

2002 Controls Design Challenge

R. A. Hess¹

T. K Vetter²

*Dept. of Mechanical and Aeronautical Engineering
University of California, Davis, CA*

S. R. Wells³

Dept. of Aeronautics

U.S. Air Force Academy, Colorado Springs, CO

Table of Contents

I. Challenge Statement

- A. Introduction
- B. Challenge Goal and Vehicle Description
- C. Design Challenge
 - 1. Desired Control System Response Types
 - 2. Controller Simulink[®] Digital Implementation
 - 3. Pilot Models
 - 4. Sensor Models
 - 5. Linear Models – Flight Conditions
 - 6. Piloting Tasks
 - 7. Performance/Handling Qualities Specifications
 - a. Desired Handling Qualities
 - b. Desired Tracking Performance
 - c. Desired Stability Margins
 - d. Structural Coupling Considerations
 - e. Control Activity
 - f. Scheduling Considerations – Off-nominal Flight Conditions
 - 8. Failure Modes
 - 9. MATLAB[®] Models and M-file for Simulink[®] Failure Simulation
 - 10. Design Challenge Synopsis
- D. Appendix I Listing for IceFunctionsDC.m
- E. Appendix II – Innovative Control Effector (ICE) Vehicle Description

¹ Professor, and Vice Chair

² Graduate Student

³ Assistant Professor

II. A Design Challenge Solution

A. Introduction

B. Solution

1. Controller Description
2. Solution Specifics
 - a. Handling Qualities
 - b. Tracking Performance
 - c. Stability Margins
 - d. Structural Coupling Considerations
 - e. Control Activity
 - f. Scheduling Considerations - Off-Nominal Flight Conditions
3. Discussion

C. Conclusions

A. Introduction

This document is intended to provide the specifications and requirements for a flight control system design challenge. The response to the challenge will involve documenting whether the particular design has met the stated requirements through analysis and computer simulation. The response should be written in the general format of a technical publication with corresponding length limits, e.g., an approximate maximum length of 45 units, with each full-size figure and double-spaced typewritten page constituting one unit.

B. Challenge Goal and Vehicle Description

The underlying goal of the challenge is the design and computer simulation of a high-performance piloted aircraft in a series of different flight conditions. Of particular importance to the design will be the ability of the aircraft to sustain significant system failures/damage, here to be represented by variations in control surface actuator characteristics and vehicle dynamics. The vehicle that is the subject of the challenge is the Innovative Control Effector (ICE) vehicle shown in Fig. I.1. A detailed description of

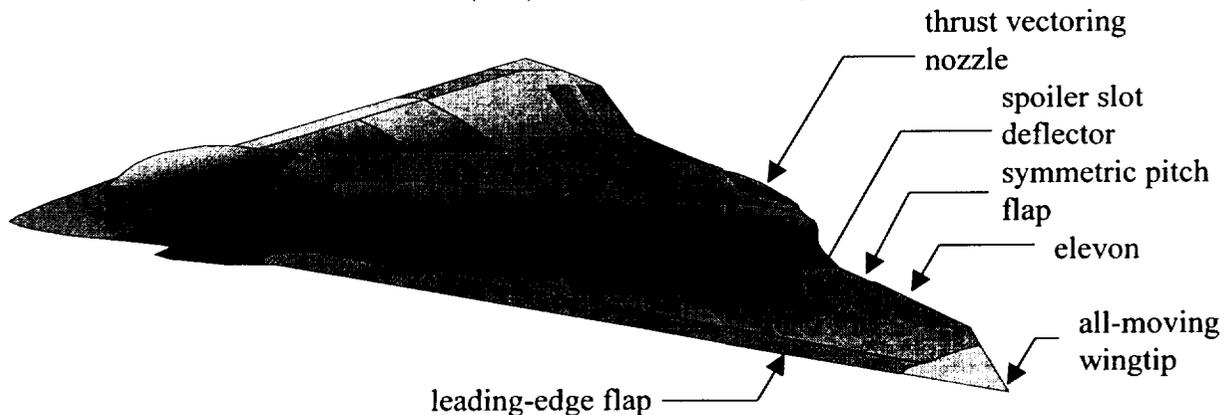


Figure I.1 The ICE vehicle

and the derivation of a linear model is given in the Appendix.

C. Design Challenge

1. Desired Control System Response Types

The control system response types are straightforward: For longitudinal control, a pitch-rate command system should be created. There is no requirement for attitude hold in this system. For lateral control, roll-rate and sideslip-command systems should be created, again with no requirement for attitude hold in the roll rate system. Commands to the pitch-rate and roll-rate systems are assumed to be generated by cockpit inceptor commands, e.g., longitudinal and lateral control column inputs. No pilot-generated sideslip commands will be in evidence, i.e., the system should maintain zero sideslip with the pilot's "feet on the floor". The vehicle models that will be provided have been linearized about a series of different flight conditions. The maneuvers to be described will be transient in nature and no thrust changes from trim values will be required.

The desired characteristics of the rate command systems will be predicated upon meeting handling qualities requirements in the pitch and roll axes to be specified in Section I.C.5.2. However desired frequency-domain characteristics can be specified as:

$$\begin{aligned} \frac{\theta}{\theta_c} &\approx \frac{100}{s(s+5)(s+20)} \\ \frac{\phi_s}{\phi_{s_c}} &\approx \frac{100}{s(s+5)(s+20)} \\ \frac{\beta}{\beta_c} &\approx \frac{100}{(s+5)(s+20)} \end{aligned} \quad \text{all for } 0.5 \leq \omega \leq 50 \text{ rad/sec} \quad (1)$$

where θ and θ_c represent the Euler pitch attitude and attitude commands, and ϕ_s and ϕ_{s_c} represent roll attitude and attitude commands about the x-stability axis, i.e. describing rolling motion about the vehicle velocity vector in equilibrium flight.

Finally, in keeping with the design philosophy of the ICE vehicle, the controller must utilize all the available control surface effectors.

2. Controller Simulink® Implementation

The controller must be implemented as a discrete device in the Simulink® computer simulation that will form an important part of the Design Challenge response. A frame rate of 80 Hz may be assumed in the discretization.

3. Pilot Models

A pair of analytical pilot models will be specified as part of the computer simulation of the pilot/vehicle system. Figure I.2 is the Simulink® diagrams for the pilot models. No pilot model for the beta loop is necessary because of the assumption of “feet-on-the-floor” activity by the pilot. The pilot models of Fig. 2 have been created assuming the dynamics of Eq. 1 are in evidence in the θ and ϕ_s loops. The gains of 2.67 on visual error

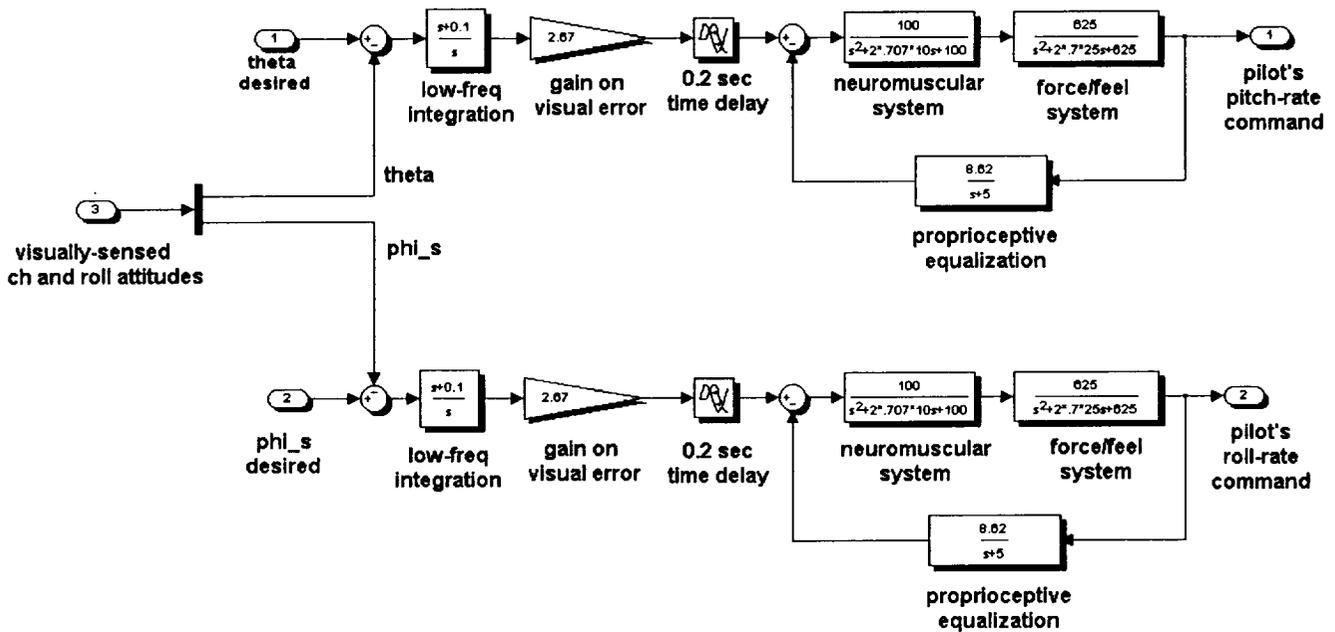


Figure I.2 Pilot models

in each model will produce crossover frequencies of 1.0 rad/sec in each of the loops, a value representative of moderate pilot control activity. Figure I.3 is the Bode plot of the open-loop transfer function of the pilot/vehicle system for both the θ and ϕ -loops when the dynamics of Eq. 1 are in evidence.

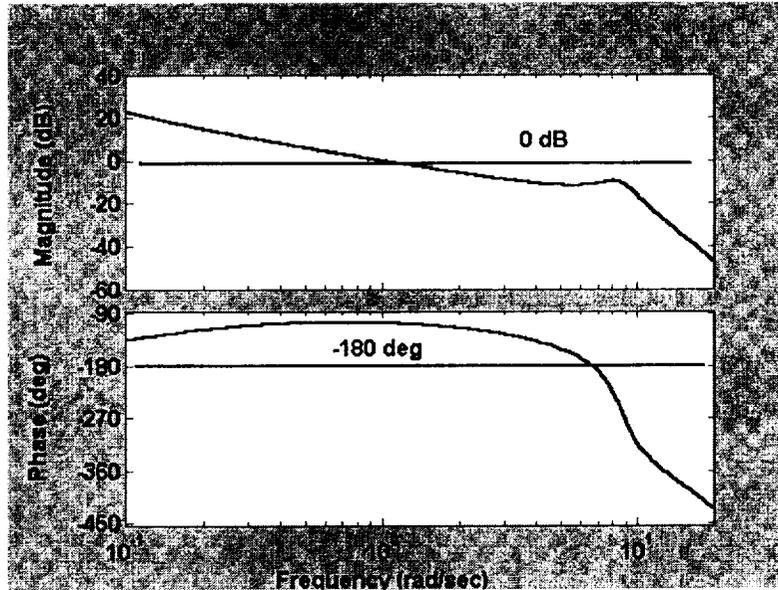


Figure I.3 Bode diagram of nominal pilot/vehicle open-loop system

3. Sensor Models

It will be assumed that any and all of the output variables specified in the Appendix for the linear models to be discussed will be available for feedback in the control system design. These include v , α_w , q_b , θ , β_w , p_s , r_s , ϕ , $a_{x_{cg}}$, $a_{y_{cg}}$, and $a_{n_{cg}}$. No sensor dynamics will be modeled with the exception of a 0.02 sec time delay assumed for each sensed variable. Additive sensor noise must be included in each measured variable as shown below in Fig. I.4.

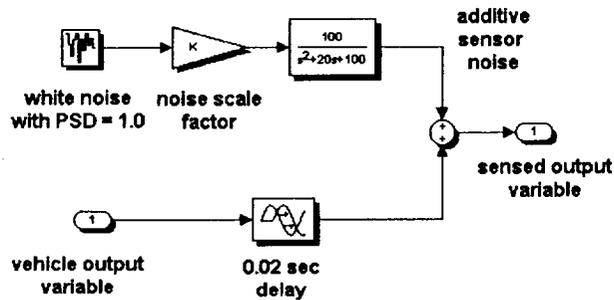


Figure I.4 Sensor noise models

The noise scale factor is given as $K = 0.15625$ for all sensed variable except for θ and ϕ , for which $K = 0$, and for $a_{x_{cg}}$, $a_{y_{cg}}$, and $a_{n_{cg}}$ for which $K = 0.015625$. The variables for θ and ϕ should not be differentiated in the control law to be determined.

With $K = 0.15625$, the root mean square (RMS) value of the output of the filter in Fig. I.4 will be 0.25. Thus, for example, the RMS noise on α_w would be 0.25 deg, etc.

4. Linear Models – Flight Conditions

Linear models for the ICE vehicle are enclosed in the Appendix for the following four flight conditions:

Level Flight:	Mach No. = 0.3	Altitude = 15,000 ft
	Mach No. = 0.6	Altitude = 25,000 ft
	Mach No. = 0.9	Altitude = 35,000 ft
Steady Turning Flight	Mach No. = 0.6	Altitude = 25,000 ft

The latter condition involves a trim bank angle of 60 deg.

5. Piloting Tasks

The piloting tasks will consist of the pilot following a series of simultaneous pitch and roll attitude commands for a period of 50 sec in each flight condition. The commands consist of a series of filtered 5 sec pulses, alternating in sign as shown in Fig. I.5. The amplitude of the pulses depends upon the flight condition and the command, i.e., whether it is a pitch or roll command. As will be described in Section I.C.8, in comparing the command input to the corresponding pilot/vehicle response, a 1.0 sec delay can be added to the recorded command signal. This is only for the purposes of performance assessment. The delay is not included in the input to the pilot/vehicle system in the Simulink[®] simulation. Figure I.6 shows the command generator.

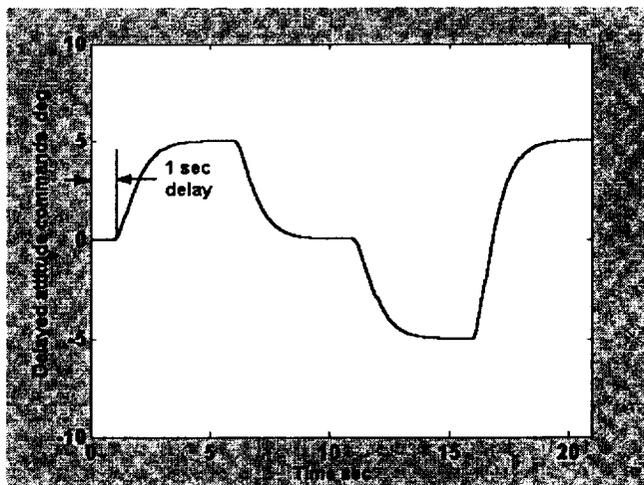


Figure I.5 Typical attitude command for pilot/vehicle task

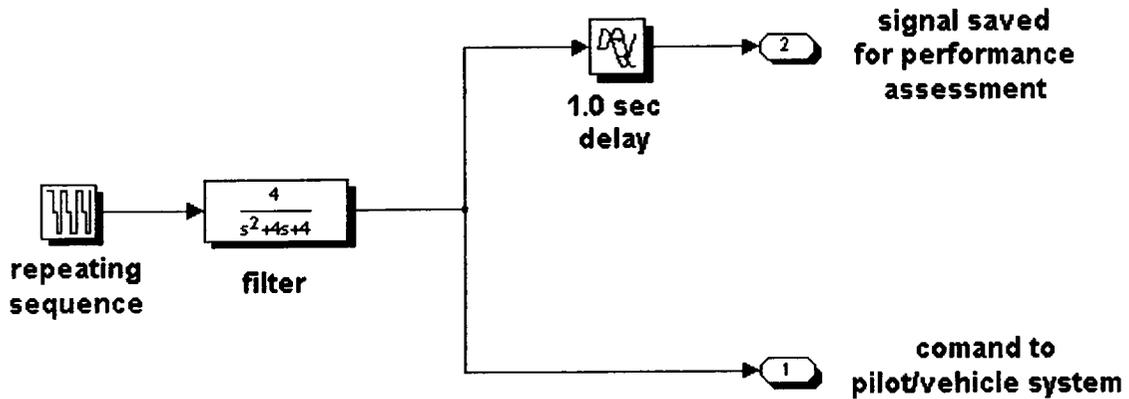


Figure I.6 Command signal generation

The amplitudes of the filtered pulses are given as

Level Flight:	Mach No. = 0.3	Altitude = 15,000 ft	$\theta_{amp} = 5 \text{ deg};$	$\phi_{amp} = 30 \text{ deg}$
	Mach No. = 0.6	Altitude = 25,000 ft	$\theta_{amp} = 5 \text{ deg};$	$\phi_{amp} = 30 \text{ deg}$
	Mach No. = 0.9	Altitude = 35,000 ft	$\theta_{amp} = 0 \text{ deg};$	$\phi_{amp} = 5 \text{ deg}$

Steady Turning Flight	Mach No. = 0.6	Altitude = 25,000 ft	$\theta_{amp} = 0 \text{ deg};$	$\phi_{amp} = 5 \text{ deg}$
-----------------------------	----------------	----------------------	---------------------------------	------------------------------

6. Performance/Handling Qualities Specifications

a. Desired Handling Qualities

Handling qualities specifications are to be analytically determined using bandwidth/phase delay as applied to the pitch-rate and roll-rate systems. Figure I.7 is to be used to define bandwidth and phase delay, and Fig. I.8 is to be used to assign handling qualities levels. NB: The transfer functions to be used in applying the measures of Fig. I.7 are θ/θ_{cFS} and ϕ/ϕ_{cFS} where θ_{cFS} and ϕ_{cFS} include the dynamics of the force/feel system given by

$$\text{force/feel dynamics} = \frac{625}{s^2 + 35s + 625} \quad (2)$$

Note that the units on the force/feel system dynamics are of no importance in this analysis.

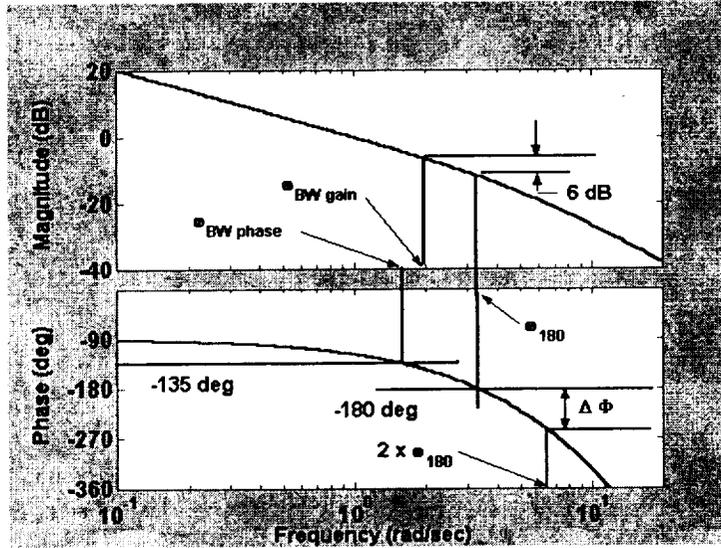


Figure I.7 Determining bandwidth for handling qualities determination
 Bode diagram is for either θ/θ_{cFS} or ϕ/ϕ_{cFS}

Phase delay τ_p is determined from Fig. I.7 as:

$$\tau_p = \frac{\Delta\Phi}{(57.3)(2 \times \omega_{180})} \text{ sec} \quad (3)$$

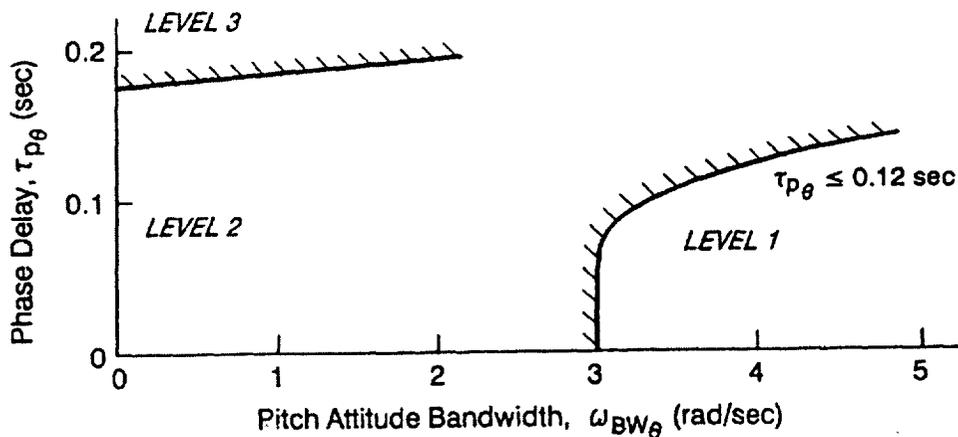


Figure I.8 Handling qualities level - bandwidth/phase delay mapping for pitch axis; for roll axis, the roll bandwidth determined from Fig. 7 shall be at least 1 rad/sec for Level 1. Phase delay shall be no more that 0.14 sec for Level 1 and 0.2 sec for Level 2

b. Desired Tracking Performance

The pilot/vehicle performance specifications in each of the flight conditions are:

pitch-attitude tracking:

$$\theta_{amp} = 5 \text{ deg or } 0 \text{ deg}$$

Desired: no sustained oscillations; maximum error of ± 2 deg

Adequate: no sustained oscillations; maximum error of ± 3 deg

roll-attitude tracking:

$$\phi_{amo} = 30 \text{ deg}$$

Desired: no sustained oscillations; maximum error of ± 10 deg

Adequate: no sustained oscillations; maximum error of ± 15 deg

$$\phi_{amo} = 5 \text{ deg}$$

Desired: no sustained oscillations; maximum error of ± 2 deg

Adequate: no sustained oscillations; maximum error of ± 3 deg

sideslip tracking:

All attitude command amplitudes

Desired: maximum excursions of ± 2 deg

Adequate: maximum excursions of ± 5 deg

c. Desired Stability Margins

Stability margins in the control system design are to be determined in classical single-loop fashion by sequentially cutting each control loop before each actuator as indicated in Fig. I.9.

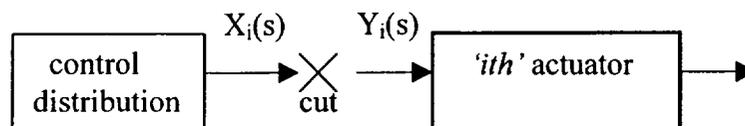


Figure I.9 Determining loop break point

From the Bode diagram of $\frac{X_i}{Y_i}(j\omega)$ the following stability margins should be in evidence: Gain Margin ≥ 6 dB; Phase Margin ≥ 30 deg

NB: It is obvious that this stability analysis is not as rigorous as one obtained by inserting a perturbation matrix of the form $P = \text{diag}(K_1 e^{-j\phi_1}, \dots, K_n e^{-j\phi_n})$ before the actuators and assessing closed-loop stability when K_i and ϕ_i are varied within some desired region in the gain and phase parameter space. The single-loop approach was adopted for the sake of simplicity here given the number of actuators involved in the vehicle.

d. Structural Coupling Considerations

Although no elastic modes have been included in the linearized ICE models, to minimize excitation of structural modes the following requirements on closed-loop transfer functions should be met for $\omega \leq 25$ rad/sec:

$$\begin{aligned} \left| \frac{q_b}{\theta_c}(j\omega) \right| &\leq -10 \text{ dB} \\ \left| \frac{p_b}{\phi_c}(j\omega) \right| &\leq -10 \text{ dB} \\ \left| \frac{a_{n_{cg}}}{\theta_c}(j\omega) \right| &\leq -20 \text{ dB} \end{aligned} \quad (4)$$

e. Control Activity

In each of the flight conditions for the tracking tasks to be considered, control activity should meet the following criterion:

The half-power frequency in the command signal to each actuator should be less than or equal to the actuator bandwidth (here assumed equivalent to the undamped natural frequency of the actuator).

In equation form, this requirement can be stated as

$$(\omega_{0.5})_i \leq (\omega_n)_i \quad (5)$$

where $(\omega_{0.5})_i$ is obtained as

$$0.5 = \int_0^{(\omega_{0.5})_i} \Phi_{\delta_{c_i}, \delta_{c_i}}(\omega) d\omega \quad (6)$$

with δ_{c_i} denoting the command to the i^{th} actuator and $\Phi_{xx}(\omega)$ denoting the power spectral density of the signal $x(t)$. Finally, $(\omega_n)_i$ denotes the undamped natural frequency of the i^{th} actuator. The requirement of Eq. 5 places a limit on the control activity required by the control system and the sensor noise transmission that occurs as a result of the design.

f. Scheduling Considerations

In the analysis and computer simulation of the pilot/vehicle system, the flight control design may be tuned to each flight condition being considered. The one exception to this is the turning flight condition, where the level flight dynamics must be considered in the control system design. No scheduling algorithms need be developed. However, a discussion of the approach to scheduling should be included. Finally, to demonstrate the robustness of the design approach, the pilot/vehicle system tasks should be performed at each of the four flight conditions using the control system designed for the Mach 0.6 Altitude 25,000 ft flight condition. No handling qualities evaluations need be included, however, tracking task performance should be ascertained using the requirements of Section I.C.7.b.

7. Failure Modes

After computer simulation of the nominal system in the specified flight conditions, the following system failure(s) should be evaluated via Simulink[®] simulation. These failures should be evaluated at each flight condition with the tracking tasks specified in Section I.C.5.

<u>Failure No.</u>	<u>Failure Description</u>
1	left elevon actuator amplitude limits reduced to ± 15 deg rate limits reduced to ± 15 deg/sec
2	symmetric pitch flap actuator with hard-over of +5 deg
3	left spoiler actuator with rate limits reduced to ± 10 deg/sec
4	left leading edge outboard flap with hard-over of +5 deg
5	pitch nozzle actuator with rate limits reduced to ± 10 deg/sec
6	elements of linear plant A and B matrices each changed by ± 20 %. This change should not include those elements of the A matrix that describe kinematic relationships. To ensure uniformity, the sign of the 20% variations will be dependent upon the column of the A and B matrices, with even- numbered columns receiving a +20% change and odd- numbered columns receiving a - 20% change.

It is desired that the pilot/vehicle system be able to tolerate all six of these failures simultaneously. In the Simulink[®] simulation, the failures should be introduced at $t = 20$ sec. Section I.C.8 briefly defines an M-file that will automate the failures.

In the event that the simultaneous failures prove too severe, a series of Simulink® simulations can be undertaken in which the failures are added incrementally. If this procedure is followed, the order in which the failures are accumulated should follow the failure numbers given above. That is, first, No. 1, then Nos. 1 and 2, etc.

8. MATLAB® Models and M-file for Simulink® Failure Simulation

A MATLAB® M-file called “IceFunctionsDC.m” is provided that automates the failures just described. The file can be used with a Simulink® simulation module called ICEdc.mdl containing the state-space vehicle model, actuator models, input commands, pilot models and reference models. Figure I.10 shows ICEdc.mdl.

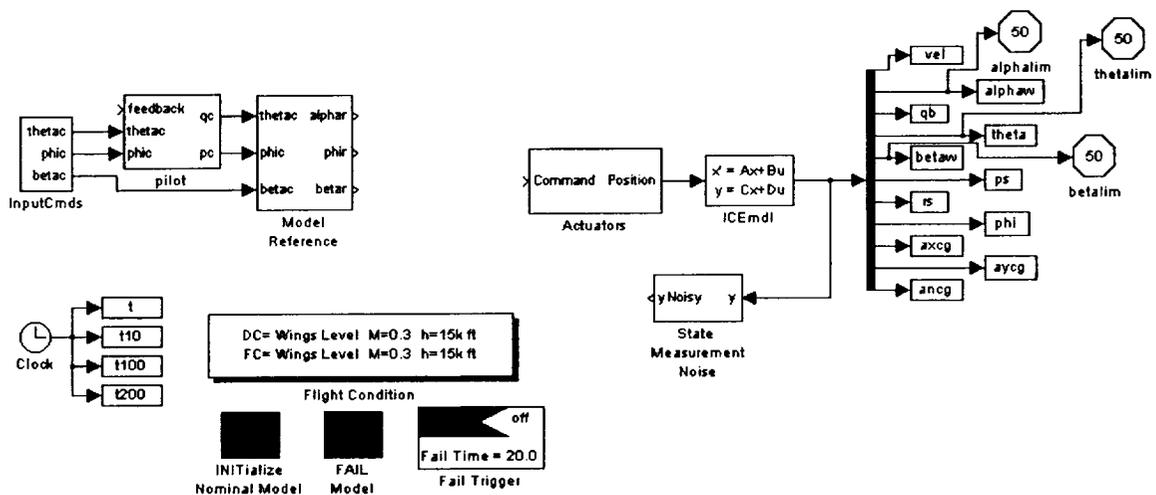


Figure I.10 The ICEdc.mdl in Simulink®

By left clicking on any of the blocks above, the underlying model structures can be viewed. By clicking on the shadowed box containing “DC = Wings Level...” the menu shown in Fig. I.11 appears.

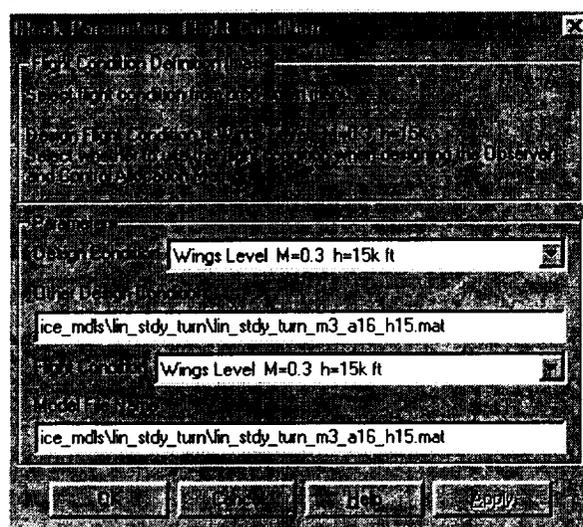


Figure. I.11 ICEdc.mdl menu

The menu of Fig. I.11 allows the user to select flight conditions, and “design” conditions. If desired, the user can load their control designs into MATLAB, using the “Other Design Conditions” label. Likewise, the Flight Conditions can be loaded into MATLAB using the “Model File Name.” For example, in Fig. I.11, the “Design Condition” is Wings Level, Mach No. = 0.3 and Altitude = 15,000 ft, i.e. this is the flight condition for which the control system has been designed or “tuned”. The flight condition selected is also Wings Level, Mach No = 0.3 and Altitude = 15,000 ft. The octagonal elements on the right hand side of the model will terminate the Simulink® simulation when the associated output variables exceed the values within the elements, e.g., here 50, indicating instability. The Model Reference is also included. Depending upon the particular design philosophy adopted by the user, this block may or may not be necessary, i.e. the reference model dynamics may be implicit in the design.

Returning to Fig. I.10, one initializes the system by left clicking on the shaded box labeled INITIALize Nominal Model. This sets the actuators to nominal conditions and the vehicle model to the flight condition specified in the menu of Fig. I.11. By left clicking on the Fail Trigger box, one can select a failure time, now set at a default value of 20 sec. Alternatively, the user can manually fail the system by left clicking on the FAIL model box any time during the simulation run. The failure conditions (including the $\pm 20\%$ variation in appropriate elements of the A and B matrices) must be specified by the user through the m-file IceFunctionsDC.m. listed in Appendix I..

9.0 Design Challenge Summary

The Design Challenge can be summarized as follows:

Vehicle: ICE aircraft, linearized about four flight conditions.

Controller: Designed by participant. May be tuned to each of the four flight conditions, with the exception that for the turning flight condition, the controller is tuned to the wings-level condition. Each of the control effectors must be utilized in the design. Controller is to be discretized for Simulink® computer simulation.

Handling Qualities: Ascertained through bandwidth/phase delay for each of the four flight conditions, for nominal vehicle only (no failures).

Tracking Performance: Ascertained through Simulink® computer simulation for each flight condition. Should include nominal and failed cases. Failure should occur 20 seconds into a 50 second run.

Stability Margins: Ascertained through linear analysis for each of the four flight conditions, for nominal vehicle only (no failures).

- Structural Coupling: Ascertained through linear analysis for each of the four flight conditions, for nominal vehicle only (no failures).
- Control Activity: Ascertained through Simulink® computer simulation for each flight condition, for nominal vehicle only (no failures).
- Scheduling Considerations: Scheduling need not be implemented, but means for accommodating scheduling (if necessary) should be discussed.
- Off-nominal Flight Cond's: Ascertained through Simulink® computer simulation for each of the four flight conditions. Controller designed for Mach No. = 0.6, Altitude = 25,000 ft should be employed in remaining three flight conditions. Tracking performance should be assessed.

D. Appendix I Listing for IceFunctionsDC.m

```
function [errFlag,tstep,sysDC,sysFC,sysFail,icFC,icDC] = IceFunctionsDC(inVar);
%
%This m-file contains all required functions for the ICE Simulink model
%
%Inputs:
% inVar - input variable, may be a string or numeric vector. The following cases are defined:
%   'Init' - Update model (state space and actuators) with nominal system
%   'Fail' - Update model (state space and actuators)with failed system
%   [0 failFlag] - fail trigger input vector
%   [u x1 x2 ...] - call various user defined functions with arguments x1 x2 ...
%
%
%Outputs:
% errFlag - error flag, 0=no error, 1=error
% tstep - simulation time step
% SysFC - system state space matrices for current flight condition
% SysDC - Design condition state space matrices
% SysFail - failed system state space matrices
% uDC,xDC,yDC - Initial conditions and inputs for Design condition
% SIMULINK BLOCKS - various blocks updated for each different case

currentSys = gcs; %parent name of current SIMULINK object
slashSpot = findstr(currentSys,'/'); %find fwd slashes in current sys object
if size(slashSpot,2) > 0
    fileName = currentSys(1:slashSpot(1)-1); %delete everything to left of last slash
else
    fileName = currentSys;
end
end
```

```

% Determine which function to switch to
if ischar(inVar) %is the input a character or a number
    if strcmp(inVar,'Init')
        myCase = 1; %Initialization of Nominal model
    elseif strcmp(inVar,'Fail')
        myCase = 2; %Initialization of Failed model
    else
        myCase = 0; %Invalid input string
    end
elseif inVar(1)==0 %is input a number?%Check fail trigger
    if inVar(2)==1 %IF FailFlag==true
        myCase = 2; %Fail the vehicle
    else
        myCase = 3; %Do nothing, no error flag returned
    end
elseif inVar(1)==1 %Do Something per your design.
    myCase = 4;
elseif inVar(1)==2 %Do Somethin per your design.
    myCase = 5;
elseif inVar(1)==3 %Do something per your design.
    myCase = 6;
else
    myCase = 0; %Invalid input
end
errFlag = 1;

% Output Variable Defintions

tstep = 0.0005; %simulation time step

% Initialize system state space matrices for designed condition
% This is only included for possible aid in simulating flight envelope testing.
% It does nothing in the given SIMULINK file!
DC = get_param(strcat(fileName,'/Flight Condition'),'DC'); %Read desired flight condition
dsgnFile=get_param(strcat(fileName,'/Flight Condition'),'dsgnFile'); %Read possible input file
if strcmp(DC,'Other...')==1; %if Dropdown says "other" right dsgnFile to DC
    DC=dsgnFile;
end %if

% Initialize system state space matrices for flight condition
% This writes to the block FSAVmodel whatever model is in the Flight dropdown or if "other"
% is selected then it reads the input file
FC = get_param(strcat(fileName,'/Flight Condition'),'FC');
modFile=get_param(strcat(fileName,'/Flight Condition'),'ModelFile');
if strcmp(FC,'Other...') ==1;
    FC=modFile;
end %if

% Call function to write FC to model, DC only writes DC to sysDC nothing else
[sysFC,sysDC,sysFail,icFC,icDC]=getICESystem(DC,FC);

switch myCase
case 0
    errFlag = 1;

```

```

-----
% NOMINAL NOMINAL
-----
case 1
% Initialize Nominal system and actuators in Simulink
% Inputs:
%   A,B,C,D - nominal state space matrices
%   fileName - current SIMULINK file name
% Outputs:
%   SIMULINK OBJECTS - update state space system block and all actuator components

%Left Elevon (2nd Order, with limits); G(s) = k * w^2/(s^2 + 2*damp*w*s + w^2)
delay = 0.00001; %Transport delay (sec)
k = 1; %gain
w = 63.2455532; %frequency (rad/s)
damp = 1.106797181; %damping ratio
dMaxHi = 30; %max position, upper limit
dMaxLo = -30; %max position, lower limit
ddotMax = 150; %max rate (+/- unit/sec), enter positive number
hardOver = 0; %hard-over switch (0=no hard-over, 1=hard-over)
jamPos = 0; %hard-over actuator position (unit)
setActuators(fileName,'03',[delay;k;w;damp;dMaxHi;dMaxLo;ddotMax;hardOver;jamPos]);

%Right Elevon (2nd Order, with limits); G(s) = k * w^2/(s^2 + 2*damp*w*s + w^2)
delay = 0.00001; %Transport delay (sec)
k = 1; %gain
w = 63.2455532; %frequency (rad/s)
damp = 1.106797181; %damping ratio
dMaxHi = 30; %max position, upper limit
dMaxLo = -30; %max position, lower limit
ddotMax = 150; %max rate (+/- unit/sec), enter positive number
hardOver = 0; %hard-over switch (0=no hard-over, 1=hard-over)
jamPos = 0; %hard-over actuator position (unit)
setActuators(fileName,'13',[delay;k;w;damp;dMaxHi;dMaxLo;ddotMax;hardOver;jamPos]);

%Symmetric Pitch Flap (2nd Order, with limits); G(s) = k * w^2/(s^2 + 2*damp*w*s + w^2)
delay = 0.00001; %Transport delay (sec)
k = 1; %gain
w = 63.2455532; %frequency (rad/s)
damp = 1.106797181; %damping ratio
dMaxHi = 30; %max position, upper limit
dMaxLo = -30; %max position, lower limit
ddotMax = 50; %max rate (+/- unit/sec), enter positive number
hardOver = 0; %hard-over switch (0=no hard-over, 1=hard-over)
jamPos = 0; %hard-over actuator position (unit)
setActuators(fileName,'4',[delay;k;w;damp;dMaxHi;dMaxLo;ddotMax;hardOver;jamPos]);

```

```

%Left All Moving Tip (2nd Order, with limits); G(s) = k * w^2/(s^2 + 2*damp*w*s + w^2)
delay = 0.00001; %Transport delay (sec)
k = 1; %gain
w = 63.2455532; %frequency (rad/s)
damp = 1.106797181; %damping ratio
dMaxHi = 60; %max position, upper limit
dMaxLo = 0; %max position, lower limit
ddotMax = 150; %max rate (+/- unit/sec), enter positive number
hardOver = 0; %hard-over switch (0=no hard-over, 1=hard-over)
jamPos = 0; %hard-over actuator position (unit)
setActuators(fileName,'05',[delay;k;w;damp;dMaxHi;dMaxLo;ddotMax;hardOver;jamPos]);

%Right All Moving Tip (2nd Order, with limits); G(s) = k * w^2/(s^2 + 2*damp*w*s + w^2)
delay = 0.00001; %Transport delay (sec)
k = 1; %gain
w = 63.2455532; %frequency (rad/s)
damp = 1.106797181; %damping ratio
dMaxHi = 60; %max position, upper limit
dMaxLo = 0; %max position, lower limit
ddotMax = 150; %max rate (+/- unit/sec), enter positive number
hardOver = 0; %hard-over switch (0=no hard-over, 1=hard-over)
jamPos = 0; %hard-over actuator position (unit)
setActuators(fileName,'15',[delay;k;w;damp;dMaxHi;dMaxLo;ddotMax;hardOver;jamPos]);

%Left Spoiler (2nd Order, with limits); G(s) = k * w^2/(s^2 + 2*damp*w*s + w^2)
delay = 0.00001; %Transport delay (sec)
k = 1; %gain
w = 63.2455532; %frequency (rad/s)
damp = 1.106797181; %damping ratio
dMaxHi = 60; %max position, upper limit
dMaxLo = 0; %max position, lower limit
ddotMax = 150; %max rate (+/- unit/sec), enter positive number
hardOver = 0; %hard-over switch (0=no hard-over, 1=hard-over)
jamPos = 0; %hard-over actuator position (unit)
setActuators(fileName,'09',[delay;k;w;damp;dMaxHi;dMaxLo;ddotMax;hardOver;jamPos]);

%Right Spoiler (2nd Order, with limits); G(s) = k * w^2/(s^2 + 2*damp*w*s + w^2)
delay = 0.00001; %Transport delay (sec)
k = 1; %gain
w = 63.2455532; %frequency (rad/s)
damp = 1.106797181; %damping ratio
dMaxHi = 60; %max position, upper limit
dMaxLo = 0; %max position, lower limit
ddotMax = 150; %max rate (+/- unit/sec), enter positive number
hardOver = 0; %hard-over switch (0=no hard-over, 1=hard-over)
jamPos = 0; %hard-over actuator position (unit)
setActuators(fileName,'19',[delay;k;w;damp;dMaxHi;dMaxLo;ddotMax;hardOver;jamPos]);

```

```

%Left Leading Edge Outboard Flap (2nd Order, with limits); G(s) = k * w^2/(s^2 + 2*damp*w*s + w^2)
delay = 0.00001; %Transport delay (sec)
k = 1; %gain
w = 42.22321636; %frequency (rad/s)
damp = 1.3952987; %damping ratio
dMaxHi = 40; %max position, upper limit
dMaxLo = -40; %max position, lower limit
ddotMax = 40; %max rate (+/- unit/sec), enter positive number
hardOver = 0; %hard-over switch (0=no hard-over, 1=hard-over)
jamPos = 0; %hard-over actuator position (unit)
setActuators(fileName,'02',[delay;k;w;damp;dMaxHi;dMaxLo;ddotMax;hardOver;jamPos]);

%Right Leading Edge Outboard Flap (2nd Order, with limits); G(s) = k * w^2/(s^2 + 2*damp*w*s + w^2)
delay = 0.00001; %Transport delay (sec)
k = 1; %gain
w = 42.22324636; %frequency (rad/s)
damp = 1.3952987; %damping ratio
dMaxHi = 40; %max position, upper limit
dMaxLo = -40; %max position, lower limit
ddotMax = 40; %max rate (+/- unit/sec), enter positive number
hardOver = 0; %hard-over switch (0=no hard-over, 1=hard-over)
jamPos = 0; %hard-over actuator position (unit)
setActuators(fileName,'12',[delay;k;w;damp;dMaxHi;dMaxLo;ddotMax;hardOver;jamPos]);

%Pitch Nozzle (2nd Order, with limits); G(s) = k * w^2/(s^2 + 2*damp*w*s + w^2)
delay = 0.00001; %Transport delay (sec)
k = 1; %gain
w = 39.18905204; %frequency (rad/s)
damp = 1.001376608; %damping ratio
dMaxHi = 15; %max position, upper limit
dMaxLo = -15; %max position, lower limit
ddotMax = 60; %max rate (+/- unit/sec), enter positive number
hardOver = 0; %hard-over switch (0=no hard-over, 1=hard-over) NOTE: INTEGER
jamPos = 0; %hard-over actuator position (unit)
setActuators(fileName,'10',[delay;k;w;damp;dMaxHi;dMaxLo;ddotMax;hardOver;jamPos]);

%Yaw Nozzle (2nd Order, with limits); G(s) = k * w^2/(s^2 + 2*damp*w*s + w^2)
delay = 0.00001; %Transport delay (sec)
k = 1; %gain
w = 39.18905204; %frequency (rad/s)
damp = 1.001376608; %damping ratio
dMaxHi = 15; %max position, upper limit
dMaxLo = -15; %max position, lower limit
ddotMax = 60; %max rate (+/- unit/sec), enter positive number
hardOver = 0; %hard-over switch (0=no hard-over, 1=hard-over) NOTE: INTEGER
jamPos = 0; %hard-over actuator position (unit)
setActuators(fileName,'20',[delay;k;w;damp;dMaxHi;dMaxLo;ddotMax;hardOver;jamPos]);

```

```

set_param(strcat(fileName,'/Actuators'),'BackgroundColor','white')

%Initialize the state space model and Input amplitudes
set_param(strcat(fileName,'/ICEmdl'),'A',mat2str(sysFC.a,6),'B',mat2str(sysFC.b,6),'C',
mat2str(sysFC.c,6),'D',mat2str(sysFC.d,6),'X0',mat2str(icFC.x0,6));
set_param(strcat(fileName,'/ICEmdl'),'BackgroundColor','white');
set_param(strcat(fileName,'/InputCmds/thetaamp'),'Value',num2str(icFC.thetaamp));
set_param(strcat(fileName,'/InputCmds/phiamp'),'Value',num2str(icFC.phiamp));

disp('IceFunctionsDC>> Initialization of model completed...')
disp('      Outputs [errFlag,tstep,sysDC,sysFC,sysFail,icFC,icDC] written.  ')

errFlag = 0;

%-----
% FAILED FAILED
%-----
case 2
% Initialize FAILED system and actuators in Simulink
% Inputs:
%   Afail,Bfail,Cfail,Dfail - FAILED state space matrices
%   fileName - current SIMULINK file name
% Outputs:
%   SIMULINK OBJECTS - update state space system block and all actuator components

%FAILED Left Elevon (2nd Order, with limits); G(s) = k * w^2/(s^2 + 2*damp*w*s + w^2)
delay = 0.00001; %Transport delay (sec)
k = 1; %gain
w = 63.2455532; %frequency (rad/s)
damp = 1.106797181; %damping ratio
dMaxHi = 30; %max position (unit), upper limit (Nom:30)
dMaxLo = -30; %max position (unit), lower limit (Nom:30)
ddotMax = 15; %max rate (+/- unit/sec), enter positive number (Nom:150)
hardOver = 0; %hard-over switch (0=no hard-over, 1=hard-over)
jamPos = 0; %hard-over actuator position (unit)
setActuators(fileName,'03',[delay;k;w;damp;dMaxHi;dMaxLo;ddotMax;hardOver;jamPos]);

%FAILED Right Elevon (2nd Order, with limits); G(s) = k * w^2/(s^2 + 2*damp*w*s + w^2)
delay = 0.00001; %Transport delay (sec)
k = 1; %gain
w = 63.2455532; %frequency (rad/s)
damp = 1.106797181; %damping ratio
dMaxHi = 30; %max position, upper limit
dMaxLo = -30; %max position, lower limit
ddotMax = 150; %max rate (+/- unit/sec), enter positive number
hardOver = 0; %hard-over switch (0=no hard-over, 1=hard-over)
jamPos = 0; %hard-over actuator position (unit)
setActuators(fileName,'13',[delay;k;w;damp;dMaxHi;dMaxLo;ddotMax;hardOver;jamPos]);

```

```

%FAILED Symmetric Pitch Flap (2nd Order, with limits); G(s) = k * w^2/(s^2 + 2*damp*w*s + w^2)
delay = 0.00001; %Transport delay (sec)
k = 1; %gain
w = 63.2455532/6.32455532; %frequency (rad/s)
damp = 1.106797181; %damping ratio
dMaxHi = 30; %max position, upper limit
dMaxLo = -30; %max position, lower limit
ddotMax = 50; %max rate (+/- unit/sec), enter positive number
hardOver = 1; %hard-over switch (0=no hard-over, 1=hard-over)
jamPos = 5; %hard-over actuator position (unit)
setActuators(fileName, '4', [delay;k;w;damp;dMaxHi;dMaxLo;ddotMax;hardOver;jamPos]);

%FAILED Left All Moving Tip (2nd Order, with limits); G(s) = k * w^2/(s^2 + 2*damp*w*s + w^2)
delay = 0.00001; %Transport delay (sec)
k = 1; %gain
w = 63.2455532; %frequency (rad/s)
damp = 1.106797181; %damping ratio
dMaxHi = 60; %max position, upper limit
dMaxLo = 0; %max position, lower limit
ddotMax = 150; %max rate (+/- unit/sec), enter positive number
hardOver = 0; %hard-over switch (0=no hard-over, 1=hard-over)
jamPos = 60; %hard-over actuator position (unit)
setActuators(fileName, '05', [delay;k;w;damp;dMaxHi;dMaxLo;ddotMax;hardOver;jamPos]);

%FAILED Right All Moving Tip (2nd Order, with limits); G(s) = k * w^2/(s^2 + 2*damp*w*s + w^2)
delay = 0.00001; %Transport delay (sec)
k = 1; %gain
w = 63.2455532; %frequency (rad/s)
damp = 1.106797181; %damping ratio
dMaxHi = 60; %max position, upper limit
dMaxLo = 0; %max position, lower limit
ddotMax = 150; %max rate (+/- unit/sec), enter positive number
hardOver = 0; %hard-over switch (0=no hard-over, 1=hard-over)
jamPos = 0; %hard-over actuator position (unit)
setActuators(fileName, '15', [delay;k;w;damp;dMaxHi;dMaxLo;ddotMax;hardOver;jamPos]);

%FAILED Left Spoiler (2nd Order, with limits); G(s) = k * w^2/(s^2 + 2*damp*w*s + w^2)
delay = 0.00001; %Transport delay (sec)
k = 1; %gain
w = 63.2455532; %frequency (rad/s)
damp = 1.106797181; %damping ratio
dMaxHi = 60; %max position, upper limit
dMaxLo = 0; %max position, lower limit
ddotMax = 10; %max rate (+/- unit/sec), enter positive number (Nom:150)
hardOver = 0; %hard-over switch (0=no hard-over, 1=hard-over)
jamPos = 0; %hard-over actuator position (unit)
setActuators(fileName, '09', [delay;k;w;damp;dMaxHi;dMaxLo;ddotMax;hardOver;jamPos]);

```

```

%FAILED Right Spoiler (2nd Order, with limits); G(s) = k * w^2/(s^2 + 2*damp*w*s + w^2)
delay = 0.00001; %Transport delay (sec)
k = 1; %gain
w = 63.2455532; %frequency (rad/s)
damp = 1.106797181; %damping ratio
dMaxHi = 60; %max position, upper limit
dMaxLo = 0; %max position, lower limit
ddotMax = 150; %max rate (+/- unit/sec), enter positive number
hardOver = 0; %hard-over switch (0=no hard-over, 1=hard-over)
jamPos = 0; %hard-over actuator position (unit)
setActuators(fileName, '19', [delay;k;w;damp;dMaxHi;dMaxLo;ddotMax;hardOver;jamPos]);

%FAILED Left Leading Edge Outboard Flap (2nd Order, with limits); G(s) = k * w^2/(s^2 + 2*damp*w*s + w^2)
delay = 0.00001; %Transport delay (sec)
k = 0.5; %gain
w = 42.22321636; %frequency (rad/s)
damp = 1.3952987; %damping ratio
dMaxHi = 40; %max position, upper limit
dMaxLo = -40; %max position, lower limit
ddotMax = 40; %max rate (+/- unit/sec), enter positive number
hardOver = 1; %hard-over switch (0=no hard-over, 1=hard-over)
jamPos = 5; %hard-over actuator position (unit)
setActuators(fileName, '02', [delay;k;w;damp;dMaxHi;dMaxLo;ddotMax;hardOver;jamPos]);

%FAILED Right Leading Edge Outboard Flap (2nd Order, with limits); G(s) = k * w^2/(s^2 + 2*damp*w*s + w^2)
delay = 0.00001; %Transport delay (sec)
k = 1; %gain
w = 42.22324636; %frequency (rad/s)
damp = 1.3952987; %damping ratio
dMaxHi = 40; %max position, upper limit
dMaxLo = -40; %max position, lower limit
ddotMax = 40; %max rate (+/- unit/sec), enter positive number
hardOver = 0; %hard-over switch (0=no hard-over, 1=hard-over)
jamPos = 0; %hard-over actuator position (unit)
setActuators(fileName, '12', [delay;k;w;damp;dMaxHi;dMaxLo;ddotMax;hardOver;jamPos]);

%FAILED Pitch Nozzle (2nd Order, with limits); G(s) = k * w^2/(s^2 + 2*damp*w*s + w^2)
delay = 0.00001; %Transport delay (sec)
k = 1; %gain
w = 39.18905204/3.918905204; %frequency (rad/s)
damp = 1.001376608; %damping ratio
dMaxHi = 15; %max position, upper limit
dMaxLo = -15; %max position, lower limit
ddotMax = 10; %max rate (+/- unit/sec), enter positive number(Nom:60)
hardOver = 0; %hard-over switch (0=no hard-over, 1=hard-over) NOTE:INTEGER
jamPos = 0; %hard-over actuator position (unit)
setActuators(fileName, '10', [delay;k;w;damp;dMaxHi;dMaxLo;ddotMax;hardOver;jamPos]);

```

```

%FAILED Yaw Nozzle (2nd Order, with limits); G(s) = k * w^2/(s^2 + 2*damp*w*s + w^2)
delay = 0.00001; %Transport delay (sec)
k = 1; %gain
w = 39.18905204/3.918905204; %frequency (rad/s)
damp = 1.001376608; %damping ratio
dMaxHi = 15; %max position, upper limit
dMaxLo = -15; %max position, lower limit
ddotMax = 60; %max rate (+/- unit/sec), enter positive number
hardOver = 0; %hard-over switch (0=no hard-over, 1=hard-over) NOTE:INTEGER
jamPos = 0; %hard-over actuator position (unit)
setActuators(fileName,'20',[delay;k;w;damp;dMaxHi;dMaxLo;ddotMax;hardOver;jamPos]);

set_param(strcat(fileName,'/Actuators'),'BackgroundColor','red')

%Initialize the FAILED state space model and controll allocation matrix
set_param(strcat(fileName,'/ICEmdl'),'A',mat2str(sysFail.a,6),'B',mat2str(sysFail.b,6),'C',
mat2str(sysFail.c,6),'D',mat2str(sysFail.d,6));
%in the above line do not set Initial conditions because we want to have continuous states during the simulation.
set_param(strcat(fileName,'/ICEmdl'),'BackgroundColor','red');

sound(1:100); %make a 1 second 100hz sound to indicate the model was failed...
disp('IceFunctions>> Model Failed successfully!')
errFlag = 0;

%-----
% Do Not Fail      Do Not Fail      Do Not Fail      Do Not Fail
%-----
case 3
    errFlag = 0;

%-----
% Case 4      Case 4      Case 4      Case 4      Case 4      Case 4      Case 4
%-----
%user definable function:... call as IceFunctionsDC([1 x1,x2 x3...]);
case 4

errFlag = 0;

%-----
% Case 5      Case 5
%-----
case 5

errFlag = 0;

%-----
% Case 6      Case 6
%-----
case 6

errFlag = 0;

%-----
% End Select      End Select      End Select      End Select      End Select      End Select
%-----
end %End of select case

%disp('Finished IceFunctions')
return; %IceFunctionDC.m

```

```

%=====

function setActuators(fileName,actuatorName,values)
% This function writes the actuator properties to each actuator
%inputs:
%   fileName: from ICEfunctionsDC the name of the SIMULINK file
%   actuatorName: The name of the actuator as defined in iceDC
%   values: The values necessary to define the actuator
%outputs:
%   Writes to a actuator in filename given.  Block containing actuators must be named
%   'Actuators'

%Set the parameters for a second-order acuator model in SIMULINK
set_param(strcat(fileName,'/Actuators/Delay',actuatorName),'DelayTime',num2str(values(1)));
set_param(strcat(fileName,'/Actuators/Gain',actuatorName),'Gain',num2str(values(2)));
set_param(strcat(fileName,'/Actuators/sw',actuatorName),'Gain',num2str(values(3)^2));
set_param(strcat(fileName,'/Actuators/d2w',actuatorName),'Gain',num2str(2*values(4)*values(3)));
set_param(strcat(fileName,'/Actuators/d Integrator',actuatorName),'UpperSaturationLimit',num2str(values(5)),
'LowerSaturationLimit',num2str(values(6)));
set_param(strcat(fileName,'/Actuators/ddot Integrator',actuatorName),'UpperSaturationLimit',num2str(values(7)),
'LowerSaturationLimit',num2str(-values(7)));
set_param(strcat(fileName,'/Actuators/Hard-over Switch',actuatorName),'sw',num2str(values(8)));
set_param(strcat(fileName,'/Actuators/jamPosition',actuatorName),'Value',num2str(values(9)));

return %setActuators

%=====

function [sysFC,sysDC,sysFail,icFC,icDC]=getICESystem(DC,FC);

%ICE Linear Models
%Nominal Model Definitions

unames = ['de3 ' ; 'del3 ' ; 'de4 ' ; 'de5 ' ; 'del5 ' ; 'de9 ' ; 'del9 ' ; 'de2 ' ; 'del2 ' ; 'de10 ' ; 'de20 '];
xnames = ['u ' ; 'u ' ; 'qb ' ; 'theta ' ; 'v ' ; 'pb ' ; 'rb ' ; 'phi '];
ynames = ['vel ' ; 'alpha' ; 'qb ' ; 'theta ' ; 'betaw ' ; 'ps ' ; 'rs ' ; 'phi ' ; 'axcg ' ; 'aycg ' ; 'ancg '];

%First set DC then FC
for i=1:1:2,
    if i==1;
        swtch=DC;
    elseif i==2;
        swtch=FC;
    else
        disp('error in getICESystem');
    end %if

%The following section is identicle to loading the selected model by choosing 'other' and typing inthe file
%the difference is only that they are included in this file.
%The 't' in the matrix names indicates that it is a temporary variable which will be modified. These
% are the exact same matrices found in the model files.

```

The listing for the next section of the m-file has been omitted here, and contains the state space descriptions of the ICE vehicle at the various flight conditions, including the initial conditions on the vehicle states. The latter values are not needed in implementing the linear models, but are useful for describing the equilibrium flight conditions.

```

%set input command magnitudes
thetaamp=0; %deg
phiamp=5; %deg

otherwise %if the text in DC or FC (depending on i) does not match any of the preloaded cases
% then read them from the file loaded and name them appropriately...
mdl=load(swch);
At = mdl.a;
Bt = mdl.b;
Ct = mdl.c;
Dt = mdl.d;
u0t= mdl.u0;
x0t= mdl.x0;
y0t= mdl.y0;

end %select swch
%If you so desire to modify the outputs or augment any input/output matrix
%this is the location to do so...
At = At;
Bt = Bt;
Ct = Ct;
Dt = Dt;
u0t= u0t;
x0t= x0t;
y0t= zeros(8,1); %This Sets initial conditions on the states to zero! your call...
%where to write the temp variables? are we calculating DC or FC
if i==1 % then write to DC

sysDC=ss(At,Bt,Ct,Dt);
icDC=struct('u0',u0t,'x0',x0t,'y0',y0t);
elseif i==2 %then write to FC
sysFC=ss(At,Bt,Ct,Dt);
icFC=struct('u0',u0t,'x0',x0t,'y0',y0t,'thetaamp',thetaamp,'phiamp',phiamp);

%Also define Failed model based on FC
%Failed Model Definition
%Afail = A*1.00;Afail(4,:)=A(4,:);Afail(8,:)=A(8,:);Bfail = B*0.75;Cfail = C;Dfail = D;
sysFail=sysFC; %initialize sysFail
for i = 1:8
    for j = 1:8
        sysFail.a(i,j)=sysFC.a(i,j)*(1+0.2*(-1)^j);
    end
end
for i = 1:8
    for j = 1:11
        sysFail.b(i,j)=sysFC.b(i,j)*(1+0.2*(-1)^j);
    end
end
sysFail.a(4,:)=sysFC.a(4,:); %Correct Kinematic lines back to nominal
sysFail.a(8,:)=sysFC.a(8,:); %Correct Kinematic lines back to nominal
sysFail.c = sysFC.c;
sysFail.d = sysFC.d;
else
disp('getICESystem>> error in the output variable definition')
end %if
end % for loop

return %getICESystem(DC,FC)

```

E. Appendix II – Innovative Control Effector (ICE) Vehicle Description

Introduction

A set of linear models and trim files has been provided to aid in the development of a controller for an advanced tailless fighter. This document contains the variable definitions and units corresponding to those models. A cursory description of the nonlinear vehicle is offered with the decomposition of aerodynamic database. Source of variations in linear models, i.e., stability and control derivatives, for three different trim conditions is traced back to the decomposition of the aerodynamic database. Mass properties, vehicle geometry, cg location, and a thrust vectoring model available in the open literature have also been collected. The result of the linear model-derived reconfigurable controller will be applied to the nonlinear simulation at Langley for comparison with in house controllers.

Variable List

aerods	dimensional aerodynamic force component along negative x-stability axis, slug-ft/sec ²
aerols	dimensional aerodynamic force component along negative z- stability axis, slug-ft/sec ²
alphaw	angle of attack, deg
alphdtw	angle of attack rate, deg/sec
altic	altitude, ft
amtl(DE5)	left all moving tail position, deg
amtr(DE15)	right all moving tail position, deg
anacc	normal acceleration at accelerometer, g's
ancg	normal acceleration at cg, g's
ancgstb	normal acceleration at cg along negative z-stability axis, g's
axcg	longitudinal acceleration at cg, g's
ayacc	lateral acceleration at accelerometer, g's
aycg	lateral acceleration at cg, g's
betadtw	sideslip angular rate, deg/sec
betaw	sideslip angle, deg
cdrag	stability axis aerodynamic force coefficient along negative x axis
clift	stability axis aerodynamic force coefficient along negative z axis
cpitch	stability axis aerodynamic pitching moment coefficient
croll	stability axis aerodynamic rolling moment coefficient
cside	stability axis aerodynamic force coefficient along y axis
cyaw	stability axis aerodynamic yawing moment coefficient
dirtrmx	directional trim input
dpnoz(DE10)	pitch nozzle position, deg
dynoz(DE20)	yaw nozzle position, deg
erthzdt	velocity along inertial negative z-axis, ft/sec
erthxic	initial position along inertial x-axis, ft
erthyic	initial position along inertial y-axis, ft
flapl(DE4)	symmetric pitch flap position, deg
gamma	flight path angle, deg
grs_thr	gross thrust, slug-ft/sec ²

latrmx	lateral trim input
lefol(DE2)	left leading-edge outboard flap position, deg
lefor(DE12)	right leading-edge outboard flap position, deg
lontrmx	longitudinal trim input
mach	mach number
net_thr	net thrust, slug-ft/sec ²
pbdot	body axis roll acceleration, deg/sec ²
pbic, pb	body axis roll rate, deg/sec
phidot	euler bank angular rate, deg/sec
phiic, phi	euler bank angle, deg
plain	power lever angle, deg
ps	stability axis roll rate, deg/sec
psidot	euler heading angular rate, deg/sec
psiic, psi	euler heading angle, deg
qbdot	body axis pitching acceleration, deg/sec ²
qbic, qb	body axis pitch rate, deg/sec
rbdot	body axis yaw acceleration, deg/sec ²
rbic, rb	body axis yaw rate, deg/sec
rs	stability axis yaw rate, deg/sec
ssdl(DE9)	left spoiler-slot deflector position, deg
ssdr(DE19)	right spoiler-slot deflector position, deg
taill(DE3)	left elevon position, deg
tailr(DE13)	right elevon position, deg
thetdot	euler pitch angular rate, deg/sec
thetaic, theta	euler pitch angle, deg
udot	acceleration along body x-axis, ft/sec ²
uic, u	velocity along body x-axis, ft/sec
vcas	calibrated airspeed, kts
veas	equivalent airspeed, kts
vel (velkts)	true airspeed, ft/sec (kts)
vic, v	velocity along body y-axis, ft/sec
vtotdot	rate of change of true airspeed, ft/sec ²
wdot	acceleration along body z-axis, ft/sec ²
wic, w	velocity along body z-axis, ft/sec
wtic	weight

ICE Vehicle (101-3)

The controller will be applied to the tailless fighter configuration developed under the Innovative Control Effectors (ICE) program. Configuration 101-3 of this program is shown in figure 1. The control effectors include elevons, pitch flaps, all moving tips, thrust vectoring, spoiler slot deflectors, and outboard leading edge flaps. The conventional control effectors are defined as the elevons, pitch flap, and leading edge flaps. The innovative control effectors are defined as the thrust vectoring, all moving tips and spoiler slot deflectors. Challenges associated with control using the all moving tips and spoiler slot deflectors include zero lower deflection limits, strong multi-axes effects and effector interactions (latter, not apparent in linear models).[1]

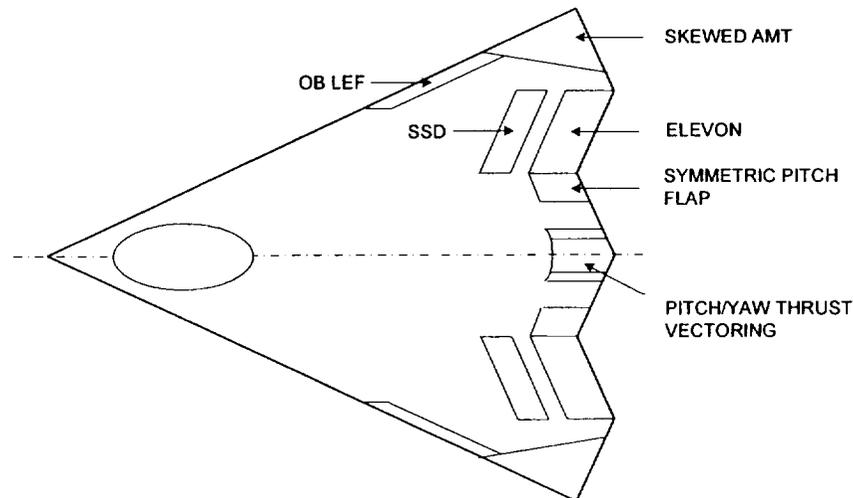


Figure 1. Model Layout

The full nonlinear simulation of this vehicle remains proprietary. Lockheed, however, has granted NASA Langley Research Center permission to supply Prof. Hess with a set of linear models spanning the subsonic flight envelope for his reconfigurable control work. The supplemental information provided in this package concerning mass properties, etc., has been accumulated from the open literature [1], [2], allowing the researcher to track variations in stability and control derivatives to pertinent increments in the aerodynamic data base. The ultimate aim is to integrate the proposed control in the actual nonlinear simulation at NASA Langley Research Center to further validate the methodology and compare the results with the current in-house effort. Any activity towards addressing stability/performance robustness issues as well as nonlinear simulation implementation issues prior to NASA involvement will increase the likelihood of a successful implementation.

Linear Models

The set of linear models provided corresponds to three types of trim conditions: 1g wings-level, steady level turn, and symmetric pull up/push down trim [3]. The corresponding flight conditions span a range of Mach [.3-.9], altitude [15-35k ft], and angle of attack. These models provide an adequate set of stability and control derivative variations associated with subsonic flight.

The pertinent trim variables are summarized in three MATLAB files: *trmmap_WL1G.mat*, *trmmap_ST.mat*, and *trmmap_PU.mat*. These trim file summaries catalog the linear models respectively found in subdirectories **lin_wng_lev_1g**, **lin_stdy_turn**, and **lin_pull_up**. The MATLAB files within them are labeled as follows

lin_wng_lev_1g_mX_hZ.mat
lin_stdy_turn_mX_aY_hZ.mat
lin_pull_up_mX_aY_hZ.mat

where

$X = 10 \cdot mach$
 $Y = alphaw$
 $Z = altic / 1000.$

Along with the matrices of the state space representation, each file contains trim values of state, input, and output (x_0 , u_0 , and y_0), and the variable names ($xnames$, $unames$, and $ynames$). Units are given above in the List of Variables. Actuator dynamics, position limits, and rate limits can be found in Table 1.

A few comments concerning the choice of trim conditions are in order. The loaded conditions of steady level turn and symmetric pull up/push down have been included to provide an independent range of angle of attack and Mach for a given altitude. This has been done to provide a better characterization of vehicle's aerodynamic database. Note, in 1g wings-level flight, for a given weight, speed, and altitude, there is only one angle of attack that will generate the necessary lift.

The ability to independently vary Mach and angle of attack, however, comes with a cost. In 1g wings-level flight, longitudinal and lateral/directional dynamics are decoupled and the gravity vector orientation relative to the trimmed vehicle is fixed. In symmetric pull up/push down trim, the longitudinal and lateral/directional dynamics are still decoupled, but the relative orientation of the gravity vector is changing (nonzero θ_{dot}). In the steady level turn, the relative orientation of the gravity vector is constant (zero ϕ_{dot} and θ_{dot}) but the longitudinal and lateral/directional dynamics are coupled. The effect of the aerodynamic database on the respective derivatives is available, however, since the data supplied in this package includes the vehicle's inertial properties and the corresponding constant components of angular rate and velocity. Dependence on Mach and angle of attack, for example, is available from both 1g wings-level and pull up/push down trim conditions. An additional dependence on side-slip angle is available from the level steady turn trim conditions at equivalent Mach and angle of attack values.

Table 1. Actuator Dynamics

	<i>Amplitude Limit</i>	<i>Rate Limit</i>
Elevon (taill/DE3), (tailr/DE13): $\frac{(40)(100)}{(s+40)(s+100)}$	± 30 deg	150 deg/sec
Symmetric Pitch Flap (pflap/DE4): $\frac{(40)(100)}{(s+40)(s+100)}$	± 30 deg	50 deg/sec
Skewed All Moving Tip (amtl/DE5), (amtr/DE15): $\frac{(40)(100)}{(s+40)(s+100)}$	[0, 60] deg	150 deg/sec
Spoiler-Slot Deflector (ssdl/DE9), (ssdr/DE19): $\frac{(40)(100)}{(s+40)(s+100)}$	[0, 60] deg	150 deg/sec
Outboard Leading Edge Flap (leflol/DE2), (leflor/DE12): $\frac{(17.828)(100)}{(s+17.828)(s+100)}$	± 40 deg	40 deg/sec
Pitch/Yaw Thrust Vectoring (dpnoz/DE10), (dynoz/DE20): $\frac{(37.186)(41.3)}{(s+37.186)(s+41.3)}$	± 15 deg	60 deg/sec
Power Lever Angle (plain)*		
Afterburner: $\frac{1}{.55s+1}$	[30, 90] deg	22 deg/sec
Afterburner out: $\frac{1}{.625s+1}$	[90, 127] deg	14 deg/sec

*dynamics currently not in proprietary simulation, but proposed for LaRC version

Nonlinear Simulation and Linear Models

Some features of the nonlinear simulation are considered in the context of the linear models generated. Components of the trimming forces and moments are discussed along with the impact of various aerodynamic increments on the stability and control derivatives for the flight conditions considered. The mass properties data and thrust-vectoring model are available in the open literature. Consider first a cursory look at the mathematical framework of the nonlinear vehicle description.

The ICE model is governed by the standard set of equations of motion for a flat-earth, rigid-body, symmetric aircraft [4].

$$\dot{u} = -q_b w_b + r_b v_b - g \sin \theta + \frac{X}{m} \quad (1a)$$

$$\dot{v} = -r_b u + p_b w + g \cos \theta \sin \phi + \frac{Y}{m} \quad (1b)$$

$$\dot{w} = -p_b v + q_b u + g \cos \theta \cos \phi + \frac{Z}{m} \quad (1c)$$

$$\dot{q}_b = \frac{1}{I_{yy}} \left\{ M - p_b r_b (I_{xx} - I_{zz}) + r_b^2 I_{xz} - p_b^2 I_{xz} \right\} \frac{180}{\pi} \quad (1d)$$

$$\begin{Bmatrix} \dot{p}_b \\ \dot{r}_b \end{Bmatrix} = \begin{bmatrix} I_{xx} & -I_{xz} \\ -I_{xz} & I_{zz} \end{bmatrix}^{-1} \begin{Bmatrix} L - q_b r_b (I_{zz} - I_{yy}) + p_b q_b I_{xz} \\ N - p_b q_b (I_{yy} - I_{zz}) - q_b r_b I_{xz} \end{Bmatrix} \frac{180}{\pi} \quad (1e)$$

$$\dot{\theta} = q_b \cos \phi - r_b \sin \phi \quad (1f)$$

$$\dot{\phi} = p_b + q_b \tan \theta \sin \phi + r_b \tan \theta \cos \phi \quad (1g)$$

The model constants are listed in Table 2 [1]. The force and moment vectors, $\hat{F}^T = [X \ Y \ Z]$ and $\hat{M}^T = [L \ M \ N]$, respectively, expand as

$$\begin{aligned} \hat{F} &= \hat{F}_a + \hat{F}_T \\ \hat{F}_a &= \hat{F}_{a,o} + \hat{F}_{a,dyn} + \hat{F}_{a,\delta} \\ \hat{F}_T &= \hat{F}_t + \hat{F}_{rd} \end{aligned} \quad (2)$$

$$\begin{aligned} \hat{M} &= \hat{M}_a + \hat{M}_T \\ \hat{M}_a &= \hat{M}_{a,o} + \hat{M}_{a,dyn} + \hat{M}_{a,\delta} \\ \hat{M}_T &= \hat{M}_t + \hat{M}_{rd} \end{aligned} \quad (3)$$

where subscripts ‘*a*’ and ‘*T*’ correspond to the aerodynamic and thrust related forces and moments. The aerodynamic force and moments are further decomposed into ‘*a,o*’ components of the baseline aircraft with no effectors deflected, ‘*a,dyn*’ components due to angular rates, and ‘*a, δ* ’ force and moment increments due to control deflection. The thrust force and moments are decomposed into ‘*t*’ components due to gross thrust and ‘*rd*’ components due to ram drag. Although it is unlikely that anything more than a first-order approximation of the aerodynamic data can be obtained from the trim files and linear models provided (maybe not with the decomposition above), it is of interest how these terms influence the stability and control derivatives particularly when characterizing uncertainty. In the coming subsections, the aerodynamic data build up is considered along with its influence on the linear models.

Table 2. Model Constants

Symbol	Description	Value
I_{xx}	x-axis moment of inertia	35,479 slug-ft ²
I_{yy}	y-axis moment of inertia	78,451 slug-ft ²
I_{zz}	z-axis moment of inertia	110,627 slug-ft ²
I_{xz}	xz-axes product of inertia	-525 slug-ft ²
S_{ref}	wing planform area	808.6 ft ²
\bar{c}	wing mean aerodynamic chord	28.75 ft
b	wing span	37.5 ft
m	Aircraft mass	1017.9 slug
W	Aircraft gross weight	32,750 lb
d_{TWL}	waterline distance between cg and thrust application point	-.417 ft**
d_{TFS}	fuselage station distance between cg and thrust application point	18.79 ft**
d_{DWL}	waterline distance between cg and ram drag application point	-0.3308 ft
d_{DFS}	fuselage station distance between cg and ram drag application point	-12.66 ft
d_{ACWL}	waterline distance between cg and accelerometer	-1.039 ft
d_{PSFS}	fuselage station distance between cg and pilot station	-13.55 ft
d_{ACFS}	fuselage distance between cg and accelerometer	-11.5008 ft

* relative to cg position of 38% of mean aerodynamic chord, nominal weight

** positive WL, FS distance is above and behind cg

Aerodynamic Forces and Moments

The total aerodynamic forces and moments at trim can be obtained from the trim file as follows

$$X_{a,s} = -\bar{q}S_{ref} \cdot c_{drag} \quad (4a)$$

$$Y_{a,s} = \bar{q}S_{ref} \cdot c_{side} \quad (4b)$$

$$Z_{a,s} = -\bar{q}S_{ref} \cdot c_{lift} \quad (4c)$$

$$L_{a,s} = \bar{q}S_{ref}b \cdot c_{roll} \quad (4d)$$

$$M_{a,s} = \bar{q}S_{ref}\bar{c} \cdot c_{pitch} \quad (4e)$$

$$N_{a,s} = \bar{q}S_{ref}b \cdot c_{yaw} \quad (4f)$$

These components include trim forces and moments due to the baseline vehicle's orientation and speed with respect to the air stream, due to any trim angular velocities (excluding inertial effects), and due to the trim control deflections. Note these components are expressed with respect to stability axis. Transforming these components to body axis as reflected in the equations of motion is straightforward. The decomposition of components into the corresponding 'a,o', 'a,dyn', 'a,delta' terms may not be so straightforward or possible. Some additional background information on the dynamic term is provided.

Rotary and Forced Oscillation Terms

The direct method in AIAA-88-4357 is used to add the rotational and oscillatory increments to the force and moment components for $\alpha < 30^\circ$. The effect of angular rates on the aerodynamic moments and forces is modeled using data from two sources: 1) a rotary balance test where the vehicle rotates about the velocity vector and 2) a forced oscillation test where the vehicle is forced to oscillate about each body axis.

First the angular velocity vector is equivalently expressed as

$$\begin{aligned}\hat{\omega} &= p_b \hat{j}_b + q_b \hat{j}_b + r_b \hat{k}_b = \omega_V \hat{i}_w + \hat{\omega}_{os} \\ \omega_V &= \hat{\omega} \cdot \hat{i}_w = p_b \cos \alpha \cos \beta + q_b \sin \beta + r_b \sin \alpha \cos \beta\end{aligned}\quad (5)$$

where \hat{i}_w is a unit vector along the velocity vector, ω_V and $\hat{\omega}_{os}$ are respectively the rotary (along the velocity vector) and oscillatory components of the angular velocity vector. The oscillatory component is further decomposed along the body axes as

$$\hat{\omega}_{os} = p_{os} \hat{i}_b + q_{os} \hat{j}_b + r_{os} \hat{k}_b \quad (6)$$

where

$$p_{os} = p_b - \omega_V \cos \alpha \cos \beta \quad (7a)$$

$$q_{os} = q_b - \omega_V \sin \beta \quad (7b)$$

$$r_{os} = r_b - \omega_V \sin \alpha \cos \beta. \quad (7c)$$

From these expressions, it is clear that this decomposition provides an equivalent representation of the angular velocity. The components of the oscillatory term and the rotary angular velocity are made non-dimensional as follows

$$\hat{\omega}'_{os} = \frac{bp_{os}}{2V} \hat{i}_b + \frac{\bar{c}q_{os}}{2V} \hat{j}_b + \frac{br_{os}}{2V} \hat{k}_b \quad (8a)$$

$$\omega'_V = \frac{b\omega_V}{2V}. \quad (8b)$$

The direct method uses the non-dimensional terms to define dynamic effect of angular rates on the forces and moments.

$$\hat{F}_{a,dyn} = \hat{F}_{a,ro}(M, \alpha, \beta, \omega'_V) \quad (9a)$$

$$\hat{M}_{a,dyn} = \hat{M}_{a,ro}(M, \alpha, \beta, \omega'_V) + \Pi_{os}(M, \alpha) \hat{\omega}'_{os} \quad (9b)$$

Subscripts 'a,ro' and 'os' denote respectively the rotary and oscillatory components. The matrix Π_{os} consists of the following oscillatory dynamic derivatives

$$\Pi_{os} = \begin{bmatrix} L_{p,os} & 0 & L_{r,os} \\ 0 & M_{q,os} & 0 \\ N_{p,os} & 0 & N_{r,os} \end{bmatrix}. \quad (10)$$

The rotary term is zero whenever ω_V is zero, a result of either $\hat{\omega} = 0$ or $\hat{\omega}$ is perpendicular to \hat{i}_w . The former occurs in 1g wing-level trim and the latter occurs in both the symmetric pull up/push down and steady level turn. Consequently, the trim values of force and moments do not have any rotary component. The rotary components do influence the derivatives. It should be mentioned that the Mach dependence in the rotary terms exclusively resides with dynamic pressure, \bar{q} , used to dimensional-ize the corresponding force and moment coefficients. The axial force rotary component is zero.

At trim the steady effects due to constant angular rates, if any, result from the oscillatory component, or

$$\hat{F}_{a,dyn,trim} = 0 \quad (11a)$$

$$\hat{M}_{a,dyn,trim} = \Pi_{os}(M, \alpha) \hat{\omega}'_{os}. \quad (11b)$$

The aerodynamic trim force and moment components, then, take on the following form

1-g, wings level: $p'_{os} = q'_{os} = r'_{os} = \beta = 0$

$$\hat{F}_{a,trim} = \hat{F}_{a,o}(M, \alpha, \beta) + \hat{F}_{a,\delta}(M, \alpha, \delta) \quad (12a)$$

$$\hat{M}_{a,trim} = \hat{M}_{a,o}(M, \alpha, \beta) + \hat{M}_{a,\delta}(M, \alpha, \delta) \quad (12b)$$

pull up/push down: $p'_{os} = r'_{os} = \beta = 0$, $q'_{os} = \bar{c}q_b/2V$

$$\hat{F}_{a,trim} = \hat{F}_{a,o}(M, \alpha, \beta) + \hat{F}_{a,\delta}(M, \alpha, \delta) \quad (13a)$$

$$\hat{M}_{a,trim} = \hat{M}_{a,o}(M, \alpha, \beta) + \begin{bmatrix} 0 \\ M_{q,os}q'_{os} \\ 0 \end{bmatrix} + \hat{M}_{a,\delta}(M, \alpha, \delta) \quad (13b)$$

steady level turn: $p'_{os} = bp_b/2V$, $q'_{os} = \bar{c}q_b/2V$, $r'_{os} = br_b/2V$, $\beta = \text{constant}$

$$\hat{F}_{a,trim} = \hat{F}_{a,o}(M, \alpha, \beta) + \hat{F}_{a,\delta}(M, \alpha, \delta) \quad (14a)$$

$$\hat{M}_{a,trim} = \hat{M}_{a,o}(M, \alpha, \beta) + \begin{bmatrix} L_{p,os}p'_{os} + L_{r,os}r'_{os} \\ M_{q,os}q'_{os} \\ N_{p,os}p'_{os} + N_{r,os}r'_{os} \end{bmatrix} + \hat{M}_{a,\delta}(M, \alpha, \delta). \quad (14b)$$

If a first-order approximation is assumed for $\hat{F}_{a,\delta}$ and $\hat{M}_{a,\delta}$ about $\delta = 0$ (trim effector deflections are small), so

$$\hat{F}_{a,\delta} \cong F'_{a,\delta} \delta \quad (15a)$$

$$\hat{M}_{a,\delta} \cong M'_{a,\delta} \delta, \quad (15b)$$

$\hat{F}_{a,o}$ and $\hat{M}_{a,o}$ can be approximated for 1g-level flight. Note, $F'_{a,\delta}$ and $M'_{a,\delta}$ are available from the B matrix. Altitude effects can be removed by normalizing with \bar{q} . It will be shown that for both 1g-level flight and symmetric pull up, the term $M_{q,os}$ is explicit in the A-matrix. As a result, $\hat{F}_{a,o}$ and $\hat{M}_{a,o}$ can be approximated for symmetric pull up. Side-slip angle effects, only on $\hat{F}_{a,o}$, can be determined from the difference obtained in the term for the symmetric pull up and steady level turn at the same (α, M) flight conditions. Side-slip angle effects on $\hat{M}_{a,o}$ (except for the pitching component) as well as identifying the remaining dynamic derivatives may not, however, be possible.

Stability Derivatives and the Nonlinear Aerodynamic Database

Let us assume that A' is the linear system's A-matrix with all the differential contributions due to inertial terms (e.g., wq_b , uq_b , $I_{xz}p_bq_b$, $(I_{zz} - I_{yy})_bq_b$) and gravity terms removed. Only the aerodynamic contribution remains in A' . Pursuing a construction similar to that use in [4], the differential forces and moments determine the linear models, i.e., what is leftover in A'

$$\begin{aligned} d\hat{F}_a - \frac{\partial \hat{F}_a}{\partial \delta} d\delta &= \frac{\partial \hat{F}_a}{\partial \alpha} d\alpha + \frac{\partial \hat{F}_a}{\partial \beta} d\beta + \frac{\partial \hat{F}_a}{\partial M} dM + \frac{\partial \hat{F}_a}{\partial p_b} dp_b + \frac{\partial \hat{F}_a}{\partial q_b} dq_b + \frac{\partial \hat{F}_a}{\partial r_b} dr_b \\ d\hat{M}_a - \frac{\partial \hat{M}_a}{\partial \delta} d\delta &= \frac{\partial \hat{M}_a}{\partial \alpha} d\alpha + \frac{\partial \hat{M}_a}{\partial \beta} d\beta + \frac{\partial \hat{M}_a}{\partial M} dM + \frac{\partial \hat{M}_a}{\partial p_b} dp_b + \frac{\partial \hat{M}_a}{\partial q_b} dq_b + \frac{\partial \hat{M}_a}{\partial r_b} dr_b \end{aligned} \quad (16)$$

where all partial derivatives are evaluated at the trim condition. The terms on the right influence the A-matrix whereas the second term (negative of) on the left corresponds to the B-matrix. Substituting the differentials $d\alpha$, $d\beta$, and dM for expressions in table 3 expressed in terms of du , dv , and dw provide the derivatives used to construct A. Note when the trim β (v) is zero as in 1-g level flight or symmetric pull up, dM influences only the du dw columns of A, whereas $d\beta$ influences only the dv column of A. In a steady level turn, dM and $d\beta$ influence all columns of A corresponding to translational velocity. From its definition, $d\alpha$ influences only the du dw columns of A regardless of the trim.

It should be mentioned that the columns corresponding to du and dw are also influenced by the differential force and moments due to thrust, dF_T and dM_T , since the application of ram drag is dependent on angle of attack as will be shown later.

Table 3. Transformation to translational velocity

$$\begin{aligned}
 d\alpha &= -w(u^2 + w^2)^{-1} du + u(u^2 + w^2)^{-1} dw \\
 d\beta &= -\frac{uv}{V^2}(u^2 + w^2)^{-1/2} du + \frac{1}{V^2}(u^2 + w^2)^{1/2} dv - \frac{wv}{V^2}(u^2 + w^2)^{-1/2} dw \\
 dM &= \frac{1}{a}\left(\frac{u}{V}\right)du + \frac{1}{a}\left(\frac{v}{V}\right)dv + \frac{1}{a}\left(\frac{w}{V}\right)dw
 \end{aligned}$$

*all values at trim; a is the speed of sound.

The contribution of the baseline aerodynamics and the increment due to control deflection

$$\begin{aligned}
 d(\hat{F}_{a,o} + \hat{F}_{a,\delta}) - \frac{\partial \hat{F}_{a,\delta}}{\partial \delta} d\delta &= \frac{\partial(\hat{F}_{a,o} + \hat{F}_{a,\delta})}{\partial \alpha} d\alpha + \frac{\partial \hat{F}_{a,o}}{\partial \beta} d\beta + \frac{\partial(\hat{F}_{a,o} + \hat{F}_{a,\delta})}{\partial M} dM \\
 d(\hat{M}_{a,o} + \hat{M}_{a,\delta}) - \frac{\partial \hat{M}_{a,\delta}}{\partial \delta} d\delta &= \frac{\partial(\hat{M}_{a,o} + \hat{M}_{a,\delta})}{\partial \alpha} d\alpha + \frac{\partial \hat{M}_{a,o}}{\partial \beta} d\beta + \frac{\partial(\hat{M}_{a,o} + \hat{M}_{a,\delta})}{\partial M} dM
 \end{aligned} \tag{17}$$

does not influence the columns of A corresponding to dp_b , dq_b , and dr_b . The dynamic aerodynamic increment

$$\begin{aligned}
 d\hat{F}_{a,dyn} &= \frac{\partial \hat{F}_{a,dyn}}{\partial \alpha} d\alpha + \frac{\partial \hat{F}_{a,dyn}}{\partial \beta} d\beta + \frac{\partial \hat{F}_{a,dyn}}{\partial M} dM + \frac{\partial \hat{F}_{a,dyn}}{\partial p_b} dp_b + \frac{\partial \hat{F}_{a,dyn}}{\partial q_b} dq_b + \frac{\partial \hat{F}_{a,dyn}}{\partial r_b} dr_b \\
 d\hat{M}_{a,dyn} &= \frac{\partial \hat{M}_{a,dyn}}{\partial \alpha} d\alpha + \frac{\partial \hat{M}_{a,dyn}}{\partial \beta} d\beta + \frac{\partial \hat{M}_{a,dyn}}{\partial M} dM + \frac{\partial \hat{M}_{a,dyn}}{\partial p_b} dp_b + \frac{\partial \hat{M}_{a,dyn}}{\partial q_b} dq_b + \frac{\partial \hat{M}_{a,dyn}}{\partial r_b} dr_b
 \end{aligned} \tag{18}$$

potentially affects all six columns. Next, expansions of the partials in equation 18 are developed. The result will be the dynamic aerodynamic increment has no influence on the derivatives corresponding to translational velocities in 1g-level flight. The influence of the dynamic term expands, however, as trim pitch rate becomes nonzero for symmetric pull up flight, and the rest of angular rate components and side-slip angle become nonzero in the level steady turn.

For partial derivatives taken with respect to $x_i = \alpha, \beta$, or M , the initial form is the same

$$\frac{\partial \hat{F}_{a,dyn}}{\partial x_i} = \frac{\partial \hat{F}_{a,ro}}{\partial x_i} + \frac{\partial \hat{F}_{a,ro}}{\partial \omega'_V} \frac{\partial \omega'_V}{\partial x_i} \tag{19a}$$

$$\frac{\partial \hat{M}_{a,dyn}}{\partial x_i} = \frac{\partial \hat{M}_{a,ro}}{\partial x_i} + \frac{\partial \hat{M}_{a,ro}}{\partial \omega'_V} \frac{\partial \omega'_V}{\partial x_i} + \frac{\partial \Pi_{os}}{\partial x_i} \hat{\omega}'_{os} + \Pi_{os} \left(\frac{\partial \hat{\omega}'_{os}}{\partial x_i} + \frac{\partial \hat{\omega}'_{os}}{\partial \omega_V} \frac{\partial \omega_V}{\partial x_i} \right) \tag{19b}$$

Since the rotary component is zero whenever ω'_V is zero (true for all trim), the first term in (19a-b) is always zero. The second term is zero for $x_i = M$ in both equations, since

$$\frac{\partial \omega'_V}{\partial M} = -\frac{a}{V} \omega'_V = 0. \quad (20)$$

The third term in (19b) is zero when $x_i = \beta$. The components of (19a-b) can be found in tables 4 and 5, where the following shorthand is used. $L_{\omega'_v,ro}, M_{\omega'_v,ro}, N_{\omega'_v,ro}$ are the components of $\partial \hat{M}_{a,ro} / \partial \omega'_V$; $X_{\omega'_v,ro}, Y_{\omega'_v,ro}, Z_{\omega'_v,ro}$ are the components of $\partial \hat{F}_{a,ro} / \partial \omega'_V$; $L_{p,os,\alpha}$ and $L_{p,os,M}$ are the partials of $L_{p,os}$ with respect to α and M , respectively; s_α and c_β are $\sin(\alpha)$ and $\cos(\beta)$, respectively; stability axis roll and yaw rate are denoted as p_s and r_s , respectively.

Table 4. Dynamic Force Partial Derivatives with respect to (α, β, M)

$\frac{\partial X_{dyn}}{\partial \alpha} = 0$	$\frac{\partial X_{dyn}}{\partial \beta} = 0$	$\frac{\partial X_{dyn}}{\partial M} = 0$
$\frac{\partial Y_{dyn}}{\partial \alpha} = \frac{b}{2V} Y_{\omega'_v,ro} (r_s c_\beta)$	$\frac{\partial Y_{dyn}}{\partial \beta} = \frac{b}{2V} Y_{\omega'_v,ro} (q_b c_\beta - p_s s_\beta)$	$\frac{\partial Y_{dyn}}{\partial M} = 0$
$\frac{\partial Z_{dyn}}{\partial \alpha} = \frac{b}{2V} Z_{\omega'_v,ro} (r_s c_\beta)$	$\frac{\partial Z_{dyn}}{\partial \beta} = \frac{b}{2V} Z_{\omega'_v,ro} (q_b c_\beta - p_s s_\beta)$	$\frac{\partial Z_{dyn}}{\partial M} = 0$

Table 5. Dynamic Moment Partial Derivatives with respect to (α, β, M)

$$\frac{\partial L_{dyn}}{\partial \alpha} = \frac{b}{2V} \left(L_{\omega'_v,ro} r_s c_\beta + L_{p,os\alpha} p_b + L_{r,os\alpha} r_b - r_s \left(L_{p,os} c_\alpha c_\beta^2 + L_{r,os} s_\alpha c_\beta^2 \right) \right)$$

$$\frac{\partial M_{dyn}}{\partial \alpha} = \frac{b}{2V} M_{\omega'_v,ro} r_s c_\beta + \frac{\bar{c}}{2V} \left(M_{q,os,\alpha} q_b - \frac{r_s}{2} M_{q,os} s_{2\beta} \right)$$

$$\frac{\partial N_{dyn}}{\partial \alpha} = \frac{b}{2V} \left(N_{\omega'_v,ro} r_s c_\beta + N_{p,os\alpha} p_b + N_{r,os\alpha} r_b - r_s \left(N_{p,os} c_\alpha c_\beta^2 + N_{r,os} s_\alpha c_\beta^2 \right) \right)$$

$$\frac{\partial L_{dyn}}{\partial \beta} = \frac{b}{2V} \left(L_{\omega'_v,ro} (q_b c_\beta - p_s s_\beta) + \left(L_{p,os} \left(\frac{1}{2} p_s c_\alpha s_{2\beta} - q_b c_\alpha c_\beta^2 \right) + L_{r,os} \left(\frac{1}{2} p_s s_\alpha s_{2\beta} - q_b s_\alpha c_\beta^2 \right) \right) \right)$$

$$\frac{\partial M_{dyn}}{\partial \beta} = \frac{b}{2V} M_{\omega'_v,ro} (q_b c_\beta - p_s s_\beta) + \frac{\bar{c}}{2V} \left(M_{q,os} \left(p_s s_\beta^2 - \frac{1}{2} q_b s_{2\beta} \right) \right)$$

$$\frac{\partial N_{dyn}}{\partial \beta} = \frac{b}{2V} \left(N_{\omega'_v,ro} (q_b c_\beta - p_s s_\beta) + \left(N_{p,os} \left(\frac{1}{2} p_s c_\alpha s_{2\beta} - q_b c_\alpha c_\beta^2 \right) + N_{r,os} \left(\frac{1}{2} p_s s_\alpha s_{2\beta} - q_b s_\alpha c_\beta^2 \right) \right) \right)$$

$$\frac{\partial L_{dyn}}{\partial M} = \frac{b}{2V} \left(\left(L_{p,os,M} - \frac{a}{V} L_{p,os} \right) p_b + \left(L_{r,os,M} - \frac{a}{V} L_{r,os} \right) r_b \right)$$

$$\frac{\partial M_{dyn}}{\partial M} = \frac{\bar{c}}{2V} \left(M_{q,os,M} - \frac{a}{V} M_{q,os} \right) q_b$$

$$\frac{\partial N_{dyn}}{\partial M} = \frac{b}{2V} \left(\left(N_{p,os,M} - \frac{a}{V} N_{p,os} \right) p_b + \left(N_{r,os,M} - \frac{a}{V} N_{r,os} \right) r_b \right)$$

In 1-g level flight, all terms in tables 4 & 5 vanish due a zero angular velocity vector at trim. For symmetric pull up, there are nonzero entries for $\partial \hat{M}_{a,dyn} / \partial \beta$ and the components $\partial M_{dyn} / \partial \alpha$ and $\partial M_{dyn} / \partial M$ due to the nonzero trim q_b . For level steady turn, all expressions not specified as zero are nonzero.

For partial derivatives of equation 18 taken with respect to angular velocity, the form is

$$\frac{\partial \hat{F}_{a,dyn}}{\partial \hat{\omega}} = \frac{\partial \hat{F}_{a,ro}}{\partial \omega'_v} \frac{\partial \omega'_v}{\partial \hat{\omega}} \quad (21a)$$

$$\frac{\partial \hat{M}_{a,dyn}}{\partial \hat{\omega}} = \frac{\partial \hat{M}_{a,ro}}{\partial \omega'_v} \frac{\partial \omega'_v}{\partial \hat{\omega}} + \Pi_{os} \left(\frac{\partial \hat{\omega}'_{os}}{\partial \hat{\omega}} + \frac{\partial \hat{\omega}'_{os}}{\partial \omega'_v} \frac{\partial \omega'_v}{\partial \hat{\omega}} \right) \quad (21b)$$

Tables 6 and 7 contain the nine components associated with each equation. The shorthand used in tables 4 and 5 is again used here.

Table 6. Dynamic Force Partial Derivatives with respect to (p_b, q_b, r_b)

$\frac{\partial X}{\partial p_b} = \frac{\partial X_{dyn}}{\partial p_b} = 0$	$\frac{\partial X}{\partial q_b} = \frac{\partial X_{dyn}}{\partial q_b} = 0$	$\frac{\partial X}{\partial r_b} = \frac{\partial L_{dyn}}{\partial r_b} = 0$
$\frac{\partial Y}{\partial p_b} = \frac{\partial Y_{dyn}}{\partial p_b} = \frac{b}{2V} Y_{\omega'_{v,ro}} c_{\alpha} c_{\beta}$	$\frac{\partial Y}{\partial q_b} = \frac{\partial Y_{dyn}}{\partial q_b} = \frac{b}{2V} Y_{\omega'_{v,ro}} s_{\beta}$	$\frac{\partial Y}{\partial r_b} = \frac{\partial Y_{dyn}}{\partial r_b} = \frac{b}{2V} Y_{\omega'_{v,ro}} s_{\alpha} c_{\beta}$
$\frac{\partial Z}{\partial p_b} = \frac{\partial Z_{dyn}}{\partial p_b} = \frac{b}{2V} Z_{\omega'_{v,ro}} c_{\alpha} c_{\beta}$	$\frac{\partial Z}{\partial q_b} = \frac{\partial Z_{dyn}}{\partial q_b} = \frac{b}{2V} Z_{\omega'_{v,ro}} s_{\beta}$	$\frac{\partial Z}{\partial r_b} = \frac{\partial Z_{dyn}}{\partial r_b} = \frac{b}{2V} Z_{\omega'_{v,ro}} s_{\alpha} c_{\beta}$

Table 7. Dynamic Moment Partial Derivatives with respect to (p_b, q_b, r_b)

$$\begin{aligned} \frac{\partial L}{\partial p_b} &= \frac{\partial L_{dyn}}{\partial p_b} = \frac{b}{2V} \left(L_{\omega'_{v,ro}} c_{\alpha} c_{\beta} + L_{p,os} \left(1 - c_{\alpha}^2 c_{\beta}^2 \right) - \frac{1}{2} L_{r,os} s_{2\alpha} c_{\beta}^2 \right) \\ \frac{\partial M}{\partial p_b} &= \frac{\partial M_{dyn}}{\partial p_b} = \frac{b}{2V} M_{\omega'_{v,ro}} c_{\alpha} c_{\beta} - \frac{\bar{c}}{4V} M_{q,os} c_{\alpha} s_{2\beta} \\ \frac{\partial N}{\partial p_b} &= \frac{\partial N_{dyn}}{\partial p_b} = \frac{b}{2V} \left(N_{\omega'_{v,ro}} c_{\alpha} c_{\beta} + N_{p,os} \left(1 - c_{\alpha}^2 c_{\beta}^2 \right) - \frac{1}{2} N_{r,os} s_{2\alpha} c_{\beta}^2 \right) \\ \frac{\partial L}{\partial q_b} &= \frac{\partial L_{dyn}}{\partial q_b} = \frac{b}{2V} \left(L_{\omega'_{v,ro}} s_{\beta} - \frac{1}{2} \left(L_{p,os} s_{2\beta} c_{\alpha} + L_{r,os} s_{2\beta} s_{\alpha} \right) \right) \\ \frac{\partial M}{\partial q_b} &= \frac{\partial M_{dyn}}{\partial q_b} = \frac{b}{2V} M_{\omega'_{v,ro}} s_{\beta} + \frac{\bar{c}}{2V} M_{q,os} \left(1 - s_{\beta}^2 \right) \\ \frac{\partial N}{\partial q_b} &= \frac{\partial N_{dyn}}{\partial q_b} = \frac{b}{2V} \left(N_{\omega'_{v,ro}} s_{\beta} - \frac{1}{2} \left(N_{p,os} c_{\alpha} s_{2\beta} + N_{r,os} s_{\alpha} s_{2\beta} \right) \right) \\ \frac{\partial L}{\partial r_b} &= \frac{\partial L_{dyn}}{\partial r_b} = \frac{b}{2V} \left(L_{\omega'_{v,ro}} s_{\alpha} c_{\beta} - \frac{1}{2} L_{p,os} s_{2\alpha} c_{\beta}^2 + L_{r,os} \left(1 - s_{\alpha}^2 c_{\beta}^2 \right) \right) \\ \frac{\partial M}{\partial r_b} &= \frac{\partial M_{dyn}}{\partial r_b} = \frac{b}{2V} M_{\omega'_{v,ro}} s_{\alpha} c_{\beta} - \frac{\bar{c}}{4V} M_{q,os} s_{\alpha} s_{2\beta} \\ \frac{\partial N}{\partial r_b} &= \frac{\partial N_{dyn}}{\partial r_b} = \frac{b}{2V} \left(N_{\omega'_{v,ro}} s_{\alpha} c_{\beta} - \frac{1}{2} N_{p,os} s_{2\alpha} c_{\beta}^2 + N_{r,os} \left(1 - s_{\alpha}^2 c_{\beta}^2 \right) \right) \end{aligned}$$

To be noted, when $\beta = 0$ as in 1g-wings level and symmetric pull up flight, all partials with respect to q_b vanish except

$$\frac{\partial M}{\partial q_b} = \frac{\bar{c}}{2V} M_{q,os}. \quad (22)$$

For these two cases, the longitudinal pitch rate (perturbation) does not effect the lateral/directional moments. However, the lateral/directional roll and yaw rate (perturbations) do influence (albeit small from observation) the pitching moment through the rotary term. It appears that both the rotary and oscillatory components influence the lateral/directional moments in response to perturbations of the lateral/directional angular rate components. In the steady level turn, the longitudinal pitch rate (perturbation) does effect the lateral/directional moments providing some of the aerodynamic coupling at this trim condition. The rest of the aerodynamic coupling arises from the expressions in tables 3 and 4.

To close this discussion on the source of the linear models, the thrust-vectoring model is presented followed by the trim control used.

Thrust Equations

Thrust vectoring is applied to both the pitch and yaw axes. The thrust force and moment components may be expressed as:

$$X_t = T_g \cos(\delta_{ip}) \cos(\delta_{iy}) \quad (23a)$$

$$Y_t = T_g \cos(\delta_{ip}) \sin(\delta_{iy}) \quad (23b)$$

$$Z_t = -T_g \sin(\delta_{ip}) \quad (23c)$$

$$L_t = -l_z Y_t \quad (23d)$$

$$M_t = l_x Z_t + l_z X_t \quad (23e)$$

$$N_t = -l_x Y_t \quad (23f)$$

where T_g is gross thrust (lb), δ_{ip} (dpnoz) and δ_{iy} (dynoz) are the pitch and yaw nozzle deflections, respectively, and where l_z and l_x are respectively the distances (ft) below and behind the cg for the thrust application point. In terms of variables defined above, $T_g = \text{grs_thr}$; $l_z = -d_{TWL}$; $l_x = d_{TFS}$.

Force and moments due to ram drag, D_{ram} (lb), are expressed (best guess) as

$$D_{ram} = T_g - T_{net} \quad (24)$$

$$X_{rd} = -D_{ram} \cos(\alpha) \quad (25a)$$

$$Y_{rd} = 0 \quad (25b)$$

$$Z_{rd} = -D_{ram} \sin(\alpha) \quad (25c)$$

$$L_{rd} = 0 \quad (25d)$$

$$M_{rd} = -l_{x,rd} Z_{rd} + l_{z,rd} X_{rd} \quad (25e)$$

$$N_{rd} = 0 \quad (25f)$$

where T_{net} is net thrust, and where $l_{x,rd}$ and $l_{z,rd}$ are the distances(ft) ahead and below cg for the ram drag application point. From the trim file and Table 1.0, $T_{net} = \text{net_thr}$; $l_{x,rd} = -d_{DFS}$; $l_{z,rd} = -d_{DWL}$.

The differential force and moments due to thrust are used in linear models.

$$d\hat{F}_T = d\hat{F}_t + d\hat{F}_{rd} = \frac{\partial \hat{F}_t}{\partial \delta_{ip}} d\delta_{ip} + \frac{\partial \hat{F}_t}{\partial \delta_{ty}} d\delta_{ty} + \frac{\partial \hat{F}_{rd}}{\partial \alpha} d\alpha \quad (26)$$

$$d\hat{M}_T = d\hat{M}_t + d\hat{M}_{rd} = \frac{\partial \hat{M}_t}{\partial \delta_{ip}} d\delta_{ip} + \frac{\partial \hat{M}_t}{\partial \delta_{ty}} d\delta_{ty} + \frac{\partial \hat{M}_{rd}}{\partial \alpha} d\alpha \quad (27)$$

Equations 23 and 25 determine the partials above. The first two partial derivatives of equations 26 and 27 determine the columns of B corresponding to thrust vectoring. The third modifies the du, dw columns of the A-matrix.

Trim control used in linear models

Lastly, three inputs, 'lontrim', 'lattrim', and 'dirtrim' drive the control effectors to respectively trim the vehicle about the longitudinal, lateral, and directional axes.

$$elevl = lontrim + lattrim$$

$$elevr = lontrim - lattrim$$

$$pflapx = 0.1111 \cdot lontrim$$

$$leflox = 0.07111 \cdot (lontrim + lattrim)$$

$$lefrox = 0.07111 \cdot (lontrim - lattrim)$$

$$dpnoz = 0.16 \cdot lontrim$$

$$dynoz = 0.16 \cdot dirtrim$$

$$ssdl = ssdr = amtl = amtr = 0$$

For each effector, the weighting used is the squared value of the effector's rate limit normalized with the elevon's rate limit. Due to difficulties with the trimming routine, the unilateral controls (those restricted to positive deflections) were not used in the trim. The linear models for the four pertinent flight conditions are presented next.

Linear Models

State space representations for linearized ICE models for the flight conditions of interest for the design challenge are given on the following pages. The state, output and input vectors in these linear models are defined as:

$$x^T = [u \ w \ qb \ \theta \ v \ pb \ rb \ \phi]$$

$$y = [vel \ \alpha \ qb \ \theta \ \beta \ ps \ rs \ \phi \ axcg \ aycg \ azcg]$$

$$u = [dE3 \ dE13 \ dE4 \ dE5 \ dE15 \ dE9 \ dE19 \ dE2 \ dE12 \ dE10 \ dE20]$$

Mach No. = 0.3, Altitude = 15,000 ft (wings level)

a =

5.9570e-003	3.1987e-002	-1.3310e+000	-5.4507e-001	-1.2579e-003	0	0	0
-6.3377e-002	-5.8625e-001	5.3737e+000	-1.3501e-001	-1.9032e-003	2.3209e-004	5.7485e-005	0
-5.2181e-002	8.1704e-002	-5.9437e-001	0	1.1259e-002	1.7539e-003	4.3442e-004	0
0	0	1.0000e+000	0	0	0	0	0
3.3216e-007	8.2150e-008	0	0	1.1094e-002	1.3318e+000	-5.3735e+000	5.4507e-001
8.5005e-006	2.1012e-006	0	0	-1.2207e+000	-6.0701e-001	3.9515e-001	0
-2.9015e-006	-7.1571e-007	0	0	-1.2626e-001	2.9703e-003	2.0646e-002	0
0	0	0	0	0	1.0000e+000	2.4768e-001	0

>> b

b =

Columns 1 through 8

-5.9693e-003	-5.9693e-003	-4.0856e-003	-1.4174e-002	-1.4174e-002	-2.5238e-002	-2.5238e-002	1.1500e-002
-2.6797e-001	-2.6797e-001	-2.3059e-001	-8.4039e-002	-8.4039e-002	1.1122e-001	1.1122e-001	9.7043e-002
-1.3850e+000	-1.3850e+000	-1.2023e+000	-3.5207e-001	-3.5207e-001	2.6650e-001	2.6650e-001	2.3341e-001
0	0	0	0	0	0	0	0
-9.1236e-003	9.1236e-003	0	-2.4849e-004	2.4849e-004	2.1933e-002	-2.1933e-002	-2.8901e-002
2.4117e+000	-2.4117e+000	0	5.3690e-001	-5.3690e-001	-1.1963e+000	1.1963e+000	-9.8532e-001
3.7657e-002	-3.7657e-002	0	-5.6093e-002	5.6092e-002	-1.2350e-001	1.2350e-001	1.8590e-001
0	0	0	0	0	0	0	0

Columns 9 through 11

1.1500e-002	-1.1805e-005	0
9.7043e-002	-1.2395e-001	0
2.3341e-001	-1.7312e+000	0
0	0	0
2.8901e-002	0	1.2395e-001
9.8532e-001	0	-6.6733e-002
-1.8590e-001	0	-1.2273e+000
0	0	0

>> c

c =

9.7067e-001	2.4042e-001	0	0	0	0	0	0
-4.3428e-002	1.7534e-001	0	0	0	0	0	0
0	0	1.0000e+000	0	0	0	0	0
0	0	0	1.0000e+000	0	0	0	0
0	0	0	0	1.8063e-001	0	0	0
0	0	0	0	0	9.7067e-001	2.4042e-001	0
0	0	0	0	0	-2.4042e-001	9.7067e-001	0
0	0	0	0	0	0	0	1.0000e+000
1.8515e-004	9.9420e-004	0	0	-3.9096e-005	0	0	0
1.0324e-008	2.5533e-009	0	0	3.4482e-004	2.7166e-005	6.7286e-006	0
1.9698e-003	1.8221e-002	0	0	5.9153e-005	-7.2136e-006	-1.7867e-006	0

>> d

d =

Columns 1 through 8

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
-1.8553e-004	-1.8553e-004	-1.2698e-004	-4.4053e-004	-4.4053e-004	-7.8444e-004	-7.8444e-004	3.5743e-004	0
-2.8357e-004	2.8357e-004	0	-7.7233e-006	7.7233e-006	6.8170e-004	-6.8170e-004	-8.9826e-004	0
8.3287e-003	8.3287e-003	7.1669e-003	2.6120e-003	2.6120e-003	-3.4567e-003	-3.4567e-003	-3.0162e-003	0

Columns 9 through 11

0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
3.5743e-004	-3.6690e-007	0
8.9826e-004	0	3.8524e-003
-3.0162e-003	3.8524e-003	0

Mach No. = 0.6, Altitude = 25,000 ft (wings level)

a =

```
-4.4568e-003  6.4410e-002 -1.1655e+000 -5.5816e-001 -3.2481e-003      0      0      0
-5.2647e-002 -5.5615e-001  1.0575e+001 -6.1516e-002  1.2911e-002 -9.1850e-005 -1.0123e-005      0
-1.0290e-002 -1.7800e-001 -5.8883e-001      0  2.8958e-002 -3.7347e-003 -4.1161e-004      0
      0      0  1.0000e+000      0      0      0      0      0
      0  4.2110e-018      0      0 -3.4652e-003  1.1675e+000 -1.0575e+001  5.5816e-001
  1.5401e-019  4.6198e-019      0      0 -8.8575e-001 -8.0642e-001  1.7919e-001      0
-1.0408e-017 -3.1220e-017      0      0 -1.4300e-001  7.6043e-003 -6.1956e-003      0
      0      0      0      0      0  1.0000e+000  1.1021e-001      0
```

>> b

b =

Columns 1 through 8

```
-9.4029e-003 -9.4029e-003 -2.7374e-002 -1.7097e-002 -1.7097e-002 -9.7633e-002 -9.7633e-002 -8.3672e-003
-5.9531e-001 -5.9531e-001 -3.0032e-001 -3.4040e-001 -3.4040e-001  2.3746e-001  2.3746e-001  4.6687e-003
-4.2251e+000 -4.2251e+000 -2.0481e+000 -1.6221e+000 -1.6221e+000  7.9212e-001  7.9212e-001 -1.0211e-001
      0      0      0      0      0      0      0      0
  4.5640e-003 -4.5640e-003      0  9.6242e-003 -9.6242e-003  5.6742e-002 -5.6742e-002  1.2412e-002
  5.9008e+000 -5.9008e+000      0  2.5672e+000 -2.5673e+000 -2.7252e+000  2.7252e+000 -6.1473e-002
-4.2362e-002  4.2362e-002      0 -1.5954e-001  1.5958e-001 -4.7640e-001  4.7640e-001 -6.8770e-002
      0      0      0      0      0      0      0      0
```

Columns 9 through 11

```
-8.3672e-003 -8.4481e-005      0
  4.6687e-003 -7.7582e-002      0
-1.0211e-001 -1.0836e+000      0
      0      0      0
-1.2412e-002      0  7.7582e-002
  6.1473e-002      0 -4.1771e-002
  6.8770e-002      0 -7.6822e-001
      0      0      0
```

>> c

c =

9.9398e-001	1.0955e-001	0	0	0	0	0	0	0	0
-1.0297e-002	9.3426e-002	0	0	0	0	0	0	0	0
0	0	1.0000e+000	0	0	0	0	0	0	0
0	0	0	1.0000e+000	0	0	0	0	0	0
0	0	0	0	9.3992e-002	0	0	0	0	0
0	0	0	0	0	9.9398e-001	1.0955e-001	0	0	0
0	0	0	0	0	0	-1.0955e-001	9.9398e-001	0	0
0	0	0	0	0	0	0	0	1.0000e+000	0
-1.3852e-004	2.0019e-003	0	0	-1.0095e-004	0	0	0	0	0
0	1.3088e-019	0	0	-1.0770e-004	6.2015e-005	6.8348e-006	0	0	0
1.6363e-003	1.7286e-002	0	0	-4.0128e-004	2.8548e-006	3.1463e-007	0	0	0

>> d

d =

Columns 1 through 8

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
-2.9225e-004	-2.9225e-004	-8.5080e-004	-5.3140e-004	-5.3140e-004	-3.0345e-003	-3.0345e-003	-2.6006e-004	0	0
1.4185e-004	-1.4185e-004	0	2.9913e-004	-2.9913e-004	1.7636e-003	-1.7636e-003	3.8579e-004	0	0
1.8503e-002	1.8503e-002	9.3342e-003	1.0580e-002	1.0580e-002	-7.3805e-003	-7.3805e-003	-1.4511e-004	0	0

Columns 9 through 11

0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
-2.6006e-004	-2.6258e-006	0
-3.8579e-004	0	2.4113e-003
-1.4511e-004	2.4113e-003	0

Mach No. = 0.6, Altitude = 25,000 ft (steady turn, bank angle = 60 deg)

a =

```

8.8389e-003 -4.5946e-002 -2.0301e+000 -5.5886e-001 3.9864e-002 0 7.6618e-003 0
1.4825e-002 -8.5038e-001 1.0444e+001 -2.7508e-002 -8.9107e-003 -8.0981e-003 -8.4805e-005 -4.8347e-001
8.4736e-002 -7.5301e-001 -7.2466e-001 0 -1.3201e-001 3.6525e-002 -1.0136e-002 0
0 0 5.0158e-001 0 0 0 -8.6511e-001 -8.8912e-002
-4.4602e-002 -8.6740e-003 1.1329e-006 -4.7444e-002 1.1079e-002 2.0316e+000 -1.0443e+001 2.8031e-001
-2.1457e-003 6.4421e-003 -4.0192e-002 0 -1.4822e+000 -8.8882e-001 2.9510e-001 0
-6.9972e-004 2.5245e-003 3.8442e-003 0 -2.3373e-001 -2.3010e-002 2.2584e-002 0
0 0 8.4895e-002 8.9769e-002 0 1.0000e+000 4.9222e-002 -8.7771e-016

```

>> b

b =

Columns 1 through 10

```

3.9019e-002 3.9019e-002 3.7961e-002 -2.7021e-002 -2.7039e-002 -8.0999e-002 -8.1138e-002 -5.3058e-003 -5.3058e-003 9.3231e-004
-5.0400e-001 -5.0400e-001 -3.0129e-001 -2.0913e-001 -2.0889e-001 2.3251e-001 2.3314e-001 2.2552e-002 2.2552e-002 -1.9504e-001
-3.9298e+000 -3.9298e+000 -1.9451e+000 -1.0485e+000 -1.0468e+000 6.9107e-001 6.9652e-001 -9.9962e-002 -9.9962e-002 -2.7238e+000
0 0 0 0 0 0 0 0 0 0
4.5591e-003 -4.5591e-003 0 1.4190e-002 -1.4249e-002 5.6379e-002 -5.6504e-002 2.3896e-003 -2.3896e-003 -3.6715e-007
5.7041e+000 -5.7041e+000 0 1.5498e+000 -1.5444e+000 -3.0085e+000 3.0183e+000 -3.1068e-001 3.1068e-001 1.9768e-007
1.0613e-001 -1.0613e-001 0 -2.1517e-001 2.1538e-001 -3.9858e-001 3.9945e-001 -1.3590e-002 1.3590e-002 3.6355e-006
0 0 0 0 0 0 0 0 0 0

```

Column 11

```

7.6809e-005
0
2.3792e-005
0
1.9504e-001
-1.0501e-001
-1.9313e+000
0

```

>> c

c =

```

9.8163e-001 1.9081e-001 0 0 7.2015e-004 0 0 0
-1.7934e-002 9.2265e-002 0 0 0 0 0 0
0 0 1.0000e+000 0 0 0 0 0
0 0 0 1.0000e+000 0 0 0 0
-6.6444e-005 -1.2915e-005 0 0 9.3992e-002 0 0 0
-8.1498e-004 4.1927e-003 0 0 0 9.8163e-001 1.9081e-001 0
-9.9344e-007 5.1108e-006 0 0 0 0 -1.9081e-001 9.8163e-001 0
0 0 0 0 0 0 0 1.0000e+000
2.7472e-004 9.6267e-004 0 0 -1.4710e-004 0 0 0
-1.5245e-007 1.5897e-006 3.5211e-008 0 3.4434e-004 4.7996e-005 9.3295e-006 0
1.9299e-003 2.6431e-002 9.9481e-009 0 5.4814e-004 1.3560e-005 2.6358e-006 0

```

>> a

d =

Columns 1 through 10

0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
1.2128e-003	1.2128e-003	1.1799e-003	-8.3983e-004	-8.4040e-004	-2.5175e-003	-2.5218e-003	-1.6491e-004	-1.6491e-004	2.8977e-005	
1.4170e-004	-1.4170e-004	0	4.4105e-004	-4.4286e-004	1.7523e-003	-1.7562e-003	7.4271e-005	-7.4271e-005	-1.1411e-008	
1.5665e-002	1.5665e-002	9.3644e-003	6.5000e-003	6.4924e-003	-7.2266e-003	-7.2461e-003	-7.0093e-004	-7.0093e-004	6.0621e-003	

Column 11

0
0
0
0
0
0
0
0
0
2.3873e-006
6.0621e-003
0

Mach No. = 0.9 Altitude = 35,000 ft (wings level)

a =

-1.1400e-002	4.9835e-002	-1.1490e+000	-5.5995e-001	-4.3042e-003	0	0	0
-7.1202e-002	-6.1261e-001	1.5239e+001	-4.2221e-002	8.9952e-003	3.9819e-004	3.0024e-005	0
-2.6984e-001	-4.5007e-001	-6.0733e-001	0	2.4971e-002	-7.0499e-004	-5.3157e-005	0
0	0	1.0000e+000	0	0	0	0	0
1.2015e-017	0	0	0	-8.1296e-003	1.1506e+000	-1.5239e+001	5.5995e-001
0	0	0	0	-7.5942e-001	-7.5934e-001	1.3915e-001	0
0	0	0	0	-1.2068e-001	8.4167e-003	-1.4168e-002	0
0	0	0	0	0	1.0000e+000	7.5402e-002	0

>> b

b =

Columns 1 through 8

-1.7943e-002	-1.7943e-002	-6.1980e-002	-2.2331e-002	-2.2331e-002	-1.4240e-001	-1.4240e-001	-1.3153e-002
-8.3378e-001	-8.3378e-001	-2.9910e-001	-5.2179e-001	-5.2179e-001	3.6383e-001	3.6383e-001	1.8624e-002
-7.3127e+000	-7.3127e+000	-2.5975e+000	-2.4202e+000	-2.4202e+000	2.8081e+000	2.8081e+000	-1.0550e-001
0	0	0	0	0	0	0	0
1.8059e-002	-1.8059e-002	0	4.7930e-004	-4.7930e-004	6.1131e-002	-6.1131e-002	2.3480e-002
8.3878e+000	-8.3878e+000	0	4.3692e+000	-4.3693e+000	-4.1643e+000	4.1643e+000	-1.5024e-001
-1.5399e-001	1.5399e-001	0	-2.3691e-001	2.3695e-001	-7.9366e-001	7.9366e-001	-1.0674e-001
0	0	0	0	0	0	0	0

>> c

c =

9.9717e-001	7.5188e-002	0	0	0	0	0	0
-4.9201e-003	6.5251e-002	0	0	0	0	0	0
0	0	1.0000e+000	0	0	0	0	0
0	0	0	1.0000e+000	0	0	0	0
0	0	0	0	6.5436e-002	0	0	0
0	0	0	0	0	9.9717e-001	7.5188e-002	0
0	0	0	0	0	-7.5188e-002	9.9717e-001	0
0	0	0	0	0	0	0	1.0000e+000
-3.5433e-004	1.5489e-003	0	0	-1.3378e-004	0	0	0
3.7342e-019	0	0	0	-2.5268e-004	4.9947e-005	3.7661e-006	0
2.2130e-003	1.9041e-002	0	0	-2.7958e-004	-1.2376e-005	-9.3318e-007	0

d =

Columns 1 through 8

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
-5.5769e-004	-5.5769e-004	-1.9264e-003	-6.9408e-004	-6.9408e-004	-4.4259e-003	-4.4259e-003	-4.0881e-004		
5.6129e-004	-5.6129e-004	0	1.4897e-005	-1.4897e-005	1.9000e-003	-1.9000e-003	7.2978e-004		
2.5915e-002	2.5915e-002	9.2962e-003	1.6218e-002	1.6218e-002	-1.1308e-002	-1.1308e-002	-5.7885e-004		

Columns 9 through 11

0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
-4.0881e-004	-2.9148e-006	0
-7.2978e-004	0	3.0557e-003
-5.7885e-004	3.0557e-003	0

References

- [1] Buffington, J.M., Modular Control Law Design for the Innovative Control Effectors (ICE) Tailless Fighter Aircraft Configuration 101-3, Final Report 5/1/97-5/1/99, AFRL-VA-WP-TR-1999-3057, June 1999.
- [2] Ostroff, A.J., and B.J. Bacon, Force and Moment Approach for Achievable Dynamics using Nonlinear Dynamic Inversion, AIAA 99-4001, Proceedings 1999 AIAA Guidance, Navigation, and Control Conference, Aug. 1999.
- [3] Roskam, J., *Airplane Flight Dynamics and Automatic Flight Controls, Part I*. Published by Roskam Aviation and Engineering Corporation, Ottawa, KS, 1979.
- [4] McRuer, D., Ashkenas, I., and Graham, D., *Aircraft Dynamics and Automatic Control*. Princeton University Press, Princeton, New Jersey, 1973.

II. A Design Challenge Solution

A. Introduction

One “solution” to the design challenge described in Part I of this document will now be presented. The controller design is similar to that presented in Ref. 1, albeit somewhat simpler in implementation. The design approach involves sliding mode control (SMC) utilizing asymptotic observers and reference model “hedging” to reduce the SMC system’s sensitivity to parasitic dynamics, here consisting of the actuator dynamics that are ignored in the design procedure.

B. Solution

A. Controller Description

Figure II.1 shows the basic control system structure. Figure II.2 is a detailed

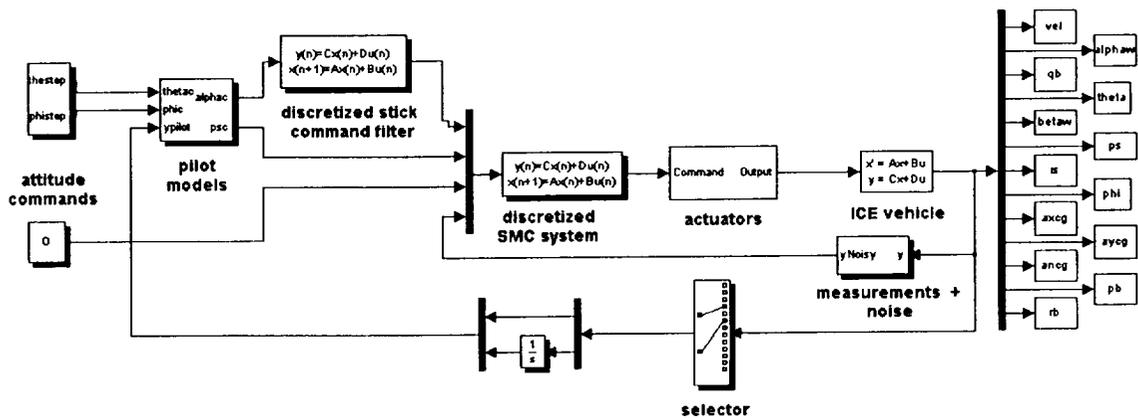


Figure II.1 The SMC system structure

representation of the continuous form of the “discretized SMC system” identified in Fig. II.1. Figure II.3 is a more detailed representation of the “sliding mode controllers” shown in Fig. II.2. Likewise, Figs. II.4, II.5 and II.6 are detailed representations of the “hedge system”, “reference models” and “asymptotic observers” of Fig. II.2. It should be noted that all of the possible sensed quantities available with the ICE model were input to the

observers. The control allocation matrix was created using a pseudo-inverse design technique [1].

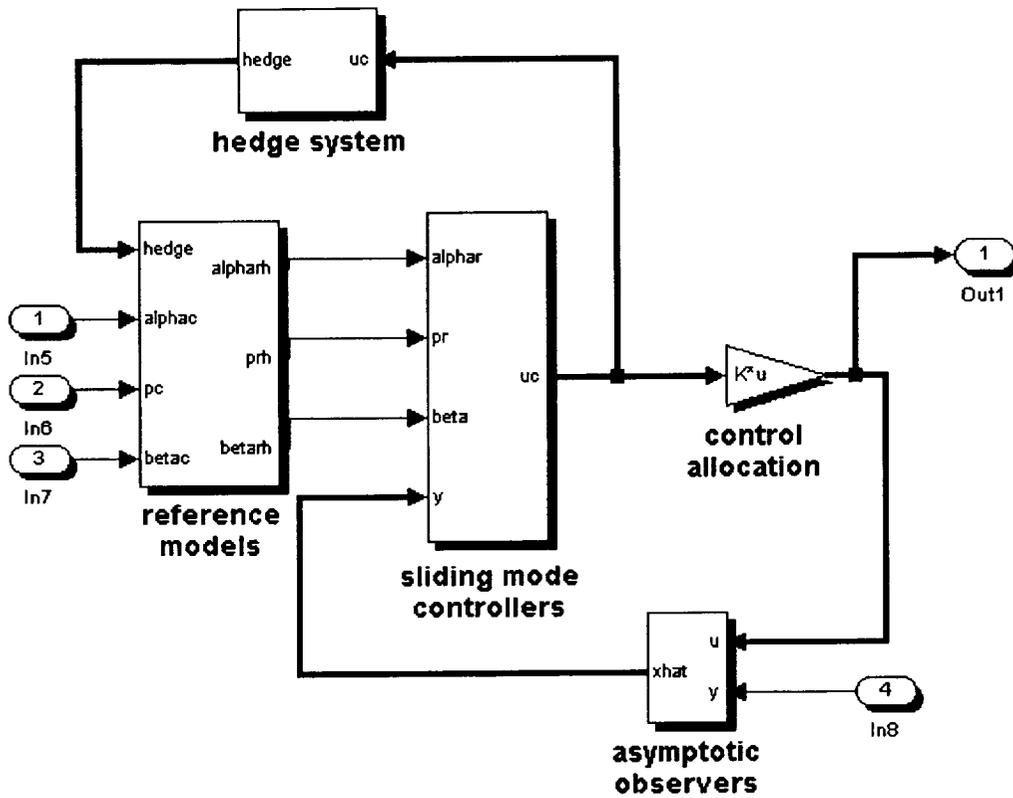


Figure II.2 Detail of continuous form of “discretized SMC system” in Fig. II.1

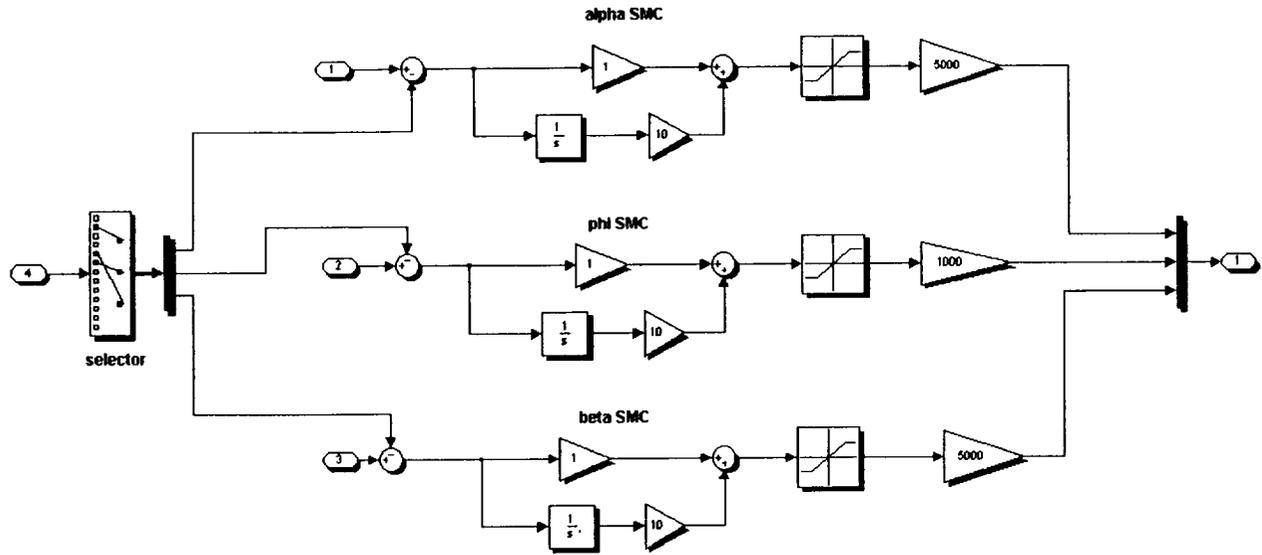


Figure II.3 Detail of “sliding model controllers” in Fig. II.2

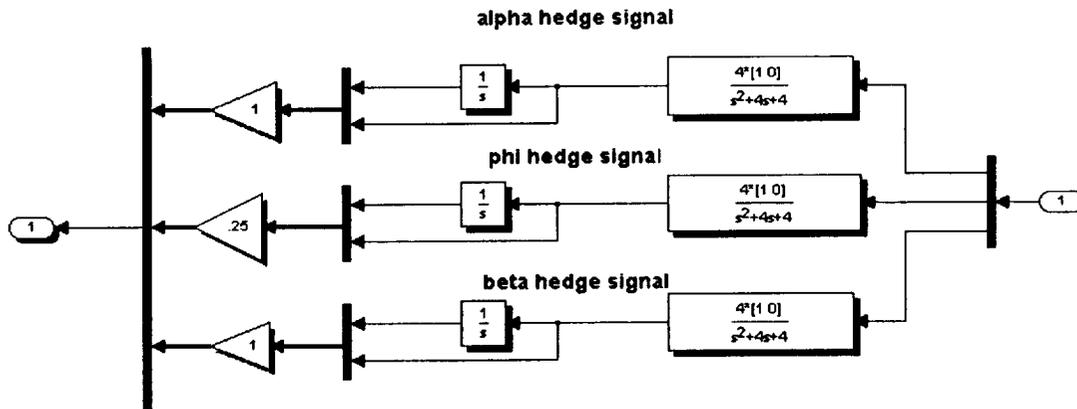


Figure II.4 Detail of “hedge system” of Fig. II.2

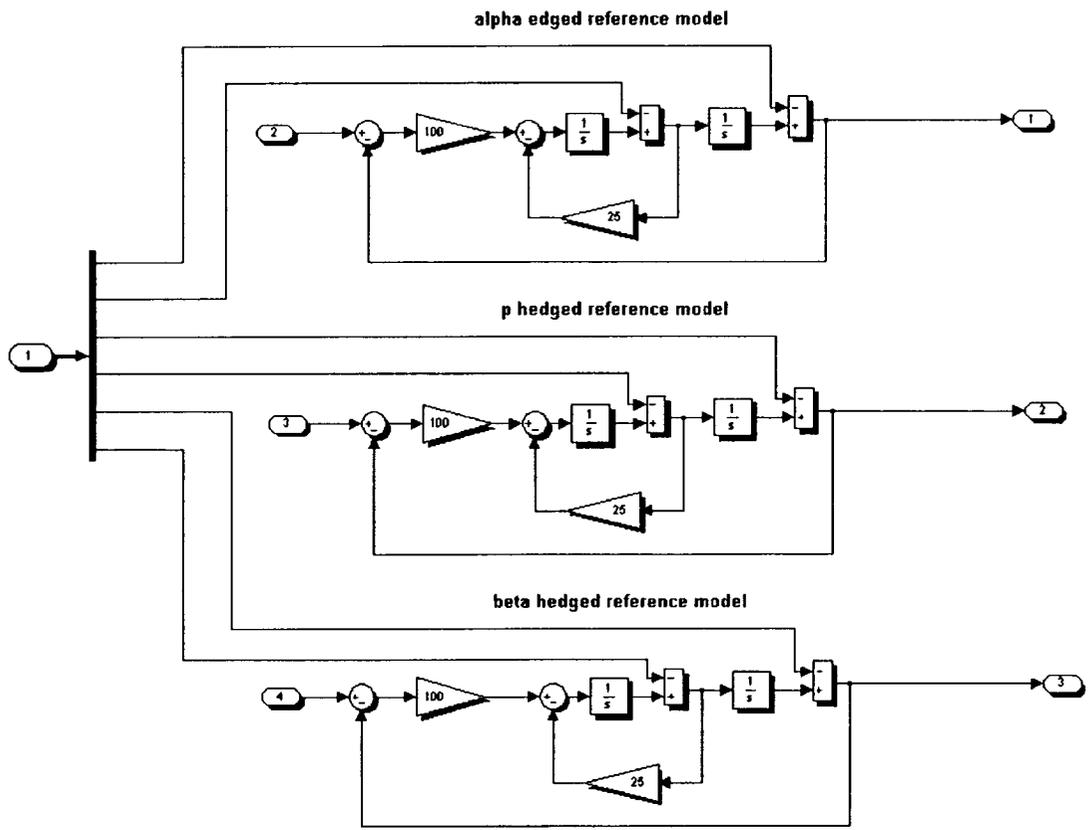


Figure II.5 Detail of “reference models” of Fig. II.2

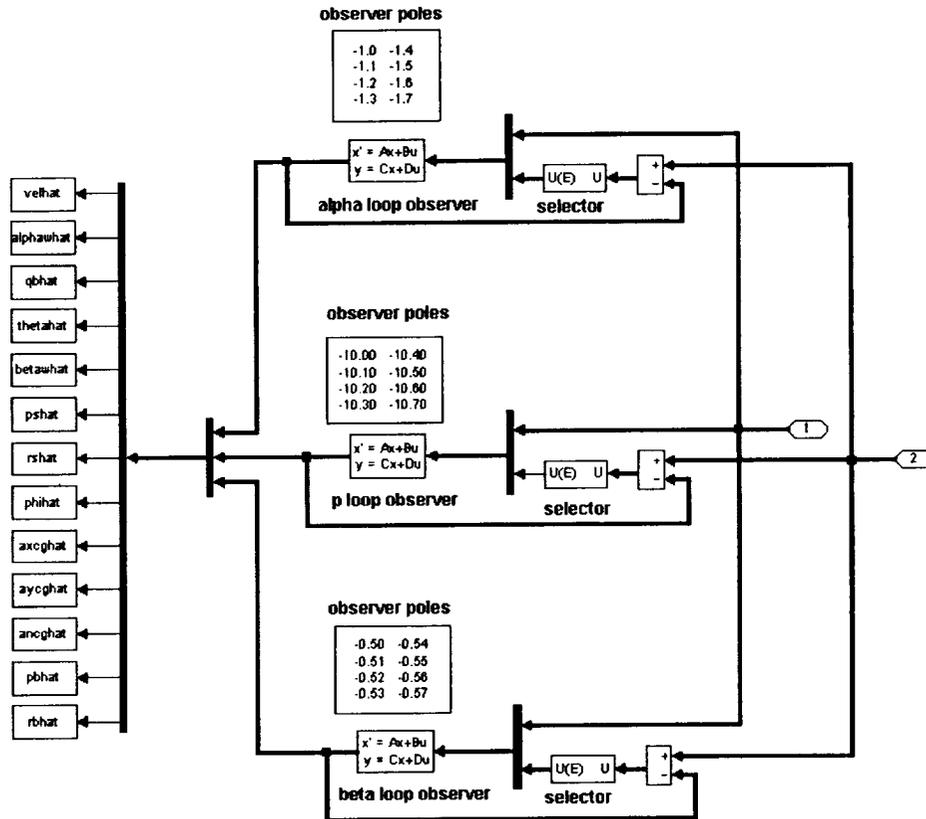


Figure II.6 Detail of “asymptotic observers” in Fig. II.2

Details of the design procedure will be omitted here. The reader is referred to Ref. 1 for a thorough discussion of the approach. It should be mentioned here, however, that the SMC design to be evaluated here is not an “optimum” one. For example, the hedging logic was created for a single flight condition (Mach No. = 0.6, Altitude = 25,000 ft) and was left invariant for the remainder of the flight conditions evaluated.

It should be noted that the pitch-rate command system was obtained by first designing an alpha-command system. Then a control stick command filter was placed in series with the pilot’s longitudinal stick command. The filter was a simple lag/lead network that produced rate-like pitch attitude responses when input to the alpha-command system. Since an alpha-command system was actually in evidence, no phugoid damping was provided by the control system. However, this produced no problems as the pilot control of pitch attitude provided excellent phugoid damping.

2.0 Solution Results

Handling Qualities

The bandwidth/phase delay values for each of the four flight conditions are given below.

Mach No. = 0.3, Altitude = 15,000 ft (wings level)

	Pitch	Roll
$\omega_{BW_{phase}}$ (rad/sec)	3.69	2.35
τ_p (sec)	0.071	0.11

Mach No. = 0.6, Altitude = 25,000 ft (wings level)

	Pitch	Roll
$\omega_{BW_{phase}}$ (rad/sec)	3.69	2.35
τ_p (sec)	0.071	0.11

Mach No. = 0.6, Altitude = 25,000 ft (steady turn)

	Pitch	Roll
$\omega_{BW_{phase}}$ (rad/sec)	4.42	2.32
τ_p (sec)	0.065	0.12

Mach No. = 0.9, Altitude = 35,000 ft

	Pitch	Roll
$\omega_{BW_{phase}}$ (rad/sec)	5.14	2.32
τ_p (sec)	0.063	0.12

Referring to Fig. I.8 in Part I of this document, these values are each seen to predict Level 1 handling qualities.

b. Tracking Performance

Tracking performance was examined for flight conditions for nominal and failed cases. In the case of failures, the failure was introduced 20 seconds into a 50 second run. The entire failure ensemble described in Section I.C.8 was accommodated by the controller. Performance was in the desired categories as described in Section I.C.7.b except for the following:

Mach No. = 0.3, Altitude = 15,000 ft, wings-level, nominal
only adequate performance for beta

Mach No. = 0.6, Altitude = 25,000 ft; steady turn ,nominal
only adequate performance for theta

Mach No. = 0.6, Altitude = 25,000 ft; steady turn, failed
only adequate performance for theta

Mach No. = 0.9, Altitude = 35,000 ft, failed
adequate performance not attained for phi

The last of these conditions was the most serious in terms of performance, where adequate performance was not in evidence for the phi loop. Figures II.7 through II.18 demonstrate the performance on a selected set of flight conditions and vehicle modes. Figure II.18 demonstrates the significant rate limiting occurring in control effector dE3 (left elevon) after the failure.

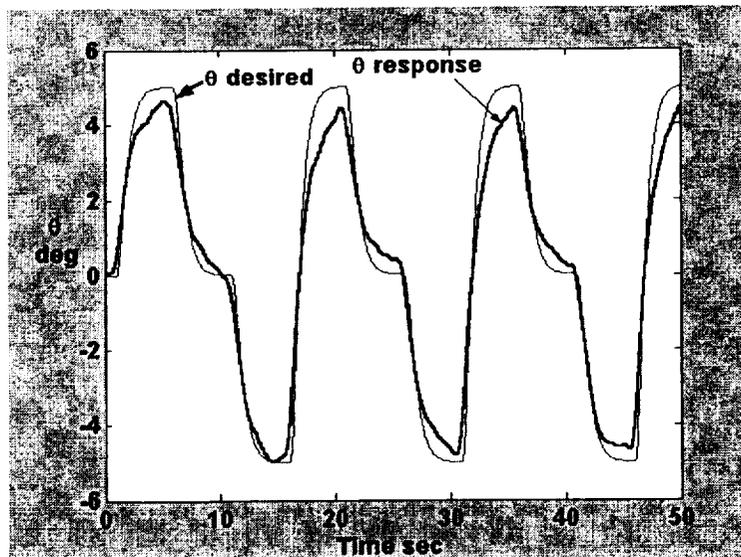


Figure II.7 θ tracking Mach No. = 0.3 Altitude = 15,000 ft, wings level, nominal

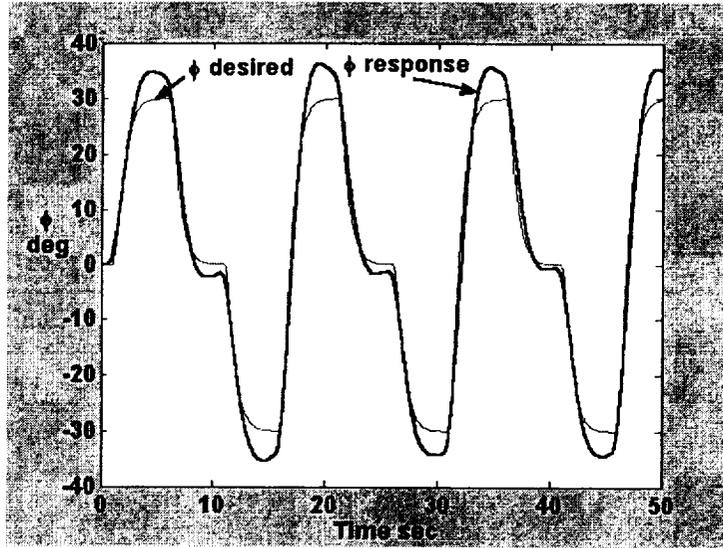


Figure II.8 ϕ tracking Mach No. = 0.3 Altitude = 15,000 ft, wings level, nominal

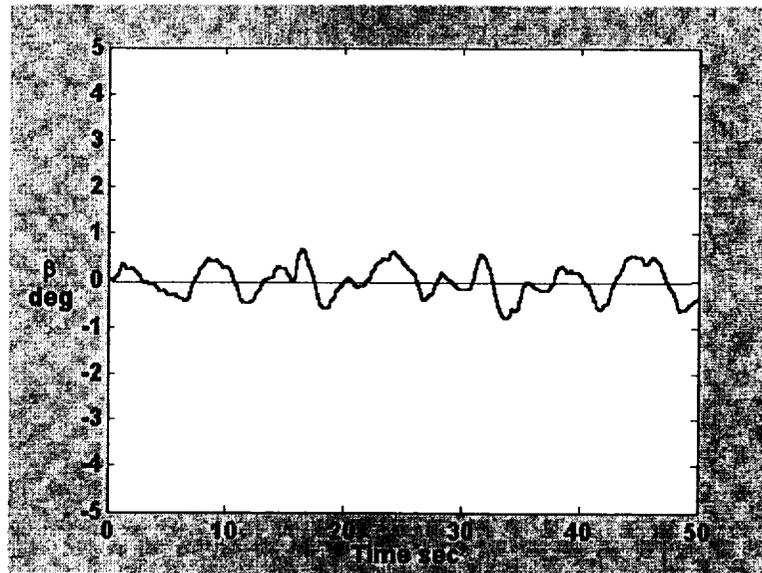


Figure II.9 β tracking Mach No. = 0.3 Altitude = 15,000 ft, wings level, nominal

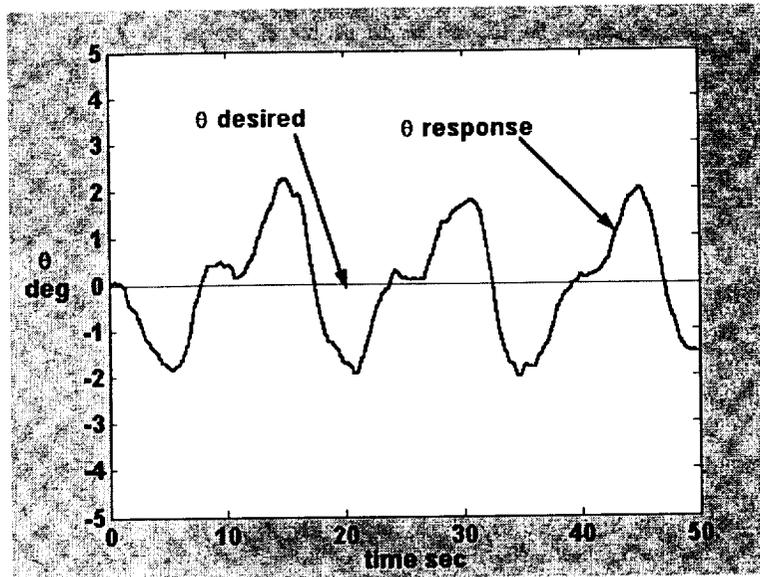


Figure II.10 θ tracking Mach No. = 0.6 Altitude = 25,000 ft, steady turn, nominal

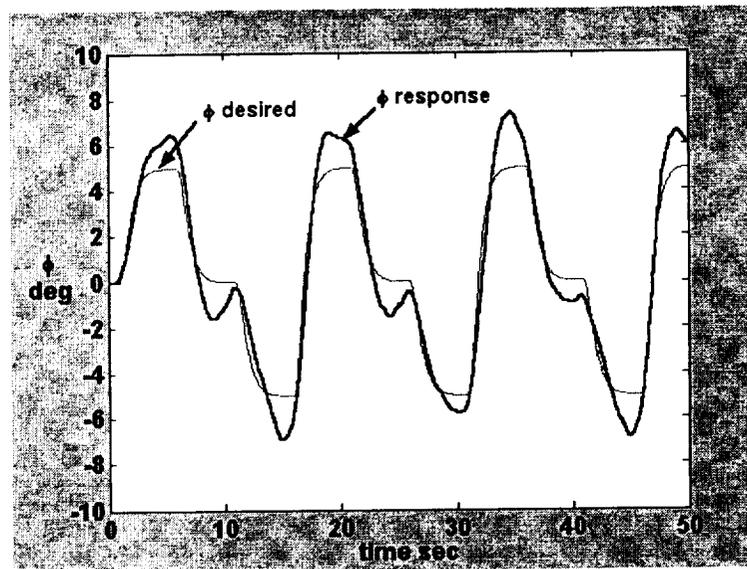


Figure II.11 ϕ tracking Mach No. = 0.6 Altitude = 25,000 ft, steady turn, nominal

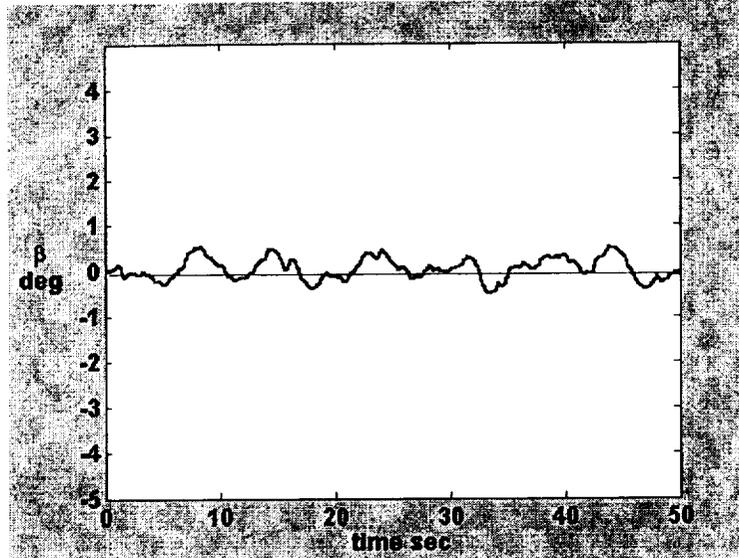


Figure II.12 β tracking Mach No. = 0.6 Altitude = 25,000 ft, steady turn, nominal

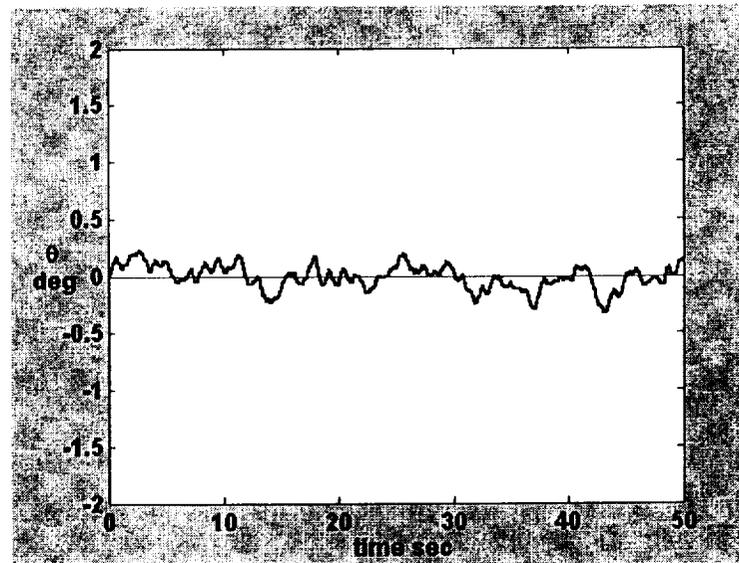


Figure II.13 θ tracking Mach No. = 0.9 Altitude = 35,000 ft, wings level, nominal

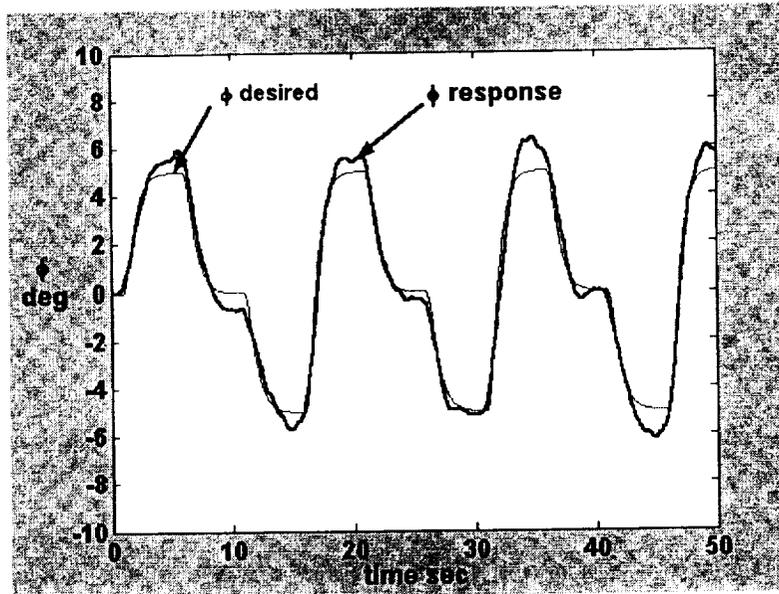


Figure II.14 ϕ tracking Mach No. = 0.9 Altitude = 35,000 ft, wings level, nominal

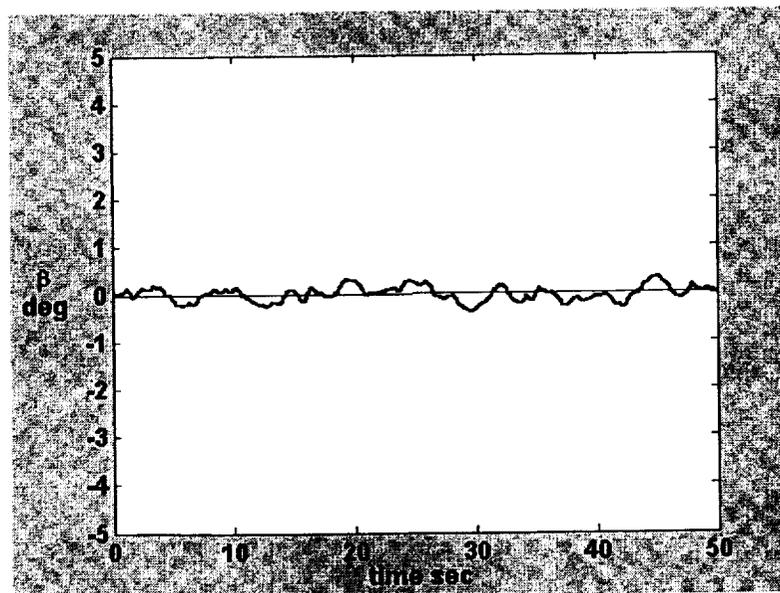


Figure II.15 β tracking Mach No. = 0.9 Altitude = 35,000 ft, wings level, nominal

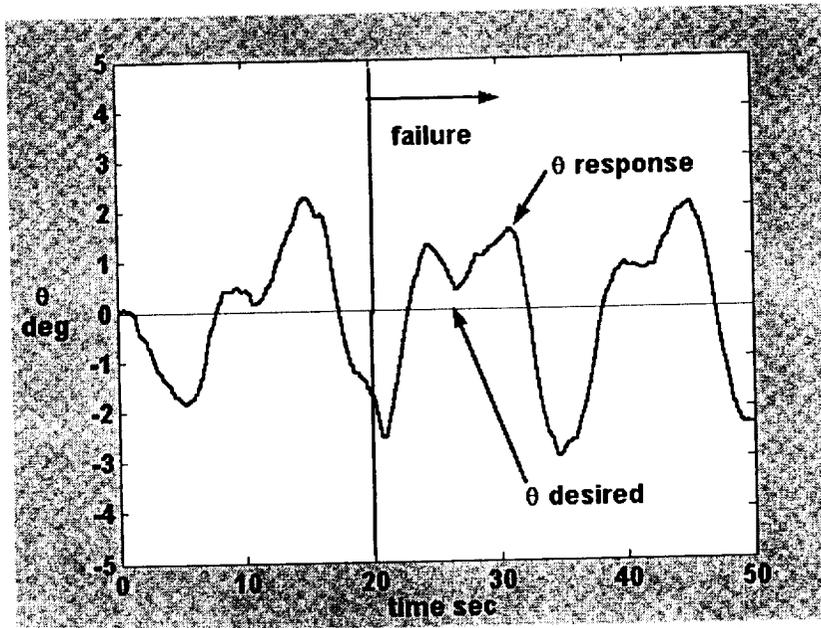


Figure II.16 θ tracking Mach No. = 0.6 Altitude = 25,000 ft, steady turn, failure

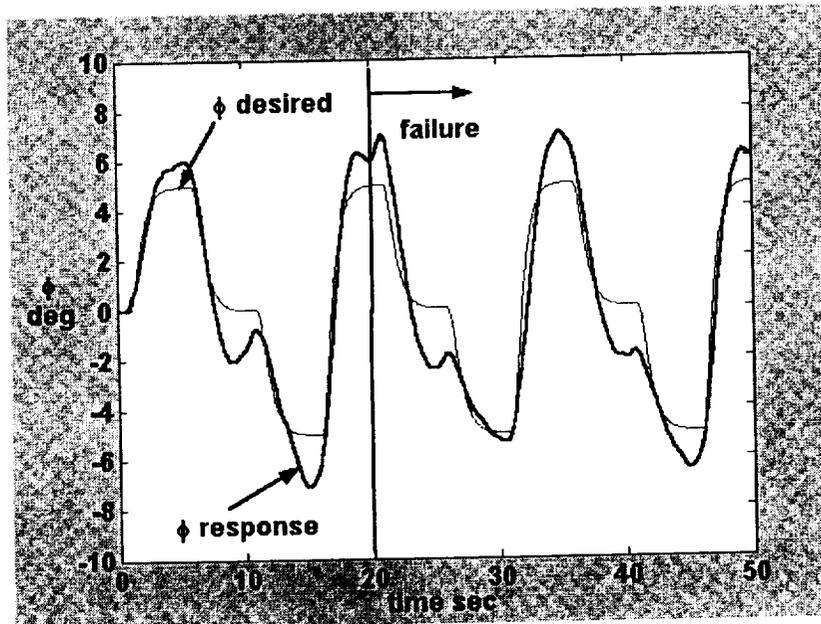


Figure II.17 ϕ tracking Mach No. = 0.6 Altitude = 25,000 ft, steady turn, failure

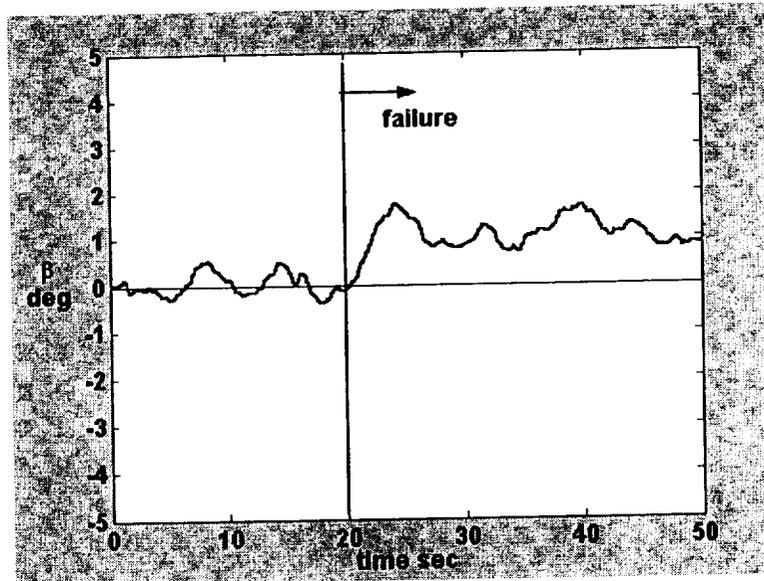


Figure II.17 β tracking Mach No. = 0.6 Altitude = 25,000 ft, steady turn, failure

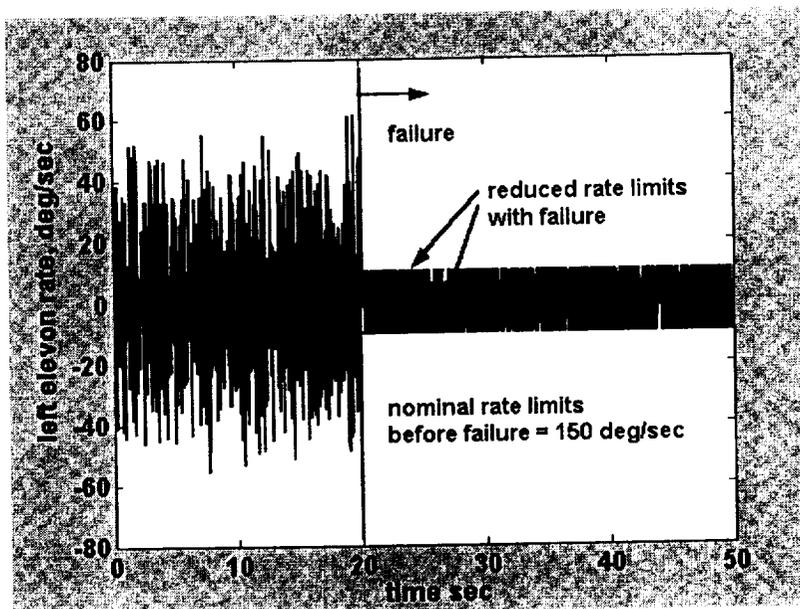


Figure II.18 left elevon rate, Mach No. = 0.6 Altitude = 25,000 ft, steady turn, failure

c. Stability Margins

The stability margin criteria of Section I.C.7.c were met in all but one case. This involved the yaw thrust loop in the flight condition Mach No. = 0.6, Altitude = 25,000 ft, steady-turn. A gain margin of only 3 dB resulted. The corresponding phase margin of 34 degrees met the criterion.

d. Structural Coupling Considerations

The structural coupling criteria of Section I.C.7.d were violated in four cases. All three involved the magnitude of the transfer function between pilot pitch input and q_b . The violations were

Mach No. = 0.3, Altitude = 15,000 ft, wings-level; -9.27 dB rather than -10 dB

Mach No. = 0.6, Altitude = 25,000 ft, wings-level; -8.21 dB rather than -10 dB

Mach No. = 0.9, Altitude = 35,000 ft, wings-level; -7.07 dB rather than -10 dB

Mach No. = 0.6, Altitude = 25,000 ft, steady-turn; -8.7 dB rather than -10 dB

Figure II.18 shows one of the transfer functions above, with the magnitude violation indicated. These violations were not considered serious, as they essentially could be interpreted as a slight increase in the 25 rad/sec criterion frequency

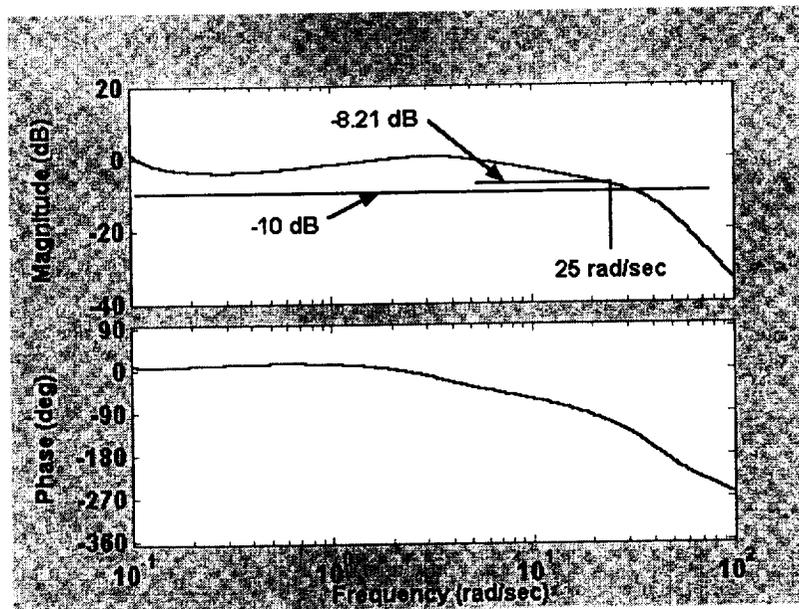


Figure II.18 Transfer function between pilot pitch input and q_b showing violation of structural coupling magnitude for $\omega \geq 25$ rad/sec

5. Control Activity

Five violations of the control activity criterion of Section I.C.7.e occurred. They are as follows:

Mach No. = 0.9, Altitude = 35,00 ft, level flight; commands to actuators dE2, dE12, dE10 and dE20 exceeded the 75% figure.

Mach No. = 0.6, Altitude = 25,000 ft, steady turn, commands to actuator dE12 exceeded the 75% figure.

It should be noted that these five violations were out of a total of $11 \times 4 = 44$ total possibilities (11 actuator commands \times 4 flight conditions). The violations were each on the order of 10% above the 75% criterion value).

6. Off-Nominal Flight Conditions

The off-nominal flight conditions were examined with the controller tuned for Mach No. = 0.6, Altitude = 25,000 ft, but with vehicle dynamics associated with Mach No. = 0.3, Altitude = 15,000 ft and Mach No. = 0.9, Altitude = 35,000 ft. Figures II.19 through II.24 show the results. One tracking performance violation occurred for beta tracking for the with Mach No. = 0.3, Altitude = 15,000 ft case where only adequate performance could be obtained.

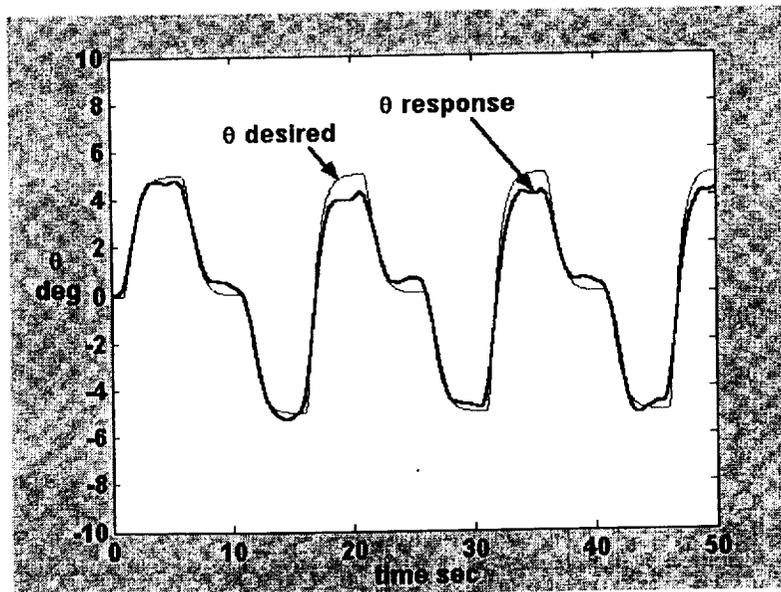


Figure II.19 θ tracking for off-nominal condition; controller tuned for Mach No. = 0.6, Altitude = 25,000 ft, but vehicle dynamics are Mach No. = 0.3, altitude = 15,000 ft

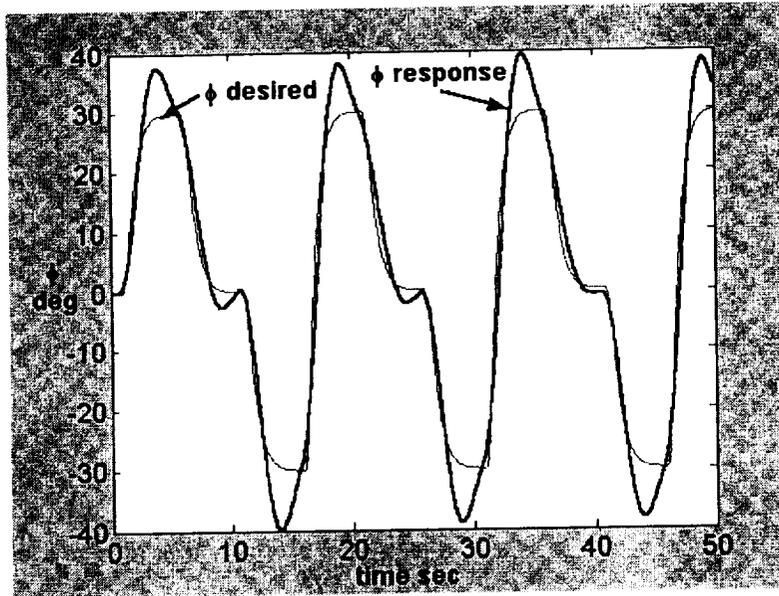


Figure II.20 ϕ tracking for off-nominal condition; controller tuned for Mach No. = 0.6, Altitude = 25,000 ft, but vehicle dynamics are Mach No. = 0.3, altitude = 15,000 ft

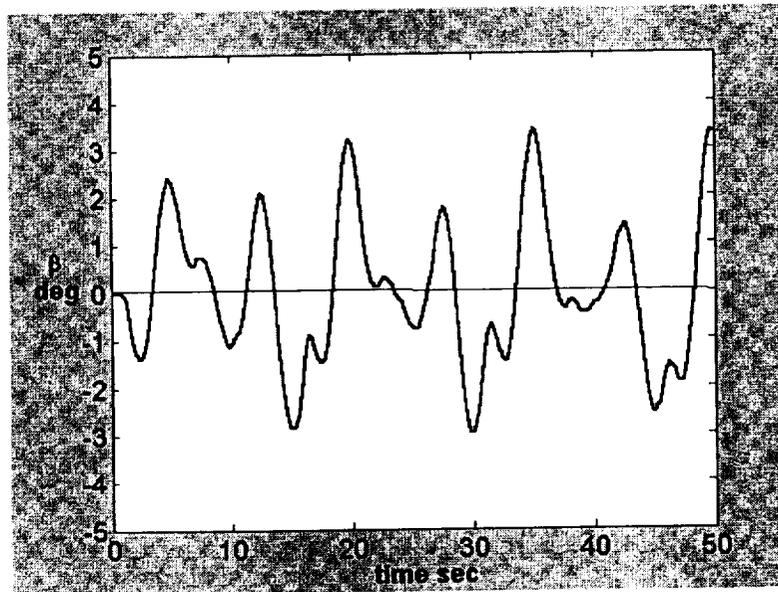


Figure II.21 β tracking for off-nominal condition; controller tuned for Mach No. = 0.6, Altitude = 25,000 ft, but vehicle dynamics are Mach No. = 0.3, altitude = 15,000 ft

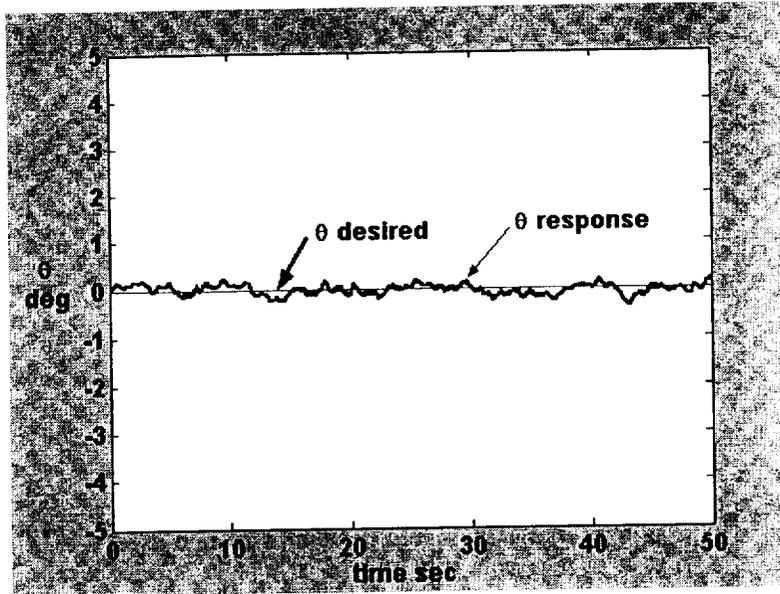


Figure II.22 θ tracking for off-nominal condition; controller tuned for Mach No. = 0.6, Altitude = 25,000 ft, but vehicle dynamics are Mach No. = 0.9, altitude = 35,000 ft

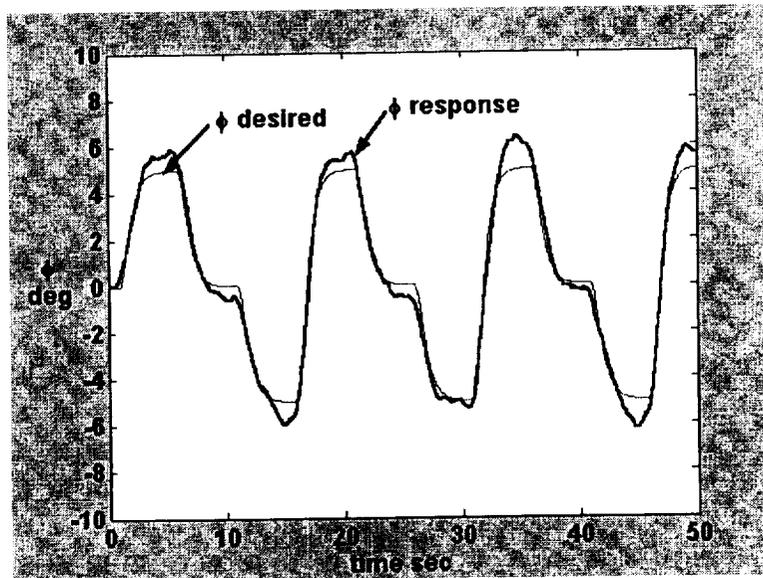


Figure II.23 ϕ tracking for off-nominal condition; controller tuned for Mach No. = 0.6, Altitude = 25,000 ft, but vehicle dynamics are Mach No. = 0.9, altitude = 35,000 ft

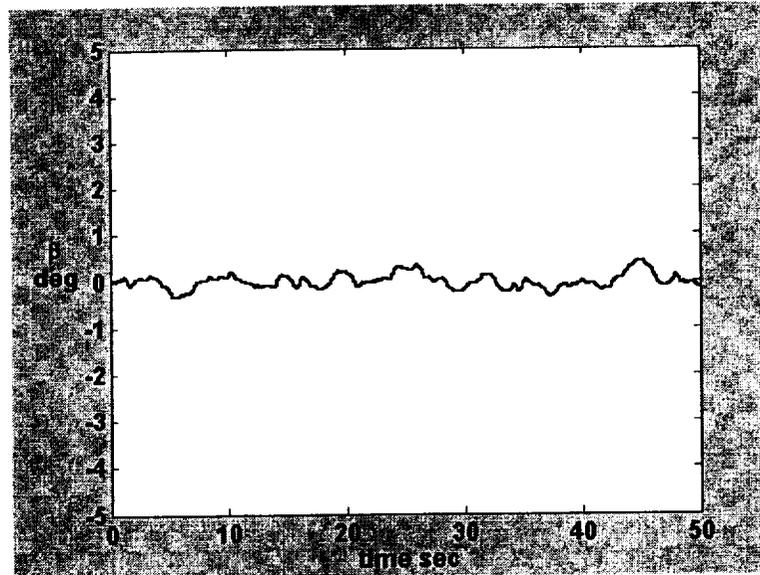


Figure II.24 β tracking for off-nominal condition; controller tuned for Mach No. = 0.6, Altitude = 25,000 ft, but vehicle dynamics are Mach No. = 0.9, altitude = 35,000 ft

Scheduling the controller designed herein would be based upon Mach No. and Altitude. In the implementation used, the only the asymptotic observers and the control distribution matrix were a function of flight condition. The SMC and hedging systems were independent of flight condition. It should be noted that is likely that an improved designs could be obtained by consideration of tuning the hedging system to the flight condition.

3. Discussion

There was some concern about the amount of control activity in the design just presented. Therefore, for comparison, a simple controller obtained with classical loop-shaping techniques was designed and evaluated in the same tracking task for flight condition Mach No. = 0.3, Altitude = 15,000 ft. This latter design had no asymptotic observers, and only three variables were assumed to be measured: α , ψ , and β . Figures II.25 and II.26 compare the control activity in the left elevon (dE03). As can be seen, the SMC design has slightly higher frequency content, but significantly less motion amplitude. Thus, the SMC design was considered to be acceptable from the standpoint of sensor noise transmission.

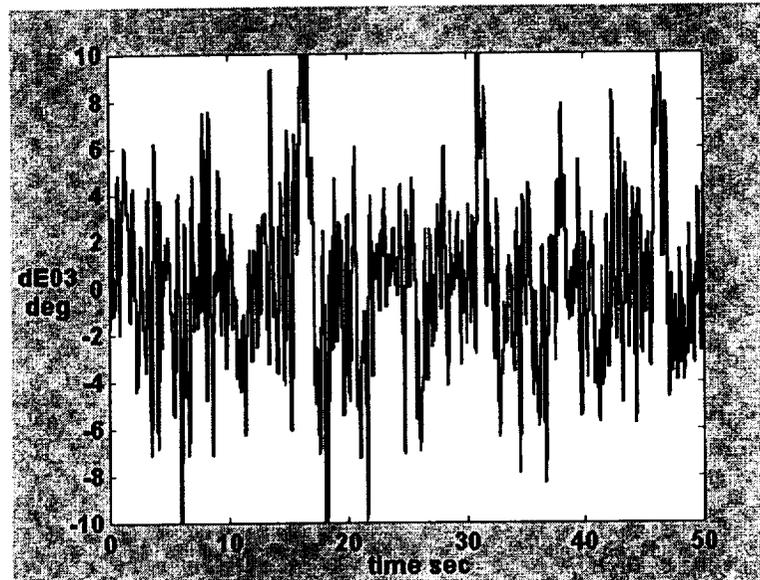


Figure II.25 Left elevon activity (dE03) for Mach No. = 0.3, Altitude = 15,000 ft, for loop-shape design

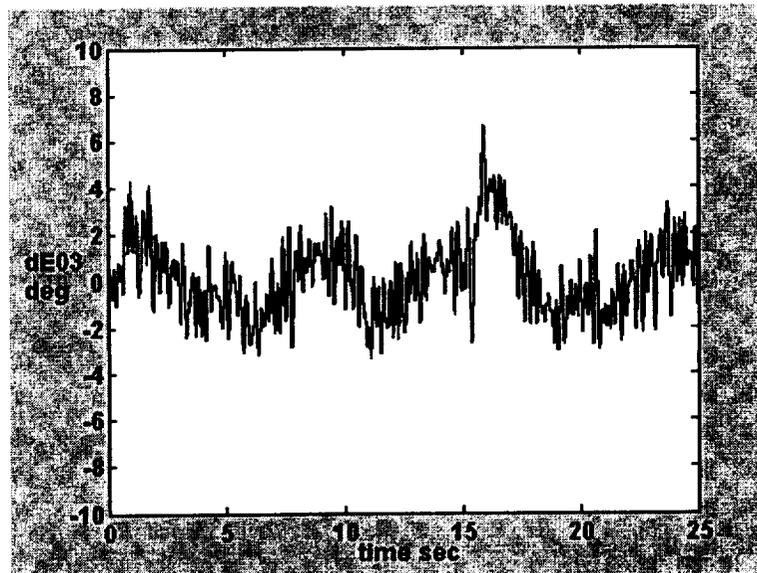


Figure II.26 Left elevon activity (dE03) for Mach No. = 0.3, Altitude = 15,000 ft, for SMC design

C. Conclusions

The controller design that was described and exercised in Part II was intended to serve as a test case for the design challenge. As pointed out in this section, a number of violations of the design specifications occurred, although none were “show-stoppers” in the sense that system stability was compromised. The SMC approach allowed the

pilot/vehicle system to accommodate the entire suite of failures described in Section I.C.8, which is notable.

E. References

- [1] Wells, S. R. and Hess, R. A., "MIMO Sliding Mode Control for a Tailless Fighter Aircraft – An Alternative to Reconfigurable Architectures," AIAA Paper No. 2002-4650.