5-2013

# Control Law Calculation and Verification Methods for the Variable Stability Navion In-Flight Simulation Aircraft

Joe Ming Yin Siu
jsiu@utk.edu

To the Graduate Council:

I am submitting herewith a thesis written by Joe Ming Yin Siu entitled "Control Law Calculation and Verification Methods for the Variable Stability Navion In-Flight Simulation Aircraft." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Aviation Systems.

Borja Martos, Major Professor

We have read this thesis and recommend its acceptance:

Peter Solies, Ahmad Vakili

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

# Control Law Calculation and Verification Methods for the Variable Stability Navion In-Flight Simulation Aircraft

A Thesis Presented for the
Master of Science
Degree
The University of Tennessee, Knoxville

Joe Ming Yin Siu
May 2013

## DEDICATION

To my beautiful wife, Marie-Michèle.

## ACKNOWLEDGEMENTS

I am very fortunate to have Mr. Borja Martos as my thesis advisor and mentor. His academic knowledge, flight test experience, and encouragement were instrumental to the completion of this thesis. I am also thankful to Dr. Peter Solies and Dr. Ahmad Vakili for being my academic committee members, and for their exceptional teaching styles and skills. Should I pursue a career in teaching in the future, I will strive to be as dedicated and caring as these three professors.

## ABSTRACT

The University of Tennessee Space Institute's (UTSI) variable stability research aircraft, Ryan Navion N66UT, was extensively modified by the Princeton University in the 1960's. When UTSI acquired the aircraft from Princeton, volumes of calibration data, charts, and schematics manuals were transferred to UTSI.

Based on the study and research of available Princeton documents, methods of calculating flight control laws were "reverse-engineered". The Variable Stability Navion employs an implicit model following structure to achieve in-flight simulation of other aircraft's flying quality. Mathematical formulas were derived to calculate stability derivative potentiometer settings, for the analog response feedback flight controls system. MATLAB scripts were created to generate potentiometer settings, for both longitudinal and lateral-directional in-flight simulations, and Simulink models were used to verify the results.

This report outlines the calculation and verification process to allow the Variable Stability Navion to simulate different aircraft's stability and dynamic responses. Sample calculations are focused on the in-flight simulation of the Twin Otter. This report also serves as a step-by-step guide for deriving desired flight control laws.

# TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

## NOMENCLATURE

| | |
|---|---|
| **Aircraft Geometry, Mass, and Inertia** | |
| x,y,z | reference body axes |
| m | aircraft mass, slugs |
| g | gravitational constant, 32.17 ft/s$^2$ |
| S | wing reference area, ft$^2$ |
| $I_{xx}$, $I_{yy}$, $I_{zz}$ | moment of inertia referred to body axes x,y, and z, slug-ft$^2$ |
| $I_{xz}$ | product of inertia, slug-ft$^2$ |
| $\bar{c}$ | mean aerodynamic chord, ft |
| b | wing span, ft |
| | |
| **Aircraft Motion Variables** | |
| V | total velocity, ft/s |
| $\alpha, \beta$ | angle of attack and sideslip, rad |
| Q | dynamic pressure, lb/ft$^2$ |
| p,q,r | angular rates (roll, pitch, yaw) about x,y,z body axes, rad/s |
| $\psi, \theta, \phi$ | Euler orientation angles (heading, pitch, bank), rad |
| $\rho$ | air density, slugs/ft$^3$ |
| | |
| **Aircraft Dynamic Characteristics** | |
| $\omega_{nsp}$ | undamped natural frequency of short period mode, rad/s |
| $\zeta_{sp}$ | damping ratio of short period mode |
| $\omega_{np}$ | undamped natural frequency of phugoid mode, rad/s |
| $\zeta_{p}$ | damping ratio of phugoid mode |
| $\omega_{nDR}$ | undamped natural frequency of Dutch roll |
| $\zeta_{DR}$ | damping ratio of Dutch roll |
| $\tau_r$ | roll mode time constant, s |
| $\tau_s$ | spiral mode time constant, s |
| **Forces and Moments** | |
| T | thrust, lbf |
| L | lift, lbf |
| D | drag, lbf |
| L,M,N | moments (roll, pitch, yaw) about x,y, and z body axes, ft-lbf |
| X,Y,Z | components of force along x,y, and z body axes, lbf |
| $C_L$ | lift coefficient |
| $C_D$ | drag coefficient |
| $C_m$ | pitching moment coefficient |
| $C_n$ | yawing moment coefficient |
| $C_l$ | rolling moment coefficient |
| $C_y$ | side-force coefficient |

| **Control System Variables** | |
|---|---|
| $\delta_{PS}$ | pitch stick deflection, deg forward |
| $\delta_{RS}$ | roll stick deflection, deg left |
| $\delta_{FP}$ | foot pedal deflection, left pedal inch forward |
| $\delta_{TL}$ | throttle lever, cm forward |
| $\delta_{FL}$ | flap lever, units up |
| $\delta_e$ | elevator deflection, rad trailing edge down (TED) |
| $\delta_a$ | aileron deflection, rad right wing TED |
| $\delta_r$ | rudder deflection, rad trailing edge left |
| $\delta_t$ | throttle, cm |
| $\delta_f$ | flaps, rad TED |
| | |
| **Equations of Motion** | |
| $\Delta$ | small perturbation variable |
| o | trimmed condition |
| $\frac{\partial \delta_e}{\partial \alpha}$ | feedback gain (e.g, angle of attack feedback to elevator) |
| $M_\alpha$ | dimensional stability derivative (e.g, pitching moment due to angle of attack) |

# CHAPTER 1
# INTRODUCTION

## 1.1 Background

The University of Tennessee Space Institute (UTSI) operates a variable stability research aircraft, the Ryan Navion N66UT. The aircraft was extensively modified by the Princeton University in the 1960's. The modifications allowed the Navion to simulate the flying characteristics of other aircraft, and pilots could evaluate a wide range of flying qualities. When UTSI acquired the aircraft from Princeton, volumes of calibration data charts and schematics manuals were transferred to UTSI.

The aircraft was last flown in 2006 for the Icing short course. It is anticipated that it will be brought back to operational status by 2014. The aircraft will once again be used for demonstration of stability and control in the Experimental Flight Mechanics course, and potentially be used for stability flight research.

## 1.2 Problem Statement

Over the years, the UTSI's operational knowledge of the variable stability Navion has been lost due to unserviceability of the aircraft and the change of faculty staff. To allow smooth operation of the Navion when it is brought back to serviceable condition, a comprehensive study and research of all the available Princeton archived documents is required.

In the past, the Navion in-flight simulation operations were based on hand-calculations of individual stability and control derivative potentiometer settings. With the advent of digital technology and powerful software, the hand-calculation method needs to be adopted into an automated computer based method, which would allow faster and more accurate calculation of flight control laws. The calculated potentiometer settings would also require lab-based computer simulations to predict flight behavior prior to the actual test flight.

## 1.3 Purpose and Scope of Study

Based on the study and research of available Princeton manuals, methods of calculating flight control law stability derivative settings are "reverse-engineered". Mathematical formulas are derived to calculate stability derivative potentiometer settings, for the analog response feedback flight controls.

MATLAB computer scripts are created to generate potentiometer settings for both longitudinal and lateral-directional in-flight simulations. The MATLAB scripts can also be executed in conjunction with Simulink aircraft models to graphically show how the Variable Stability Navion behaves with new control laws.

This report outlines the calculation procedures and verification process to allow the Variable Stability Navion to simulate the flying quality of a de Havilland DHC-6 Twin Otter aircraft. The MATLAB scripts and Simulink Models can be found in APPENDIX F. This report also serves as a user's manual for the Navion aircrew to implement the MATLAB scripts to suit other intended needs.

## 1.4 Aircraft Description

### 1.4.1 Airframe

The Navion is an all-metal, low wing aircraft powered by a single engine. It was designed and built by North American Aviation in the 1940s for the role of military personnel and cargo carrier, and trainer for officer flight training program. In the 1960's, Princeton University modified the Navion from the four-seat configuration to a two-seat configuration to accommodate the variable stability electronics. UTSI acquired the aircraft in 1988 and has maintained and used the Navion for the role of research and training pilots and engineers.

The aircraft is powered by a 285 hp Teledyne-Continental IO-520 BA engine driving a McCauley three-blade constant speed propeller. The aircraft's maximum takeoff weight is 3150 lbs.

N66UT is modified to have independent controls on five axes: elevator for pitching moment, throttle for longitudinal force, direct flaps for normal force, aileron for rolling moment, and rudder for yawing moment. A three-view diagram of the Navion is included in APPENDIX A.


Figure 1.1 Ryan Navion N66UT (UTSI, 2004)

### 1.4.2 Variable Stability System

The aircraft cockpit is modified with control panels for the variable stability potentiometers. Figure 1.2 shows the longitudinal stability feedback gain potentiometers, and Figure 1.3 shows the lateral-directional potentiometers. The evaluation pilot is seated in the left seat while the safety pilot sits in the right seat (Figure 1.4). For the evaluation pilot side, the Navion is installed with an analog fly-by-wire irreversible flight control system, which uses electrical signals to command hydraulic power-actuated control surfaces deflections. The signals are from various cockpit controllers and motion sensors; they are summed to provide a net signal to each servo-actuator to generate desired surface deflections. The Navion is designed with redundant servos and sensors to guard against hardware failures, and allow the safety pilot to override the evaluation pilot to take control of the aircraft. For the safety pilot side, the flight control system remains reversible, that is, the pilot's stick and pedals are directly connected to the control surfaces through cables and rods (Aviation, 2004).

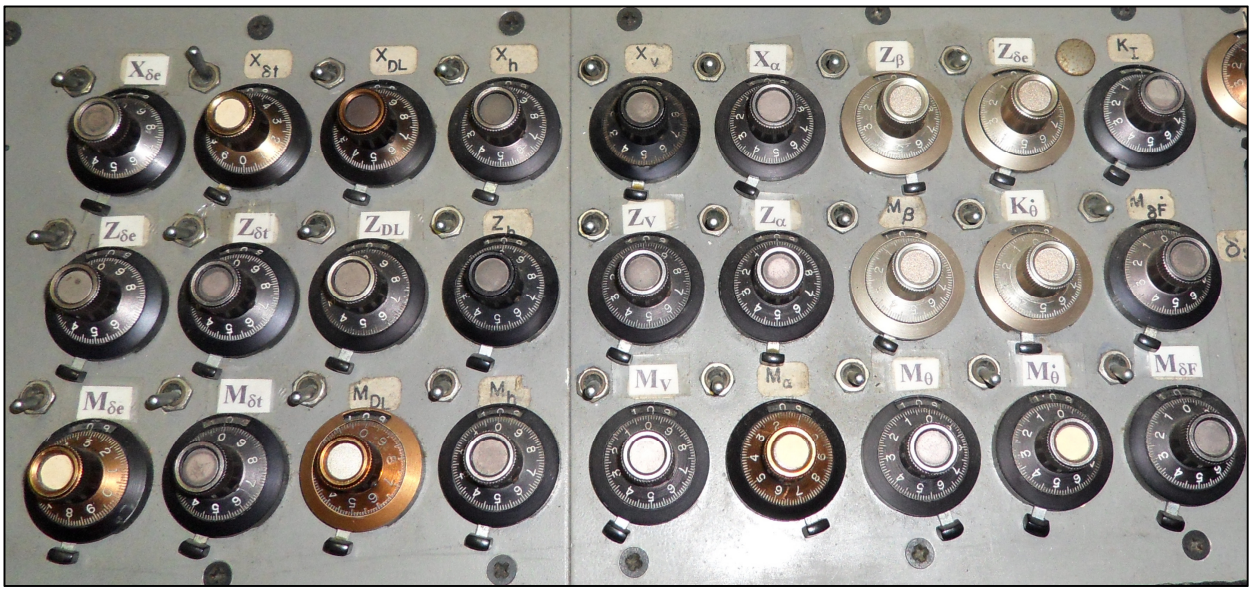


Figure 1.2 Longitudinal Stability Feedback Gain Potentiometers


Figure 1.3 Lateral-Directional Stability Feedback Gain Potentiometers


Figure 1.4 Navion Cockpit View

# CHAPTER 2
# EQUATIONS OF MOTION

## 2.1 Rigid Body Equations of Motion

The purpose of this report is to describe the method of altering the Variable Stability Navion's flying characteristics, and its response to control inputs. An aircraft's flying characteristics can be mathematically described by a set of non-linear differential equations, called equations of motion. The generalized aircraft equations of motion are derived from Newton's second law, as shown in equations (2.1) and (2.2) using a body-axis system shown in Figure 2.1.

$$\sum \vec{F} = \frac{d}{dt}\left(m\vec{V}\right) \tag{2.1}$$

$$\sum \vec{M} = \frac{d}{dt}\vec{H} \tag{2.2}$$

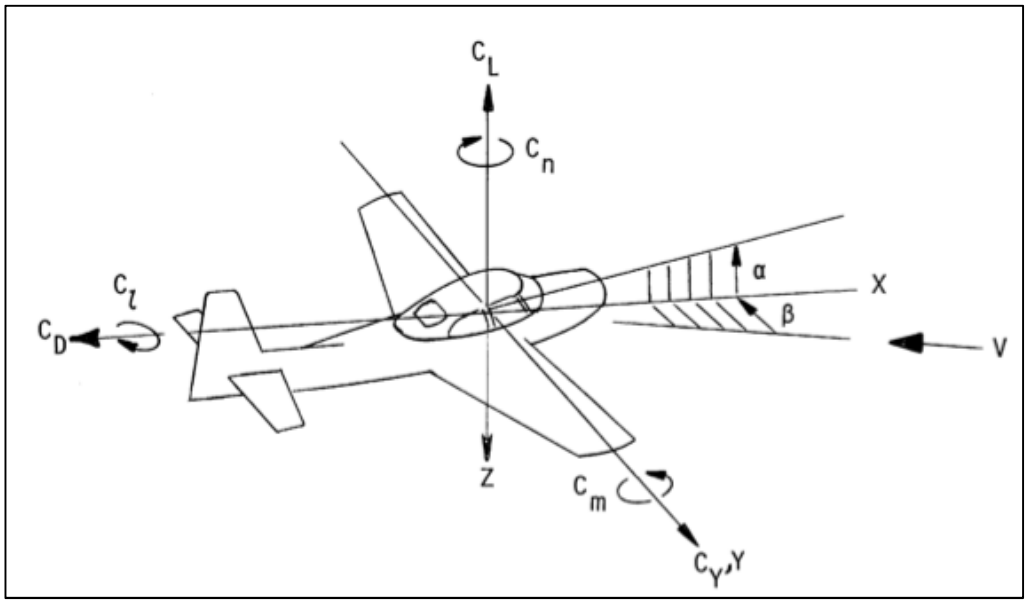


Figure 2.1 Wind and Body Axes (Shivers et al. 1970)

Since the study of stability is for small time duration and limited to a very localized flying area, it is justifiable to use a flat earth model as the inertial reference, and the earth's rotation can be ignored. For the Navion, the aircraft body can be assumed to be rigid, symmetric about the xy-plane (left side is an identical mirror image of the right side), and

with all forces and moments acting about the center of gravity. A set of six equations is derived, and can be grouped into force equations (2.3) (Nelson, 2007, p.105):

$$X - mg\sin\theta = m(\dot{u} + qw - rv)$$

$$Y + mg\cos\theta\sin\phi = m(\dot{v} + ru - pw) \tag{2.3}$$

$$Z + mg\cos\theta\cos\phi = m(\dot{w} + pv - qu)$$

and moment equations (2.4),

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

$$M = I_{yy}\dot{q} + rp(I_{xx} - I_{zz}) + I_{xz}(p^2 - r^2) \tag{2.4}$$

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

The six aircraft equations of motion can be further simplified by decoupling them into a set of longitudinal equations, and a set of lateral-directional equations. Physically, for the longitudinal case, it means when the aircraft's pitch motion is being analyzed, the aircraft needs to maintain wings level with no sideslip, as shown in equations (2.5) and (2.6).

$$p = r = \phi = \beta = 0 \tag{2.5}$$

therefore (2.3) becomes:

$$X - mg\sin\theta = m(\dot{u} + qw)$$

$$Z + mg\cos\theta = m(\dot{w} - qu) \tag{2.6}$$

$$M = I_{yy}\dot{q}$$

For the lateral-directional case, when the aircraft's roll and yaw motions are being analyzed, the aircraft should have zero pitch rate, as shown in equations (2.7) and (2.8).

$$q = 0 \tag{2.7}$$

therefore (2.4) becomes:

$$Y + mg\cos\theta\sin\phi = m\left(\dot{v} + ru - pw\right)$$

$$L = I_{xx}\dot{p} - I_{xz}\dot{r} \tag{2.8}$$

$$N = -I_{xz}\dot{p} + I_{zz}\dot{r}$$

# CHAPTER 3
# LONGITUDINAL EQUATIONS OF MOTION LINEARIZATION

*"The discussion of the equations of motion of an aeroplane in their most general form, and the search for cases in which these equations admit of exact integration, is probably a mathematical problem which may well occupy the attention of pure mathematicians for the next half century."* (Bryan, 1911, p.180)

## 3.1 Linearization

The decoupled longitudinal equations of motion (2.6) can only be solved by numerical integration techniques due to their non-linear nature. To further simplify these equations, the small perturbation theory is used to linearize the equations to obtain close solutions. The approach is to consider the aircraft flying in a steady-state condition (i.e. straight line flight with wings level at constant airspeed), and each motion variable is subjected to a small disturbance. For example, if an aircraft is flying with a trimmed longitudinal velocity of 100 kts, and a small disturbance of 1 kt is encountered, the longitudinal velocity can be expressed as equation (3.1).

$$u = u_0 + \Delta u = 100 \text{ kt} + 1 \text{ kt} \tag{3.1}$$

Similarly, all the motion variables are replaced by known trimmed values and unknown small perturbations. Following the linearization approach stated in (Bryan, 1911; Nelson, 2007, p.106; Cook, 2013, p.93; Yechout, 2003, p.245; Etkin & Reid, 1996, p.109), the trimmed values and perturbations are substituted into the longitudinal equations. Simplifications can be made by applying small-angle approximation, and assuming products of perturbations are negligible. Some terms will become zero because the initial condition is set to be trimmed constant airspeed, straight and level flight. The initial conditions are:

$$v_0 = q_0 = r_0 = p_0 = \phi_0 = \beta_0 = 0 \tag{3.2}$$

## 3.2 Longitudinal Force (X) Equation

The X force equation (2.6) can first be rearranged into this form:

$$\dot{u} = -qw - g\sin\theta + \frac{X}{m} \tag{3.3}$$

Each variable is then replaced by its trimmed value and perturbation variable:

$$(\dot{u}_0 + \Delta\dot{u}) = -(q_0 + \Delta q)(w_0 + \Delta w) - g\sin(\theta_0 + \Delta\theta) + \frac{(X_0 + \Delta X)}{m} \tag{3.4}$$

Multiplying all the terms, and substituting in the following trigonometry identity and small angle approximation:

$$\sin(\theta_0 + \Delta\theta) = \sin\theta_0\cos\Delta\theta + \cos\theta_0\sin\Delta\theta \cong \sin\theta_0 + \Delta\theta\cos\theta_0 \tag{3.5}$$

yields:

$$\Delta\dot{u} + \dot{u}_0 = -q_0 w_0 - q_0\Delta w - \Delta q w_0 - \Delta q\Delta w - g\sin\theta_0 - g\Delta\theta\cos\theta_0 + \frac{X_0}{m} + \frac{\Delta X}{m} \tag{3.6}$$

At equilibrium, or steady state, all the perturbation variables Δ become zero, the equation becomes:

$$\dot{u}_0 = -q_0 w_0 - g\sin\theta_0 + \frac{X_0}{m} \tag{3.7}$$

This expression can be removed from equation (3.6):

$$\Delta\dot{u} = -\underbrace{q_0}_{=0}\Delta w - \Delta q w_0 - \underbrace{\Delta q\Delta w}_{\approx 0} - g\Delta\theta\cos\theta_0 + \frac{\Delta X}{m} \tag{3.8}$$

Finally, apply the initial conditions and remove products of perturbation, the X force equation of motion is linearized:

$$\Delta\dot{u} = \frac{\Delta X}{m} - (g\cos\theta_0)\Delta\theta - w_0\Delta q \tag{3.9}$$

## 3.3 Normal Force (Z) Equation

The Z force equation (2.6) can first be rearranged into this form:

$$\dot{w} = qu + g\cos\theta + \frac{Z}{m} \tag{3.10}$$

Each variable is then replaced by its trimmed value and perturbation variable:

$$(\dot{w}_0 + \Delta\dot{w}) = (q_0 + \Delta q)(u_0 + \Delta u) + g\cos(\theta_0 + \Delta\theta) + \frac{(Z_0 + \Delta Z)}{m} \tag{3.11}$$

Multiplying all the terms, and substituting in the following trigonometry identity and small angle approximation:

$$\cos(\theta_0 + \Delta\theta) = \cos\theta_0 \cos\Delta\theta - \sin\theta_0 \sin\Delta\theta \cong \cos\theta_0 - \Delta\theta \sin\theta_0 \tag{3.12}$$

yields:

$$\Delta\dot{w} + \dot{w}_0 = q_0 u_0 + q_0 \Delta u + \Delta q u_0 + \Delta q \Delta u + g\cos\theta_0 - \Delta\theta g \sin\theta_0 + \frac{Z_0}{m} + \frac{\Delta Z}{m} \tag{3.13}$$

At equilibrium, or steady state, all the perturbation variables Δ become zero, the equation becomes:

$$\dot{w}_0 = q_0 u_0 + g\cos\theta_0 + \frac{Z_0}{m} \tag{3.14}$$

This expression can be removed from equation (3.13):

$$\Delta\dot{w} = \underbrace{q_0}_{=0} \Delta u + \Delta q u_0 + \underbrace{\Delta q \Delta u}_{\approx 0} - \Delta\theta g \sin\theta_0 + \frac{\Delta Z}{m} \tag{3.15}$$

Finally, apply the initial conditions and remove products of perturbation, the Z force equation of motion is linearized:

$$\Delta\dot{w} = \frac{\Delta Z}{m} - \left(g \sin\theta_0\right)\Delta\theta + u_0 \Delta q \tag{3.16}$$

## 3.4 Pitching Moment (M) Equation

The pitch moment equation (2.6) can first be rearranged into this form:

$$\dot{q} = \frac{M}{I_{yy}} \tag{3.17}$$

Each variable is then replaced by its trimmed value and perturbation variable:

$$\left(\dot{q}_0 + \Delta\dot{q}\right) = \frac{\left(M_0 + \Delta M\right)}{I_{yy}} \tag{3.18}$$

At equilibrium, or steady state, all the perturbation variables Δ become zero, the equation becomes:

$$\dot{q}_0 = \frac{M_0}{I_{yy}} \tag{3.19}$$

This expression can be removed from equation (3.18):

$$\Delta\dot{q} = \frac{\Delta M}{I_{yy}} \tag{3.20}$$

In Summary, the longitudinal equations of motion can now be rewritten as a set of linear equations (3.21):

$$\Delta\dot{u} = \frac{\Delta X}{m} - \left(g\cos\theta_0\right)\Delta\theta - w_0\Delta q$$

$$\Delta\dot{w} = \frac{\Delta Z}{m} - \left(g\sin\theta_0\right)\Delta\theta + u_0\Delta q \tag{3.21}$$

$$\Delta\dot{q} = \frac{\Delta M}{I_{yy}}$$

## 3.5 Taylor Series Approximation of Forces and Moments

The perturbed force and moment variables in the linearized longitudinal equations of motion (3.21) can be approximated by using first-order Taylor series expansion:

$$\Delta X = \frac{\partial X}{\partial u}\Delta u + \frac{\partial X}{\partial w}\Delta w + \frac{\partial X}{\partial \dot{w}}\Delta\dot{w} + \frac{\partial X}{\partial q}\Delta q + \frac{\partial X}{\partial \delta_e}\Delta\delta_e + \frac{\partial X}{\partial \delta_t}\Delta\delta_t + \frac{\partial X}{\partial \delta_f}\Delta\delta_f$$

$$\Delta Z = \frac{\partial Z}{\partial u}\Delta u + \frac{\partial Z}{\partial w}\Delta w + \frac{\partial Z}{\partial \dot{w}}\Delta\dot{w} + \frac{\partial Z}{\partial q}\Delta q + \frac{\partial Z}{\partial \delta_e}\Delta\delta_e + \frac{\partial Z}{\partial \delta_t}\Delta\delta_t + \frac{\partial Z}{\partial \delta_f}\Delta\delta_f \tag{3.22}$$

$$\Delta M = \frac{\partial M}{\partial u}\Delta u + \frac{\partial M}{\partial w}\Delta w + \frac{\partial M}{\partial \dot{w}}\Delta\dot{w} + \frac{\partial M}{\partial q}\Delta q + \frac{\partial M}{\partial \delta_e}\Delta\delta_e + \frac{\partial M}{\partial \delta_t}\Delta\delta_t + \frac{\partial M}{\partial \delta_f}\Delta\delta_f$$

The only significant higher order derivative term commonly encountered is $\frac{\partial M}{\partial \dot{w}}$. For the purpose of this thesis, and base on the Variable Stability Navion's hardware design, the perturbed force and moment variables can be represented by the following simplified Taylor series:

$$\Delta X = \frac{\partial X}{\partial u}\Delta u + \frac{\partial X}{\partial w}\Delta w + \frac{\partial X}{\partial \delta_t}\Delta \delta_t$$

$$\Delta Z = \frac{\partial Z}{\partial u}\Delta u + \frac{\partial Z}{\partial w}\Delta w + \frac{\partial Z}{\partial q}\Delta q + \frac{\partial Z}{\partial \delta_e}\Delta \delta_e + \frac{\partial Z}{\partial \delta_f}\Delta \delta_f \tag{3.23}$$

$$\Delta M = \frac{\partial M}{\partial u}\Delta u + \frac{\partial M}{\partial w}\Delta w + \frac{\partial M}{\partial \dot{w}}\Delta \dot{w} + \frac{\partial M}{\partial q}\Delta q + \frac{\partial M}{\partial \delta_e}\Delta \delta_e$$

The longitudinal force *ΔX* includes the aerodynamic forces and the applied force due to engine thrust. The normal force *ΔZ* includes the aerodynamic forces and the applied forces due to elevator and flap deflections. The pitching moment *ΔM* includes the aerodynamic moments and the applied moment due to elevator deflection.

It is convenient to use angle of attack *Δα* instead of the normal velocity *Δw*. They are related to each other by the small angle approximation theory:

$$\Delta \alpha \cong \tan^{-1}\frac{\Delta w}{u_0} = \frac{\Delta w}{u_0} \tag{3.24}$$

also,

$$\Delta \dot{\alpha} = \frac{\Delta \dot{w}}{u_0} \tag{3.25}$$

Substituting the Taylor series representation of forces and moment (3.23) into the small perturbation longitudinal linearized equations of motion equation (3.21) yields:

$$\Delta \dot{u} = \frac{\left(\frac{\partial X}{\partial u}\right)}{m}\Delta u + \frac{\left(\frac{\partial X}{\partial w}\right)}{m}\Delta w - w_0 \Delta q - \left(g\cos\theta_0\right)\Delta\theta + \frac{\left(\frac{\partial X}{\partial \delta_t}\right)}{m}\Delta \delta_t$$

$$\Delta \dot{w} = \frac{\left(\frac{\partial Z}{\partial u}\right)}{m}\Delta u + \frac{\left(\frac{\partial Z}{\partial w}\right)}{m}\Delta w + \left(u_0 + \frac{\left(\frac{\partial Z}{\partial w}\right)}{m}\right)\Delta q - \left(g\sin\theta_0\right)\Delta\theta + \frac{\left(\frac{\partial Z}{\partial \delta_e}\right)}{m}\Delta \delta_e + \frac{\left(\frac{\partial Z}{\partial \delta_f}\right)}{m}\Delta \delta_f \tag{3.26}$$

$$\Delta \dot{q} = \frac{\left(\frac{\partial M}{\partial u}\right)}{I_{yy}}\Delta u + \frac{\left(\frac{\partial M}{\partial w}\right)}{I_{yy}}\Delta w + \frac{\left(\frac{\partial M}{\partial \dot{w}}\right)}{I_{yy}}\Delta \dot{w} + \frac{\left(\frac{\partial M}{\partial q}\right)}{I_{yy}}\Delta q + \frac{\left(\frac{\partial M}{\partial \delta_e}\right)}{I_{yy}}\Delta \delta_e$$

## 3.6 Dimensional Stability Derivatives

The small perturbation longitudinal linearized equations of motion (3.26) can be rewritten into a more compact form by introducing dimensional stability derivatives, for example, the "pitch damping" derivative can be calculated using the stability coefficient (3.27):

$$M_q = \frac{\left(\frac{\partial M}{\partial q}\right)}{I_{yy}} = \frac{C_{m_q} Q S \bar{c}^2}{2u_0 I_{yy}} \tag{3.27}$$

where the dynamic pressure $Q$, can be expressed as equation (3.28) in low Mach number flight conditions.

$$Q = \frac{1}{2}\rho V_t^2 = \frac{1}{2}\rho_0 V_e^2 \cong \frac{1}{2}\rho_0 V_c^2 \tag{3.28}$$

The sources of aircraft stability data are from aircraft system identification flight testing (Klein & Morelli, 2006), wind-tunnel testing, or by computer simulations. Unfortunately, textbooks and literatures do not always present the stability derivatives in the same format or use the same formulas. In some textbooks (Etkin & Reid, 1996; Yechout, 2003), the aircraft characteristics are presented as dimensional derivatives in the body-axes. In (Nelson, 2007), aircraft aerodynamic characteristics are presented in dimensionless coefficient form in the wind-axes. Finally, in (Cook, 2013), both dimensional and dimensionless derivatives are presented, however, they are calculated differently using the British formulas. The (Nelson, 2007, p.123) definitions will be used for this thesis.

Since the trimmed aircraft has a small angle of attack during straight and level cruise flight condition, the following assumptions relating wind-axis to body-axis are valid (Yechout, 2003, p.178):

$$Z = -D\sin\alpha - L\cos\alpha \cong -L \tag{3.29}$$

and

$$X = -D\cos\alpha + L\sin\alpha \cong -D \tag{3.30}$$

The Variable Stability Navion is designed to use dimensional stability derivatives. It is important to convert stability data into the compatible dimensional form. Table 3.1 summarizes each dimensional derivative used by the Navion.

Table 3.1. Summary of Longitudinal Derivatives

| | |
|---|---|
| $X_u = \frac{\left(\frac{\partial X}{\partial u}\right)}{m} = \frac{-\left(C_{D_u} + 2C_{D_0}\right)QS}{mu_0}$ | $X_w = \frac{\left(\frac{\partial X}{\partial w}\right)}{m} = \frac{-\left(C_{D_\alpha} - C_{L_0}\right)QS}{mu_0}$ |
| $X_\alpha = u_0 X_w = \frac{-\left(C_{D_\alpha} + C_{L_0}\right)QS}{m}$ | $X_{\delta t} = \frac{\left(\frac{\partial X}{\partial \delta t}\right)}{m}$ |
| $Z_u = \frac{\left(\frac{\partial Z}{\partial u}\right)}{m} = \frac{-\left(C_{L_u} + 2C_{L_0}\right)QS}{mu_0}$ | $Z_q = \frac{\left(\frac{\partial Z}{\partial q}\right)}{m} = \frac{-C_{L_q} QS\bar{c}}{2mu_0}$ |
| $Z_w = \frac{\left(\frac{\partial Z}{\partial w}\right)}{m} = \frac{-\left(C_{L_\alpha} - C_{D_0}\right)QS}{mu_0}$ | $Z_\alpha = u_0 Z_w = \frac{-\left(C_{L_\alpha} + C_{D_0}\right)QS}{m}$ |
| $Z_{\delta_e} = \frac{\left(\frac{\partial Z}{\partial \delta_e}\right)}{m} = \frac{-C_{L_{\delta_e}} QS}{m}$ | $Z_{\delta_f} = \frac{\left(\frac{\partial Z}{\partial \delta_f}\right)}{m}$ |
| $M_u = \frac{\left(\frac{\partial M}{\partial u}\right)}{I_{yy}} = \frac{C_{m_u} QS\bar{c}}{u_0 I_{yy}}$ | $M_q = \frac{\left(\frac{\partial M}{\partial q}\right)}{I_{yy}} = \frac{C_{m_q} QS\bar{c}^2}{2u_0 I_{yy}}$ |
| $M_w = \frac{\left(\frac{\partial M}{\partial w}\right)}{I_{yy}} = \frac{C_{m_\alpha} QS\bar{c}}{u_0 I_{yy}}$ | $M_\alpha = u_0 M_w = \frac{C_{m_\alpha} QS\bar{c}}{I_{yy}}$ |
| $M_{\dot{w}} = \frac{\left(\frac{\partial M}{\partial \dot{w}}\right)}{I_{yy}} = \frac{C_{m_\alpha} QS\bar{c}^2}{2u_0^2 I_{yy}}$ | $M_{\dot{\alpha}} = u_0 M_{\dot{w}} = \frac{C_{m_{\dot{\alpha}}} QS\bar{c}^2}{2u_0 I_{yy}}$ |
| $M_{\delta e} = \frac{\left(\frac{\partial M}{\partial \delta e}\right)}{I_{yy}} = \frac{C_{m_{\delta e}} QS\bar{c}}{I_{yy}}$ | |

(Nelson, 2007, p.123, Yechout, p.289)

The following example illustrates the calculation of the Navion $M_\alpha$ dimensional derivative:

Calculating Navion $M_\alpha$. Should the stability derivatives be given as dimensionless coefficients, a conversion is required to dimensionalize the derivatives. The $C_{m_\alpha}$ value given in (Nelson, 2007, p.400) is for a specific weight, center of gravity, atmospheric condition, altitude, airspeed, and configuration. In this case, a $C_{m_\alpha}$ value of -0.683 is given

for the basic Navion in trimmed flight of 104 KCAS (176 ft/s) at sea level (0 ft Hp). By using calibrated airspeed to calculate dynamic pressure $Q$, the dependency to altitude can be minimized:

$$Q = \frac{1}{2}\rho_0 V_c^{\,2} = \frac{1}{2}\left(0.00238 \text{ slug/ft}^3\right)\left(176 \text{ ft/s}\right)^2 = 36.9 \text{ lbf/ft}^2 \tag{3.31}$$

$$\left(M_\alpha\right)_{Navion} = \frac{C_{m_\alpha} Q S \overline{c}}{I_{yy}} = \frac{\left(-0.683\right)\left(36.9 \text{ lbf/ft}^2\right)\left(184 \text{ ft}^2\right)\left(5.70 \text{ ft}\right)}{\left(3000 \text{ slugs}\cdot\text{ft}^2\right)} = -8.80 \text{ s}^{-2} \tag{3.32}$$

As shown in the above example, each dimensional stability derivative can be calculated for a specific aircraft. Three significant figures will be used throughout this report's calculations. It must be noted that a great source of calculation errors can arise due to incompatible units and sign conventions, attention must be paid to ensure all data are converted into a standardized format.

## 3.7 Basic Navion and Twin Otter Longitudinal Dimensional Derivatives

The basic Navion geometric, mass, and aerodynamic characteristics data are for trimmed airspeed of 104 KCAS (M 0.158) at sea level (Nelson, 2007, p.400). Thrust and flap derivatives are taken from old Princeton hand-written workbooks. The following table summarizes the basic Navion longitudinal dimensional derivatives.

Table 3.2 Basic Navion Longitudinal Dimensional Derivatives

| | |
|---|---|
| $X_u = -0.0451 \text{ s}^{-1}$ | $X_w = 0.0361 \text{ s}^{-1}$ |
| $X_\alpha = 6.35 \text{ s}^{-1}$ | $X_{\delta_t} = 3.59 \text{ ft/s}^2/\text{cm}$ |
| $Z_u = -0.370 \text{ s}^{-1}$ | $Z_q = -4.88 \text{ ft/s}$ |
| $Z_w = -2.02 \text{ s}^{-1}$ | $Z_\alpha = -356 \text{ ft/s}^2$ |
| $Z_{\delta_e} = -28.2 \text{ ft/s}^2$ | $Z_{\delta_f} = -70.9 \text{ ft/s}^2$ |
| $M_u = 0 \text{ ft}^{-1}\text{s}^{-1}$ | $M_q = -2.08 \text{ ft/s}$ |
| $M_w = -0.05 \text{ ft}^{-1}\text{s}^{-1}$ | $M_\alpha = -8.80 \text{ s}^{-2}$ |
| $M_{\dot{w}} = -0.00517 \text{ ft}^{-1}$ | $M_{\dot{\alpha}} = -0.910 \text{ s}^{-1}$ |
| $M_{\delta_e} = -11.9 \text{ s}^{-2}$ | |

The Twin Otter geometric, mass, and aerodynamic characteristics data are for trimmed airspeed of 104 KCAS (M 0.158) at sea level (Brigg et al., 2000). The following table summarizes the clean configuration Twin Otter longitudinal dimensional derivatives.

Table 3.3. Twin Otter Longitudinal Dimensional Derivatives

| | |
|---|---|
| $X_u = -0.0255\ \text{s}^{-1}$ | $X_w = 0.0218\ \text{s}^{-1}$ |
| $X_\alpha = 3.84\ \text{s}^{-1}$ | $X_{\delta_t}$ not available |
| $Z_u = -0.237\ \text{s}^{-1}$ | $Z_q = -20.2$ ft/s |
| $Z_w = -1.78\ \text{s}^{-1}$ | $Z_\alpha = -312\ \text{ft/s}^2$ |
| $Z_{\delta_e} = -33.3\ \text{ft/s}^2$ | $Z_{\delta_f}$ not available |
| $M_u = 0\ \text{ft}^{-1}\text{s}^{-1}$ | $M_q = -2.89$ ft/s |
| $M_w = -0.0341\ \text{ft}^{-1}\text{s}^{-1}$ | $M_\alpha = -5.99\ \text{s}^{-2}$ |
| $M_{\dot{w}}$ not available | $M_{\dot{\alpha}}$ not available |
| $M_{\delta_e} = -7.96\ \text{s}^{-2}$ | |

## 3.8 Linearized Equations of Motion with Dimensional Stability Derivatives

Substituting the dimensional stability derivatives from Table 3.1 into the small perturbation linearized longitudinal equations of motion (3.26) yields:

$$\Delta\dot{u} = X_u\Delta u + X_w\Delta w - w_0\Delta q - \left(g\cos\theta_0\right)\Delta\theta + X_{\delta_t}\Delta\delta_t$$

$$\Delta\dot{w} = Z_u\Delta u + Z_w\Delta w + \left(u_0 + Z_q\right)\Delta q - \left(g\sin\theta_0\right)\Delta\theta + Z_{\delta_e}\Delta\delta_e + Z_{\delta_f}\Delta\delta_f \qquad (3.33)$$

$$\Delta\dot{q} = M_u\Delta u + M_w\Delta w + M_{\dot{w}}\Delta\dot{w} + M_q\Delta q + M_{\delta_e}\Delta\delta_e$$

Two final changes are required to arrange the above set of equations into compatible format for the Variable Stability Navion. The first change is to recognize that the entire $\Delta\dot{w}$ equation can be substituted into the $\Delta\dot{q}$ equation; this eliminates the need for a $\dot{w}$ feedback sensor, as it is not available in the aircraft. The second change is to convert the $\Delta\dot{w}$ equation into $\Delta\dot{\alpha}$ equation by dividing the entire equation by $u_0$ as per equation (3.25). This change facilitates calculations because the Navion uses an angle of

attack vane as feedback, rather than a normal velocity sensor. The resulting equation set becomes:

$$\Delta\dot{u} = X_u\Delta u + X_\alpha\Delta\alpha - w_0\Delta q - (g\cos\theta_0)\Delta\theta + X_{\delta_t}\Delta\delta_t$$

$$\Delta\dot{\alpha} = \left(\frac{Z_u}{u_0}\right)\Delta u + \left(\frac{Z_\alpha}{u_0}\right)\Delta\alpha + \left(1+\frac{Z_q}{u_0}\right)\Delta q - \left(\frac{g}{u_0}\sin\theta_0\right)\Delta\theta + \left(\frac{Z_{\delta_e}}{u_0}\right)\Delta\delta_e + \left(\frac{Z_{\delta_f}}{u_0}\right)\Delta\delta_f \tag{3.34}$$

$$\Delta\dot{q} = \left(M_u + M_{\dot{\alpha}}\frac{Z_u}{u_0}\right)\Delta u + \left(M_\alpha + M_{\dot{\alpha}}\frac{Z_\alpha}{u_0}\right)\Delta\alpha + \left[M_q + M_{\dot{\alpha}}\left(1+\frac{Z_q}{u_0}\right)\right]\Delta q$$
$$-\left(M_{\dot{\alpha}}\frac{g}{u_0}\sin\theta_0\right)\Delta\theta + \left(M_{\delta_e} + M_{\dot{\alpha}}\frac{Z_{\delta_e}}{u_0}\right)\Delta\delta_e + \left(M_{\dot{\alpha}}\frac{Z_{\delta_f}}{u_0}\right)\Delta\delta_f$$

## 3.9 Model fidelity

The longitudinal model derived thus far only applies to small perturbations. The dimensional stability derivatives are assumed to be constant within the small range about the trimmed condition. For example, the lift curve slope $C_{L\alpha}$ in Figure 3.1 (Shivers et al., 1970) has a nearly constant slope between -4° to 12° angles of attack. Should the aircraft slow down significantly from the initial trimmed airspeed, and the angle of attack rises to the stall angle, the $C_{L\alpha}$ value would no longer be the initial value at the linear region. In this instant, the $Z_\alpha$ value originally calculated is no longer valid for useful analysis.

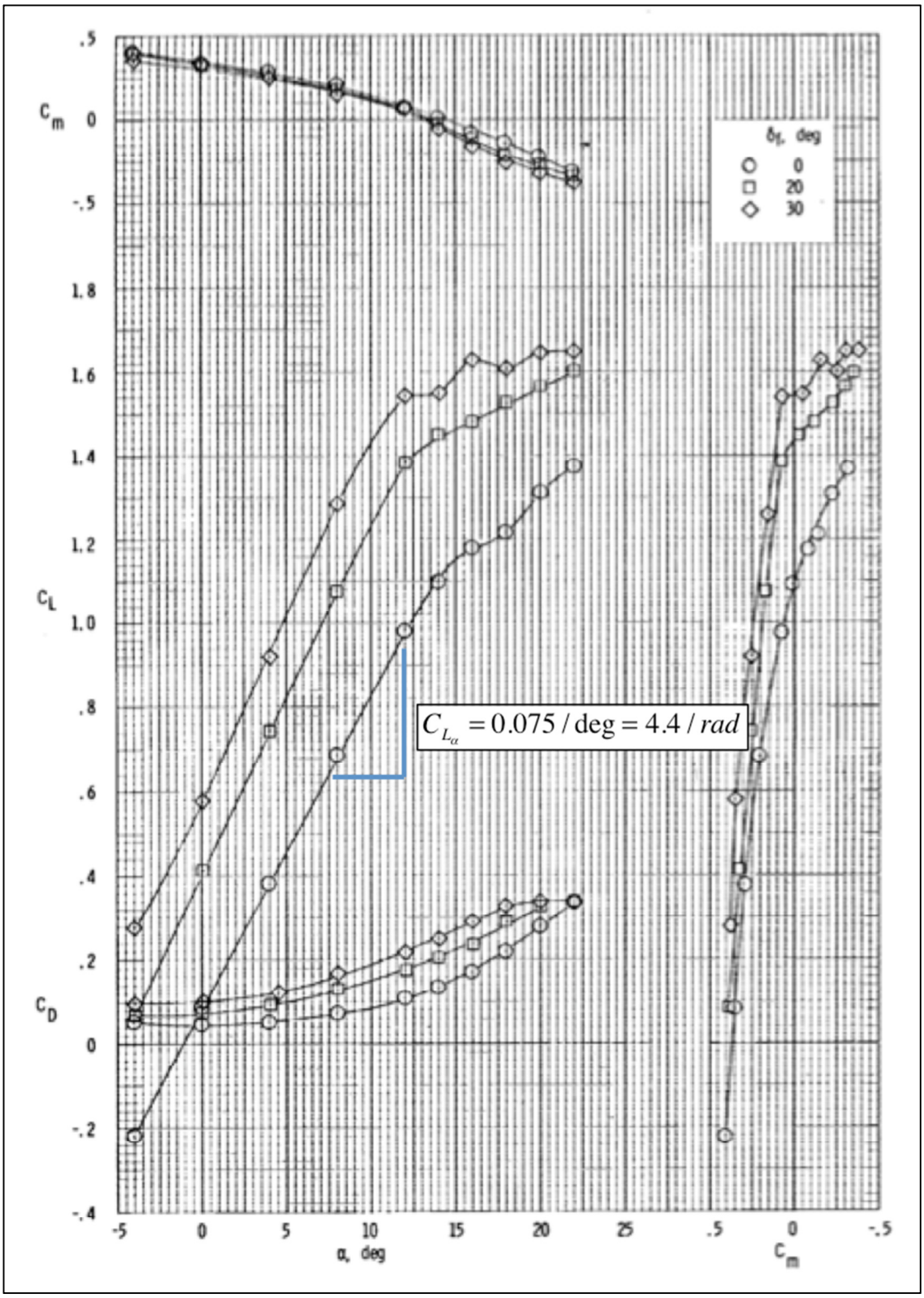


Figure 3.1 $C_L$, $C_D$, and $C_m$ Curves of the Navion (Shivers et al., 1970)

## 3.10 Simulink Simulation

As shown in equation (3.34), the longitudinal equations of motion can be linearized into simple three degrees of freedom (3DOF) model. Using the MATLAB Simulink model (APPENDIX E), the linearized 3DOF model is subjected to pitch doublet inputs, to disturb the aircraft flying straight and level with constant trimmed airspeed. The time histories of the desired aircraft are then compared with the basic Navion. This simple model can be very helpful with immediate troubleshooting and visualization of motions.

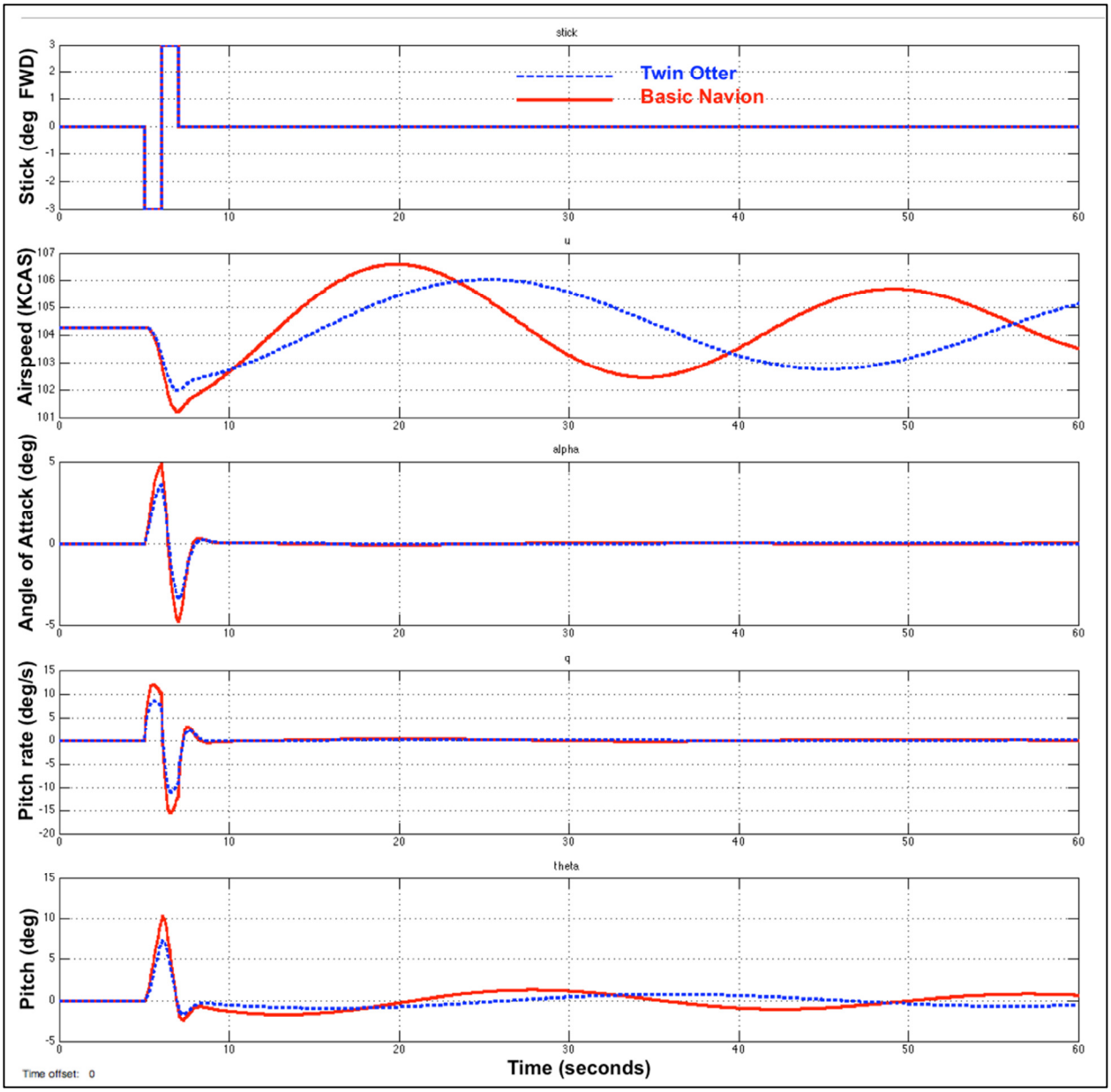


Figure 3.2 Basic Navion and Twin Otter Pitch Doublet Response

The above figure demonstrates how each aircraft responds to the same pilot stick input pitch doublet. Many state variables can be plotted, and each one reveals the dynamic response of the aircraft. As expected, the basic Navion do not behave the same way as the basic Twin Otter to the same pilot pitch doublet input.

# CHAPTER 4
# LONGITUDINAL IMPLICIT MODEL FOLLOWING

## 4.1 Variable Stability Implementation

The Variable Stability Navion uses a response feedback system to simulate the stability characteristic of other aircraft. This is an application of feedback gains in the classical control theory. The following figure represents angle of attack to elevator feedback gain structure. The main elements of this structure are the sensor to observe the state variable, in this case an angle of attack vane; and the signal amplifier that sets the feedback gain to the elevator actuator.

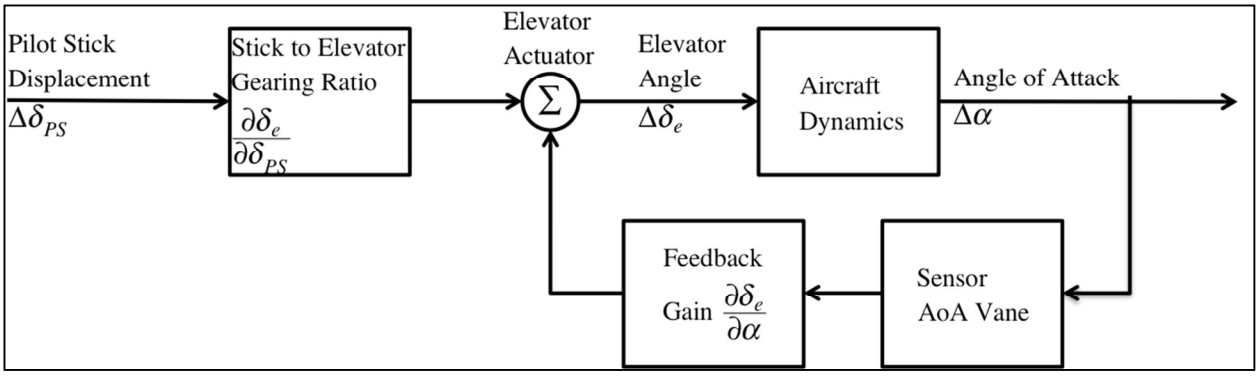


Figure 4.1 Simple Angle of Attack Feedback System

The mechanism of using the Variable Stability Navion to simulate another aircraft is by the using an implicit model following technique. The technique requires prior knowledge of the basic Navion and the desired aircraft's stability derivatives, which is usually accomplished by parameter identification flight test or wind tunnel testing. For the purpose of this thesis, the desired aircraft will be a clean configuration Twin Otter.

Once all the dimensional stability derivatives are calculated for both the Navion and the Twin Otter, as described in CHAPTER 3, the Navion can simulate the Twin Otter by using feedback gains to match each stability derivative term-by-term. The following example illustrates how the Variable Stability Navion uses angle of attack to elevator feedback gain $\frac{\partial\delta_e}{\partial\alpha}$, to match the aircraft pitch rates $\Delta\dot{q}$ caused by a gust of wind $\Delta\alpha$.

At equilibrium, both the Twin Otter and Navion are flying straight and level at a constant trimmed airspeed, a gust of wind causes both aircraft's angle of attack to increase by $\Delta\alpha$.

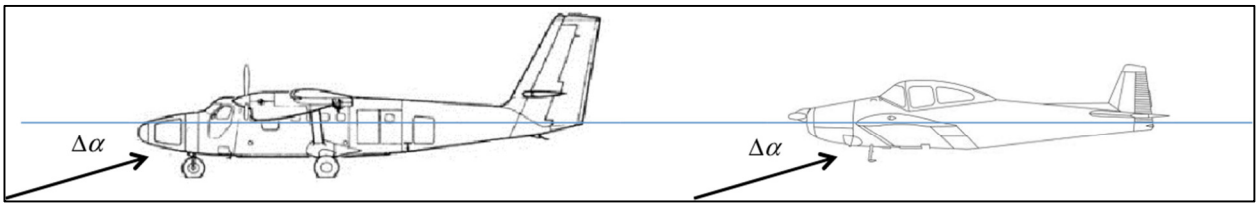


Figure 4.2 Aircraft Flying in Trimmed Condition

Both aircraft's longitudinal stabilities cause the aircraft to pitch down. The Navion's $M_\alpha$ of -8.8 $s^{-2}$ is greater than the Twin Otter's $M_\alpha$ of -6.0 $s^{-2}$; the Navion would experience a higher pitch down rate $\Delta\dot{q}$ than the Twin Otter, as shown in the figure below:

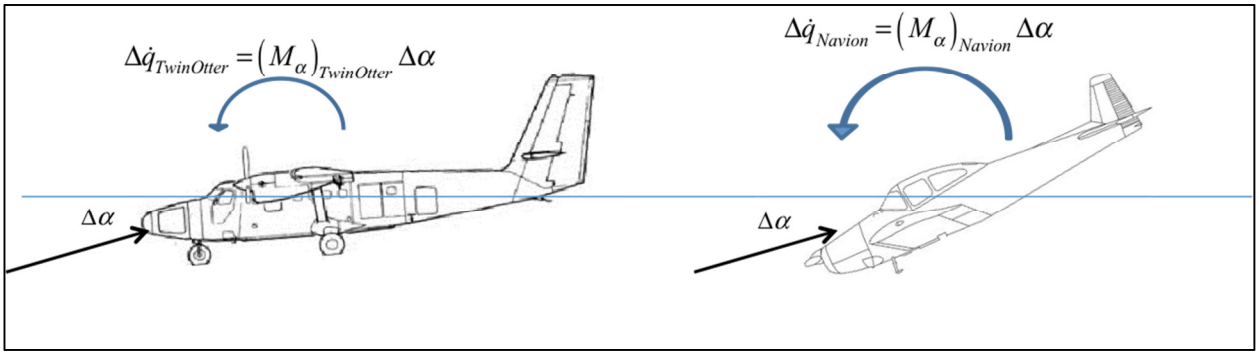


Figure 4.3 Different Pitching Moments due to Angle of Attack

The Navion can simulate the same pitch rate $\Delta\dot{q}$ as the Twin Otter by using its elevator to compensate for the difference. The amount of elevator to deflect is automatically commanded by the angle of attack to elevator feedback gain $\frac{\partial\delta_e}{\partial\alpha}$. In essence, this $\frac{\partial\delta_e}{\partial\alpha}$ gain matches the basic Navion's $M_\alpha$ to the Twin Otter's $M_\alpha$.

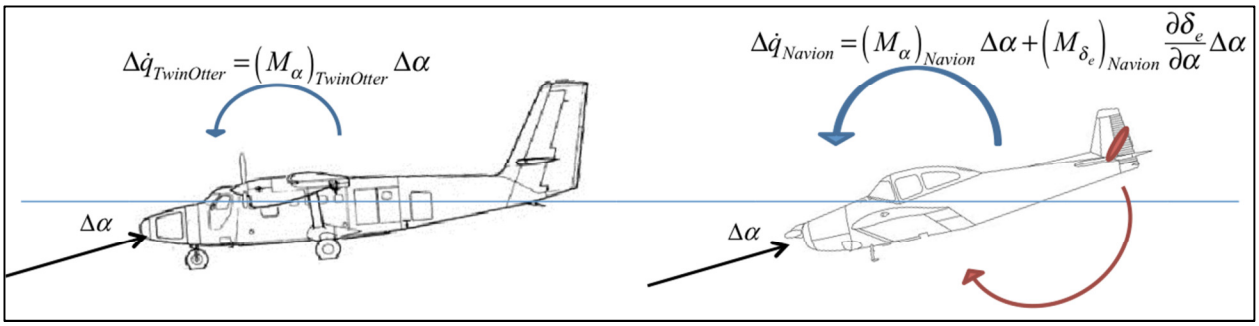


Figure 4.4 $M_\alpha$ Matching using Angle of Attack to Elevator Feedback

The purpose of this thesis is to describe the systematic methods of calculating these feedback gains for the Variable Stability Navion to simulate other aircraft.

## 4.2 Term-by-Term Stability Derivative Matching Method

To match the Navion to a desired aircraft's pitch axis, each term of the $\Delta\dot{q}$ equation of motion for both aircraft needs to be matched (Kidd et al, 1961).

$$\Delta\dot{q}_{Desired} = \Delta\dot{q}_{Navion} \tag{4.1}$$

equation (4.1) with only the $\Delta\alpha$ term:

$$(M_\alpha)_{Desired}\,\Delta\alpha = (M_\alpha)_{Navion}\,\Delta\alpha + (M_{\delta_e})_{Navion}\left(\frac{\partial\delta_e}{\partial\alpha}\right)\Delta\alpha \tag{4.2}$$

The objective is to find the feedback gain values, which can be dialed in the Navion Variable Stability control panel via potentiometers. The process involves first obtaining the dimensional stability derivatives for both the Navion and the desired aircraft. Once all the dimensional stability derivatives are calculated, each derivative can be matched term-by-term by finding the required feedback gain. With the feedback gains calculated, they can now be translated to actual potentiometer settings via calibration graphs and equations.

It would be unreasonable to expect the Variable Stability Navion to simulate highly maneuverable aircraft such as a fighter jet. The longitudinal in-flight simulation is limited by the Navion's structural limitation, hydraulic actuator response, thrust, flaps, and elevator deflection limits.

## 4.3 Independent Axis Control Assumption and Effect of $M_{\dot{\alpha}}$

In the longitudinal case, all three equations of motion can be altered. For $\Delta\dot{u}$ equation, the Navion uses the throttle to generate thrust to compensate the difference in X derivatives. For the $\Delta\dot{w}$ equation, the flaps are used to generate lift to compensate the difference in Z derivatives. For the $\Delta\dot{q}$ equation, the elevator is used to generate pitching moment to compensate the difference in M derivatives.

The above paragraph assumes that each axis can be controlled independently, but there exists cross-coupling of axes that introduce unintended forces and moments. For example, when the elevator deflects to generate pitching moment to match $M_\alpha$ derivative, the elevator also produces drag and lift forces. The calculation method in this section is simplified to only consider one control for each axis.

Furthermore, in equation (3.34), the angle of attack feedback $\Delta\alpha$ not only affects $M_\alpha$, it also affects the term $M_{\dot{\alpha}}\frac{Z_\alpha}{u_0}$ (due to linear combination for $M_{\dot{\alpha}}\Delta\dot{\alpha}$). It is possible to obtain feedback gains required for each derivative by solving multiple simultaneous equations. However, in this section, the effect of $M_{\dot{\alpha}}$ will be ignored to keep the calculation process simple.

The omission of control cross-coupling contributions in different axes, and the omission of $M_{\dot{\alpha}}$ do not significantly degrade the overall in-flight simulation, as can be shown later in this chapter by Simulink demonstration. A more rigorous calculation method to include all cross-coupling control contributions and $M_{\dot{\alpha}}$ contribution will be described in CHAPTER 5 Longitudinal State-Space Representation.

## 4.4 Aerodynamic Stability Derivatives Matching

There are differences in calculating aerodynamic stability derivative gains versus control derivative gains. The calculations for both are described in this section.

### 4.4.1 $M_\alpha$ Feedback Gain

This example shows how to program the Variable Stability Navion to have the same "pitch stiffness" as the Twin Otter, that is, angle of attack feedback command to the Navion's elevator to match Twin Otter's $M_\alpha$.

Step 1. Calculating Navion $M_{\alpha.}$

$$\left(M_{\alpha}\right)_{Navion}=\frac{C_{m_{\alpha}}QS\overline{c}}{I_{yy}}=\frac{\left(-0.683\right)\left(36.9\text{ lbf/ft}^2\right)\left(184\text{ ft}^2\right)\left(5.70\text{ ft}\right)}{\left(3000\text{ slugs}\cdot\text{ft}^2\right)}=-8.80\text{ s}^{-2} \tag{4.3}$$

Step 2. Calculating Navion $M_{\delta e.}$

$$\left(M_{m_{\delta_e}}\right)_{Navion}=\frac{C_{m_{\delta_e}}QS\overline{c}}{I_{yy}}=\frac{\left(-0.923\right)\left(36.9\text{ lbf/ft}^2\right)\left(184\text{ ft}^2\right)\left(5.70\text{ ft}\right)}{\left(3000\text{ slugs}\cdot\text{ft}^2\right)}=-11.9\text{ s}^{-2} \tag{4.4}$$

Step 3. Calculating Twin Otter $M_{\alpha.}$ The Twin Otter stability derivatives are also given as dimensionless coefficients, which were obtained by extensive NASA flight test programs (Bragg et al. 2000).

$$\left(M_{\alpha}\right)_{TwinOtter}=\frac{C_{m_{\alpha}}QS\overline{c}}{I_{yy}}=\frac{\left(-1.31\right)\left(36.9\text{ lbf/ft}^2\right)\left(423\text{ ft}^2\right)\left(6.50\text{ ft}\right)}{\left(22100\text{ slugs}\cdot\text{ft}^2\right)}=-5.99\text{ s}^{-2} \tag{4.5}$$

Step 4. Matching Twin Otter $M_{\alpha}$ Using Angle of Attack Feedback to Elevator. The goal is to use the Navion to simulate a different $M_{\alpha}$ value. The Navion's elevator needs to deflect to provide the difference in pitching moment due to angle of attack. This can be accomplished by matching pitch rates $\Delta\dot{q}$ due to angle of attack:

$$\Delta\dot{q}_{TwinOtter}=\Delta\dot{q}_{Navion} \tag{4.6}$$

$$\left(M_{\alpha}\right)_{TwinOtter}\Delta\alpha=\left(M_{\alpha}\right)_{Navion}\Delta\alpha+\left(M_{\delta_e}\right)_{Navion}\left(\frac{\partial\delta_e}{\partial\alpha}\right)\Delta\alpha \tag{4.7}$$

divide both sides by $\Delta\alpha$ and rearrange to isolate the gain variable, $\frac{\partial\delta_e}{\partial\alpha}$:

$$\frac{\partial\delta_e}{\partial\alpha}=\frac{\left(M_{\alpha}\right)_{TwinOtter}-\left(M_{\alpha}\right)_{Navion}}{\left(M_{\delta_e}\right)_{Navion}}=\frac{\left(-5.99\right)-\left(-8.80\right)}{\left(-11.9\right)}=-0.236 \tag{4.8}$$

Step 5. Apply Calibration to Find Potentiometer Setting. Once the $\frac{\partial\delta_e}{\partial\alpha}$ gain value is determined, the corresponding potentiometer setting can be calculated or looked up on the calibration curve (Figure 4.5). The corresponding potentiometer setting is approximately 17.

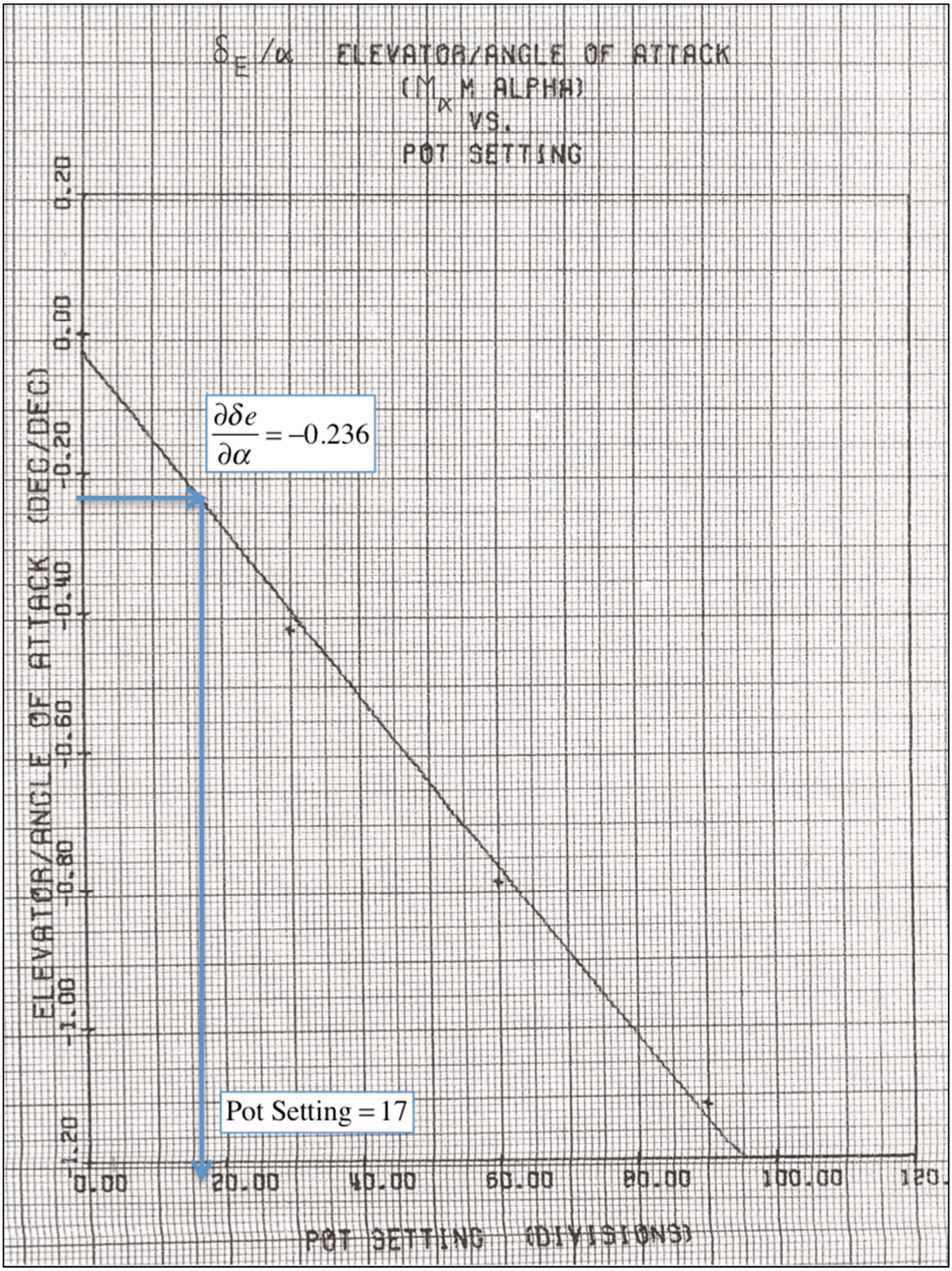


Figure 4.5 $M_\alpha$ Potentiometer Calibration Curve

The equation for this highly linear calibration curve is available from the Princeton calibration data manual (Nichols & Malagon, 1981). Throughout the years of the Variable Stability Navion's operations at Princeton and UTSI, aircraft sensors and electronics have undergone repairs and replacements. The available potentiometer calibration data will likely be outdated and invalid. However, for the purpose of this thesis, the old calibration data will be used to demonstrate the process. It is imperative that new calibrations be conducted on the aircraft prior to it re-entering to service. A suggested ground calibration test card is included in Table 14.1. The equation for the potentiometer calibration is:

$$\frac{\partial \delta_e}{\partial \alpha} = -0.0124 \times M_\alpha Potentiometer - 0.0254 \tag{4.9}$$

substituting the $\frac{\partial \delta_e}{\partial \alpha}$ value, and rearranging yields:

$$M_\alpha Potentiometer = \frac{\frac{\partial \delta_E}{\partial \alpha} + 0.0254}{-0.0124} = \frac{(-0.236) + 0.0254}{-0.0124} = 17.0 \tag{4.10}$$

Therefore, by dialing 17 on the potentiometer labeled $M_\alpha$, the Variable Stability Navion will now simulate the Twin Otter's $M_\alpha$ of -5.99 $s^{-2}$ at 104 KCAS.

The same process can be applied to match $M_u$ and $M_q$ derivatives to calculate feedback gains to elevator and corresponding potentiometer settings.

### 4.4.2 $X_\alpha$ Feedback Gain

The matching mechanism is similar for the X derivatives, but instead of using elevator, the Variable Stability Navion uses the throttle to accelerate and decelerate the aircraft in the X-axis. The objective is to match the $\Delta \dot{u}$ equation (3.34) term-by-term.

$$\Delta \dot{u}_{TwinOtter} = \Delta \dot{u}_{Navion} \tag{4.11}$$

$$\left(X_\alpha\right)_{TwinOtter} \Delta\alpha = \left(X_\alpha\right)_{Navion} \Delta\alpha + \left(X_\alpha\right)_{Navion} \left(\frac{\partial \delta_t}{\partial \alpha}\right) \Delta\alpha \tag{4.12}$$

The angle of attack to throttle feedback gain $\frac{\partial \delta_t}{\partial \alpha}$ is then isolated, and the corresponding $X_\alpha$ potentiometer setting can be calculated by the calibration equation.

The same process can be applied to match $X_u$ derivative to calculate feedback gain to throttle and its corresponding potentiometer setting.

### 4.4.3 $Z_\alpha$ Feedback Gain

For the Z derivatives, the Variable Stability Navion uses the flaps to control motions in the Z-axis. The objective is to match the $\Delta\dot{\alpha}$ equation (3.34) term-by-term.

$$\Delta\dot{\alpha}_{TwinOtter} = \Delta\dot{\alpha}_{Navion} \tag{4.13}$$

$$\left(\frac{Z_\alpha}{u_0}\right)_{TwinOtter}\Delta\alpha = \left(\frac{Z_\alpha}{u_0}\right)_{Navion}\Delta\alpha + \left(\frac{Z_{\delta_f}}{u_0}\right)_{Navion}\left(\frac{\partial\delta_f}{\partial\alpha}\right)\Delta\alpha \tag{4.14}$$

The angle of attack to flaps feedback gain $\frac{\partial\delta_f}{\partial\alpha}$ is then isolated, and the corresponding $Z_\alpha$ potentiometer setting can be calculated by the calibration equation.

The same process can be applied to match $Z_u$ and $Z_q$ derivatives to calculate feedback gains to flaps and potentiometer settings.

## 4.5 Control Derivatives Matching

Because the Variable Stability Navion has an irreversible flight control system on the evaluation pilot side, there are no physical linkages between the pilot's controls (stick, throttle lever, and flap lever) to the actual control surfaces (elevator, throttle, and flaps). In conventional reversible systems, the linkages between the cockpit control and control surfaces are referred to as gearing. The same analogy can be used to describe the control feed forward gains used in the Variable Stability Navion.

### 4.5.1 Basic Navion $M_{\delta e}$ Feed Forward Gain

The matching of control derivatives is different from the aerodynamic stability derivative procedure. When the Variable Stability Navion is to be flown as a basic Navion, the pilot simply has to turn all the aerodynamic stability derivative potentiometers to zero, which cuts the sensor feedbacks to the control surfaces. However, since the Variable Stability Navion has an irreversible flight control system, the pilot is required to program in the control surface to cockpit control inceptor gearing ratios. The $M_{\delta e}$ matching can be viewed as setting the stick to elevator gearing ratio.

Step 1. Find Stick to Elevator Gearing Ratio. The basic Navion has full pitch stick range $\Delta\delta_{PS}$ of 29.1°, and full elevator deflection authority $\Delta\delta_e$ of 48.3° (Nichols & Malagon, 1981). The gearing can be calculated as follows: (elevator trailing edge down, stick foward, and pitch up moment are positive)

$$\frac{\partial \delta_e}{\partial \delta_{PS}} = \frac{\delta_{e,\max} - \delta_{e,\min}}{\delta_{PS,\max} - \delta_{PS,\min}} = \frac{48.3°}{29.1°} = 1.66 \tag{4.15}$$

that is, for every 1° movement of the stick, 1.66° elevator deflection is commanded.

Step 2. Apply Calibration to Find Potentiometer Setting. The potentiometer setting can be looked up from the calibration curve, or calculated using the calibration equation:

$$\frac{\partial \delta_e}{\partial \delta_{PS}} = 0.0236 \times M_\alpha Potentiometer + 0.0055 \tag{4.16}$$

Substituting in the $\frac{\partial \delta_e}{\partial \delta_{PS}}$ value, and rearrange yields

$$M_{\delta_e} Potentiometer = \frac{\frac{\partial \delta_e}{\partial \delta_{PS}} + 0.0055}{-0.0236} = \frac{(1.66) + 0.0055}{-0.0236} = -70.6 \tag{4.17}$$

Therefore, the pilot needs to flip the toggle switch down for negative, and dial 71 on the potentiometer labeled $M_{\delta e}$. The Variable Stability Navion will now simulate the elevator to stick gearing of a basic unmodified Navion.

### 4.5.2 Twin Otter $M_{\delta e}$ Feed Forward Gain

To match the Twin Otter $M_{\delta e}$, is to match the aircraft response that the pilot expects when pulling or pushing on the stick. When the pilot pulls on the stick in a Twin Otter, the aircraft is expected to pitch up with a certain pitch rate. This example describes the procedure to mimic the control power of the Twin Otter in the Variable Stability Navion.

Step 1. Calculating Navion $M_{\delta e}$.

$$\left(M_{\delta_e}\right)_{Navion} = \frac{C_{m_{\delta_e}} Q S \bar{c}}{I_{yy}} = \frac{(-0.923)\left(36.9 \text{ lbf/ft}^2\right)\left(184 \text{ ft}^2\right)(5.70 \text{ ft})}{\left(3000 \text{ slugs} \cdot \text{ft}^2\right)} = -11.9 \text{ s}^{-2} \tag{4.18}$$

Step 2. Calculating Twin Otter $M_{\delta e}$.

$$\left(M_{\delta_e}\right)_{TwinOtter} = \frac{C_{m_{\delta_e}} Q S \bar{c}}{I_{yy}} = \frac{(-1.74)\left(36.9 \text{ lbf/ft}^2\right)\left(423 \text{ ft}^2\right)(6.50 \text{ ft})}{\left(22100 \text{ slugs} \cdot \text{ft}^2\right)} = -7.96 \text{ s}^{-2} \tag{4.19}$$

Step 3. Matching Twin Otter $M_{\delta e}$ using pitch stick feed forward gain to elevator. The control derivatives are the relationship between the elevator angle and the pitching moment that it generates. To find the elevator angle, one must know the gearing ratio between the pitch stick $\Delta\delta_{PS}$ and the elevator deflection angle $\Delta\delta_e$. The basic Navion's elevator to stick gearing ratio $\frac{\partial\delta_e}{\partial\delta_{ps}}$ is known, but the Twin Otter's gearing ratio is not readily available. For this reason, one can assume the Twin Otter has the same elevator to stick gearing ratio as the basic Navion. The objective is once again, to match the $\Delta\dot{q}$ equations.

$$\left(M_{\delta_e}\right)_{TwinOtter}\left(\frac{\partial\delta_e}{\partial\delta_{PS}}\right)_{gearing}\Delta\delta_{PS}=\underbrace{\left(M_{\delta_e}\right)_{Navion}\left(\frac{\partial\delta_e}{\partial\delta_{PS}}\right)_{gearing}\Delta\delta_{PS}}_{\text{pilot commanded elevator deflection}}+\underbrace{\left(M_{\delta_e}\right)_{Navion}\left(\frac{\partial\delta_e}{\partial\delta_{PS}}\right)_{extra}\Delta\delta_{PS}}_{\text{extra elevator deflection to match Twin Otter}} \quad (4.20)$$

where the total required elevator to stick feed forward gain is:

$$\frac{\partial\delta_e}{\partial\delta_{PS}}=\left(\frac{\partial\delta_e}{\partial\delta_{PS}}\right)_{gearing}+\left(\frac{\partial\delta_e}{\partial\delta_{PS}}\right)_{extra} \quad (4.21)$$

substituting equation (4.21) into (4.20), the equation becomes:

$$\left(M_{\delta_e}\right)_{TwinOtter}\left(\frac{\partial\delta_e}{\partial\delta_{PS}}\right)_{gearing}\Delta\delta_{PS}=\left(M_{\delta_e}\right)_{Navion}\frac{\partial\delta_e}{\partial\delta_{PS}}\Delta\delta_{PS} \quad (4.22)$$

dividing both sides by $\Delta\delta_{PS}$, and rearranging to isolate $\frac{\partial\delta_e}{\partial\delta_{PS}}$:

$$\frac{\partial\delta_e}{\partial\delta_{PS}}=\frac{\left(M_{\delta_e}\right)_{TwinOtter}\left(\frac{\partial\delta_e}{\partial\delta_{PS}}\right)_{gearing}}{\left(M_{\delta_e}\right)_{Navion}}=\frac{\left(-7.96\ \mathrm{s}^{-1}\right)\times(1.66)}{\left(-11.9\ \mathrm{s}^{-1}\right)}=1.11 \quad (4.23)$$

Step 4. Applying Calibration to Find Potentiometer Setting. As previously shown, the equation for the calibration curve can be used to find the potentiometer setting:

$$M_{\delta_e}Potentiometer=-\frac{\frac{\partial\delta_e}{\partial\delta_{PS}}+0.0055}{-0.0236}=\frac{(1.11)+0.0055}{-0.0236}=-47.3 \quad (4.24)$$

Therefore, the pilot can now place the toggle switch down for negative, and dial 47 on the potentiometer labeled $M_{\delta e}$. The Variable Stability Navion will now simulate the Twin

Otter's $M_{\delta e}$ of -7.96 $s^{-2}$. Comparing to the basic Navion's $M_{\delta e}$ of -11.9 $s^{-2}$, a reduction in elevator power will be simulated.

### 4.5.3 $Z_{\delta_e}$ Feed Forward Gain

The objective is once again, to match the $\Delta\dot{\alpha}$ equations.

$$\left(\frac{Z_{\delta_e}}{u_0}\right)_{TwinOtter}\left(\frac{\partial\delta_e}{\partial\delta_{PS}}\right)_{gearing}\Delta\delta_{PS}=\left(\frac{Z_{\delta_e}}{u_0}\right)_{Navion}\left(\frac{\partial\delta_e}{\partial\delta_{PS}}\right)_{gearing}\Delta\delta_{PS}+\left(\frac{Z_{\delta f}}{u_0}\right)_{Navion}\frac{\partial\delta_f}{\partial\delta_{PS}}\Delta\delta_{PS} \qquad (4.25)$$

multiplying both sides by $\frac{u_0}{\Delta\delta_{PS}}$, and rearranging to isolate $\frac{\partial\delta_f}{\partial\delta_{PS}}$:

$$\frac{\partial\delta_f}{\partial\delta_{PS}}=\frac{\left[\left(Z_{\delta_e}\right)_{TwinOtter}-\left(Z_{\delta_e}\right)_{Navion}\right]\left(\frac{\partial\delta_e}{\partial_{PS}}\right)_{gearing}}{\left(Z_{\delta_f}\right)_{Navion}}=\frac{\left[\left(-33.3\right)-\left(-28.2\right)\right]\times(1.66)}{\left(-70.9\right)}=0.119 \qquad (4.26)$$

the $Z_{\delta_e}$ potentiometer setting can now be calculated or looked up from calibration curve.

## 4.6 Twin Otter Longitudinal Setup Summary

As shown by the above examples, to program the Variable Stability Navion to simulate the longitudinal flying characteristic of a Twin Otter would require each dimensional stability derivative to be matched and each potentiometer setting calculated individually. This cumbersome process can be simplified by using MATLAB scripts to calculate all the potentiometer settings. The MATLAB scripts are in APPENDIX F.

Table 4.1 summarizes the potentiometer settings that the pilot needs to dial into the Variable Stability Navion to simulate a Twin Otter for standard atmospheric condition at trimmed calibrated airspeed of 104 KCAS, at any altitude.

Table 4.1 Longitudinal Potentiometer Setting Summary

| Potentiometer Label | Gains | Potentiometer Settings |
|---|---|---|
| $X_u$ | 0.00544 | 1 |
| $X_\alpha$ | -0.699 | -1 |
| $X_{\delta t}$ | 0.931 | -51 |
| $Z_u$ | -0.00188 | -6 |
| $Z_\alpha$ | -0.524 | 9 |
| $Z_q$ | 0.216 | N/A |
| $Z_{\delta e}$ | 0.121 | 4 |
| $Z_{\delta f}$ | 10.0 | 20 |
| $M_u$ | 0 | 0 |
| $M_\alpha$ | -0.236 | 17 |
| $M_q$ | 0.0682 | -5 |
| $M_{\delta e}$ | 1.11 | -47 |

## 4.7 Simulink Simulation Results

The next step is to show that once the Variable Stability Navion is configured with the required potentiometer settings, the Variable Stability Navion behaves the same way as the Twin Otter when disturbed. The time histories match well as shown in Figure 4.6.

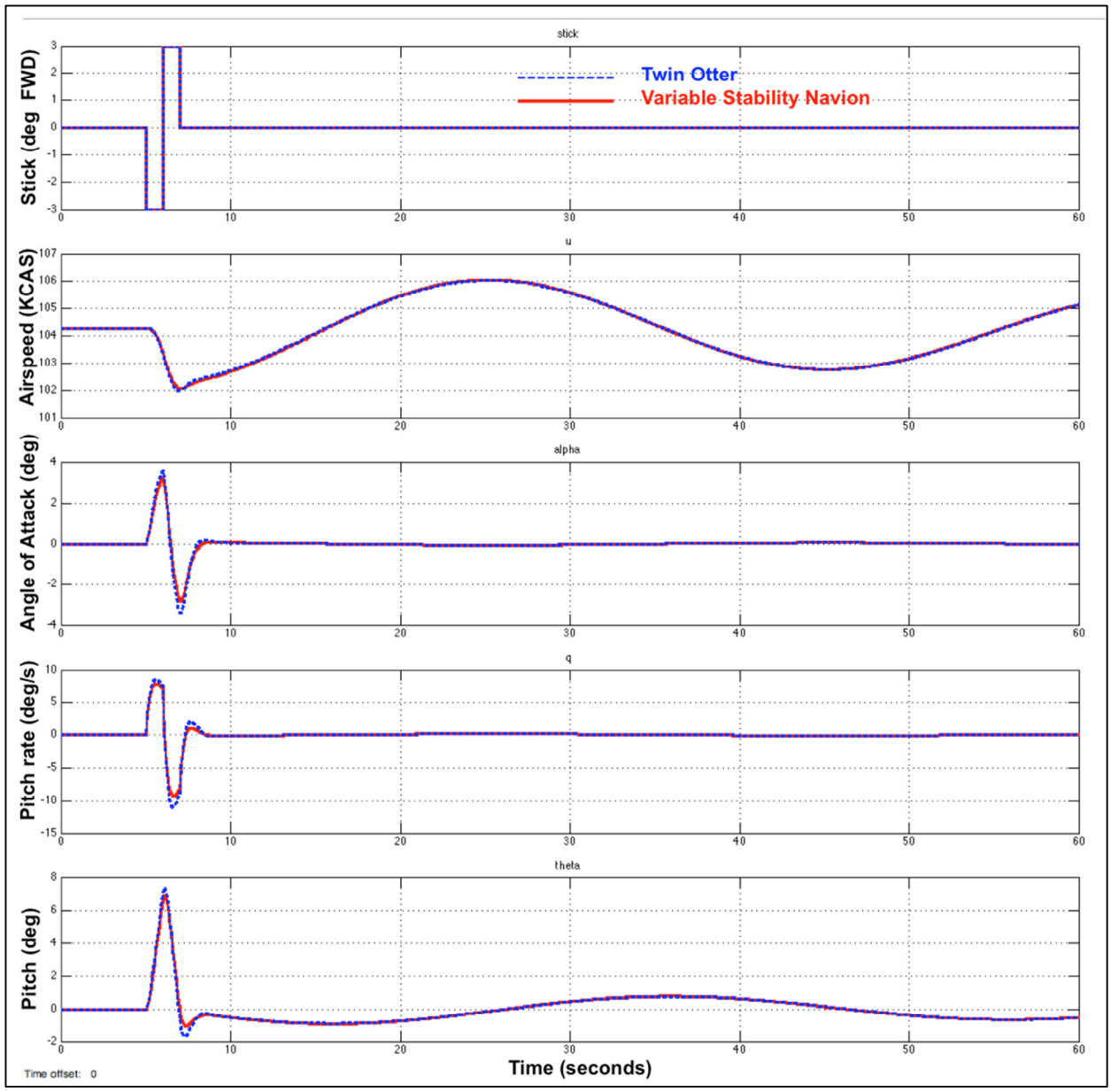


Figure 4.6 Variable Stability Navion Simulating Twin Otter Longitudinal Response

It is interesting to see what the Variable Stability Navion is doing to mimic the Twin Otter. The time histories of the Navion's elevator, throttle, and flap modulations can show how the Navion uses the response feedbacks to control the aircraft.

For the evaluation pilot, the elevator, throttle, and flap movements are not apparent because the irreversible flight control system does not relay those movements to control stick, throttle, and flap levers. However, the safety pilot will see his control stick, throttle, and flap levers move because they are directly connected to the control surfaces.

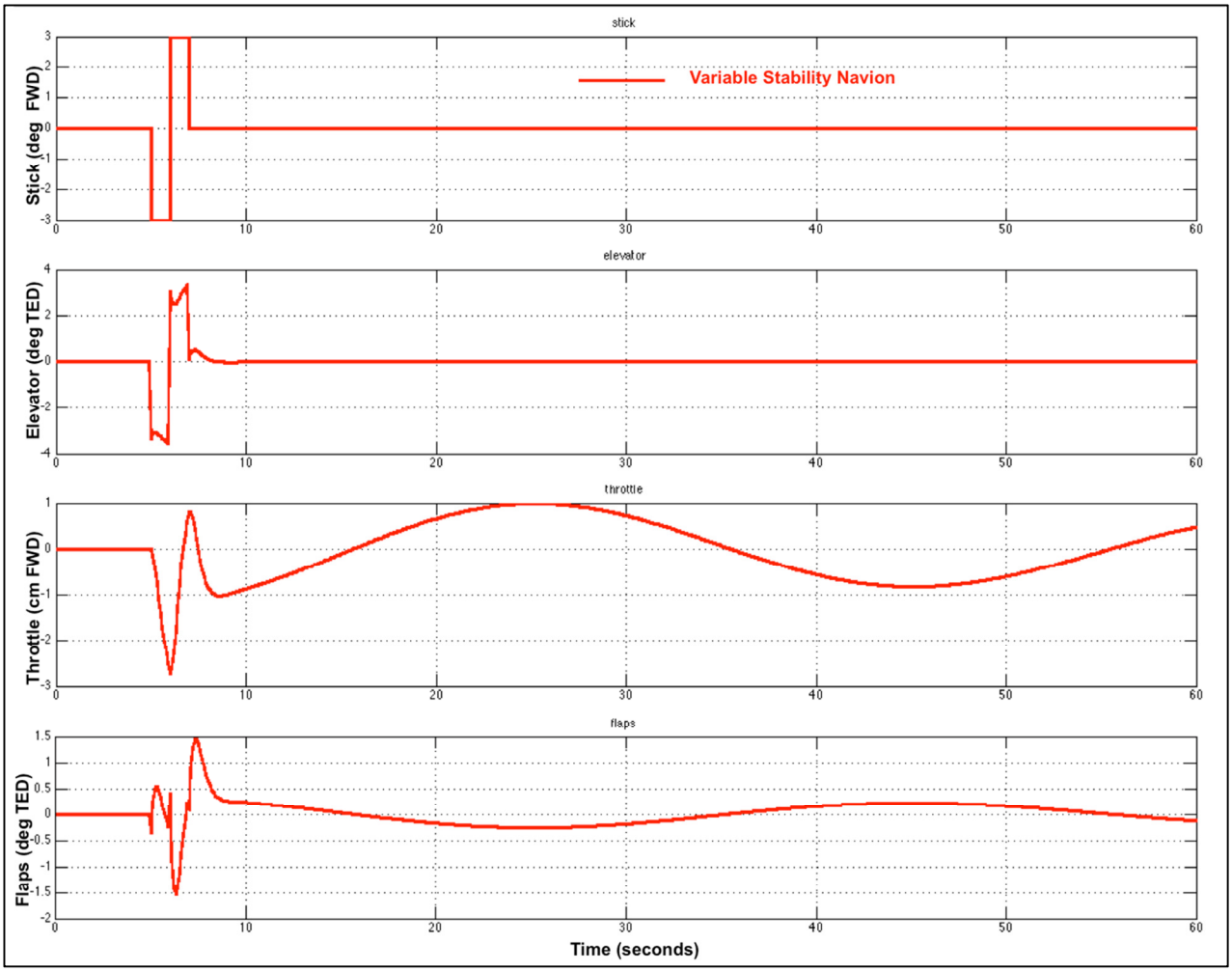


Figure 4.7 Elevator, Throttle, and Flaps Commanded by Feedback

# CHAPTER 5
# LONGITUDINAL STATE-SPACE REPRESENTATION

In this chapter, an alternative method of calculating individual potentiometer settings is presented. The previous method of term-by-term matching is simple and intuitive; however, the following method using state-space is more elegant and complete.

## 5.1 State-Space Representation

The linearized longitudinal equations of motion (3.34) are simple, ordinary linear differential equations with constant coefficients, made up of aircraft stability derivatives, mass and inertia. These equations can now be written in matrix form, called the state-space form. This form is well suited for software such as MATLAB to solve. Rewriting equation (3.34) yields the following state-space representation:

$$\begin{bmatrix} \Delta\dot{u} \\ \Delta\dot{\alpha} \\ \Delta\dot{q} \\ \Delta\dot{\theta} \end{bmatrix} = \begin{bmatrix} X_u & X_\alpha & -w_0 & -g\cos\theta_0 \\ \dfrac{Z_u}{u_0} & \dfrac{Z_\alpha}{u_0} & \dfrac{1+Z_q}{u_0} & -\dfrac{g}{u_0}\sin\theta_0 \\ M_u + M_{\dot\alpha}\dfrac{Z_u}{u_0} & M_\alpha + M_{\dot\alpha}\dfrac{Z_\alpha}{u_0} & M_q + M_{\dot\alpha}\dfrac{1+Z_q}{u_0} & -M_{\dot\alpha}\dfrac{g}{u_0}\sin\theta_0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta\alpha \\ \Delta q \\ \Delta\theta \end{bmatrix} + \begin{bmatrix} 0 & X_{\delta_t} & 0 \\ \dfrac{Z_{\delta_e}}{u_0} & 0 & \dfrac{Z_{\delta_f}}{u_0} \\ M_{\delta_e} + M_{\dot\alpha}\dfrac{Z_{\delta_e}}{u_0} & 0 & M_{\dot\alpha}\dfrac{Z_{\delta_f}}{u_0} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\delta_e \\ \Delta\delta_t \\ \Delta\delta_f \end{bmatrix} \tag{5.1}$$

The above equation can be represented as

$$\dot{x} = Ax + B\eta \tag{5.2}$$

where $\boldsymbol{x}$ is the state vector, $\boldsymbol{\eta}$ is the control vector, $\boldsymbol{A}$ is the plant matrix, and $\boldsymbol{B}$ is the control matrix.

The longitudinal equations of motion, when written as state-space form in equation (5.1), can be manipulated and solved easily by MATLAB. The objective is to find the set of feedback gain values that can augment the Navion's plant matrix, $A_{Navion}$, to the Twin Otter's plant matrix $A_{Twin\ Otter}$.

Step 1. Find Navion Plant Matrix, $A_{Navion}$. This can be calculated by calculating all dimensional stability derivatives using formulas listed in Table 3.1, and calculated in Table 3.2.

Usually, the aircraft's initial attitude $\theta_0$ is given; in this case however, the $\theta_0$ value is not readily available. Therefore, to further simplify the state-space representation, the initial pitch attitude $\theta_0$ is assumed to be near 0 for conventional aircraft trimmed for straight and level flight. The initial normal velocity $w_0$ can also be assumed to be zero, and the forward airspeed $u_0$ simply becomes $V_{trim}$.

$$A_{Navion} = \begin{bmatrix} -0.0450 & 6.34 & 0 & -32.2 \\ -0.210 & -2.02 & 0.972 & 0 \\ 0.00190 & -6.95 & -2.96 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{5.3}$$

Step 2. Find Twin Otter Plant Matrix, $A_{Twin\ Otter}$. Dimensional derivatives already calculate in Table 3.3 are inserted in the matrix.

$$A_{TwinOtter} = \begin{bmatrix} -0.0255 & 3.83 & 0 & -32.2 \\ -0.0013 & -1.77 & 0.885 & 0 \\ 0 & -5.99 & -2.89 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{5.4}$$

Step 3. Find Navion Control Matrix, $B_{Navion}$.

$$B_{Navion} = \begin{bmatrix} 0 & 3.59 & 0 \\ -0.160 & 0 & -0.403 \\ -11.7 & 0 & 0.366 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.5}$$

Step 4. Find Feedback Gain Matrix, $K$. To match the Navion's plant matrix to the Twin Otter's, a state feedback gain matrix is required. The equation of plant matrix matching using positive feedback gain is:

$$A_{TwinOtter} = A_{Navion} + B_{Navion}K \tag{5.6}$$

The process of isolating and calculating $K$ involves complex matrix operations, but MATLAB can handle it easily using the command:

$$K = B_{Navion} \setminus (A_{TwinOtter} - A_{Navion}) \tag{5.7}$$

where

$$K = \begin{bmatrix} \frac{\partial \delta_e}{\partial u} & \frac{\partial \delta_e}{\partial \alpha} & \frac{\partial \delta_e}{\partial q} & \frac{\partial \delta_e}{\partial \theta} \\ \frac{\partial \delta_t}{\partial u} & \frac{\partial \delta_t}{\partial \alpha} & \frac{\partial \delta_t}{\partial q} & \frac{\partial \delta_t}{\partial \theta} \\ \frac{\partial \delta_f}{\partial u} & \frac{\partial \delta_f}{\partial \alpha} & \frac{\partial \delta_f}{\partial q} & \frac{\partial \delta_f}{\partial \theta} \end{bmatrix} = \begin{bmatrix} 0.0001 & -0.1004 & -0.0005 & 0 \\ -0.0054 & -0.6984 & 0 & 0 \\ 0.0019 & -0.5766 & 0.2158 & 0 \end{bmatrix} \tag{5.8}$$

Step 5. Find Gains for the Controls. To match $Z_{\delta e}$ and $M_{\delta e}$ requires solving the $\Delta\dot{\alpha}$ and $\Delta\dot{q}$ equations simultaneously for $\frac{\partial \delta_e}{\partial \delta_{PS}}$ and $\frac{\partial \delta_f}{\partial \delta_{PS}}$:

$$\left(\frac{Z_{\delta_e}}{u_0}\right)_{TwinOtter}\left(\frac{\partial \delta_e}{\partial \delta_{PS}}\right)_{gearing}\Delta\delta_{PS} = \left(\frac{Z_{\delta_e}}{u_0}\right)_{Navion}\frac{\partial \delta_e}{\partial \delta_{PS}}\Delta\delta_{PS} + \left(\frac{Z_{\delta_f}}{u_0}\right)_{Navion}\frac{\partial \delta_f}{\partial \delta_{PS}}\Delta\delta_{PS}$$

$$\begin{aligned}\left(M_{\delta_e} + M_{\dot{\alpha}}\frac{Z_{\delta_e}}{u_0}\right)_{TwinOtter}\left(\frac{\partial \delta_e}{\partial \delta_{PS}}\right)_{gearing}\Delta\delta_{PS} = &\left(M_{\delta_e} + M_{\dot{\alpha}}\frac{Z_{\delta_e}}{u_0}\right)_{Navion}\frac{\partial \delta_e}{\partial \delta_{PS}}\Delta\delta_{PS} \\ &+\left(M_{\dot{\alpha}}\frac{Z_{\delta_f}}{u_0}\right)_{Navion}\frac{\partial \delta_f}{\partial \delta_{PS}}\Delta\delta_{PS}\end{aligned} \tag{5.9}$$

Step 6. Apply Calibration to Find Potentiometer Setting. Using calibration equations from the calibration manual, the following table summarizes the required potentiometer settings to simulate Twin Otter longitudinally.

Table 5.1 Longitudinal Potentiometer Setting by State-Space Method

| Potentiometer Label | Gains | Potentiometer Settings |
|---|---|---|
| $X_u$ | 0.00544 | 1 |
| $X_\alpha$ | -0.698 | -1 |
| $X_{\delta t}$ | 0.931 | -51 |
| $Z_u$ | -0.00192 | -6 |
| $Z_\alpha$ | -0.577 | 10 |
| $Z_q$ | 0.216 | N/A |
| $Z_{\delta e}$ | 0.329 | 12 |
| $Z_{\delta f}$ | 10.0 | 20 |
| $M_u$ | 0.000103 | -2 |
| $M_\alpha$ | -1.00 | 6 |
| $M_q$ | 0.000472 | 0 |
| $M_{\delta e}$ | 1.13 | -47 |

## 5.2 Simulink Simulation Results

The state-space method yields perfect time history match as shown in Figure 5.1. The result is better than the term-by-term method because it takes cross-coupling and $\Delta\dot{\alpha}$ into consideration during calculation.

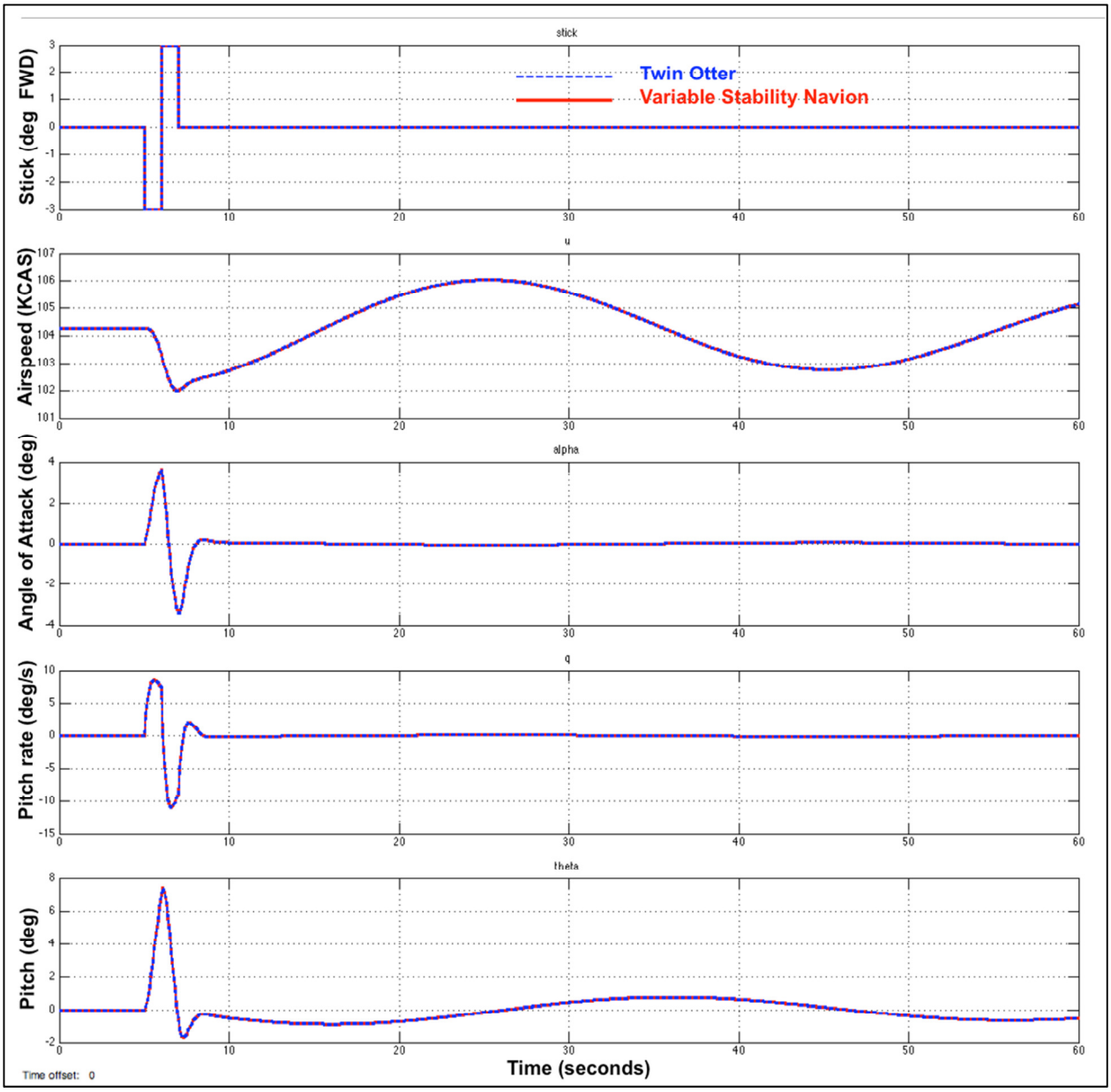


Figure 5.1 Navion Matching to Twin Otter Using State-Space Method

# CHAPTER 6
# SHORT PERIOD OSCILLATION AND PHUGOID APPROXIMATIONS

The pilot is not specifically aware of the individual stability derivatives when conducting routine flying tasks such as pitch attitude capture and maintenance. What the pilot perceives in the cockpit is the pitch oscillations, which are described as short period mode and long period mode (phugoid). These oscillations can be modeled as simple mass-spring-damper dynamic systems, with specific damping ratios $\zeta$ and natural frequencies $\omega_n$. These parameters give a more complete picture of the aircraft's flying qualities.

The general mass-spring-damper second-order differential equation is:

$$\frac{d^2x}{dt^2} + 2\zeta\omega_n \frac{dx}{dt} + \omega_n^2 x = f(t) \tag{6.1}$$

## 6.1 Approximation Equations

The phugoid is lightly damped and has a long period; whereas the short period mode is usually heavily damped and has a very short period. To clearly illustrate both dynamic modes, Figure 6.1 shows an aircraft disturbed from trimmed pitch attitude, and the short period oscillation is triggered. The short period oscillation damps out quickly, and the remaining motion is the phugoid (with a period of 30 seconds).

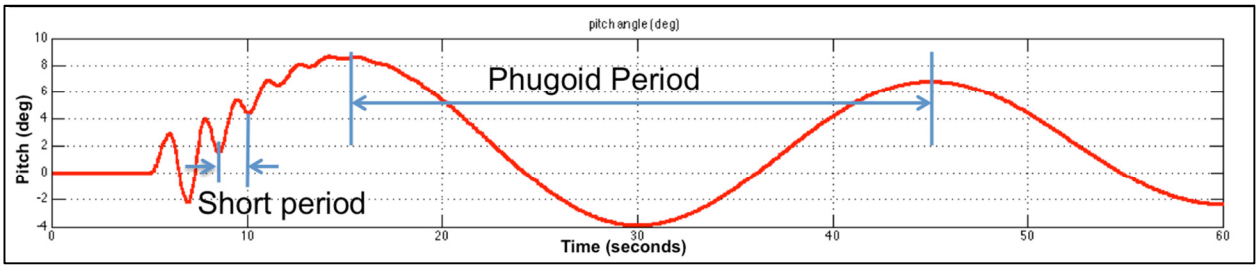


Figure 6.1 The Phugoid and Short Period Motions

The longitudinal stick-fixed dynamic response approximations are summarized in Table 6.1. The phugoid approximations are derived with the assumption that the change in angle of attack is negligible. The short period approximations are derived with the assumption that the oscillation is too short for change in longitudinal velocity (Nelson, 2007, p.155).

Table 6.1 Phugoid and Short-Period Mode Approximations

| | Long Period (Phugoid) | Short Period |
|---|---|---|
| Natural Frequency | $\omega_{np} = \sqrt{\frac{-Z_u g}{u_0}}$ | $\omega_{nsp} = \sqrt{\frac{Z_\alpha M_q}{u_0} - M_\alpha}$ |
| Damping Ratio | $\varsigma_p = \frac{-X_u}{2\omega_{np}}$ | $\varsigma_{sp} = -\frac{M_q + M_{\dot{\alpha}} + \frac{Z_\alpha}{u_0}}{2\omega_{nsp}}$ |

The phugoid and short period for both the Navion and Twin Otter can be easily calculated by using the short period and phugoid estimation equations listed above. The results are listed in Table 6.2 and Table 6.3. For comparison, the exact solutions calculated by MATLAB damp() function are also presented. In general, the short period approximation is much more accurate than the phugoid approximation.

Table 6.2 Basic Navion Phugoid and Short Period Approximations

| Basic Navion | Short Period | | | Phugoid | | |
|---|---|---|---|---|---|---|
| | Damping Ratio ζ | Natural Frequency $\omega_n$ (rad/s) | Undamped Period (s) | Damping Ratio ζ | Natural Frequency $\omega_n$ (rad/s) | Undamped Period (s) |
| Approximation Equations | 0.569 | 3.61 | 1.74 | 0.0867 | 0.260 | 24.2 |
| Exact Solution by MATLAB damp() function | 0.699 | 3.57 | 1.76 | 0.0783 | 0.216 | 29.1 |
| Difference | -19% | 1% | -1% | 11% | 20% | -17% |

Table 6.3 Twin Otter Phugoid and Short Period Approximations

| Twin Otter | Short Period | | | Phugoid | | |
|---|---|---|---|---|---|---|
| | Damping Ratio ζ | Natural Frequency $\omega_n$ (rad/s) | Undamped Period (s) | Damping Ratio ζ | Natural Frequency $\omega_n$ (rad/s) | Undamped Period (s) |
| Approximation Equations | 0.699 | 3.34 | 1.88 | 0.0614 | 0.208 | 30.2 |
| Exact Solution by MATLAB damp() function | 0.722 | 3.23 | 1.95 | 0.0502 | 0.157 | 40.0 |
| Difference | -3% | 3% | -4% | 22% | 32% | -25% |

## 6.2 Short Period Matching

The previous two methods (term-by-term, and state-space) assume all the stability derivatives are known for both the basic Navion, and the desired aircraft to be simulated.

If the in-flight simulation is focused on the flying qualities of the desired aircraft, then the following method will demonstrate how to match a desired aircraft's dynamic response rather than matching all stability derivatives.

The short period oscillation approximation equations listed in Table 6.1 can be used to generate specific damping ratio $\zeta_{sp}$ and natural frequency $\omega_{nsp}$. In this example, the Twin Otter's short period will be matched by the Variable Stability Navion. That is, matching short period approximation $\zeta_{sp}$ of 0.699, and $\omega_{nsp}$ of 3.34 rad/s.

The Short Period approximation equations are (Nelson, p.155):

$$\omega_{nsp} = \sqrt{\frac{Z_\alpha M_q}{u_0} - M_\alpha} \tag{6.2}$$

$$\zeta_{sp} = -\frac{M_q + M_{\dot{\alpha}} + \frac{Z_\alpha}{u_0}}{2\omega_{nsp}} \tag{6.3}$$

Although there are four variables (dimensional derivatives) in two equations, by fixing the $Z_\alpha$ and $M_{\dot{\alpha}}$ to the basic Navion values, the two equations can be solved simultaneously for $M_q$ and $M_\alpha$. Also, the benefit of fixing $Z_\alpha$ is that only the elevator

feedback is required to match short period oscillations. The system of equations to be solved becomes:

$$M_q = -2\zeta_{sp}\omega_{nsp} - M_{\dot{\alpha}} - \frac{Z_\alpha}{u_0} \tag{6.4}$$

$$M_\alpha = \frac{Z_\alpha M_q}{u_0} - \omega_{nsp}^2 \tag{6.5}$$

The following dimensional stability derivatives can be used to generate Twin Otter's short period oscillation:

$$M_q = -2(0.699)(3.34 \text{ rad/s}) - (-0.910 \text{ s}^{-1}) - \frac{-356 \text{ ft/s}^2}{176 \text{ ft/s}} = -1.74 \text{ s}^{-1} \tag{6.6}$$

$$M_\alpha = \frac{(-356 \text{ ft/s}^2)(-1.74 \text{ s}^{-1})}{176 \text{ ft/s}} - (3.34 \text{ rad/s})^2 = -7.64 \text{ s}^{-2} \tag{6.7}$$

Using the same term-by-term dimensional stability derivative matching procedure as described in paragraph 4.4.1, the gain and potentiometer settings for $M_\alpha$ and $M_q$ can be calculated.

Table 6.4 Twin Otter Short Period Matching

| Potentiometer Label | Gains | Potentiometer Settings |
|---|---|---|
| $M_\alpha$ | -0.0974 | 6 |
| $M_q$ | -0.0288 | 1 |

### 6.2.1 Proposed Training Syllabus

Flight test pilot schools use variable stability aircraft to demonstrate a wide range of flying qualities. Calspan SRl Corporation created a training syllabus for this purpose using a Variable Stability Learjet. The combinations of different short period $\zeta_{sp}$ and $\omega_{nsp}$ as suggested by Calspan (Ball et al., 1993, p.2-24) can be adopted by the Variable Stability Navion:

Table 6.5 Short Period Simulation Potentiometer Settings

| Description | $\omega_{nsp}$ (rad/s) | $\zeta_{sp}$ | $\frac{\partial \delta_e}{\partial \alpha}$ | $M_\alpha$ Pot | $\frac{\partial \delta_e}{\partial q}$ | $M_q$ Pot |
|---|---|---|---|---|---|---|
| Med Freq Med Damp | 4 | 0.5 | -0.424 | -36 | -0.0851 | 5 |
| Med Freq Low Damp | 4 | 0.2 | -0.832 | -69 | -0.287 | 17 |
| Med Freq Zero Damp | 4 | 0 | 1.10 | -91 | -0.421 | 25 |
| Med Freq High Damp | 4 | 0.7 | 0.152 | -14 | 0.0494 | -3 |
| Fast High Damp | 6 | 0.7 | 1.36 | -111 | 0.285 | -18 |
| Very Fast High Damp | 8 | 0.7 | 3.23 | -263 | 0.520 | -32 |
| Slow High Damp | 2 | 0.7 | -0.381 | 29 | -0.186 | 11 |
| Twin Otter | 3.34 | 0.699 | -0.0974 | 6 | -0.0288 | 1 |

For the proposed training syllabus, the following time histories confirm that the desired short period responses are simulated:

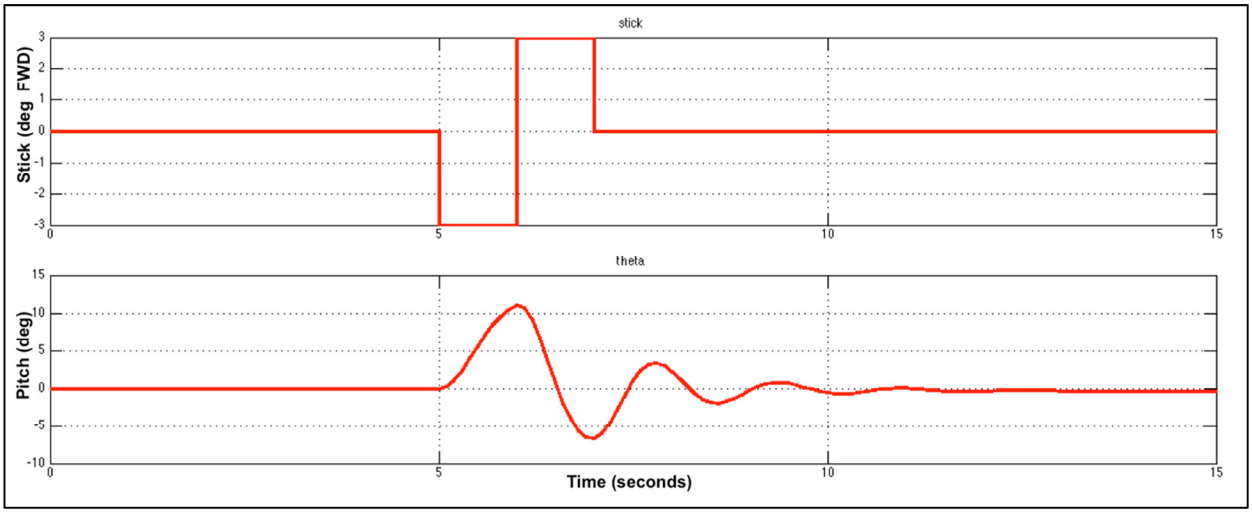


Figure 6.2 Short Period $\zeta_{sp}$=0.2 and $\omega_{nsp}$=4 rad/s Matching

## 6.3 Phugoid Matching

The long period oscillation, or phugoid, is usually described by the damped period $T_p$ and damping ratio $\zeta_p$. The phugoid approximation equations are (Nelson, p.155):

$$\omega_{n_p} = \frac{\frac{2\pi}{T_p}}{\sqrt{1-\zeta_p^2}} = \sqrt{\frac{-Z_u g}{u_0}} \tag{6.8}$$

$$\zeta_p = \frac{-X_u}{2\omega_{n_p}} \tag{6.9}$$

isolating the dimensional derivatives yields:

$$Z_u = -\frac{u_0 \omega_{n_p}^2}{g} \tag{6.10}$$

$$X_u = -2\zeta_p \omega_{np} \tag{6.11}$$

For phugoid, the desired $\zeta_p$ and $\omega_{np}$ can easily be simulated by calculating the required $Z_u$ and $X_u$. The $Z_u$ and $X_u$ values can then be equated to potentiometer settings. To simulate the Twin Otter phugoid, and the basic Learjet's phugoid ($T_p$=75 seconds, $\zeta_p$=0.05), the Variable Stability Navion will need the following potentiometer settings based on the above calculations.

Table 6.6 Phugoid Simulation Potentiometer Settings

| Description | $\omega_{np}$ (rad/s) | $\zeta_p$ | $\frac{\partial \delta_f}{\partial u}$ | $Z_u$ Pot | $\frac{\partial \delta_t}{\partial u}$ | $X_u$ Pot |
|---|---|---|---|---|---|---|
| Twin Otter | 0.208 | 0.0614 | -0.0019 | -6 | 0.0054 | 1 |
| Learjet | 0.0839 | 0.05 | -0.0047 | -15 | 0.0102 | 0 |

Matching the short period and phugoid using approximation equations yields good results for matching the dynamic response. Figure 6.1 shows the Variable Stability Navion matching the Twin Otter's dynamic response closely.

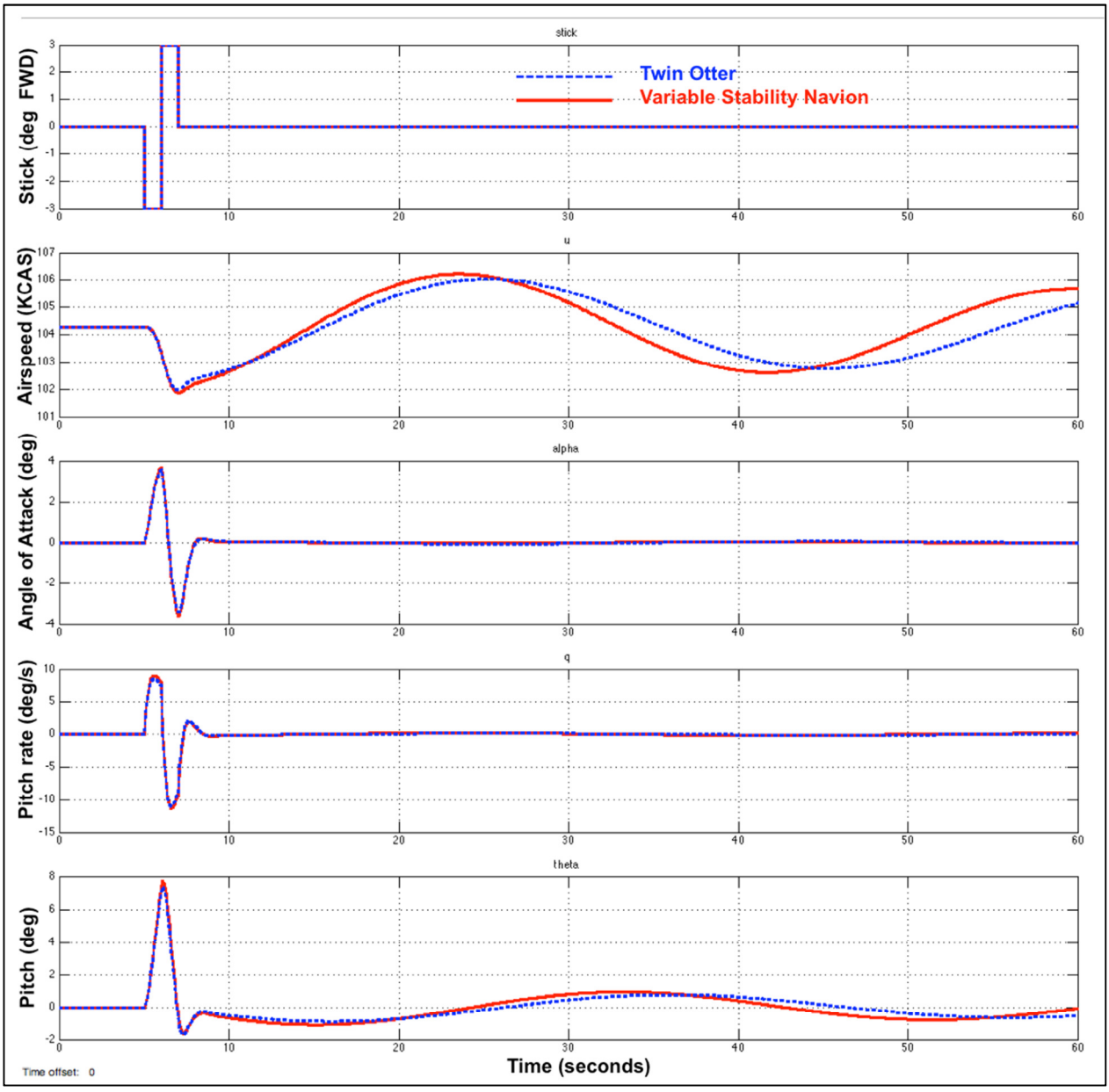


Figure 6.3 Estimated Short Period and Phugoid Matching for Twin Otter

For the proposed training syllabus, the following time histories confirm that the desired phugoid is simulated:

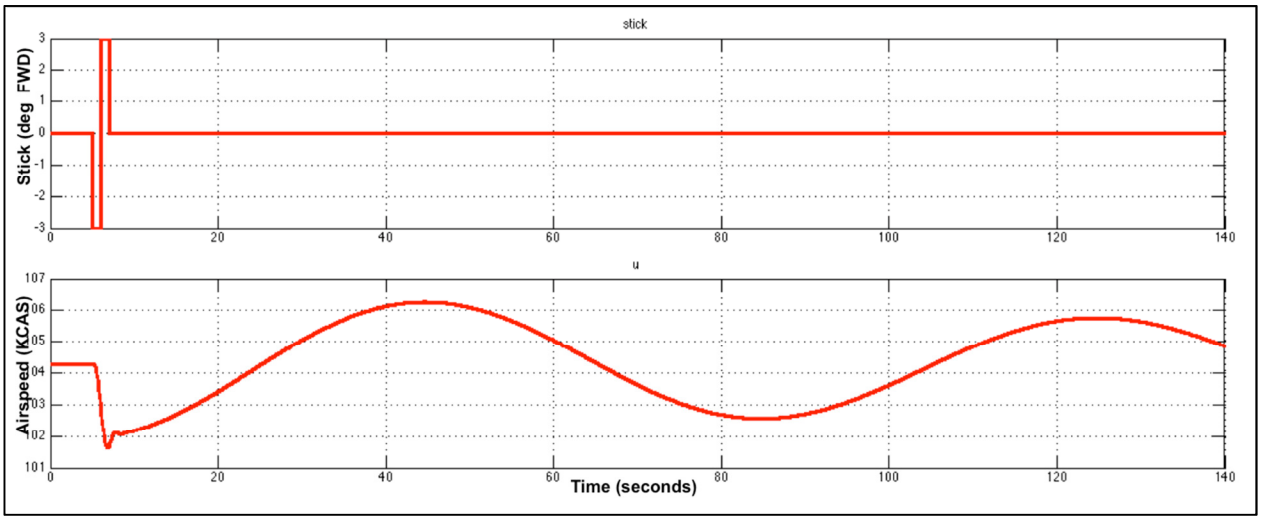


Figure 6.4 Phugoid Period $T_p$=75 s and $\zeta_p$ =0.05 Matching

The above examples show that to simulate both short period oscillation and phugoid requires adjusting four potentiometers: $M_q$, $Z_u$, $M_\alpha$, and $X_u$. The control law requires feedback gains to the elevator, throttle, and flaps.

# CHAPTER 7
# LONGITUDINAL POLE PLACEMENT METHOD

The short period and phugoid estimation and matching method described in CHAPTER 7 are sufficient for in-flight demonstration of different natural frequencies and damping ratios. The estimation method requires feedback gains to all three controls: elevator, throttle, and flaps.

With the help of MATLAB, a more elegant and more accurate method can be used to match the desired short period and phugoid. It is possible to use only the elevator to simulate both short period oscillation and phugoid. The pole placement method can quickly accomplish this task.

## 7.1 Pole Placement Method

Step 1. Assembling Short Period and Phugoid Poles. On the complex plane, one can locate the characteristic roots (eigenvalues) for both the short period and phugoid. The roots are represented as complex conjugates on the complex plane as shown in Figure 7.1. The poles (eigenvalues) are defined by the following equations:

$$\lambda_{sp} = -\zeta_{sp}\omega_{n_{sp}} \pm i\omega_{n_{sp}}\sqrt{1-\zeta_{sp}^{2}} \tag{7.1}$$

$$\lambda_{p} = -\zeta_{p}\omega_{n_{p}} \pm i\omega_{n_{p}}\sqrt{1-\zeta_{p}^{2}} \tag{7.2}$$

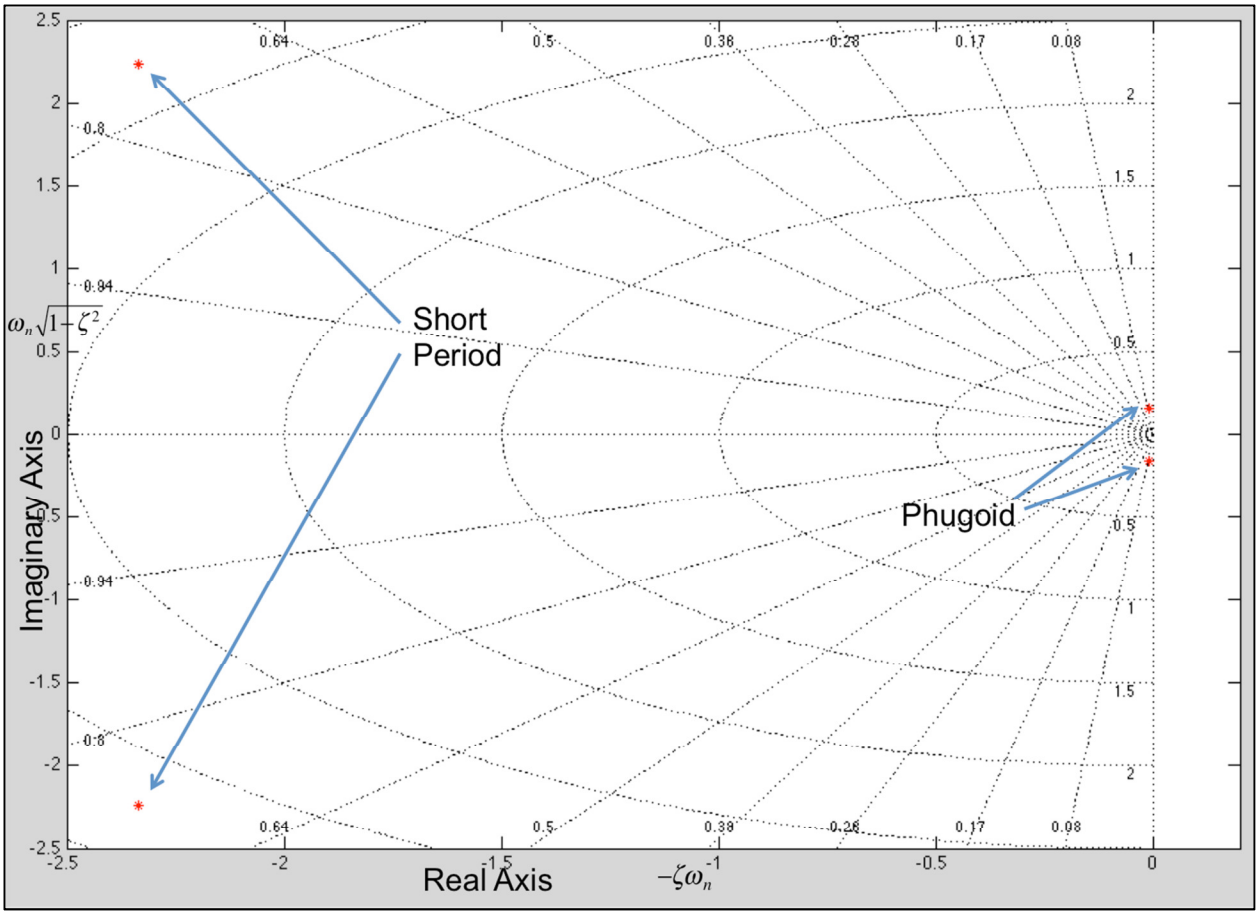


Figure 7.1 Short Period and Phugoid in Complex Plane

Rather than using the approximation equations to calculate Twin Otter's short period and phugoid's frequencies and damping ratios, a more accurate and easier method is to use the MATLAB damp() function to find the short period and phugoid's damping ratios $\zeta$ and natural frequencies $\omega_n$. The damp() function requires taking in the aircraft's plant matrix $A$ to calculate $\zeta$ and $\omega_n$. The Twin Otter phugoid and short period calculated by the damp() function is shown in Table 7.1.

Table 7.1 Twin Otter Phugoid and Short Period Characteristics

| | Eigenvalue | Damping Ratio $\zeta$ | Natural Frequency $\omega_n$ (rad/s) | Undamped Period (s) |
|---|---|---|---|---|
| Short Period | -2.33+2.23i | 0.722 | 3.23 | 1.95 |
| | -2.33-2.23i | | | |
| Phugoid | -0.00790+0.157i | 0.0502 | 0.157 | 40.0 |
| | -0.00790-0.157i | | | |

Step 2. Find Feedback Gains to Place Poles. Again, the state-space representation of the longitudinal equations of motion will be used. The basic Navion's plant matrix *A* was previously calculated in equation (5.3), the control matrix *B*, however, will be modified such that only the elevator can be used to control the aircraft. The *B* matrix now becomes:

$$B = \begin{bmatrix} 0 & 0 & 0 \\ \frac{Z_{\delta_e}}{u_0} & 0 & 0 \\ M_{\delta_e} + M_{\dot{\alpha}} \frac{Z_{\delta_e}}{u_0} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ -0.160 & 0 & 0 \\ -11.7 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{7.3}$$

Using MATLAB's place() command, it calculates the elevator feedback gains that can force the Navion to simulate different short period oscillations and phugoid. Attention must be given when using MATLAB to calculate feedback gains because MATLAB assumes negative feedback. The feedback gains matrix *K* is:

$$K = \begin{bmatrix} \frac{\partial \delta_e}{\partial u} & \frac{\partial \delta_e}{\partial \alpha} & \frac{\partial \delta_e}{\partial q} & \frac{\partial \delta_e}{\partial \theta} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0.0003 & -0.1446 & -0.0271 & 0.0080 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{7.4}$$

Step 3. Apply Calibration to Find Potentiometer Setting. These individual elevator feedback gain values can be converted into potentiometer settings as previously shown. The following table summarizes the elevator potentiometer settings required to simulate different phugoid and short period characteristics.

Table 7.2 Longitudinal Phugoid and Short Period Pole Placement Summary

| Description | $\omega_{n,sp}$ (rad/s) | $\zeta_{sp}$ | $T_p$ (s) | $\zeta_p$ | $M_u$ Pot | $M_\alpha$ Pot | $M_q$ Pot | $M_\theta$ Pot |
|---|---|---|---|---|---|---|---|---|
| Med Freq Med Damp | 4 | 0.5 | 75 | 0.05 | -26 | -38 | 5 | 1 |
| Med Freq Low Damp | 4 | 0.2 | | | -36 | -71 | 18 | 1 |
| Med Freq Zero Damp | 4 | 0 | | | -43 | -92 | 27 | 1 |
| Med Freq High Damp | 4 | 0.7 | | | -46 | -119 | -16 | 2 |
| Fast High Damp | 6 | 0.7 | | | -41 | -103 | -21 | 3 |
| Very Fast High Damp | 8 | 0.7 | | | -89 | -277 | -30 | 4 |
| Slow High Damp | 2 | 0.7 | | | -8 | 30 | 10 | 1 |
| Twin Otter | 3.23 | 0.722 | 40 | 0.0502 | -6 | 10 | 1 | 1 |

## 7.2 Simulink Results

The pole placement method is intended to match oscillation frequencies and damping ratios, not individual equations of motion. The pole placement method produces excellent results for simulating specific phugoid and short period characteristics, yet reduces the complexity by using only feedbacks to the elevator. The Simulink time histories for matching Twin Otter are shown in Figure 7.2.

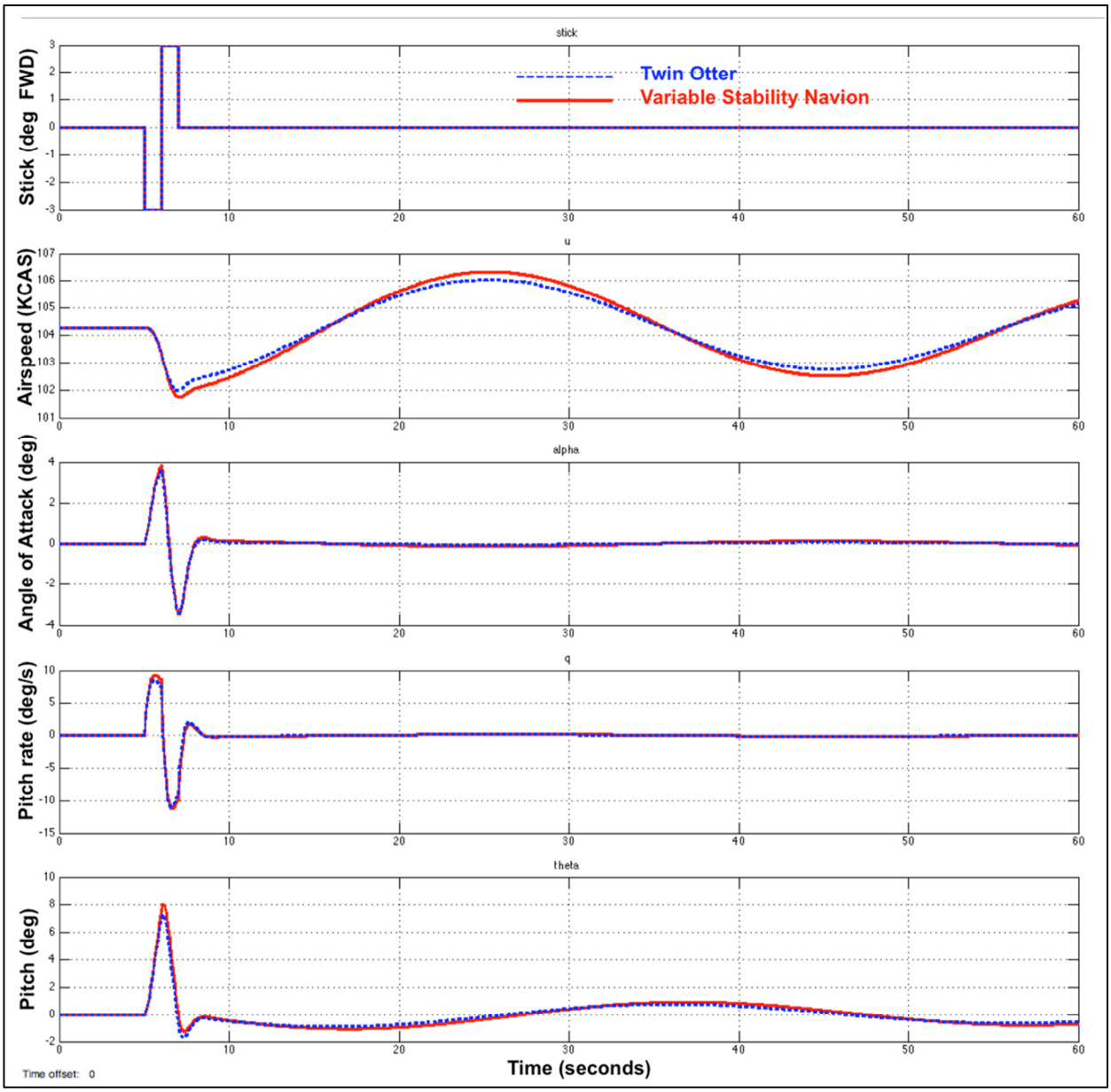


Figure 7.2 Matching Twin Otter Phugoid and Short Period Using Pole Placement Method

The training syllabus demonstration of wide range of frequencies and damping ratio also produce good results. Figure 7.3 shows the simulation of zero damping of short period mode. Once the motion is triggered, the oscillations do not dampen out.

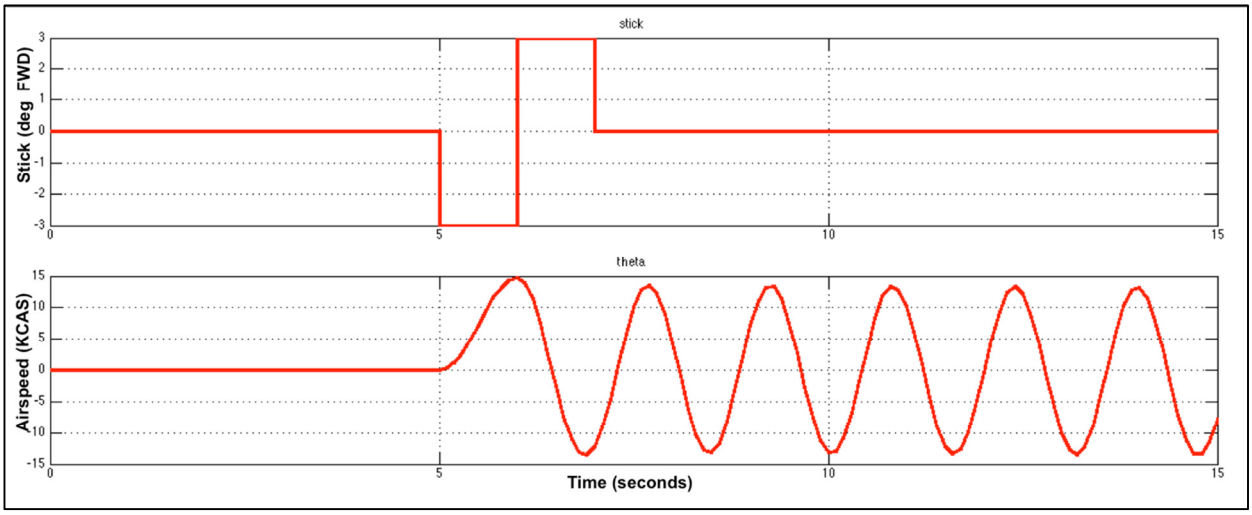


Figure 7.3 Short Period with Zero Damping using Pole Placement Method

# CHAPTER 8
# LATERAL-DIRECTIONAL EQUATIONS OF MOTION LINEARIZATION

## 8.1 Linearization

Similar to the longitudinal equations of motion, the lateral-directional set can also be linearized. The initial condition is set to be trimmed constant airspeed, straight and level flight. Therefore, the initial conditions are:

$$v_0 = q_0 = r_0 = p_0 = \phi_0 = \beta_0 = 0 \tag{8.1}$$

## 8.2 Side-Force (Y) Equation Linearization

The Y force equation (2.8) can first be rearranged into this form:

$$\dot{v} = -ru + pw + g\cos\theta\sin\phi + \frac{Y}{m} \tag{8.2}$$

Each variable is then replaced by its trimmed value and perturbation variable:

$$\begin{aligned}(\dot{v}_0 + \Delta v) = &-(r_0 + \Delta r)(u_0 + \Delta u) + (p_0 + \Delta p)(w_0 + \Delta w) \\ &+ g\cos(\theta_0 + \Delta\theta)\sin(\phi_0 + \Delta\phi) + \frac{(Y_0 + \Delta Y)}{m}\end{aligned} \tag{8.3}$$

Multiplying all the terms, and substituting in trigonometry identity and small angle approximation:

$$\begin{aligned}\dot{v}_0 + \Delta v = &-r_0u_0 - r_0\Delta u - \Delta r u_0 - \Delta r\Delta u + p_0w_0 + p_0\Delta w + \Delta p w_0 + \Delta p\Delta w \\ &+ g\cos\theta_0\sin\phi_0 + g\Delta\phi\cos\theta_0\cos\phi_0 - g\Delta\theta\sin\theta_0\sin\phi_0 \\ &- g\Delta\theta\Delta\phi\sin\theta_0\cos\phi_0 + \frac{Y_0}{m} + \frac{\Delta Y}{m}\end{aligned} \tag{8.4}$$

At equilibrium, or steady state, all the perturbation variables $\Delta$ become zero, the equation becomes:

$$\dot{v}_0 = -r_0u_0 + p_0w_0 + g\cos\theta_0\sin\phi_0 + \frac{Y_0}{m} \tag{8.5}$$

Therefore this expression can be removed from equation (8.4):

$$\Delta v = -\underbrace{r_0 \Delta u}_{=0} - \underbrace{\Delta r \Delta u}_{=0} - \Delta r u_0 + \underbrace{p_0 \Delta w}_{=0} + \Delta p w_0 + \underbrace{\Delta p \Delta w}_{=0} + g \Delta\phi \cos\theta_0 \underbrace{\cos\phi_0}_{=1}$$
$$-g \Delta\theta \sin\theta_0 \underbrace{\sin\phi_0}_{=0} - g \underbrace{\Delta\theta \Delta\phi}_{=0} \sin\theta_0 \cos\phi_0 + \frac{\Delta Y}{m} \tag{8.6}$$

Finally, apply the initial conditions and remove products of perturbation, the Y force equation of motion is linearized:

$$\Delta\dot{v} = \frac{\Delta Y}{m} + w_0 \Delta p - u_0 \Delta r + g \cos\theta_0 \Delta\phi \tag{8.7}$$

## 8.3 Rolling Moment (L) Equation Linearization

The rolling moment equation (2.8) can first be rearranged into this form:

$$\dot{p} - \frac{I_{xz}}{I_{xx}} \dot{r} = \frac{L}{I_{xx}} \tag{8.8}$$

Each variable is then replaced by its trimmed value and perturbation variable:

$$\left(\dot{p}_0 + \Delta\dot{p}\right) - \frac{I_{xz}}{I_{xx}} \left(\dot{r}_0 + \Delta r\right) = \frac{\left(L_0 + \Delta L\right)}{I_{xx}} \tag{8.9}$$

Multiplying all the terms yields:

$$\dot{p}_0 + \Delta\dot{p} - \frac{I_{xz}}{I_{xx}} \dot{r}_0 - \frac{I_{xz}}{I_{xx}} \Delta r = \frac{L_0}{I_{xx}} + \frac{\Delta L}{I_{xx}} \tag{8.10}$$

At equilibrium, or steady state, all the perturbation variables *Δ* become zero, the equation becomes:

$$\dot{p}_0 - \frac{I_{xz}}{I_{xx}} \dot{r}_0 = \frac{L_0}{I_{xx}} \tag{8.11}$$

Therefore this expression can be removed from equation (8.10), the rolling moment equation of motion is linearized:

$$\Delta\dot{p} - \frac{I_{xz}}{I_{xx}} \Delta\dot{r} = \frac{\Delta L}{I_{xx}} \tag{8.12}$$

## 8.4 Yawing Moment (N) Equation Linearization

The rolling moment equation (2.8) can first be rearranged into this form:

$$\dot{r} - \frac{I_{xz}}{I_{zz}}\dot{p} = \frac{N}{I_{zz}} \tag{8.13}$$

Each variable is then replaced by its trimmed value and perturbation variable:

$$\left(\dot{r}_0 + \Delta r\right) - \frac{I_{xz}}{I_{zz}}\left(\dot{p}_0 + \Delta\dot{p}\right) = \frac{\left(N_0 + \Delta N\right)}{I_{zz}} \tag{8.14}$$

Multiplying all the terms yields:

$$\dot{r}_0 + \Delta r - \frac{I_{xz}}{I_{zz}}\dot{p}_0 - \frac{I_{xz}}{I_{zz}}\Delta\dot{p} = \frac{N_0}{I_{zz}} + \frac{\Delta N}{I_{zz}} \tag{8.15}$$

At equilibrium, or steady state, all the perturbation variables Δ become zero, the equation becomes:

$$\dot{r}_0 - \frac{I_{xz}}{I_{zz}}\dot{p}_0 = \frac{N_0}{I_{zz}} \tag{8.16}$$

Therefore this expression can be removed from equation(8.15), the yawing moment equation of motion is linearized:

$$\Delta\dot{r} - \frac{I_{xz}}{I_{zz}}\Delta\dot{p} = \frac{\Delta N}{I_{zz}} \tag{8.17}$$

In Summary, the lateral-directional equations of motion can now be rewritten as linear equations (8.18):

$$\Delta\dot{v} = \frac{\Delta Y}{m} + w_0\Delta p - u_0\Delta r + g\cos\theta_0\Delta\phi$$

$$\Delta\dot{p} - \frac{I_{xz}}{I_{xx}}\Delta\dot{r} = \frac{\Delta L}{I_{xx}} \tag{8.18}$$

$$\Delta\dot{r} - \frac{I_{xz}}{I_{zz}}\Delta\dot{p} = \frac{\Delta N}{I_{zz}}$$

The lateral-directional linearized equations of motion require more algebraic manipulation to isolate $\Delta\dot{p}$ and $\Delta\dot{r}$. After some arrangement of the equations above, the following equations are obtained:

$$\Delta\dot{v} = \frac{\Delta Y}{m} + w_0\Delta p - u_0\Delta r + g\cos\theta_0\Delta\phi$$

$$\Delta\dot{p} = \left(\frac{\Delta L}{I_{xx}} + \frac{\Delta N}{I_{zz}}\frac{I_{xz}}{I_{xx}}\right)\left(\frac{1}{1 - I_{xz}^2/I_{xx}I_{zz}}\right) \tag{8.19}$$

$$\Delta\dot{r} = \left(\frac{\Delta N}{I_{zz}} + \frac{\Delta L}{I_{xx}}\frac{I_{xz}}{I_{zz}}\right)\left(\frac{1}{1 - I_{xz}^2/I_{xx}I_{zz}}\right)$$

## 8.5 First-Order Taylor Approximation of Forces and Moments

The perturbed force and moment variables in the linearized lateral-directional equations of motion (8.19) can be approximated by using first-order Taylor series:

$$\Delta Y = \frac{\partial Y}{\partial v}\Delta v + \frac{\partial Y}{\partial p}\Delta p + \frac{\partial Y}{\partial r}\Delta r + \frac{\partial Y}{\partial \delta_a}\Delta\delta_a + \frac{\partial Y}{\partial \delta_r}\Delta\delta_r$$

$$\Delta L = \frac{\partial L}{\partial v}\Delta v + \frac{\partial L}{\partial p}\Delta p + \frac{\partial L}{\partial r}\Delta r + \frac{\partial L}{\partial \delta_a}\Delta\delta_a + \frac{\partial L}{\partial \delta_r}\Delta\delta_r \tag{8.20}$$

$$\Delta N = \frac{\partial N}{\partial v}\Delta v + \frac{\partial N}{\partial p}\Delta p + \frac{\partial N}{\partial r}\Delta r + \frac{\partial N}{\partial \delta_a}\Delta\delta_a + \frac{\partial N}{\partial \delta_r}\Delta\delta_r$$

It is convenient to use sideslip angle $\Delta\beta$ instead of the lateral velocity $\Delta v$. They are related to each other by the small angle approximation theory:

$$\Delta\beta \cong \tan^{-1}\frac{\Delta v}{u_0} = \frac{\Delta v}{u_0} \tag{8.21}$$

Substituting the Taylor series representation of forces and moment into the small perturbation lateral-directional linearized equations of motion (8.19) yields:

$$\Delta\dot{v} = \frac{\left(\frac{\partial Y}{\partial v}\right)}{m}\Delta v + \left[\frac{\left(\frac{\partial Y}{\partial p}\right)}{m} + w_0\right]\Delta p + \left[\frac{\left(\frac{\partial Y}{\partial r}\right)}{m} - u_0\right]\Delta r$$
$$+ g\cos\theta_0\Delta\phi + \frac{\left(\frac{\partial Y}{\partial \delta_a}\right)}{m}\Delta\delta_a + \frac{\left(\frac{\partial Y}{\partial \delta_r}\right)}{m}\Delta\delta_r$$

$$\Delta\dot{p} = \left\{\begin{array}{l} \left[\frac{\left(\frac{\partial L}{\partial v}\right)}{I_{xx}} + \frac{\left(\frac{\partial N}{\partial v}\right)}{I_{zz}}\frac{I_{xz}}{I_{xx}}\right]\Delta v \\ + \left[\frac{\left(\frac{\partial L}{\partial p}\right)}{I_{xx}} + \frac{\left(\frac{\partial N}{\partial p}\right)}{I_{zz}}\frac{I_{xz}}{I_{xx}}\right]\Delta p + \left[\frac{\left(\frac{\partial L}{\partial r}\right)}{I_{xx}} + \frac{\left(\frac{\partial N}{\partial r}\right)}{I_{zz}}\frac{I_{xz}}{I_{xx}}\right]\Delta r \\ + \left[\frac{\left(\frac{\partial L}{\partial \delta_a}\right)}{I_{xx}} + \frac{\left(\frac{\partial N}{\partial \delta_a}\right)}{I_{zz}}\frac{I_{xz}}{I_{xx}}\right]\Delta\delta_a + \left[\frac{\left(\frac{\partial L}{\partial \delta_r}\right)}{I_{xx}} + \frac{\left(\frac{\partial N}{\partial \delta_r}\right)}{I_{zz}}\frac{I_{xz}}{I_{xx}}\right]\Delta\delta_r \end{array}\right\}\left(\frac{1}{1 - I_{xz}^2/I_{xx}I_{zz}}\right)$$

$$\Delta\dot{r} = \left\{\begin{array}{l} \left[\frac{\left(\frac{\partial N}{\partial v}\right)}{I_{zz}} + \frac{\left(\frac{\partial L}{\partial v}\right)}{I_{xx}}\frac{I_{xz}}{I_{zz}}\right]\Delta v \\ + \left[\frac{\left(\frac{\partial N}{\partial p}\right)}{I_{zz}} + \frac{\left(\frac{\partial L}{\partial p}\right)}{I_{xx}}\frac{I_{xz}}{I_{zz}}\right]\Delta p + \left[\frac{\left(\frac{\partial N}{\partial r}\right)}{I_{zz}} + \frac{\left(\frac{\partial L}{\partial r}\right)}{I_{xx}}\frac{I_{xz}}{I_{zz}}\right]\Delta r \\ + \left[\frac{\left(\frac{\partial N}{\partial \delta_a}\right)}{I_{zz}} + \frac{\left(\frac{\partial L}{\partial \delta_a}\right)}{I_{xx}}\frac{I_{xz}}{I_{zz}}\right]\Delta\delta_a + \left[\frac{\left(\frac{\partial N}{\partial \delta_r}\right)}{I_{zz}} + \frac{\left(\frac{\partial L}{\partial \delta_r}\right)}{I_{xx}}\frac{I_{xz}}{I_{zz}}\right]\Delta\delta_r \end{array}\right\}\left(\frac{1}{1 - I_{xz}^2/I_{xx}I_{zz}}\right) \quad (8.22)$$

### 8.5.1 Dimensional Stability Derivatives

The cumbersome equation above can be rewritten in a more concise form by using dimensional stability derivatives (Nelson, 2007, p.123).

Table 8.1 Summary of Lateral-Directional Stability Derivatives

| | |
|---|---|
| $Y_\beta = \frac{QSC_{y_\beta}}{m}$ | $Y_v = \frac{\left(\frac{\partial Y}{\partial v}\right)}{m} = \frac{Y_\beta}{u_0}$ |
| $Y_p = \frac{\left(\frac{\partial Y}{\partial p}\right)}{m} = \frac{QSbC_{y_p}}{2mu_0}$ | $Y_r = \frac{\left(\frac{\partial Y}{\partial r}\right)}{m} = \frac{QSbC_{y_r}}{2mu_0}$ |
| $Y_{\delta_a} = \frac{\left(\frac{\partial Y}{\partial \delta_a}\right)}{m} = \frac{QSC_{y_{\delta_a}}}{m}$ | $Y_{\delta_r} = \frac{\left(\frac{\partial Y}{\partial \delta_r}\right)}{m} = \frac{QSC_{y_{\delta_r}}}{m}$ |
| $L_\beta = \frac{QSbC_{l_\beta}}{I_{xx}}$ | $L_v = \frac{\left(\frac{\partial L}{\partial v}\right)}{I_{xx}} = \frac{L_\beta}{u_0}$ |
| $L_p = \frac{\left(\frac{\partial L}{\partial p}\right)}{I_{xx}} = \frac{QSb^2C_{l_p}}{2I_{xx}u_0}$ | $L_r = \frac{\left(\frac{\partial L}{\partial r}\right)}{I_{xx}} = \frac{QSb^2C_{l_r}}{2I_{xx}u_0}$ |
| $L_{\delta_a} = \frac{\left(\frac{\partial L}{\partial \delta_a}\right)}{I_{xx}} = \frac{QSbC_{l_{\delta_a}}}{I_{xx}}$ | $L_{\delta_r} = \frac{\left(\frac{\partial L}{\partial \delta_r}\right)}{I_{xx}} = \frac{QSbC_{l_{\delta_r}}}{I_{xx}}$ |
| $N_\beta = \frac{QSbC_{n_\beta}}{I_{zz}}$ | $N_v = \frac{\left(\frac{\partial N}{\partial v}\right)}{I_{zz}} = \frac{N_\beta}{u_0}$ |
| $N_p = \frac{\left(\frac{\partial N}{\partial p}\right)}{I_{zz}} = \frac{QSb^2C_{n_p}}{2I_{zz}u_0}$ | $N_r = \frac{\left(\frac{\partial N}{\partial r}\right)}{I_{zz}} = \frac{QSb^2C_{n_r}}{2I_{zz}u_0}$ |
| $N_{\delta_a} = \frac{\left(\frac{\partial N}{\partial \delta_a}\right)}{I_{zz}} = \frac{QSbC_{n_{\delta_a}}}{I_{zz}}$ | $N_{\delta_r} = \frac{\left(\frac{\partial N}{\partial \delta_r}\right)}{I_{zz}} = \frac{QSbC_{n_{\delta_r}}}{I_{zz}}$ |

(Nelson, 2007, p.123; Yechout, 2003, p.292)

## 8.6 Basic Navion and Twin Otter Lateral-Directional Dimensional Derivatives

The basic Navion geometric, mass, and aerodynamic characteristics data are for trimmed airspeed of 104 KCAS (M 0.158) at sea level (Nelson, 2007, p.400). The following table summarizes the basic Navion lateral-directional dimensional derivatives.

Table 8.2 Basic Navion Lateral-Directional Derivatives

| | |
|---|---|
| $Y_\beta = -44.7 \text{ ft/s}^{-2}$ | $Y_v = -0.254 \text{ s}^{-1}$ |
| $Y_p = 0 \text{ ft/s}$ | $Y_r = 0 \text{ ft/s}$ |
| $Y_{\delta_a} = 0 \text{ ft/s}^2$ | $Y_{\delta_r} = 12.5 \text{ ft/s}^2$ |
| $L_\beta = -15.0 \text{ s}^{-2}$ | $L_v = -0.909 \text{ s}^{-1}$ |
| $L_p = -8.41 \text{ s}^{-1}$ | $L_r = 2.19 \text{ s}^{-1}$ |
| $L_{\delta_a} = -29.0 \text{ s}^{-2}$ | $L_{\delta_r} = 23.1 \text{ s}^{-2}$ |
| $N_\beta = 4.56 \text{ s}^{-2}$ | $N_v = 0.0259 \text{ ft}^{-1}\text{s}^{-1}$ |
| $N_p = -0.350 \text{ s}^{-1}$ | $N_r = -0.761 \text{ s}^{-1}$ |
| $N_{\delta_a} = -0.225 \text{ s}^{-2}$ | $N_{\delta_r} = -4.62 \text{ s}^{-2}$ |

The Twin Otter geometric, mass, and aerodynamic characteristics data are for trimmed airspeed of 104 KCAS (M 0.158) at sea level (Brigg et al., 2000). The following table summarizes the clean configuration Twin Otter lateral-dimensional dimensional derivatives.

Table 8.3 Twin Otter Lateral-Directional Derivatives

| | |
|---|---|
| $Y_\beta = -32.9 \text{ ft/s}^{-2}$ | $Y_v = -0.187 \text{ s}^{-1}$ |
| $Y_p = -2.02 \text{ ft/s}$ | $Y_r = 4.05 \text{ ft/s}$ |
| $Y_{\delta_a} = 0 \text{ ft/s}^2$ | $Y_{\delta_r} = 8.22 \text{ ft/s}^2$ |
| $L_\beta = -5.16 \text{ s}^{-2}$ | $L_v = -0.0293 \text{ s}^{-1}$ |
| $L_p = -5.96 \text{ s}^{-1}$ | $L_r = 0.715 \text{ s}^{-1}$ |
| $L_{\delta_a} = -9.68 \text{ s}^{-2}$ | $L_{\delta_r} = 0.968 \text{ s}^{-2}$ |
| $N_\beta = 3.05 \text{ s}^{-2}$ | $N_v = 0.0173 \text{ ft}^{-1}\text{s}^{-1}$ |
| $N_p = -0.338 \text{ s}^{-1}$ | $N_r = -1.01 \text{ s}^{-1}$ |
| $N_{\delta_a} = -0.0305 \text{ s}^{-2}$ | $N_{\delta_r} = -3.66 \text{ s}^{-2}$ |

## 8.7 Linearized Equations of Motion with Dimensional Stability Derivatives

Substituting the dimensional stability derivatives from Table 8.1 into the small perturbation linearized lateral-direction equations of motion (8.22) yields:

$$\Delta\dot{v} = Y_v\Delta v + \left(Y_p + w_0\right)\Delta p + \left(Y_r - u_0\right)\Delta r + g\cos\theta_0\Delta\phi + Y_{\delta_a}\Delta\delta_a + Y_{\delta_r}\Delta\delta_r$$

$$\Delta\dot{p} = \begin{bmatrix} \left(L_v + N_v\dfrac{I_{xz}}{I_{xx}}\right)\Delta v \\ +\left(L_p + N_p\dfrac{I_{xz}}{I_{xx}}\right)\Delta p + \left(L_r + N_r\dfrac{I_{xz}}{I_{xx}}\right)\Delta r \\ +\left(L_{\delta_a} + N_{\delta_a}\dfrac{I_{xz}}{I_{xx}}\right)\Delta\delta_a + \left(L_{\delta_r} + N_{\delta_r}\dfrac{I_{xz}}{I_{xx}}\right)\Delta\delta_r \end{bmatrix}\left(\frac{1}{1 - I_{xz}^2/I_{xx}I_{zz}}\right)$$

$$\Delta\dot{r} = \begin{bmatrix} \left(N_v + L_v\dfrac{I_{xz}}{I_{zz}}\right)\Delta v \\ +\left(N_p + L_p\dfrac{I_{xz}}{I_{zz}}\right)\Delta p + \left(N_r + L_r\dfrac{I_{xz}}{I_{zz}}\right)\Delta r \\ +\left(N_{\delta_a} + L_{\delta_a}\dfrac{I_{xz}}{I_{zz}}\right)\Delta\delta_a + \left(N_{\delta_r} + L_{\delta_r}\dfrac{I_{xz}}{I_{zz}}\right)\Delta\delta_r \end{bmatrix}\left(\frac{1}{1 - I_{xz}^2/I_{xx}I_{zz}}\right) \qquad (8.23)$$

Finally, equation (8.23) can be rearranged into a more usable format for the Variable Stability Navion. The change converts the $\Delta\dot{v}$ equation into $\Delta\dot{\beta}$ equation by dividing the entire equation by $u_0$ as per equation (8.21). This change facilitates calculations because the Navion uses a sideslip angle vane as feedback, rather than a lateral velocity sensor. The resulting equation set becomes:

$$\Delta\dot{\beta} = \frac{Y_\beta}{u_0}\Delta\beta + \left(\frac{Y_p}{u_0} + \frac{w_0}{u_0}\right)\Delta p + \left(\frac{Y_r}{u_0} - 1\right)\Delta r + \frac{g}{u_0}\cos\theta_0\Delta\phi + \frac{Y_{\delta_a}}{u_0}\Delta\delta_a + \frac{Y_{\delta_r}}{u_0}\Delta\delta_r$$

$$\Delta\dot{p} = \begin{bmatrix} \left(L_\beta + N_\beta\frac{I_{xz}}{I_{xx}}\right)\Delta\beta \\ +\left(L_p + N_p\frac{I_{xz}}{I_{xx}}\right)\Delta p + \left(L_r + N_r\frac{I_{xz}}{I_{xx}}\right)\Delta r \\ +\left(L_{\delta_a} + N_{\delta_a}\frac{I_{xz}}{I_{xx}}\right)\Delta\delta_a + \left(L_{\delta_r} + N_{\delta_r}\frac{I_{xz}}{I_{xx}}\right)\Delta\delta_r \end{bmatrix}\left(\frac{1}{1 - I_{xz}^2/I_{xx}I_{zz}}\right)$$

$$\Delta\dot{r} = \begin{bmatrix} \left(N_\beta + L_\beta\frac{I_{xz}}{I_{zz}}\right)\Delta\beta \\ +\left(N_p + L_p\frac{I_{xz}}{I_{zz}}\right)\Delta p + \left(N_r + L_r\frac{I_{xz}}{I_{zz}}\right)\Delta r \\ +\left(N_{\delta_a} + L_{\delta_a}\frac{I_{xz}}{I_{zz}}\right)\Delta\delta_a + \left(N_{\delta_r} + L_{\delta_r}\frac{I_{xz}}{I_{zz}}\right)\Delta\delta_r \end{bmatrix}\left(\frac{1}{1 - I_{xz}^2/I_{xx}I_{zz}}\right) \quad (8.24)$$

## 8.8 Simulink Simulation

The linearized 3-DOF lateral-directional model uses rudder doublet input to disturb the aircraft flying straight and level with constant trimmed airspeed. The time histories of the Twin Otter are then compared with the basic Navion. As expected, the two aircraft do not respond the same way to the pilot rudder doublet input as shown in Figure 8.1.

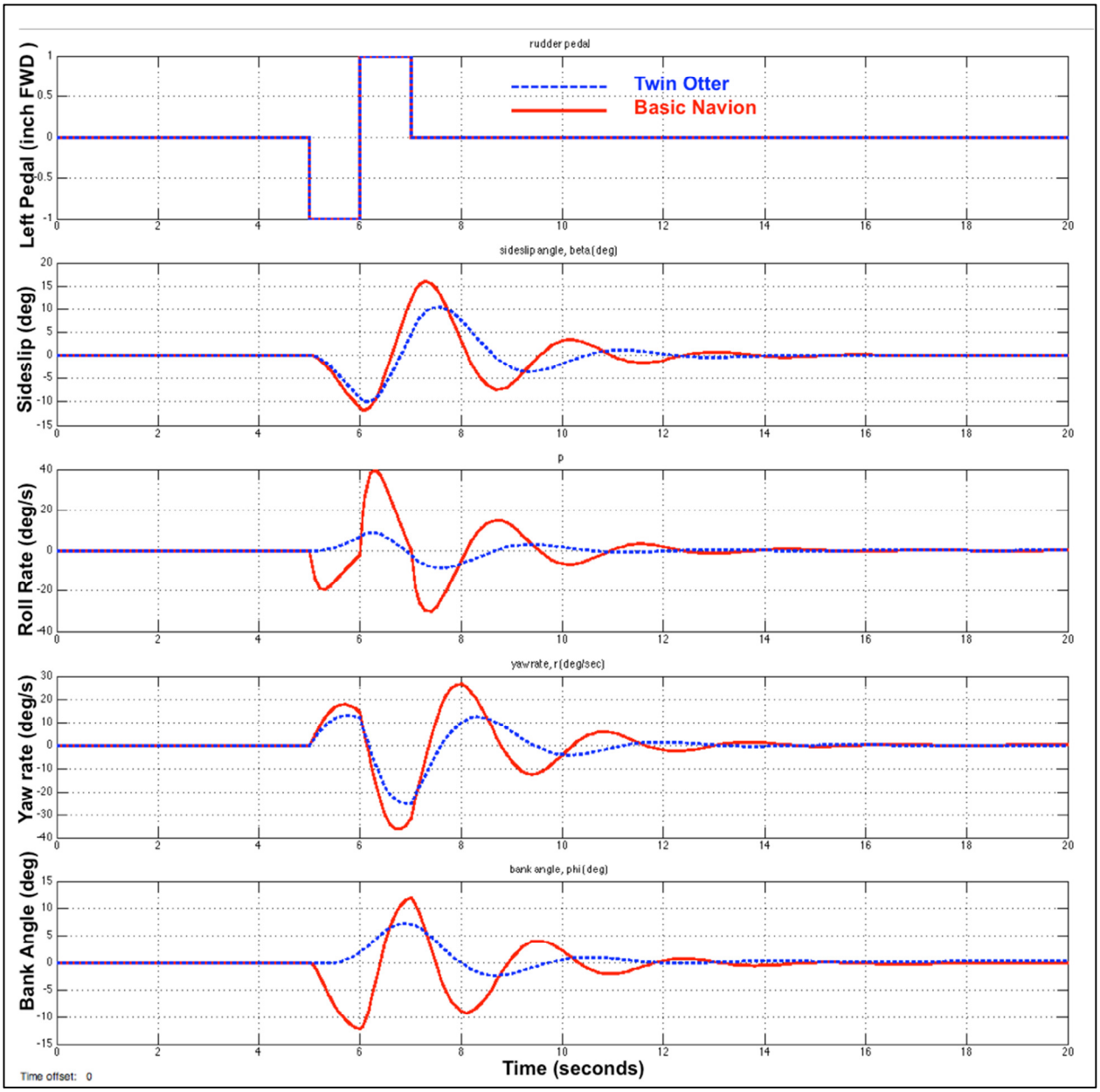


Figure 8.1 Basic Navion and Twin Otter Rudder Doublet Response

# CHAPTER 9
# LATERAL-DIRECTIONAL IMPLICIT MODEL FOLLOWING

## 9.1 Variable Stability Implementation

The Variable Stability Navion, N66UT, can only match the $\Delta\dot{p}$ and $\Delta\dot{r}$ equations of motion term-by-term. For the $\Delta\dot{p}$ equation, the ailerons primarily generate rolling moment to compensate the difference in *L* derivatives. For the $\Delta\dot{r}$ equation, the rudder primarily generates yawing moment to compensate the difference in *N* derivatives.

Unfortunately, the complete matching of individual lateral-directional dimensional derivatives is not possible due to lack of side-force generators on the Variable Stability Navion. Therefore, the $\Delta\dot{v}$ equation cannot be satisfied and the *Y* derivatives cannot be matched term-by-term. For general aviation aircraft, side-force is rarely apparent to the pilot during routine flight tasks, as such, the Variable Stability Navion's inability to match $\Delta\dot{v}$ equation directly is still suffice for in-flight simulation.

## 9.2 Term-by-Term Stability Derivative Matching Methods

Matching the $\Delta\dot{p}$ and $\Delta\dot{r}$ equations become tedious due to the $I_{xz}$ term as shown in equation (8.24). For the basic Navion, the $I_{xz}$ value is zero, but many aircraft such as the Twin Otter have a non-zero value for $I_{xz}$. For the case of Twin Otter, it is justifiable to ignore the $I_{xz}$ because $I_{xz}$ is much smaller than $I_{xx}$ and $I_{zz}$, and thus not have significant contribution in the lateral-directional time histories.

## 9.3 Independent Axis Control Assumption

The term-by-term dimensional derivative matching process is similar to the longitudinal case. An example would be to match the "weather-vane" derivative, $N_r$, which is to match the $\Delta\dot{r}$ equations by using yaw rate feedback to rudder:

$$\left(N_r\right)_{TwinOtter}\Delta r = \left(N_r\right)_{Navion}\Delta r + \left(N_{\delta_r}\right)_{Navion}\frac{\partial\delta_r}{\partial r}\Delta r \tag{9.1}$$

By matching each derivative on each axis independently, a set of feedback gains is calculated. The simulation result is poor matching in the roll axis for the Twin Otter during a rudder doublet, as shown in Figure 9.1. During an aileron double, the match result is fair, as shown in Figure 9.2.

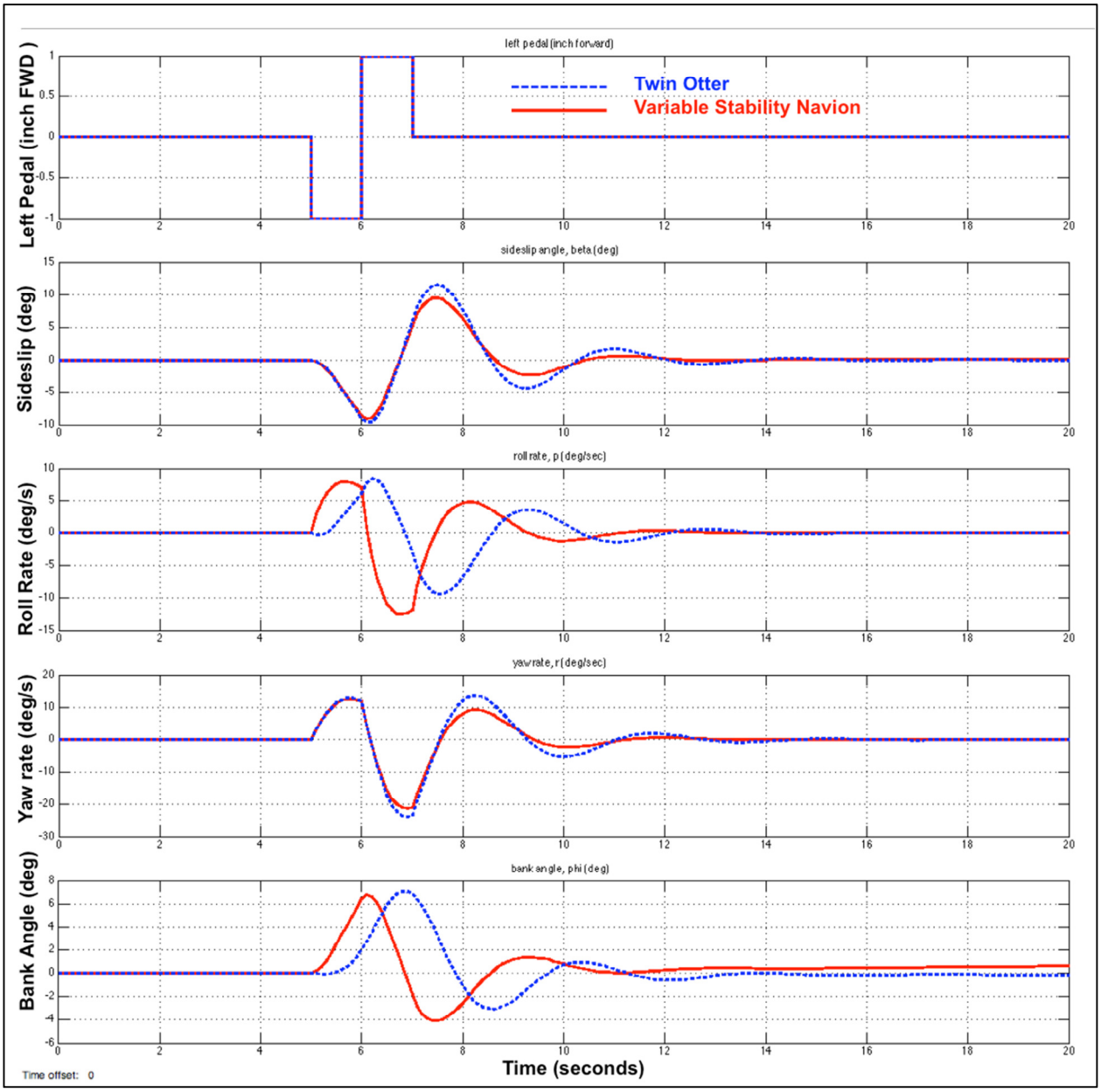


Figure 9.1 Lateral-Directional Term-by-Term Matching Rudder Doublet

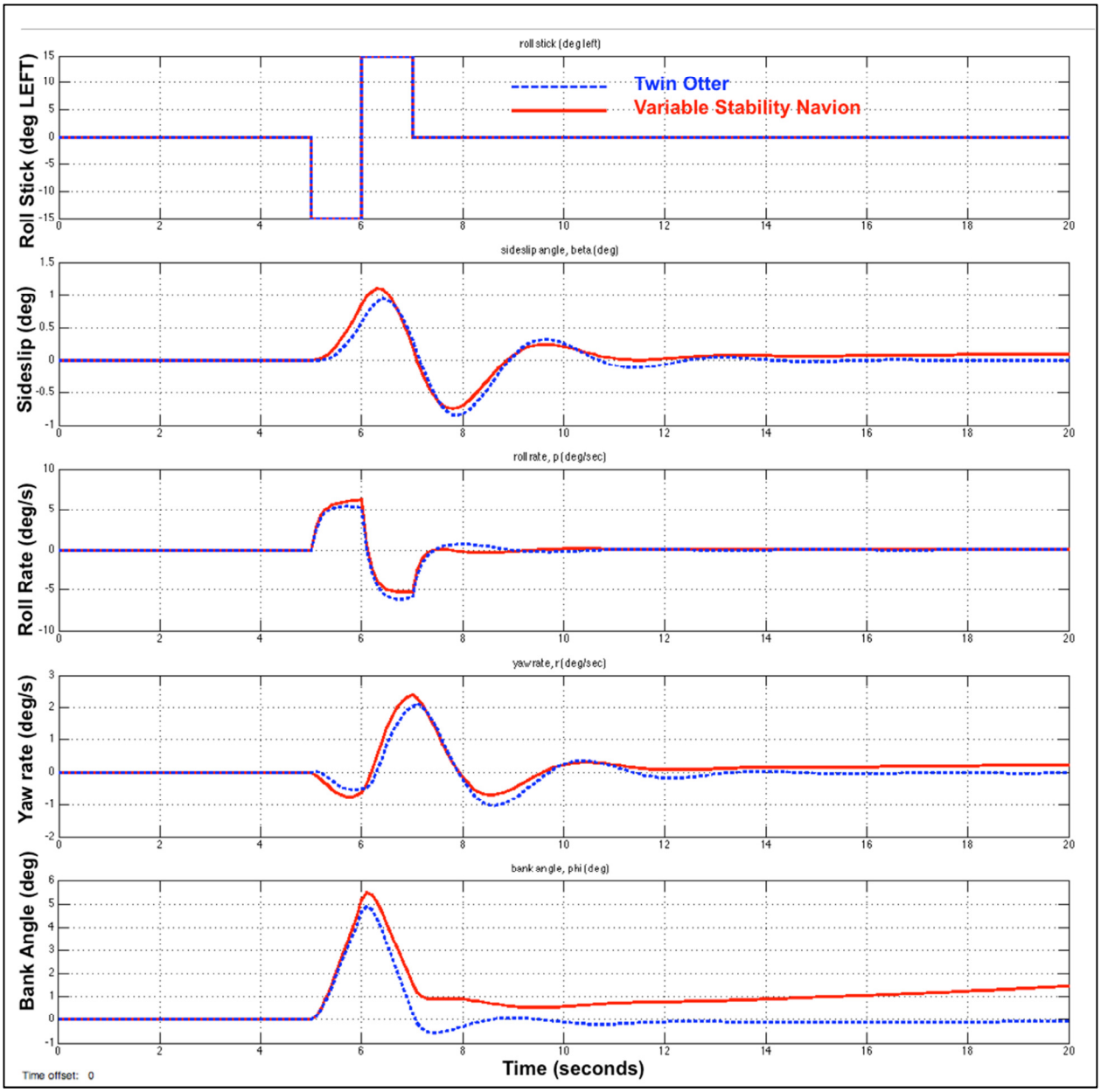


Figure 9.2 Lateral-Directional Term-by-Term Matching Aileron Doublet

The problem is due to the highly coupled nature of the roll and yaw axes. The rudder produces a yawing moment, but it also generates rolling moment. Likewise for the ailerons, their primary use is to roll the aircraft, but some yawing moment is produced. It is not possible to simply ignore the cross-coupling affects because they have significant contributions to the aircraft dynamic motions. The assumption of each axis can be controlled independently is not valid. The control cross-coupling is most obvious to pilots in the form of rudder roll and adverse yaw.

Since this method of matching derivatives on individual axis does not produce good results, a more rigorous method needs to be devised.

## 9.4 Matching Derivatives in Pairs

Each stability derivative can be matched by using both the rudder and the ailerons. This creates a situation where there are five sets of two equations with two unknowns. Stability derivatives would need to be solved simultaneously in pairs, for example, $L_r$ and $N_r$:

$$\begin{aligned}
\left(N_r\right)_{TwinOtter}\Delta r &= \left(N_r\right)_{Navion}\Delta r + \left(N_{\delta_r}\right)_{Navion}\frac{\partial \delta_r}{\partial r}\Delta r + \left(N_{\delta_a}\right)_{Navion}\frac{\partial \delta_a}{\partial r}\Delta r \\
\left(L_r\right)_{TwinOtter}\Delta r &= \left(L_r\right)_{Navion}\Delta r + \left(L_{\delta_a}\right)_{Navion}\frac{\partial \delta_a}{\partial r}\Delta r + \left(L_{\delta_r}\right)_{Navion}\frac{\partial \delta_r}{\partial r}\Delta r
\end{aligned} \tag{9.2}$$

where the variables to be solved simultaneously are $\frac{\partial \delta_r}{\partial r}$ and $\frac{\partial \delta_a}{\partial r}$.

Similarly, the other derivative pairs to be solved are: ($L_p$ and $N_p$), ($L_\beta$ and $N_\beta$), ($L_{\delta r}$ and $N_{\delta \mathrm{r}}$), and ($L_{\delta a}$ and $N_{\delta a}$). Once all the feedback gains are solved, the Simulink simulation results show good time history matching for rudder doublets (Figure 9.3). Despite the omission of $I_{xz}$, and no side-force matching, the two time histories overlap each other almost perfectly. As for aileron doublet in Figure 9.4, the matching is not exact but it is better than the term-by-term method.

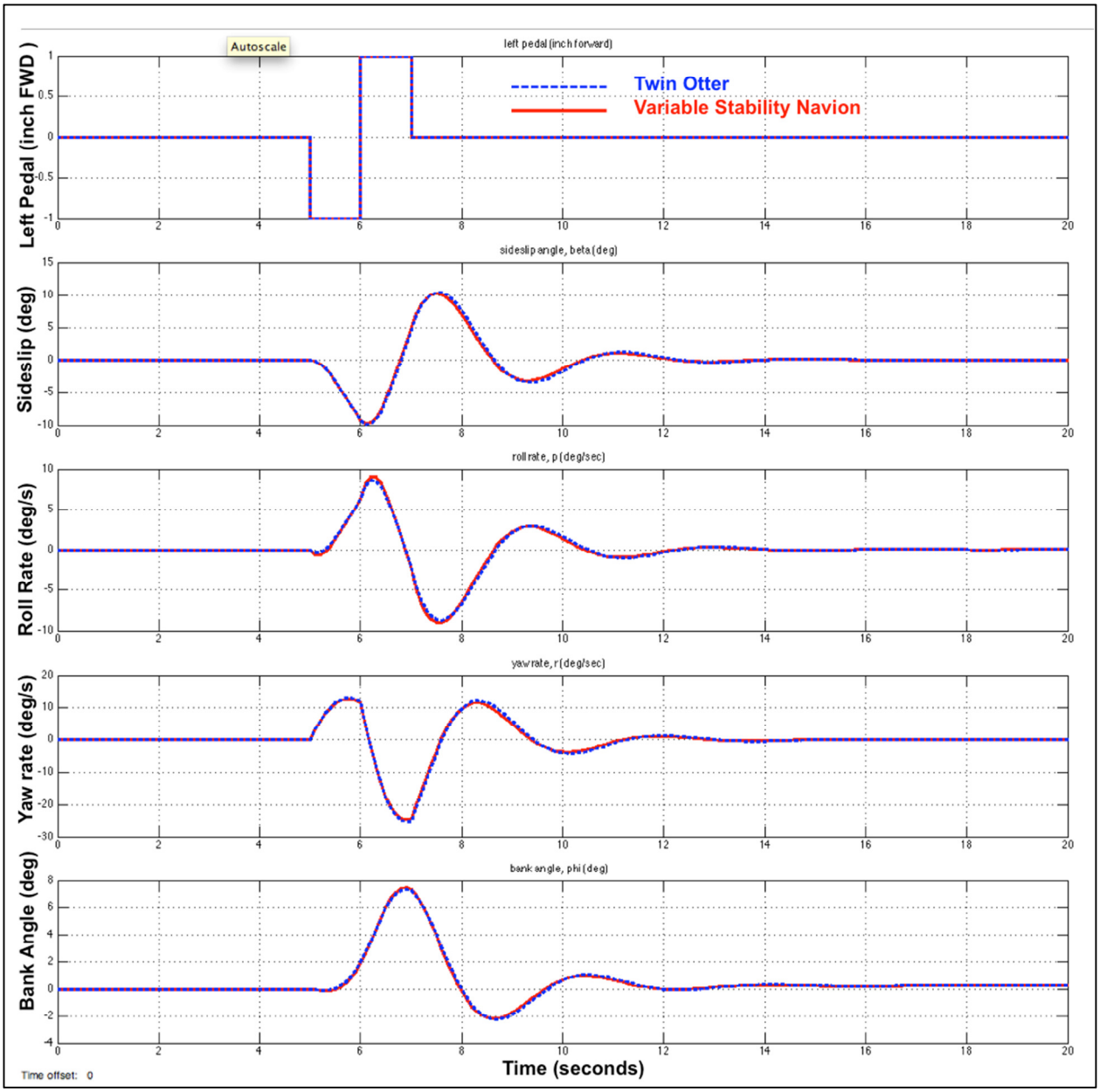


Figure 9.3 Simulating Twin Otter by Pair Matching Method Rudder Doublet Response

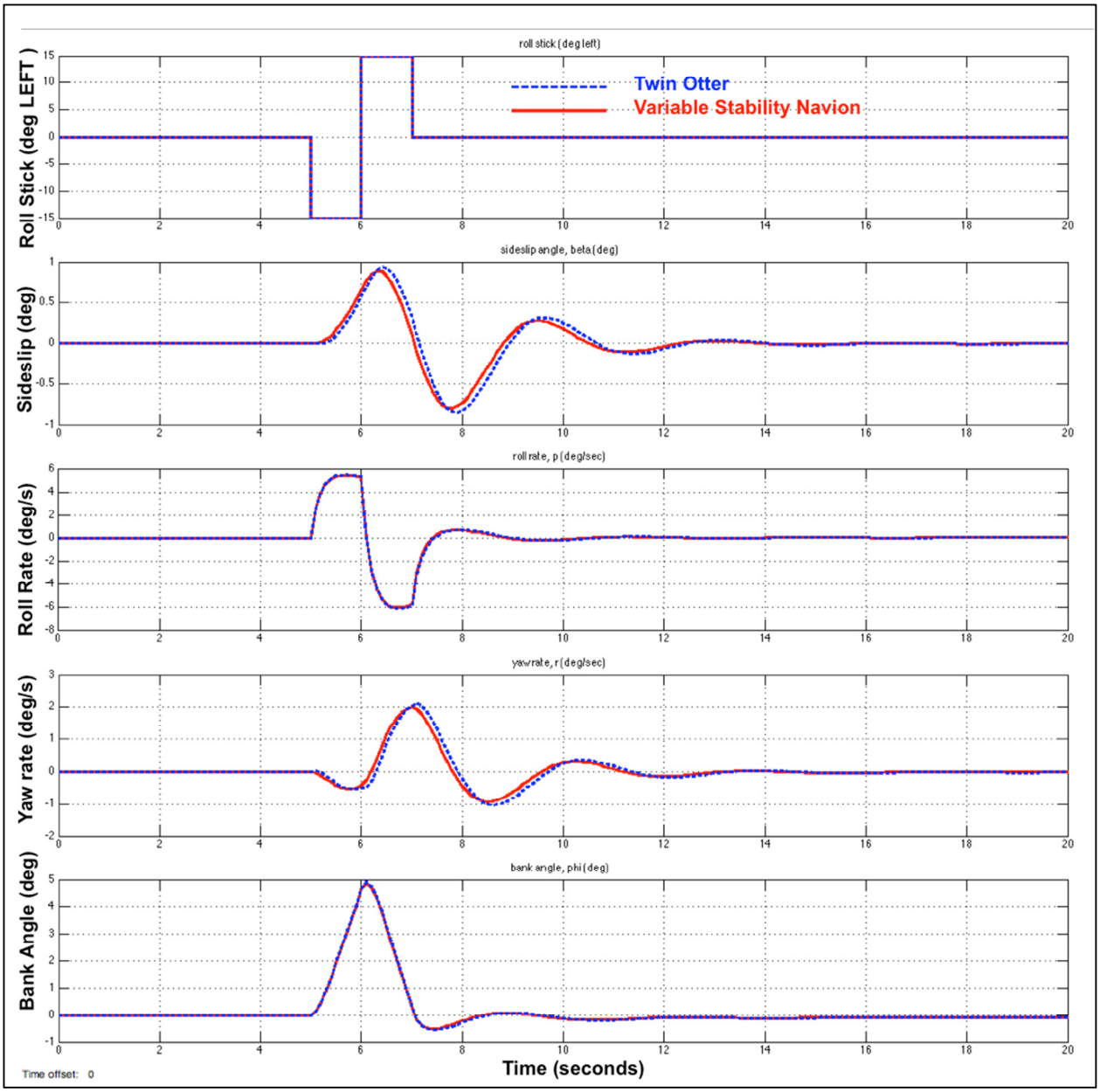


Figure 9.4 Simulating Twin Otter by Pair Matching Method Aileron Doublet Response

## 9.5 Twin Otter Lateral-Directional Setup Summary

Table 9.1 summarizes the potentiometer settings that the pilot needs to dial into the Variable Stability Navion to simulate a Twin Otter for standard atmospheric condition at trimmed calibrated airspeed of 104 KCAS, at any altitude. At the writing of this report, the lateral-directional potentiometer calibrations are not available. Once ground calibration has been conducted, the potentiometer settings can be easily calculated using the feedback gains provided below.

Table 9.1 Lateral-Directional Potentiometer Setting Summary

| Potentiometer Label | Gains | Potentiometer Settings |
|---|---|---|
| $L_\beta$ | -0.110 | Calibrations Not Available |
| $L_p$ | -0.0836 | |
| $L_r$ | 0.0913 | |
| $L_{\delta a}$ | 0.0768 | |
| $L_{\delta r}$ | 5.32 | |
| $N_\beta$ | 0.331 | |
| $N_p$ | 0.0014 | |
| $N_r$ | 0.0503 | |
| $N_{\delta a}$ | -0.0022 | |
| $N_{\delta r}$ | 7.05 | |

# CHAPTER 10
# LATERAL-DIRECTIONAL STATE-SPACE REPRESENTATION

As shown in the term-by-term matching method, multiple sets of equations needed to be solved simultaneously. This is precisely the advantage of using state-space form to allow MATLAB to solve it. The linearized lateral-directional equations of motion can be written in the state-space form. Rewriting equations (8.24) yields the following state-space representation:

$$\begin{bmatrix} \Delta\dot{\beta} \\ \Delta\dot{p} \\ \Delta\dot{r} \\ \Delta\dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{Y_\beta}{u_0} & \frac{Y_p}{u_0}+\frac{w_0}{u_0} & \frac{Y_r}{u_0}-1 & \frac{g}{u_0}\cos\theta_0 \\ \left(L_\beta + N_\beta \frac{I_{xz}}{I_{xx}}\right)\kappa & \left(L_p + N_p \frac{I_{xz}}{I_{xx}}\right)\kappa & \left(L_r + N_r \frac{I_{xz}}{I_{xx}}\right)\kappa & 0 \\ \left(N_\beta + L_\beta \frac{I_{xz}}{I_{zz}}\right)\kappa & \left(N_p + L_p \frac{I_{xz}}{I_{zz}}\right)\kappa & \left(N_r + L_r \frac{I_{xz}}{I_{zz}}\right)\kappa & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\beta \\ \Delta p \\ \Delta r \\ \Delta\phi \end{bmatrix} + \begin{bmatrix} \frac{Y_{\delta_a}}{u_0} & \frac{Y_{\delta_r}}{u_0} \\ \left(L_{\delta_a} + N_{\delta_a} \frac{I_{xz}}{I_{xx}}\right)\kappa & \left(L_{\delta_r} + N_{\delta_r} \frac{I_{xz}}{I_{xx}}\right)\kappa \\ \left(N_{\delta_a} + L_{\delta_a} \frac{I_{xz}}{I_{zz}}\right)\kappa & \left(N_{\delta_r} + L_{\delta_r} \frac{I_{xz}}{I_{zz}}\right)\kappa \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\delta_a \\ \Delta\delta_r \end{bmatrix} \tag{10.1}$$

where

$$\kappa = \frac{1}{1 - I_{xz}^2 / I_{xx} I_{zz}} \tag{10.2}$$

For the state-space method, the $I_{xz}$ term does not need to be omitted because MATLAB can handle it easily. The process of matching the Navion's lateral-directional plant matrix to the Twin Otter's plant matrix is the same as the longitudinal case described in CHAPTER 5. The lateral-directional feedback gain matrix solved by MATLAB is expressed by:

$$K = B_{Navion} \backslash (A_{TwinOtter} - A_{Navion}) \tag{10.3}$$

$$K = \begin{bmatrix} \frac{\partial \delta_a}{\partial \beta} & \frac{\partial \delta_a}{\partial p} & \frac{\partial \delta_a}{\partial r} & \frac{\partial \delta_a}{\partial \phi} \\ \frac{\partial \delta_r}{\partial \beta} & \frac{\partial \delta_r}{\partial p} & \frac{\partial \delta_r}{\partial r} & \frac{\partial \delta_r}{\partial \phi} \end{bmatrix} = \begin{bmatrix} -0.0898 & -0.0507 & 0.0901 & 0 \\ 0.3645 & 0.0411 & 0.0460 & 0 \end{bmatrix} \tag{10.4}$$

The result of the state-space method produces excellent results, as shown in Figure 10.1 and Figure 10.2. The side-force generated by the rudder is enough to match the $\Delta\dot{\beta}$ equation.

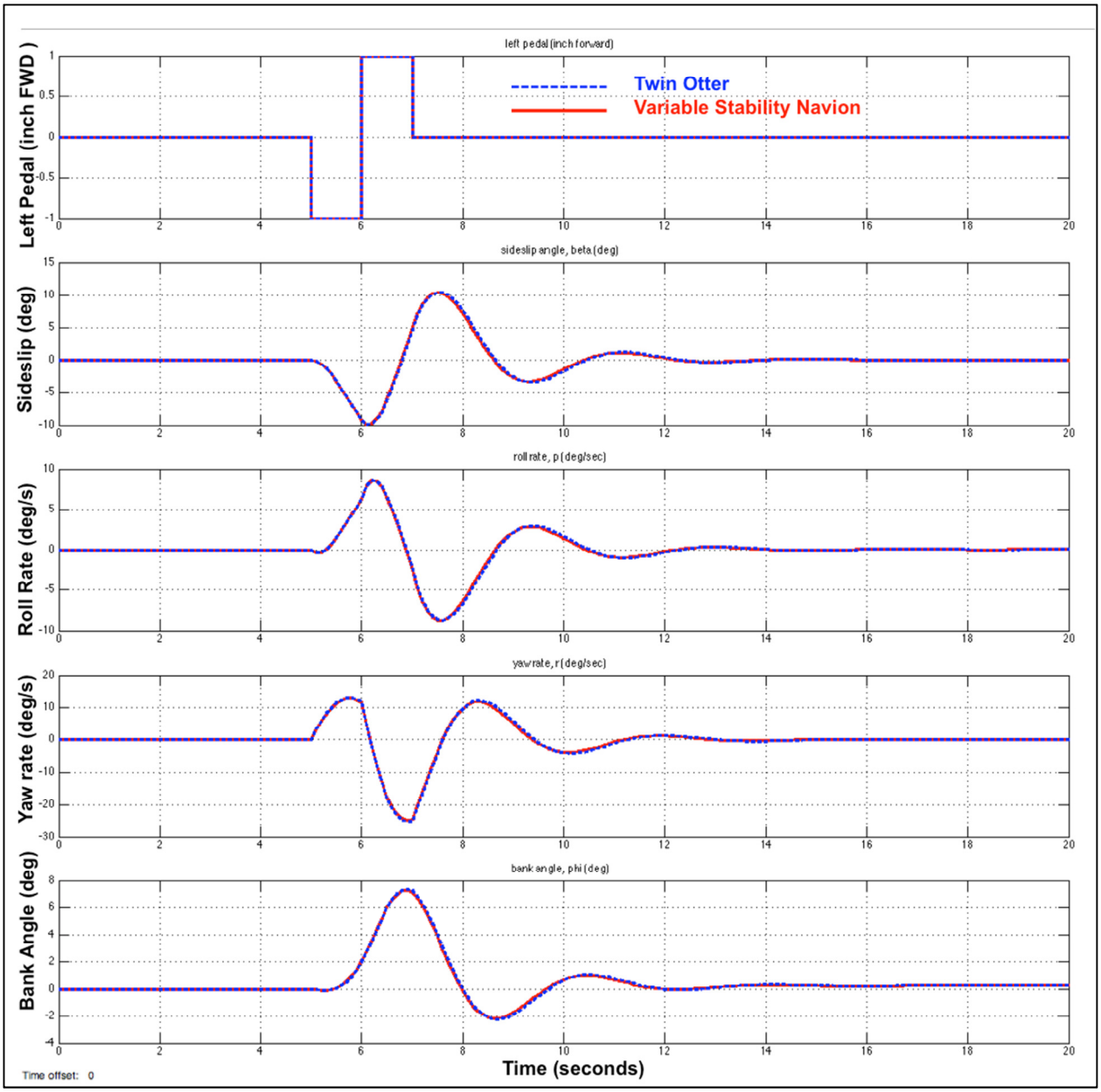


Figure 10.1 Simulating Twin Otter by State-Space Method Rudder Doublet Response

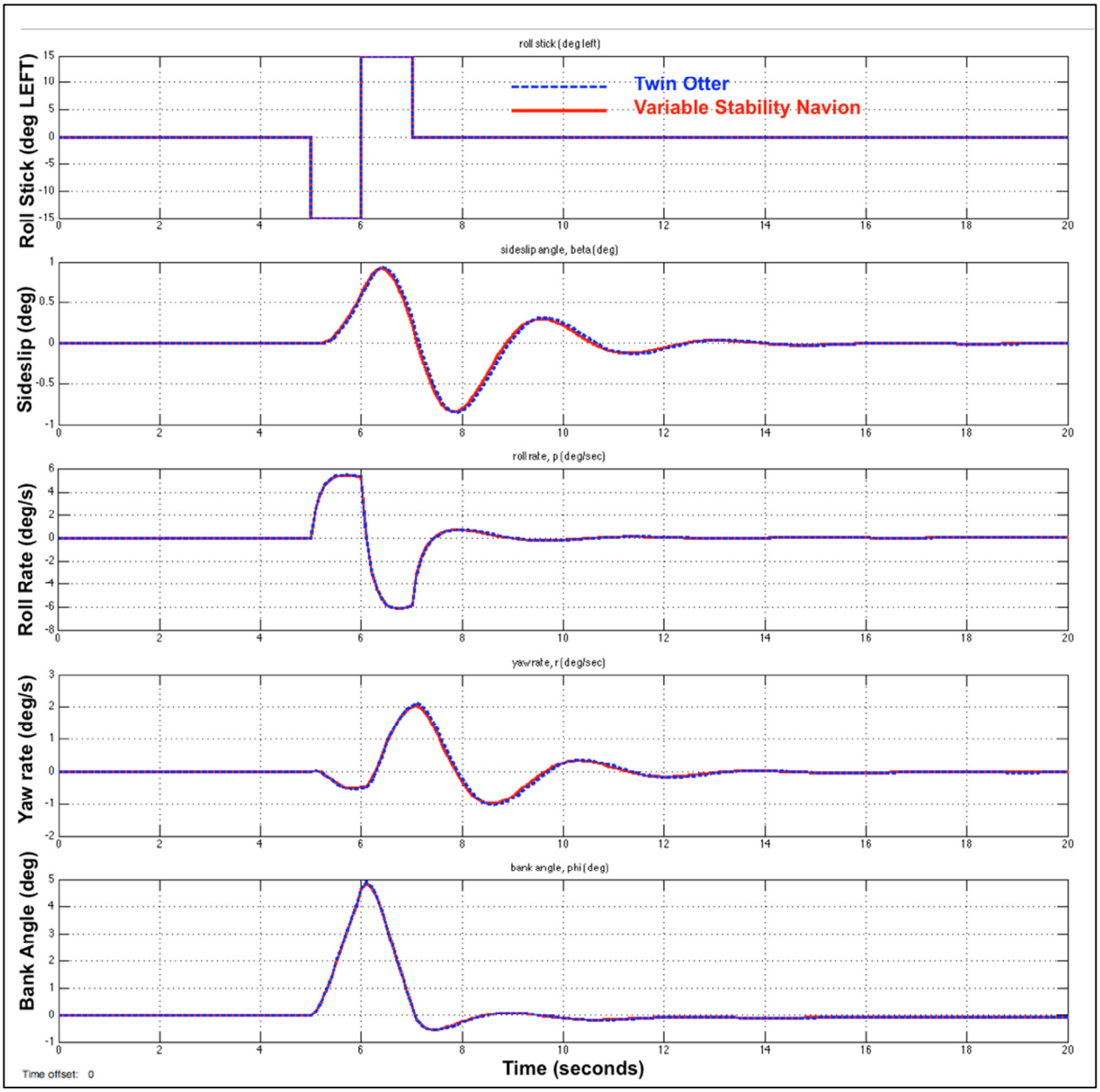


Figure 10.2 Simulating Twin Otter by State-Space Method Aileron Doublet Response

# CHAPTER 11
# DUTCH ROLL, SPIRAL ROLL, AND ROLL MODE APPROXIMATIONS

## 11.1 Approximation Equations

Unlike the longitudinal dynamic responses, the lateral-directional dynamic responses have three modes: Dutch roll, spiral roll, and roll modes.

### 11.1.1 Dutch Roll Mode

The Dutch roll is an oscillatory motion in both roll and yaw axes. Figure 11.1 shows the aircraft disturbed from trimmed condition, in this case a rudder doublet is used to trigger the Dutch roll. This mode is a nuisance to the pilot because it causes the aircraft to roll and yaw undesirably, which increases pilot workload in maintaining heading or bank angle.

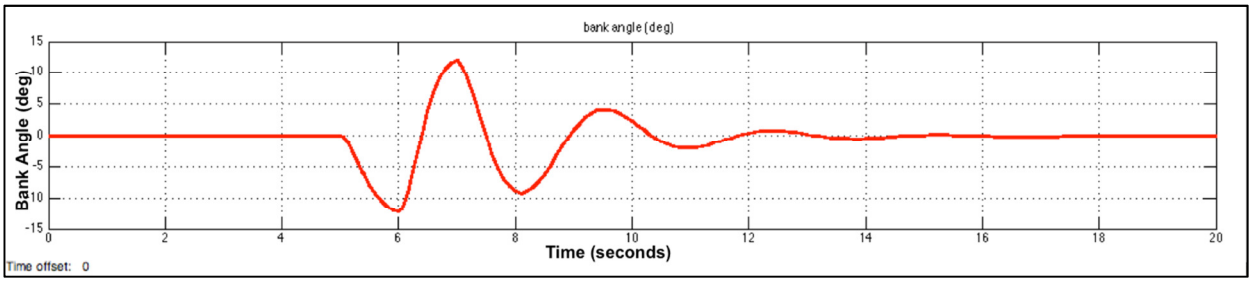


Figure 11.1 Convergent Dutch Roll Oscillation

### 11.1.2 Spiral Mode

The spiral roll is the aircraft's tendency to roll into a turn or roll out of a turn. When the spiral mode is divergent, the pilot is required to hold the aileron in opposite direction as the turn to prevent the aircraft from tightening the turn; it can be dangerous when flying in clouds. Convergent spiral mode requires the pilot to hold the stick in the direction of turn, should the pilot relaxes on the stick, the aircraft would roll out of the turn, which is desirable in general aviation, especially for student pilots. In Figure 11.2, the aircraft is initially at 20° angle of bank turn, the pilot then returns the stick back to neutral position and the aircraft continues to roll into the turn. Although the aircraft has an unstable spiral mode, its increase in bank angle is very small over 20 seconds.

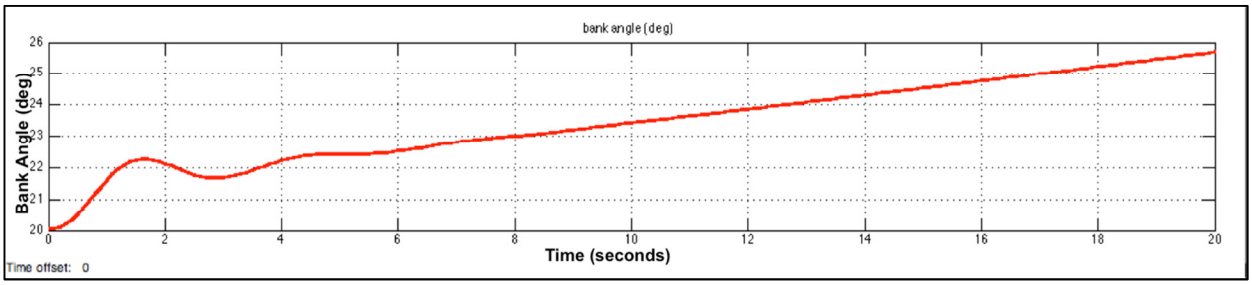


Figure 11.2 Slightly Divergent Spiral Roll

### 11.1.3 Roll Mode

The roll mode is the mode that allows the aircraft to roll or bank. When the pilot puts the stick to the left or right, he expects the aircraft to roll in that direction timely. For military fighter jets, the aircraft is expected to roll sharply and predictably. In large transport aircraft, the roll should be gentle, but also predictable. Figure 11.3 shows the pilot puts in an abrupt right stick input and the aircraft took approximately 0.5 seconds to reach the steady roll rate of 25°/second.

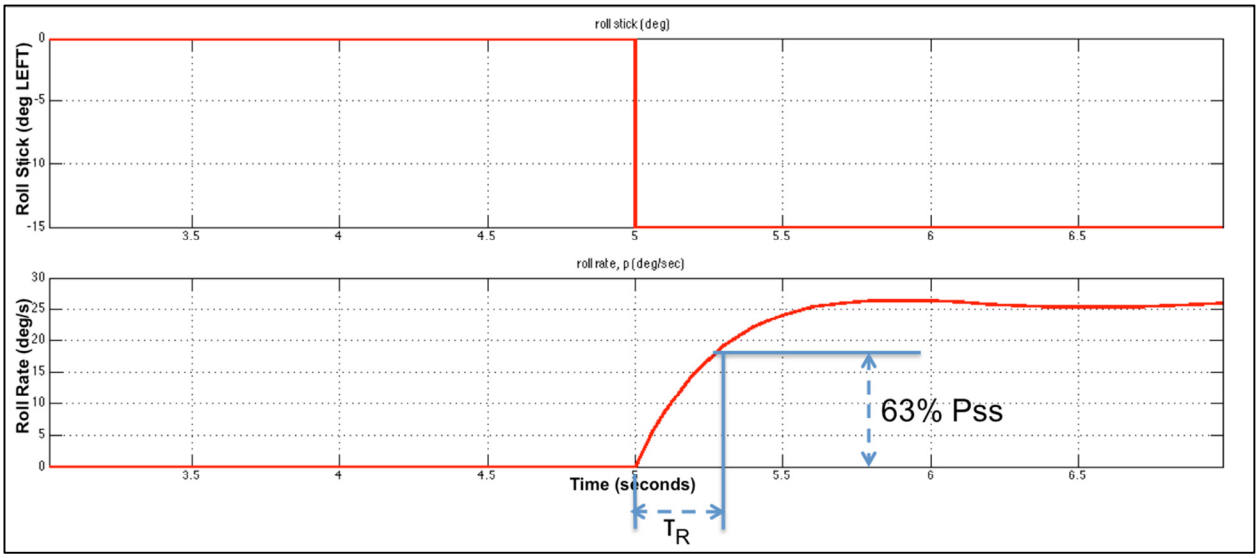


Figure 11.3 Roll Mode

## 11.2 Lateral-Directional Approximation Summary

The Dutch roll oscillations can be modeled as simple mass-spring-damper dynamic system, with specific damping ratios and natural frequencies. The roll mode and spiral roll are not oscillatory, but they can be expressed in terms of time constants τ (time to reach 63% of steady state value), or time to half/double amplitude. The lateral-directional stick-fixed dynamic response approximations are summarized in Table 11.1 (Nelson, 2007, p.198).

Table 11.1 Dutch Roll, Roll, and Spiral Mode Approximations

| | Dutch Roll | Spiral | Roll |
|---|---|---|---|
| Natural Frequency | $\omega_{n_{DR}} = \sqrt{\frac{Y_\beta N_r - N_\beta Y_r + u_0 N_\beta}{u_0}}$ | | |
| Damping Ratio | $\zeta_{DR} = -\frac{1}{2\omega_{n_{DR}}}\left(\frac{Y_\beta + u_0 N_r}{u_0}\right)$ | | |
| Time Constant | | $\tau_s = \frac{L_\beta}{-L_\beta N_r + L_r N_\beta}$ Positive for Convergence | $\tau_r = -\frac{1}{L_p}$ |
| Time to Half or Double (Cook, 2013, p.217) | | $t_{s_{\frac{1}{2}}}$ or $t_{s_2} = \tau_s \ln 2$ | $t_{r_{\frac{1}{2}}}$ or $t_{r_2} = \tau_r \ln 2$ |

The Dutch roll, roll mode, and spiral roll for both the Navion and Twin Otter can be easily calculated by using the approximation equations above. The results are listed in the following tables, and are compared to the exact solutions. The spiral mode approximation equation has very poor and unreliable results.

Table 11.2 Basic Navion Dutch Roll, Roll Mode, and Spiral Mode Approximations

| Basic Navion | Dutch Roll | | | Spiral | | Roll | |
|---|---|---|---|---|---|---|---|
| | Damping Ratio $\zeta$ | Natural Frequency $\omega_n$ (rad/s) | Un-damped Period (s) | Time Constant $\tau_s$ (s) | Time to half or double $t_{1/2 \text{ or } 2}$ | Time Constant $\tau_s$ (s) | Time to half or double $t_{1/2 \text{ or } 2}$ |
| Approximation Equations | 0.233 | 2.18 | 2.88 | 7.36 | 5.10 | 0.119 | 0.0825 |
| Exact Solution by MATLAB damp() function | 0.203 | 2.40 | 2.61 | -122 | -84.6- | 0.119 | 0.0822 |
| Difference | 14% | -9% | 10% | -106% | -106% | 0% | 0% |

Table 11.3 Twin Otter Dutch Roll, Roll Mode, and Spiral Mode Approximations

| Basic Navion | Dutch Roll | | | Spiral | | Roll | |
|---|---|---|---|---|---|---|---|
| | Damping Ratio $\zeta$ | Natural Frequency $\omega_n$ (rad/s) | Un-damped Period (s) | Time Constant $\tau_s$ (s) | Time to half or double $t_{1/2 \text{ or } 2}$ | Time Constant $\tau_s$ (s) | Time to half or double $t_{1/2 \text{ or } 2}$ |
| Approximation Equations | 0.337 | 1.78 | 3.53 | 1.69 | 1.17 | 0.168 | 0.117 |
| Exact Solution by MATLAB damp() function | 0.293 | 1.88 | 3.34 | -38 | -26.3 | 0.166 | 0.115 |
| Difference | 2% | -5% | 6% | -104% | -106% | 1% | 2% |

## 11.3 Dutch Roll Matching

If a desired aircraft's Dutch roll natural frequency and damping ratio are known, then the Variable Stability Navion can simulate the Dutch roll by selecting feedback gain for $N_\beta$ and $N_r$:

$$N_\beta = \frac{Y_\beta N_r - u_0 \omega_{n_{DR}}^2}{Y_r - u_0} \tag{11.1}$$

$$N_r = \frac{u_0 2\omega_{n_{DR}} \zeta_{DR} + Y_\beta}{-u_0} \tag{11.2}$$

Once the required $N_\beta$ and $N_r$ are calculated, the feedback gains can be found by using the term-by-term pairs method. The following table summarizes the gains required to simulate a wide range of Dutch roll frequencies and damping ratios as suggested by the proposed training syllabus (Ball, 1993, p.3-28).

Table 11.4 Dutch Roll Simulation Potentiometer Settings

| Description | $\omega_{nDR}$ (rad/s) | $\zeta_{DR}$ | $\frac{\partial \delta_r}{\partial \beta}$ | $N_\beta$ Pot | $\frac{\partial \delta_r}{\partial r}$ | $N_r$ Pot |
|---|---|---|---|---|---|---|
| Light Damping | 2 | 0.05 | 0.159 | N/A | -0.170 | N/A |
| No Damping | 2 | 0 | 0.159 | | -0.212 | |
| Divergent | 2 | -0.08 | 0.159 | | -0.278 | |
| Yaw Damper | 2 | 0.3 | 0.159 | | 0.0388 | |
| Deadbeat Learjet | 2 | 0.7 | 0.159 | | 0.373 | |
| Fast | 4 | 0.7 | -2.34 | | 0.957 | |
| Slow | 1 | 0.7 | 0.785 | | 0.0805 | |
| Twin Otter | 1.78 | 0.337 | 0.260 | | 0.0387 | |

An example of matching Dutch roll is shown in Figure 11.4. This would simulate a neutral Dutch roll in-flight.

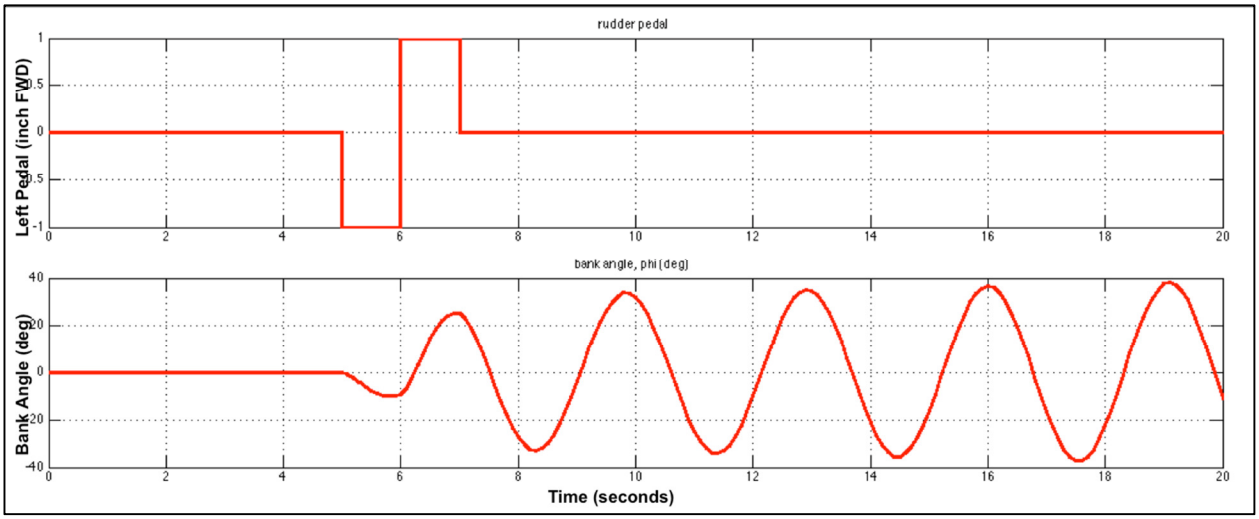


Figure 11.4 Simulation of Neutral Dutch Roll $\zeta_{DR=}0$ and $\omega_{nDR}$=2 rad/s

## 11.4 Roll Mode Matching

If a desired aircraft's roll mode time constant is known, then the Variable Stability Navion can simulate the roll mode by selecting feedback gain for $L_p$:

$$L_p = -\frac{1}{\tau_R} \tag{11.3}$$

Once the required $L_p$ is calculated, the feedback gain can be found by using the term-by-term pairs method. The following table summarizes the gain settings required to simulate three roll mode time constants suggested by the proposed training syllabus (Ball, 1993, p.3-28).

Table 11.5 Roll Mode Potentiometer Settings

| Description | $\tau_R$ (s) | $\frac{\partial \delta_a}{\partial p}$ | $L_p$ Pot |
|---|---|---|---|
| Medium | 0.3 | -0.169 | N/A |
| Short | 0.15 | -0.0576 | |
| Long | 1.3 | -0.254 | |
| Twin Otter | 0.168 | -0.0816 | |

An example of matching roll mode is shown in Figure 11.5, which is the medium roll mode case.

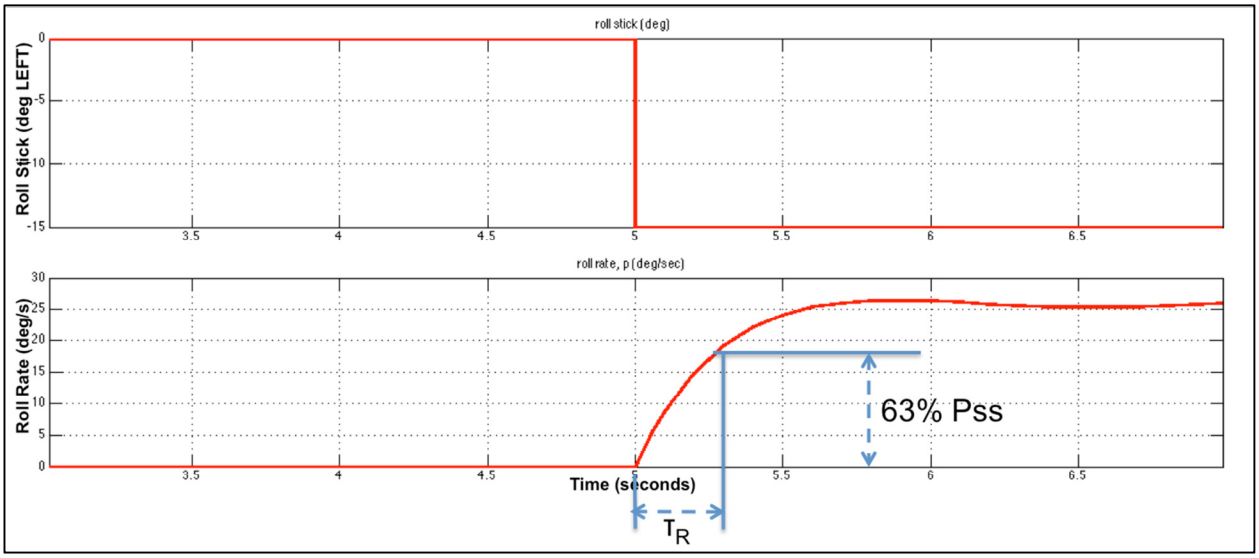


Figure 11.5 Matching Roll Mode $\tau_R$ =0.3 s

## 11.5 Spiral Roll Matching

If a desired aircraft's spiral roll time constant is known, then the Variable Stability Navion can simulate the spiral roll mode by selecting feedback gain for $L_\beta$:

$$L_\beta = \frac{\tau_s L_r N_\beta}{\tau_s N_r - 1} \tag{11.4}$$

Once the required $L_\beta$ is calculated, the feedback gain can be found by using the term-by-term pairs method. The spiral approximation method yields poor results based on its inaccurate spiral mode time constant approximation. As such, a trial and error approach is required to acquire desired values. The following table summarizes the potentiometer settings required to simulate three roll mode constants suggested by the proposed training syllabus (Ball, 1993, p.3-30).

Table 11.6 Spiral Roll Simulation Potentiometer Settings

| Description | $\tau_s$ (s) | $\frac{\partial \delta_a}{\partial \beta}$ | $L_\beta$ Pot |
|---|---|---|---|
| Neutral | 100 | -0.0892 | N/A |
| Convergent | 10 | -0.0289 | |
| Divergent | 0 | -0.532 | |
| Twin Otter | 1.69 | 1.70 | |

# CHAPTER 12
# LATERAL-DIRECTIONAL POLE PLACEMENT METHOD

This chapter will show how the Dutch roll, spiral, and roll modes are located on the complex plane. The method described here is similar to the longitudinal pole placement procedure described in CHAPTER 7.

## 12.1 Pole Placement Method

Step 1. Assembling Short Period and Phugoid Poles. The eigenvalues, or poles, for the lateral-directional case are in the form one real eigenvalue for the roll mode, one real eigenvalue for the spiral mode, and a complex conjugate pair eigenvalues for the Dutch roll mode. The eigenvalues are defined by the following equations:

$$\lambda_r = -\frac{1}{\tau_r} \tag{12.1}$$

$$\lambda_s = -\frac{1}{\tau_s} \tag{12.2}$$

$$\lambda_{DR} = -\zeta_{DR}\omega_{n_{DR}} \pm i\omega_{n_{DR}}\sqrt{1-\zeta_{DR}^2} \tag{12.3}$$

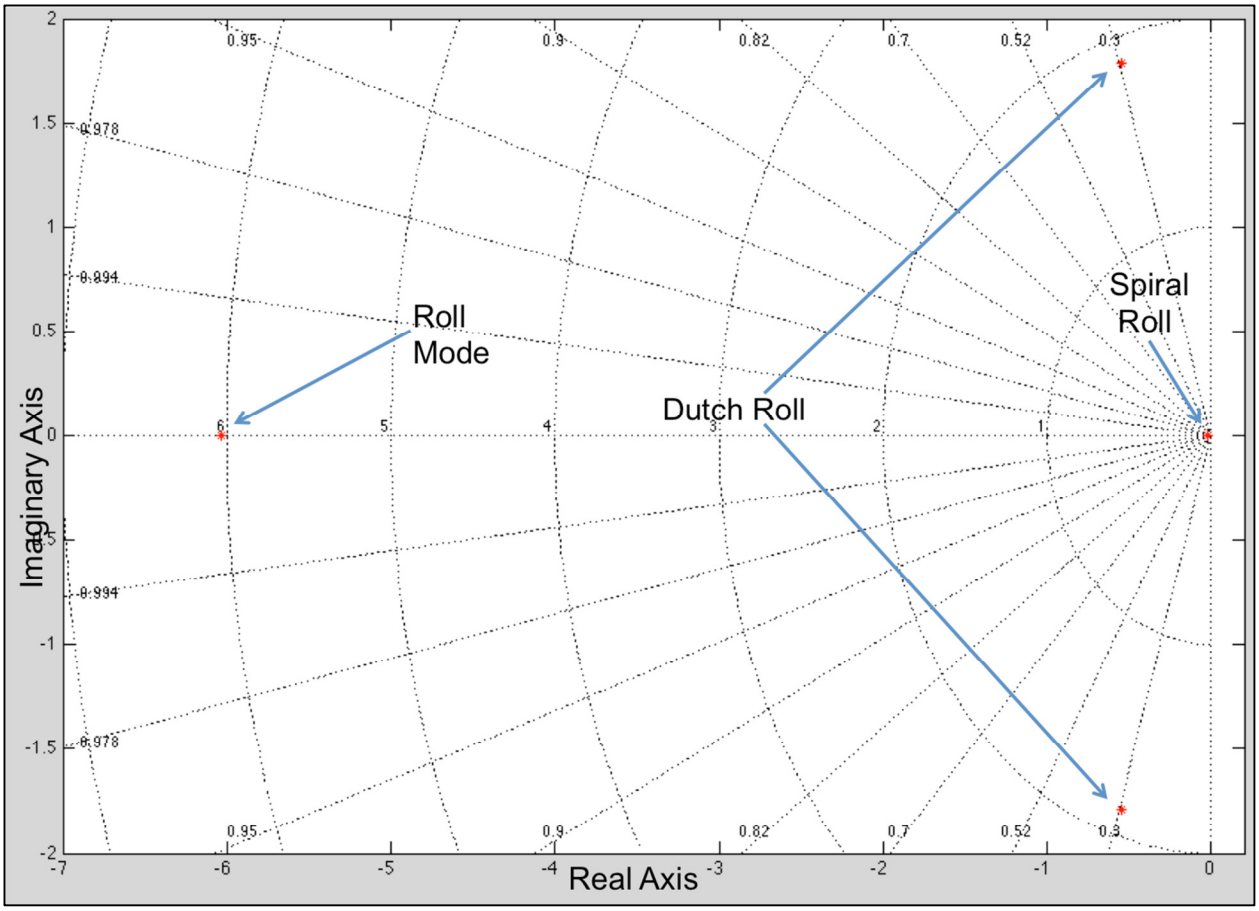


Figure 12.1 Dutch Roll, Spiral, and Roll Mode in Complex Plane

The MATLAB damp() function requires taking in the aircraft's plant matrix $A$ to calculate the eigenvalues. The Twin Otter's Dutch roll, spiral, and roll mode calculated by the damp() function are shown in Table 12.1

Table 12.1 Twin Otter Dutch Roll, Spiral, and Roll Mode Characteristics

| Twin Otter | Eigenvalue | Damping Ratio $\zeta$ | Natural Frequency $\omega_n$ (rad/s) | Time Constant (s) |
|---|---|---|---|---|
| Dutch Roll | 0.550+1.79i | 0.293 | 1.88 | |
| | 0.550-1.79i | | | |
| Spiral Roll | -0.0236 | 1 | 0.0236 | 42.4 |
| Roll Mode | -6.04 | 1 | 6.04 | 0.166 |

Step 2. Find Feedback Gains to Place Poles. Using MATLAB's place() command, it calculates the rudder feedback gains that can force the poles of the basic Navion to the locations of Twin Otter's poles. The control matrix *B* will be modified such that only the rudder can be used to control the aircraft's response. The *B* matrix now becomes:

$$B = \begin{bmatrix} 0 & \dfrac{Y_{\delta_r}}{u_0} \\ 0 & \left( L_{\delta_r} + N_{\delta_r} \dfrac{I_{xz}}{I_{xx}} \right) \kappa \\ 0 & \left( N_{\delta_r} + L_{\delta_r} \dfrac{I_{xz}}{I_{zz}} \right) \kappa \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0.0707 \\ 0 & 23.1 \\ 0 & -4.61 \\ 0 & 0 \end{bmatrix} \tag{12.4}$$

Using MATLAB's place() command, it calculates the rudder feedback gains that can force the Navion to simulate different lateral-directional dynamics. Attention must be given when using MATLAB to calculate feedback gains because MATLAB assumes negative feedback. The feedback gains matrix *K* is:

$$K = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \dfrac{\partial \delta_r}{\partial \beta} & \dfrac{\partial \delta_r}{\partial p} & \dfrac{\partial \delta_r}{\partial r} & \dfrac{\partial \delta_r}{\partial \phi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.4676 & 0.0935 & -0.0123 & -0.0219 \end{bmatrix} \tag{12.5}$$

## 12.2 Simulink Results

The pole placement method is intended to match oscillation frequencies and damping ratios, not individual equations of motion. The pole placement method produces good results for simulating lateral-dimensional dynamics. Figure 12.2 and Figure 12.3 demonstrate the pole placement method of matching Twin Otter's Dutch roll mode.

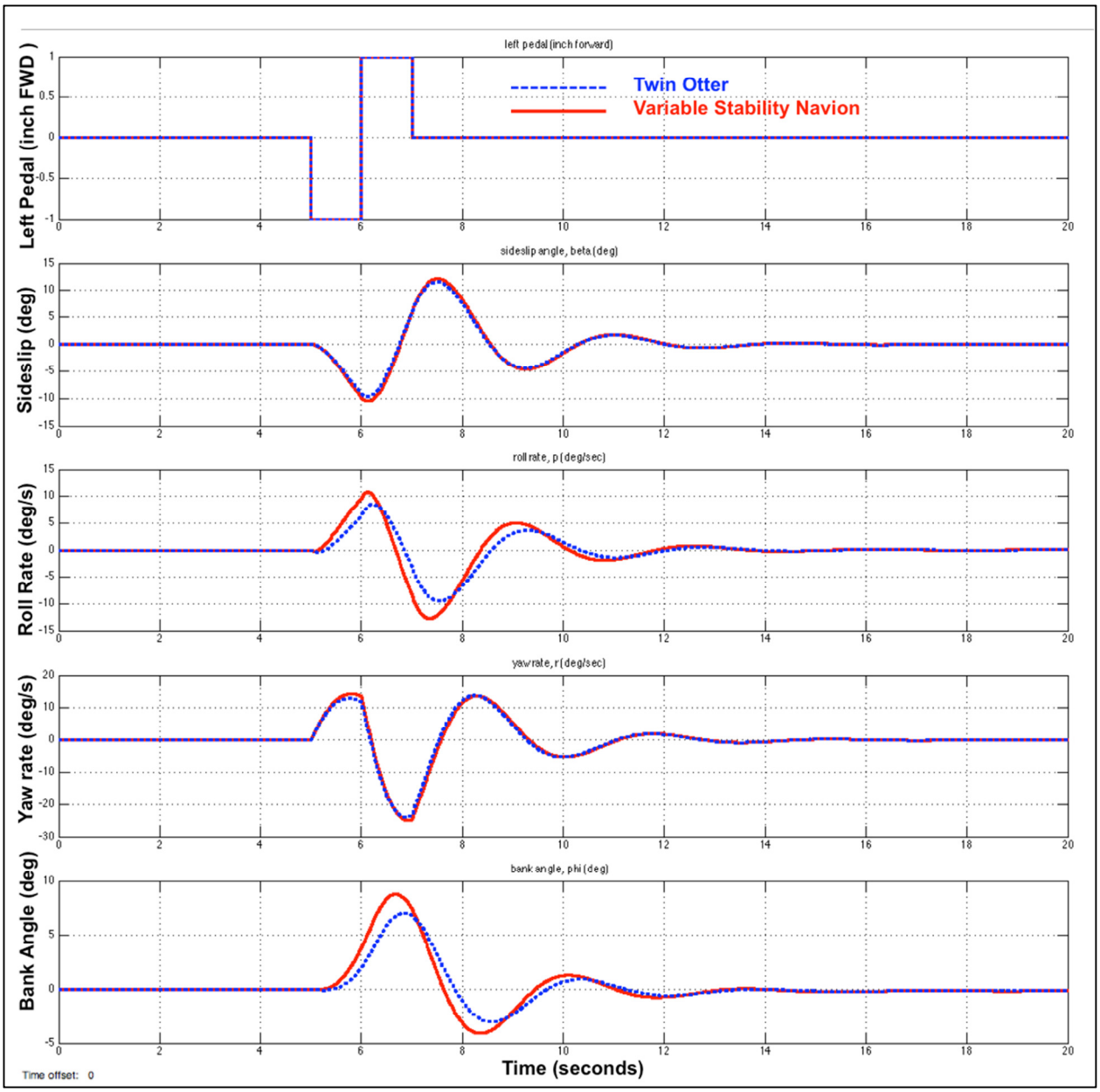


Figure 12.2 Matching Twin Otter Dutch Roll by Pole Placement Method (Rudder Doublet)

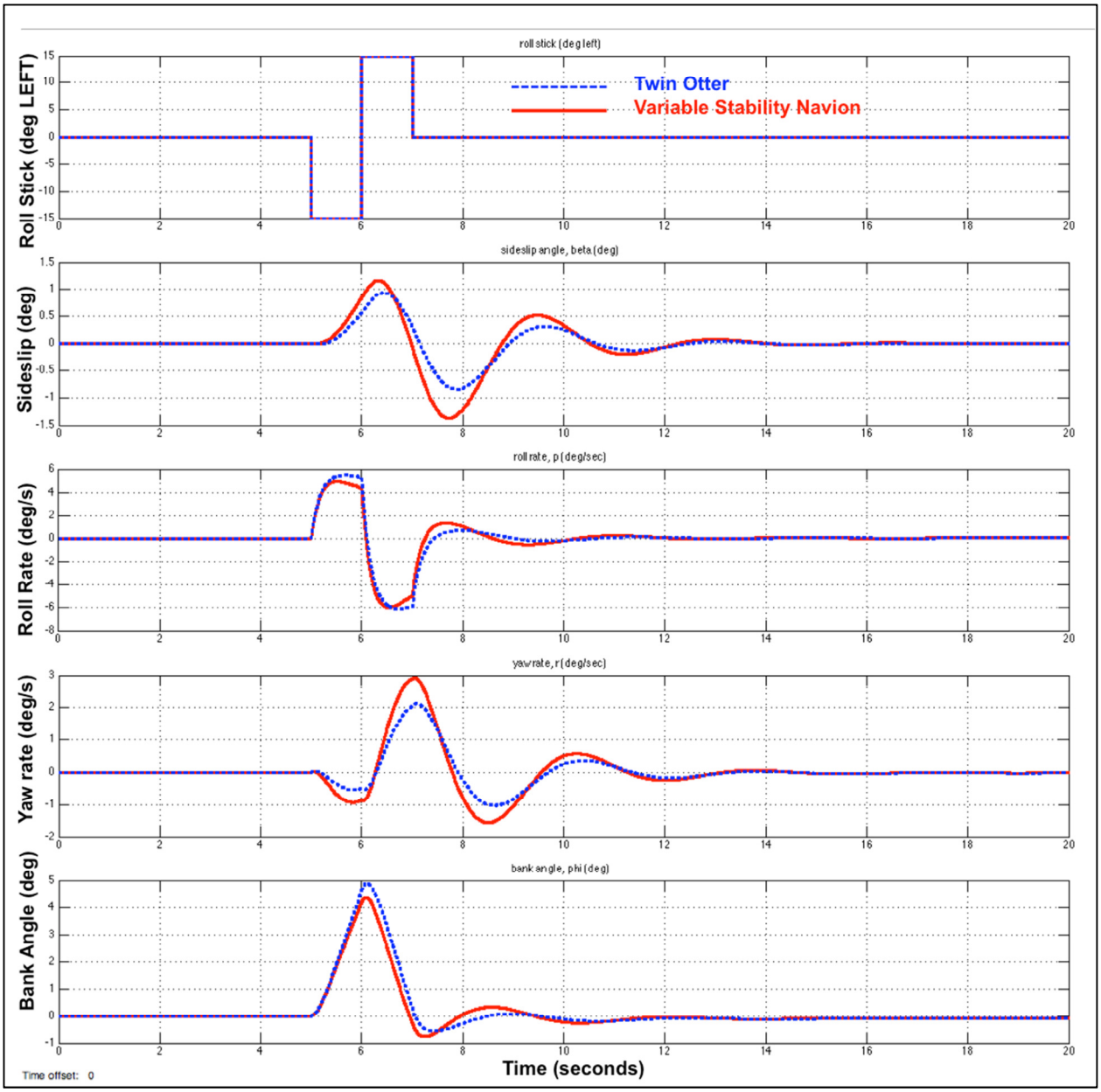


Figure 12.3 Matching Twin Otter Dutch Roll Using Pole Placement Method (Aileron Doublet)

To demonstrate matching of roll mode time constant, an aileron step input is used, as illustrated in the following figure:

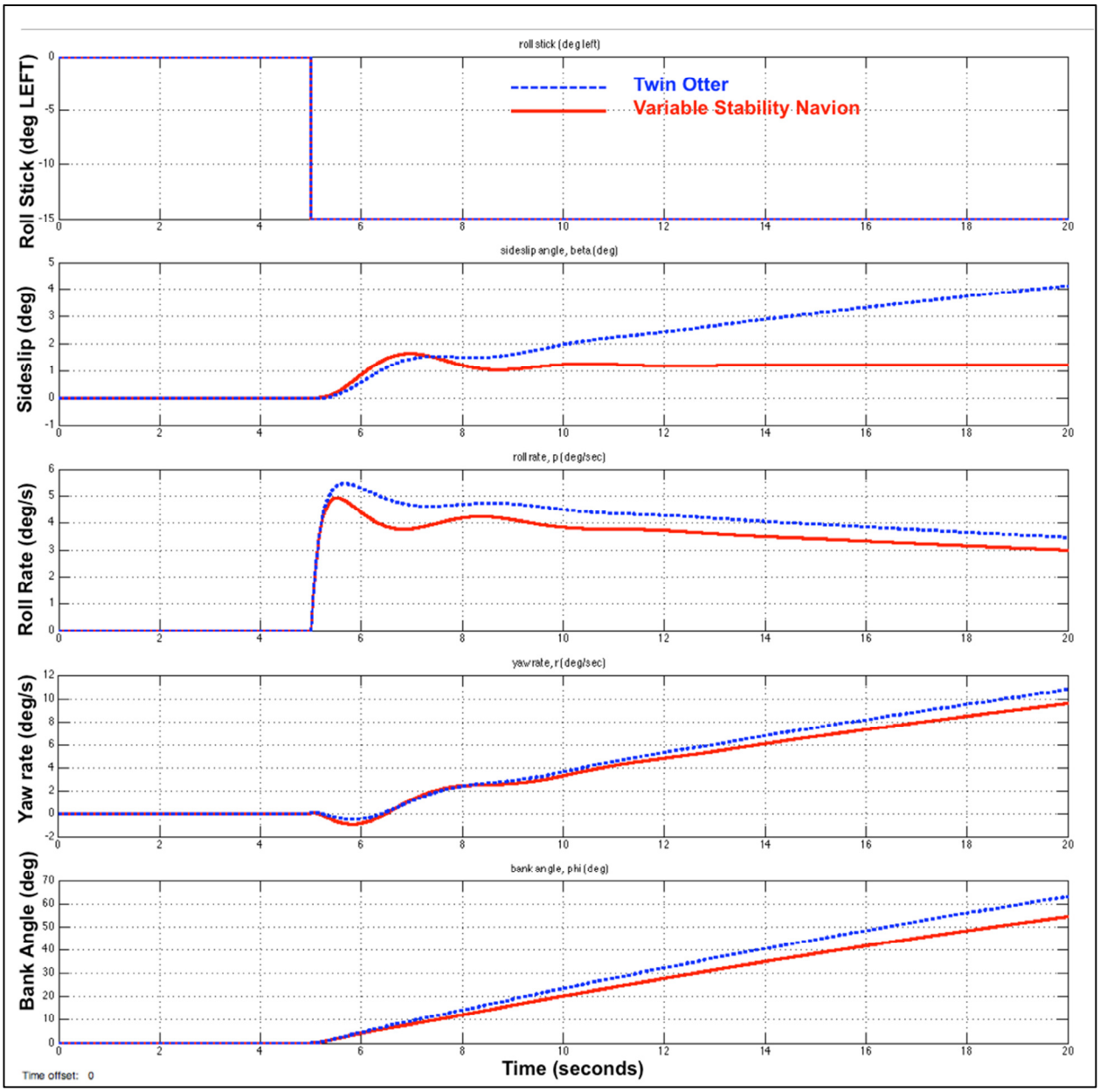


Figure 12.4 Matching Twin Otter Roll Mode Using Pole Placement Method (Aileron Step)

To demonstrate matching of spiral roll mode time constant, the aircraft is setup to have a 20° angle of bank, and at time zero, the aircraft's convergent spiral mode rolls the aircraft out slowly, as illustrated in the following figure:

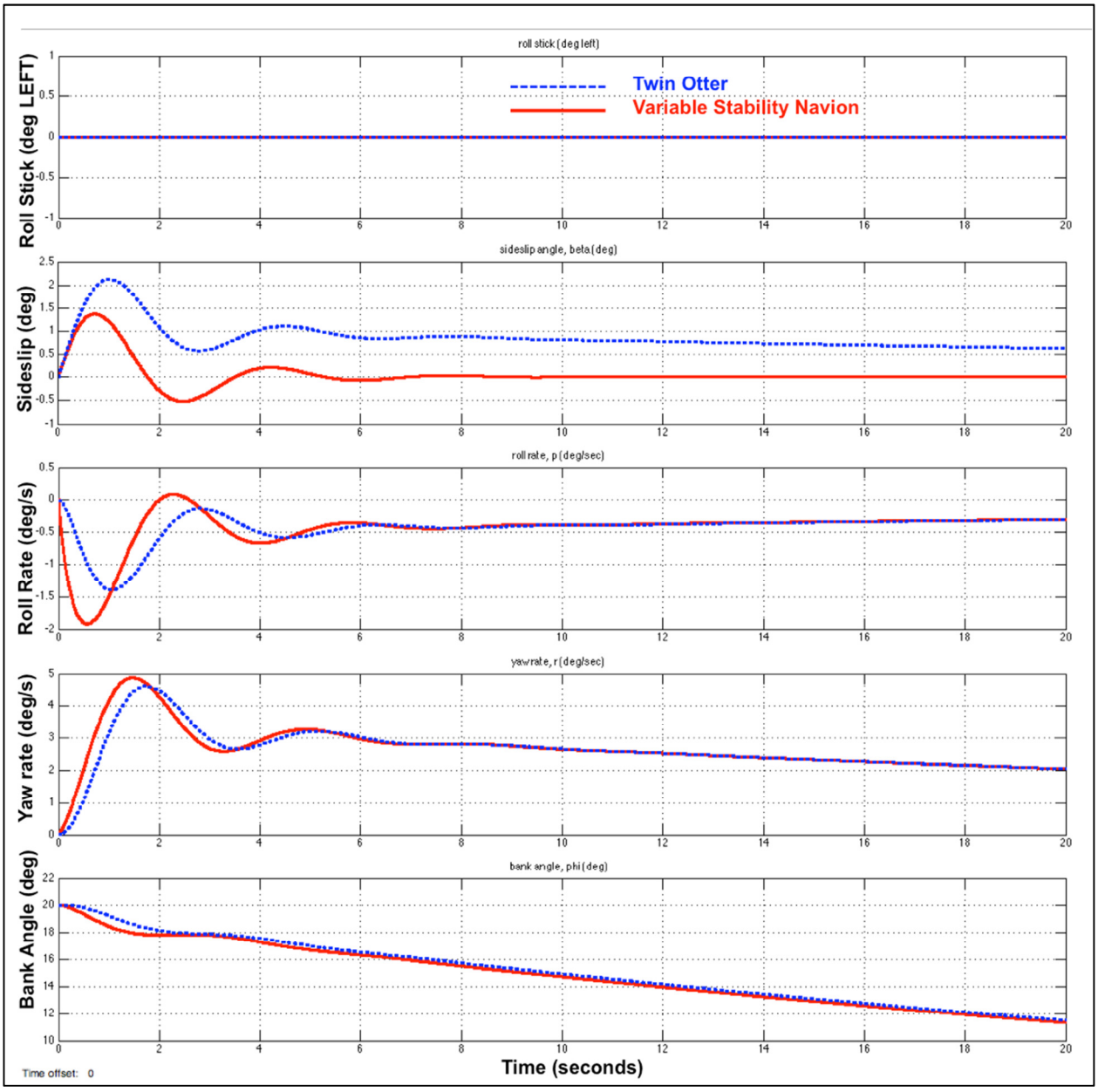


Figure 12.5 Matching Twin Otter Spiral Roll Using Pole Placement Method (Time to Half Amplitude)

Finally, to illustrate the pole placement method's ability to choose specific Dutch roll characteristics from the Calspan training syllabus (Ball et al., 1993, p.3-28), a Dutch roll with damping ratio of 0.05, and natural frequency of 2 rad/s is chosen, as shown in Figure 12.6.

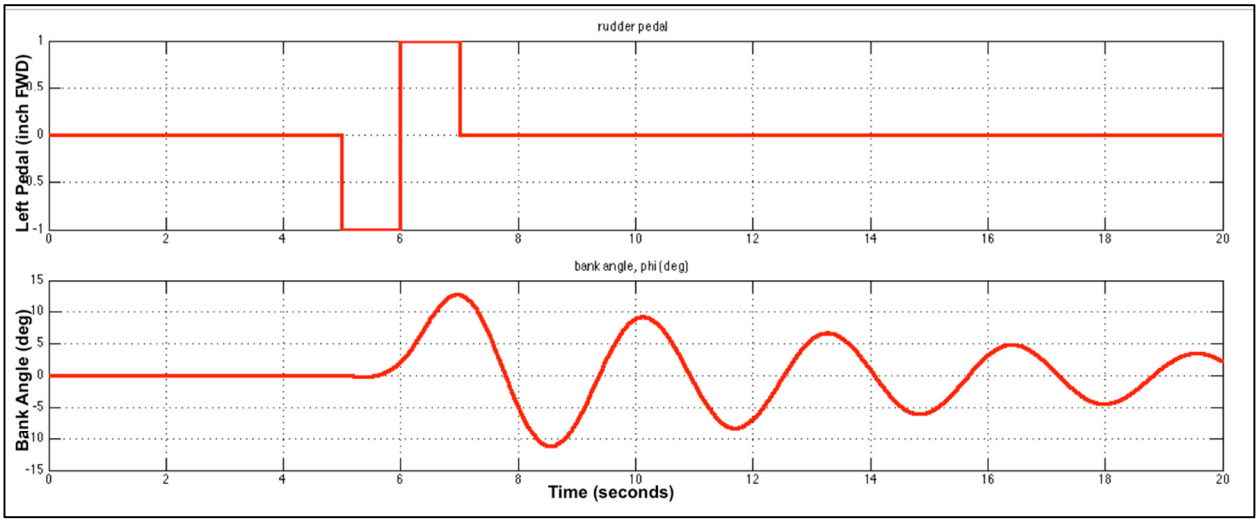


Figure 12.6 Matching Dutch Roll $\zeta_{DR}$=0.05 and $\omega_{nDR}$=2 rad/s Using Pole Placement Method

# CHAPTER 13
# 6-DOF MODEL AND FLIGHT GEAR SIMULATION

*"All models are wrong, but some are useful." - Statistician George E.P. Box, 1979.*

## 13.1 Verification and Validation

The ultimate validation of the control laws described in this thesis is to compare them with actual flight test data. In absence of flight data, there are a few simulation options to allow reliable verifications.

## 13.2 3DOF Decoupled Linear Flight Model

This Simulink model decouples longitudinal and lateral-directional equations of motion into two separate state-spaces (APPENDIX E). NASA has demonstrated by flight test that linear models can provide good predictions of dynamics for flight vehicles (Bosworth, 1992). This model is simple and easy to implement.

## 13.3 6DOF Non-Linear Flight Model

To simulate the control laws and flight dynamics with higher fidelity, a 6DOF non-linear equations of motion Simulink model can be used. The 6-DOF model also includes simple actuator and servo dynamics to simulate rate limit and physical surface deflection stops (APPENDIX E).

## 13.4 FlightGear Simulation

The MATLAB scripts and the Simulink 6DOF Model can be connected to the desktop Flight Gear simulator to allow a hands-on "pilot-in-the-loop" flying experience. Using a joystick, users can visually see how the Navion simulates desired aircraft, and how the aircraft responds to control inputs.

This FlightGear simulator can help students "chair-fly" and experiment with the Variable Stability Navion prior to the actual demonstration flight in air.

Since this simulator is built based on a set of constant stability coefficients, the limitation is that the flight must remain at or near the desired trimmed speed.


Figure 13.1 Desktop Flight Gear Simulation

# CHAPTER 14
# CONCLUSIONS AND RECOMMENDATIONS

## 14.1 Conclusion

Multiple methods are shown in this thesis to allow the Variable Stability Navion aircrew to calculate potentiometer settings required to simulate response of different aircraft.

For longitudinal motions, four methods were introduced: term-by-term stability derivative matching, state-space method, short period and phugoid approximations, and pole placement method. All four methods can successfully produce control law required to simulate desired aircraft response. The methods yield satisfactory results in matching the Variable Stability Navion to the Twin Otter, with the state-space and pole placement methods being the most accurate.

For the lateral-directional motions, four methods were introduced: term-by-term stability derivative pair matching, state-space method, Dutch roll/spiral roll/roll mode approximations, and pole placement method. The state-space and pole placement methods can most successfully produce control law required to match the Twin Otter and other desired responses.

These methods are coded as MATLAB scripts to allow timely calculation of flight control laws. The calculated flight control laws can be verified rapidly by Simulink simulations, and pilot-in-the-loop flight simulation using FlightGear desktop software. The pilot will have an idea of how the Variable Stability Navion operates and how it would react prior to the actual flight. The familiarization of the variable stability operation will enhance the in-flight demonstration value, and maximize learning experience during limited flight time.

## 14.2 Recommendations

### 14.2.1 Potentiometer Calibration

The only official potentiometer calibration manual was released in 1981. Throughout the year, the aircraft had been modified several times for different test programs. Sensors have been upgraded or replaced. A comprehensive sensors and potentiometer calibration ground test should be conducted to produce an updated calibration data manual.

For example, to calibrate the $M_{\delta e}$ potentiometer, a suggested process would be to set the potentiometer to 0, 30, 60, and 90. At each potentiometer setting, the pitch stick in the

cockpit is displaced and displacement measured. The corresponding elevator deflection is also measured and recorded.

Table 14.1 $M_{\delta e}$ Potentiometer Calibration Ground Test Card

| **Pot = 0** | | | **Pot = 30** | | | **Pot = 60** | | | **Pot = 90** | |
|---|---|---|---|---|---|---|---|---|---|---|
| **$\delta_{ps}$** | **$\delta_e$** | | **$\delta_{ps}$** | **$\delta_e$** | | **$\delta_{ps}$** | **$\delta_e$** | | **$\delta_{ps}$** | **$\delta_e$** |
| Full Fwd | | | Full Fwd | | | Full Fwd | | | Full Fwd | |
| 10° Fwd | | | 10° Fwd | | | 10° Fwd | | | 10° Fwd | |
| 0° | | | 0° | | | 0° | | | 0° | |
| 10° Aft | | | 10° Aft | | | 10° Aft | | | 10° Aft | |
| Full Aft | | | Full Aft | | | Full Aft | | | Full Aft | |
| $\frac{\partial \delta_e}{\partial \delta_{PS}}$ | | | $\frac{\partial \delta_e}{\partial \delta_{PS}}$ | | | $\frac{\partial \delta_e}{\partial \delta_{PS}}$ | | | $\frac{\partial \delta_e}{\partial \delta_{PS}}$ | |

The same procedure can be used for other physically adjustable sensors such as angle of attack vane, sideslip vane, and airspeed (connected to pitot test set).

For sensors that cannot be adjusted physically on the ground, such as pitch rate and pitch angle, electrical voltage signal would need to be injected at the sensor input.

### 14.2.2 Potentiometer Scaling

Rather than having a conversion step to find potentiometer values (between -90 to 90), the potentiometers should be labeled or scaled such that the pilot can dial in the actual feedback gain value. This would reduce confusion and error in calculations.

### 14.2.3 System Identification Flight Test

The available Navion stability derivatives are from textbooks and previous flight tests for the basic Navion. To obtain stability derivatives for different configurations (altitude, speed, weight, etc), a system identification flight test program should be conducted.

## LIST OF REFERENCES

Aviation System and Flight Research Department. (2004). Variable Stability Operations Manual. Tullahoma: The University of Tennessee Space Institute.

Ball, J., Berthe, C., et al. (1993). Learjet Flight Syllabus and Background Material for the U.S. Air Force/U.S. Navy Test Pilot School Variable Stability Programs. New York: Calspan SRL Corporation.

Bosworth,.J. (1992). Linearized Aerodynamic and Control Law Models of the X-29A Airplane and Comparison with Flight Data. California: NASA Dryden Flight Research Facility.

Bragg, M., Hutchison, T., et al. (2000). AIAA-2000-0360 Effect of Ice Accretion on Aircraft Flight Dynamics. Reno: American Institute of Aeronautics and Astronautics, Inc.

Bryan, G.H. (1911). Stability in Aviation – An Introduction to Dynamical Stability as Applied in the Motion of Aeroplanes. London: MacMillan and Co. Limited.

Cook, M. (2013). Flight Dynamics Principles: a Linear Systems Approach to Aircraft Stability and Control. 3rd ed. Oxford: Butterworth-Heinemann.

Etkin, B., Reid, L. (1996). Dynamics of Flight: Stability and Control. 3rd ed. Hoboken: John Wiley & Sons. Inc.

Kidd, E., Bull, G., Harper, R. (1961). In-Flight Simulation Theory and Application. Paris: North Atlantic Treaty Organization.

Nelson, R. (2007) Flight Stability and Automatic Control. 2nd ed. New York: McGraw Hill Companies, Inc.

Nichols, R., Malagon, M. (1981). VRA and ARA Calibration Data Manual. New Jersey: Princeton University.

Ranaudo, R., Batterson, J., et al. (1989). AIAA-89-0754 Determination of Longitudinal Aerodynamic Derivatives Using Flight Data From an Icing Research Aircraft. Reno: American Institute of Aeronautics and Astronautics, Inc.

Ranaudo, R., Mikkelsen, K., et al. (1986). AIAA-86-9758 The Measurement of Aircraft Performance and Stability and Control After Flight Through Natural Icing Condition. Las Vegas: American Institute of Aeronautics and Astronautics, Inc.

Ratvasky, T., Rauaudo, R. (1993). AIAA-93-0398 Icing Effects on Aircraft Stability and Control Determined from Flight Data. Reno: American Institute of Aeronautics and Astronautics, Inc.

Shivers, J., Fink, M., Ware, G. (1970). Full-Scale Wind-Tunnel Investigation of the Static Longitudinal and Lateral Characteristics of a Light Single-Engine Low-Wing Airplane. Washington: National Aeronautics and Space Administration.

Yechout, T. (2003). Introduction to Aircraft Flight mechanics: Performance, Static Stability, Dynamic Stability, and Classical Feedback Control. Virginia: American Institute of Aeronautics and Astronautics, Inc.

# APPENDICES

## APPENDIX A
## THREE VIEW DIAGRAM OF NAVION

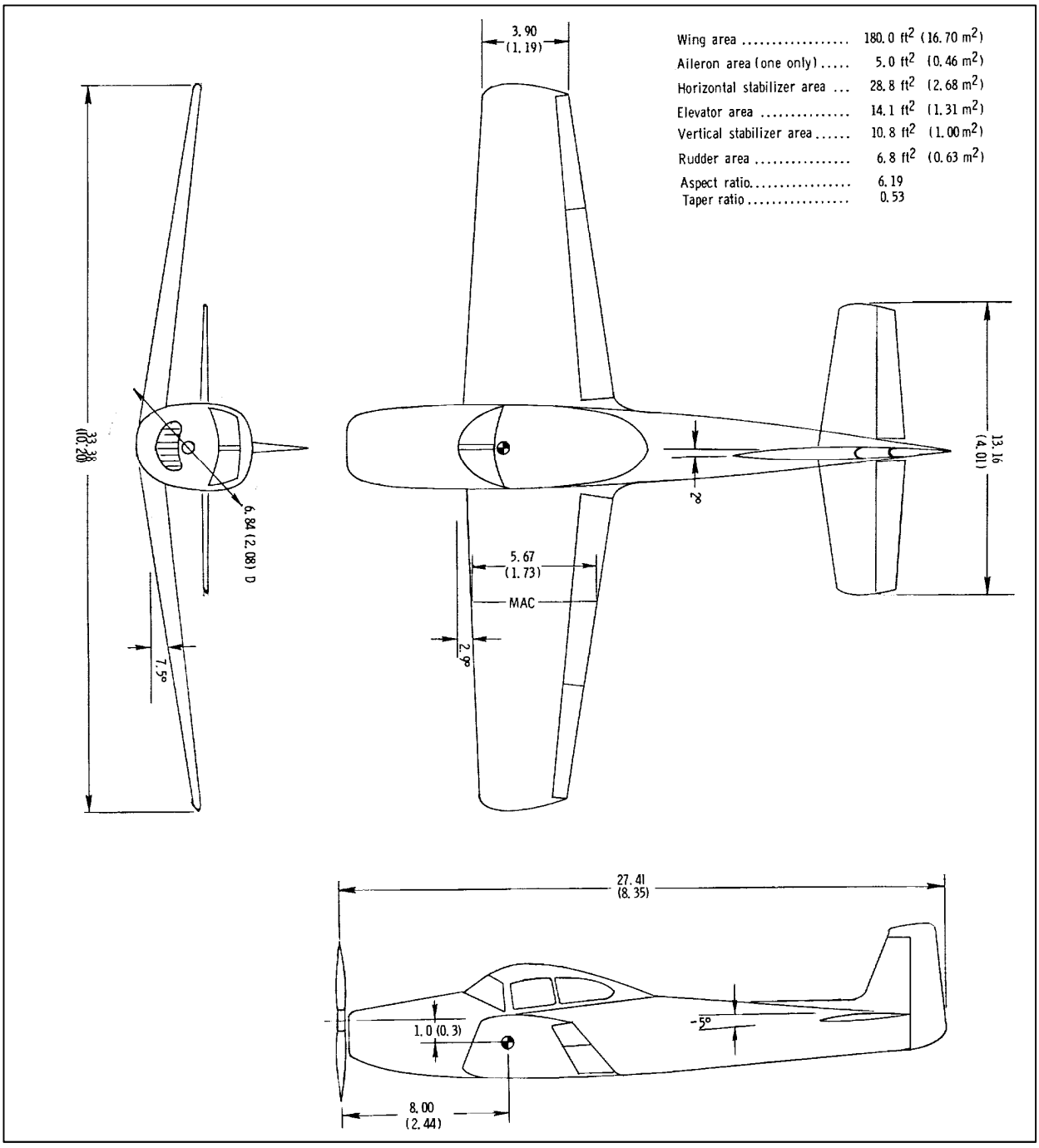


Figure A.1 Three View Diagram of Navion (Shivers et al, 1970)

## APPENDIX B
## BASIC CLEAN NAVION DERIVATIVES

Table B.1 Basic Navion Stability Coefficients

| **Center of Gravity and Mass Characteristics** | | **Reference Geometry** | |
|---|---|---|---|
| W | 2750 lbs | S | 184 $ft^2$ |
| CG | 29.5% MAC | b | 33.4 ft |
| $I_{xx}$ | 1048 $slug.ft^2$ | $\bar{c}$ | 5.7 ft |
| $I_{yy}$ | 3000 $slug.ft^2$ | | |
| $I_{zz}$ | 3530 $slug.ft^2$ | **Initial Conditions** | |
| $I_{xz}$ | 0 $slug.ft^2$ | Hp | Sea Level |
| | | M | 0.158 |
| | | V | 104 KTAS= 104 KCAS= 176 ft/s |
| | | | |
| **Longitudinal** | | **Lateral** | |
| $C_L$ | 0.41 | $C_{y\beta}$ | -0.564 |
| $C_D$ | 0.05 | $C_{l\beta}$ | -0.0740 |
| $C_{L\alpha}$ | 4.44 | $C_{n\beta}$ | 0.0710 |
| $C_{D\alpha}$ | 0.330 | $C_{lp}$ | -0.410 |
| $C_{m\alpha}$ | -0.683 | $C_{lr}$ | 0.107 |
| $C_{L\dot{\alpha}}$ | 0.0 | $C_{nr}$ | -0.125 |
| $C_{m\dot{\alpha}}$ | -4.36 | $C_{l\delta a}$ | -0.134 |
| $C_{Lq}$ | 3.80 | $C_{n\delta a}$ | -0.00350 |
| $C_{mq}$ | -9.96 | $C_{y\delta r}$ | 0.157 |
| $C_{LM}$ | 0.0 | $C_{l\delta r}$ | 0.107 |
| $C_{DM}$ | 0.0 | $C_{n\delta r}$ | -0.0720 |
| $C_{mM}$ | 0.0 | $C_{np}$ | -0.0575 |
| $C_{L\delta e}$ | 0.355 | | |
| $C_{m\delta e}$ | -0.923 | | |

(Nelson, 2007, p.400)

# APPENDIX C
# CLEAN TWIN OTTER DERIVATIVES

Table C.1 Clean Twin Otter Stability Coefficients

| **Center of Gravity and Mass Characteristics** | | **Reference Geometry** | |
|---|---|---|---|
| W | 9149 lbs | S | 422.5 $ft^2$ |
| CG | unspecified | b | 65 ft |
| $I_{xx}$ | 15694 $slug.ft^2$ | $\bar{c}$ | 6.5 ft |
| $I_{yy}$ | 22126 $slug.ft^2$ | | |
| $I_{zz}$ | 33179 $slug.ft^2$ | **Initial Conditions** | |
| $I_{xz}$ | 1056 $slug.ft^2$ | Hp | unspecified |
| | | M | unspecified |
| | | V | 120 KCAS= 202.5 ft/s |
| | | | |
| **Longitudinal** | | **Lateral** | |
| $C_L$ | 0.36 | $C_{y\beta}$ | -0.6 |
| $C_D$ | 0.041 | $C_{l\beta}$ | -0.08 |
| $C_{L\alpha}$ | 5.66 | $C_{n\beta}$ | 0.1 |
| $C_{D\alpha}$ | 0.31 | $C_{lp}$ | -0.5 |
| $C_{m\alpha}$ | -1.31 | $C_{lr}$ | 0.06 |
| $C_{L\dot{\alpha}}$ | unspecified | $C_{nr}$ | -0.18 |
| $C_{m\dot{\alpha}}$ | unspecified | $C_{l\delta a}$ | -0.15 |
| $C_{Lq}$ | 19.97 | $C_{n\delta a}$ | -0.001 |
| $C_{mq}$ | -34.2 | $C_{y\delta r}$ | 0.15 |
| $C_{LM}$ | unspecified | $C_{l\delta r}$ | 0.015 |
| $C_{DM}$ | Unspecified | $C_{n\delta r}$ | -0.12 |
| $C_{mM}$ | Unspecified | $C_{yp}$ | -0.2 |
| $C_{L\delta e}$ | 0.608 | $C_{yr}$ | 0.4 |
| $C_{m\delta e}$ | -1.74 | $C_{np}$ | -0.06 |

(Bragg et al, 2000; Ranaudo et al, 1989)

## APPENDIX D
## ALL ICED TWIN OTTER DERIVATIVES

Table D.1 All Iced Twin Otter Stability Coefficients

| **Center of Gravity and Mass Characteristics** | | **Reference Geometry** | |
|---|---|---|---|
| W | 9149 lbs | S | 422.5 $ft^2$ |
| CG | unspecified | b | 65 ft |
| $I_{xx}$ | 15694 $slug.ft^2$ | $\bar{c}$ | 6.5 ft |
| $I_{yy}$ | 22126 $slug.ft^2$ | | |
| $I_{zz}$ | 33179 $slug.ft^2$ | **Initial Conditions** | |
| $I_{xz}$ | 1056 $slug.ft^2$ | Hp | unspecified |
| | | M | unspecified |
| | | V | 120 KCAS= 202.5 ft/s |
| | | | |
| **Longitudinal** | | **Lateral** | |
| $C_L$ | 0.38 | $C_{y\beta}$ | -0.48 |
| $C_D$ | 0.062 | $C_{l\beta}$ | -0.072 |
| $C_{L\alpha}$ | 5.094 | $C_{n\beta}$ | 0.08 |
| $C_{D\alpha}$ | 0.31 | $C_{lp}$ | -0.45 |
| $C_{m\alpha}$ | -1.180 | $C_{lr}$ | 0.06 |
| $C_{L\dot{\alpha}}$ | Unspecified | $C_{nr}$ | -0.169 |
| $C_{m\dot{\alpha}}$ | Unspecified | $C_{l\delta a}$ | -0.135 |
| $C_{Lq}$ | 19.7 | $C_{n\delta a}$ | -0001 |
| $C_{mq}$ | -33 | $C_{y\delta r}$ | 0.138 |
| $C_{LM}$ | Unspecified | $C_{l\delta r}$ | 0.0138 |
| $C_{DM}$ | Unspecified | $C_{n\delta r}$ | -0.11 |
| $C_{mM}$ | Unspecified | $C_{yp}$ | -0.2 |
| $C_{L\delta e}$ | 0.550 | $C_{yr}$ | 0.4 |
| $C_{m\delta e}$ | -1.566 | $C_{np}$ | -0.06 |

(Bragg et al, 2000; Ranaudo et al, 1989)

**APPENDIX E**
**SIMULINK FILES**

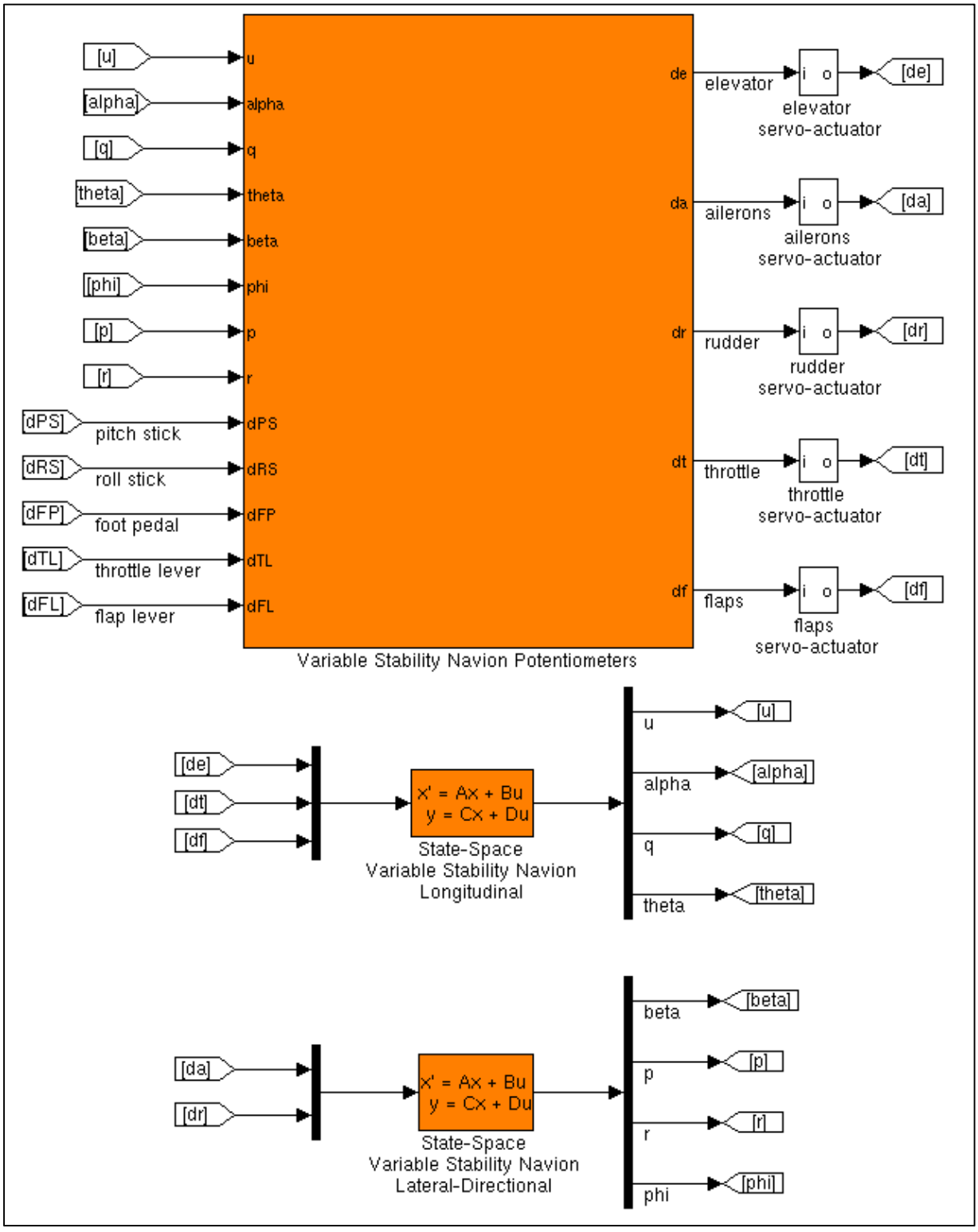


Figure E.1 Decoupled Linearized 3DOF Simulink Model

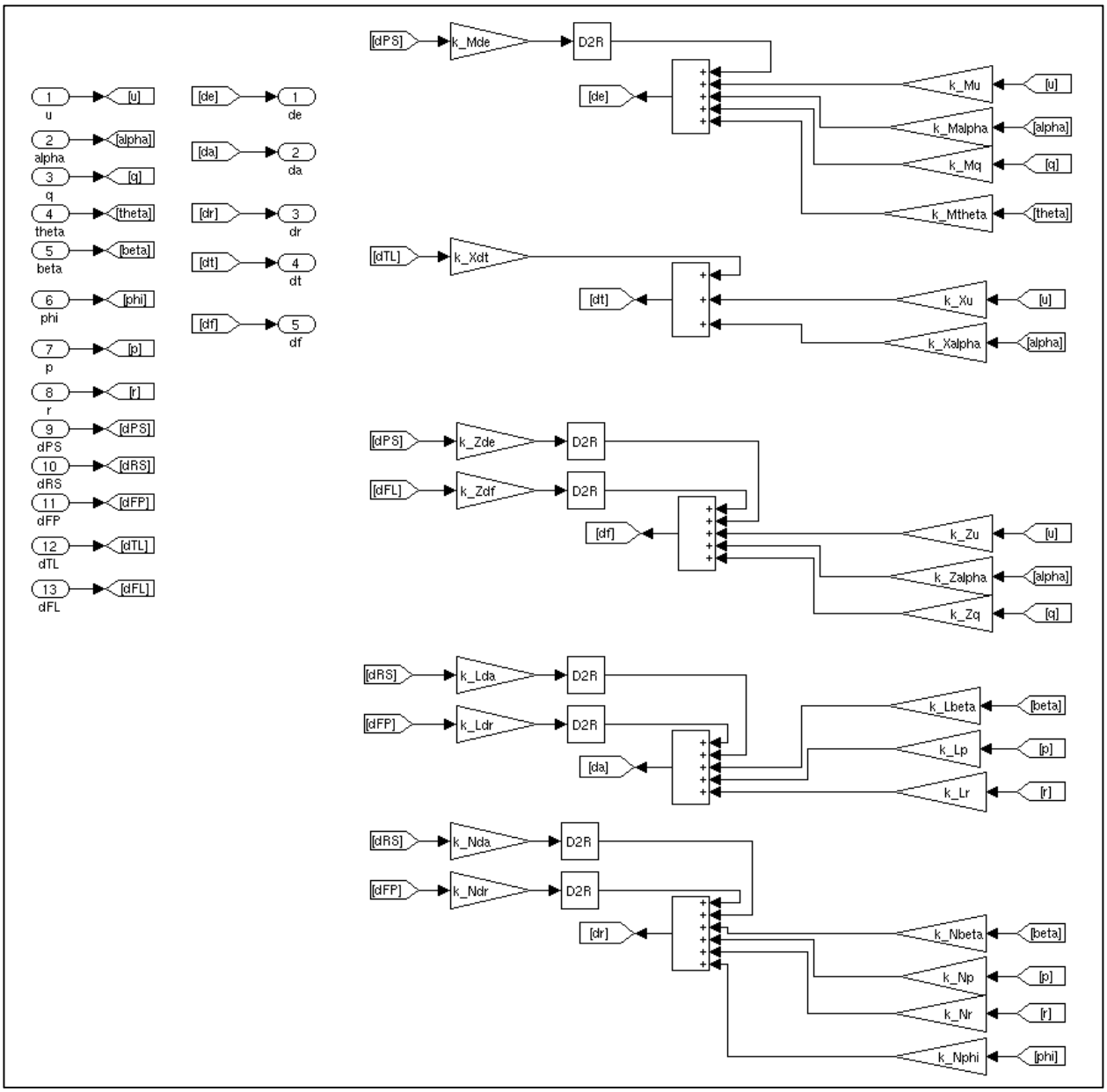


Figure E.2 Variable Stability Navion Potentiometers

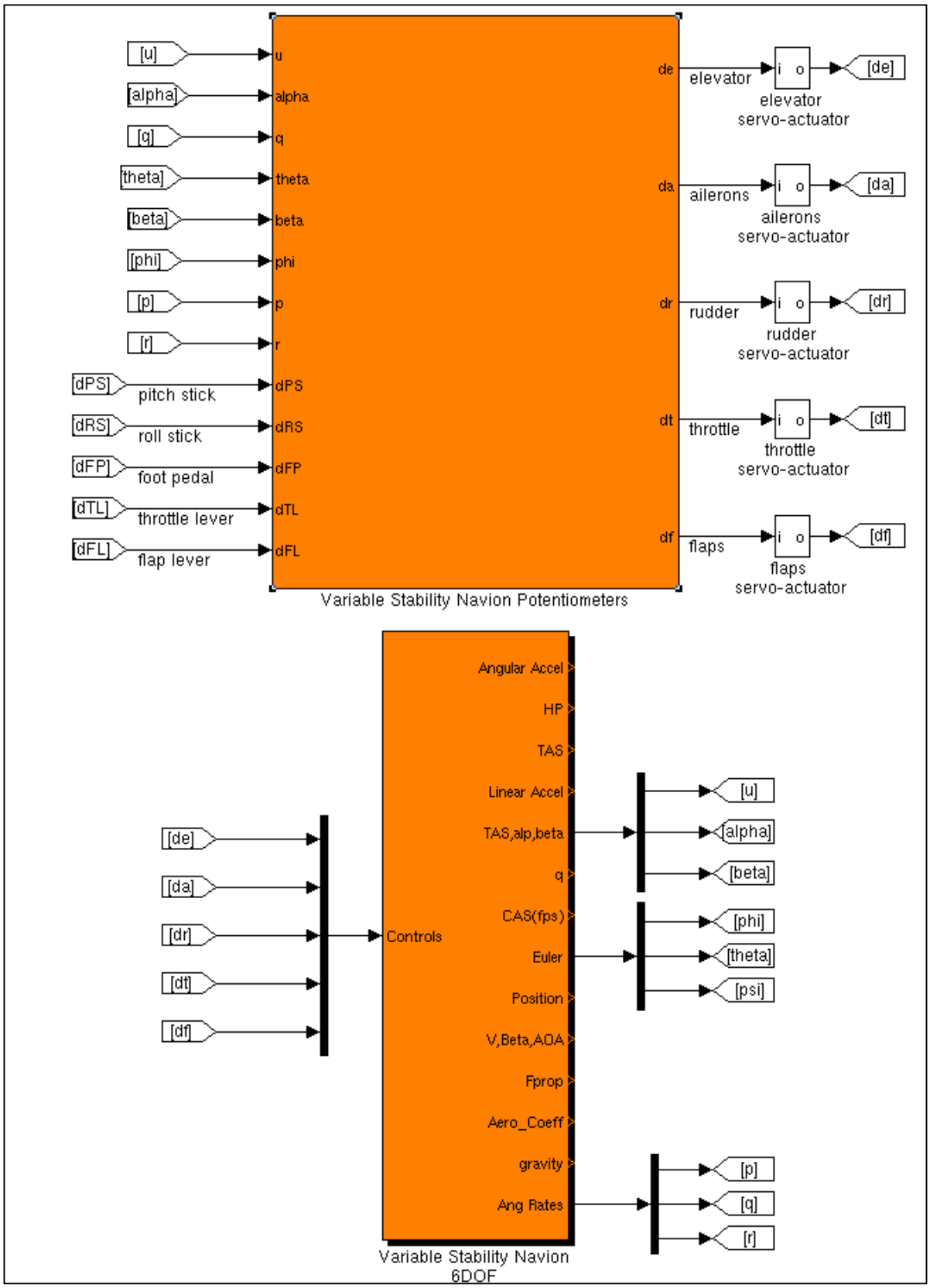


Figure E.3 6DOF Simulink Model

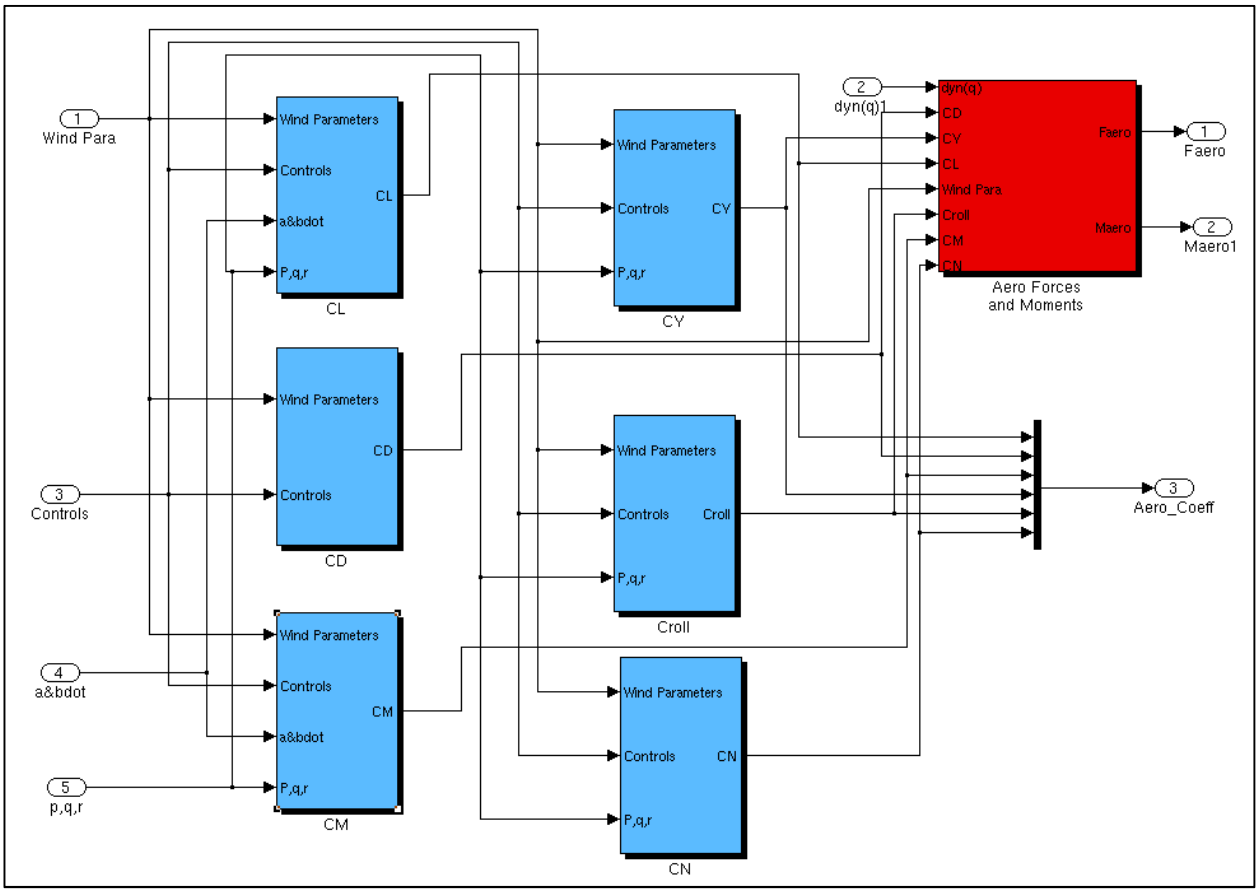


Figure E.4 6DOF Aerodynamics

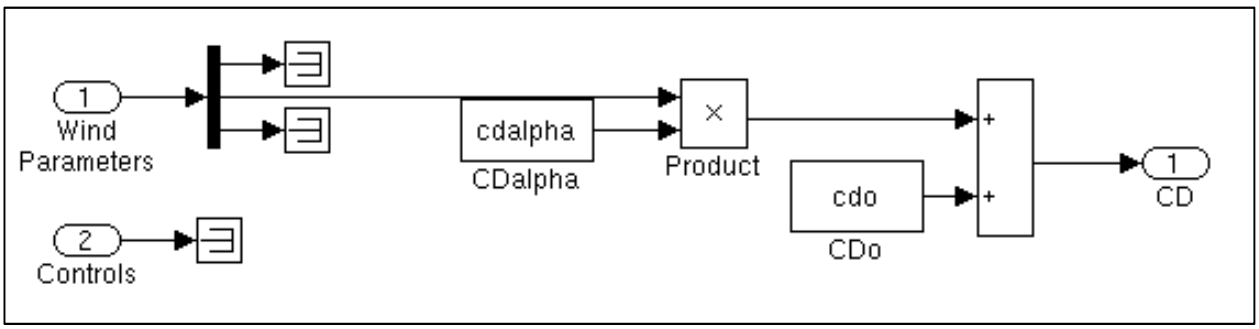


Figure E.5 $C_D$ Equation Block

# APPENDIX F
# MATLAB SCRIPTS

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% File Name: setup.m
% Joe Siu, UTSI, 5 March 2013
% Top level script to demonstrate Variable Stability Navion in-flight
% simulations
%
% Users are to "pick and choose" scripts to run by uncommenting required
% lines, and commenting out not required ones.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clc;
clear all;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Setup Variable Stability Navion (N66UT)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

run N66UT;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Setup Desired Aircraft to be simulated by N66UT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 run basic_Navion;     % run this to simulate basic Navion
% run TwinOtter;          % run this to simulate Twin Otter

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Feedback Gain Calculation Calculation Methods
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%% Longitudinal

% run long_implicit;                 % Term by term matching method
 run long_statespace;               % State-space method
% run long_approximation;            % Apporximation method
% run long_pole_placement;           % Pole Placement method


%%%% Lateral-Directional

% run latdir_implicit;               % Term by term matching method
% run latdir_implicit_pairs;         % Pair matching method
% run latdir_statespace;             % State-space method
% run latdir_approximation;          % Approximation method
 run latdir_pole_placement;         % Pole Placement Method

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Potentiometer setup
% Using 1981 ARA Calibration Data Manual
% Sign convention: TED, stick FWD, throttle FWD are POSITIVE
%                       stick LEFT, pedal LEFT are POSITIVE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% run potentiometers;   % only longitudinal available, not required for
                        % simulink because it takes pure feedback gains
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Trim aircraft to straight and level
% Two models to choose from: 3DOF or 6DOF
% Returns initial throttle and pitch stick settings
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% run N66UT_3DOF_trim;      % trim out Navion UT66
% run desired_3DOF_trim;    % trim out desired aircraft (Twin Otter)

 run N66UT_6DOF_trim;       % trim out Navion UT66
 run desired_6DOF_trim;     % trim out desired aircraft (Twin Otter)
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% File name: N66UT.m
% Joe Siu, UTSI, 5 March 2013
% Setup Variable Stability Navion Stability Derivatives and initial
% conditions
%
% Non-dimensionalized derivatives CL, CD, CM, CY, CL, and CN are from
% Nelson 2nd ed textbook, p.400.
%
% Calculation of dimensionalized derivatives are based on Nelson
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Mass characteristics and reference geometry
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

g=32.17;    % ft/s^2, gravitational constant
we=7.2921150e-5;    % earth's angular velocity in radians
sw=184;     % ft^2, wing area
b=33.4;     % ft, wing span
cbar = 5.7; % ft, mean chord length
weight = 2750;  % lbf
mass = weight/g;% slugs
Iyy=3000;    % sl-ft^2, Iyy
Ixx=1048;   % sl-ft^2
Izz=3530;   % sl-ft^2
Ixz=0;      % sl-ft^2

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Initial trimmed conditions for straight and level flight
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Trim velocity components
initu=176;  % ft/s (104 KCAS) Assume IAS=CAS=EAS, Do not use TAS
initv=0;    % ft/s
initw=0;    % ft/s (0 for stability axis)
initVtrue=176;  % ft/s

%%% Initial angular rates
initp=0;    % rad/s, roll rate
initq=0;    % rad/s, pitch rate
initr=0;    % rad/s, yaw rate

%%% Initial attitude
inittheta=0;% rad, pitch
initphi=0;  % rad, bank angle
initpsi=0;  % rad, yaw angle
initalpha=0;% rad, AoA
initbeta=0;  % rad, sideslip

%%% Initial Position
initx=0;    % ft
inith=0;    % ft
initnorth=0;
initeast=(-122*pi)/180;
initdown=0; % ft

rho0=0.002377;      % slug/ft^-3, standard sea level density
Q=0.5*rho0*initu^2; % lbf/ft^2, dynamic pressure

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Aerodynamic Stability Coefficients (Nelson, P.400)
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% CL (Lift)
clalpha=4.44;
clalphadot=0;
clq=3.8;
clde=0.355;
clo=0.41;
clu=0;

%%% CD (Drag)
cdalpha=0.33;
cdo=0.05;
cdu=0;

%%% CM (Pitching Moment)
cmalpha=-0.683;
cmdele=-0.923;
cmalphadot=-4.36;
cmq=-9.96;

%%% CY (Side-Force)
cybeta=-0.564;
cydr=0.157;
cyp=0;
cyr=0;
cyda=0;

%%% Cl (Rolling Moment)
clbeta=-0.074;
clp=-0.410;
clr=0.107;
clda=-0.134;
cldr=0.107;

%%% CN (Yawing Moment)
cnbeta=0.071;
cnp=-0.0575;
cnr=-0.125;
cndr=-0.072;
cnda=-0.0035;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Dimensionalized stability derivatives
% Sign convention: TED, stick FWD, throttle FWD are POSITIVE
%                       stick LEFT, pedal LEFT are POSITIVE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% X derivatives
Xu=-(cdu+2*cdo)*sw*Q/(initu*mass);
Xalpha=-(cdalpha-clo)*Q*sw/mass;
Xw=Xalpha/initu;
Xde=0;
Xdt=3.59; %(ft/s^2)/cm
Xdf=0;

%%% Z derivatives
Zu=-(clu+2*clo)*Q*sw/(initu*mass);
Zalpha=-(clalpha+cdo)*Q*sw/mass;
Zw=Zalpha/initu;
Zalphadot=-clalphadot*cbar*Q*sw/(2*initu*mass);
Zwdot=Zalphadot/initu;
```

```
Zq=-clq*cbar*Q*sw/(2*initu*mass);
Zde=-clde*Q*sw/mass;
Zdt=0;
Zdf=-70.9;  %(ft/s^2)/rad

%%% M derivatives
Mu=0;
Malpha=cmalpha*Q*sw*cbar/Iyy;
Malphadot=cmalphadot*cbar*Q*sw*cbar/(2*initu*Iyy);
Mw=Malpha/initu;
Mwdot=Malphadot/initu;
Mq=cmq*cbar*Q*sw*cbar/(2*initu*Iyy);
Mde=cmdele*Q*sw*cbar/Iyy;
Mdt=0;
Mdf=0;
Mtheta=0;

%%% Y derivatives
Ybeta=Q*sw*cybeta/mass;
Yv=Ybeta/initu;
Yp=Q*sw*b*cyp/(2*mass*initu);
Yr=Q*sw*b*cyr/(2*mass*initu);
Yda=Q*sw*cyda/mass;
Ydr=Q*sw*cydr/mass;

%%% L derivatives
Lbeta=Q*sw*b*clbeta/Ixx;
Lv=Lbeta/initu;
Lp=Q*sw*b^2*clp/(2*Ixx*initu);
Lr=Q*sw*b^2*clr/(2*Ixx*initu);
Lda=Q*sw*b*clda/Ixx;
Ldr=Q*sw*b*cldr/Ixx;

%%% N derivatives
Nbeta=Q*sw*b*cnbeta/Izz;
Nv=Nbeta/initu;
Np=Q*sw*b^2*cnp/(2*Izz*initu);
Nr=Q*sw*b^2*cnr/(2*Izz*initu);
Nda=Q*sw*b*cnda/Izz;
Ndr=Q*sw*b*cndr/Izz;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Longitudinal State-Space
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

A=[ Xu          Xalpha          0           -g;
    Zu/initu    Zalpha/initu    1+Zq/initu  0;
    Mu+Malphadot*Zu/initu  Malpha+Malphadot*Zalpha/initu...
    Mq+Malphadot*(1+Zq/initu)   0;
    0           0               1           0];

B=[0                        Xdt     0;
   Zde/initu                0       Zdf/initu;
   Mde+Malphadot*Zde/initu  0       Malphadot*Zdf/initu;
   0                        0       0];

C=eye(4);

D=zeros(4,3);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Lateral-Directional State-Space
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

keppa=1/(1-Ixz^2/(Ixx*Izz));

A_LD=[Ybeta/initu   Yp/initu   Yr/initu-1       +g/initu;
    (Lbeta+Nbeta*Ixz/Ixx)*keppa   (Lp+Np*Ixz/Ixx)*keppa
(Lr+Nr*Ixz/Ixx)*keppa  0;
    (Nbeta+Lbeta*Ixz/Izz)*keppa   (Np+Lp*Ixz/Izz)*keppa
(Nr+Lr*Ixz/Izz)*keppa  0;
        0           1           0                0];

B_LD=[0                         Ydr/initu;
     (Lda+Nda*Ixz/Ixx)*keppa  (Ldr+Ndr*Ixz/Ixx)*keppa   ;
     (Nda+Lda*Ixz/Izz)*keppa  (Ndr+Ldr*Ixz/Izz)*keppa   ;
     0                        0   ];

C_LD=eye(4);

D_LD=zeros(4,2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Cockpit control to surfaces gearing
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Longitudinal
stick_gearing=48.3/29.1; %full range elevator/full range stick (deg/deg)
throttle_gearing=9.4/10.1; %full throttle/full throttle lever fwd (cm/cm)
flap_gearing=20/2; %full 20 deg flaps down/2 units flap lever up

%%% Lateral-Directional
wheel_gearing=37.6/160; %full range aileron/full range wheel (deg/deg)
pedal_gearing=45/4.88; %full range rudder/full range pedal (deg/inch)
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% File name: TwinOtter.m
% Joe Siu, UTSI, 5 March 2013
% Setup Twin Otter Stability Derivatives and initial conditions
%
% Non-dimensionalized derivatives CL, CD, CM, CY, CL, and CN are from
% NASA Flight test
%
% Calculation of dimensionalized derivatives are based on Nelson
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Mass characteristics and reference geometry
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

des_sw=422.5;   % ft^2, wing area
des_b=65;       % ft, wing span
des_cbar = 6.5; % ft, mean chord length
des_weight = 9149;  % lbf
des_mass = des_weight/32.2; % slugs

des_Iyy=22126;  % sl-ft^2
des_Ixx=15694;  % sl-ft^2
des_Izz=33179;  % sl-ft^2
des_Ixz=1056;   % sl-ft^2

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Initial trimmed conditions for straight and level flight
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Trim velocity components
des_initu=176;  % ft/s (104 KCAS) Assume IAS=CAS=EAS, Do not use TAS
des_initv=0;    % ft/s
des_initw=0;    % ft/s (0 for stability axis)
des_initVtrue= 176;  % ft/s

%%% Initial angular rates
des_initp=0;    % rad/s, roll rate
des_initq=0;    % rad/s, pitch rate
des_initr=0;    % rad/s, yaw rate

%%% Initial attitude
des_initttheta=0;% rad, pitch
des_initphi=0;  % rad, bank angle
des_initpsi=0;  % rad, yaw angle
des_initalpha=0;% rad, AoA
des_initbeta=0;  % rad, sideslip

%%% Initial Position
des_initx=0;    % ft
des_inith=0;    % ft
des_initnorth=0;
des_initeast=(-122*pi)/180;
des_initdown=0; % ft

des_rho0=0.002377;      % slug/ft^-3, standard sea level density
des_Q=0.5*des_rho0*des_initu^2; % lbf/ft^2, dynamic pressure


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Aerodynamic Stability Coefficients (AIAA and NASA)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%% CL (Lift)
des_clalpha=5.66;
des_clalphadot=0;
des_clq=19.97;
des_clde=0.608;
des_clo=0.38;
des_clu=0;

%%% CD (Drag)
des_cdalpha=0.31;
des_cdo=0.041;
des_cdu=0;

%%% CM (Pitching Moment)
des_cmalpha=-1.31;
des_cmdele=-1.74;
des_cmalphadot=0;
des_cmq=-34.2;

%%% CY (Side-Force)
des_cybeta=-0.6;
des_cydr=0.15;
des_cyp=-0.2;
des_cyr=0.4;
des_cyda=0;

%%% Cl (Rolling Moment)
des_clbeta=-0.08;
des_clp=-0.5;
des_clr=0.06;
des_clda=-0.15;
des_cldr=0.015;

%%% CN (Yawing Moment)
des_cnbeta=0.1;
des_cnp=-0.06;
des_cnr=-0.18;
des_cndr=-0.12;
des_cnda=-0.001;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Dimensionalized stability derivatives
% Sign convention: TED is POSITIVE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% X derivatives
des_Xu=-(des_cdu+2*des_cdo)*des_sw*des_Q/(des_initu*des_mass);
des_Xalpha=-(des_cdalpha-des_clo)*des_Q*des_sw/des_mass;
des_Xw=des_Xalpha/des_initu;
des_Xdt=3.59;  % not available, use basic Navion's
des_Xdf=0;
des_Xde=0;

%%% Z derivatives
des_Zu=-(des_clu+2*des_clo)*des_Q*des_sw/(des_initu*des_mass);
des_Zalpha=-(des_clalpha+des_cdo)*des_Q*des_sw/des_mass;
des_Zw=des_Zalpha/des_initu;
des_Zalphadot=-des_clalphadot*des_cbar*des_Q*des_sw/(2*des_initu*des_mass);
des_Zwdot=des_Zalphadot/des_initu;
des_Zq=-des_clq*des_cbar*des_Q*des_sw/(2*des_initu*des_mass);
des_Zde=-des_clde*des_Q*des_sw/des_mass;
```

```
des_Zdf=-70.9;  %(ft/s^2)/rad, not available, use basic Navion's
des_Zdt=0;

%%% M derivatives
des_Mu=0;
des_Malpha=des_cmalpha*des_Q*des_sw*des_cbar/des_Iyy;
des_Mw=des_Malpha/des_initu;
des_Malphadot=des_cmalphadot*des_cbar*des_Q*des_sw*des_cbar/(2*des_initu*des_I
yy);
des_Mwdot=des_Malphadot/des_initu;
des_Mq=des_cmq*des_cbar*des_Q*des_sw*des_cbar/(2*des_initu*des_Iyy);
des_Mde=des_cmdele*des_Q*des_sw*des_cbar/des_Iyy;
des_Mdt=0;
des_Mdf=0;
des_Mtheta=0;

%%% Y derivatives
des_Ybeta=des_Q*des_sw*des_cybeta/des_mass;
des_Yv=des_Ybeta/des_initu;
des_Yp=des_Q*des_sw*des_b*des_cyp/(2*des_mass*des_initu);
des_Yr=des_Q*des_sw*des_b*des_cyr/(2*des_mass*des_initu);
des_Yda=des_Q*des_sw*des_cyda/des_mass;
des_Ydr=des_Q*des_sw*des_cydr/des_mass;

%%% L derivatives
des_Lbeta=des_Q*des_sw*des_b*des_clbeta/des_Ixx;
des_Lv=des_Lbeta/des_initu;
des_Lp=des_Q*des_sw*des_b^2*des_clp/(2*des_Ixx*des_initu);
des_Lr=des_Q*des_sw*des_b^2*des_clr/(2*des_Ixx*des_initu);
des_Lda=des_Q*des_sw*des_b*des_clda/des_Ixx;
des_Ldr=des_Q*des_sw*des_b*des_cldr/des_Ixx;

%%% N derivatives
des_Nbeta=des_Q*des_sw*des_b*des_cnbeta/des_Izz;
des_Nv=des_Nbeta/des_initu;
des_Np=des_Q*des_sw*des_b^2*des_cnp/(2*des_Izz*des_initu);
des_Nr=des_Q*des_sw*des_b^2*des_cnr/(2*des_Izz*des_initu);
des_Nda=des_Q*des_sw*des_b*des_cnda/des_Izz;
des_Ndr=des_Q*des_sw*des_b*des_cndr/des_Izz;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Longitudinal State-Space
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

des_A=[ des_Xu   des_Xalpha   0         -g;
    des_Zu/des_initu   des_Zalpha/des_initu   1+des_Zq/des_initu  0;
    des_Mu+des_Malphadot*des_Zu/des_initu
des_Malpha+des_Malphadot*des_Zalpha/des_initu
des_Mq+des_Malphadot*(1+des_Zq/des_initu)   0;
    0    0    1      0];

des_B=[0    des_Xdt  0;
   des_Zde/des_initu  0    des_Zdf/des_initu;
   des_Mde+des_Malphadot*des_Zde/des_initu  0
des_Malphadot*des_Zdf/des_initu;
   0    0    0];

des_C=eye(4);

des_D=zeros(4,3);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Lateral-Directional State-Space
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

des_keppa=1/(1-des_Ixz^2/(des_Ixx*des_Izz));

des_A_LD=[des_Ybeta/des_initu   des_Yp/des_initu   des_Yr/des_initu-1
g/des_initu;
(des_Lbeta+des_Nbeta*des_Ixz/des_Ixx)*des_keppa
(des_Lp+des_Np*des_Ixz/des_Ixx)*des_keppa
(des_Lr+des_Nr*des_Ixz/des_Ixx)*des_keppa  0;
(des_Nbeta+des_Lbeta*des_Ixz/des_Izz)*des_keppa
(des_Np+des_Lp*des_Ixz/des_Izz)*des_keppa
(des_Nr+des_Lr*des_Ixz/des_Izz)*des_keppa  0;
0    1    0      0];

des_B_LD=[0    des_Ydr/des_initu;
   (des_Lda+des_Nda*des_Ixz/des_Ixx)*des_keppa
(des_Ldr+des_Ndr*des_Ixz/des_Ixx)*des_keppa   ;
   (des_Nda+des_Lda*des_Ixz/des_Izz)*des_keppa
(des_Ndr+des_Ldr*des_Ixz/des_Izz)*des_keppa   ;
   0    0   ];

des_C_LD=eye(4);

des_D_LD=zeros(4,2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Cockpit control to surfaces gearing
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Longitudinal
des_stick_gearing=48.3/29.1; %full range elevator/full range stick (deg/deg)
des_throttle_gearing=9.4/10.1; %full throttle fwd/full throttle lever fwd
(cm/cm)
des_flap_gearing=20/2; %full 20 deg flaps down/2 units flap lever up

%%% Lateral-Directional
des_wheel_gearing=37.6/160; %full range aileron/full range wheel (deg/deg)
des_pedal_gearing=45/4.88; %full range rudder/full range pedal (deg/inch)
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% File name: long_implicit.m
% Joe Siu, UTSI, 5 March 2013
% Longitudinal Implicit model following (term-by-term method)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Control Gain Feedbacks setup
% Elevator (de) to match pitch rate (q_dot)
% Throttle (dt) to match longitudinal acceleration (u_dot)
% Direct Lift Flaps (DL) to match normal acceleration (w_dot)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Feedback to Throttle
k_Xalpha=((des_Xalpha-Xalpha)/Xdt);
k_Xw=(des_Xw-Xw)/Xdt;
k_Xde=(des_Xde-Xde)/Xdt;
k_Xdt=(des_Xdt-Xdt)*throttle_gearing/Xdt+throttle_gearing;
k_Xdf=(des_Xdf-Xdf)/Xdt;
k_Xu=((des_Xu-Xu)/Xdt);

%%% Feedback to Elevator
k_Malpha=(des_Malpha-Malpha)/Mde;
k_Mw=(des_Mw-Mw)/Mde;
k_Mq=(des_Mq-Mq)/Mde;
k_Mde=des_Mde*stick_gearing/Mde;
k_Mdt=(des_Mdt-Mdt)/Mde;
k_Mdf=(des_Mdf-Mdf)/Mde;
k_Mtheta=(des_Mtheta-Mtheta)/Mde;
k_Mu=((des_Mu-Mu)/Mde);

%%% Feedback to Flaps
k_Zalpha=(des_Zalpha-Zalpha-Zde*k_Malpha)/Zdf;
k_Zw=(des_Zw-Zw-Zde*k_Mw)/Zdf;
k_Zdt=(des_Zdt-Zdt)/Zdf;
k_Zdf=(des_Zdf-Zdf)*flap_gearing/Zdf+flap_gearing;
k_Zu=((des_Zu-Zu-Zde*k_Mu)/Zdf);
k_Zde=(des_Zde-Zde)*stick_gearing/Zdf;
k_Zq=(des_Zq-Zq)/Zdf;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% File name: long_statespace.m
% Joe Siu, UTSI, 5 March 2013
% Longitudinal State-Space Method
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Rebuild des_A_ matrix because Approximation method needs it
des_A2=[ des_Xu   des_Xalpha   0         -g;
    des_Zu/des_initu   des_Zalpha/des_initu   1+des_Zq/des_initu  0;
    des_Mu+des_Malphadot*des_Zu/des_initu
des_Malpha+des_Malphadot*des_Zalpha/des_initu
des_Mq+des_Malphadot*(1+des_Zq/des_initu)   0;
    0    0    1      0];

%%% Calculate Gain Matrix
K=B\(des_A2-A); % because des_A2=A+B*K for positive feedback

k_Mu=K(1,1);
k_Malpha=K(1,2);
k_Mq=K(1,3);
k_Mtheta=K(1,4);

k_Xu=K(2,1);
k_Xalpha=K(2,2);

k_Zu=K(3,1);
k_Zalpha=K(3,2);
k_Zq=K(3,3);

% aug_A=A+B*K;  % Augmented A Matrix

%%% Controls
syms k_Mde k_Zde
S =
solve((des_Mde+des_Malphadot*des_Zde/des_initu)*stick_gearing==(Mde+Malphadot*
Zde/initu)*(k_Mde)+(Malphadot*Zdf/initu)*(k_Zde),...

des_Zde/des_initu*stick_gearing==(Zde/initu)*(k_Mde)+Zdf/initu*(k_Zde));
k_Mde=double(S.k_Mde);
k_Zde=double(S.k_Zde);

k_Xdt=(des_Xdt-Xdt)*throttle_gearing/Xdt+throttle_gearing;

k_Zdf=(des_Zdf-Zdf)*flap_gearing/Zdf+flap_gearing;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% File name: long_approximation.m
% Joe Siu, UTSI, 5 March 2013
% Short Period Oscillation (SPO) and Phugoid Estimation
% Equations from Nelson p.155
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Basic Navion's Longitudinal Approximations

spo_omega=sqrt(Zalpha*Mq/initu-Malpha);  % rad/s
spo_zeta=-(Mq+Zalphadot+Zalpha/initu)/(2*spo_omega);
ph_omega= sqrt(-Zu*g/initu);  % rad/s
ph_zeta=-Xu/(2*ph_omega);

%%% Desired Aircraft

des_spo_omega=sqrt(des_Zalpha*des_Mq/des_initu-des_Malpha);
des_spo_zeta=-(des_Mq+des_Zalphadot+des_Zalpha/des_initu)/(2*des_spo_omega);
des_ph_omega=sqrt(-des_Zu*g/des_initu);
des_ph_zeta=-des_Xu/(2*des_ph_omega);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calspan Learjet Training Syllabus Settings (p.2-24)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%  des_spo_omega=4; % rad/s
%  des_spo_zeta=-2;
%  des_ph_omega=0.0839; % rad/s
%  des_ph_zeta=0.05;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Use feedback gains to simulate desired SPO and Phugoid
% Rearrange the approximation equations (Nelson p.155)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

des_Mq=-2*des_spo_zeta*des_spo_omega-Malphadot-Zalpha/initu;
des_Malpha=Zalpha*des_Mq/initu-des_spo_omega^2;
des_Zu=-initu*des_ph_omega^2/g;
des_Xu=-2*des_ph_zeta*des_ph_omega;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% For Simulink Long_3DOF_linear_model use:
% The des_A matrix is still the desired aircraft, for comparison
% Required to reset all the potentiometers back to basic Navion, except the
% estimated ones (Mq, Malpha, Zu, Xu), and controls (Mde, Zde)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% X derivatives
des_Xalpha=Xalpha;

%%% Z derivatives
des_Zalpha=Zalpha;
des_Zalphadot=Zalphadot;
des_Zq=Zq;

%%% M derivatives
des_Mu=Mu;
des_Malphadot=Malphadot;
```

```
des_initu=initu;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Control Gain Feedbacks setup
% Using Statespace method for most accurate results
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

run long_statespace;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% File name: long_pole_placement.m
% Joe Siu, UTSI, 5 March 2013
% Short Period Oscillation (SPO) and Phugoid
% Pole Placement Method
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Control Matrix only uses elevator. Throttle and Flaps are untouched
B2=[0    0  0;
   Zde/initu  0    0;
   Mde+Malphadot*Zde/initu  0  0;
   0    0    0];

%%% Assemble the desired poles
[omega,zeta,P]=damp(des_A);  % find the spo and phugoid poles of desired
aircraft

%%% draw the poles on complex plane
% plot(P,'r*');
% sgrid;

%%% The required gain feedback matrix
%%% ATTENTION!!! MATLAB place() uses negative feedbacks

K=place(A,B2,P); % for desired aircraft


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calspan Learjet Training Syllabus Settings
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

spo_zeta=0.7;   %Required SP damping ratio
spo_omega=4;    %Required SP natural frequency rads/sec
ph_zeta=0.260;   %Required Phugoid damping ratio
ph_T=25;       %Required phugoid damped period


spo_omega_damp=spo_omega*sqrt(1-spo_zeta^2);%SP damped frequency
ph_omega_damp=(2*pi)/ph_T; %Required Phugoid damped frequency rads/sec
ph_omega=ph_omega_damp/sqrt(1-ph_zeta^2);%Phugoid natural frequency

%%% Assemble Longitudinal the poles
p1=-spo_zeta*spo_omega+spo_omega_damp*i;
p2=-spo_zeta*spo_omega-spo_omega_damp*i;
p3=-ph_zeta*ph_omega+ph_omega_damp*i;
p4=-ph_zeta*ph_omega-ph_omega_damp*i;

P=[p1; p2; p3; p4];

%%% The required gain feedback matrix
%%% ATTENTION!!! MATLAB place() uses negative feedbacks

 K=place(A,B2,P);  % for Calspan Sylabus


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Gain Feedbacks setup
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Controls
syms k_Mde k_Zde
```

```
S =
solve((des_Mde+des_Malphadot*des_Zde/des_initu)*stick_gearing==(Mde+Malphadot*
Zde/initu)*(k_Mde)+(Malphadot*Zdf/initu)*(k_Zde),...

des_Zde/des_initu*stick_gearing==(Zde/initu)*(k_Mde)+Zdf/initu*(k_Zde));
k_Mde=double(S.k_Mde);
k_Zde=double(S.k_Zde);

k_Xdt=(des_Xdt-Xdt)*throttle_gearing/Xdt+throttle_gearing;

k_Zdf=(des_Zdf-Zdf)*flap_gearing/Zdf+flap_gearing;


%%% stability
K=-K; % to turn negative feedback to positive feedback

%%% Feedback to Elevator
k_Mu=K(1,1);       % negative in front to flip back to positive feedback
k_Malpha=K(1,2);
k_Mq=K(1,3);
k_Mtheta=K(1,4);

%%% Feedback to Throttle (should be all zeros)
k_Xu=K(2,1);       % negative in front to flip back to positive feedback
k_Xalpha=K(2,2);
k_Xq=K(2,3);
k_Xtheta=K(2,4);

%%% Feedback to Flaps (should be all zeros)
k_Zu=K(3,1);       % negative in front to flip back to positive feedback
k_Zalpha=K(3,2);
k_Zq=K(3,3);
k_Ztheta=K(3,4);


%%% The augmented A matrix  (Nelson p.371)
%%% check: damp(des_A) = damp(A_aug) due to gain feedback
A_aug=A+B2*K;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% File name: latdir_implicit.m
% Joe Siu, UTSI, 5 March 2013
% Lateral-directional Implicit model following (term-by-term method)
% This method is not as accurate as other methods because it ignores cross
% coupling effects
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Control Gain Feedbacks setup
% Ailerons (da) to match roll rate (p_dot)
% Rudder (dr) to match yaw rate (r_dot)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Simplified by setting Ixz=0

%%% Feedback to Ailerons
k_Lbeta=(des_Lbeta-Lbeta)/Lda;
k_Lp=(des_Lp-Lp)/Lda;
k_Lr=(des_Lr-Lr)/Lda;
k_Lphi=0;
k_Ldr=(des_Ldr-Ldr)*pedal_gearing/Lda;
k_Lda=(des_Lda-Lda)*wheel_gearing/Lda+wheel_gearing;


%%% Feedback to Rudder
k_Nbeta=(des_Nbeta-Nbeta)/Ndr;
k_Np=(des_Np-Np)/Ndr;
k_Nr=(des_Nr-Nr)/Ndr;
k_Nphi=0;
k_Ndr=(des_Ndr-Ndr)*pedal_gearing/Ndr+pedal_gearing;
k_Nda=(des_Nda-Nda)*wheel_gearing/Ndr;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% File name: latdir_implicit_pairs.m
% Joe Siu, UTSI, 5 March 2013
% Lateral-directional Implicit model following (pairs method)
% Systems of two equations with two variables
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Simplified by setting Ixz=0

syms k_Lbeta k_Nbeta
S =
solve(des_Lbeta==Lbeta+Lda*(k_Lbeta)+Ldr*(k_Nbeta),des_Nbeta==Nbeta+Nda*(k_Lbe
ta)+Ndr*(k_Nbeta));
k_Lbeta=double(S.k_Lbeta);
k_Nbeta=double(S.k_Nbeta);

syms k_Lp k_Np
S = solve(des_Lp==Lp+Lda*(k_Lp)+Ldr*(k_Np),des_Np==Np+Nda*(k_Lp)+Ndr*(k_Np));
k_Lp=double(S.k_Lp);
k_Np=double(S.k_Np);

syms k_Lr k_Nr
S = solve(des_Lr==Lr+Lda*(k_Lr)+Ldr*(k_Nr),des_Nr==Nr+Nda*(k_Lr)+Ndr*(k_Nr));
k_Lr=double(S.k_Lr);
k_Nr=double(S.k_Nr);

k_Lphi=0;
k_Nphi=0;

%%% controls
syms k_Lda k_Nda
S =
solve(des_Lda*wheel_gearing==Lda*(k_Lda)+Ldr*(k_Nda),des_Nda*wheel_gearing==Nd
a*(k_Lda)+Ndr*(k_Nda));
k_Lda=double(S.k_Lda);
k_Nda=double(S.k_Nda);

syms k_Ldr k_Ndr
S =
solve(des_Ldr*pedal_gearing==Ldr*(k_Ndr)+Lda*(k_Ldr),des_Ndr*pedal_gearing==Nd
r*(k_Ndr)+Nda*(k_Ldr));
k_Ldr=double(S.k_Ldr);
k_Ndr=double(S.k_Ndr);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% File name: latdir_statespace.m
% Joe Siu, UTSI, 5 March 2013
% Lateral-Directional State-Space Method
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Rebuild des_A_LD matrix because Approximation method needs it
des_A_LD2=[des_Ybeta/des_initu   des_Yp/des_initu   des_Yr/des_initu-1
g/des_initu;
(des_Lbeta+des_Nbeta*des_Ixz/des_Ixx)*des_keppa
(des_Lp+des_Np*des_Ixz/des_Ixx)*des_keppa
(des_Lr+des_Nr*des_Ixz/des_Ixx)*des_keppa  0;
(des_Nbeta+des_Lbeta*des_Ixz/des_Izz)*des_keppa
(des_Np+des_Lp*des_Ixz/des_Izz)*des_keppa
(des_Nr+des_Lr*des_Ixz/des_Izz)*des_keppa  0;
0    1    0      0];

%%% Calculate Gain Matrix
K_LD=B_LD\(des_A_LD2-A_LD);     % for positive feedback

k_Lbeta=K_LD(1,1);
k_Lp=K_LD(1,2);
k_Lr=K_LD(1,3);
k_Lphi=K_LD(1,4);

k_Nbeta=K_LD(2,1);
k_Np=K_LD(2,2);
k_Nr=K_LD(2,3);
k_Nphi=K_LD(2,4);

% A_LD_aug=A_LD+B_LD*K_LD;      % augmented A_LD matrix

%%% controls (includes Ixz contribution)
syms k_Lda k_Nda
S = solve(((des_Lda+des_Nda*des_Ixz/des_Ixx)*des_keppa)*wheel_gearing== ...
    ((Lda+Nda*Ixz/Ixx)*keppa)*(k_Lda)+((Ldr+Ndr*Ixz/Ixx)*keppa)*(k_Nda)...
    ,((des_Nda+des_Lda*des_Ixz/des_Izz)*des_keppa)*wheel_gearing==...

(((Nda+Lda*Ixz/Izz)*keppa)*des_keppa)*(k_Lda)+((Ndr+Ldr*Ixz/Izz)*keppa)*(k_Nda
));
k_Lda=double(S.k_Lda);
k_Nda=double(S.k_Nda);


syms k_Ldr k_Ndr
S = solve(((des_Ldr+des_Ndr*des_Ixz/des_Ixx)*des_keppa)*pedal_gearing==...
    ((Ldr+Ndr*Ixz/Ixx)*keppa)*(k_Ndr)+((Lda+Nda*Ixz/Ixx)*keppa)*(k_Ldr)...
    ,((des_Ndr+des_Ldr*des_Ixz/des_Izz)*des_keppa)*pedal_gearing==...
    ((Ndr+Ldr*Ixz/Izz)*keppa)*(k_Ndr)+((Nda+Lda*Ixz/Izz)*keppa)*(k_Ldr));
k_Ldr=double(S.k_Ldr);
k_Ndr=double(S.k_Ndr);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% File name: latdir_approximation.m
% Joe Siu, UTSI, 11 March 2013
% Roll Mode, Spiral Roll, and Dutch Roll Estimation
% Equations from Nelson p.198
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Basic Navion's Lateral-Directional Approximations

taur=-1/Lp;  % seconds
tr=taur*log(2); % seconds, time to half of double
taus=Lbeta/(Lr*Nbeta-Lbeta*Nr); % seconds
ts=taus*log(2); % seconds, time to half of double
dr_omega=sqrt((Ybeta*Nr-Nbeta*Yr+initu*Nbeta)/initu); % rad/sec
dr_zeta=-1/(2*dr_omega)*((Ybeta+initu*Nr)/initu);

%%% Desired Aircraft

des_taur=-1/des_Lp;   % seconds
des_tr=des_taur*log(2); % seconds, time to half of double
des_taus=des_Lbeta/(des_Lr*des_Nbeta-des_Lbeta*des_Nr);  % seconds
des_ts=des_taus*log(2); % seconds, time to half of double
des_dr_omega=sqrt((des_Ybeta*des_Nr-
des_Nbeta*des_Yr+des_initu*des_Nbeta)/des_initu); % rad/sec
des_dr_zeta=-1/(2*des_dr_omega)*((des_Ybeta+des_initu*des_Nr)/des_initu);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calspan Learjet Training Syllabus Settings (p.3-28)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% des_taur=1.3; % sec
% des_taus=0;    % sec
% des_dr_omega=2; % rad/sec
% des_dr_zeta=-0.08;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Use feedback gains to simulate desired Dutch Roll, spiral and roll Modes
% Rearrange the approximation equations (Nelson p.198)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

des_Lp=-1/des_taur;     % for matching roll mode
des_Nr=(initu*2*des_dr_omega*des_dr_zeta+Ybeta)/(-initu); % for matching Dutch
roll
des_Nbeta=(Ybeta*des_Nr-initu*des_dr_omega^2)/(Yr-initu);   % for matching
Dutch roll
des_Lbeta=(des_taus*Lr*des_Nbeta)/(1+des_taus*des_Nr);  % for matching spiral
mode

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% For Simulink LatDir_3DOF_linear_model use:
% The des_A_LD matrix is still the desired aircraft, for comparison
% Required to reset all the potentiometers back to basic Navion, except the
% estimated ones (Lp,Lbeta,Nbeta,Nr), and controls (Yda, Ydr, Lda,Ldr,Nda,Ndr)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Y derivatives
des_Ybeta=Ybeta;
des_Yp=Yp;
des_Yr=Yr;

%%% L derivatives
des_Lr=Lr;
```

```
%%% N derivatives
des_Np=Np;

des_Iyy=Iyy;
des_Ixx=Ixx;
des_Izz=Izz;
des_Ixz=Ixz;
des_keppa=keppa;
des_initu=initu;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Control Gain Feedbacks setup
% Using Statespace method for most accurate results
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

run latdir_statespace;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% File name: latdir_pole_placement.m
% Joe Siu, UTSI, 5 March 2013
% Dutch Roll, Roll Mode, and Spiral Mode
% Pole Placement Method
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Control Matrix only uses rudder. Aileron is untouched
B_LD2=[ 0   Ydr/initu;
        0  (Ldr+Ndr*Ixz/Ixx)*keppa   ;
        0  (Ndr+Ldr*Ixz/Izz)*keppa   ;
        0   0   ];

%%% Assemble the desired poles
[omega,zeta,P_LD]=damp(des_A_LD);  % find the spo and phugoid poles of desired
aircraft

% draw the poles on complex plane
% plot(P_LD,'r*');
% sgrid;

%%% The required gain feedback matrix
%%% ATTENTION!!! MATLAB place() uses negative feedbacks

 K_LD=place(A_LD,B_LD2,P_LD); % for desired aircraft

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calspan Learjet Training Syllabus Settings
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

dr_zeta=0.4;  % Required DR damping ratio
dr_omega=2;  % Required DR natural frequency rad/s
r_tau=1/6.04;   % Required Roll mode time constant
sp_tau=10;    % Required Spiral mode time constant

dr_omega_damp=dr_omega*sqrt(1-dr_zeta^2);  % DR damped frequency

% Assemble Lateral-Directional Poles
p1_LD=-dr_zeta*dr_omega+dr_omega_damp*i;
p2_LD=-dr_zeta*dr_omega-dr_omega_damp*i;
p3_LD=-1/sp_tau;
p4_LD=-1/r_tau;

P_LD=[p1_LD; p2_LD; p3_LD; p4_LD];

%%% The required gain feedback matrix
%%% ATTENTION!!! MATLAB place() uses negative feedbacks

 K_LD=place(A_LD,B_LD2,P_LD);   % for Calspan Sylabus



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Gain Feedbacks setup
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% controls
syms k_Lda k_Nda
S = solve(((des_Lda+des_Nda*des_Ixz/des_Ixx)*des_keppa)*wheel_gearing== ...
    ((Lda+Nda*Ixz/Ixx)*keppa)*(k_Lda)+((Ldr+Ndr*Ixz/Ixx)*keppa)*(k_Nda)...
    ,((des_Nda+des_Lda*des_Ixz/des_Izz)*des_keppa)*wheel_gearing==...
```

```
(((Nda+Lda*Ixz/Izz)*keppa)*des_keppa)*(k_Lda)+((Ndr+Ldr*Ixz/Izz)*keppa)*(k_Nda
));
k_Lda=double(S.k_Lda);
k_Nda=double(S.k_Nda);


syms k_Ldr k_Ndr
S = solve(((des_Ldr+des_Ndr*des_Ixz/des_Ixx)*des_keppa)*pedal_gearing==...
    ((Ldr+Ndr*Ixz/Ixx)*keppa)*(k_Ndr)+((Lda+Nda*Ixz/Ixx)*keppa)*(k_Ldr)...
    ,((des_Ndr+des_Ldr*des_Ixz/des_Izz)*des_keppa)*pedal_gearing==...
    ((Ndr+Ldr*Ixz/Izz)*keppa)*(k_Ndr)+((Nda+Lda*Ixz/Izz)*keppa)*(k_Ldr));
k_Ldr=double(S.k_Ldr);
k_Ndr=double(S.k_Ndr);


%%% stability
K_LD=-K_LD;  % to turn negative feedback to positive feedback

%%% Feedback to Ailerons (should be all zeros)
k_Lbeta=K_LD(1,1); % negative in front to flip back to positive feedback
k_Lp=K_LD(1,2);
k_Lr=K_LD(1,3);
k_Lphi=K_LD(1,4);

%%% Feedback to Rudder
k_Nbeta=K_LD(2,1); % negative in front to flip back to positive feedback
k_Np=K_LD(2,2);
k_Nr=K_LD(2,3);
k_Nphi=K_LD(2,4);


%%% The augmented A matrix  (Nelson p.371)
%%% check: damp(des_A_LD) = damp(A_LD_aug) due to gain feedback
A_LD_aug=A_LD+B_LD2*K_LD;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% File name: potentiometers.m
% Joe Siu, UTSI, 5 March 2013
% Potentiometer setup
% Using 1981 ARA Calibration Data Manual
% Sign convention: TED, stick FWD, throttle FWD are POSITIVE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%% Only Longitudinal Potentiometer Calibrations are available
%%% Calibration curves are highly linear in y=mx+b form
%%% Unit conversion factors are included

p_Mu=(k_Mu*(180/pi)/(3600/6076)-0.00098)/(-0.00419);
p_Malpha=(k_Malpha+0.0254)/(-0.0124);
p_Mq=(k_Mq+0.0060)/(-0.0164);
p_Mtheta=(k_Mtheta-0.008)/(-0.0170);
p_Mde=(k_Mde+0.0055)/(-0.0236);
p_Xdt=(k_Xdt+0.0015)/-0.0184;
p_Xu=(k_Xu*6076/3600+0.0127)/(0.0155);
p_Xalpha=(k_Xalpha*pi/180-0.0020)/(0.0130);
p_Zalpha=(k_Zalpha-0.0050)/(-0.0567);
p_Zde=(k_Zde-0.0143)/0.0267;
p_ZDL=(k_ZDL+0.476)/0.526;
p_Zu=(k_Zu*(180/pi)/(3600/6076)+0.0181)/0.0285;
```

## VITA

Captain Joe Siu enrolled in the Royal Canadian Air Force in 1998 and graduated from the Royal Military College in 2002 with a bachelor degree in Computer Engineering. He attended the Canadian Forces School of Aerospace Technology and Engineering in 2003 to become qualified as an Aerospace Engineering Officer. In 2006, Captain Siu was selected for a year-long training program at the Empire Test Pilots' School, U.K. to become a Qualified Flight Test Engineer.

Captain Siu's military tours with the CF-18 Hornet include a posting in the CF-18 Weapon System Manager Detachment Mirabel, Quebec, in 2004, where he managed the air-to-air weapon software. In 2007, as a CF-18 Flight Test Engineer, he served in the Aerospace Engineering Test Establishment, Cold Lake, Alberta. Captain Siu has directed flight test programs such as the CF-18 Hornet modernization, Link-16 data system upgrade, landing gear prototype and landing technique evaluation, advanced multi-role infrared sensor developmental evaluation, flight control software upgrade, operational flight program software upgrade, enhanced precision guided bomb trial, and Airbus air refueling trial. Captain Siu has flown more than 500 hours in 25 different aircraft. Some aircraft to highlight are the F-18 Hornet, JAS-39 Gripen, Jaguar T2, Tornado F3, Mirage 2000D, Hawk, Tutor, Alpha Jet, Harvard, Tucano, E-3 Sentry AWACS, P-3 Orion, DC-3, and Airbus A380.

Outside the military, Captain Siu is a Professional Engineer in the Association of Professional Engineers and Geophysicists of Alberta. He enjoys playing ice-hockey, traveling, and is a volunteer firefighter for the City of Cold Lake.

Currently, Captain Siu is assigned to the University of Tennessee Space Institute to earn a Master's degree in Aviation Systems, and a Master's degree in Aerospace Engineering. He currently resides in Tullahoma, TN, with his wife, Marie-Michele, who is also a Royal Canadian Air Force officer pursuing her Master's degrees at UTSI.