Flight Simulation of WVU YF-22 Aircraft

Submitted as partial completion of the requirements
for graduation as a University Honors Scholar

Jason Gross
Nick Hansford
Kerri Phillips
Blake Waldie

Dr. Mario Perhinschi

Department of Mechanical and Aerospace Engineering
Morgantown, West Virginia

Spring 2008

Abstract

A flight simulator for the WVU YF-22 research aircraft was created using the Flight Dynamics and Controls Toolbox in Matlab and Simulink. This simulator allows for the user to control the aircraft motion and throttle with a joystick interface. A decoupled control surface non-linear model of the aircraft was developed in order to test various flight failure scenarios on individual control surfaces. Before modeling the failure scenarios, an accurate engine model, wind model, and a virtual reality world were developed to enhance the accuracy and usefulness of the simulation. Four failures were modeled in the simulator, and a graphical user interface was developed to provide the user with the option of selecting which failures to test for a given simulation. The four failures that were modeled include locked control surface, missing or damaged control surface, sensor failure, and icing. Each failure model was implemented into the simulation environment, and the user had specific options and conditionals for each failure mode. The locked control surface and missing or damaged control surface failure modes provided six individually controlled surfaces which were successfully implemented into individual failures to achieve desirable results during simulation testing. The sensor failure and icing failure models also provided results that validated the accuracy of the failure models. The design and implementation of the simulation environment and the failure modes are detailed in the report, with successful simulation trials that document the behavior of the aircraft during nominal and failure flight conditions.

This simulator was initially developed as coursework for MAE 593L – Advanced Flight Simulation. The goals for the design team were to create first a PC based simulation environment, with additional goals being to reproduce failure scenarios of a stuck control surface, and missing or damaged control surface. A final goal of the design was to make the simulator compatible with the MOBUS 6 DOF flight simulator located on the Evansdale campus. Concurrently with these goals, an icing failure scenario was developed by Blake Waldie for thesis submission per MAE 496. The icing failure scenario is integrated with the simulator design and cannot be fully described without detailing the simulator development. Therefore this document is also submitted as Blake Waldie's thesis for MAE 496.

Table of Contents

List of Figures

List of Symbols

Symbol	Description	Unit
	English	
A	Aerodynamic coefficient	-
C	Aerodynamic coefficient	-
E	Aerodynamic efficiency parameter	-
	Greek	
α	Angle of attack	deg or rad
β	Angle of sideslip	deg or rad
δ	Deflection angle	deg or rad
ψ	Perturbed value of airplane heading angle	rad
φ	Perturbed value of airplane bank angle	rad
	Subscript	
0	Zero surface deflection	-
a	Aileron	-
e	Elevator (Stabilator)	-
F	Force	-
f	Flap	-
L	Left surface	-
l	Rolling moment	-
M	Moment	-

m	Pitching moment	-
n	Yawing moment	-
p	Roll rate	-
q	Pitch rate	-
R	Right surface	-
r	Yaw rate	-
r	Rudder	-
u	Aerodynamic efficiency surface parameter	-
X	X direction	-
Y	Y direction	-
Z	Z direction	-

Introduction

The goal of the project was to develop a flight simulation of the WVU YF-22 research aircraft that allowed for continuous user input. The simulation was to consider in-flight conditions and to exclude take-off and landing. A non-linear mathematical model of the aircraft was developed by utilizing the known aerodynamic coefficients and incorporating them into the equations of motion. After producing the nominal aerodynamic model of the YF-22 and running it through the developed simulation, failure modeling was conducted. Several different failures were incorporated into the simulation scheme, and a graphical user interface (GUI) was made available for selection of conditions.

The failure conditions included a locked control surface, a missing or damaged control surface, a sensor failure, and icing conditions. The locked control surface failure was designed to occur on a left or right aileron, stabilator, or rudder at user-specified angles and failure times with the inclusion of transition time. The missing or damaged control surface caused an alteration in the efficiency of the control surface, and it was designed that the user could select the percentage of the surface that has been damaged. The included sensor faults affected the output of the angle of attack and the bank angle. In addition, icing conditions were studied and the detrimental lifting effects of ice accretion were modeled.

The simulation was developed to run with continuous user control on a personal computer (PC), with the final goal of implementing it on the Mobus 6 DOF Flight Simulator.

Literature Review

Flight Simulation

Flight simulators have been used for aerospace research and testing since as early as 1910[7]. These simulators have evolved over the years from the stationary Billing Flight Trainer, which was mounted on a universal joint facing into the wind as shown in Figure 1, to the most technologically advanced flight simulators such as the Boeing 787 flight simulator shown in Figures 2 and 3.

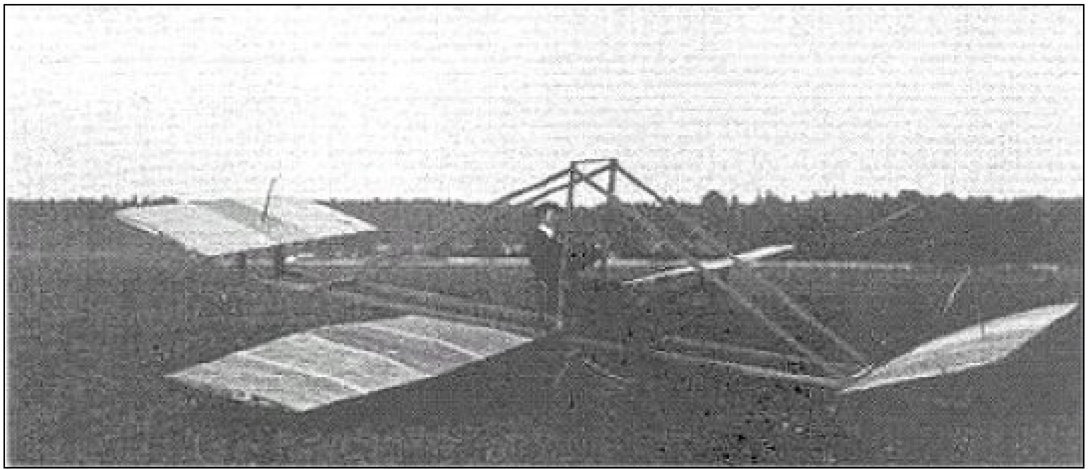


Figure 1: The Billing Flight Trainer (1910) [8]

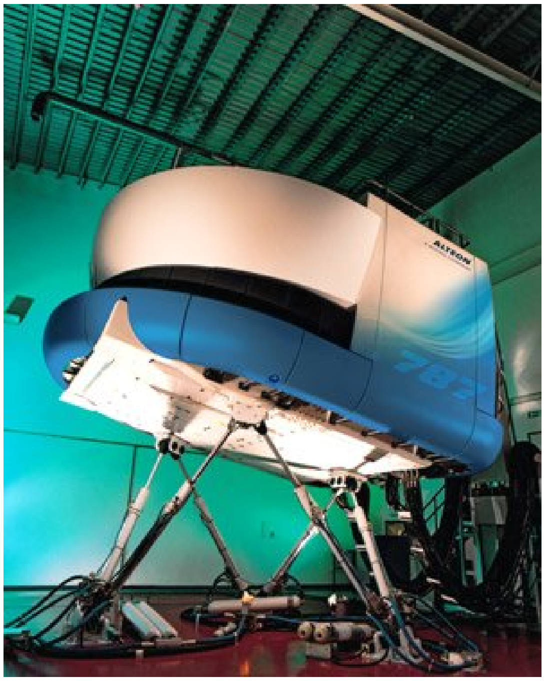


Figure 2: Alteon Boeing 787 Dreamliner Simulator[5]

Figure 3: Boeing 787 Dreamliner Simulator Cockpit[4]

Flight simulation is a cost effective, efficient, and safe means of testing and evaluating an aerospace system. The risks associated with the design, construction, and flight tests of the prototype of an aircraft may be greatly reduced with the use of flight simulation. In addition, pilot training may be conducted on simulators to familiarize and prepare them to fly the actual aircraft, thus reducing the initial risk. Flight simulation also allows an engineer to produce and evaluate aircraft designs quickly and at a reduced cost. Flight simulation is useful for preliminary aircraft design, performance evaluation, subsystem design and evaluation, system integration, and test design and support, amongst others[7]. The current simulators have six degrees of freedom and are based on a Stewart Platform, which provides this freedom of motion. Flight simulators are based on relaying visual, aural, and motion cues to the user to more accurately simulate a flight environment. Various flight scenarios may be modeled in simulators from steady, level flight to failure modes of an aircraft. A simulator incorporates the aircraft kinematic and dynamic equations of motion and the aircraft response to the atmospheric surroundings. As technology advances and more knowledge is gained about flight simulation, the models are improved to incorporate more realistic aircraft behavior. Both linear and non-linear aircraft models are used for flight simulation purposes depending on the application, and the performance of components such as the engine and the landing gear may be incorporated into the model. Software interfaces such as Matlab and Simulink are typically used to design and run these simulations on a personal computer or in the large six degree of freedom simulators[10].

Locked Control Surface Modeling

"A control 'surface blockage' implies that, after the failure occurrence time, the deflection of the control surface is no longer controllable since it either remains fixed at the value corresponding to the failure time or at a position reached shortly after the failure time (which is not commanded but as a result of the failure)[8]." For this failure modeling scenario, it is necessary to simulate the alteration of control signal inputs; however, it is not necessary to alter the aerodynamic forces and moments. This is due to the fact that using the aircraft model, simulating the locked surface will in itself alter the performance of the aerodynamic forces and moments.

Missing or Damaged Control Surface Modeling

One method of modeling missing portions of control surfaces is to "represent the alteration of aerodynamic forces and moments through an equivalent loss of aerodynamic efficiency[8]". To do this, "the contribution of each individual control surface to the total external forces and moments is isolated and expressed in terms of one single parameter this is easily accessible[8]. This efficiency parameter is modeled as a number between 0 and 1, where this efficiency parameter is denoted by E_{u_k}. If there are *m* number of control surfaces, each indicated by an index value from *k=1…m* on an aircraft, then the resulting effect on the overall aerodynamic forces and moments by a damaged control surface modeled by Equation 1[8].

$$A_{FM} = \left(A_{FM}\right)_{WB} + \sum_{k=1}^{m} (A_{FM})_{c_k} E_{u_k} \qquad \textbf{(1)}$$

Where $(A_{FM})_{WB}$ denotes the aerodynamic forces an moments over the wing body, or primary lifting surfaces, and $(A_{FM})_{c_k}$ represents the aerodynamic forces and moments created over the control surfaces.

Sensor Failure Modeling

An integral part to the proper execution of a dynamic system is being able to properly analyze and monitor its operation. The capability to detect faults and anomalies in the operation of the devices which record and control their behavior allows for prompt correction and a quick recovery to acceptable system performance. Sensors onboard an aircraft are one such example of a device, which control and monitor aircraft dynamic behavior by communicating with the human pilot and the flight computer.

The basic approach for detecting faults in aircraft sensor failures is having physical redundancy. In the event of a faulty or erroneous signal from one of the sensors, the flight computer will detect and remove the anomaly by comparing the flawed sensor to the output of the back up sensors. The fundamental concept of this approach, which is common with other schemes, is that a basis for comparison exists. A group of sensors performing a certain operation are compared at regular intervals. When one of these sensors produces an output that does not coincide with the others, it is concluded to be incorrect and removed from the control scheme or feedback operation[2].

More modern approaches to sensor failure detection vary in the way estimated and actual values for comparison are obtained. Analytical calculations are performed based on other aircraft states to provide an estimate for the desired value. Sensor values are then compared against these estimates, or 'observed' data taken from the mathematical model to identify and isolate any failures. Besides an observer, a neural network is another analytical scheme used in sensor failure detection and isolation. Neural nets are mathematical models that are used to estimate complex non-linear systems. Training algorithms are employed in order to improve the neural networks ability to approximate a specific function[3]. One of the main benefits to these analytical methods is the elimination of dependence upon physical redundancy. Removing the need for multiple back ups reduces the weight penalty and thus improves aircraft performance.

Icing

Ice accretion on aerodynamic surfaces has been a problem known to the aviation field as early as the 1940s, but not until recently have attempts been made to reproduce these negative effects in a computer based simulation environment. NASA's Aviation Safety Program in particular has spearheaded the development of an icing effects simulator, along with the Bihrle Research Center and Wichita State University. One of the main goals of this research is to provide a methodology of simulating icing conditions which other flight simulators can implement in their pilot training programs.

The modeling process assumes the presence of an un-iced model already in place. This model includes the basic aerodynamic and thrust moment coefficients for the longitudinal and lateral dynamics, such as the pitching moment C_m and rolling moment C_l. These total coefficients are described with their basic dependencies, along with the buildup containing the individual components of the total coefficient. Examples of the pitching and rolling moment coefficient modeling structure are shown with aerodynamic and thrust contributions[1].

Pitching Moment Coefficient

$$C_{M,Total} = f(\alpha,\beta,\delta_F)$$
$$+\Delta C_{M,\delta e}(\alpha,\delta_E,\delta_F)$$
$$+\Delta C_{M,Rot}(\alpha,\Omega b/2V\cdot\sin(\beta),|\beta|,\delta_F)$$
$$+\Delta C_{M,Q}(\alpha,q\bar{c}/2V,\delta_F)$$

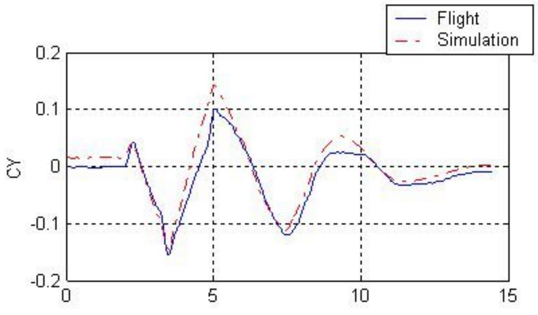
Thrust Increments

$$\Delta C_{M,CT}(\alpha,CT,\delta_F)$$
$$\Delta C_{M,A}(\alpha,CT,\delta_F)$$

Rolling Moment Coefficient

$$C_{l,Total} = f(\alpha,\beta,\delta_F)$$
$$+\Delta C_{l,\delta a}\cdot\sin(\delta_A),\ C_{l,\delta a} = f(\alpha,|\delta_A|,\delta_F)$$
$$+\Delta C_{l,\delta R}\cdot\sin(\delta_R),\ C_{l,\delta R} = f(\alpha,|\delta_R|)$$
$$+\Delta C_{l,Rot}\cdot\sin(\delta_\beta)$$
$$C_{l,Rot} = f(\alpha,\Omega b/2V\cdot\sin(\beta),|\beta|,\delta_F)$$
$$+\Delta C_{l,P}(\alpha,pb/2V,\delta_F)$$
$$+\Delta C_{l,R}(\alpha,rb/2V,\delta_F)$$

These aerodynamic data were collected during low speed wind tunnel testing and compared with data collected during test flights using a validation tool called 'Overdrive' from the D-Six modeling environment. Scaling of some terms were required in order to properly match the wind tunnel data with those of the flight tests. Figure 4 shows examples of comparisons between these results.



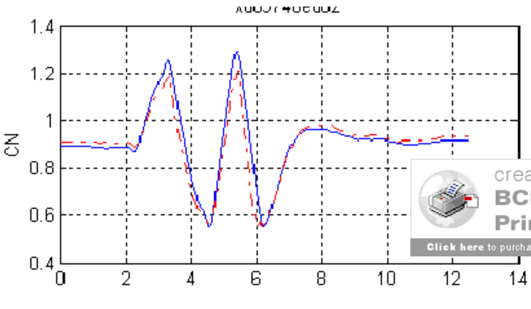


Figure 4: Overdrive Results for Lateral (left) and Longitudinal Moment Comparison (right) of the Flight and Simulated Data

The Overdrive results for the base mathematical model provided a calibration routine needed in order to correctly estimate the aerodynamic coefficients under the effect of icing conditions. Commonly occurring ice shapes were added to the wings of the aircraft scaled model to be tested in the wind tunnel. Conditions of the icing tests were identical to those during the tests resulting in the base model. Using the same data collection method as the base tests, the aerodynamic terms were collected again under the influence of icing conditions and scaled according to the results of the Overdrive comparison.

Cebeci (2003) listed several reasons as to why reproducing ice formation is a very difficult process:

- The capricious behavior of water on the airfoil surface. The formation of water rivulets and the changing paths that those rivulets follow are highly unpredictable. This directly affects the resulting ice shape.
- When ice starts accumulating, the resulting surface roughness varies significantly from one case to another and from one location on the surface to another. This is also very difficult to predict. Roughness plays a key role on the heat transfer between the water and the airflow, which drives the water freezing rate. The final ice shape is therefore very sensitive to the evolution of local surface roughness.
- Tezok & Fritz (1997) have shown from wind tunnel testing that ice density may experience important variations for different cases. In some instances, they measured values close to 300 kg/m^3, significantly lower than the typical values

used for ice shape determination (800 to 917 kg/m^3). The ice density is affected by the amount of air trapped in the ice.

- The physical model used in current ice accretion codes needs improving, especially if it is also to be used for three-dimensional flows. Studies conducted by Olsen & Walker (1986) confirm this need.

Model Design

Flight Dynamics and Controls Toolbox

The Flight Dynamics and Controls (FDC) Toolbox[9] was utilized as a starting point for the simulation design. FDC has been used by West Virginia University researchers to develop simulations for formation flight; therefore, it was selected as a viable resource. The team began with the basic "OLOOP1" layout and began to modify its contents to develop a simulation for failure modeling. Figure 5 provides an overview of the modified layout.

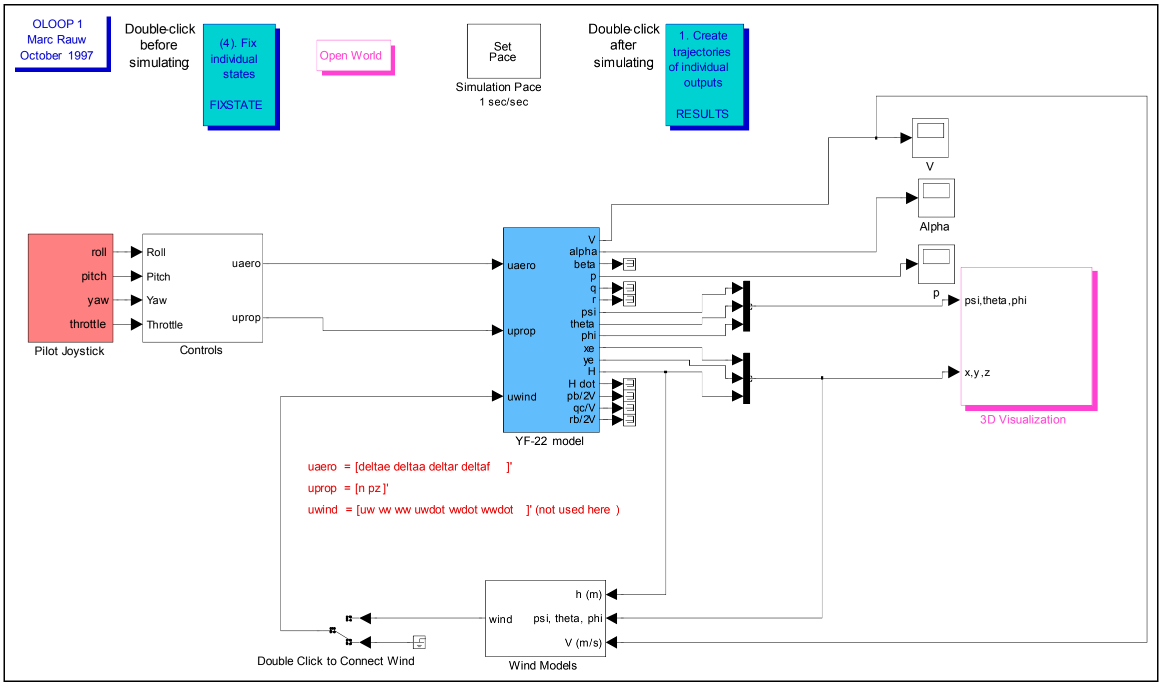


Figure 5: YF-22 Flight Simulator Main System Layout

Decoupled Control Surfaces

The YF-22 aircraft has three pairs of control surfaces: ailerons, rudders, and stabilators. Each of these surface pairs required separation in order to control each of the six individually. Individual control of each of the six surfaces allowed the locked and damaged control surface failure scenarios to be injected during simulations. The decoupling was modeled in the blue YF-22 model block, shown in Figure 5. Inside this block is the YF-22 dynamics and output equations subsystem, which contains the "Aerodynamics Group" subsystem. This subsystem accepts angle of attack, sideslip angle, and control surface deflection input from each of the three surface pairs as well as the throttle commands. The typical output of the coupled system was a vector of 19 values which included the following information:

$$[1,\ \alpha,\ \alpha^2,\ \alpha^3,\ \beta,\ \beta^2,\ \beta^3,\ \frac{pb}{2V},\ \frac{qc}{V},\ \frac{rb}{2V},\quad \delta_e,\ \delta_f,\ \delta_a,\ \delta_r,\ \alpha\delta_{f,}\ \alpha\delta_r,\ \alpha\delta_a,\ \delta_e\ \beta^2\ 0]'$$

The only alterations made to this vector was to diverge two signals from the control surface inputs and multiply them by a gain of 0.5, so to divide them into the six individual surfaces. This increased the length of the output vector to 22, as shown in Figure 6. The control surface deflections that were multiplied by the angle of attack or sideslip angle were not separated in this case because they were being multiplied by a matrix of zeros in the “Matrix Gain” block, so the signal being multiplied did not affect the model. The “Matrix Gain” block multiplies the vector output by the “AM Matrix,” which is a collection of the aerodynamic force and moment coefficients for the aircraft. The initial matrix was a 6x19; however, it was altered to a 6x22 to account for the decoupled surfaces. Also, when there are individual deflections between two control surfaces, such as the left and right stabilator, coupling of the longitudinal and lateral-directional dynamics occurs. The AM Matrix accounts for this coupling by adding yawing and rolling moment coefficients which are affected by individual deflections of the stabilators. This coupling is present due to the different lift and drag occurring on the surfaces, thus causing a roll and yaw of the aircraft when a single stabilator is deflected.

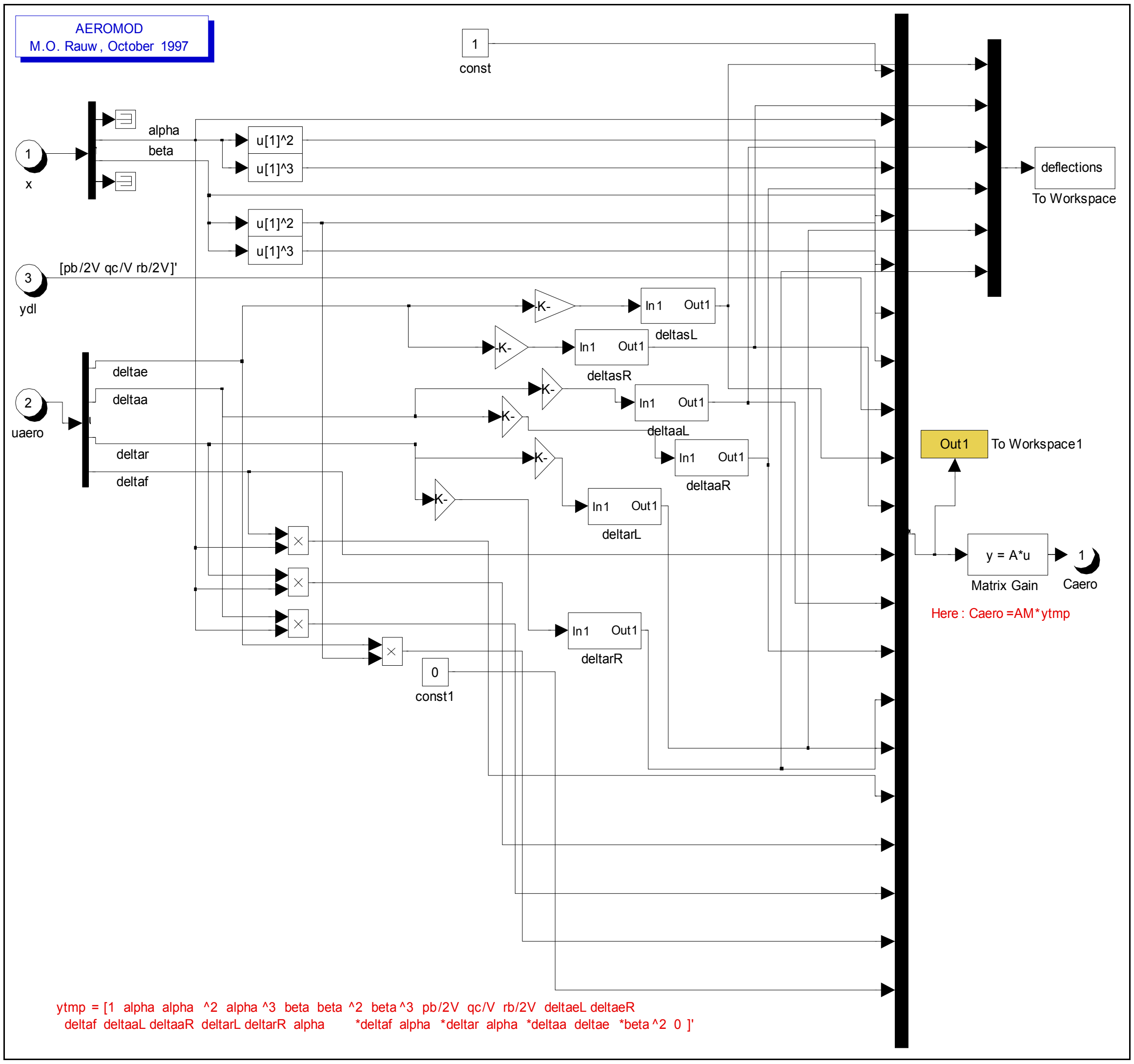


Figure 6: Decoupled Control Surface Model

The AM Matrix is included in the following section. This 6x22 matrix was inverted before being sent to the “Matrix Gain” block in the decoupled control surface model.

```
CX0     CY0     CZ0     Cl0     Cm0     Cn0  ;
CXa     0       CZa     0       Cma     0    ;
CXa2    0       0       0       Cma2    0    ;
CXa3    0       CZa3    0       0       0    ;
0       CYb     0       Clb     0       Cnb  ;
0       0       0       0       Cmb2    0    ;
0       0       0       0       0       Cnb3 ;
0       CYp     0       Clp     0       Cnp  ;
CXq     0       CZq     0       Cmq     Cnq  ;
0       CYr     0       Clr     Cmr     Cnr  ;
0       0       CZdeL   CldeL   CmdeL   CndeL;
0       0       CZdeR   CldeR   CmdeR   CndeR;
CXdf    0       CZdf    0       Cmdf    0    ;
0       CYdaL   0       CldaL   0       CndaL;
0       CYdaR   0       CldaR   0       CndaR;
CXdrL   CYdrL    0      CldrL   0       CndrL;
CXdrR   CYdrR    0      CldrR   0       CndrR;
CXadf   0       CZadf   0       0       0    ;
0       CYdra   0       0       0       0    ;
0       0       0       Cldaa   0       0    ;
0       0       CZdeb2  0       0       0    ;
0       CYbdot  0       0       0       0    ;
```

The decoupled model was verified by running a flight scenario with the coupled control surfaces and running the exact flight again with the decoupled control surfaces and ensuring that the output was the same.

Joystick Implementation

In order to design a simulation where the user could pilot the aircraft, an interface had to be established with a joystick. A joystick model block was taken from the Aerospace Blockset in Simulink and implemented into the model. This block, shown in Figure 7, outputs the roll, pitch, yaw, and throttle commands based on the joystick connected to the USB port on the computer. The roll and pitch channels were obtained from the side-to-side and back-and-forth movement of the joystick, respectively, while the yaw command was input from a slider on the base of the joystick.

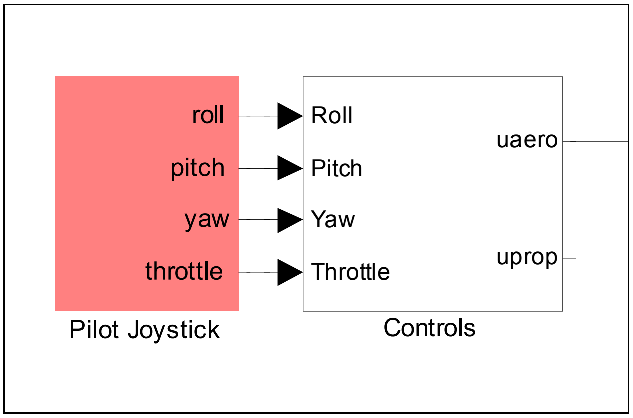


Figure 7: Joystick Interface Block with Pitch, Roll, Yaw, and Throttle Controls

The joystick input block provides the simulation with a raw value for the three axes or rotation ranging from -1 to 1, where 0 is recorded for no input and -1 and 1 represent the joysticks saturation limit in the negative and positive direction respectively. Thus, the input deflections had to be scaled according to maximum allowable deflections for the stabilator, aileron, and rudder. After consulting with Dr. Srik Gururajan, the maximum allowable deflections for the control surfaces were set as listed in Table 1.

Table 1: Deflections set for Joystick Saturation Limits

Control Surface	**Maximum Allowable Deflection**
Stabilator	±20°
Aileron	±30°
Rudder	±15°

The angular values were output in radians, so they could be multiplied by the aerodynamic coefficients in the AM Matrix. The throttle gain corresponded to the maximum thrust output of the jet engine on the aircraft, which was 97.86 N (22 lb). The "uaero" and "uprop" variables being fed into the system, as shown in Figure 8, are initial inputs for the controls and throttle command. These values were then sent into the YF-22 model system block.

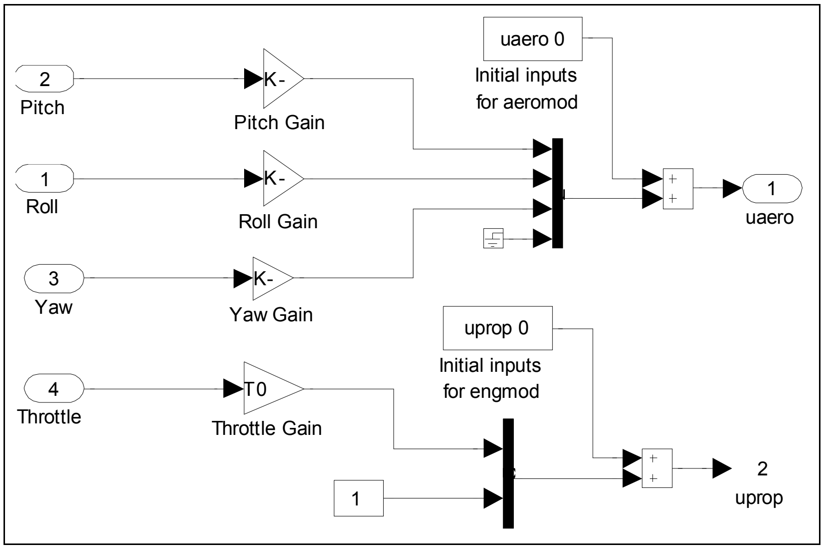


Figure 8: Controls Subsystem and Initial Inputs

Modal Analysis

A modal analysis of the WVU YF-22 research aircraft was conducted using a dynamic and aerodynamic model created in the Simulink modeling environment. Given a set of stability and control derivatives for the WVU Model YF-22 aircraft, the eigenvalues of the state-variable model were determined. These eigenvalues were then used to asses the stability and handling qualities of the aircraft and the validity of the simulation code.

Two Matlab functions were utilized to find the Eigenvalues and Eigenvectors of the aircraft system matrix. Both of these Matlab functions accepted a Simulink model as an input. The first of these was the Matlab 'trim' command which took the aircraft Simulation model and returned a list of trim states and their derivatives.

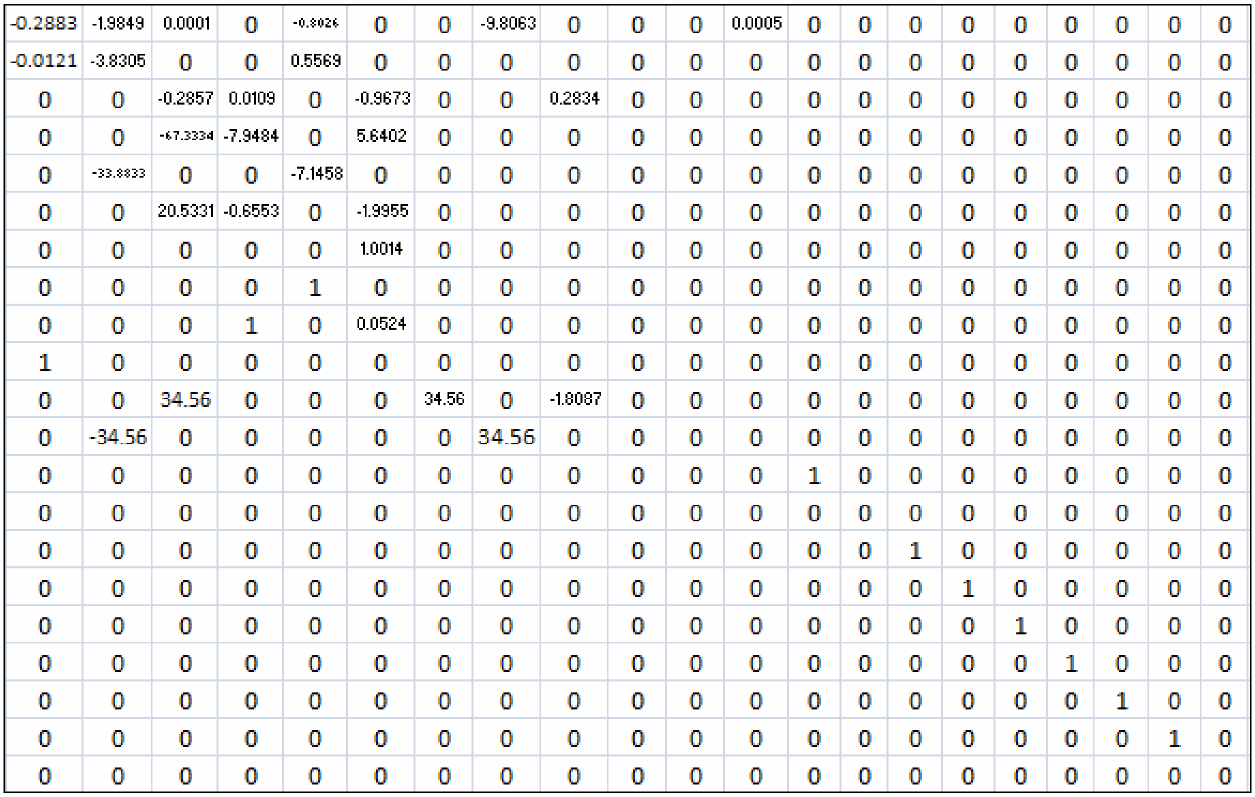
Once the trim states were found, the 'linmod' function was executed. For this function, the trim states and inputs along with the non-linear Simulink model were input into the function. With this, the 'linmod' function returned a system of linear state-space matrices. The only matrix that returned non-zero values was the A matrix. Once the A matrix was computed, the Eigenvalues and Eigenvectors were easily extracted for analysis.

After running the Matlab "trim" function, the values for x, u, y, and dx were found. Both u and y were empty vectors, while x was a 21 x 1 matrix and dx was a 12 x 1 matrix.

```
x =
  34.5600
   0.0524
        0
        0
        0
        0
        0
   0.0524
        0
        0
        0
 100.0000
        0
     -Inf
        0
  10.0000
 -10.0000
  10.0000
 100.0000
        0
        0
        0
        0
```

```
dx =
  -4.9823
   0.1509
   0.0000
   0.0001
   0.0000
  -0.0001
        0
        0
        0
  34.5600
        0
  -0.0000
```

After determining the values for x and u, these vectors were used to determine the A, B, C, and D matrices. Figure 9 shows the result for the A matrix, and the B, C, and D matrices are empty.



-0.2883	-1.9849	0.0001	0	-0.8026	0	0	-9.8063	0	0	0	0.0005	0	0	0	0	0	0	0	0	0
-0.0121	-3.8305	0	0	0.5569	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	-0.2857	0.0109	0	-0.9673	0	0	0.2834	0	0	0	0	0	0	0	0	0	0	0	0
0	0	-67.3334	-7.9484	0	5.6402	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	-33.8833	0	0	-7.1458	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	20.5331	-0.6553	0	-1.9955	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1.0014	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0.0524	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	34.56	0	0	0	34.56	0	-1.8087	0	0	0	0	0	0	0	0	0	0	0	0
0	-34.56	0	0	0	0	0	34.56	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	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	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 9: State Space Matrices after Linearization using ‘linmod’ Function in Matlab

Figure 10 displays the Eigenvalues of the YF-22 aircraft from the Simulink model after running the Matlab commands for “trim” and “linmod.” The figure displays both the longitudinal and lateral directional roots representing short period, phugoid, dutch roll, roll, and spiral. All of the aircraft modes are stable because they lie within the negative region of the real axis.

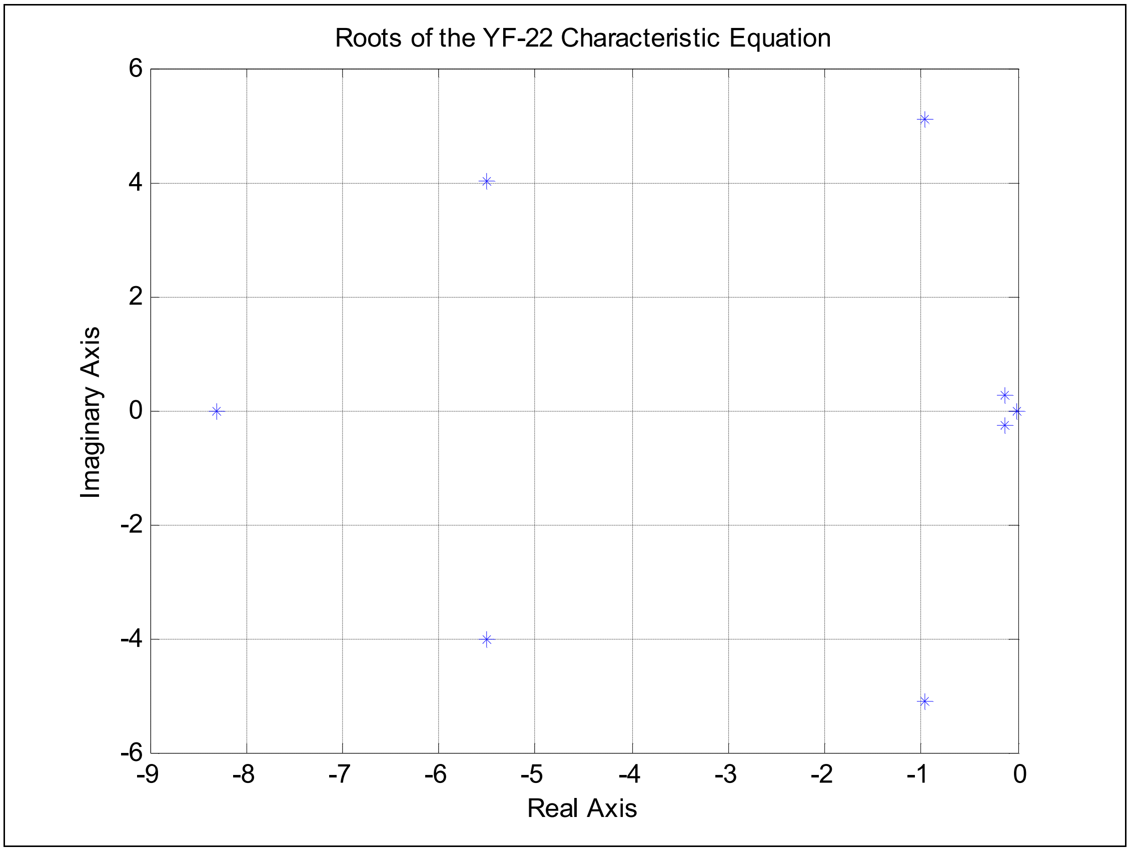


Figure 10: Roots of the YF-22 Characteristic Equation

The longitudinal dynamics are represented by two pairs of complex conjugate roots in the figure. The short period is characterized by high damping and a high natural frequency; therefore, the complex conjugate roots are as follows:

-5.4970 + 4.0128i
-5.4970 - 4.0128i

The phugoid is characterized by low damping and a low natural frequency, and they are as follows:

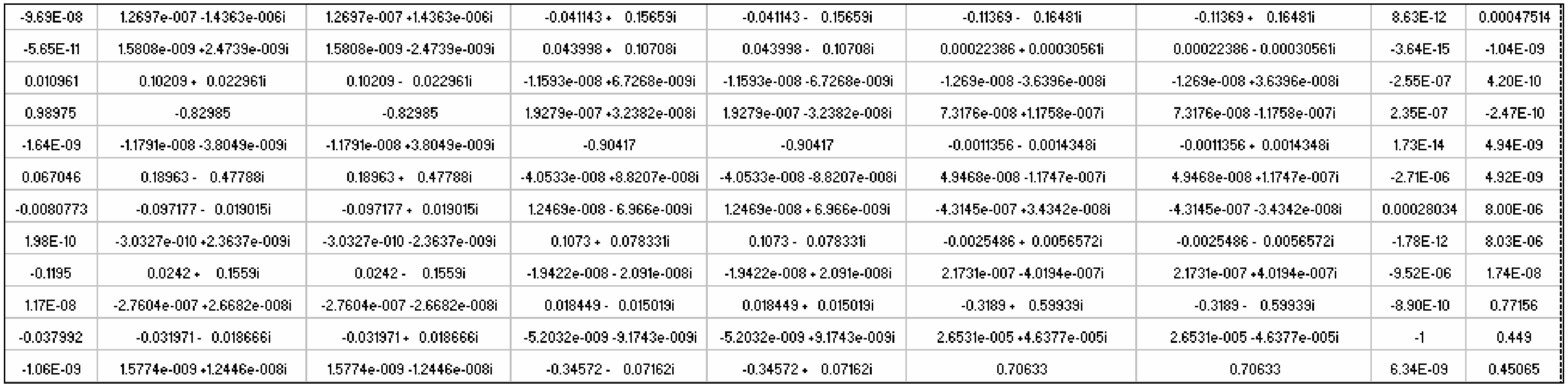
-0.1357 + 0.2618i
-0.1357 - 0.2618i

The lateral-directional dynamics include the roll, spiral, and dutch roll. The dutch roll is represented by a complex conjugate pair of roots, while the roll and spiral are both individual, real numbers. The dutch roll typically has a higher damping and natural frequency than the phugoid; however, it has typically lower values than the short period. The following pair of roots represents the dutch roll for the YF-22:

-0.9540 + 5.1110i
-0.9540 - 5.1110i

The spiral time constant is a small, real number and was found to be -0.0097. The roll time constant was much larger, which is typical for model aircraft, and was calculated to be -8.3120.

Figure 11 displays the calculated eigenvectors for the YF-22 aircraft.



-9.69E-08	1.2697e-007 -1.4363e-006i	1.2697e-007 +1.4363e-006i	-0.041143 + 0.15659i	-0.041143 - 0.15659i	-0.11369 - 0.16481i	-0.11369 + 0.16481i	8.63E-12	0.00047514
-5.65E-11	1.5808e-009 +2.4739e-009i	1.5808e-009 -2.4739e-009i	0.043998 + 0.10708i	0.043998 - 0.10708i	0.00022386 + 0.00030561i	0.00022386 - 0.00030561i	-3.64E-15	-1.04E-09
0.010961	0.10209 + 0.022961i	0.10209 - 0.022961i	-1.1593e-008 +6.7268e-009i	-1.1593e-008 -6.7268e-009i	-1.269e-008 -3.6396e-008i	-1.269e-008 +3.6396e-008i	-2.55E-07	4.20E-10
0.98975	-0.82985	-0.82985	1.9279e-007 +3.2382e-008i	1.9279e-007 -3.2382e-008i	7.3176e-008 +1.1758e-007i	7.3176e-008 -1.1758e-007i	2.35E-07	-2.47E-10
-1.64E-09	-1.1791e-008 -3.8049e-009i	-1.1791e-008 +3.8049e-009i	-0.90417	-0.90417	-0.0011356 - 0.0014348i	-0.0011356 + 0.0014348i	1.73E-14	4.94E-09
0.067046	0.18963 - 0.47788i	0.18963 + 0.47788i	-4.0533e-008 +8.8207e-008i	-4.0533e-008 -8.8207e-008i	4.9468e-008 -1.1747e-007i	4.9468e-008 +1.1747e-007i	-2.71E-06	4.92E-09
-0.0080773	-0.097177 - 0.019015i	-0.097177 + 0.019015i	1.2469e-008 - 6.966e-009i	1.2469e-008 + 6.966e-009i	-4.3145e-007 +3.4342e-008i	-4.3145e-007 -3.4342e-008i	0.00028034	8.00E-06
1.98E-10	-3.0327e-010 +2.3637e-009i	-3.0327e-010 -2.3637e-009i	0.1073 + 0.078331i	0.1073 - 0.078331i	-0.0025486 + 0.0056572i	-0.0025486 - 0.0056572i	-1.78E-12	8.03E-06
-0.1195	0.0242 + 0.1559i	0.0242 - 0.1559i	-1.9422e-008 - 2.091e-008i	-1.9422e-008 + 2.091e-008i	2.1731e-007 -4.0194e-007i	2.1731e-007 +4.0194e-007i	-9.52E-06	1.74E-08
1.17E-08	-2.7604e-007 +2.6682e-008i	-2.7604e-007 -2.6682e-008i	0.018449 - 0.015019i	0.018449 + 0.015019i	-0.3189 + 0.59939i	-0.3189 - 0.59939i	-8.90E-10	0.77156
-0.037992	-0.031971 - 0.018666i	-0.031971 + 0.018666i	-5.2032e-009 -9.1743e-009i	-5.2032e-009 +9.1743e-009i	2.6531e-005 +4.6377e-005i	2.6531e-005 -4.6377e-005i	-1	0.449
-1.06E-09	1.5774e-009 +1.2446e-008i	1.5774e-009 -1.2446e-008i	-0.34572 - 0.07162i	-0.34572 + 0.07162i	0.70633	0.70633	6.34E-09	0.45065

Figure 11: Eigenvectors of the YF-22 Aircraft ModelEngine Modeling

The engine on the YF-22 is a single jet engine, and it had to be accurately modeled in the aircraft simulation. The "uprop" vector input is the value output from the "Controls" block. The initial gain was calculated to be $1/T_E$, where T_E is the engine thrust control time constant. A single feedback loop was created to model a first order transfer function that describes the behavior of the engine. An integrator was included in the feed forward section of the loop to set a pole on the bottom of the transfer function. The feedback was multiplied by a gain of $1/T_E$. The output of the engine model was a single value representing the thrust, which was alterable as the slider on the joystick was moved. The other elements included in Figure 12 were to account for prop wash and torque effects from the propeller if necessary in the aircraft design. Since the YF-22 has a jet engine, these components were excluded from the model.

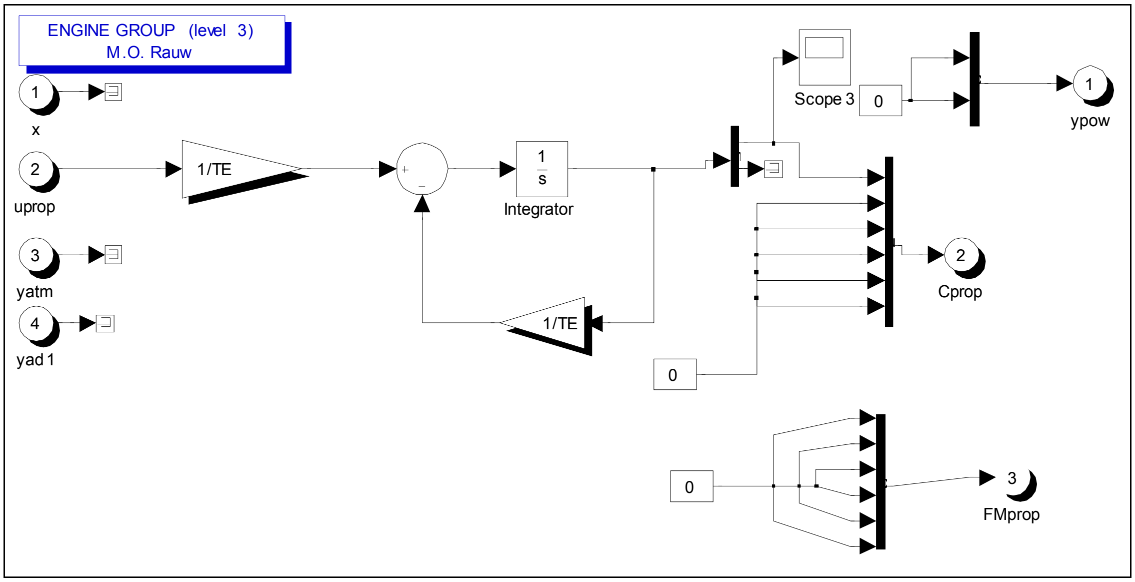


Figure 12: WVU YF-22 Jet Engine Model

Figure 13 shows a sample of engine output from the designated model. The engine was taken from full throttle to low throttle, back up to full throttle between 400 and 1000 time steps, and then down to low throttle again incrementally before being taken back up again. The response is not instantaneous, which is similar to the behavior of a real jet engine. This is due to the successful modeling of the first order transfer function for the jet engine.

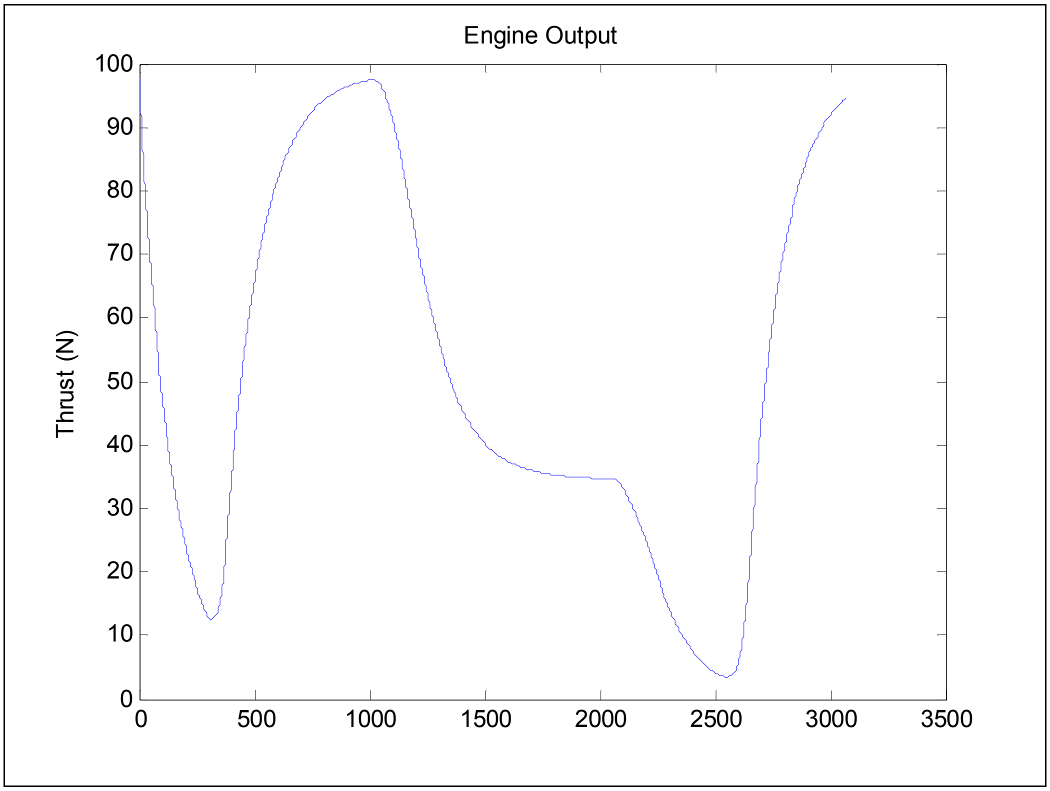


Figure 13: Engine Output Sample

Wind Modeling

Aircraft do not always fly in calm air; therefore, wind profiles had to be modeled in the simulation. A subsystem was developed with an attached switch to give the user a choice of whether or not to include the wind models. Figure 14 depicts this subsystem layout, which takes the altitude, Euler angles, and aircraft velocity into account for the wind models.

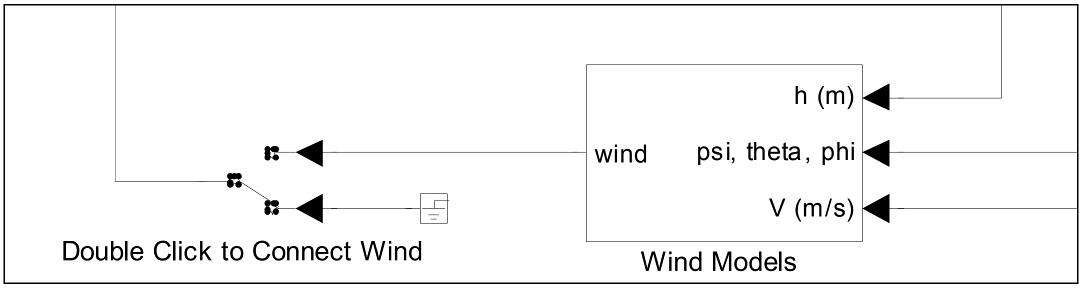


Figure 14: Wind Model Subsystem

Figure 15 shows the layout inside of the “Wind Models” subsystem. The wind conditions modeled included the wind shear and the Dryden Wind Turbulence Model. A block from the Aerospace Blockset was included to convert the Euler angles into the directional cosine matrix which was required as an input for both wind models.

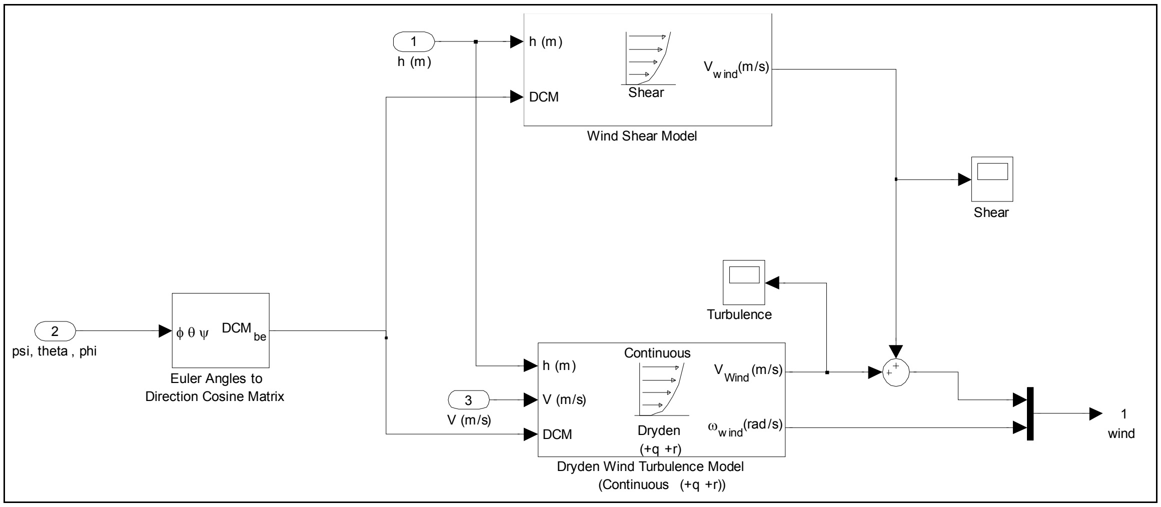


Figure 15: Wind Shear and Dryden Wind Turbulence Models

Figure 16 provides an example of the shear wind profile with turbulence effects during a flight simulation of the aircraft. Near the end of the simulation (around 35 seconds) the aircraft began to roll severely due to the strong wind shear.

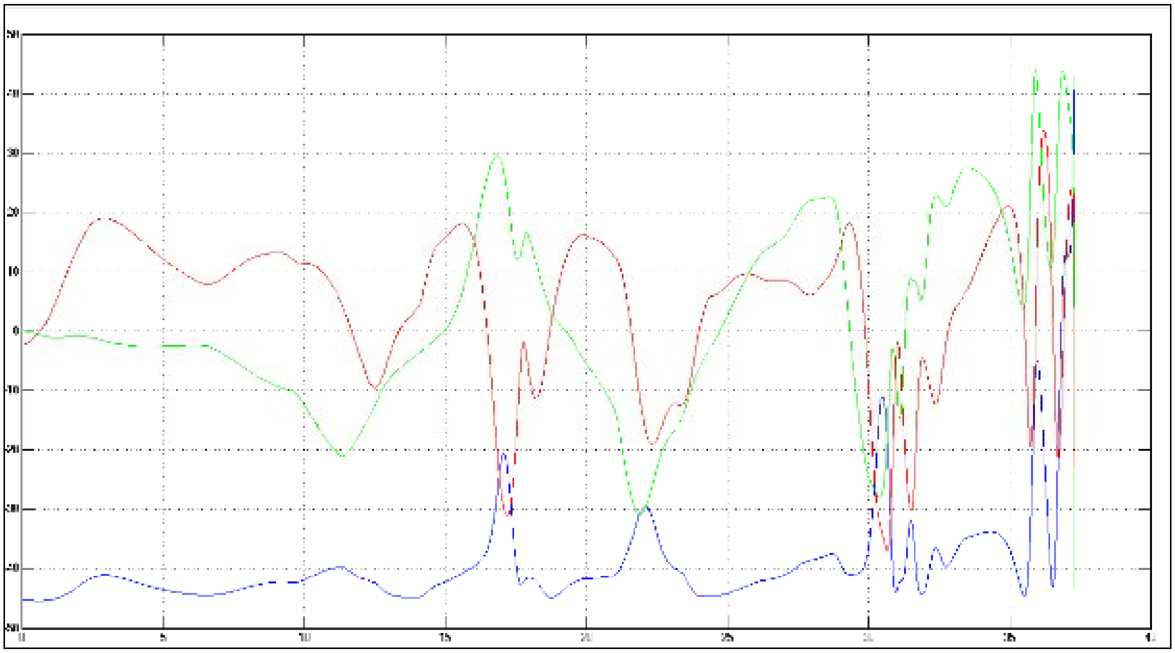


Figure 16: Wind Shear and Turbulence Model Response

Virtual Reality Toolbox

The Matlab Virtual Reality Toolbox was used to provide visualization to the user during simulations. In order to incorporate this toolbox, a specific “world” had to be downloaded to provide background information for the flight environment and an aircraft design for the F-22 was used. The "3D Visulaization" block is shown in Figure 17. It requires the Euler angles and the positional information of the aircraft as inputs to the visualization environment.

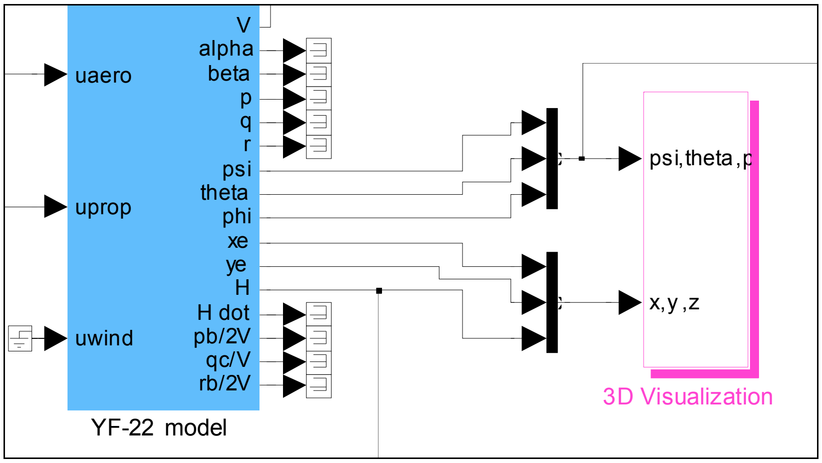


Figure 17: Virtual Reality 3D Visualization Interface

Figure 18 details the "Visualization" subsystem. WVU Researchers used this subsystem to model five aircraft formation flight, so there are five individual aircraft options in the Virtual Reality (VR) Sink. The blue aircraft was selected to for use in the simulation; therefore, the other four aircraft were grounded. The Euler angles and the positional information were fed into a block that computed them into rotational and translational information, respectively. The "Closed Universe" block was used to limit the translational position being sent to the VR Sink.

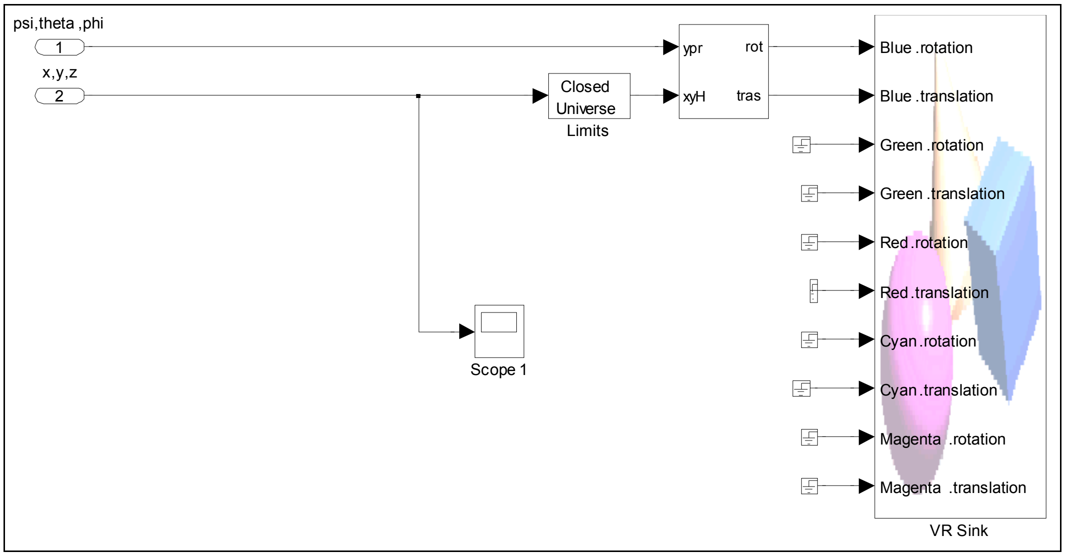


Figure 18: Virtual Reality Sink with Inputs for the Blue Aircraft

Figure 19 provides an example of the simulation environment with a view from behind the blue aircraft. The views may be changed during simulation to allow the user to have a cockpit view, a side view, and a front view as well. The toolbar on the bottom of the screen allows the user to change the view incrementally to suit his or her needs for that particular simulation.

Figure 19: YF-22 Virtual World

Failure Modeling

Locked Control Surface

For the modeling of locked control surfaces, the manipulation of the simulation consists of altering the control surface input deflections. This alteration consists of modeling a failure scenario in which a control surface gets locked at a specific deflection angle at a specific time, and the failure occurs over some transition time. To do this, a Simulink subsystem was developed. This Simulink subsystem was then placed within each of the six control surface input signals.

The basic logic of this Simulink subsystem consisted of two layers. The first layer consists of determining whether or not the user selected a failure to occur on a particular surface. This was done by introducing binary flags for each surface. If it was determined that a failure is to occur on a particular surface, the second layer of logic consisted of determining if the time of the simulation was before or after the user indicated time of failure. Finally, if it was determined that the time of the simulation was equal to or after the user defined time of failure then the deflection for that particular surface was set to the user defined locked position. This failure at a defined locked position, it also set to occur over a specific transition time in the form of first order dynamics. Figure 20 shows the Simulink subsystem developed to inject a locked surface failure.

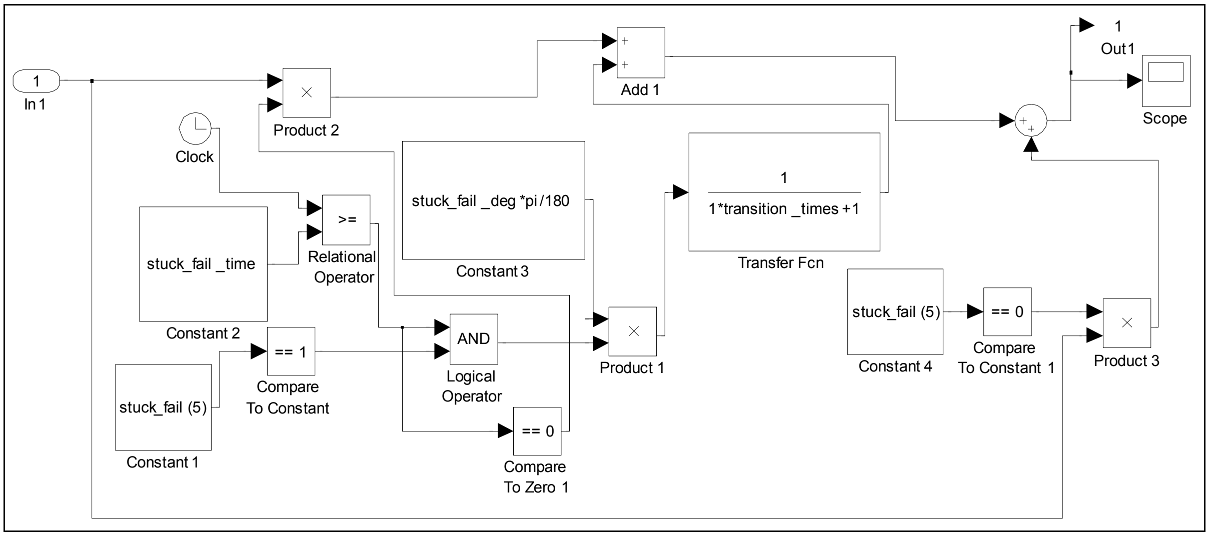


Figure 20: Locked Control Surface Conditional Subsytem

In order to display that this model was successful, two sets of plots in which two different surfaces were locked at two different deflections at two different times are provided.

The first scenario shows the left stabilator was locked at 4 degrees starting at 10 seconds, as shown in Figure 21.

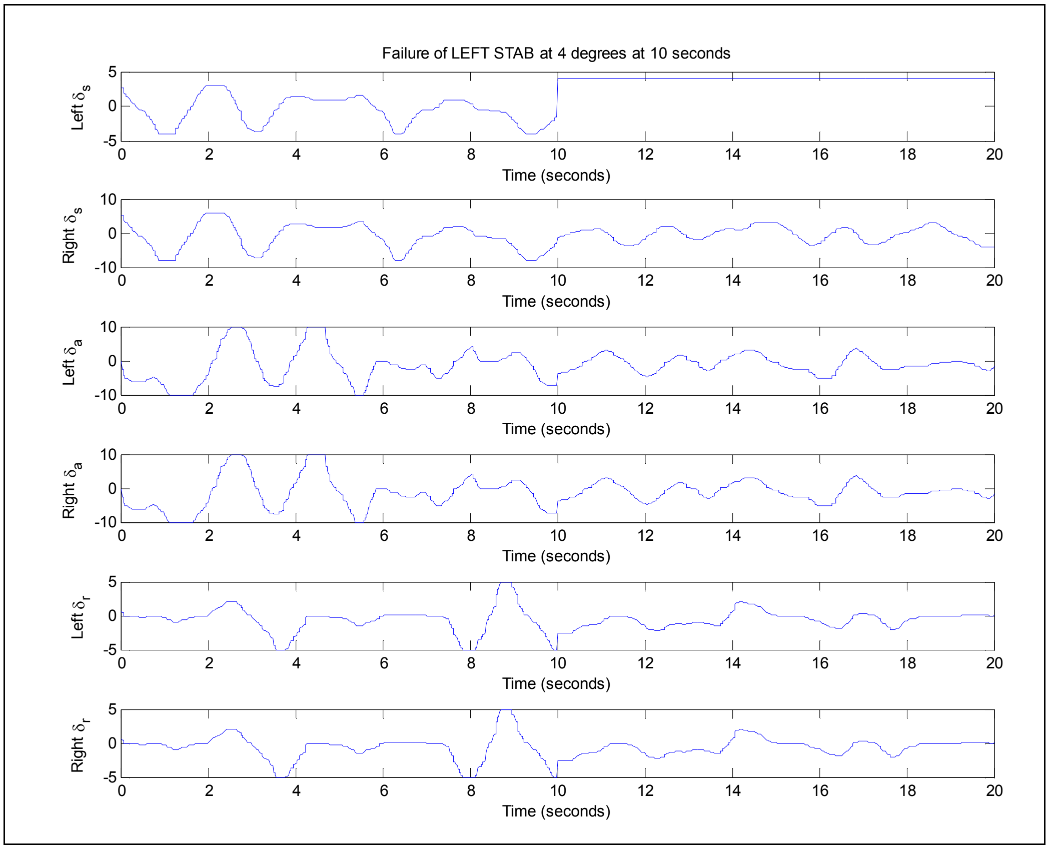


Figure 21: Locked Left Stabilator at 4 Degrees at 10 Seconds

For the second scenario, the right aileron was locked at 5 degrees beginning at 6 seconds, as displayed in Figure 22.

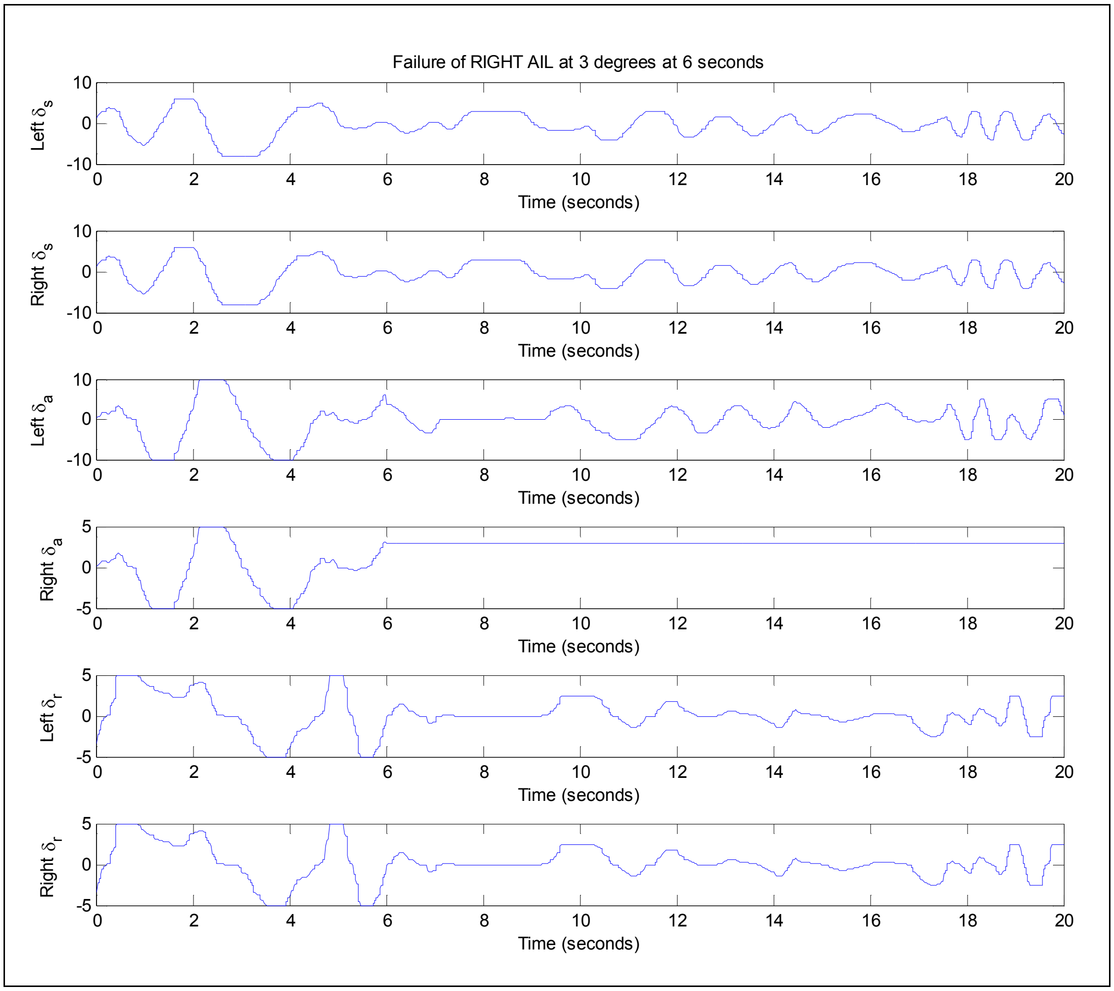


Figure 22: Locked Right Aileron at 5 Degrees at 6 Seconds

In summary, Figures 21 and 22 display that the modeling of the locked control surfaces in which the user can define which surface to fail, at a specific deflection, and starting a specific time over a transition period.

Damaged or Missing Control Surface

In order to the model the missing portion of a control surface, the actual aerodynamic forces and moments aerodynamic must be manipulated. This manipulation was performed through the use efficiency parameters, as discussed in the literature review section of this report. These efficiency parameters are values between 0 and 1 and are user selected within the graphical user interface of the simulator. In addition, the time of failure is user selected.

Once the efficiency parameters of the control surfaces to be failed are indicated and the time of failure is indicated, a new aerodynamic forces and moment coefficient matrix is defined to represent the aircraft model post failure. Within the simulation, the aerodynamic forces and moment coefficient matrix is referred to as the aircraft model matrix, or AM matrix. The calculation of post-failure AM matrix is performed as an element-wise matrix multiplication between the original AM matrix and a matrix filled with ones unless user specified efficiency parameters are defined. If a user specified efficiency parameter is defined for a particular control surface, then that parameter values is placed into a matrix in the appropriate location so that the force and moment coefficients corresponding to that particular control surface is affected after the element-wise multiplication operation.

Once the post-failure AM matrix has been calculated, a Simulink scheme was developed to allow the user specified time of failure to determine when the simulation would implement the failure model, as shown in Figure 23.

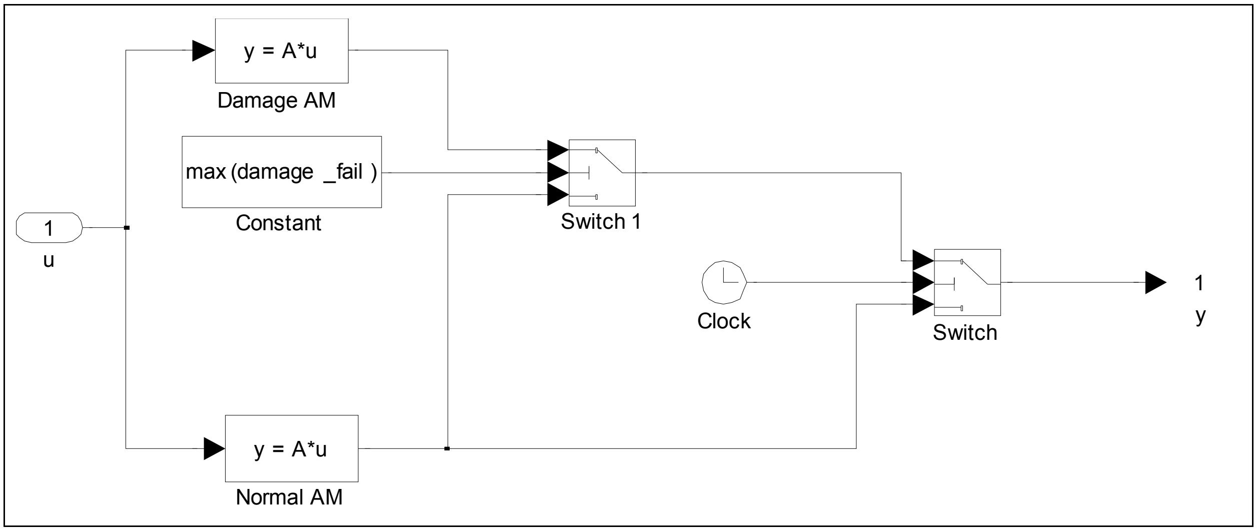


Figure 23: Damaged or Missing Control Surface Failure Block

Within Figure 23, switches are compared to the clock and the user specified time of failure. Once the time of the simulation is equal to or past the time of failure, the post-failure AM matrix is fed into the simulation. The post-failure AM matrix is labeled as "Damage AM" in Figure 23.

Sensor Failure

Undetected sensor failures onboard an aircraft can lead to serious dynamic problems of the closed-loop aircraft motion, and potentially unrecoverable flight conditions[3]. The underlying assumption is that sensor information is fed back into the flight computer as part of the control scheme. All modern aircraft have some form of this closed-loop control design, however the analytical model used in the simulation of the YF-22 test aircraft is an open-loop simulation; there is no control scheme or feedback loops used in the model. Therefore, an erroneous output of the simulation would only negatively affect the analysis of the flight characteristics instead of its actual input or performance or handling qualities.

Despite the absence of a control scheme used for the simulation, outputs were created to be 'sensor' outputs of an actual aircraft and compared to the outputs of the analytical model in the Simulink environment. These sensor readings were taken from the model and manipulated to reproduce common sensor failures.

Two common sensor failures simulated in this method are a bias, and the presence of noise. A signal bias is when the measured output of the sensor is exaggerated compared to the actual output. A constant multiplier is used to produce this amplified erroneous signal output, shown in Figure 24 below is the gain block.

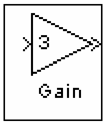


Figure 24: Simulink *gain* block used to amplify the input

A noisy signal was modeled using a white noise source in the Simulink environment and combined with the sensor output. The noise intensity and sample time can be varied in these components. A white noise block is shown in Figure 25.

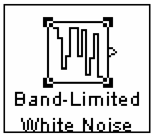


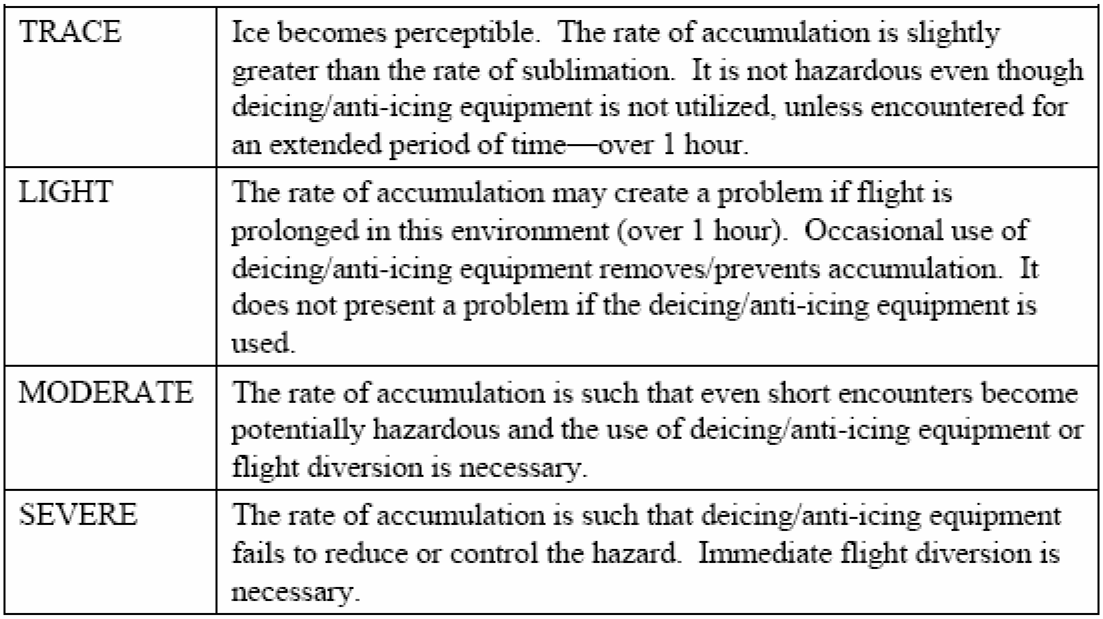
Figure 25: Simulink block to introduce continuous noise to a signal

Icing

Modeling ice accretion on aerodynamic surfaces has historically been performed in a wind tunnel environment. Instead of experimentally determined data from the wind tunnel, aerodynamic coefficients affected by ice buildup on the leading edge of the airfoil are instead estimated based on trends observed from previous experiments. Particularly, the drag and lift coefficients of both wings are separated from one another and scaled according to the specific icing condition.

The Federal Aviation Administration (FAA) considers four tiers of ice occurrence on an aircraft. The following table shows the FAA classification:

Table 2: FAA Icing Categories[7]



TRACE	Ice becomes perceptible. The rate of accumulation is slightly greater than the rate of sublimation. It is not hazardous even though deicing/anti-icing equipment is not utilized, unless encountered for an extended period of time—over 1 hour.
LIGHT	The rate of accumulation may create a problem if flight is prolonged in this environment (over 1 hour). Occasional use of deicing/anti-icing equipment removes/prevents accumulation. It does not present a problem if the deicing/anti-icing equipment is used.
MODERATE	The rate of accumulation is such that even short encounters become potentially hazardous and the use of deicing/anti-icing equipment or flight diversion is necessary.
SEVERE	The rate of accumulation is such that deicing/anti-icing equipment fails to reduce or control the hazard. Immediate flight diversion is necessary.

It can be seen from this table that the determining of icing conditions is very subjective. Further statements made in Jeck 2001, from which this table is taken, says that an aircraft under severe icing conditions often has to use all available power to remain in steady level flight. Based on this observation, aerodynamic coefficients are scaled down according to the point of barely achieving steady level flight. This flight condition will be severe icing condition, and the moderate, light and trace icing conditions will be scaled down accordingly.

To model the degradation of the lift force of the aircraft the C_Z coefficient was monitored to illustrate the degraded vertical force. For small angles of attack (α), the body axes of the plane can be considered equal to global axes of the aircraft. Therefore, the forces in the negative Z-axis of the plane can be considered the lift of the aircraft. C_X, C_Y, C_Z, C_l, C_m and C_n, are the outputs of the state variable model.

The scaled down coefficient $C_{Z\alpha}$ is actually $C_{L\alpha}$. Due to their relationship described above, this substitution is viable. A multiplying factor is applied to this term, until this lift coefficient is scaled down enough to produce the flight condition as where steady level flight is barely achievable. The resulting multiplicative factor corresponds to the debilitated lift effect of severe icing conditions. The multiplicative factor is scaled between the severe icing value and unity to produce effects of the moderate, light, and trace icing flight conditions.

The multiplying factor Cice, can be seen in Appendix B in the ln2nl.m Matlab code. The user defines the icing severity in the GUI shown in Figure 37, which determines the value of Cice in the manner previously explained.

Results

Locked Control Surface

In order to show both the simulation's capability of locked surface failure modeling, and alos locked surface failure effects on aircraft dynamics, an experiment was performed in which three separate tested were performed. Each test corresponded to a different control surface failed at a specific time, at a specific magnitude, over a specific transition. In order to show the failure effects on aircraft kinematics and dynamics the joystick was used to fly at steady level flight, then the same control inputs were fed into the simulation for each failure scenario. The results of the three tests are shown below.

Damaged or Missing Control Surface

For the damaged or missing control surface simulation testing, each of the six individual control surfaces had the capability to fail. Figure 26 compares the roll angle for two failures of the left aileron. The blue line corresponds to a 20% damaged surface (80% left aileron) and the red line corresponds to a missing left aileron. As the surface was increasingly damaged, the roll angle effectiveness was decreased for the aircraft. Both simulation results derived from the same pilot input where the aircraft was attempted to maintain steady, level flight.

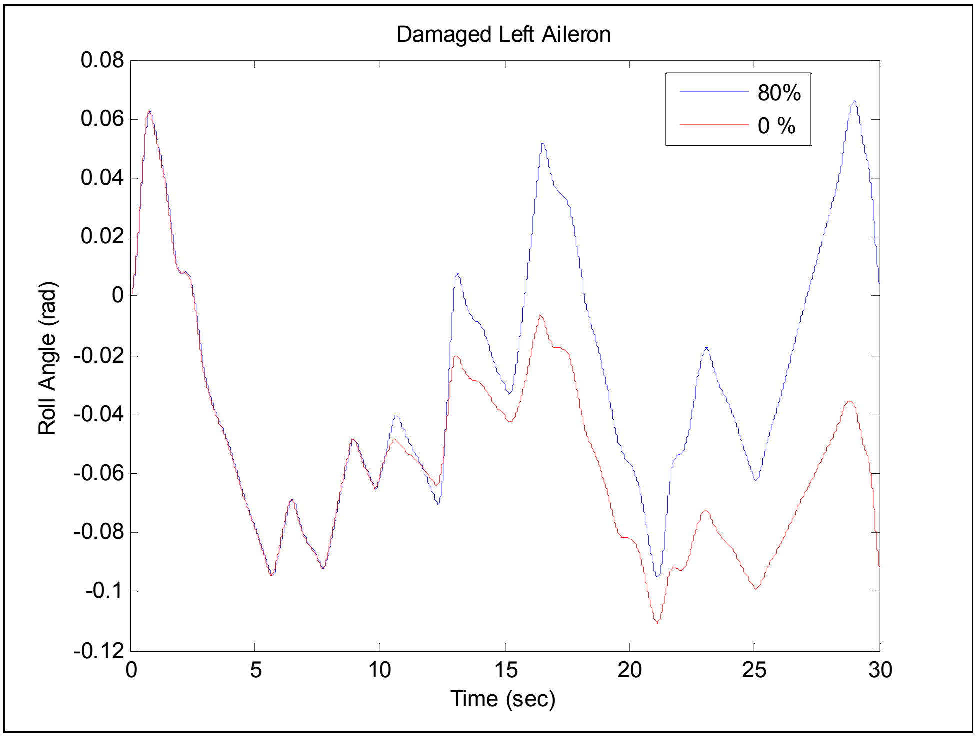


Figure 26: Damaged Left Aileron Induced Roll Angle Variations

Figure 27 compares a 50% damaged left rudder, shown in red, to a healthy rudder, shown in blue. The figure shows the roll, pitch, and yaw rates and angles and their behavior due to the damage. The damaged rudder causes less effectiveness in maneuvers requiring the surface. It also induced a slight roll to the aircraft compared to when the surface was 100%.

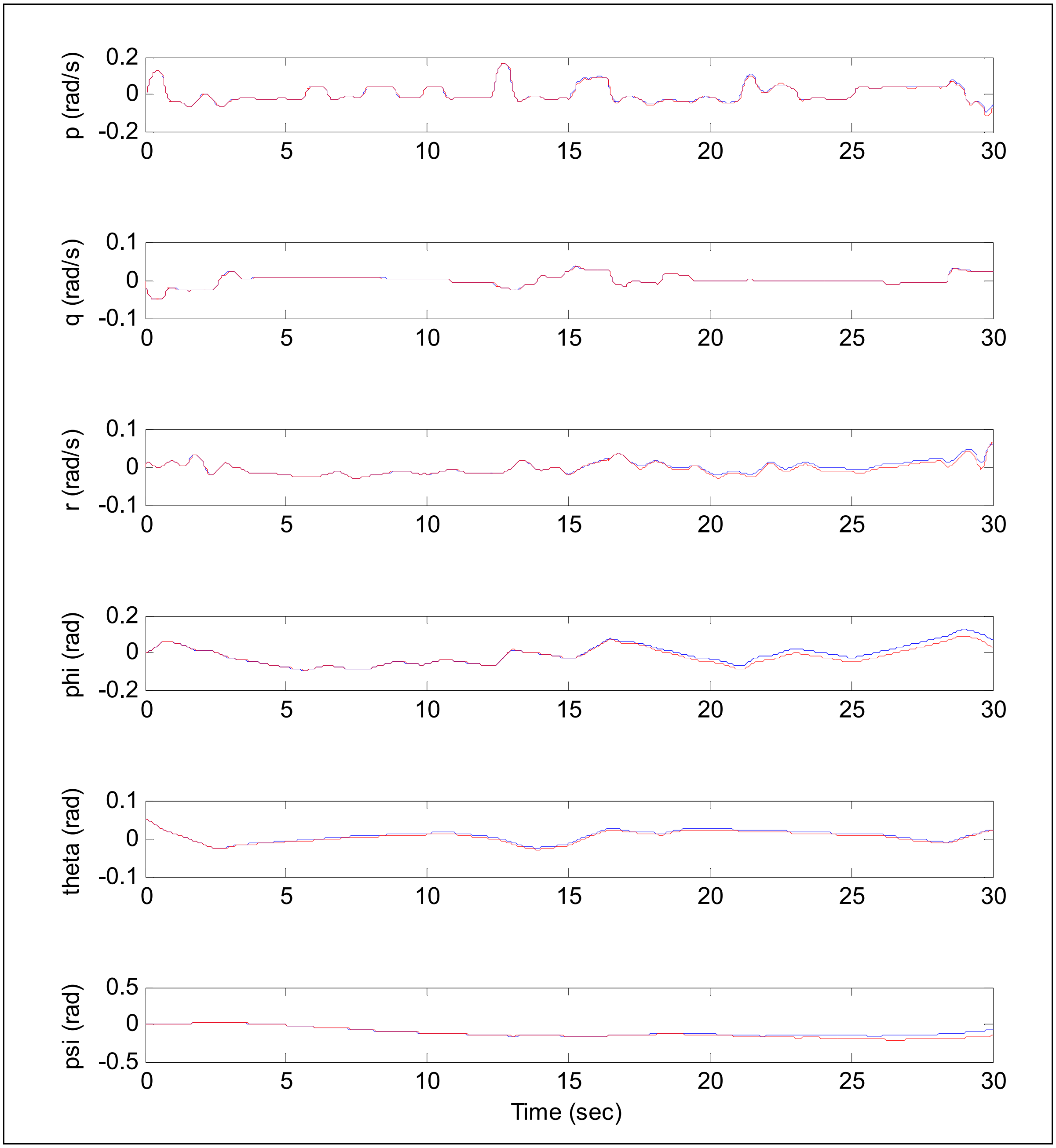


Figure 27: 50% Damaged Left Rudder versus Healthy Rudder

Figure 28 compares a 50% damaged right stabilator, red, to a healthy right stabilator, blue. It includes a comparison of the roll, pitch, and yaw rates and angles. The damage creates a less effective longitudinal surface, thus altering the aerodynamics of the aircraft, just as the lateral-directional surfaces in the previous examples. The pitch rate and pitch angle are significantly less for the same input with a damaged control surface.

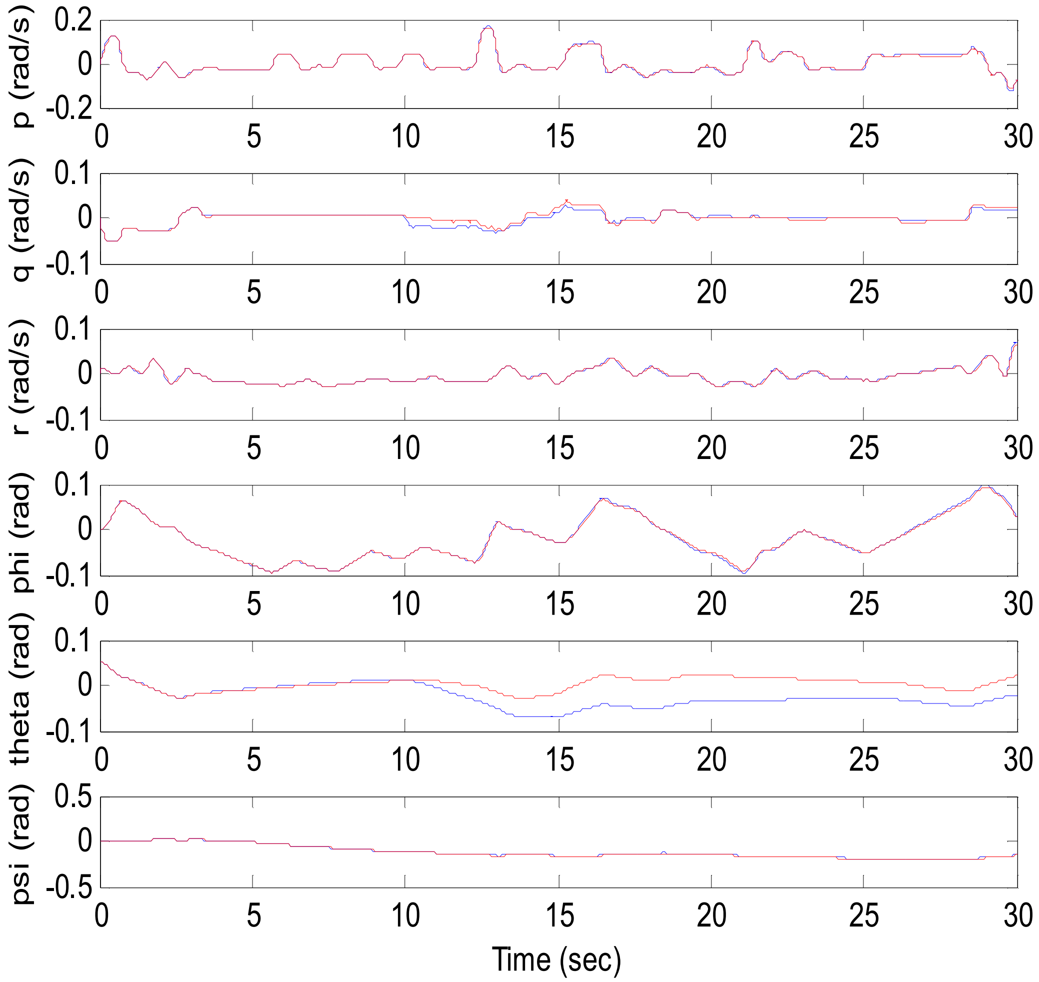


Figure 28: 50% Damaged Right Stabilator versus Healthy Stabilator

Any of the six surfaces may be damaged or missing to any user-specified degree during simulation. The less control surface that is available, the less effective a maneuver is for the aircraft, which is summarized in the results from the test simulations.

Sensor Failure

In Figure 29, the bank angle gyro failed with a constant bias of four times the actual value. The failure occurred at a simulation time of 10 seconds, and prior to this time, the sensor functioned as expected.

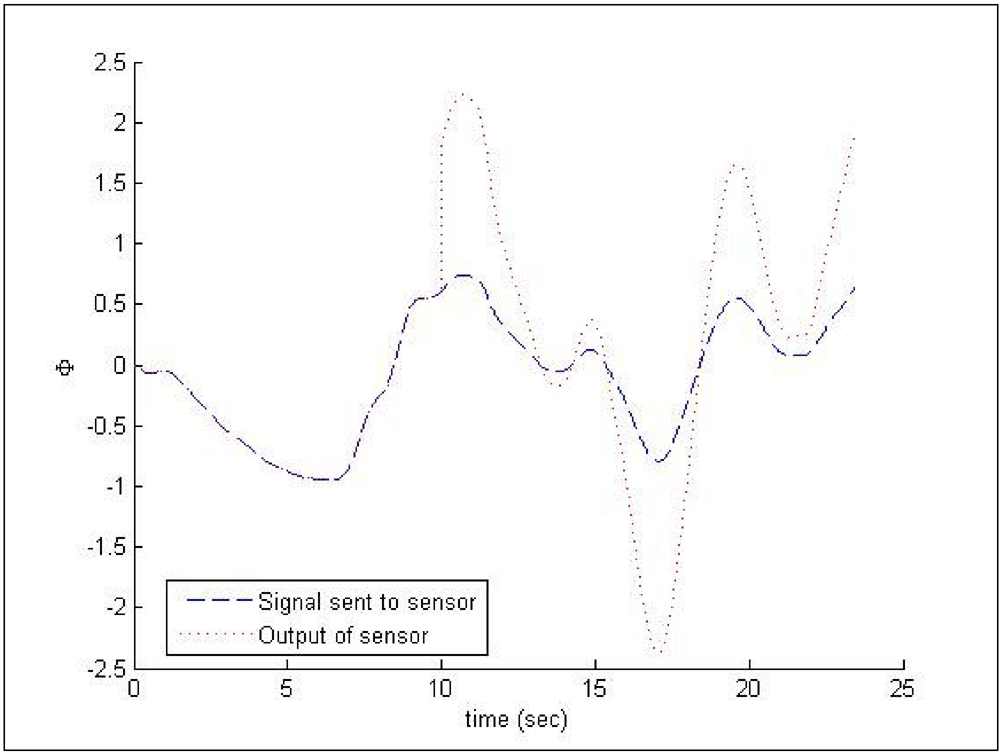


Figure 29: Phi sensor failure

In Figure 30, the alpha sensor became saturated with noise, beginning at a failure time of 10 seconds. These failures did not have a significant effect on the aircraft because the simulator did not incorporate a controller.

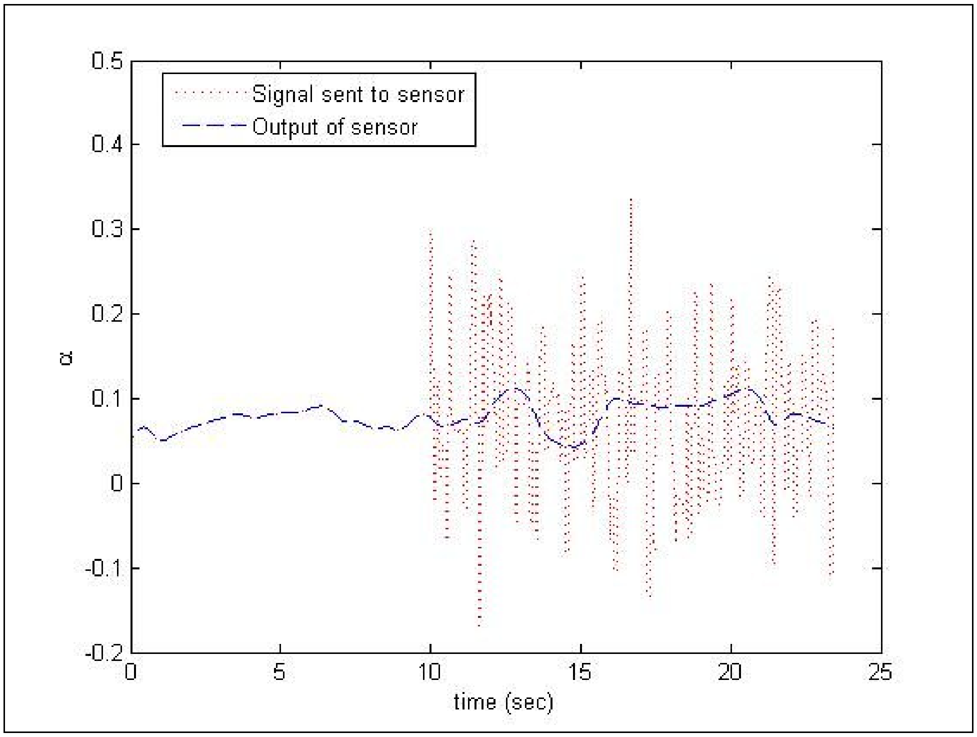


Figure 30: Alpha Sensor Failure

Icing

By changing Cice manually until maximum lift was needed just to remain at the same altitude, it was found that Cice = 0.23 after several iterations. This means that 23% $C_{L\alpha}$ reproduces the lift signature of an aircraft with a severe icing buildup on the wings. The medium, light and trace icing conditions were then scaled accordingly. Medium icing is 50% $C_{L\alpha}$, light icing is 75% $C_{L\alpha}$, and trace icing condition is 95% $C_{L\alpha}$. These effects were tested and determined to be valid given the FAA icing categories, Table 2: FAA Icing Categories

Effects of a degraded $C_{L\alpha}$ were observed by running several test flights at each icing condition; severe, moderate, light and trace. For each condition, steady state flight conditions were achieved by pilot input and the subsequent steady-state angle of attack and Cz values were recorded. Several data points were collected for each condition in order to observe the Cz vs. alpha trend for a range of 14 degrees. The following plot shows these results.

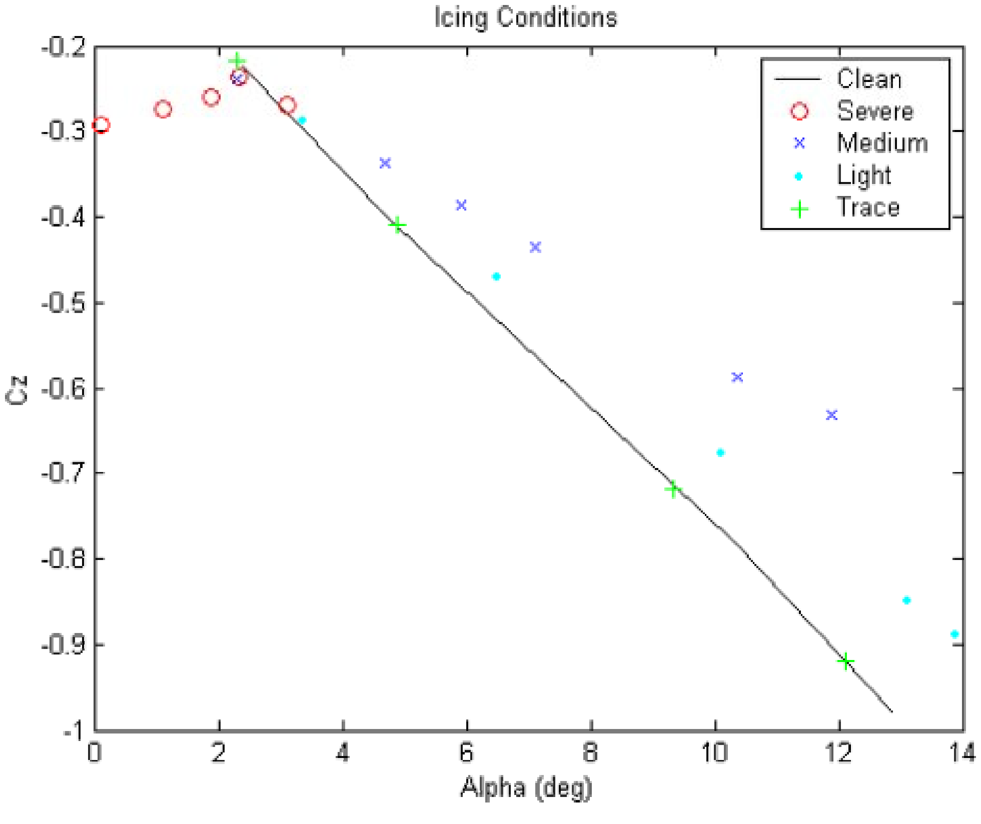
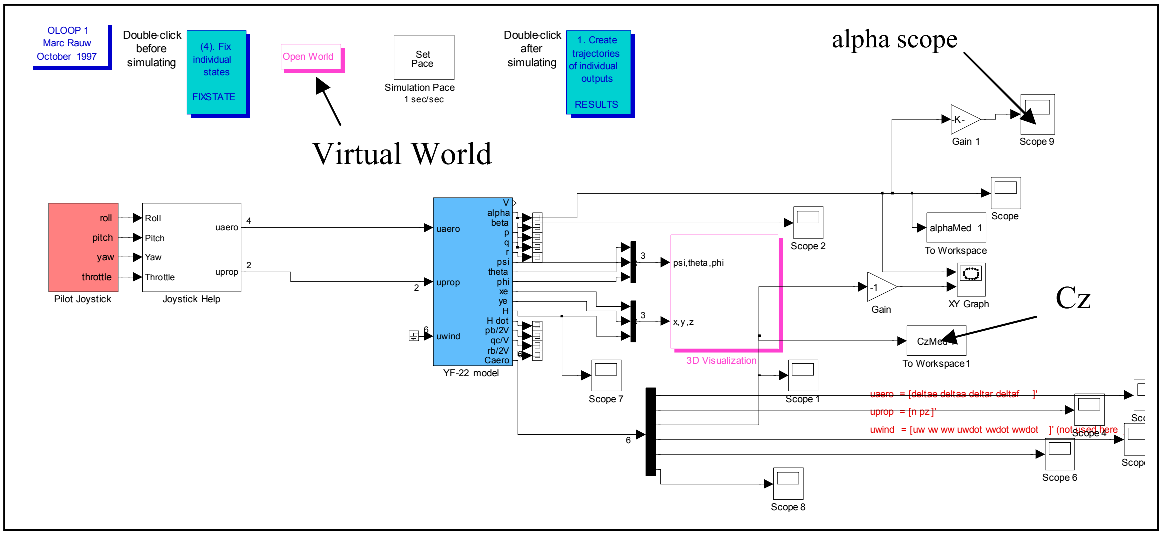


Figure 31: Icing effects on Cz_α

In this modeling environment the Z-direction is positive downwards, $C_Z = -C_L$ which explains the increasing negative values of C_Z. The only condition with unexpected results is the severe icing condition. It proved difficult to achieve steady-state flight conditions at various angles of attack when the lift of the aircraft is deteriorated to such extent. From the data points at 2.3 and 3.1 degrees, it is possible that a trend might exist as expected with the other icing conditions.


alpha scope

Virtual World

Cz

Figure 32: Simulink top layer showing extraction of the Cz coefficient from the aircraft equations

The figure above is the manipulated Simulink model to record the Cz and alpha data as the simulations were being ran. The steady-state flight conditions were achieved by observing the alpha scope and using the virtual world (Figure 19) to maintain lateral stability, which was difficult due to the joystick sensitivity.

Conclusions

A successful simulation was developed for the WVU YF-22 research aircraft which incorporated failure scenarios. The three pairs of left and right control surfaces were decoupled so there was individual control of each surface. The decoupling of the control surfaces allowed the locked control surface and missing or damaged control surface failures to be modeled with success. Sensor failures were also modeled successfully, although they did not affect the flight of the aircraft because no controller was incorporated in the simulation. A ‘severe’ icing condition was found such that no thrust is available for climb; other levels of icing were determined from this condition. Icing on the wing surfaces was modeled to show the decreased lift from the various stages of icing occurrence on the wing as described by the FAA. Due to the locked individual control surface failure, a coupling between the longitudinal and lateral-directional dynamics was observed. The simulator was successful for modeling the described failures, and the GUI was modeled so different aircraft models may be incorporated into the system if desired for future use.

Bibliography

[1] Barnhart, Dickes, Gingras, Ratvasky. *Simulation Model Development for Icing Effects Flight Training.* NASA/TM 2003-212115. April 2003.

[2] Baruh H., Choe K. *Sensor-Failure Detection Method for Flexible Structures*. AIAA Journal of Guidance, Control, and Dynamics. 1987; Vol. 10, no 5, 474-482.

[3] Campa, Fravolini, Napolitano, Seanor. *Neural-Networks Based Sensor Validation for the Flight Control System of a B777 Research Model*. WVU Internal Technical Report

[4] Cebeci, Kafyeke. *Aircraft Icing*. 2003. Annual Review, Fluid Mechanics 35:11-21.

[5] Haller, Grant M. *Boeing Dreamliner 787 Simulation.* Seattle Post-Intelligencer. May 23, 2007. http://seattlepi.nwsource.com/photos/.

[6] *New Pilot Program.* Randy's Journal. The Boeing Company. February 2007. http://www.boeing.com/randy/archives/2007/02/new_pilot_progr.html.

[7] Jeck. "A History and Interpretation of Aircraft Icing Intensity Definitions and FAA Rules for Operating in Icing Conditions". DOT/FAA/AR-01/91. Nov 2001.

[8] Perhinschi, Mario. *Handout #1: Introduction to Flight Simulation.* Advanced Flight Simulation. Department of Mechanical and Aerospace Engineering, West Virginia University. Spring 2008.

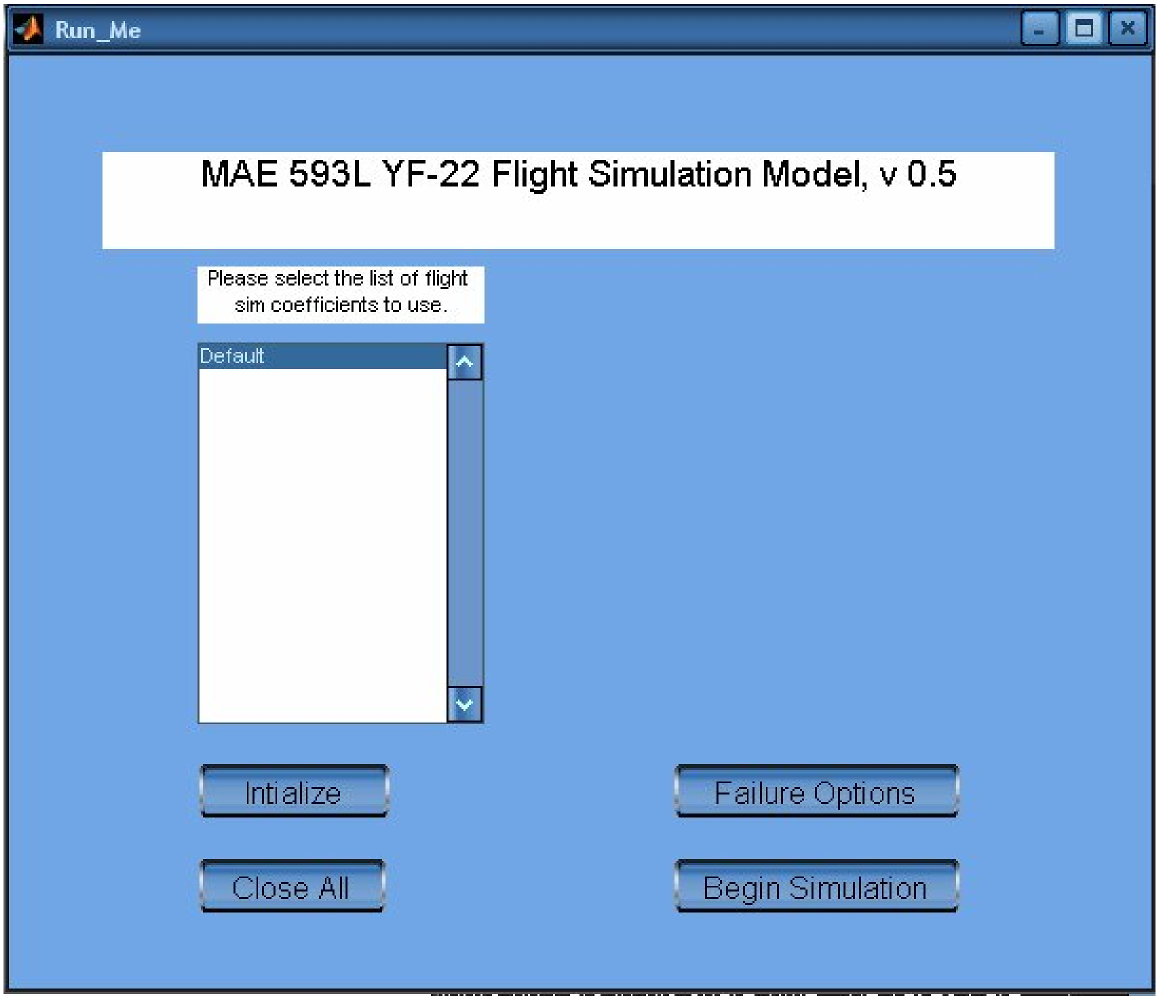
[9] Perhinschi, Mario, Campa, G., Napolitano, M.R., Lando, M., Massotti, L., Fravolini, M.L. *Modeling and Simulation of Failures for Primary Control Surfaces.* American Institute of Aeronautics and Astronautics, 2002.

[10] Rauw, Marc. *FDC, A Simulink Toolbox for Flight Dynamics and Control Analysis*. 2nd ed. May 10, 2001.

[11] Stevens, Brian L., Lewis, FrankL. *Aircraft Control and Simulation.* 2nd Edition. John Wiley & Sons, Inc. Hoboken, New Jersey. 2003.

Appendix A: GUI User Manual

To launch the application, execute the M-file, Run_Me.m, from the MATLAB command line. Upon this action, the window shown in Figure 33 will become available. The list shown is intended to be a list of available aircraft to simulate; however, currently there is only a default of the YF-22 Model. Clicking on the “Initialize” button loads the selected model into the workspace and creates a .mat file later to be loaded in the Simulink simulation. The “Close All” button forces a close of every active window and exits the simulation. The “Failure Options” button opens the Failure Modes window. The “Begin Simulation” button opens the appropriate Simulink model and initializes the remaining variables.



Run_Me

MAE 593L YF-22 Flight Simulation Model, v 0.5

Please select the list of flight sim coefficients to use.

Figure 33: Main Menu to Initialize the Simulation

The Failure Modes window is shown in Figure 34. From here, a user can select to edit the various types of failures and their respective parameters. The “Close” button closes the window and sends the failure selections back to the main window.

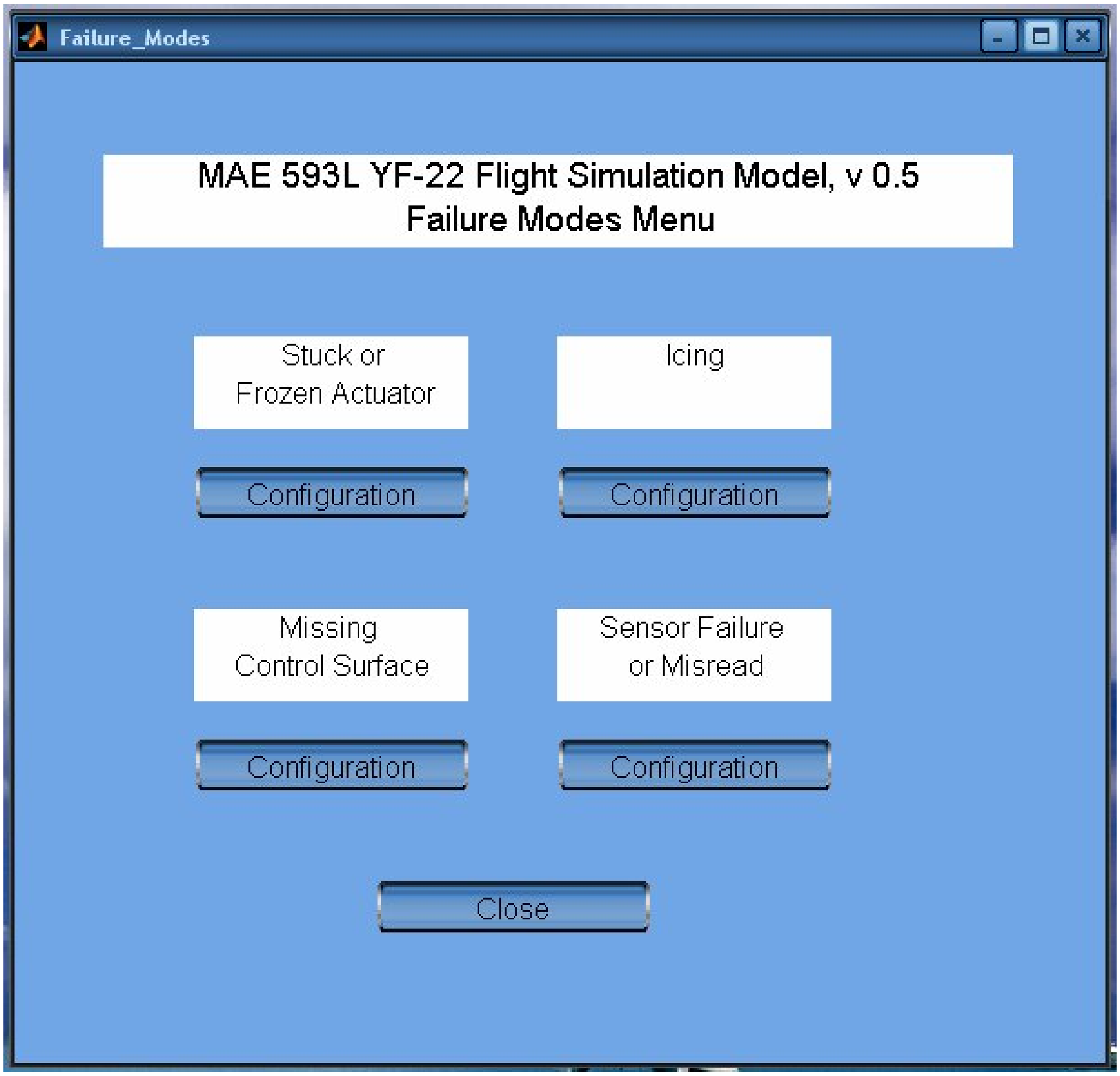
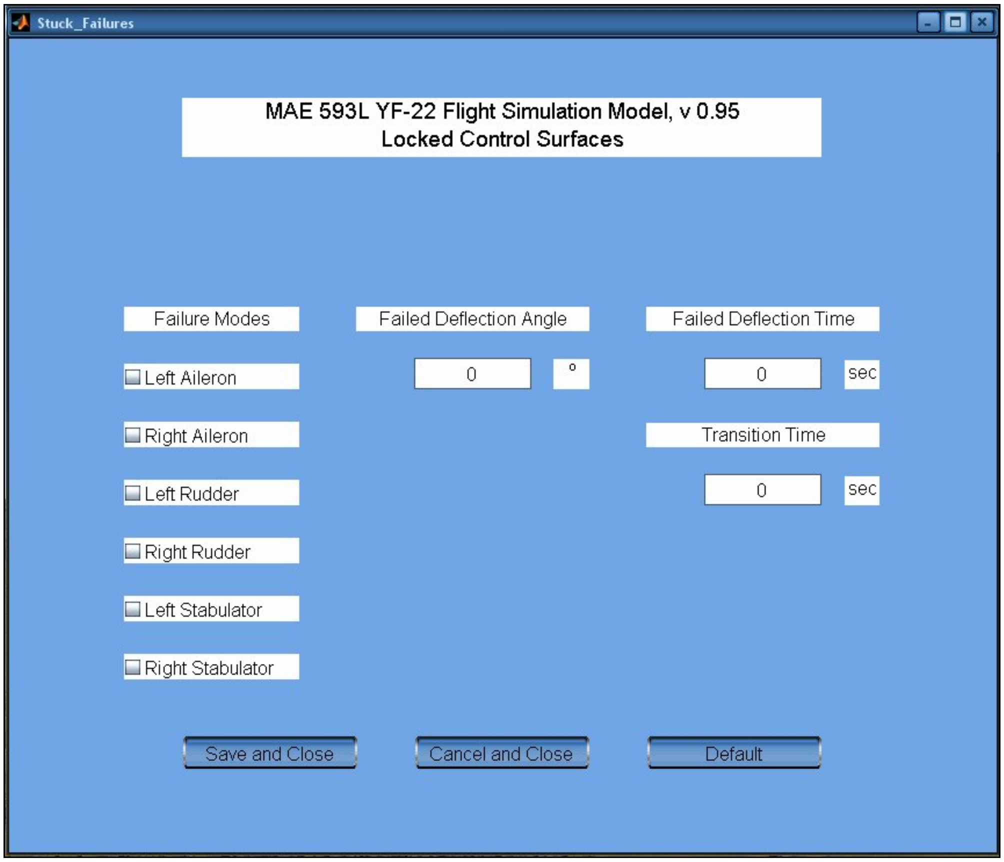


Figure 34: Failure Modes Menu

The Locked Control Surfaces window shown in Figure 35 displays the options for locking any of the six individual control surfaces at a specified time (in seconds), transition time (in seconds), and deflection angle (in degrees). The “Save and Close” button saves the configuration and closes the window. The “Cancel and Close” button ignores any changes entered in the window and closes the window. The “Default” button returns all of the parameters to default and closes the window. Checking a box activates a given failure mode and the parameters may then be selected for that control surface.



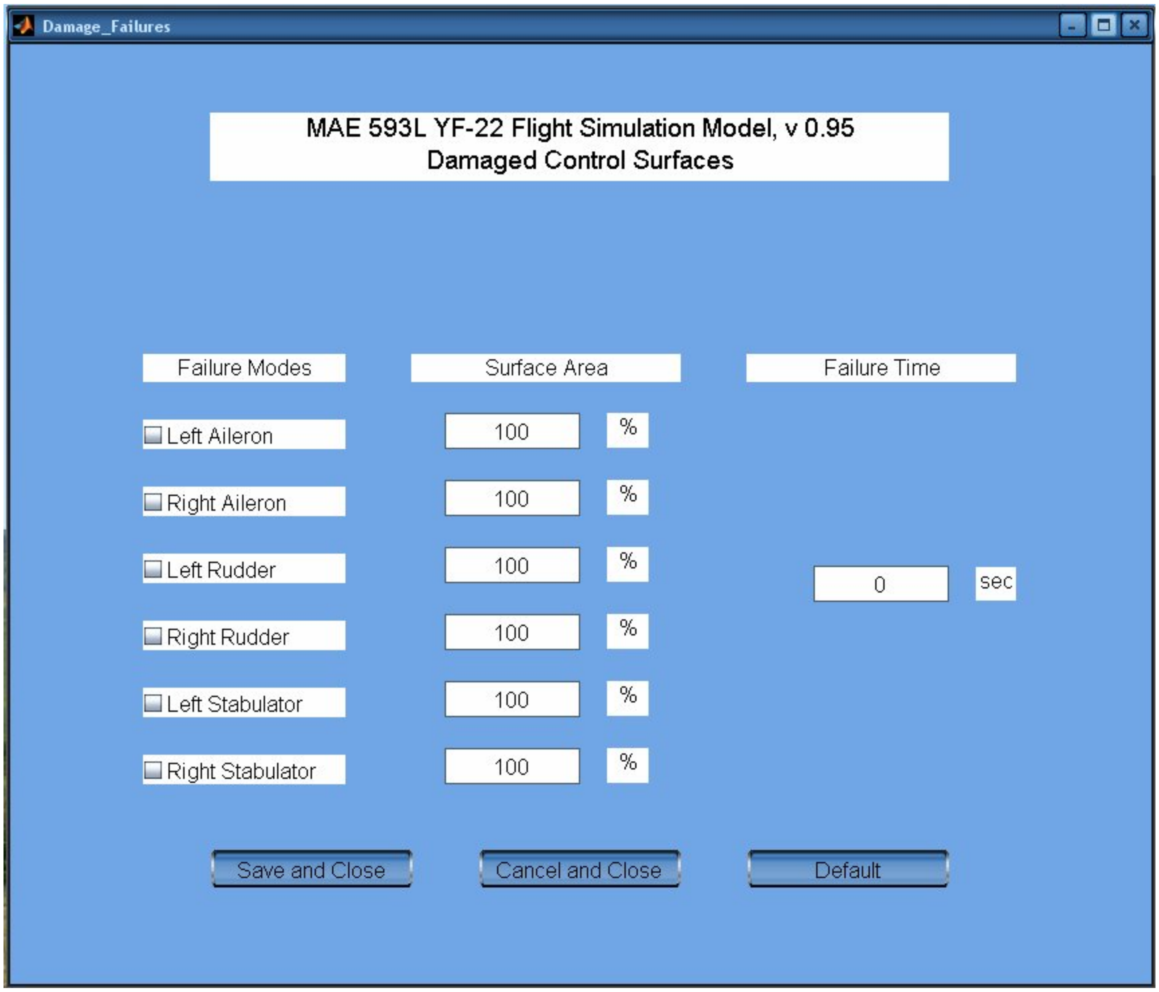
Stuck_Failures

MAE 593L YF-22 Flight Simulation Model, v 0.95
Locked Control Surfaces

Failure Modes	Failed Deflection Angle		Failed Deflection Time	
□ Left Aileron	0	°	0	sec
□ Right Aileron			Transition Time	
□ Left Rudder			0	sec
□ Right Rudder				
□ Left Stabulator				
□ Right Stabulator				

Figure 35: Locked Control Surfaces Menu

The Missing Control Surfaces window, shown in Figure 36, displays the options for removing part or all of any of the six control surfaces at a user-defined time (in seconds) and damage reduction for each control surface. The "Save and Close" button saves the configuration and closes the window. The "Cancel and Close" button ignores any changes entered in the window and closes the window. The "Default" button returns all of the parameters to default and closes the window. Checking a box activates a given failure mode.



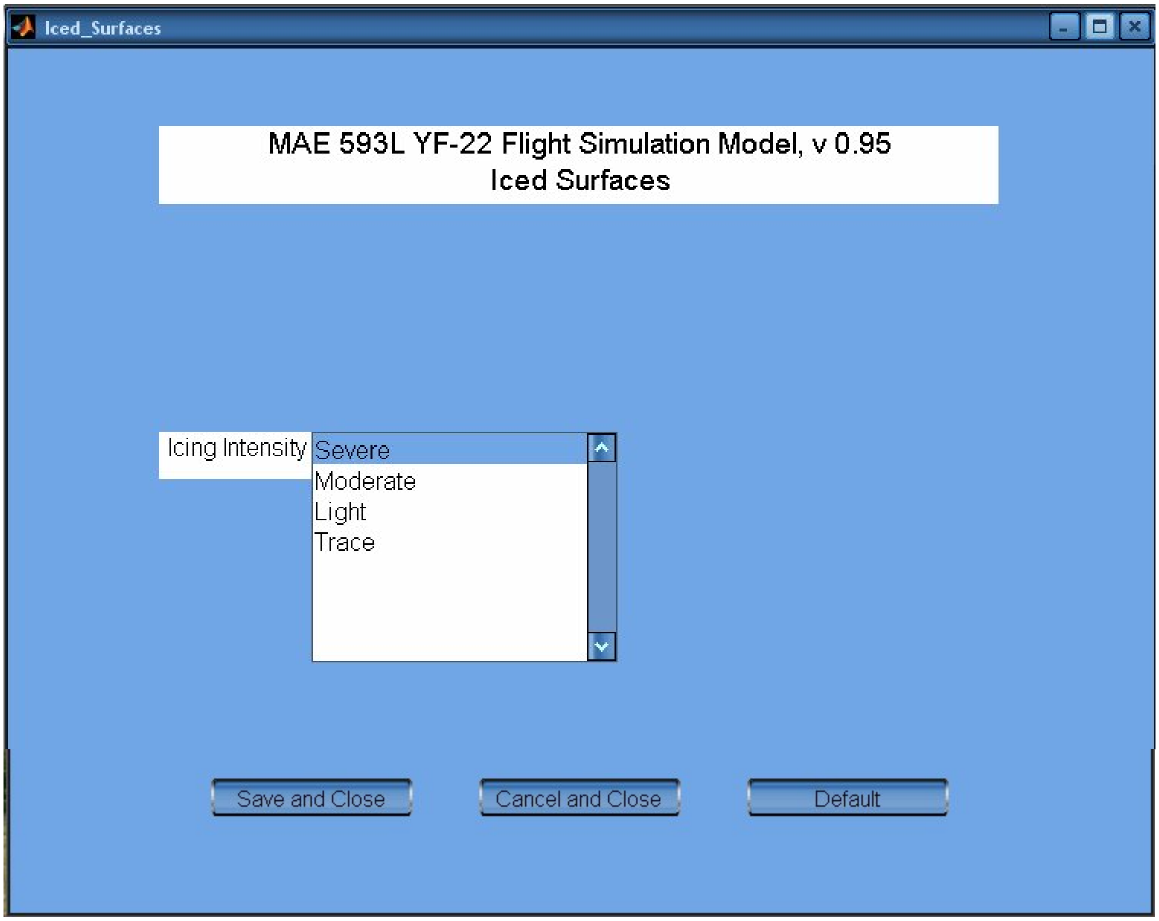
Damage_Failures

MAE 593L YF-22 Flight Simulation Model, v 0.95
Damaged Control Surfaces

Failure Modes	Surface Area	Failure Time
□ Left Aileron	100 %	0 sec
□ Right Aileron	100 %	
□ Left Rudder	100 %	
□ Right Rudder	100 %	
□ Left Stabulator	100 %	
□ Right Stabulator	100 %	

Figure 36: Missing or Damaged Control Surfaces Menu

Figure 37 displays the menu that appears after choosing the icing failure condition. The user can select any of the four icing conditions shown in the figure. Selecting one of these severity conditions will determine variables defined in the Aircraftmodel.m Matlab file and affect the relevant aerodynamic coefficients.



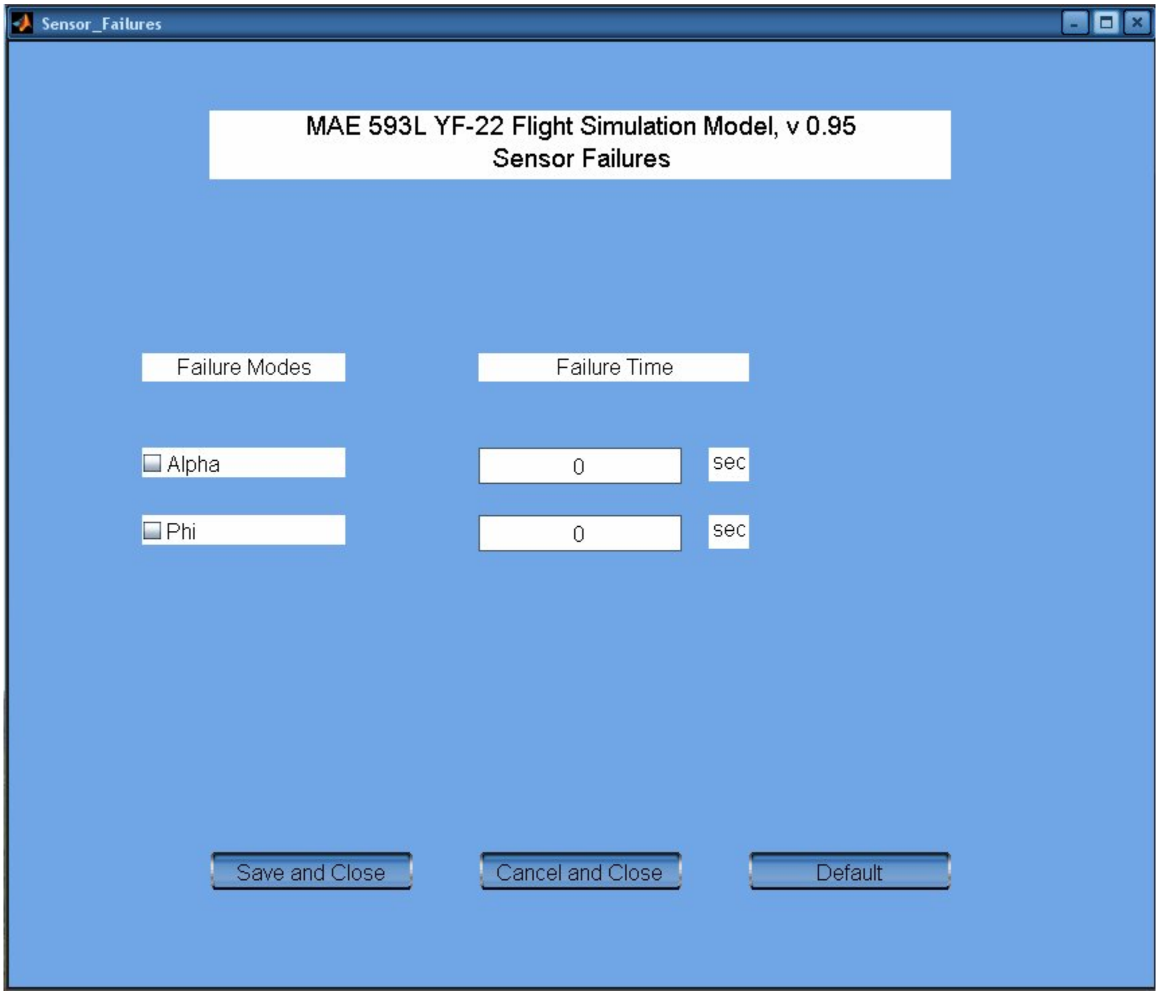
Iced_Surfaces

MAE 593L YF-22 Flight Simulation Model, v 0.95
Iced Surfaces

Icing Intensity: Severe

Figure 37: Icing Menu

The Sensor Failures window, shown in Figure 38, displays the options for damage on two of the onboard sensor models. The angle of attack vane potentiometer (alpha) and bank angle indicator (phi) sensors may be checked and a time of failure for the sensor may be specified. The "Save and Close" button saves the configuration and closes the window. The "Cancel and Close" button ignores any changes entered in the window and closes the window. The "Default" button returns all of the parameters to default and closes the window. Checking a box activates a given failure mode.


Sensor_Failures

MAE 593L YF-22 Flight Simulation Model, v 0.95
Sensor Failures

Failure Modes | Failure Time

□ Alpha 0 sec

□ Phi 0 sec

Figure 38: Sensor Failure Menu

Further, upon choosing “Begin Simulation” from the main menu, the user is greeted with the latest version of the simulation package. The user may navigate the various Simulink components and activate optional wind models and control inputs. After successful launch of the simulation, all temporary files are deleted so that failure modes don’t become attached to the model accidentally.

Appendix B: Matlab Code

AircraftModel3.m

```
%  File name:  AircraftModel3.m
%  Function:   Define the parameters for the model of WVU F22 model 
aircraft
yFLR = 0.25;

% longitudinal 2nd order model (short period model): x = [alpha q]; u = 
deltaE
Alg = [ -4.11722352855338    0.77813250614228;
       -33.88362172731505   -3.57292780929511];
Blg = [  0.54348676971364;
       -39.08466027455096];

% lateral-directional 3rd order model: x = [beta p r]; u = [deltaA 
deltaR];
Alt = [  0.429910697494514          0.09382135785621         -
1.02997260142014;
       -67.3341348830051           -7.9484571673267          
5.64021573429754;
        20.5333326004203           -0.655279367924055        -
1.99553401551354];
Blt = [   0.272449823678382         -0.771285240539336;
       -101.844564998417            33.4737614531938;
         -6.26088909938588         -24.3626805190336];

% Geometry and Mass
% GM = [c_bar       b           S        Ixx       Iyy       Izz        0 
Ixz    0          m];
  GM = [0.7649    1.9622    1.3682    1.6073    7.5085    7.1865      0 
-0.3075   0   20.6384];

  %--------Added some shizzle for the turbulence model---------
  Wingspan = GM(3);
  %wing span at 20 ft:
  W20 = 3; %m/s
  %wing direction
  Wdeg = 0;

  GM1=GM;

% trimmed condition
T0       = 97.8609;      % nominal engine thrust; N (22 lb)
V0       = 34.56;        % m/s, observed variation between +/-10m/sec 
in flight
H0       = 3000;          % m
alpha_0  = 3;            % deg
deltaE_0 = -1;           % deg

% recover the nonlinear aircraft model from the linear model
[CD,CL,CY,Cl,Cm,Cn]=ln2nl(Alg,Blg,Alt,Blt,GM,deltaE_0*pi/180,alpha_0*pi
/180,V0,H0,T0);
```

```
Ta = 1/23;               % time constant of the 1st order actuator model
KT = 0.2;                % kg/mv,        Engine thrust control gain
TE = 1;                  % sec,          Engine thrust control time
constant


% Some initial conditions
%***********************************************************************
****
uaero0 = [deltaE_0/180*pi 0 0 0];
%uaero0 = [0 0 0 0];
%uaero0=[deltaE_0/180*pi 0 0 0] 0];      % coll stab, ail, rud, flaps,
diff stab
uprop0=0.75;                             %

xinco=[V0;alpha_0/180*pi;0;0;0;0;0;alpha_0/180*pi;0;0;0;H0];
y0=[xinco;0;0;0;0;0;0;0;0;0;V0;0;0;0;0;0;0;0;0;0;0;-
1;0;0;0;deltaE_0/180*pi;deltaE_0/180*pi;0;0;0;0;0;0;0];
ro=1.225;

%***********************************************************************
****
% NLDI data
%***********************************************************************
****
ix=GM(4)/14.594/0.3048^2;
iy=GM(5)/14.594/0.3048^2;
iz=GM(6)/14.594/0.3048^2;
ixz=GM(8)/14.594/0.3048^2;

vt=V0/0.3048;   % ft/s
h=H0/0.3048;    % ft
g=32.2;

qbar=ro/14.594*0.3048/2*V0^2;
sarea=GM(3)/0.3048^2;
cmac=GM(1)/0.3048;
span=GM(2)/0.3048;

clbeta=Cl(2);
clp=Cl(3);
clr=Cl(4);
cldsn=-CL(4)/2/GM(2)*yFLR;
%cldsn=0;
cldan=Cl(5);
cldrn=Cl(6);
cldcn=0;

cnbeta=Cn(2);
cnp=Cn(3);
cnr=Cn(4);
cndsn=0;
cndan=Cn(5);
cndrn=Cn(6);
cndcn=0;
```

```
cma=Cm(2);
cmu=0;
cmq=Cm(3);
cmdsn=Cm(4);

dail_max=17;
dail_min=-17;
drud_max=9;
drud_min=-9;
dstab_max=11;
dstab_min=-11;

%***********************************************************************
****

% Compute Inertia Coefficients
Ixx=GM(4);
Iyy=GM(5);
Izz=GM(6);
Ixy=0;
Ixz=GM(8);
Iyz=0;

I=Ixx*Iyy*Izz-2*Ixy*Ixz*Iyz-Ixx*Iyz^2-Iyy*Ixz^2-Izz*Ixy^2;
I1=Iyy*Izz-Iyz^2;
I2=Ixy*Izz+Iyz*Ixz;
I3=Ixy*Iyz+Iyy*Ixz;
I4=Ixx*Izz-Ixz^2;
I5=Ixx*Iyz+Ixy*Ixz;
I6=Ixx*Iyy-Ixy^2;

Pl=I1/I;
Pm=I2/I;
Pn=I3/I;
Ppp=-(Ixz*I2-Ixy*I3)/I;
Ppq=(Ixz*I1-Iyz*I2-(Iyy-Ixx)*I3)/I;
Ppr=-(Ixy*I1+(Ixx-Izz)*I2-Iyz*I3)/I;
Pqq=(Iyz*I1-Ixy*I3)/I;
Pqr=-((Izz-Iyy)*I1-Ixy*I2+Ixz*I3)/I;
Prr=-(Iyz*I1-Ixz*I2)/I;

Ql=I2/I;
Qm=I4/I;
Qn=I5/I;
Qpp=-(Ixz*I4-Ixy*I5)/I;
Qpq=(Ixz*I2-Iyz*I4-(Iyy-Ixx)*I5)/I;
Qpr=-(Ixy*I2+(Ixx-Izz)*I4-Iyz*I5)/I;
Qqq=(Iyz*I2-Ixy*I5)/I;
Qqr=-((Izz-Iyy)*I2-Ixy*I4+Ixz*I5)/I;
Qrr=-(Iyz*I2-Ixz*I4)/I;

Rl=I3/I;
Rm=I5/I;
Rn=I6/I;
Rpp=-(Ixz*I5-Ixy*I6)/I;
Rpq=(Ixz*I3-Iyz*I5-(Iyy-Ixx)*I6)/I;
```

```
Rpr=-(Ixy*I3+(Ixx-Izz)*I5-Iyz*I6)/I;
Rqq=(Iyz*I3-Ixy*I6)/I;
Rqr=-((Izz-Iyy)*I3-Ixy*I5+Ixz*I6)/I;
Rrr=-(Iyz*I3-Ixz*I5)/I;

GM2=[Pl Pm Pn Ppp Ppq Ppr Pqq Pqr Prr
     Ql Qm Qn Qpp Qpq Qpr Qqq Qqr Qrr
     Rl Rm Rn Rpp Rpq Rpr Rqq Rqr Rrr];

  % Define stability and control derivatives of the YF-22
  % for the nonlinear aerodynamic model (Decoupled Control Surfaces)
  % ---------------------------------------------------------------
  CX0   = -CD(1);     CZ0    = -CL(1);      Cm0   =  Cm(1);
  CXa   = CD(2);      CZa    = -CL(2);      Cma   = cma;
  CXa2  = 0;          CZa3   =  0;          Cma2 = 0;
  CXa3  = 0;          CZq    = -CL(3);      Cmq   = cmq;
  CXq   = CD(3);      CZdeL  = 0.5*CL(4);   CmdeL = 0.5*cmdsn;
  CXdrL = 0.5*CD(4);  CZdeR  = 0.5*CL(4);   CmdeR = 0.5*cmdsn;
  CXdrR = 0.5*CD(4);  CZdeb2 = 0;           Cmb2 =  0;
  CXdf  = 0;          CZdf   = 0;           Cmr   =  0;
  CXadf = 0;          CZadf  = 0;           Cmdf =  0;

  CY0   = -CY(1);       Cl0    =  Cl(1);    Cn0   = Cn(1);
  CYb   = -CY(2);       Clb    = clbeta;    Cnb   =  cnbeta;
  CYp   = -CY(3);       Clp    = clp;       Cnp   = cnp;
  CYr   = -CY(4);       Clr    =  clr;      Cnr   = cnr;
  CYdaL = -0.5*CY(5);   CldaL  = 0.5*cldan; CndaL = 0.5*cndan;
  CYdaR = -0.5*CY(5);   CldaR  = 0.5*cldan; CndaR = 0.5*cndan;
  CYdrL  = -0.5*CY(6);  CldrL  = 0.5*cldrn; CndrL = 0.5*cndrn;
  CYdrR  = -0.5*CY(6);  CldrR  = 0.5*cldrn; CndrR = 0.5*cndrn;
  CYdra = 0;            Cldaa  = 0;         Cnq   =  0;
  CYbdot= 0;                                Cnb3 =  0;

  % Define the AM Matrix
AM = [ CX0     CY0     CZ0     Cl0     Cm0     Cn0  ;
       CXa     0       CZa     0       Cma     0    ;
       CXa2    0       0       0       Cma2    0    ;
       CXa3    0       CZa3    0       0       0    ;
       0       CYb     0       Clb     0       Cnb  ;
       0       0       0       0       Cmb2    0    ;
       0       0       0       0       0       Cnb3 ;
       0       CYp     0       Clp     0       Cnp  ;
       CXq     0       CZq     0       Cmq     Cnq  ;
       0       CYr     0       Clr     Cmr     Cnr  ;
       0       0       CZdeL   0       CmdeL   0    ;
       0       0       CZdeR   0       CmdeR   0    ;
       CXdf    0       CZdf    0       Cmdf    0    ;
       0       CYdaL   0       CldaL   0       CndaL;
       0       CYdaR   0       CldaR   0       CndaR;
       CXdrL   CYdrL    0      CldrL   0       CndrL;
       CXdrR   CYdrR    0      CldrR   0       CndrR;
       CXadf   0       CZadf   0       0       0    ;
       0       CYdra   0       0       0       0    ;
       0       0       0       Cldaa   0       0    ;
       0       0       CZdeb2  0       0       0    ;
       0       CYbdot  0       0       0       0    ];
```

```
AM = AM';

%eliminate the need to run the fixsate.m from FDC
xfix = 1;
```

ln2nl.m

```
function
[CD,CL,CY,Cl,Cm,Cn]=ln2nl(Alg,Blg,Alt,Blt,GM1,deltaE_0,alpha_0,V0,h0,T0)

%  File name:  ln2nl.m
%  Function:   converting linear aircraft model to nonlinear aircraft
               model
%  Version:    1.0
%  Author:     Sheng Wan
%  Date:       Feb. 12, 2003

c_bar = GM1(1);
b     = GM1(2);
S     = GM1(3);
Ixx   = GM1(4);
Iyy   = GM1(5);
Izz   = GM1(6);
Ixz   = GM1(8);
m     = GM1(10);

det_I = Ixx*Iyy*Izz-Iyy*Ixz^2;
Qm   = (Ixx*Izz - Ixz^2)/det_I;
Qpp = -Ixz*(Ixx*Izz-Ixz^2)/det_I;
Qpr = -(Ixx-Izz)*(Ixx*Izz-Ixz^2)/det_I;
Qrr = -Qpp;
Pl   = Iyy*Izz/det_I;
Pn   = Iyy*Ixz/det_I;
Ppq = (Izz-Iyy+Ixx)*Iyy*Ixz/det_I;
Pqr = -((Izz-Iyy)*Izz*Iyy+Iyy*Ixz^2)/det_I;
Rl   = Iyy*Ixz/det_I;
Rn   = Ixx*Iyy/det_I;
Rpq = ((Ixx-Iyy)*Ixx*Iyy+Iyy*Ixz^2)/det_I;
Rqr = -Ppq;

g0 = 9.80665;                 % m/s^2,     gravitational acceleration
at sea level
Tt0 = 288.15+0;               % K,         temperature
R_earth = 6371020;            % m,         radius of the Earth

g = g0*(R_earth/(R_earth+h0))^2;    %   m/s^2
Tt = Tt0-0.0065*h0;
rho0 = 101325*(Tt/Tt0)^(g/1.86584)/(287.05*Tt);
q0 = 1/2*rho0*V0^2;

Cice=1.0;
C_L_alpha  = -((m*V0*Alg(1,1)+T0*cos(alpha_0))/(q0*S))*Cice;
C_L_q      = (1-Alg(1,2))*(2*m*V0^2)/(q0*S*c_bar);
C_L_deltaE = -Blg(1)*(m*V0)/(q0*S);
C_L_0 = (m*g-T0*sin(alpha_0))/(q0*S) - C_L_alpha*alpha_0 -
      C_L_deltaE*deltaE_0;
```

```
C_m_alpha  = Alg(2,1)/(Qm*q0*S*c_bar);
C_m_q        = Alg(2,2)*(2*V0)/(Qm*q0*S*c_bar^2);
C_m_deltaE = Blg(2)/(Qm*q0*S*c_bar);
C_m_0        = -C_m_alpha*alpha_0-C_m_deltaE*deltaE_0;

C_y_beta     = Alt(1,1)*(m*V0)/(q0*S);
C_y_p        = (Alt(1,2)-sin(alpha_0))*(2*m*V0^2)/(q0*S*b);
C_y_r        = (Alt(1,3)+cos(alpha_0))*(2*m*V0^2)/(q0*S*b);
C_y_deltaA   = Blt(1,1)*(m*V0)/(q0*S);
C_y_deltaR   = Blt(1,2)*(m*V0)/(q0*S);

InvPRln = inv([Pl, Pn; Rl, Rn]);

temp     = 1/(q0*S*b)*InvPRln*[Alt(2,1); Alt(3,1)];
C_l_beta = temp(1);
C_n_beta = temp(2);
temp     = 2*V0/(q0*S*b^2)*InvPRln*[Alt(2,2); Alt(3,2)];
C_l_p = temp(1);
C_n_p = temp(2);
temp = 2*V0/(q0*S*b^2)*InvPRln*[Alt(2,3); Alt(3,3)];
C_l_r = temp(1);
C_n_r = temp(2);
temp = 1/(q0*S*b)*InvPRln*[Blt(2,1); Blt(3,1)];
C_l_deltaA = temp(1);
C_n_deltaA = temp(2);
temp = 1/(q0*S*b)*InvPRln*[Blt(2,2); Blt(3,2)];
C_l_deltaR = temp(1);
C_n_deltaR = temp(2);

C_D_0 = T0*cos(alpha_0)/(q0*S);

Cl = [0   C_l_beta  C_l_p  C_l_r  C_l_deltaA  C_l_deltaR];
Cm = [C_m_0  C_m_alpha  C_m_q  C_m_deltaE 0];
Cn = [0   C_n_beta  C_n_p  C_n_r  C_n_deltaA  C_n_deltaR];
CD = [C_D_0  0  0  0  0];
CL = [C_L_0  C_L_alpha  C_L_q  C_L_deltaE 0.0];
CY = [0   C_y_beta  C_y_p  C_y_r  C_y_deltaA  C_y_deltaR];
```

Appendix C: Raw Data

The following is the raw data collected for each icing condition; severe, medium, light, and trace. Alpha and C_Z values are achieved at steady state conditions

Severe icing condition = 23% CLalpha

alpha = 1.9	Cz = -.2586
alpha = 2.35	Cz = -.2532
alpha = 0.1	Cz = -.293
alpha = 3.1	Cz = -.268
1.1	-.274

Medium, 50%

2.32	-.2377
4.7	-.3375
5.92	-.386
7.1	-.4345
10.37	-.587
11.9	-.632

Light 75%

3.38	-.287
6.5	-.47
10.1	-.675
13.1	-.848
13.9	-.89

Trace 95%

2.3	-.218
4.9	-.41
9.35	-.717
12.1	-.92

Clean 100%

2.41	-.224
4.8	-.405
10.3	-.78
12.9	-.98

Appendix C: Team Member Tasks

Jason Gross:

- Locked Control Surface Failure Model
- Missing or Damaged Control Surface Failure Model
- Joystick Implementation
- Joystick Scaling
- Modal Analysis
- Human Pilot Model (not implemented in current simulation)
- Final Report Writing
- Final Presentation Layout

Nick Hansford:

- GUI Design
- GUI Implementation
- Overall Model Layout
- Wind Model
- Modal Analysis
- Joystick Implementation
- Final Report Writing
- Final Presentation Layout

Kerri Phillips:

- Decoupled Control Surface Model
- Virtual Reality Toolbox Integration
- Joystick Implementation
- Sensor Failure Model
- Engine Model
- Final Report Writing
- Final Presentation Layout
- Weekly Report Writing

Blake Waldie:

- Joystick Implementation
- Sensor Failure Model
- Icing Model
- Modal Analysis
- Human Pilot Model (not implemented in current simulation)
- Final Report Writing
- Final Presentation Layout