3.22}$$

$$Z = m\dot{w}_m - mg\cos\phi\cos\theta - mqu + mpv + m\lambda_w(w_m - w_p) \tag{3.23}$$

and:

$$L = -I_{xz}pq + (I_{zz} - I_{yy})qr + I_{xx}\dot{p} - I_{xz}\dot{r} + I_{xx}\lambda_p(p_m - p_p) - I_{xz}\lambda_r(r_m - r_p) \tag{3.24}$$

$$M = I_{yy}\dot{q} + (I_{xx} - I_{zz})pr + I_{xz}(p^2 - r^2) + I_{yy}\lambda_q(q_m - q_p) \tag{3.25}$$

$$N = I_{zz}\dot{r} - I_{xz}\dot{p} + I_{xz}qr + (I_{yy} - I_{xx})pq + I_{zz}\lambda_r(r_m - r_p) - I_{xz}\lambda_p(p_m - p_p) \tag{3.26}$$

3.6 TransferFunctions.f

This section explains the purpose of the main computer simulation routine, TransferFunctions.f, which is included in the appendix. TransferFunctions.f is the main program which executes the simulation. It is called at each time step. When first run, it sets constants for

the 46 stability parameters and sets the initial conditions. TransferFunctions.f calculates the values for the outputs (y), their time derivatives ($\dot{y}$), the time derivatives of the states ($\dot{x}$), the forces (X, Y, Z), and the moments (L, M, N) after each iteration of the simulation based on equations derived above. The forces and moments are calculated solely for the *CASTLE* environment, as it uses the forces and moments to calculate the aircraft's states. TransferFunction.f also performs a numerical integration at the end of each iteration of the simulation to calculate an updated value for the states (x) for the next time step.

3.7 Airframe Validity

The airframe in this research was modelled after the F-4 airframe model discussed above. It was not meant to be an exact representation of the F-4, but a sufficiently realistic aircraft platform for simulation comparison with the transfer function model.

Chapter 4

Evaluation of Simulation Model

4.1 Introduction

This chapter discusses the methods that will be used to evaluate the fidelity of the variable stability simulation model. First, handling quality ratings and their application to this research are discussed. Then, ground-based versus in-flight simulation are compared. The test pilot is introduced in the following section. Finally, the representative maneuvers used for evaluation and their results are discussed.

4.2 Handling Quality Ratings

The handling qualities of an aircraft are measured by a test pilot and are therefore qualitative, not quantitative. In order to assign numbers to these pilot ratings for comparison, the Cooper-Harper scale, shown in Figure 4.1 [11], can be used. The scale has ten points, where 1 indicates excellent and 10 the worst qualities possible. The scale is dichotomous, which improves repeatability by leading the evaluation pilot through a series of decisions regarding the task performance and the pilot workload. It is an ordinal, but not an interval scale. [10]

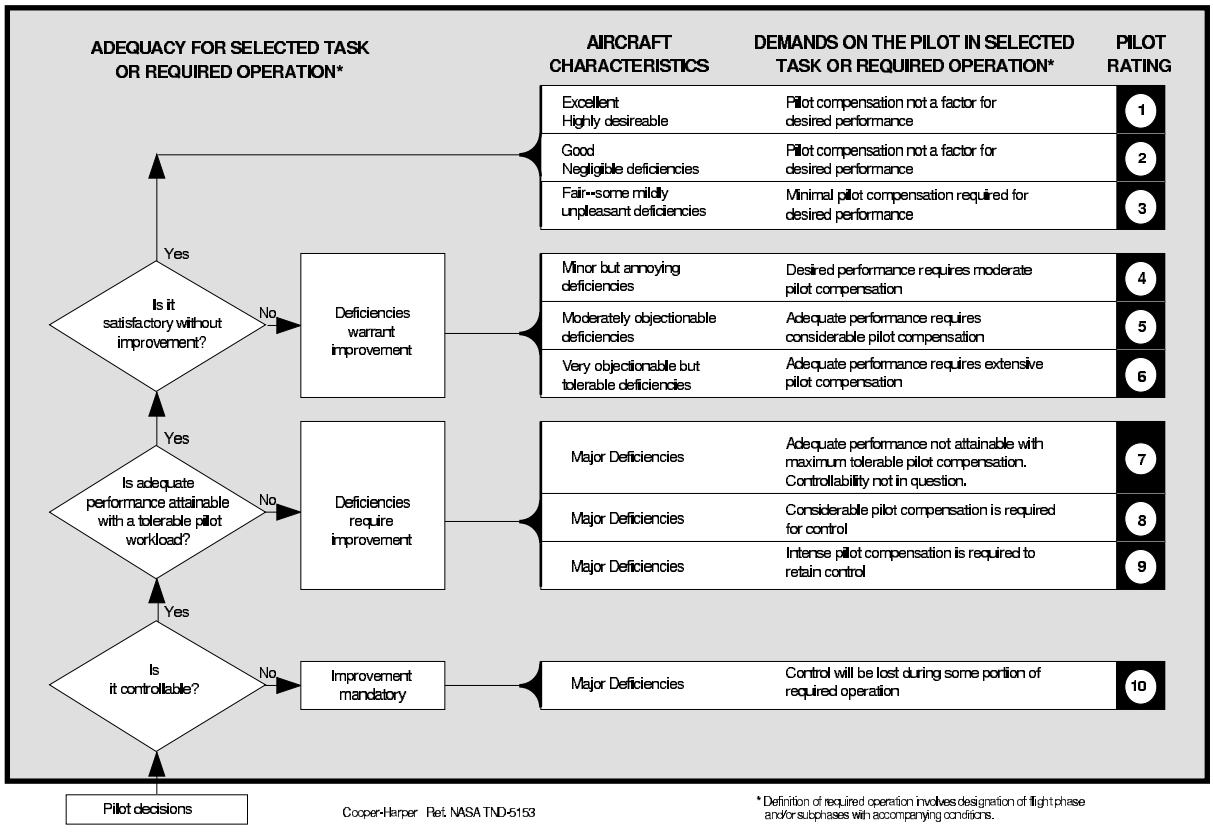

Figure 4.1: Cooper-Harper rating scale.

4.3 Ground-Based vs In-Flight Simulation

As the inspiration for this work was the NF-16 VISTA, a brief discussion of ground-based versus in-flight simulation is included. While real-time ground-based simulations offer a realistic flight environment, there are differences between simulated flight and actual flight that significantly affect task performance and completion.

Ground-based simulators are limited by the lack of realistic visual and motion cues, as well as the realistic "feel" of flying. This includes the apprehension in response to real dangers a pilot experiences while flying a real aircraft as compared to a simulator. In general, flying qualities obtained from the ground-based simulator can be considered conservative. This indicates that caution must be exercised in the interpretation of simulation results. [12]

This having been said, there are numerous advantages to a ground-based simulator over an in-flight simulator. The flight envelope of an in-flight simulator is fairly limited due to the aerodynamics of the base airframe. A ground-based simulator is much more versatile, as it has the ability to mimic the flying qualities of any aircraft. Another great advantage of the ground-based simulator is financial. It is a much more cost effective way to simulate flight.

4.4 Test Pilot

One test pilot was used to "fly" the simulator to acquire all data presented in this study. Preliminary flights were flown by other pilots, but these results were not included for analysis. The test pilot is currently an Associate Professor in the Department of Aerospace and Ocean Engineering at Virginia Polytechnic Institute and State University. After a distinguished career as a carrier based fighter pilot, he served as a test pilot instructor at the United States Naval Test Pilot School at Patuxent River Naval Air Station. He is also a member of the Society of Experimental Test Pilots.

4.5 Representative Maneuvers

To test the fidelity of the variable stability simulation created for this study, canned input maneuvers were flown with both the non-linear F-4 model and the transfer function model. As the transfer function simulation model was obtained through linearizing the full system about some reference flight condition, the small angle assumption applied and the linear model was assumed to be valid only for flight conditions near this reference condition. Therefore, this lead to a choice of inputs that did not entail any large excursions from the reference conditions.

After the validity of the transfer function model was established, the variable stability capabilities of the model were examined. The transfer function model was compared for

varying stability parameters by both pilot opinion and by the hard data acquired through the simulation. The pilot opinion was based on the Cooper-Harper rating scale previously discussed. The hard data included such outputs as time histories of aircraft states and relative aircraft positions.

4.5.1 Canned Inputs: Comparison with F-4

Canned inputs were used to compare the responses of the non-linear F-4 and transfer function models and to establish the transfer function model as a valid representation of the non-linear model. Four runs were made with each model, with each run consisting of a control input for one second, applied two seconds into the simulation. The four runs involved the following four control inputs, respectively: a 10% increase in throttle, a 10° change in elevator input (trailing edge up), a 10° aileron input (right wing down), and a 10° rudder input (trailing edge right). The reference conditions for each run were 500 fps airspeed, 10,000 ft altitude, and 3.8° angle of attack. This resulted in control deflections of 30.4% throttle and $-0.12°$ elevator. Figures 4.2 through 4.19 discussed below show the response of several states to these control inputs for both the non-linear F-4 and the transfer function model.

4.5.1.1 Throttle Input

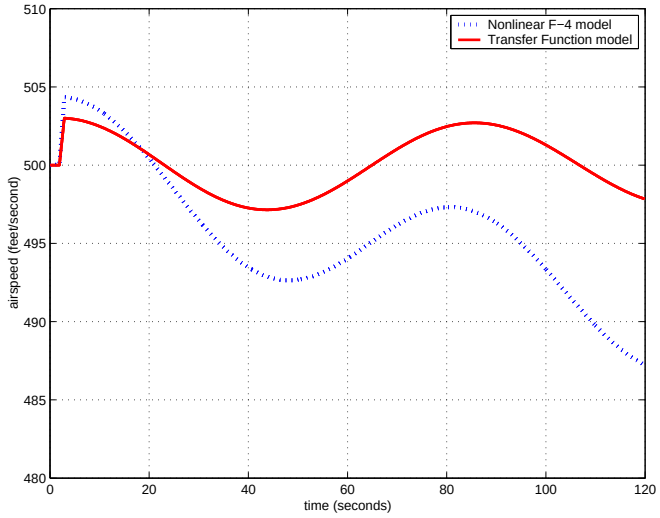

Figure 4.2: Time history of velocity in response to a 10% change in throttle input for one second.

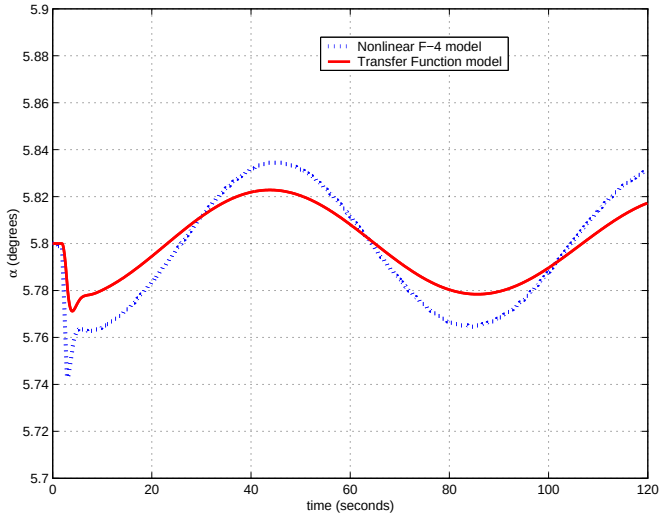

Figure 4.3: Time history of angle-of-attack in response to a 10% change in throttle input for one second.

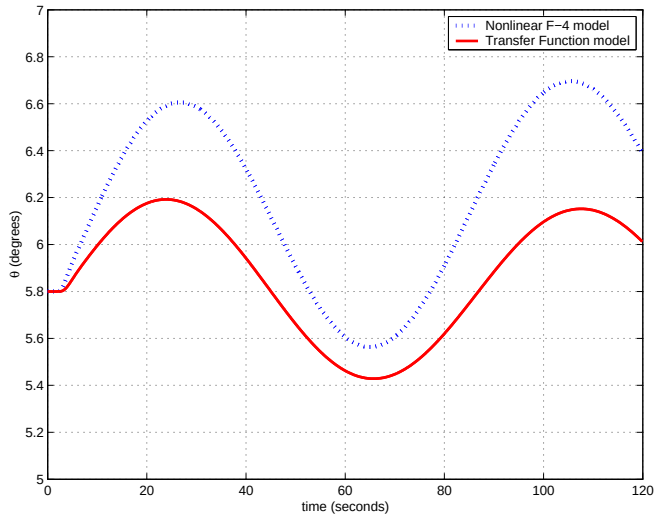

Figure 4.4: Time history of pitch angle in response to a 10% change in throttle input for one second.

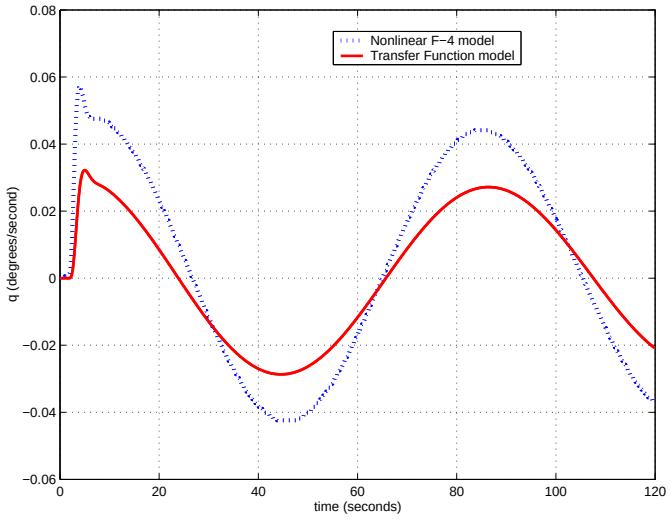

Figure 4.5: Time history of pitch rate in response to a 10% change in throttle input for one second.

Figures 4.2 through 4.5 show the longitudinal state responses to a 10% throttle increase. While the trends of the two models for each state agreed, there were discrepancies.

Figure 4.2 shows the velocity response to the throttle increase. Both models increased in velocity initially when the throttle was applied, but the nonlinear F-4 model increased to 504 fps while the transfer function model increased to 503 fps. When the throttle was returned to the trim position, both models began to exhibit the phugoid mode oscillatory response. While both phugoid modes had a period of 80 seconds, the transfer function model oscillated about the reference condition while the nonlinear model oscillated along a decreasing slope.

The angle of attack response to the throttle input can be seen in figure 4.3. Again, the phugoid oscillatory response with a period of 80 seconds was evident for both models. The amplitude of the nonlinear model response was 0.4 fps compared to 0.3 fps for the transfer function model and both are oscillating about the reference condition.

Figure 4.4 for pitch angle also exhibited the phugoid response with a period of 80 seconds. The nonlinear response oscillated with an amplitude of 0.5 degrees about 6.1 degrees while the transfer function model response had an amplitude of 0.4 degrees about the nominal 5.8 degrees.

The pitch rate response is shown in figure 4.5. The phugoid response had an amplitude of 0.045 degrees/second for the nonlinear model compared to the 0.03 degrees/second for the transfer function model.

Discrepancies could depend on the integration method used at each step of the simulation. The linear transfer function model used a simple, first order Euler integrator while the fourth-order Runge-Kutta algorithm was used by the nonlinear F-4 model for integration.

4.5.1.2 Elevator Input

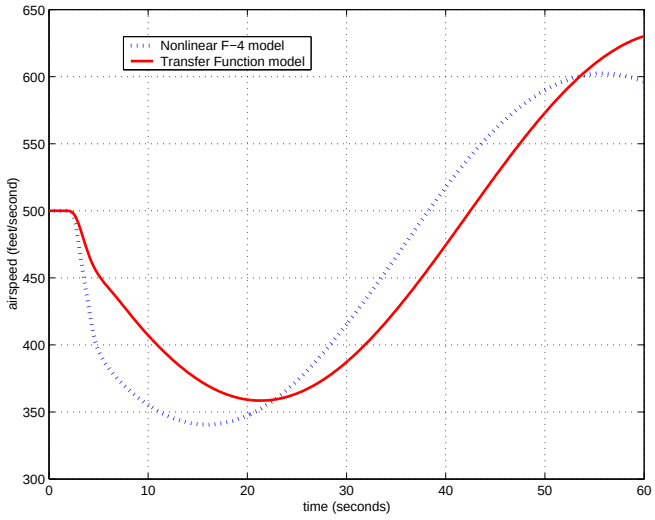

Figure 4.6: Time history of velocity in response to a 10° change in elevator input (trailing edge up) for one second.

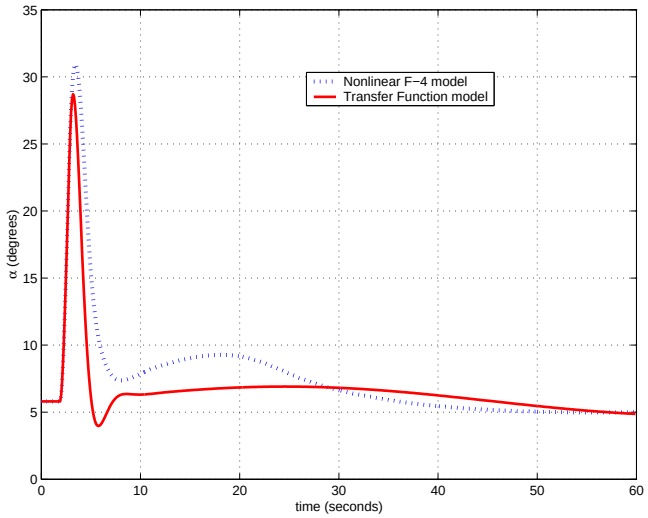

Figure 4.7: Time history of angle-of-attack in response to a 10° change in elevator input (trailing edge up) for one second.

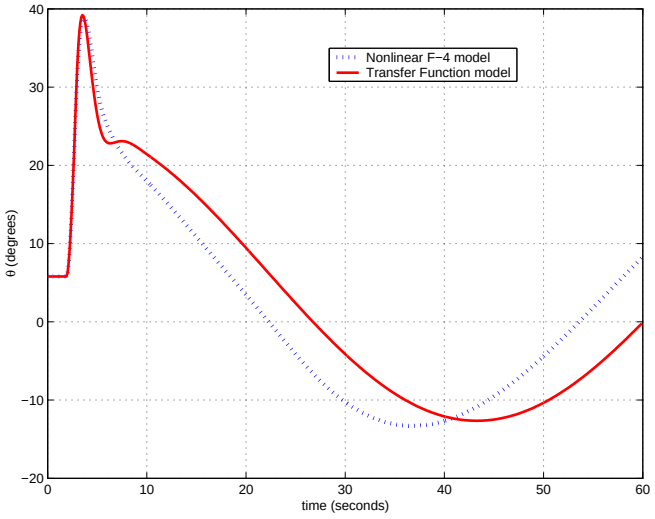

Figure 4.8: Time history of pitch angle in response to a 10° change in elevator input (trailing edge up) for one second.

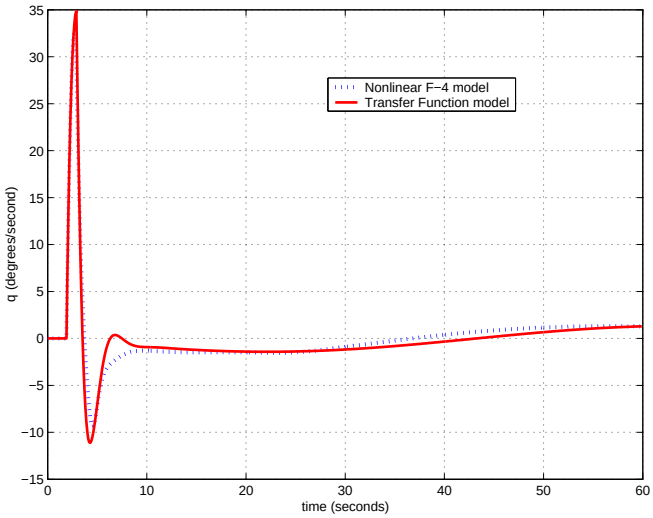

Figure 4.9: Time history of pitch rate in response to a 10° change in elevator input (trailing edge up) for one second.

Figures 4.6 through 4.9 show the longitudinal state responses to a 10° trailing edge up change in elevator deflection for one second.

Figure 4.6 shows the velocity response to the elevator deflection. Both models showed a drop in velocity as the control deflection was applied. This was followed by an oscillatory phugoid response back towards the reference conditions.

The angle of attack response to the elevator input can be seen in figure 4.7. Both models initially increased in angle of attack by 25° as the input was applied, then returned to the reference conditions. The transfer function model showed an overshoot in the short period response, then higher damping than the nonlinear model as they returned to the reference conditions.

Figure 4.8 shows the response of the pitch angle to the elevator deflection. The two models increased by 35° and then oscillated back to the initial conditions. A difference can

be seen at 7 seconds, as the transfer function response deviated from its descent. This was caused by the overshoot in the angle of attack response.

The pitch rate response is shown in figure 4.9. The transfer function model showed less damping in the short period response than the nonlinear model. The initial short period response was damped out by 10 seconds after which the phugoid oscillations were evident.

4.5.1.3 Aileron Input

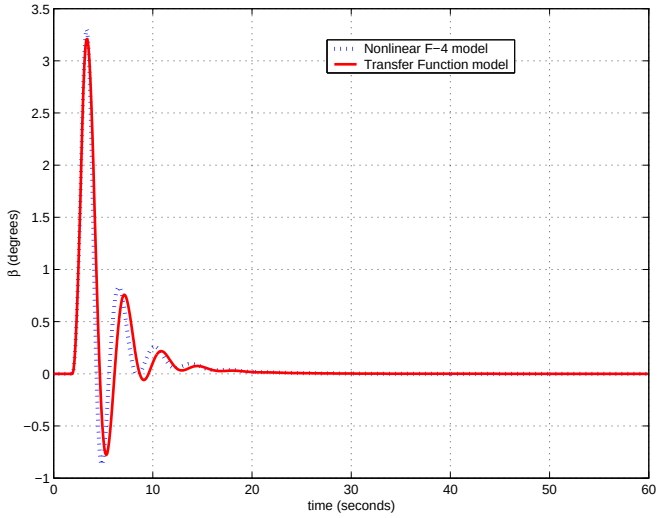

Figure 4.10: Time history of sideslip in response to a 10° aileron input for one second.

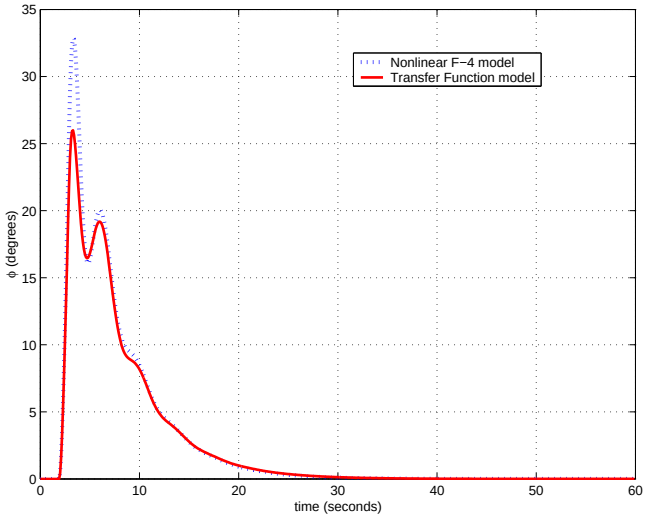

Figure 4.11: Time history of bank angle in response to a 10° aileron input for one second.

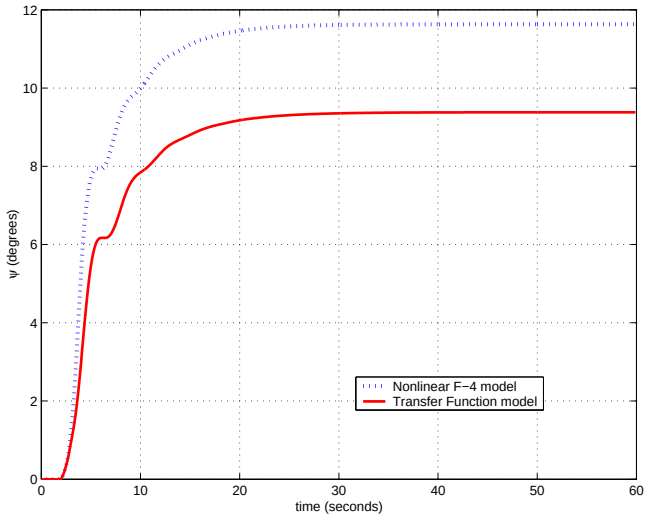

Figure 4.12: Time history of heading angle in response to a 10° aileron input for one second.

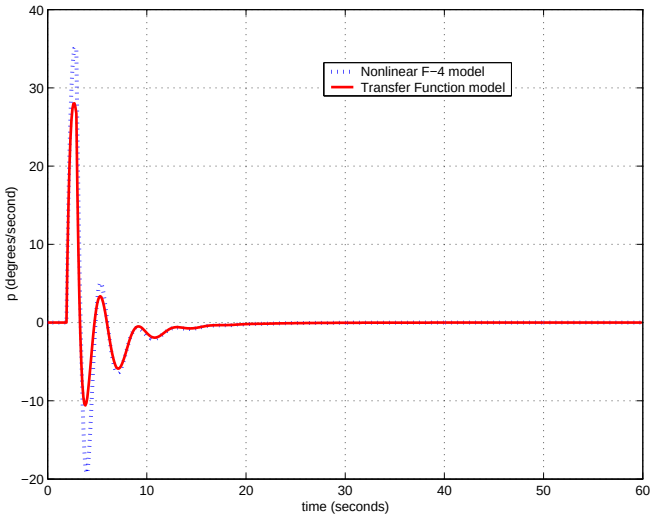

Figure 4.13: Time history of roll rate in response to a 10° aileron input for one second.

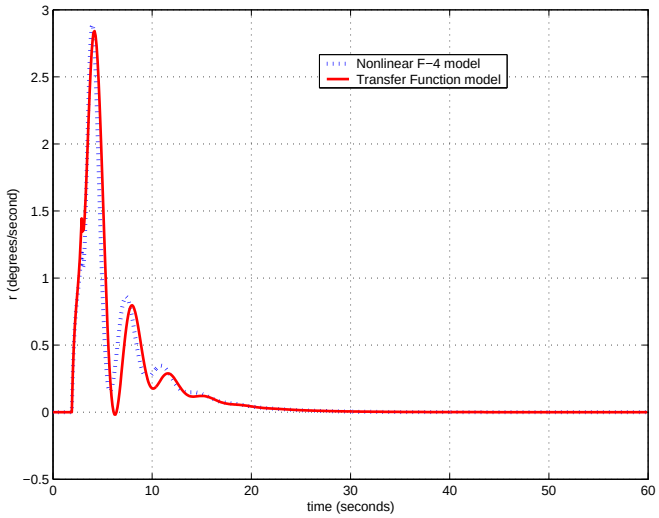

Figure 4.14: Time history of yaw rate in response to a 10° aileron input for one second.

Figures 4.10 through 4.14 show the lateral-directional state responses to a 10° aileron deflection.

Figure 4.10 shows the time history of sideslip in response to the aileron deflection. The nonlinear and transfer function models initially slipped to 3.2°. Both models then damped out to the reference conditions, with the transfer function model damping out with a slight time lag as compared to the nonlinear model.

The time history of the bank angle in response to the aileron input is shown in figure 4.11. The nonlinear model initially banked to 32° while the transfer function model banked to 26°. Both returned to the reference conditions along a first order response, with another peak at 7 seconds.

Figure 4.12 displays the heading angle response to the aileron input. The nonlinear model approached 11.5° asymptotically and the transfer function model approached 9.5°

asymptotically.

The time history of roll rate is presented in figure 4.13. The transfer function model initially achieved a roll rate of 28°/s and oscillated back to the reference condition in 20 seconds. The nonlinear model initially had a roll rate of 35°/s and damped out to the reference conditions by 20 seconds.

Figure 4.14 shows the time history of the yaw rate response to the aileron deflection. The nonlinear and transfer function models initially responded with a yaw rate of 2.8°/s. Both damped out to the reference conditions in 45 seconds.

4.5.1.4 Rudder Input

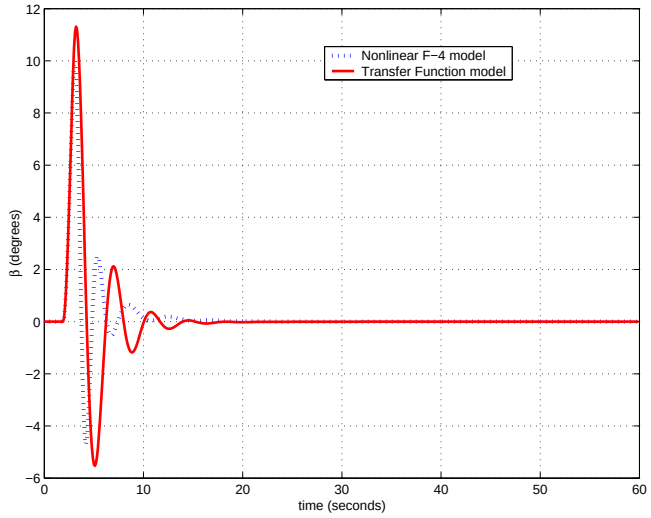

Figure 4.15: Time history of sideslip in response to a 10° rudder input for one second.

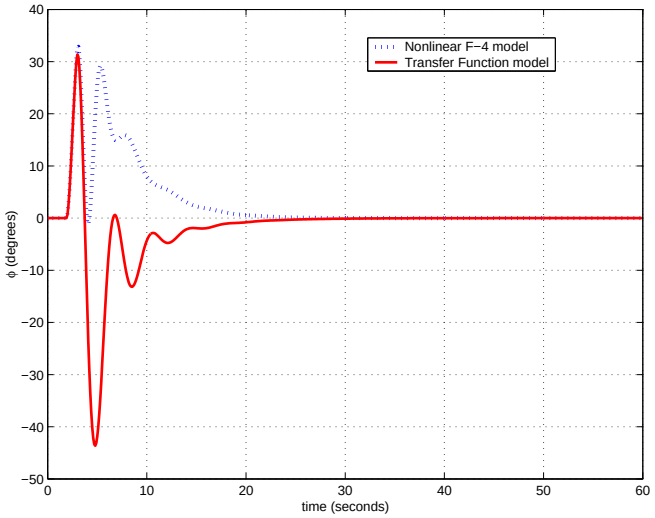

Figure 4.16: Time history of bank angle in response to a 10° rudder input for one second.

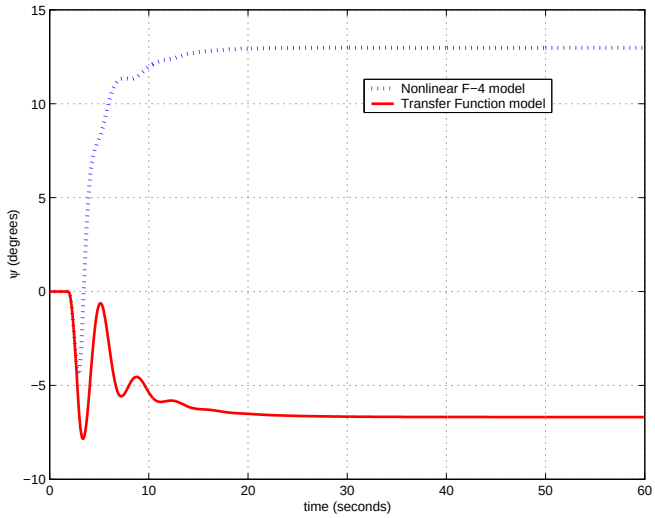

Figure 4.17: Time history of heading angle in response to a 10° rudder input for one second.

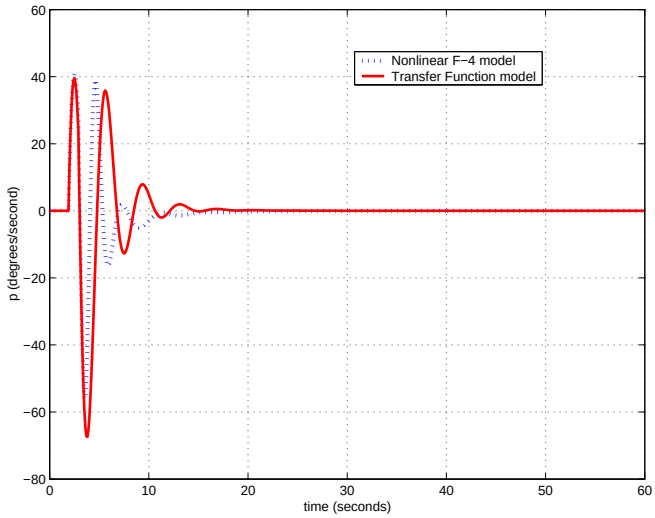

Figure 4.18: Time history of roll rate in response to a 10° rudder input for one second.

The time histories of the lateral-directional states in response to a rudder input are shown in figures 4.15 through 4.19.

Figure 4.15 shows the sideslip angle response to the rudder input. While the nonlinear model initially responded to 10° sideslip and then oscillated back to zero, the transfer function

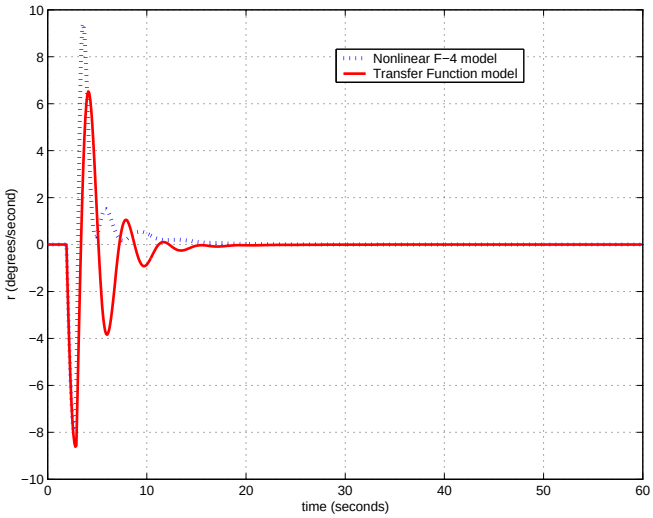

Figure 4.19: Time history of yaw rate in response to a 10° rudder input for one second.

model increased to 11° before oscillating back to the reference conditions. As in the sideslip response to aileron input, the sideslip response to rudder input damped out with a time lag as compared with the nonlinear model.

Figure 4.16 displays the roll angle response to the rudder input. The nonlinear model initially responded to 33°, then returned to zero, back up to 29°, then damped out to zero in 40 seconds. The transfer function model originally rolled to 32°, then overshot to −42°, then damped out to zero in 40 seconds. The nonlinear response seems suspect in this case, as an overshoot in bank angle was expected when the rudder input was removed.

Shown in figure 4.17 is the heading angle response to the rudder input. The two models showed the same initial response, but very different steady state responses. Again, the nonlinear response seems suspect. There was no reason for a reversal in heading after the rudder input is removed.

Figure 4.18 displays the roll rate response to the rudder input. Both the nonlinear and transfer function models initially rolled with a rate of 40°/s, then −55°/s and −65°/s, respectively. As noted above, the transfer function model damped out slower than the nonlinear model.

Figure 4.19 shows the yaw rate time history. Both the nonlinear and transfer function models originally had a yaw rate of −8°/s, then 9.5°/s and 4°/s, respectively. Then the nonlinear model damped out to zero in 30 seconds, while the transfer function model again showed a time lag in the damping. This discrepancy can be attributed to the discrepancy in heading angle, as yaw rate is the time derivative of heading angle.

While the response of the states of the transfer function model did not track those of the nonlinear model exactly, the trends were the same. These results were used to establish that the transfer function model was indeed a reasonable representation of the nonlinear model.

4.5.2 Canned Inputs: Variable Stability Testing

Canned inputs were used to show the variable stability capabilities of the transfer function model. First simulations were performed with the stability parameters obtained from linearizing the F-4 about the reference conditions. Then one stability parameter was changed and the simulation was run again. These runs were compared to evaluate the variable stability capabilities of the simulation. The short period damping ratio, ζ_{sp}, and the phugoid

natural frequency, ω_{np}, were selected as the varied parameters as they are easy to visualize and measure experimentally.

Figures 4.20 and 4.21 show the short period mode while figures 4.22 and 4.23 show the phugoid mode.

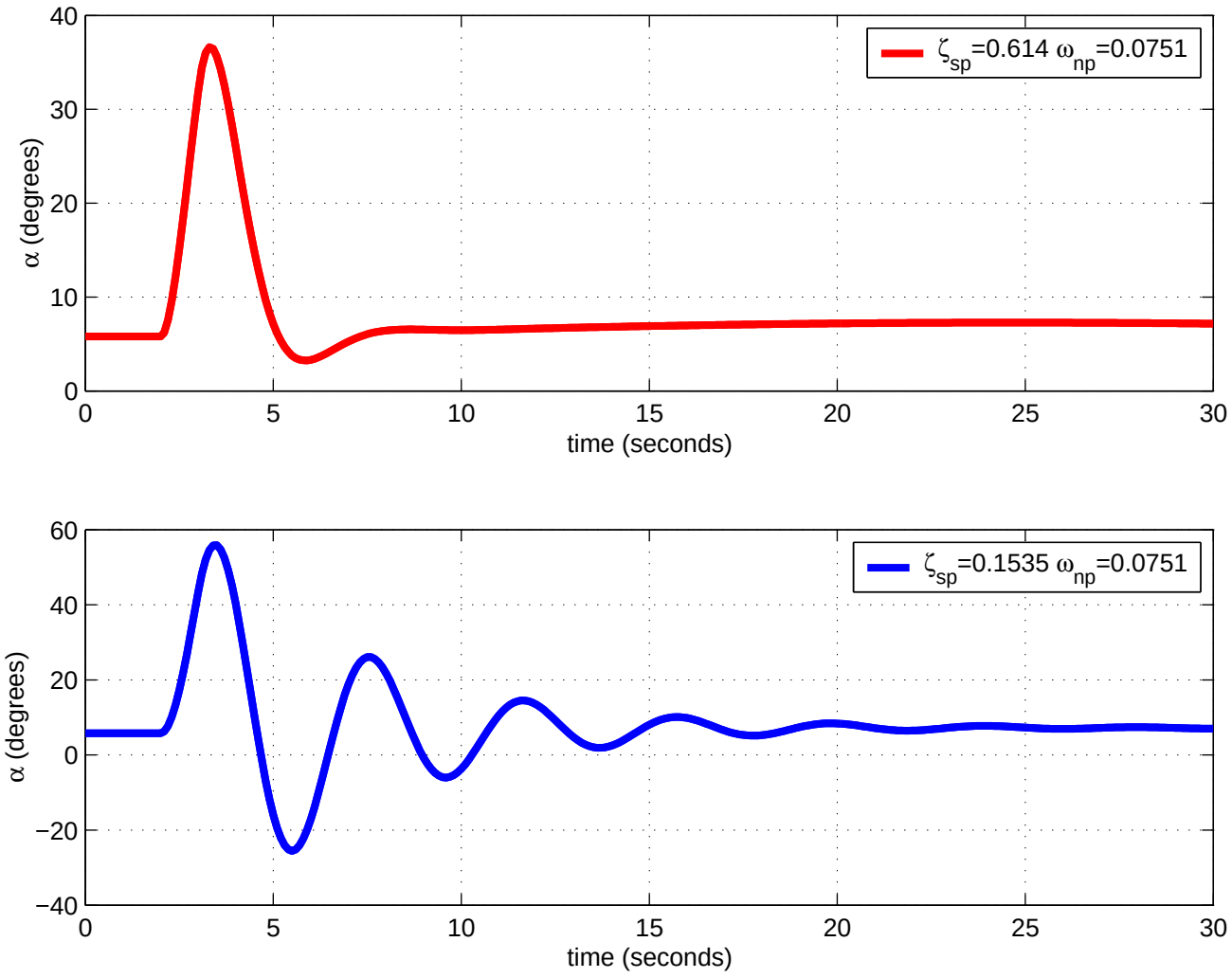

Figure 4.20: Time history of angle of attack for a one second elevator input.

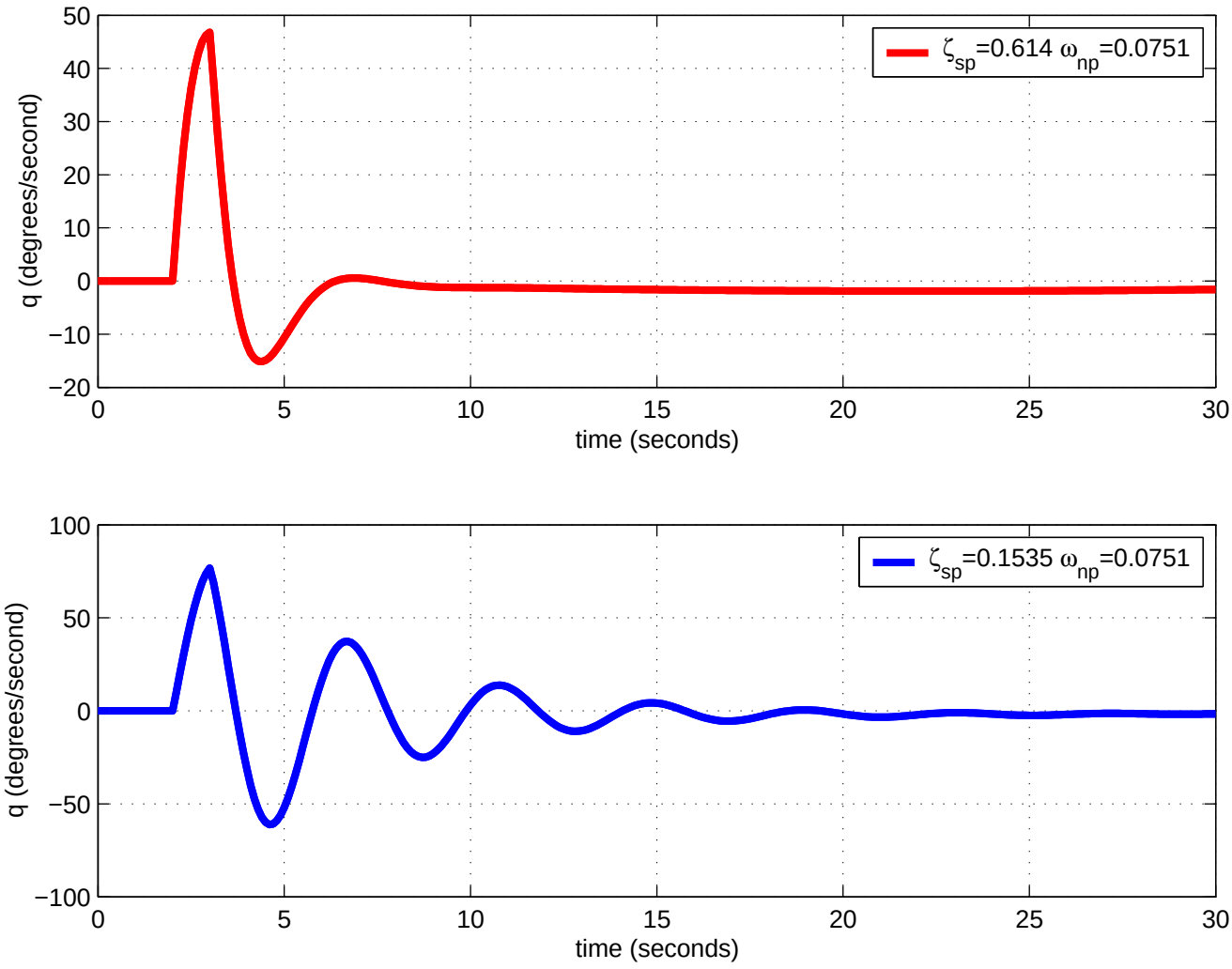

Figure 4.21: Time history of pitch rate for a one second elevator input.

Figures 4.20 and 4.21 display the time history of the angle of attack and pitch rate, respectively, for a one second elevator input. The top figure shows the response for the transfer function model with the linearized F-4 parameters, including $\zeta_{sp} = 0.614$ and $\omega_{np} = 0.0751$. The lower figure shows the response with $\zeta_{sp} = 0.1535$.

Equation 4.1 below was used to experimentally determine the damping ratio ζ based on successive peaks x_1 and x_2.

$$\zeta = \frac{ln\frac{x_1}{x_2}}{\sqrt{4\pi^2 + (ln\frac{x_1}{x_2})^2}} \tag{4.1}$$

For the theoretical $\zeta_{sp} = 0.614$ in figure 4.20, the experimental short period damping ratio of 0.574. For the lower figure with a theoretical $\zeta_{sp} = 0.1535$ in figure 4.20, the experimental short period damping ratio of 0.1434. As this data was collected at 10 Hz, errors may have been introduced in locating the successive peaks.

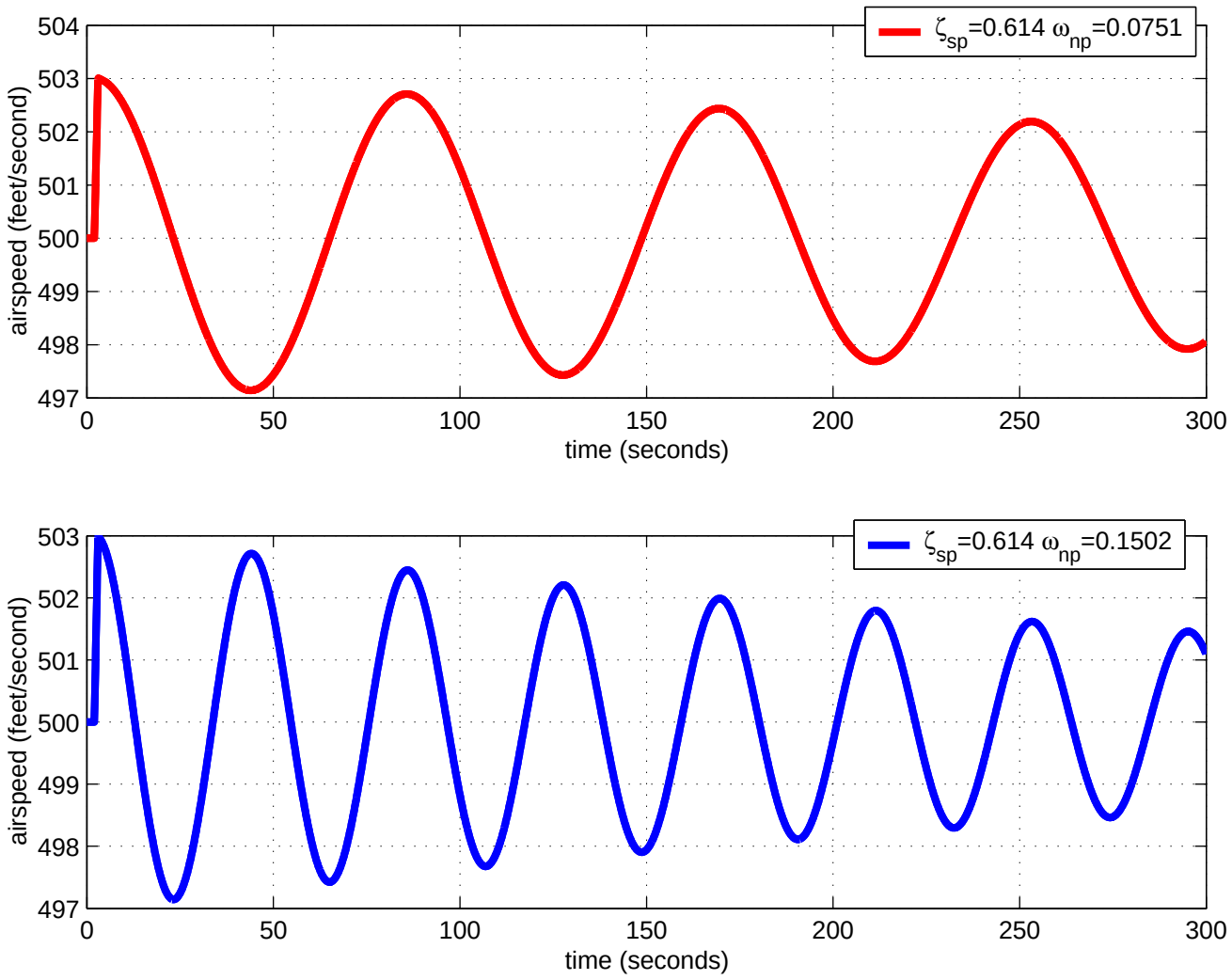

Figure 4.22: Time history of airspeed for a one second 10% throttle increase.

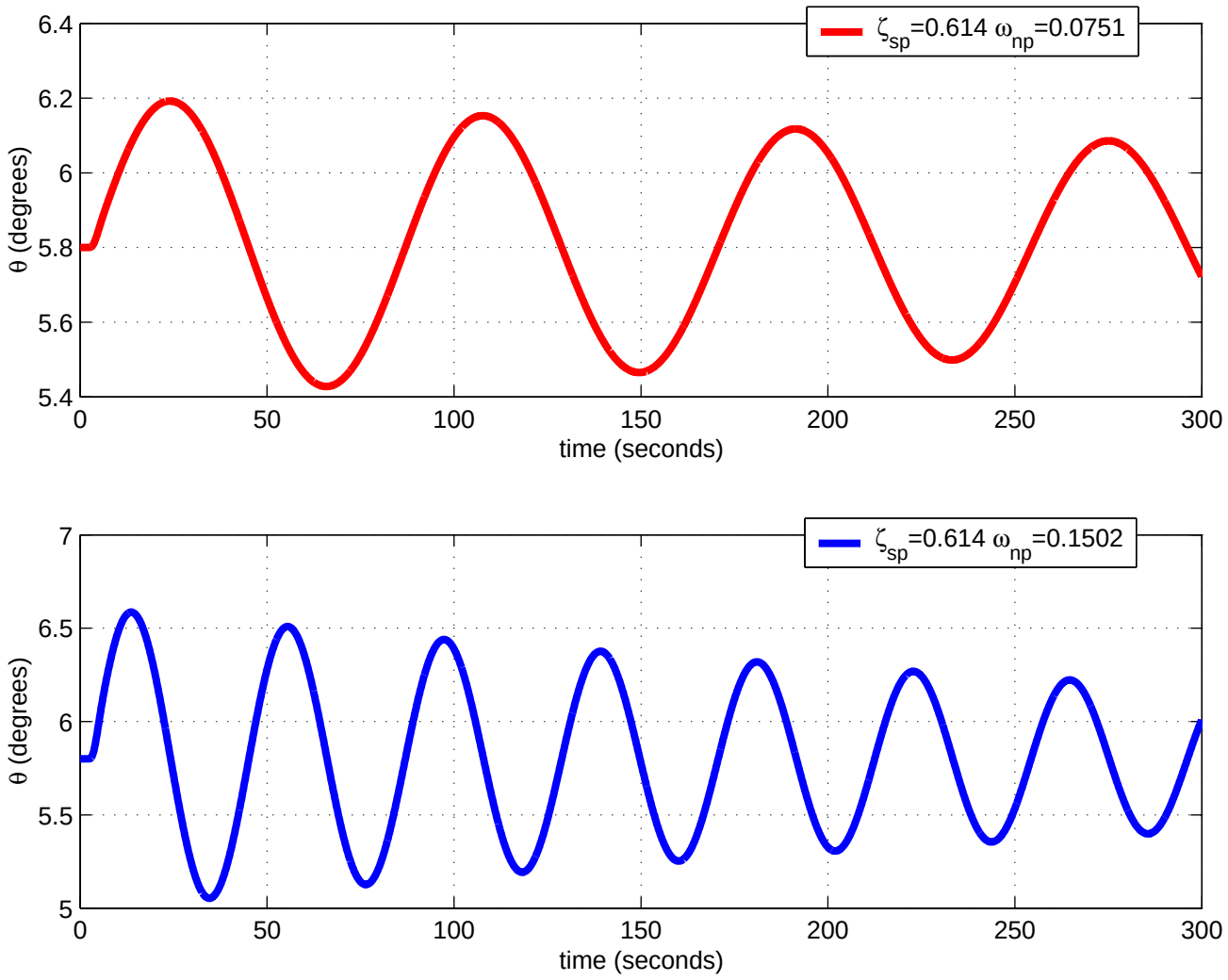

Figure 4.23: Time history of pitch angle for a one second 10% throttle increase.

Figures 4.22 and 4.23 display the time history of the airspeed and pitch angle, respectively, for a one second throttle input. The top figure shows the response for the transfer function model with the linearized F-4 parameters, including $\zeta_{sp} = 0.614$ and $\omega_{np} = 0.0751$. The lower figure shows the response with $\omega_{np} = 0.1502$.

Equation 4.2 below was used to experimentally determine the natural frequency ω_n based on the period T.

$$\omega_n = \frac{2\pi}{T} \tag{4.2}$$

For the theoretical $\omega_{np} = 0.0751$ radians per second in figure 4.22, the period was 84 seconds. This resulted in an experimental phugoid natural frequency of 0.0748 radians per second.

For the lower figure with a theoretical $\omega_{np} = 0.1502$ radians per second in figure 4.20, the period was 42 seconds. This gave an experimental natural frequency of 0.1496 radians per second.

These results confirmed that the model is a valid variable stability model.

4.5.3 Pilot-In-The-Loop Simulation: Formation Flight

To demonstrate the variable stability capabilities of the transfer function model in a real-time situation, pilot-in-the-loop simulations were flown. The task was flown three times for each configuration of stability parameters. The first configuration was the transfer function model based on the nonlinear F-4 linearized about reference conditions. The second configuration

was this base model with the short period damping ratio decreased from $\zeta_{sp} = 0.614$ to $\zeta_{sp} = 0.100$. The third configuration was the base configuration with the phugoid natural frequency increased from $\omega_{np} = 0.0751$ to $\omega_{np} = 0.200$. As mentioned previously, the short period damping ratio, ζ_{sp}, and the phugoid natural frequency, ω_{np}, were selected as the varied parameters as they are easy to visualize and measure experimentally.

This maneuver was chosen as an evaluation task for this research for two reasons. First, it is a commonly encountered task used to extend the mission range of an aircraft. Second, it requires steady flight with small control inputs to track the lead aircraft, so it is ideal for the linear transfer function model.

4.5.3.1 Description

To simulate formation flight, the pilot flew formation with a K/A-6 tanker aircraft. Both the tanker aircraft and the piloted aircraft flew at 500 fps and an altitude of 10000 feet above sea level. The transfer function model began flying 1000 feet behind the K/A-6. The pilot then positioned the aircraft in formation with the tanker. At this point, the task will began. Although proper motion cues make the task more realistic and reasonable, improper motion cues make the task much more difficult. Therefore, this simulation was run without motion cues, so the task became more difficult as no accelerations were felt by the pilot. Therefore, the tanker aircraft flew straight and level. The objective of the task was for the pilot to track the tanker aircraft and maintain a constant range for a time period of one minute.

For the formation flight task, desired performance was defined as keeping the separation distance within ±10 feet. Adequate performance was defined as keeping the separation distance within ±25 feet.

4.5.3.2 Task Evaluation

For each configuration, three runs were flown by the pilot. During each run, the pilot made comments on the flying qualities of the aircraft model. Following each run, the pilot assigned a rating to the task based on the Cooper-Harper rating scale. Data was collected throughout each run on the time histories of aircraft states, aircraft position, and control deflections. The performance of the transfer function model as a real-time variable stability model was then established.

4.5.3.3 Results

Figure 4.24 shows the relative range of the transfer function model and K/A-6 aircraft for one of the test runs. The pilot navigated from the initial range of 1000 feet to a formation position at 53 seconds elapsed time. This is where the task began, and the pilot tracked the tanker aircraft for one minute.

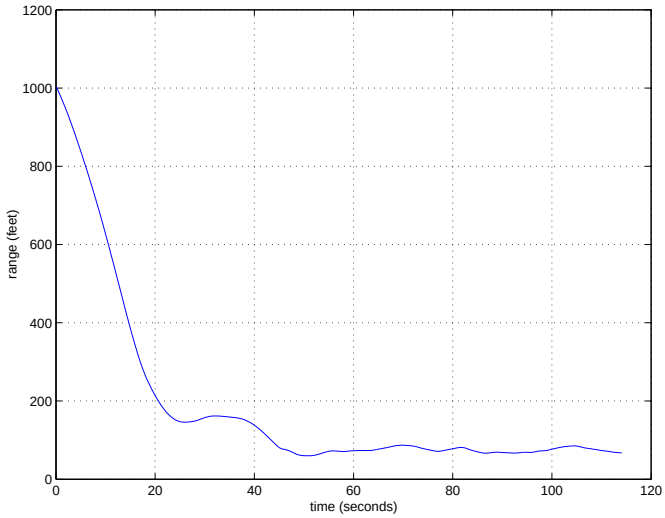

Figure 4.24: Time history of range between the piloted transfer function model and the K/A-6.

The first three runs were made with the aircraft dynamics of the transfer function model based on the nonlinear F-4 model linearized about reference conditions above. Shown in figure 4.25 are the time histories of aircraft states for run one. The pilot was able to maintain the range within ±25 feet, which falls inside the adequate parameters. Note that the aircraft states and accelerations were held fairly constant throughout the task, as is desired. The angle of attack, sideslip, and bank angle varied by ±4°, ±1°, and ±5° about the trim condition. The test pilot noted that this setup was "easy to fly" out far, but increasingly difficult in close due to the lack of acceleration cues from the motion system. The pilot also mentioned that the task seemed to take a long time. The Cooper-Harper handling qualities rating assigned by the pilot were 5, 6, and 8 for the three runs. Definitions of these ratings can be seen above in figure 4.1.

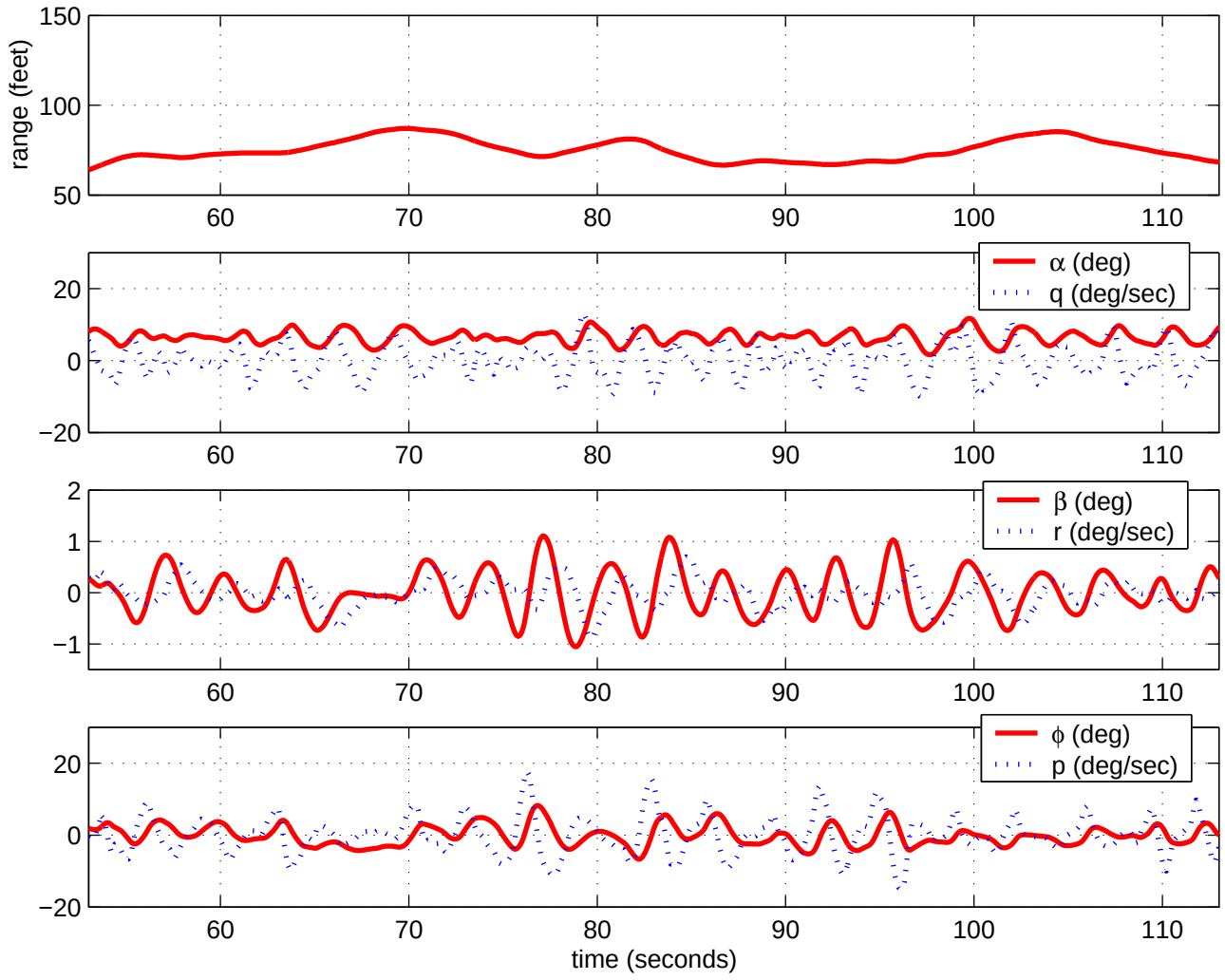

Figure 4.25: Time history of aircraft states with the transfer function model based on the nonlinear F-4 model linearized about reference conditions, parameters include $\zeta_{sp} = 0.614$ and $\omega_{np} = 0.0751$.

Runs four, five, and six were made with the aircraft dynamics of the transfer function model based on the nonlinear F-4 model linearized about reference conditions with the short period damping ratio decreased from 0.614 to 0.100. Figure 4.26 displays the aircraft state time histories for run six. Note that the pilot was unable to perform the task for the full 60

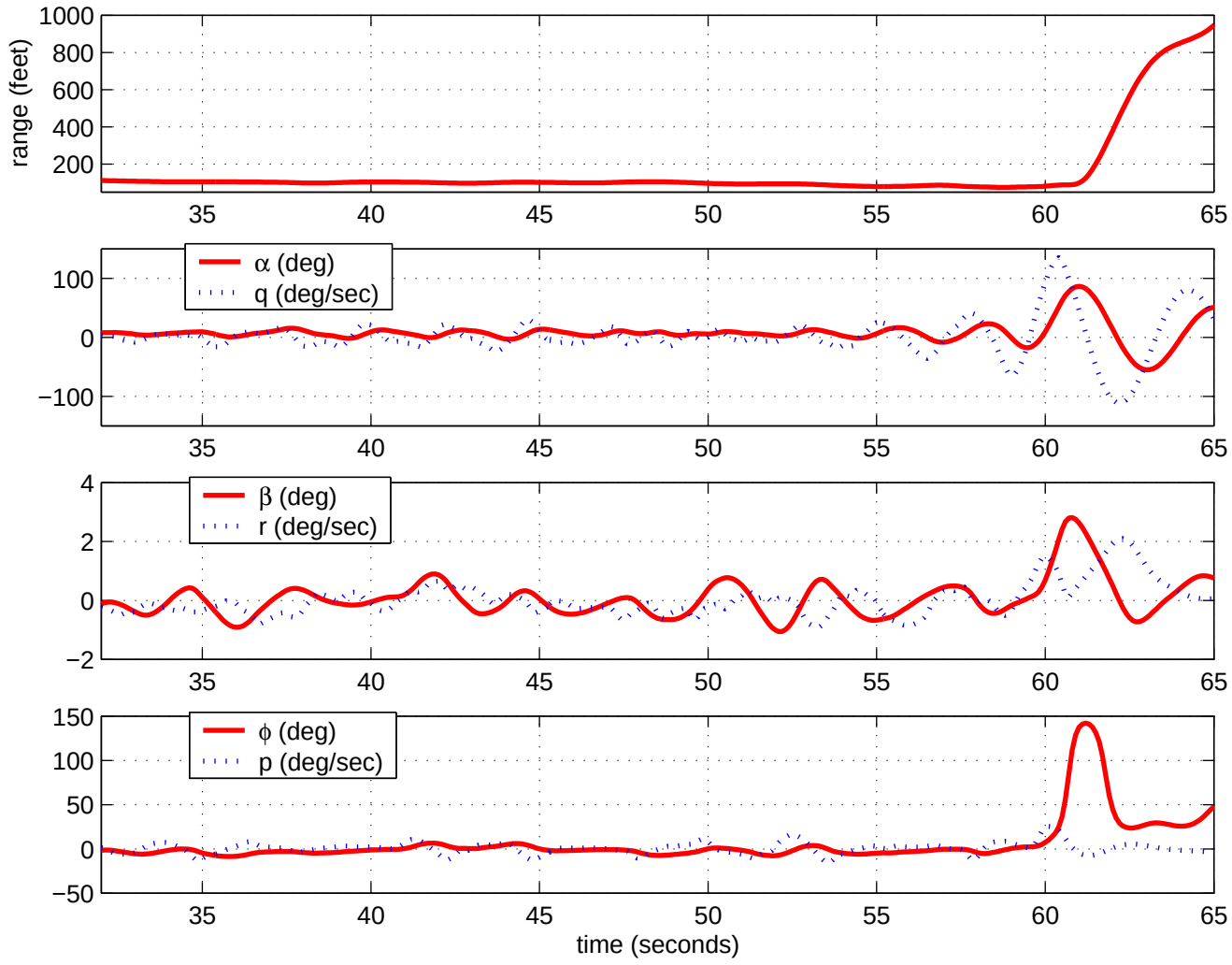

Figure 4.26: Time history of aircraft states with the modified transfer function model based on the nonlinear F-4 model linearized about reference conditions, parameters include $\omega_{np} = 0.0751$ with $\zeta_{sp} = 0.100$.

seconds, as the pilot lost sight of the tanker completely. This occurred in each of the three flights at this condition. The range was held within ±30 feet for the first 30 seconds, then the aircraft departed. This can be accounted for by the angle of attack. Starting between 50 and 55 seconds, the pilot entered a PIO (pilot induced oscillation) resulting from the

decreased short period damping, causing the angle of attack to depart from the reference conditions. Therefore, this configuration was deemed a 10 on the Cooper-Harper scale. The pilot mentioned that pitch was "easy to control" before engagement, but in close, the PIO was well developed.

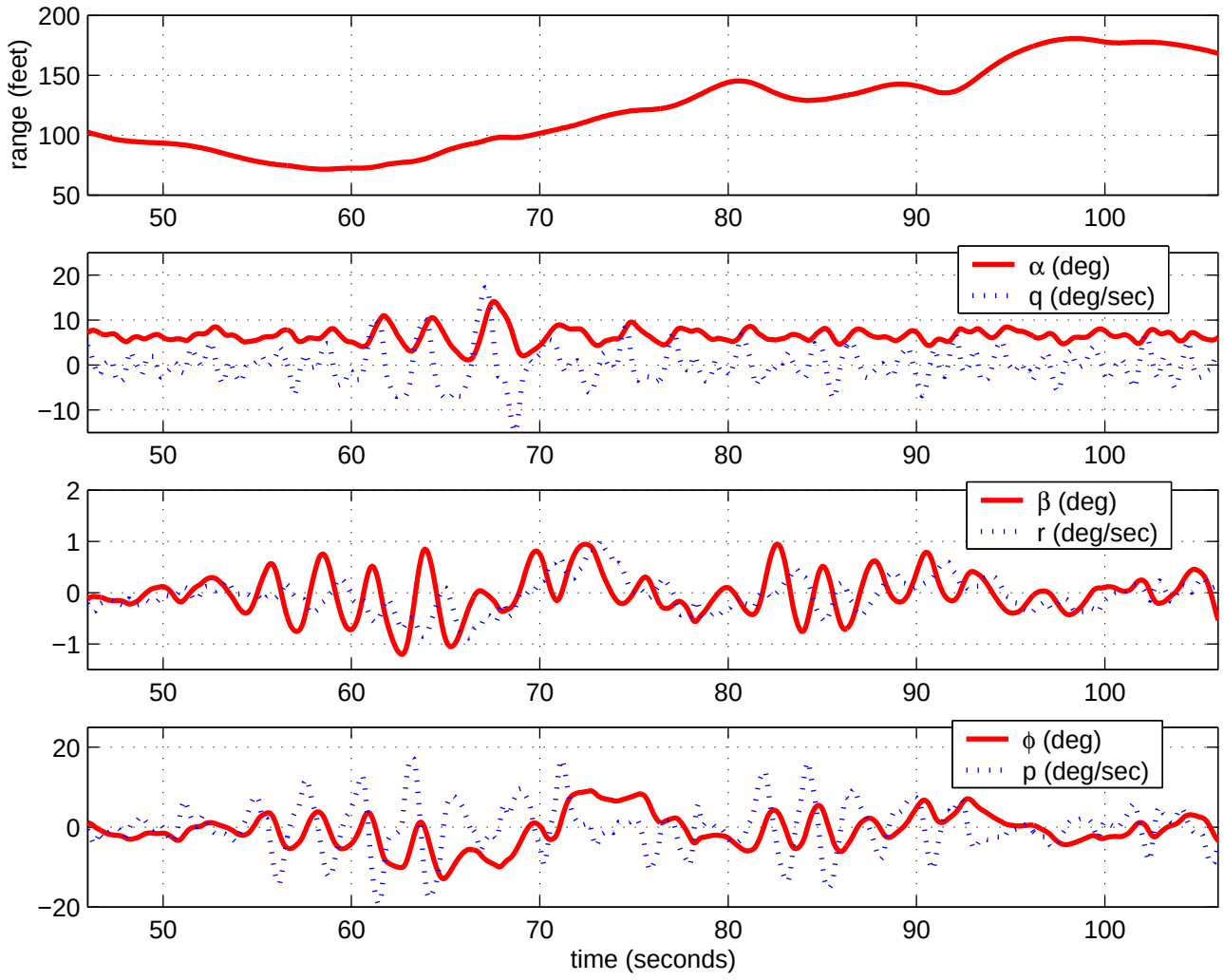

Figure 4.27: Time history of aircraft states with the modified transfer function model based on the nonlinear F-4 model linearized about reference conditions, parameters include $\zeta_{sp} = 0.614$ with $\omega_{np} = 0.200$.

The last three runs were completed with aircraft dynamics of the transfer function model based on the nonlinear F-4 model linearized about reference conditions with the phugoid natural frequency changed from 0.0751 to 0.200. Notice the large changes in range over time. The pilot noted that the aircraft was "nice in pitch". However, the closure rate and speed were difficult to control as they required large throttle corrections and the aircraft displayed some "lateral squirreliness". There was no PIO tendency evident. The pilot rated the three runs as 7, 7, and 6, but mentioned that the aircraft would have received a 3 or 4 if it were not for the throttle.

These simulation test flights demonstrated the real-time variable stability capabilities of this model.

Chapter 5

Summary and Conclusions

A variable stability simulation model was created based on 46 stability parameters, including natural frequencies, damping ratios, time constants, and gains. A ground based variable stability simulation is very versatile, as it can model any aircraft independent of aerodynamics. Such a simulation was obtained using transfer functions representing the aircraft state responses to control inputs. These transfer functions were converted into state space systems used to create the linear equations for the model. As an example application, a nonlinear F-4 Phantom model was linearized about reference conditions and used to solve for numerical values for the stability parameters.

The model was first developed as a desktop simulation and then converted into *CAS-TLE* structure for use with Virginia Tech's 2F122A flight simulator. This conversion required a simple dynamic inversion of the body axis force and moment terms. To reduce the error

in these terms, a model following scheme was incorporated.

A series of canned inputs were used to compare the transfer function model to the nonlinear F-4 model. While the response of the states of the transfer function model did not track those of the nonlinear model exactly, the trends were the same. These comparisons were used to show that the transfer function model was indeed a reasonable representation of the F-4, and thus the transfer function model can be used to model the flying qualities of other aircraft.

Real-time, pilot-in-the-loop testing was then performed to evaluate the variable stability capabilities of the model. A formation flying task was selected as this is a commonly encountered task and it requires steady flight and small control inputs, so it is ideal for a linear model. The task was flown for a series of aircraft dynamics. Each model was evaluated using pilot comments, Cooper-Harper handling qualities ratings, and data collected from the simulation. This showed the variable stability capabilities of the simulation model.

Results in this paper have demonstrated the successful creation of a variable stability simulation model. The model can be used to model a wide range of aircraft or to explore changes in the dynamics of a single aircraft. Thus, one simulator can be used to compare many aircraft or to investigate design changes to an aircraft.

Bibliography

[1] "VISTA In-Flight", *United States Air Force Test Pilot School website,* 10 May 2001, <http://www.edwards.af.mil/tps/VISTA/vista.htm>, (08 April 2002)

[2] Minor, John, Thurling, Andrew J., Ohmit, Eric, "VISTA - A 21^{st} Century UAV Testbed", *United States Air Force Test Pilot School website,* 10 May 2001, <http://www.edwards.af.mil/tps/VISTA/VISTA-21st%20Century%20UAV%20Testbed.pdf>, (17 April 2002)

[3] Durham, W. C., "AOE 5214: Aircraft Dynamics and Control", *Class notes from Fall 2000,* Aerospace and Ocean Engineering Department, Virginia Polytechnic Institute and State University, 2000.

[4] Stevens, Brian L. and Frank L. Lewis, *Aircraft Control and Simulation,* John Wiley & Sons, Inc., New York, 1992.

[5] Ogata, Katsuhiko, *System Dynamics, Second Edition,* Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1992, pp. 175-177.

[6] Hewett, Marle D., "Stability and Control of Aerospace Systems", *Course notes for Flight Dynamics,* U.S. Naval Postgraduate School, 1984.

[7] Nichols, James H., Magyar, Thomas J. and Schug, Eric C., "The Platform-Independent Aircraft Simulation Environment at Manned Flight Simulator", AIAA-98-4179,1998.

[8] Scalera, Kevin R., and Durham, W. C., "Modification of a Surplus Navy 2F122A A-6E OFT for Flight Dynamics Research and Instruction", AIAA-98-4180, 1998.

[9] Durham, W. C., *Contributions to Model Following Control Theory,* Ph.D. Dissertation, Virginia Polytechnic Institute and State University, Blacksburg, VA 1989.

[10] Hodgkinson, John, *Aircraft Handling Qualities,* American Institute of Aeronautics and Astronautics, Inc., Reston, Virginia, 1999, pp. 1-19.

[11] "Cooper-Harper Rating Scale", *United States Navy Flight Test website,* 24 September 1997, <http://flighttest.navair.navy.mil/unrestricted/ch.pdf>, (08 April 2002)

[12] Grantham, William D., "Comparison of Flying Qualities Derived From In-Flight and Ground-Based Simulators for a Jet-Transport Airplane for the Approach and Landing Pilot Tasks", NASA Technical Paper 2962, Langley Research Center, Hampton, Virginia, 1989.

Appendix A

Simulation Code

A.1 TransferFunctions.f

A.1.1 Description

TransferFunctions.f is the main Fortran program which executes the simulation, it is called at each time step by Control.f, the control subroutine. It sets constants for the 46 stability parameters and sets the initial conditions. TransferFunctions.f calculates the values for the outputs (y), their time derivatives ($\dot{y}$), the time derivatives of the states ($\dot{x}$), the forces (X, Y, Z), and the moments (L, M, N) after each iteration of the simulation based on equations derived above. TransferFunction.f also performs a numerical integration at the end of each iteration to calculate an updated value for the states (x) for the next time step.

A.1.2 Code

```
C*************************************************************************
C
C   TITLE:      TransferFunctions
C
C-------------------------------------------------------------------------
C
C   FUNCTION:   Subroutine to determine the aircraft dynamics from the
C               derived transfer functions.
C
C-------------------------------------------------------------------------
C
C   MODULE TYPE:    GENERIC
C
C-------------------------------------------------------------------------
C
C   DESIGNED BY:    Henrik Pettersson
C
C   CODED BY:       Henrik Pettersson
C
C   MAINTAINED BY:
C
C-------------------------------------------------------------------------
C
C   MODIFICATION HISTORY:
C
C   DATE            PURPOSE                             BY
C   ====            =======                             ==
C   16 April 2002   Created                             HP
C   22 May   2002   Update with model following in      HP
C                   force and moment equations
C
C-------------------------------------------------------------------------
C
C               GLOSSARY
C               ========
C
C   ASSIGNMENTS:
C
C  NONE
C
```

```
C-------------------------------------------------------------------------
C
C   INPUTS:
C
C  ALFAR      Angle Of Attack                          RAD          ANU
C  BETAR      Sideslip Angle                           RAD          ANL
C  DA         aileron deflection                       RADIANS      RWD
C  DE         elevator deflection                      RAD          TEU
C  DR         rudder deflection                        RADIANS      TEL
C  DT         Simulation time step (frame time)        SEC          --------
C  DTH        throttle (0 to +100)                     (0-1)        --------
C  G          Acceleration Due To Gravity              FT/S2        --------
C  IMODE      Sim. mode: -2=init,-1=reset,0=hold,1=ru  --------     --------
C  KALPHADE   gain for dalpha to de TF                 None         None
C  KALPHADT   gain for dalpha to dth TF                None         None
C  KBETADA    gain for dbeta to da TF                  None         None
C  KBETADR    gain for dbeta to dr TF                  None         None
C  KPHIDA     gain for dphi to da TF                   None         None
C  KPHIDR     gain for dphi to dr TF                   None         None
C  KPSIDA     gain for dpsi to da TF                   None         None
C  KPSIDR     gain for dpsi to dr TF                   --------     --------
C  KTHETADE   gain for dtheta to de TF                 None         None
C  KTHETADT   gain for dtheta to dth TF                None         None
C  KUDE       gain for dU to de TF                     None         None
C  KUDT       gain for dU to dth TF                    None         None
C  OMEGAALPHA natural frequency from dalpha to de TF   SEC-1        None
C  OMEGAD     dutch roll natural frequency             SEC-1        None
C  OMEGAP     phugoid natural frequency                SEC-1        None
C  OMEGAPHIDA natural frequency from dphi to da TF     SEC-1        None
C  OMEGAPSIDA natural frequency from dpsi to da TF     SEC-1        None
C  OMEGAPSIDR natural frequency from dpsi to dr TF     SEC-1        None
C  OMEGASP    short period natural frequency           SEC-1        None
C  OMEGAU     natural frequency from dU to de TF       SEC-1        None
C  OMEGAUDT   natural frequency from dU to dth TF      SEC-1        None
C  PB         Aircraft Roll Velocity, Body Frame       RAD/SEC      RWD
C  PHIR       Roll Angle, L-frame                      RAD          RWD
C  PSIR       Yaw Angle, L-frame                       RAD          ANR
C  QB         Aircraft Pitch Velocity, B-frame         RAD/SEC      ANU
C  RB         Aircraft Yaw Velocity, B-frame           RAD/SEC      ANR
C  TALPHA     time constant from dalpha to de TF       SEC-1        None
C  TALPHADT   time constant from dalpha to dt TF       --------     --------
C  TBETADA1   1st time constant from dbeta to da TF    SEC-1        None
```

```
C  TBETADA2   2nd time constant from dbeta to da TF    SEC-1        None
C  TBETADR1   1st time constant from dbeta to dr TF    SEC-1        None
C  TBETADR2   2nd time constant from dbeta to dr TF    SEC-1        None
C  TBETADR3   3rd time constant from dbeta to dr TF    SEC-1        None
C  THETR      Pitch Angle, L-frame                     RAD          ANU
C  TPHIDR1    1st time constant from dphi to dr TF     SEC-1        None
C  TPHIDR2    2nd time constant from dphi to dr TF     SEC-1        None
C  TPSIDA     time constant from dpsi to da TF         SEC-1        None
C  TPSIDR     time constant from dpsi to dr TF         SEC-1        None
C  TR         roll mode time constant                  SEC-1        None
C  TS         spiral mode time constant                SEC-1        None
C  TTHETA1    1st time constant from dtheta to de TF   SEC-1        None
C  TTHETA2    2nd time constant from dtheta to de TF   SEC-1        None
C  TU         time constant from dU to de TF           SEC-1        None
C  VRW        Velocity w.r.t. wind (true airspeed)     FT/SEC       N/A
C  XIXX       Vehicle X - Moment Of Inertia about Cg   SLUG-FT2     N/A
C  XIXZ       Vehicle Xz Prod Of Inertia about Cg      SLUG-FT2     N/A
C  XIYY       Vehicle Y - Moment Of Inertia about Cg   SLUG-FT2     N/A
C  XIZZ       Vehicle Z - Moment Of Inertia about Cg   SLUG-FT2     N/A
C  XMASS      Mass Of Vehicle (incl Fuel)              SLUG         N/A
C  ZETAALPHA  damping ratio from dalpha to de TF       None         None
C  ZETAD      dutch roll damping ratio                 None         None
C  ZETAP      phugoid damping ratio                    None         None
C  ZETAPHIDA  damping ratio from dphi to da TF         None         None
C  ZETAPSIDA  damping ratio from dpsi to da TF         None         None
C  ZETAPSIDR  damping ratio from dpsi to dr TF         None         None
C  ZETASP     short period damping ratio               None         None
C  ZETAU      damping ratio from dU to de TF           None         None
C  ZETAUDT    damping ratio from dU to dth TF          None         None
C
C-------------------------------------------------------------------------
C
C   OUTPUTS:
C
C  FAX        Total Aero Force, X-body                 LB           FWD
C  FAY        Total Aero Force, Y-body                 LB           RT
C  FAZ        Total Aero Force, Z-body                 LB           DOWN
C  TAL        Aero Rolling moment, X-body              FT-LB        RWD
C  TAM        Aero pitch moment, Y-body                FT-LB        ANU
C  TAN        Aero Yawing moment, Z-body               FT-LB        ANR
C
C-------------------------------------------------------------------------
```

```
C
C   LOCALS:
C
C  DAREF      reference aileron deflection           --------     --------
C  DDA        change in aileron deflection           --------     --------
C  DDE        change in elevator deflection          --------     --------
C  DDR        change in rudder deflection            --------     --------
C  DDTH       change in throttle position            --------     --------
C  DEREF      reference elevator deflection          --------     --------
C  DRREF      reference rudder deflection            --------     --------
C  DTHREF     reference throttle position            --------     --------
C  LAMDA      model following gain                   --------     --------
C  U1         temporary U for calculation            --------     --------
C  V1         temporary V for calculation            --------     --------
C  W1         temporary W for calculation            --------     --------
C  X          states                                 --------     --------
C  XD         time derivatives of states             --------     --------
C  Y          outputs                                --------     --------
C  YD         time derivatives of outputs            --------     --------
C  YDEG       outputs (in degrees)                   --------     --------
C  YREF       reference outputs                      --------     --------
C
C-------------------------------------------------------------------------

    SUBROUTINE  TRANSFERFUNCTIONS

C-------------------------------------------------------------------------
C
C   DECLARATION SECTION
C
C-------------------------------------------------------------------------
    IMPLICIT NONE

C       ** INPUTS:

    INTEGER*4 IMODE
    REAL*4 BETAR, DA, DE, DR, DT, DTH, G, ALFAR, KALPHADE, KALPHADT
    REAL*4 KBETADA, KBETADR, KPHIDA, KPHIDR, KPSIDA, KPSIDR
    REAL*4 KTHETADE, KTHETADT, KUDE, KUDT, OMEGAALPHA, OMEGAD
    REAL*4 OMEGAP, OMEGAPHIDA, OMEGAPSIDA, OMEGAPSIDR, OMEGASP
    REAL*4 OMEGAU, OMEGAUDT, PB, PHIR, PSIR, QB, RB, TALPHA
    REAL*4 TALPHADT, TBETADA1, TBETADA2, TBETADR1, TBETADR2
```

```
      REAL*4 TBETADR3, THETR, TPHIDR1, TPHIDR2, TPSIDA, TPSIDR, TR
      REAL*4 TS, TTHETA1, TTHETA2, TU, VRW, XIXX, XIXZ, XIYY, XIZZ
      REAL*4 XMASS, ZETAALPHA, ZETAD, ZETAP, ZETAPHIDA, ZETAPSIDA
      REAL*4 ZETAPSIDR, ZETASP, ZETAU, ZETAUDT

C       ** OUTPUTS:

      REAL*4 FAX, FAY, FAZ, TAL, TAM, TAN

C       ** LOCALS:

      REAL*4 DAREF, DDA, DDE, DDR, DDTH, DEREF, DRREF, DTHREF, LAMDA
      REAL*4 U1, V1, W1, X(21), XD(21), Y(12), YD(12), YDEG(12)
      REAL*4 YREF(12)

C       ** OTHER LOCALS:

      BYTE ATMPAR, CONPAR, KONSTANTS, LCLBUF, SHIPPAR, SIMPAR
      BYTE TFMODPAR
      LOGICAL*4 INITIALIZED, PILOTED, DONE
      INTEGER*4 I
      REAL*4 GTH, GR, DRLIM, D2R, LATLIM, LATSTICK, DALIM, LONLIMAFT
      REAL*4 LONLIMFOR, LONSTICK, GA, RUDPED, RUDPEDLIM, DELIMTED
      REAL*4 DELIMTEU, GE, THROTTLE, VI, ZETAPHIDR

C-------------------------------------------------------------------------
C
C   COMMON SECTION
C
C-------------------------------------------------------------------------

      COMMON/ SHELL1 / CONPAR(424)

      COMMON/ KONSTANT / KONSTANTS(40)

      COMMON/ SHELL2 / SIMPAR(1024)

      COMMON/ BK_TRANSFERFUNCTIONS / LCLBUF(408)

      COMMON/ XATMOS / ATMPAR(908)

      COMMON/ XSHIP / SHIPPAR(1464)
```

```
    COMMON/ XTFMOD / TFMODPAR(248)

C-------------------------------------------------------------------------------
C
C   EQUIVALENCE SECTION
C
C-------------------------------------------------------------------------------

C       ** INPUTS:

    EQUIVALENCE( SIMPAR(105), ALFAR)
    EQUIVALENCE( SIMPAR(109), BETAR)
    EQUIVALENCE( TFMODPAR(5), DA)
    EQUIVALENCE( TFMODPAR(9), DE)
    EQUIVALENCE( TFMODPAR(13), DR)
    EQUIVALENCE( CONPAR(145), DT)
    EQUIVALENCE( TFMODPAR(1), DTH)
    EQUIVALENCE( KONSTANTS(1), G)
    EQUIVALENCE( CONPAR(1), IMODE)
    EQUIVALENCE( TFMODPAR(21), KALPHADE)
    EQUIVALENCE( TFMODPAR(53), KALPHADT)
    EQUIVALENCE( TFMODPAR(25), KBETADA)
    EQUIVALENCE( TFMODPAR(29), KBETADR)
    EQUIVALENCE( TFMODPAR(33), KPHIDA)
    EQUIVALENCE( TFMODPAR(37), KPHIDR)
    EQUIVALENCE( TFMODPAR(45), KPSIDA)
    EQUIVALENCE( TFMODPAR(241), KPSIDR)
    EQUIVALENCE( TFMODPAR(41), KTHETADE)
    EQUIVALENCE( TFMODPAR(57), KTHETADT)
    EQUIVALENCE( TFMODPAR(17), KUDE)
    EQUIVALENCE( TFMODPAR(49), KUDT)
    EQUIVALENCE( TFMODPAR(133), OMEGAALPHA)
    EQUIVALENCE( TFMODPAR(121), OMEGAD)
    EQUIVALENCE( TFMODPAR(125), OMEGAP)
    EQUIVALENCE( TFMODPAR(141), OMEGAPHIDA)
    EQUIVALENCE( TFMODPAR(145), OMEGAPSIDA)
    EQUIVALENCE( TFMODPAR(149), OMEGAPSIDR)
    EQUIVALENCE( TFMODPAR(129), OMEGASP)
    EQUIVALENCE( TFMODPAR(137), OMEGAU)
    EQUIVALENCE( TFMODPAR(153), OMEGAUDT)
    EQUIVALENCE( SIMPAR(145), PB)
```

```
      EQUIVALENCE( SIMPAR(13), PHIR)
      EQUIVALENCE( SIMPAR(21), PSIR)
      EQUIVALENCE( SIMPAR(149), QB)
      EQUIVALENCE( SIMPAR(153), RB)
      EQUIVALENCE( TFMODPAR(73), TALPHA)
      EQUIVALENCE( TFMODPAR(245), TALPHADT)
      EQUIVALENCE( TFMODPAR(77), TBETADA1)
      EQUIVALENCE( TFMODPAR(81), TBETADA2)
      EQUIVALENCE( TFMODPAR(85), TBETADR1)
      EQUIVALENCE( TFMODPAR(89), TBETADR2)
      EQUIVALENCE( TFMODPAR(93), TBETADR3)
      EQUIVALENCE( SIMPAR(17), THETR)
      EQUIVALENCE( TFMODPAR(97), TPHIDR1)
      EQUIVALENCE( TFMODPAR(101), TPHIDR2)
      EQUIVALENCE( TFMODPAR(113), TPSIDA)
      EQUIVALENCE( TFMODPAR(117), TPSIDR)
      EQUIVALENCE( TFMODPAR(65), TR)
      EQUIVALENCE( TFMODPAR(69), TS)
      EQUIVALENCE( TFMODPAR(105), TTHETA1)
      EQUIVALENCE( TFMODPAR(109), TTHETA2)
      EQUIVALENCE( TFMODPAR(61), TU)
      EQUIVALENCE( SIMPAR(205), VRW)
      EQUIVALENCE( SIMPAR(345), XIXX)
      EQUIVALENCE( SIMPAR(357), XIXZ)
      EQUIVALENCE( SIMPAR(349), XIYY)
      EQUIVALENCE( SIMPAR(353), XIZZ)
      EQUIVALENCE( SIMPAR(369), XMASS)
      EQUIVALENCE( TFMODPAR(169), ZETAALPHA)
      EQUIVALENCE( TFMODPAR(157), ZETAD)
      EQUIVALENCE( TFMODPAR(161), ZETAP)
      EQUIVALENCE( TFMODPAR(173), ZETAPHIDA)
      EQUIVALENCE( TFMODPAR(177), ZETAPSIDA)
      EQUIVALENCE( TFMODPAR(181), ZETAPSIDR)
      EQUIVALENCE( TFMODPAR(165), ZETASP)
      EQUIVALENCE( TFMODPAR(185), ZETAU)
      EQUIVALENCE( TFMODPAR(189), ZETAUDT)

C       ** OUTPUTS:

      EQUIVALENCE( SIMPAR(393), FAX)
      EQUIVALENCE( SIMPAR(397), FAY)
      EQUIVALENCE( SIMPAR(401), FAZ)
```

```
      EQUIVALENCE( SIMPAR(469), TAL)
      EQUIVALENCE( SIMPAR(473), TAM)
      EQUIVALENCE( SIMPAR(477), TAN)

C       ** LOCALS:

      EQUIVALENCE( LCLBUF(1), DAREF)
      EQUIVALENCE( LCLBUF(5), DDA)
      EQUIVALENCE( LCLBUF(9), DDE)
      EQUIVALENCE( LCLBUF(13), DDR)
      EQUIVALENCE( LCLBUF(17), DDTH)
      EQUIVALENCE( LCLBUF(21), DEREF)
      EQUIVALENCE( LCLBUF(25), DRREF)
      EQUIVALENCE( LCLBUF(29), DTHREF)
      EQUIVALENCE( LCLBUF(33), U1)
      EQUIVALENCE( LCLBUF(37), V1)
      EQUIVALENCE( LCLBUF(41), W1)
      EQUIVALENCE( LCLBUF(45), X)
      EQUIVALENCE( LCLBUF(129), XD)
      EQUIVALENCE( LCLBUF(213), Y)
      EQUIVALENCE( LCLBUF(261), YD)
      EQUIVALENCE( LCLBUF(309), YDEG)
      EQUIVALENCE( LCLBUF(357), YREF)
      EQUIVALENCE( LCLBUF(405), LAMDA)

C-------------------------------------------------------------------------
C
C   INITIALIZATION SECTION
C
C-------------------------------------------------------------------------

      IF( IMODE.LE.-2 .OR. .NOT. Initialized ) THEN

      KUDE=0.08704
      KALPHADE=-0.001333
      KBETADA=0.01126
      KBETADR=0.0002073
      KPHIDA=0.1668
      KPHIDR=0.2479
      KTHETADE=-0.1389
      KPSIDA=0.005918
      KPSIDR=-0.02947
```

```
TU=2.553
TR=0.5275
TS=4.916
TALPHA=0.009594
TBETADA1=0.4039
TBETADA2=2.399
TBETADR1=0.003754
TBETADR2=0.5190
TBETADR3=6.3833
TPHIDR1=0.6204
TPHIDR2=-2.201
TTHETA1=1.7162
TTHETA2=435.64
TPSIDA=1.7657
TPSIDR=0.4564
OMEGAD=1.726
OMEGAP=0.0751
OMEGASP=1.550
OMEGAALPHA=0.09065
OMEGAU=8.690
OMEGAPHIDA=1.3767
OMEGAPSIDA=2.3689
OMEGAPSIDR=0.4557
ZETAD=0.2488
ZETAP=0.0172
ZETASP=0.6140
ZETAALPHA=0.004569
ZETAPHIDA=0.4399
ZETAPSIDA=-0.208
ZETAPSIDR=0.4000
ZETAU=0.09890
KUDT=30.209
OMEGAUDT=1.5473
ZETAUDT=0.61246
KALPHADT=-0.0077205
TALPHADT=0.76711
KTHETADT=0.012681

LAMDA=10

ENDIF
```

```
C-------------------------------------------------------------------------------
C
C   RESET SECTION
C
C-------------------------------------------------------------------------------

    IF( IMODE.LT.0 .OR. .NOT. Initialized ) THEN

    Initialized = .TRUE.

C   X(1,2,3,4) = LONGITUDINAL STATES DUE TO THROTTLE
C   X(5,6,7,8) = LONGITUDINAL STATES DUE TO ELEVATOR
C   X(9,10,11,12,13) = LATERAL DIRECTIONAL STATES DUE TO AILERON
C   X(14,15,16,17,18) = LATERAL DIRECTIONAL STATES DUE TO RUDDER
C   X(19) = NORTH POSITION (FT)
C   X(20) = EAST POSITION (FT)
C   X(21) = ALTITUDE (FT)

    X(1) = 0
    X(2) = 0
    X(3) = 0
    X(4) = 0
    X(5) = 0
    X(6) = 0
    X(7) = 0
    X(8) = 0
    X(9) = 0
    X(10) = 0
    X(11) = 0
    X(12) = 0
    X(13) = 0
    X(14) = 0
    X(15) = 0
    X(16) = 0
    X(17) = 0
    X(18) = 0
    X(19) = 0
    X(20) = 0
    X(21) = 0

C   XD( ) = TIME DERIVATIVES OF X( )
```

```
      XD(1) = 0
      XD(2) = 0
      XD(3) = 0
      XD(4) = 0
      XD(5) = 0
      XD(6) = 0
      XD(7) = 0
      XD(8) = 0
      XD(9) = 0
      XD(10) = 0
      XD(11) = 0
      XD(12) = 0
      XD(13) = 0
      XD(14) = 0
      XD(15) = 0
      XD(16) = 0
      XD(17) = 0
      XD(18) = 0
      XD(19) = 0
      XD(20) = 0
      XD(21) = 0

C     YREF( ) = REFERENCE OUTPUTS

      YREF(1) = 500
      YREF(2) = 5.8/57.3
      YREF(3) = 0
      YREF(4) = 0
      YREF(5) = YREF(2)
      YREF(6) = 0
      YREF(7) = 0
      YREF(8) = 0
      YREF(9) = 0
      YREF(10) = 0
      YREF(11) = 0
      YREF(12) = 10000

C     REFERENCE CONTROL DEFLECTIONS

      DTHREF = 0.304
      DEREF = -0.12
      DAREF = 0
```

```
      DRREF = 0

      DTH = DTHREF
      DE = DEREF
      DA = DAREF
      DR = DRREF

      ENDIF

C-----------------------------------------------------------------------
C
C   RUN SECTION
C
C-----------------------------------------------------------------------

C   Change in control inputs

      DDTH = DTH - DTHREF
      DDA = DA - DAREF
      DDE = DE - DEREF
      DDR = DR - DRREF

C   Y(1) = Total Velocity (ft/sec)
C   Y(2) = alpha (deg)
C   Y(3) = beta (deg)
C   Y(4) = phi (deg)
C   Y(5) = theta (deg)
C   Y(6) = psi (deg)
C   Y(7) = p (deg/sec)
C   Y(8) = q (deg/sec)
C   Y(9) = r (deg/sec)
C   Y(10) = North position (ft)
C   Y(11) = East position (ft)
C   Y(12) = altitude (ft)
C
      Y(1)=KUDT*X(1)+2*KUDT*ZETAUDT*OMEGAUDT*X(2)
     &  +KUDT*OMEGAUDT*OMEGAUDT*X(3)
     &  +KUDE*X(5)+(2*KUDE*ZETAU*OMEGAU+KUDE/TU)*X(6)
     &  +(KUDE*OMEGAU*OMEGAU+2*KUDE*ZETAU*OMEGAU/TU)*X(7)
     &  +(KUDE*OMEGAU*OMEGAU/TU)*X(8)
C
         Y(2)=KALPHADT*X(2)+KALPHADT/TALPHADT*X(3)
```

```
     &  +KALPHADE*X(5)+(2*KALPHADE*ZETAALPHA*OMEGAALPHA
     &  +KALPHADE/TALPHA)*X(6)
     &  +(KALPHADE*OMEGAALPHA*OMEGAALPHA
     &  +2*KALPHADE*ZETAALPHA*OMEGAALPHA/TALPHA)*X(7)
     &  +(KALPHADE*OMEGAALPHA*OMEGAALPHA/TALPHA)*X(8)
C
    Y(3)=KBETADA*X(10)
     &  +(KBETADA/TBETADA1+KBETADA/TBETADA2)*X(11)
     &  +(KBETADA/TBETADA1/TBETADA2)*X(12)
     &  +KBETADR*X(14)+(KBETADR/TBETADR1+KBETADR/TBETADR2
     &  +KBETADR/TBETADR3)*X(15)
     &  +(KBETADR/TBETADR1/TBETADR2+KBETADR/TBETADR1/TBETADR3
     &  +KBETADR/TBETADR2/TBETADR3)*X(16)
     &  +(KBETADR/TBETADR1/TBETADR2/TBETADR3)*X(17)
C
    Y(4)=KPHIDA*X(10)
     &  +KPHIDA*ZETAPHIDA*OMEGAPHIDA*X(11)
     &  +KPHIDA*OMEGAPHIDA*OMEGAPHIDA*X(12)
     &  +KPHIDR*X(15)
     &  +(KPHIDR/TPHIDR1+KPHIDR/TPHIDR2)*X(16)
     &  +KPHIDR/TPHIDR1/TPHIDR2*X(17)
C
    Y(5)=KTHETADT*X(4)+KTHETADE*X(6)
     &  +(KTHETADE/TTHETA1+KTHETADE/TTHETA2)*X(7)
     &  +(KTHETADE/TTHETA1/TTHETA2)*X(8)
C
    Y(6)=KPSIDA*X(10)
     &  +(2*ZETAPSIDA*OMEGAPSIDA*KPSIDA+KPSIDA/TPSIDA)*X(11)
     &  +(OMEGAPSIDA*OMEGAPSIDA*KPSIDA
     &  +2*ZETAPSIDA*OMEGAPSIDA*KPSIDA/TPSIDA)*X(12)
     &  +(OMEGAPSIDA*OMEGAPSIDA*KPSIDA/TPSIDA)*X(13)
     &  +KPSIDR*X(15)
     &  +(2*ZETAPSIDR*OMEGAPSIDR*KPSIDR+KPSIDR/TPSIDR)*X(16)
     &  +(KPSIDR*OMEGAPSIDR*OMEGAPSIDR
     &  +2*ZETAPSIDR*OMEGAPSIDR*KPSIDR/TPSIDR)*X(17)
     &  +(OMEGAPSIDR*OMEGAPSIDR*KPSIDR/TPSIDR)*X(18)
C
    Y(7)=KPHIDA*X(9)+KPHIDA*ZETAPHIDA*OMEGAPHIDA*X(10)
     &  +KPHIDA*OMEGAPHIDA*OMEGAPHIDA*X(11)
     &  +KPHIDR*X(14)
     &  +(KPHIDR/TPHIDR1+KPHIDR/TPHIDR2)*X(15)
     &  +(KPHIDR/TPHIDR1/TPHIDR2)*X(16)
```

```
C
      Y(8)=KTHETADT*X(3)+KTHETADE*X(5)
     &  +(KTHETADE/TTHETA1+KTHETADE/TTHETA2)*X(6)
     &  +(KTHETADE/TTHETA1/TTHETA2)*X(7)
C
      Y(9)=KPSIDA*X(9)
     &  +(2*ZETAPSIDA*OMEGAPSIDA*KPSIDA+KPSIDA/TPSIDA)*X(10)
     &  +(OMEGAPSIDA*OMEGAPSIDA*KPSIDA
     &  +2*ZETAPSIDA*OMEGAPSIDA*KPSIDA/TPSIDA)*X(11)
     &  +(OMEGAPSIDA*OMEGAPSIDA*KPSIDA/TPSIDA)*X(12)
     &  +KPSIDR*X(14)
     &  +(2*ZETAPSIDR*OMEGAPSIDR*KPSIDR+KPSIDR/TPSIDR)*X(15)
     &  +(KPSIDR*OMEGAPSIDR*OMEGAPSIDR
     &  +2*ZETAPSIDR*OMEGAPSIDR*KPSIDR/TPSIDR)*X(16)
     &  +(OMEGAPSIDR*OMEGAPSIDR*KPSIDR/TPSIDR)*X(17)
C
      Y(10) = X(19)
      Y(11) = X(20)
      Y(12) = X(21)
C
      Y(1) = Y(1) + YREF(1)
      Y(2) = Y(2) + YREF(2)
      Y(3) = Y(3) + YREF(3)
      Y(4) = Y(4) + YREF(4)
      Y(5) = Y(5) + YREF(5)
      Y(6) = Y(6) + YREF(6)
      Y(7) = Y(7) + YREF(7)
      Y(8) = Y(8) + YREF(8)
      Y(9) = Y(9) + YREF(9)
      Y(10) = Y(10) + YREF(10)
      Y(11) = Y(11) + YREF(11)
      Y(12) = Y(12) + YREF(12)
C
C     calculate the time derivatives of the output variables
C
      YD(1)=KUDT*XD(1)+2*KUDT*ZETAUDT*OMEGAUDT*XD(2)
     &  +KUDT*OMEGAUDT*OMEGAUDT*XD(3)
     &  +KUDE*XD(5)+(2*KUDE*ZETAU*OMEGAU+KUDE/TU)*XD(6)
     &  +(KUDE*OMEGAU*OMEGAU+2*KUDE*ZETAU*OMEGAU/TU)*XD(7)
     &  +(KUDE*OMEGAU*OMEGAU/TU)*XD(8)
C
      YD(2)=KALPHADT*XD(2)+KALPHADT/TALPHADT*XD(3)
```

```
     &  +KALPHADE*XD(5)+(2*KALPHADE*ZETAALPHA*OMEGAALPHA
     &  +KALPHADE/TALPHA)*XD(6)
     &  +(KALPHADE*OMEGAALPHA*OMEGAALPHA
     &  +2*KALPHADE*ZETAALPHA*OMEGAALPHA/TALPHA)*XD(7)
     &  +(KALPHADE*OMEGAALPHA*OMEGAALPHA/TALPHA)*XD(8)
C
      YD(3)=KBETADA*XD(10)
     &  +(KBETADA/TBETADA1+KBETADA/TBETADA2)*XD(11)
     &  +(KBETADA/TBETADA1/TBETADA2)*XD(12)
     &  +KBETADR*XD(14)+(KBETADR/TBETADR1+KBETADR/TBETADR2
     &  +KBETADR/TBETADR3)*XD(15)
     &  +(KBETADR/TBETADR1/TBETADR2+KBETADR/TBETADR1/TBETADR3
     &  +KBETADR/TBETADR2/TBETADR3)*XD(16)
     &  +(KBETADR/TBETADR1/TBETADR2/TBETADR3)*XD(17)
C
      YD(7)=KPHIDA*XD(9)
     &  +KPHIDA*ZETAPHIDA*OMEGAPHIDA*XD(10)
     &  +KPHIDA*OMEGAPHIDA*OMEGAPHIDA*XD(11)
     &  +KPHIDR*XD(14)
     &  +(KPHIDR/TPHIDR1+KPHIDR/TPHIDR2)*XD(15)
     &  +(KPHIDR/TPHIDR1/TPHIDR2)*XD(16)
C
      YD(8)=KTHETADT*XD(3)+KTHETADE*XD(5)
     &  +(KTHETADE/TTHETA1+KTHETADE/TTHETA2)*XD(6)
     &  +(KTHETADE/TTHETA1/TTHETA2)*XD(7)
C
      YD(9)=KPSIDA*XD(9)
     &  +(2*ZETAPSIDA*OMEGAPSIDA*KPSIDA+KPSIDA/TPSIDA)*XD(10)
     &  +(OMEGAPSIDA*OMEGAPSIDA*KPSIDA
     &  +2*ZETAPSIDA*OMEGAPSIDA*KPSIDA/TPSIDA)*XD(11)
     &  +(OMEGAPSIDA*OMEGAPSIDA*KPSIDA/TPSIDA)*XD(12)
     &  +KPSIDR*XD(14)
     &  +(2*ZETAPSIDR*OMEGAPSIDR*KPSIDR+KPSIDR/TPSIDR)*XD(15)
     &  +(OMEGAPSIDR*OMEGAPSIDR*KPSIDR
     &  +2*ZETAPSIDR*OMEGAPSIDR*KPSIDR/TPSIDR)*XD(16)
     &  +(OMEGAPSIDR*OMEGAPSIDR*KPSIDR/TPSIDR)*XD(17)
C
C   convert from radians to degrees
C
      YDEG(1) = Y(1)
      YDEG(2) = Y(2) * 57.3
      YDEG(3) = Y(3) * 57.3
```

```
      YDEG(4) = Y(4) * 57.3
      YDEG(5) = Y(5) * 57.3
      YDEG(6) = Y(6) * 57.3
      YDEG(7) = Y(7) * 57.3
      YDEG(8) = Y(8) * 57.3
      YDEG(9) = Y(9) * 57.3
      YDEG(10) = Y(10)
      YDEG(11) = Y(11)
      YDEG(12) = Y(12)
C
C     calculate the forces and moments
C
      FAX = XMASS*(YD(1)*COS(Y(2))*COS(Y(3))-Y(1)*YD(2)*COS(Y(3))*SIN(Y(2))
     &  -Y(1)*YD(3)*COS(Y(2))*SIN(Y(3)))+XMASS*G*SIN(Y(5))
     &  -XMASS*Y(9)*Y(1)*SIN(Y(3))+XMASS*Y(8)*Y(1)*SIN(Y(2))*COS(Y(3))
     &  +XMASS*LAMDA*(Y(1)*COS(Y(2))*COS(Y(3))-VRW*COS(ALFAR)*COS(BETAR))
C
      FAY = XMASS*(YD(1)*SIN(Y(3))+Y(1)*YD(3)*COS(Y(3)))
     &  -XMASS*G*SIN(Y(4))*COS(Y(5))-XMASS*Y(7)*Y(1)*SIN(Y(2))*COS(Y(3))
     &  +XMASS*Y(9)*Y(1)*COS(Y(2))*COS(Y(3))
     &  +XMASS*LAMDA*(Y(1)*SIN(Y(3))-VRW*SIN(BETAR))
C
      FAZ = XMASS*(YD(1)*SIN(Y(2))*COS(Y(3))-Y(1)*YD(3)*SIN(Y(2))*SIN(Y(3))
     &  +Y(1)*YD(2)*COS(Y(3))*COS(Y(2)))-XMASS*G*COS(Y(4))*COS(Y(5))
     &  -XMASS*Y(8)*Y(1)*COS(Y(2))*COS(Y(3))+XMASS*Y(7)*Y(1)*SIN(Y(3))
     &  +XMASS*LAMDA*(Y(1)*SIN(Y(2))*COS(Y(3))-VRW*SIN(ALFAR)*COS(BETAR))
C
      TAL =-XIXZ*Y(7)*Y(8)+XIZZ*Y(8)*Y(9)-XIYY*Y(8)*Y(9)+XIXX*YD(7)-XIXZ*YD(9)
     &  +XIXX*LAMDA*(Y(7)-PB)-XIXZ*LAMDA*(Y(9)-RB)
C
      TAM = XIYY*YD(8)+(XIXX-XIZZ)*Y(7)*Y(9)+XIXZ*(Y(7)*Y(7)-Y(9)*Y(9))
     &  +XIYY*LAMDA*(Y(8)-QB)
C
      TAN = XIZZ*YD(9)-XIXZ*YD(7)+XIXZ*Y(8)*Y(9)+XIYY*Y(7)*Y(8)-XIXX*Y(7)*Y(8)
     &  -XIXZ*LAMDA*(Y(7)-PB)+XIZZ*LAMDA*(Y(9)-RB)
C
C     calculate the time derivatives of the states
C
      XD(1) = (-2*ZETAP*OMEGAP-2*ZETASP*OMEGASP)*X(1)+(-OMEGAP*OMEGAP
     &  -4*ZETASP*ZETAP*OMEGASP*OMEGAP-OMEGASP*OMEGASP)*X(2)
     &  +(-2*ZETASP*OMEGASP*OMEGAP*OMEGAP
     &  -2*ZETAP*OMEGAP*OMEGASP*OMEGASP)*X(3)
```

```
     &  -OMEGASP*OMEGASP*OMEGAP*OMEGAP*X(4)+DDTH
C
      XD(2) = X(1)
      XD(3) = X(2)
      XD(4) = X(3)
C
      XD(5) = (-2*ZETAP*OMEGAP-2*ZETASP*OMEGASP)*X(5)+(-OMEGAP*OMEGAP
     &  -4*ZETASP*ZETAP*OMEGASP*OMEGAP-OMEGASP*OMEGASP)*X(6)
     &  +(-2*ZETASP*OMEGASP*OMEGAP*OMEGAP
     &  -2*ZETAP*OMEGAP*OMEGASP*OMEGASP)*X(7)
     &  -OMEGASP*OMEGASP*OMEGAP*OMEGAP*X(8)+DDE
C
      XD(6) = X(5)
      XD(7) = X(6)
      XD(8) = X(7)
C
      XD(9) = (-2*ZETAD*OMEGAD-1/TR-1/TS)*X(9)+(-OMEGAD*OMEGAD
     &  -2*ZETAD*OMEGAD/TR-2*ZETAD*OMEGAD/TS-1/TS/TR)*X(10)
     &  +(-OMEGAD*OMEGAD/TR-OMEGAD*OMEGAD/TS
     &  -2/TS/TR*ZETAD*OMEGAD)*X(11)-1/TS/TR*OMEGAD*OMEGAD*X(12)+DDA
C
      XD(10) = X(9)
      XD(11) = X(10)
      XD(12) = X(11)
      XD(13) = X(12)
C
      XD(14) = (-2*ZETAD*OMEGAD-1/TR-1/TS)*X(14)+(-OMEGAD*OMEGAD
     &  -2*ZETAD*OMEGAD/TR-2*ZETAD*OMEGAD/TS-1/TS/TR)*X(15)
     &  +(-OMEGAD*OMEGAD/TR-OMEGAD*OMEGAD/TS
     &  -2/TS/TR*ZETAD*OMEGAD)*X(16)-1/TS/TR*OMEGAD*OMEGAD*X(17)+DDR
C
      XD(15) = X(14)
      XD(16) = X(15)
      XD(17) = X(16)
      XD(18) = X(17)
C
      U1 = Y(1)*COS(Y(2))*COS(Y(3))
      V1 = Y(1)*SIN(Y(3))
      W1 = Y(1)*COS(Y(3))*SIN(Y(2))
C
      XD(19) = U1*COS(Y(5))*COS(Y(6))+V1*(SIN(Y(4))*COS(Y(6))*SIN(Y(5))
     &  -COS(Y(4))*SIN(Y(6)))+W1*(COS(Y(4))*SIN(Y(5))*COS(Y(6))
```

```
     &  +SIN(Y(4))*SIN(Y(6)))
C
      XD(20) = U1*COS(Y(5))*SIN(Y(6))+V1*(SIN(Y(4))*SIN(Y(6))*SIN(Y(5))
     &  +COS(Y(4))*COS(Y(6)))+W1*(COS(Y(4))*SIN(Y(5))*SIN(Y(6))
     &  -SIN(Y(4))*COS(Y(6)))
C
      XD(21) = U1*SIN(Y(5))
     &  -V1*(SIN(Y(4))*COS(Y(5)))-W1*(COS(Y(4))*COS(Y(5)))
C
C   numerically integrate for the next value for the states
C
      X(1) = X(1) + XD(1) * DT
      X(2) = X(2) + XD(2) * DT
      X(3) = X(3) + XD(3) * DT
      X(4) = X(4) + XD(4) * DT
      X(5) = X(5) + XD(5) * DT
      X(6) = X(6) + XD(6) * DT
      X(7) = X(7) + XD(7) * DT
      X(8) = X(8) + XD(8) * DT
      X(9) = X(9) + XD(9) * DT
      X(10) = X(10) + XD(10) * DT
      X(11) = X(11) + XD(11) * DT
      X(12) = X(12) + XD(12) * DT
      X(13) = X(13) + XD(13) * DT
      X(14) = X(14) + XD(14) * DT
      X(15) = X(15) + XD(15) * DT
      X(16) = X(16) + XD(16) * DT
      X(17) = X(17) + XD(17) * DT
      X(18) = X(18) + XD(18) * DT
      X(19) = X(19) + XD(19) * DT
      X(20) = X(20) + XD(20) * DT
      X(21) = X(21) + XD(21) * DT

C-------------------------------------------------------------------------
C
C   END OF TRANSFERFUNCTIONS MODULE
C
C-------------------------------------------------------------------------

      RETURN
      END
```

Vita

Henrik Pettersson was born on a snowy winter's day of 1977 to Monalisa and Bengt Pettersson in beautiful Göteborg, Sweden. He spent his early youth in Sweden and Jamaica before the family headed to the United States, where they settled in the Washington, D.C. area. His younger days were spent enjoying sports, with soccer and ice hockey being his true passions. Along the way, he was blessed with a younger brother, Fredrik, and sister, Lisa Marie. After high school, Henrik moved back to Sweden for one year to get in touch with his viking roots and to pursue his ice hockey and academic careers. He then made his way back across the pond to the rolling mountains of Southwest Virginia where he spent his first four years playing ice hockey for and earning a Bachelor's Degree in Aerospace Engineering from Virginia Tech in May of 2000. Upon graduation, Henrik did not roam far, but found his way to the Flight Simulation Laboratory and its many great scholars. The next two years were spent honing his dynamics and controls skills, this time earning a Master's Degree in Aerospace Engineering in June of 2002. In his time at Virginia Tech, Henrik met the greatest of friends and people.