

Supersonic Fighter Aircraft Design



In this example we will design the control system and analyze stability and performance of a supersonic fighter aircraft at a flight condition of: 20,000 (feet) altitude, level flight, at Mach: 1.4. The aircraft uses three aero surfaces in combination with two thrust vectoring engines for flight control. The aerosurfaces are: two elevons (left and right) mainly for pitch and roll control, and a vertical rudder for yaw control. The two engines are gimbaling only in pitch to enhance pitch controllability and they can also vary their thrust for speed control. The nominal thrust of each engine in this flight condition is 30,000 (lb), but they can throttle above and below this thrust, from a minimum of 10,000 (lb) to a maximum of 50,000 (lb) of thrust. The aircraft uses a vane sensor for measuring speed, and the angles of attack and sideslip. Gyros are used for measuring body rates, and an IMU for attitude. Structural flexibility is also included in the finite element dynamic model by using inertial flex coupling coefficients that couple the effector motion with flexibility.

The control objective in this flight condition in the longitudinal directions is to independently command and execute changes in altitude and velocity and to control the roll attitude in the lateral directions. The altitude and velocity are controlled with a combination of pitching and throttling the engines. The flight direction is indirectly controlled by rolling. A multivariable state-feedback control system will be designed using the LQR method. It consists of three loops that control: (a) altitude, (b) velocity, and (c) roll attitude. The control system must also be able to tolerate a certain amount of wind gust disturbances. A mixing logic matrix is also designed in order to properly

combine the engine and aero-surface deflections and to steer the vehicle in the directions demanded by the flight control system. The analysis is separated in two sections: (a) rigid-body modeling and control design, and (b) flexible vehicle modeling and control analysis. The rigid-body design and analysis is performed in directory "...\Examples\Fighter Aircraft\Rigid Body Design" and the Matlab analysis in subdirectory "...\Fighter Aircraft\Rigid Body Design\Mat_Rigid". The flexible vehicle modeling, filter design, and control analysis is performed in directory "...\Examples\Fighter Aircraft\Flex Analysis" and the Matlab analysis is in subdirectory "...\Flex Analysis\Mat_Flex".

1.0 Rigid-Body Design/ Analysis

For rigid-body design and analysis we will create two rigid vehicle state-space models: a simple design model and a more complex model for simulations and control analysis. The design model will be used to synthesize the LQR control laws. The simulation model will be combined with the actuators, mixing-logic, and the control laws and it will be used in Matlab simulations and frequency domain analysis to analyze the stability and performance of the aircraft.

1.1 Input Data for the Rigid Aircraft Models

The aircraft data for generating the two rigid-body models are in file "*Fighter_RB.Inp*", which is located in folder "*Examples\Fighter Aircraft\Rigid Body Design*". Their titles are: "*Fighter Aircraft Design Model*" and "*Fighter Aircraft Rigid Body Model*". The flag lines below the titles indicate that the rate gyro measurements are in body axes (not stability axes) and the attitude measurements are Euler angles (not integrals of body rates). The dataset for the design model is simple because it does not include any additional rate gyros and vane sensors but uses only the default outputs. It also has the tail-wags-dog options turned off (NO TWD) and, therefore, it does not include gimbal acceleration inputs and hinge moment outputs, which are required to couple the vehicle with the detailed actuator models, as we shall see in the second model. The aircraft input datasets include three aero-surfaces (left elevon, right elevon, and vertical rudder), and also two engines.

The two engines are defined to be gimbaling in a single direction (not two) and they are also throttling "*Single-Gimbal, Throttling*". The gimbaling direction for the two engines is defined to be in pitch, because the gimbaling direction is defined in the data by the maximum pitch and yaw deflections, which are 18° in pitch and 0° in yaw. The engine is constrained to rotate in the direction defined by the maximum angles which is purely pitch in this case. If the single gimbal directions of the engines were tilted, for example, let's say $+45^\circ$ from the vertical for the left engine, and -45° from vertical for the right engine, the max deflections definition for the left engine should have been $+12.7^\circ$ in pitch and $+12.7^\circ$ in yaw, and for the right engine it should be $+12.7^\circ$ in pitch and -12.7° in yaw. This definition would imply a total rotation of 18° about the skewed gimbaling axis. The two engines have a nominal thrust of 30,000 (lb) each, and their maximum thrust is 50,000 (lb). This implies that their thrust can be varied $\pm 20,000$ (lb) from nominal, using a throttle control valve.

The second flight vehicle dataset in the input file "*Fighter Aircraft Rigid Body Model*" generates a vehicle model that will be used for rigid-body simulations. Both, engines and control surfaces are defined to include the tail-wag-dog and the load-torque feedback dynamics by setting the flag above the effector data to (WITH TWD). This option generates additional inputs and outputs in the vehicle model that can be used to couple it with the actuator models, three aero-surface actuators and two engine TVC actuators. This aircraft dataset also includes three rate gyros that measure roll, pitch and yaw rotations, and a vane sensor that measures the angles of attack and sideslip and it is located

in the front of the aircraft. The wind gust input is used to excite the aircraft with a typical 30 (feet/sec) wind gust in order to analyze the flight control system response. The direction of the wind-gust is defined in the data by the incidence angles. It is perpendicular to the vehicle x axis (90° elevation) coming towards the vehicle, and at 45° from the -z axis, that is, between +z and +y body axes. After processing the datasets, the two aircraft state-space systems are saved in systems file “*Fighter_RB.Qdr*”. Their titles are the same as the titles in the input data.

The input data file “*Fighter_RB.Inp*” includes additional input datasets that are used by other Flixan utility programs to prepare additional analysis state-space models. There is system combination dataset “*Design Model*” that combines the “*Fighter Aircraft Design Model*” with the mixing-logic matrix *Kmix4* to reduce the inputs of the design model. The mixing logic matrix *Kmix4* converts the four flight control demands to aerosurface and engine deflections, and to throttle commands. It is located in the systems file “*Fighter_RB.Qdr*”. The combined system “*Design Model*” is simpler for design purposes because it has only 4 inputs, (3 angular acceleration commands: p-dot, q-dot, r-dot, and one translational x-acceleration command. The next two datasets in file “*Fighter_RB.Inp*” are system modification datasets. They are processed by the system modification Flixan utility which separates the “*Design Model*” system into pitch and lateral subsystems, “*Pitch Design Model*” and “*Lateral Design Model*” respectively. These state-space systems are used in Matlab to design two separate pitch and lateral state-feedback controllers, matrices *Kgp* and *Kgl*. The next set of input data is also a system modification dataset. Its title is “*Fighter Aircraft Rigid Body Simulation Model*”. It is used to modify the system “*Fighter Aircraft Rigid Body Model*” by creating one additional output #22, change in velocity (δV) which is needed in the simulation model for state-feedback. The input data file also contains datasets for two simple actuator models, an actuator model “*Elevator Actuator*” for all three aerosurfaces, and an actuator model “*Engine TVC Actuator*” that is used to gimbal the TVC engines in pitch. The two actuator models are also saved in the systems file “*Fighter-RB.Qdr*”.

Note that the input data file “*Fighter_RB.Inp*” includes also a batch set “*Batch for preparing models for the fighter aircraft*” located at the top of the file. This dataset can be used to process the entire file fast, in batch mode, via the systems file manager utility. However, in this tutorial example we will demonstrate the processing of each dataset by the interactive Flixan utilities.

1.2 Mixing Logic Matrix Design

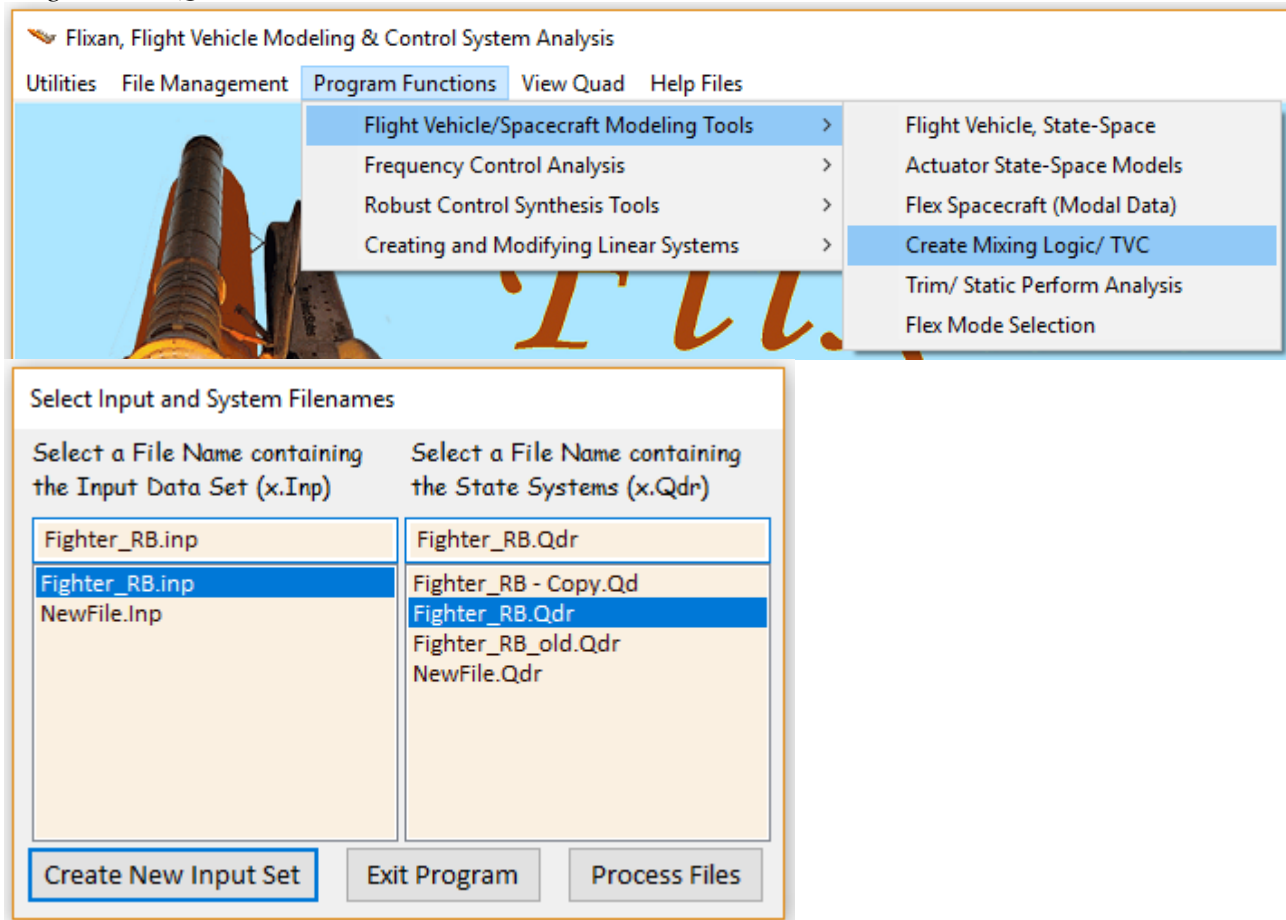
The mixing logic matrix converts the acceleration demands from the flight control system to effector commands that drive the actuators. That is, 3 aerosurface deflections, 2 pitch engine deflections, and 2 engine throttles. The matrix is designed to efficiently combine the deflection and throttle commands and to optimize the aircraft steering in four directions, that is, three rotational accelerations in roll, pitch, and yaw, and the x-axis acceleration. In essence, the mixing logic pretty much attempts to decouple the open-loop plant model into four directions that will eventually be controlled by four control loops, although in this aircraft yaw attitude is not controlled independently from roll. The Flixan program includes an algorithm for calculating the mixing logic from the aircraft data. It takes into consideration the maximum control capability of each effector and it spreads the control effort evenly among the 7 vehicle effectors so that one effector does not reach saturation before others do. The Flixan algorithm processes the mixing logic dataset “*Mixing Logic for the Fighter Aircraft (original)*” from file “*Fighter_RB.Inp*” to generate the matrix *Kmix3*. This matrix is slightly modified by the user to *Kmix4* and its title is “*Mixing Logic for Fighter Aircraft Design Model*”. The original and the modified mixing logic matrices *Kmix3* and *Kmix4* are

both saved in the output data file “*Fighter_RB.Qdr*”. *Kmix4* is also saved and in Matlab format in order to be used in the analysis.

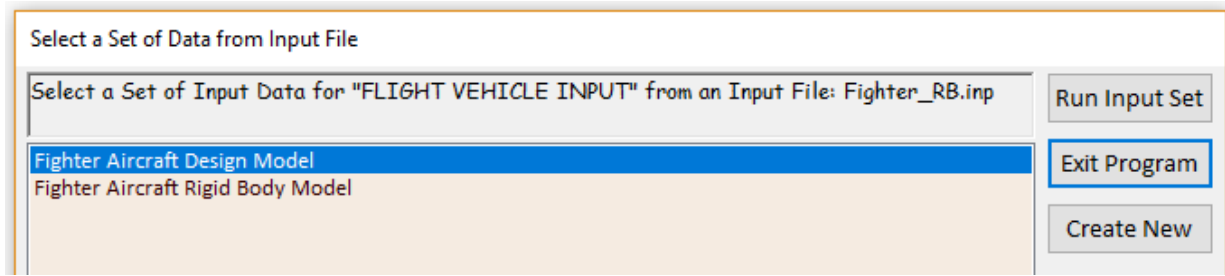
By post-multiplying the vehicle model with the mixing logic matrix we are essentially replacing the deflection command inputs and engine throttling inputs with the four acceleration command inputs in the new vehicle model. That is, three rotational acceleration inputs (roll, pitch, yaw), and one translational acceleration input along x. Note that the outputs (3 and 4) in matrix *Kmix4* and the vehicle model inputs are normalized by the engine thrust, so the variables are not engine thrusts but throttles that may vary between (-1 and +1). That is, a zero throttle output from *Kmix4* corresponds to applying nominal thrust of 30,000 (lb) at the corresponding engine (no excitation). A throttle output (+1) corresponds to a maximum thrust increase which is 50,000 (lb), and a throttle command output (-1) from the matrix corresponds to the minimum thrust which is 10,000 (lb).

Running the Mixing Logic Design Program

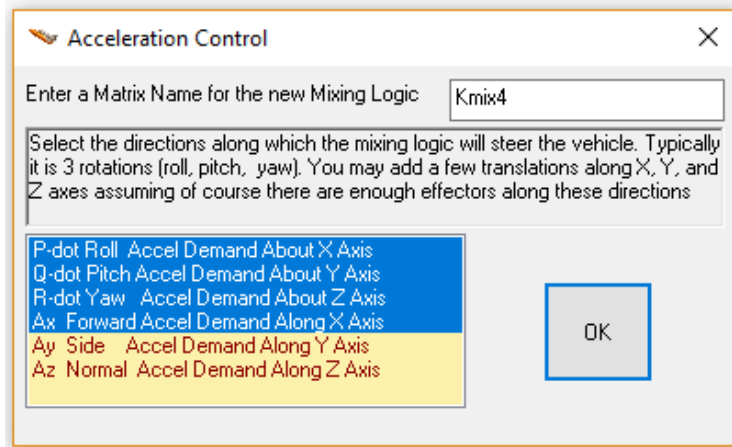
The mixing logic design program can be executed by either processing an already created mixing logic dataset via the systems file manager utility or interactively from the Flixan menu as we shall see. Start the Flixan program and go to directory “...*\Examples\Fighter Aircraft\ Rigid Body Design*”. From the Flixan main menu select “*Program Functions*”, “*Flight Vehicle/ Spacecraft Modeling Tools*”, and then “*Create Mixing Logic/ TVC*”, as shown below. From the filename selection menu select the input data file “*Fighter_RB.Inp*”, the output systems file “*Fighter_RB.Qdr*”, and click on “*Process Files*” as shown.



From the following flight vehicle data selection menu select the aircraft data title “*Fighter Aircraft Design Model*” and click on “*Run Input Set*”. The mixing matrix will be calculated based on this vehicle data.



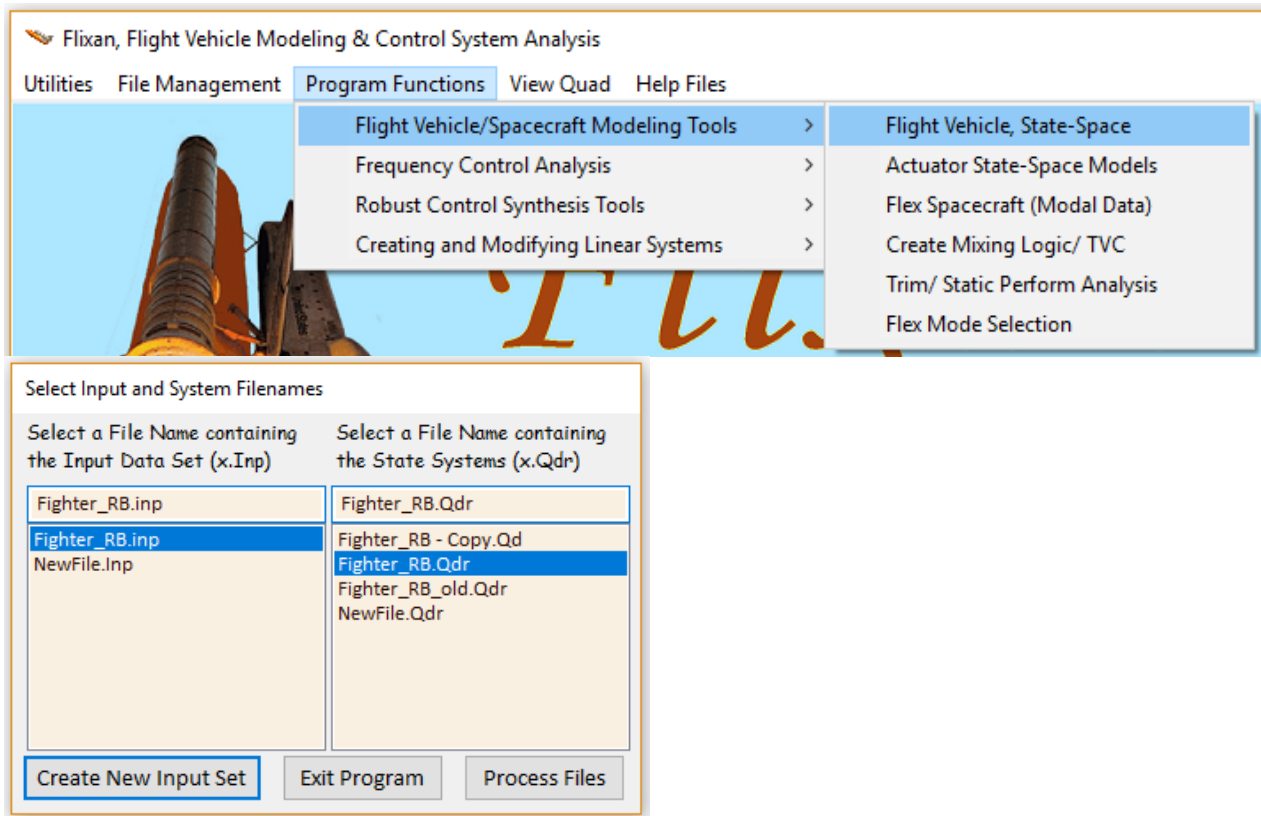
The next menu is for choosing the directions that the mixing logic matrix is expected to provide independent acceleration control. Note that the vehicle must have the proper effectors that can provide control in the directions selected. In this case we select four directions, three rotational accelerations for roll, pitch, yaw, and one translational acceleration along the x axis. Enter also a name for the effector mixing logic matrix “*Kmix4*”, and click “*OK*”. The program reads the vehicle and effector data from the input data file and calculates the (7x4) mixing logic matrix K_{mix4} .



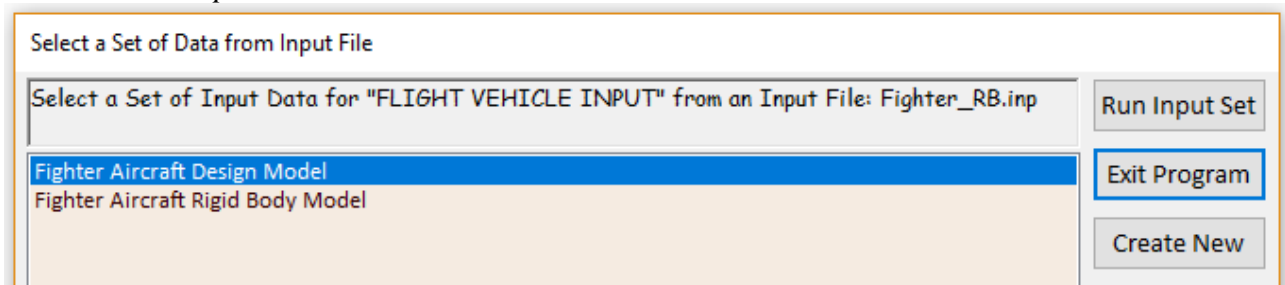
The matrix is saved by the program in the systems file “*Fighter_RB.Qdr*”. Its title is: “*Mixing Logic for the Fighter Aircraft (original)*”. It was slightly modified by the designer and saved in the same systems file under a different title “*Mixing Logic for Fighter Aircraft Design Model*”. The matrix K_{mix4} is also exported into Matlab and used in simulations. The vehicle plant will be post-multiplied by matrix K_{mix4} and used in the control design.

1.3 Flight Vehicle Modeling

Our next step is to run the flight vehicle modeling program interactively in order to create state-space systems for the two aircraft models that were described earlier. Their input data are in file “*Fighter_RB.Inp*”, and their titles are “*Fighter Aircraft Design Model*” and “*Fighter Aircraft Rigid Body Model*”, as already described. Start the Flixan program and select the same folder “...*Fighter Aircraft\Rigid Body Design*”. From the Flixan main menu select “*Program Functions*”, “*Flight Vehicle/Spacecraft Modeling Tools*”, and select “*Flight Vehicle State-Space*”. From the filename selection menu select the input data file “*Fighter_RB.Inp*”, the output systems file “*Fighter_RB.Qdr*”, and click on “*Process Files*”, as before.



From the flight vehicle data selection menu select the first title “*Fighter Aircraft Design Model*” and click on “*Run Input Set*”.



The following dialog is used for browsing and processing the flight vehicle data. It includes tabs that show various vehicle parameters. You may click on the tabs to view the vehicle data in groups.

Flight Vehicle Parameters

Vehicle System Title
Fighter Aircraft Design Model

Number of Vehicle Effectors
Gimbaling Engines or Jets. Include Tail-Wags-Dog? 2 [WITH TWD] [WITHOUT TWD]
Rotating Control Surfaces. Include Tail-Wags-Dog? 3 [WITH TWD] [WITHOUT TWD]
Reaction Wheels? 0

Momentum Control Devices
Single Gimbal CMGs? 0 [Yes] [No]
Include a 3-axes Stabilized Double Gimbal CMG System? [Yes] [No]

Number of Sensors
Gyros 0
Acceleromet 0
Aero Vanes 0
External Torques 0

Modeling Options (Flags)
Output Rates in: [Body Axes] [Stability Axes]
Turn Coordination?: [Include Turn Coordin] [Without Turn Coordin]
Aero-Elasticity Options: [Include GAFD, H-param Flex Coupl. data only] [Neither Gafd nor Hpar]
Attitude Angles: [Euler Angles] [Integrals of Rates] [LVLH Attitude]

Number of Modes
Structure Bending 0
Fuel Sloshing: 0

Reaction Wheels | Single Gimbal CMGs | Double Gimbal CMG System | Slewing Appendages | Gyros | Accelerometer | Aero Sensors | Fuel Slosh | Flex Modes | User Notes
Mass Properties | Trajectory Data | Gust/ Aero Paramet. | Aero Force Coeffs | Aero Moment Coeffs | Control Surfaces | Gimbal Engines/ RCS | External Torques

This Vehicle has 3 Control Surfaces | **Control Surface No: 1 Left Elevator** | Surface Definition | Next Surface

Surface Rotation Angles
Surface Trim Position (deg) 0.000000
Largest Positive Deflection from Trim (deg) 22.000000
Largest Negative Deflection from Trim (deg) -22.000000

Surface Location (ft) and Hinge Orientation Angles (deg)
Xcs 5.200000
Ycs -7.400000
Zcs -2.200000
Phi_cs 1.500000
Lambda_cs 0.000000

Control Surface Mass Properties
Control Surface Mass in (Slugs) 43.000000
Moment of Inertia about Hinge (slug-ft²) 125.000000
Moment Arm (feet), Surface CG to Hinge 0.70000000
Control Surface Chord (feet) 2.50000000
Control Surface Reference Area (ft²) 16.24000000

Aero Force Derivatives
Ca_delta 0.3000000E-03 Ca_delta_dot 0.000000
Cy_delta -0.1800000E-02 Cy_delta_dot 0.000000
Cz_delta -0.7000000E-02 Cz_delta_dot 0.000000
Control Force Derivatives due to Surface Deflection (1/deg) and Control Force Derivatives due to Surface Rate (1/deg/sec)

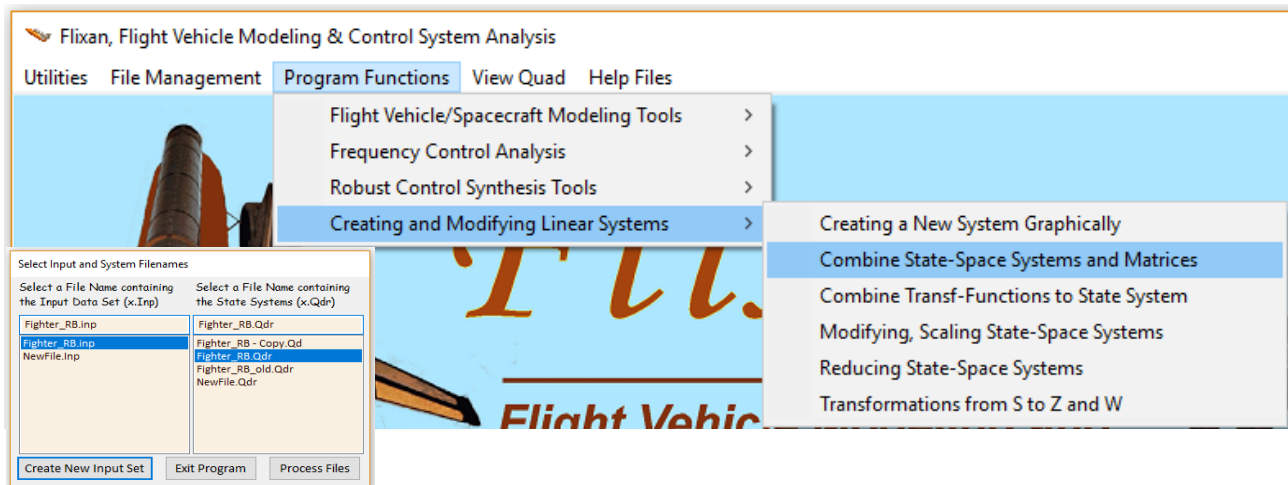
Aero Moment Derivatives
Cl_delta 0.2400000E-02 Cl_delta_dot 0.000000
Cm_delta -0.1300000E-02 Cm_delta_dot 0.000000
Cn_delta 0.8000000E-03 Cn_delta_dot 0.000000
Control Moment Derivatives due to Surface Deflection (1/deg) and Control Moment Derivatives due to Surface Rate (1/deg/sec)

Hinge Moment Derivatives
Chm_Alpha -0.5000000E-02
Chm_Beta -0.3000000E-03
Chm_Delta -0.9700000E-02
Chm_Mach 0.000000
Hinge Moment Derivatives with respect to Changes in: Alpha, Beta, Surf Deflection (1/deg) and changes in Mach Number

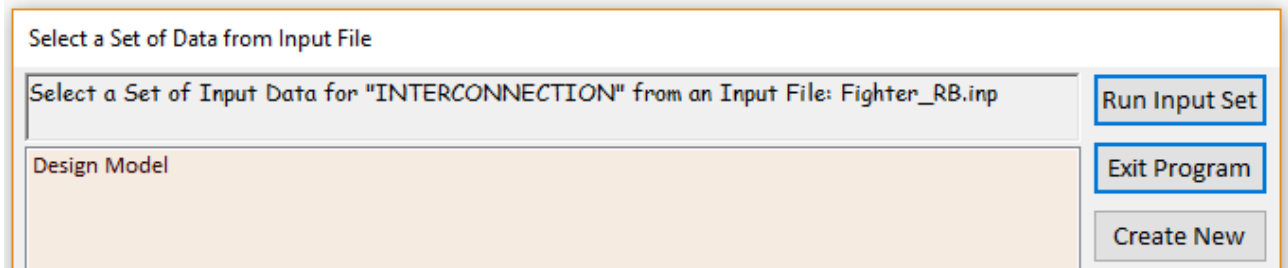
Click on “Run” button and the program will process the aircraft data and create its state-space system in file “*Fighter_RB.Qdr*” under the title “*Fighter Aircraft Design Model*”. Similarly, you may run the vehicle modeling program for a second time to process the second flight vehicle dataset “*Fighter Aircraft Rigid Body Model*” to create another state-space system that will be used for time domain simulations.

1.4 Modifying the State-Space Systems

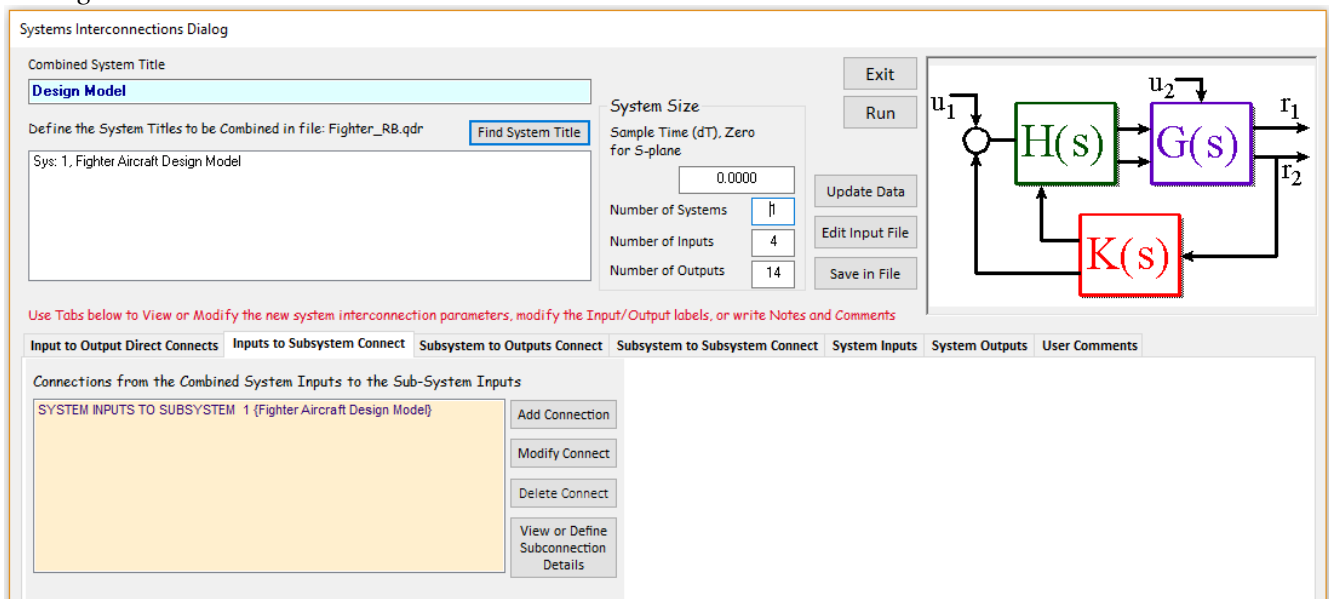
The previously created aircraft systems need some modifications before they can be used for control design and analysis. The modifications will be implemented using the Flixan systems interconnection and systems modification utilities. The first step is to use the systems combination program to post-multiply the vehicle model “*Fighter Aircraft Design Model*” by the (7x4) mixing logic matrix K_{mix4} . This will reduce and simplify the system inputs from individual effector deflections and throttle commands to 4 vehicle acceleration demands. The outputs of this new system will be the same. The interconnections dataset is in file “*Fighter_RB.Inp*” and its title is “*Design Model*”. To run the systems interconnection program, go to Flixan main menu and select “*Program Functions*”, “*Creating and Modifying Linear Systems*”, and then “*Combine State-Space Systems and Matrices*”, as shown.



From the filenames selection menu select the same input and system filenames, as before, and from the following menu select the title of the only one interconnection dataset “*Design Model*” and click on “*Run Input Set*”.



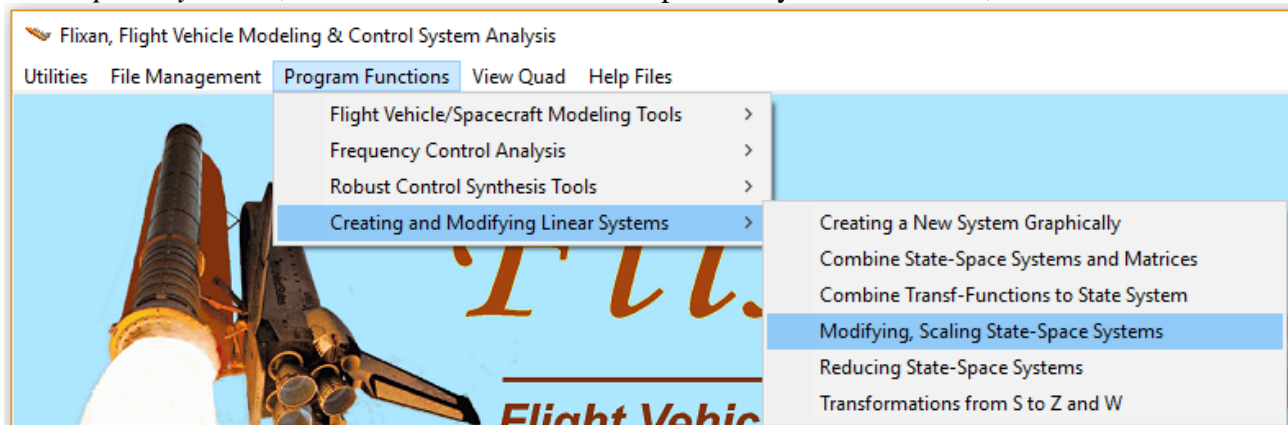
The following dialog shows the interconnection data. Click on “Run” to process the interconnection dataset and create the modified design model that will be saved in the systems file under the title “*Design Model*”.



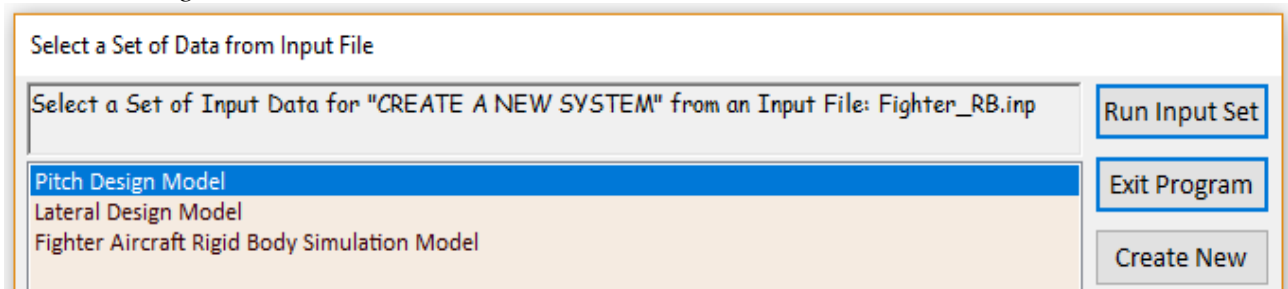
Decoupling the Design Model into Pitch and Lateral Subsystems

The new system “*Design Model*” contains pitch and lateral dynamics coupled together. In this aircraft, however, the longitudinal and lateral dynamics are not dynamically coupled, and for control design purposes it makes sense to design the longitudinal and lateral LQR controllers separately. We, therefore, need to extract from the above design model two separate subsystems, a “*Pitch Design Model*” and a “*Lateral Design Model*”. The pitch design model includes only of the longitudinal rigid-body states, inputs, and outputs, and the lateral design model includes only of the lateral rigid-body states, inputs, and outputs. The extraction of the two subsystems from the “*Design Model*” is accomplished using the systems modification program. This utility program reads the system modification instructions from datasets located in the input data file “*Fighter_RB.Inp*”. The instructions describe which of the original system variables are to be retained in the reduced system. The titles of the two system modification datasets are: “*Pitch Design Model*” and “*Lateral Design Model*”.

To process one of the system modification datasets, go to the Flixan main menu and select “*Program Functions*”, “*Creating and Modifying Linear Systems*”, and then “*Modifying, Scaling State-Space Systems*”, as shown. Select the same input and system filenames, as before.



From the following menu select the title of the pitch system extraction dataset “*Pitch Design Model*” and click on “*Run Input Set*” to process it. The extracted pitch subsystem is saved in file “*Fighter_RB.Qdr*” under the same title. A name-list of the reduced inputs, states, and outputs is also included for reference below the subsystem matrices. The same process is repeated to extract the lateral design model and the reduced lateral system is also saved in the systems file under the title “*Lateral Design Model*”.



Modifying the Simulation Model

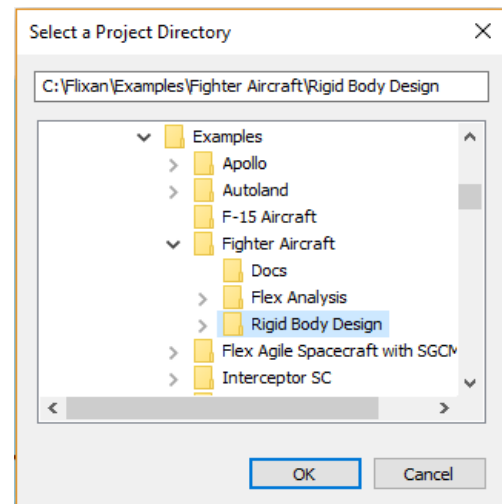
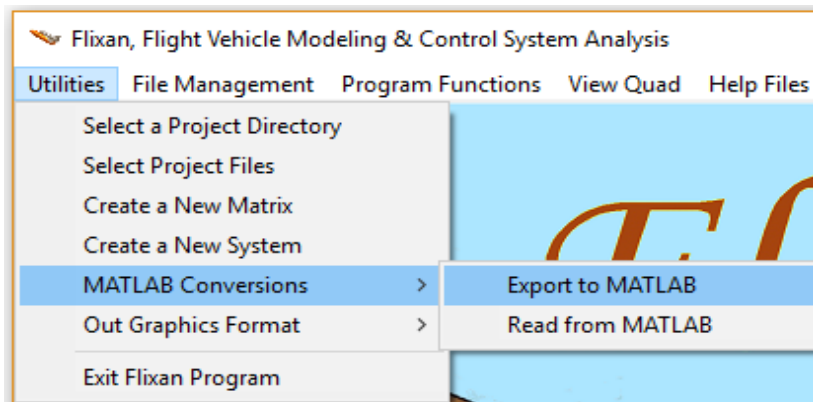
The second flight vehicle system “*Fighter Aircraft Rigid Body Model*” requires an additional modification in order to be useful for simulations. This system is missing an important variable from its output vector, the variation in aircraft velocity (δV) in (feet/sec). This is relative to the nominal speed (V_0). This variable, however, exists in the state vector, and we must modify the original system to include (δV) in the output variables. A simple solution is to create a new system from the original with a new set of outputs, extract (δV) which is the 10th state variable and include it at the bottom of the new output vector (output #22). This is implemented using the systems modification program for the third time. It reads the systems modification dataset “*Fighter Aircraft Rigid Body Simulation Model*” from the input data file, processes the system modification data which essentially modifies the original system “*Fighter Aircraft Rigid Body Model*” as instructed, and saves the new system with the additional velocity variation output in file “*Fighter_RB.Qdr*” using the same title: “*Fighter Aircraft Rigid Body Model*”.

1.5 Exporting the Data into Matlab

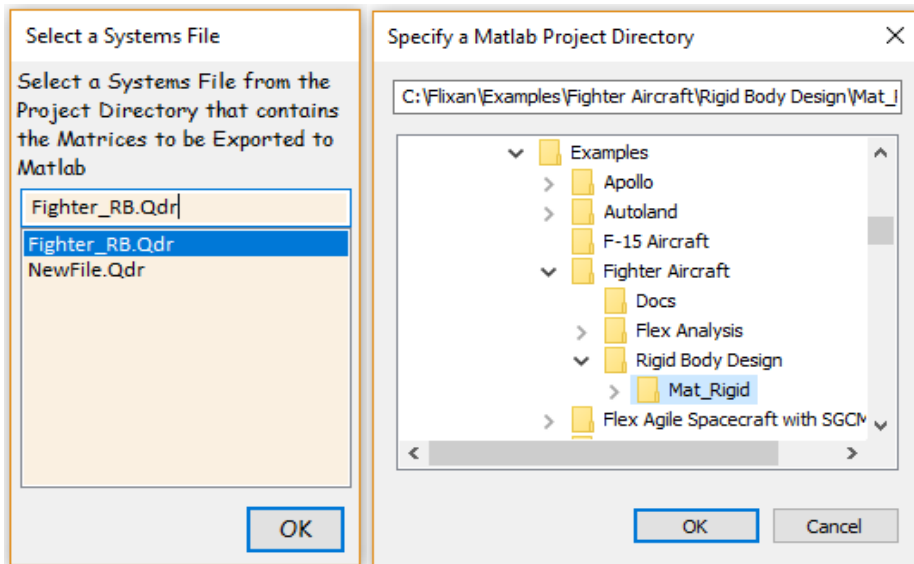
The Matlab program will be used to perform flight control design and analysis. We must therefore convert the mixing logic matrix K_{mix4} , the aircraft and actuator systems to Matlab format and save them in subdirectory “...*\Examples \Fighter Aircraft\Rigid Body Design\Mat_Rigid*”. They will be loaded into Matlab and used in the control analysis together with the other files. Flixan includes an “Export to Matlab” utility that performs this conversion.

Exporting a Matrix from the Systems File

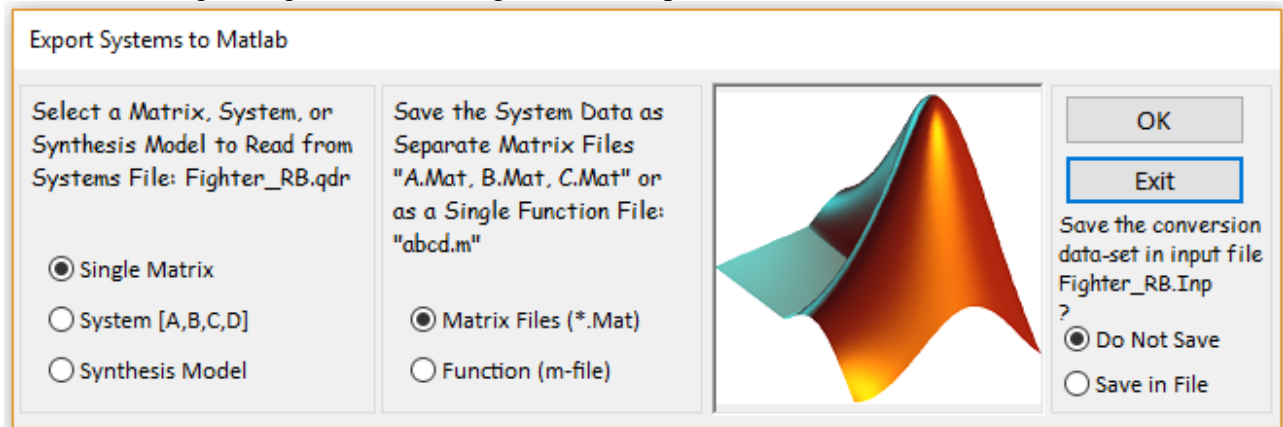
To convert matrix K_{mix4} using this utility you must first select the directory “*\Fighter Aircraft\Rigid Body Design*” where the systems file “*Fighter_RB.Qdr*” is located and from the Flixan main menu you must select the “Export to Matlab” utility, as shown below.



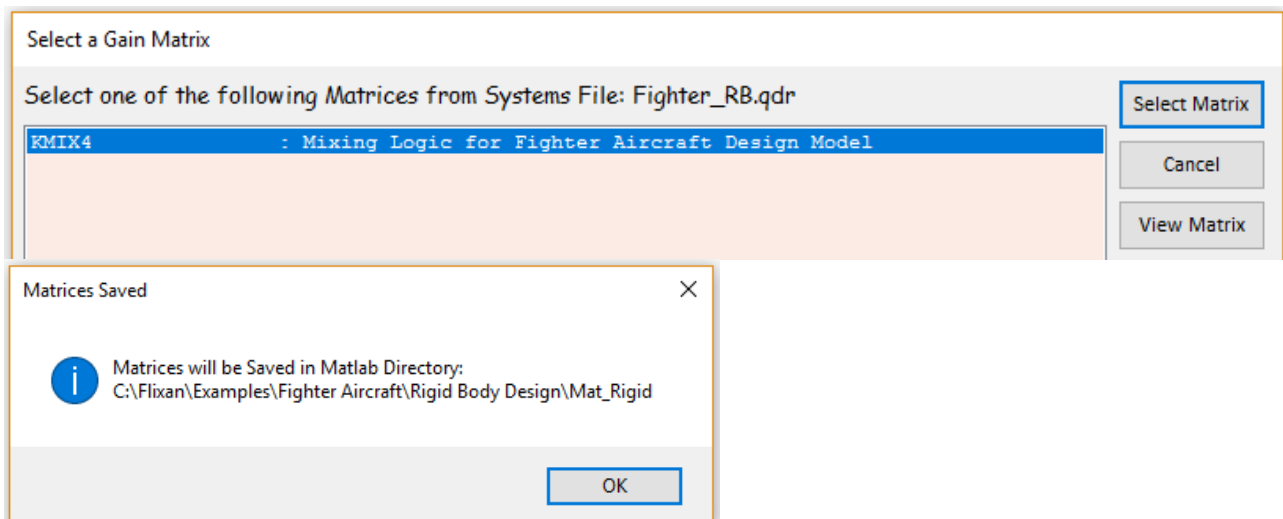
From the systems filename selection menu select the file “*Fighter_RB.Qdr*” where the matrix is located, and from the following directory selection menu select the destination folder “...*\Examples \Fighter Aircraft\ Rigid Body Design\ Mat_Rigid*” where the Matlab files are located and the matrix K_{mix4} will be saved.



In the following dialog select the “Single Matrix” option and the (*.Mat) format, and click “OK”.

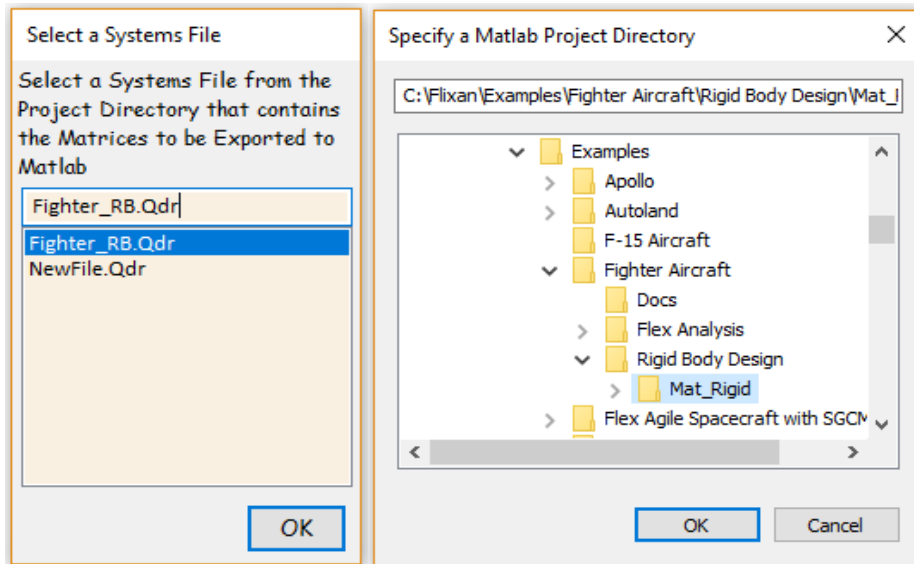


The following matrix selection menu is for select the mixing logic matrix Kmix4 from the systems file. In this case there is only one matrix to select. Click on “*Select Matrix*”, and the mixing logic matrix will be saved in file “KMIX4.MAT” in the destination directory from where it can be loaded into Matlab.

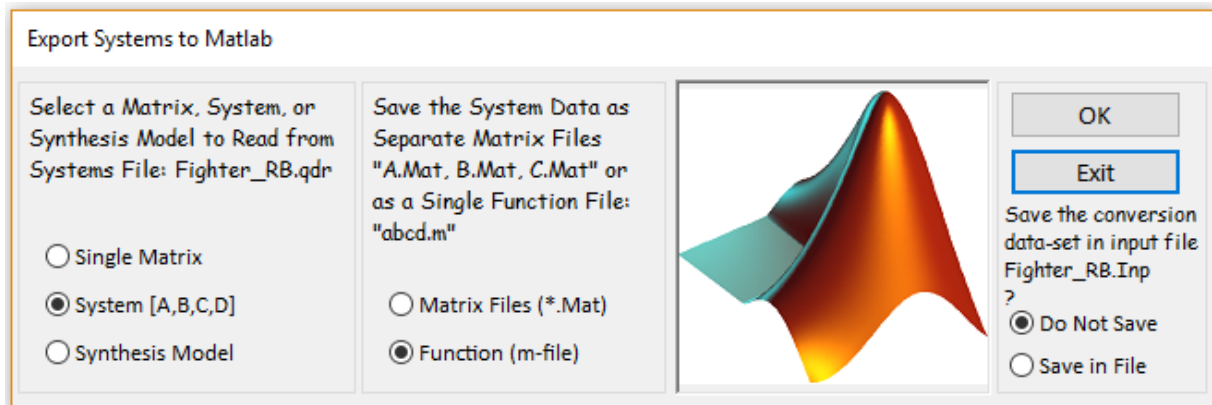


Exporting Systems from File

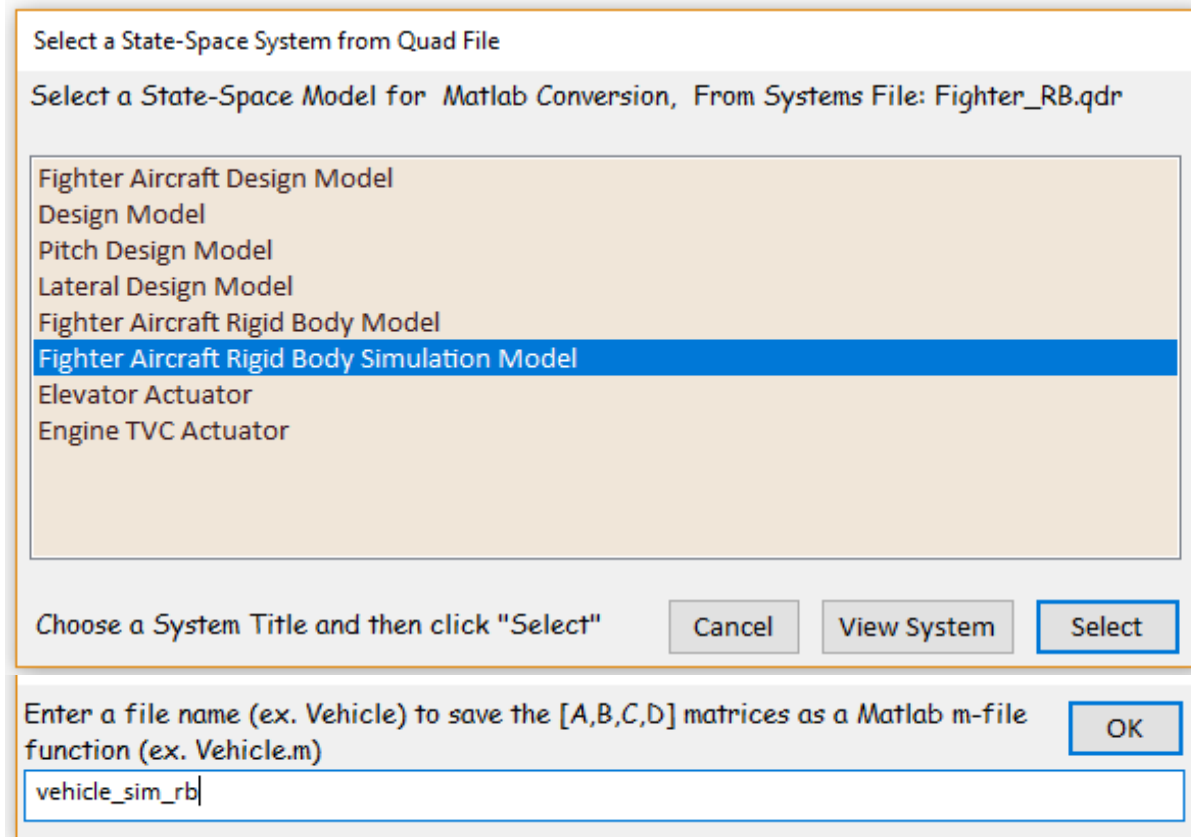
To convert one of the systems, let's say "*Fighter Aircraft Rigid Body Simulation Model*" from file "*Fighter_RB.Qdr*" to the Matlab project subdirectory you must select again the same source directory "... \Fighter Aircraft\ Rigid Body Design" where the systems file is located, and from the Flixan main menu go to "*Utilities*", "*Matlab Conversions*", and then "*Export to Matlab*". From the systems filename selection menu select the file "*Fighter_RB.Qdr*", and from the Matlab project directory selection menu select the destination directory "... \Examples \Fighter Aircraft\ Rigid Body Design\ Mat_Rigid", as before.



In the following dialog you must specify that the input data is a state-space system (A, B, C, D) and it will be converted in Matlab function format rather than as separate matrix files, then click on the "OK" button.



The following menu shows the titles of the systems which are included in file “*Fighter_RB.Qdr*”. Select the title of the system to be converted into Matlab and click “OK”. You must also enter the name of the m-file that will contain the state-space system as a function “*vehicle_sim_rb.m*”. This file will be saved in the Matlab analysis subdirectory. Note, that you should not enter the “.m” in the filename field below.

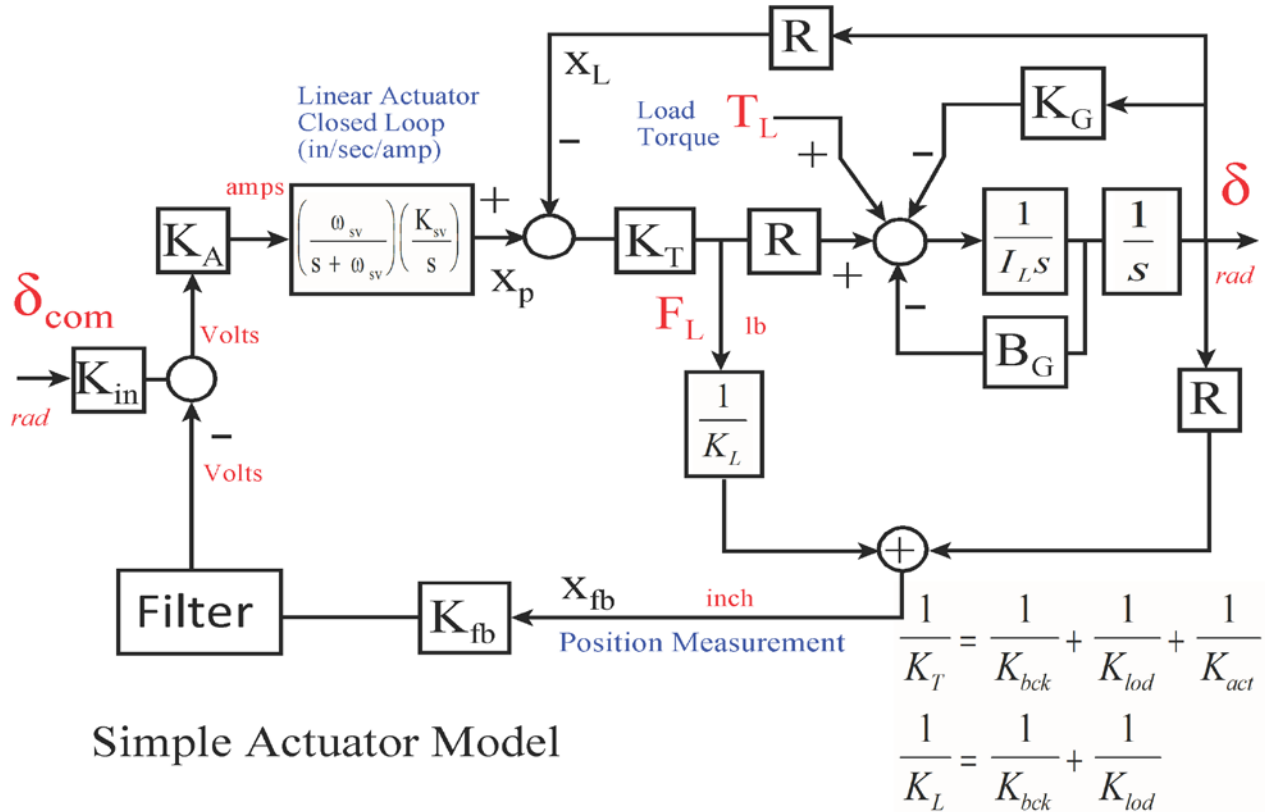


Repeat the system conversion process to transform two more systems from file “*Fighter_RB.Qdr*” to the Matlab analysis folder. The system titles are: “*Pitch Design Model*” and “*Lateral Design Model*”. The names of the corresponding Matlab functions are: “*vehi_pitch_des.m*” and “*vehi_later_des.m*” respectively. These systems are already included in the Matlab project folder.

1.6 Actuator Models

There are two actuator models included in this analysis that will be used in Matlab simulations. The input data file “*Fighter_RB.Inp*” contains the datasets for the two actuators and the Flixan actuator modeling program will be used to generate their state-space systems. The dataset titles are “*Elevator Actuator*” and “*Engine TVC Actuator*”. The first set is used to create models for all three aerosurface actuators, and the second set is for the two pitch gimbaling engines. They are “simple generic actuator” types, shown in the figure below, and described in detail the actuators section 4.3.1. The simple model consists of the actuator position servo loop which includes also the servo-valve dynamics as a first order lag, and the rotational dynamics of the load which is coupled with the position control servo via the total system rotational stiffness about the hinge. This model captures the local structural flexibility at the pivot, because the total stiffness consists of three individual stiffnesses: the backup structure at the vehicle/ actuator attachment point, the actuator shaft, and the rotational stiffness of the load about the hinge. They combine together to a total

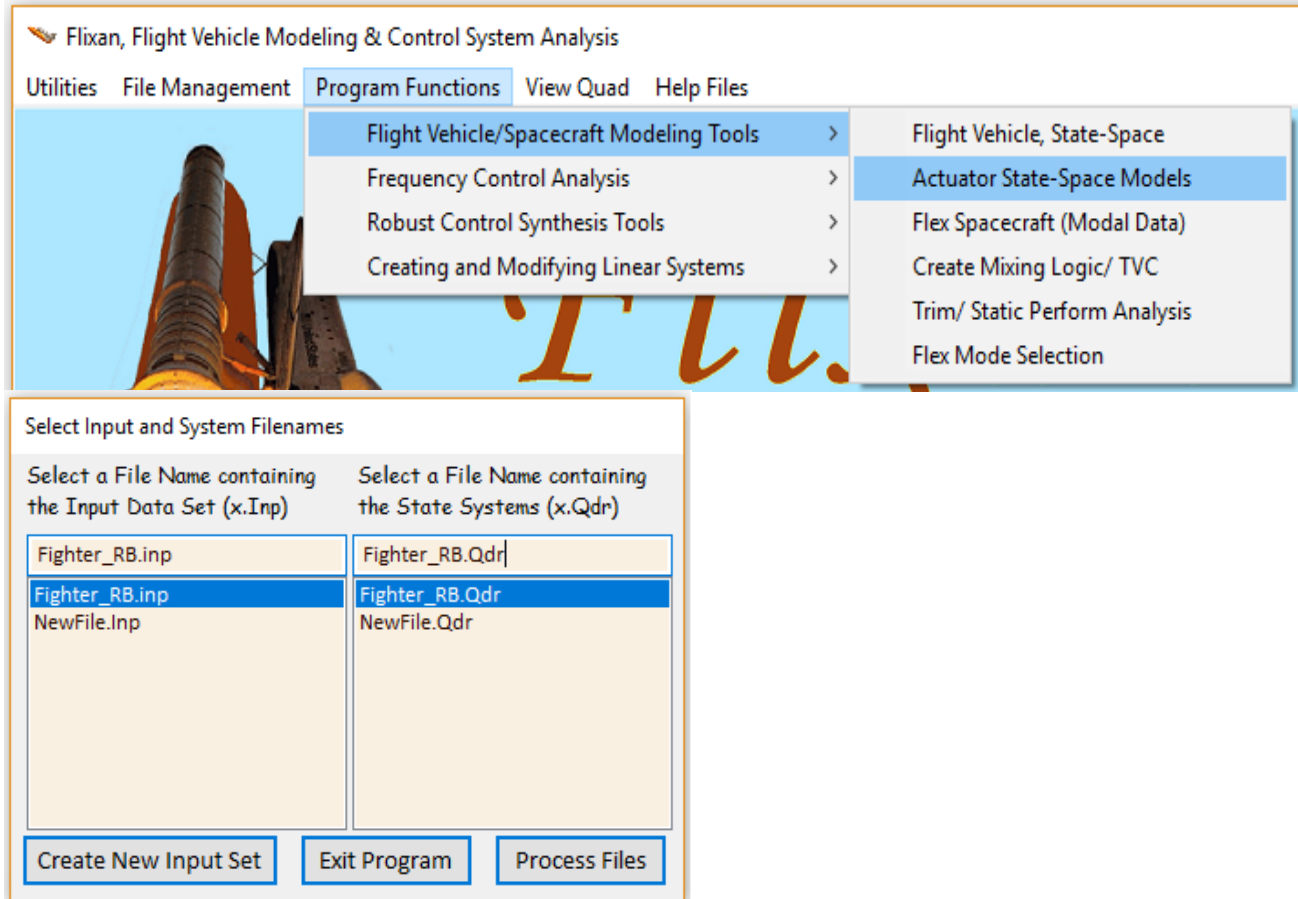
stiffness (K_T) which determines the resonance frequency of the aerosurface or nozzle about the hinge. The ideal position measurement at the input of the actuator servo loop should be (X_p). The measurement, however, is corrupted because the sensor that measures the shaft extension is affected by the compliance of the shaft itself (K_{act}). The mechanical feedback loop via ($1/K_L$) captures this measurement error. Otherwise, if the actuator shaft is perfectly rigid, ($K_T=K_L$), and the position feedback measurement (X_{fb}) is exactly (X_p). Note that this actuator model is only applicable for rigid-body models because it includes the combined stiffnesses at the vehicle backup structure and load. When dealing with finite element models, however, those stiffnesses are already included in the FEM, and therefore, they should not be included in the actuators.



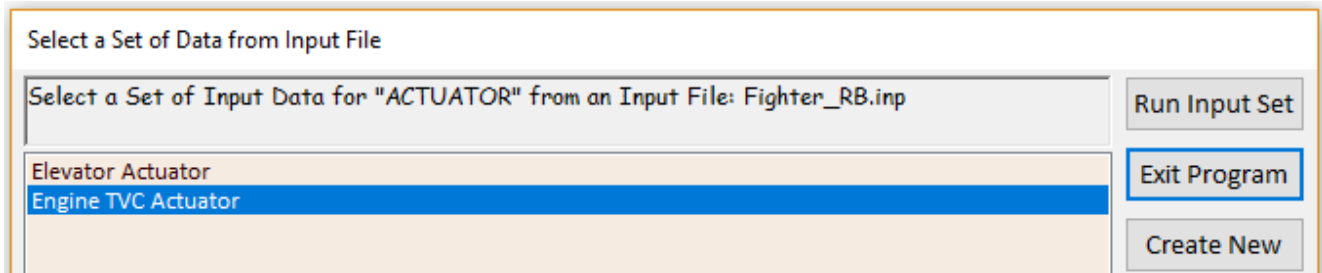
In addition to using separate models for the aero-surfaces and the TVC engines the actuator models must be modified before they can be used for flexible vehicle analysis. The reason is because in the rigid-body modeling case the local structural resonance in each actuator is captured by the total stiffness (K_T) inside the actuator model. In the flexible vehicle case, however, the actuator local resonances are included in the structural model, because the vehicle flex modes are calculated with the actuators included in the finite elements model and the hinges locked. The actuators used in flex modeling, therefore, are stiffened up by raising the backup and load resonances in order to avoid including the local resonance twice in the simulation model.

Running the Actuator Program

To generate the engine actuator state-space system, start the Flixan program and select the folder “*Examples\Fighter Aircraft\Rigid Body Design*”. Go to “*Program Functions*”, “*Flight Vehicle/Spacecraft Modeling*”, and then “*Actuator State-Space Models*”. From the filenames selection menu select the input and system files: “*Fighter_RB.Inp*”, and “*Fighter_RB.Qdr*”, as before.



The next menu includes the titles of the actuator datasets which are saved in the input file. Select the second title “*Engine TVC Actuator*” and click on the “*Run Input Set*” button to read the actuator data.



The dialog that follows shows the data used by the Flixan “simple actuator” modeling program to generate the engine TVC actuator. The $30/(s+30)$ first order transfer function captures some of the internal actuator dynamics. Click on “Run”, and the actuator state-space system will be saved in the systems file under the title “*Engine TVC Actuator*”. The definitions of the system states, inputs, and outputs are also included below the matrices.

Generic Actuator Input Data

Enter Parameters for Simple Generic Actuator Model Exit

Engine TVC Actuator Save

Run

Input Command Gain, (Kc), (Volt/Rad)	1.000
Amplifier Gain, Ka, (Amps/Volt)	8.000
Closed-Loop Actuator Servo Gain, Ksv, (ft/sec/amp)	1.000
Actuator Stiffness, Kact, (Piston+Oil/Electric), (lb/ft)	0.3800E+08
Load Stiffness, Klod, (Nozzle or Surface), (lb/ft)	0.2000E+08
Vehicle Backup Structure Stiffness, Kbck, (lb/ft)	0.6000E+07
Moment Arm (R), Actuator Rod from Gimbal, (feet)	1.000
Load Mom. of Inertia about the Gimbal (ft-lb-sec ²)	280.0
Load Gimbal Bearing Spring Constant, Kg, (ft-lb/rad)	0.000
Load Gimbal bearing Viscous Damping, Be, (ft-lb-sec)	1100.
Actuator Position Sensor Feedback Gain, Kfb, (volts/ft)	1.000

Order of Compensator C(s) in the Actuator Loop

No Compensation First Order Second Order

Enter Filter Coefficients

Define a First Order Filter Transfer Function: OK

$(T_n s + 1) / (T_d s + 1)$

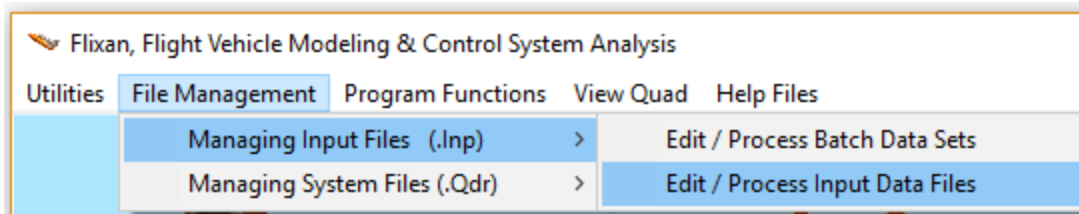
Numerator Time Constant (Tn) in (sec) 0.000

Denominator Time Constant (Td) in (sec) 0.3330E-01

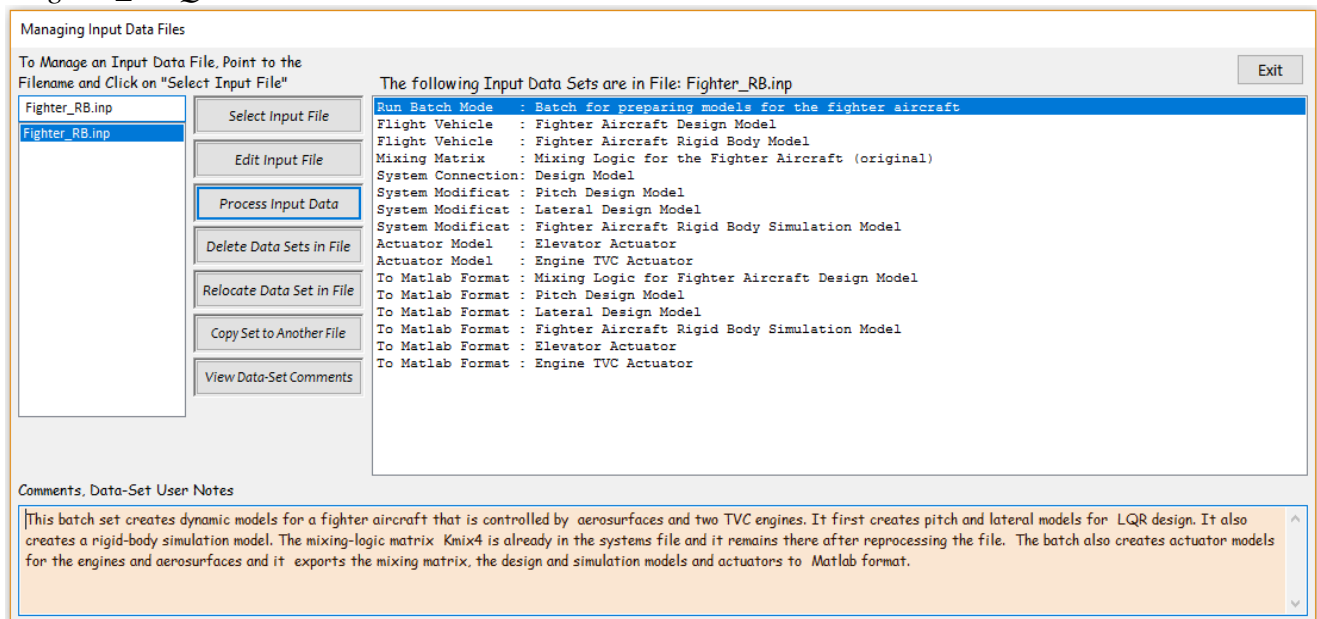
The process is repeated to generate the elevator actuator model which is also saved in the systems file under the title “*Elevator Actuator*”. The two actuator systems are also exported as Matlab m-file functions in the Matlab analysis subdirectory “*Mat_Rigid*” using the same process that was described earlier. The two m-file names are “*elevator.m*” and “*engine_tvc.m*”.

1.7 Batch Mode Processing

The interactive processing of the datasets, however, in the input file: “*Fighter_RB.Inp*” is time consuming and it is only used when creating input files for a new project/ analysis. After the process is debugged, the subsequent processing of the data files is typically performed in batch mode using a batch set. A batch dataset is included in file “*Fighter_RB.Inp*”. Its title is “*Batch for preparing models for the fighter aircraft*” and it can be processed from the File Manager utility, as shown below.



The input file manager dialog comes up, and from the menu on the left side select the input file “*Fighter_RB.Inp*” and click on “*Select Input File*”. The menu on the right shows the titles of the datasets that are saved in this input file. Select the batch set and click on “*Process Input Data*”. The batch will then process the datasets which are in the input file and save the systems in file “*Fighter_RB.Qdr*”.



1.8 Control Design

The control design and analysis in this example is performed using the Matlab program in subdirectory “\Fighter Aircraft\Rigid Body Design\Mat_Rigid”. The actuator and vehicle pitch and lateral models are already converted to Matlab format and saved in this folder. The m-file “LQR_des.m” loads the systems into Matlab workspace and performs the control design using the LQR method. Two separate designs are performed using the pitch and lateral aircraft design models in files: “*vehi_pitch_des.m*” and “*vehi_later_des.m*” respectively, because the LQR method requires simple rigid models for synthesizing the state-feedback gains.

In the longitudinal directions the objective is to control altitude and velocity independently by pitching and throttle control. However, these variables are strongly coupled together and the design model is used to properly derive the cross-state-feedback gains, via the LQR method, that will achieve this objective. The dynamic model states represent changes in: pitch attitude, rate, alpha, altitude and velocity. The original design model state vector is augmented from 5 to 7 states by including the integrals of the altitude and velocity. This augmentation is implemented in the Simulink file: “*Pitch_Vehi_int.Mdl*”, shown in figure below. The Matlab script LQR_des.m uses the augmented design model to calculate the (2x7) state feedback matrix K_{gp} . In the lateral directions the design system states are: roll attitude and rate, yaw rate, and sideslip angle beta. The original design model state vector is augmented from 4 to 5 states by including the integral of the roll attitude, as shown in figure. The yaw attitude is not included in the state vector because in this example we are not interested to control yaw independently, and a roll command is expected to control roll but cause a drift in yaw. This augmentation is implemented in Simulink file “*Later_Vehi_int.Mdl*”, shown in figure below. The LQR design script uses the augmented lateral design model to calculate the (2x5) state feedback matrix K_{gl} .

```

% LQR Design File
d2r=pi/180; r2d=180/pi;
load Kmix4.mat Kmix5 -ascii % Mixing Logic Matrix
[Av, Bv, Cv, Dv]= vehicle_sim_rb; % Simulation Model
[Ae, Be, Ce, De]= elevator; % Elevator Actuator
[At, Bt, Ct, Dt]= engine_tv; % Engine TVC Actuator

% Longitudinal LQR Design
[Ad1, Bd1, Cd1, Dd1]= vehi_pitch_des; % Load Pitch LQR Design Model
[Ai,Bi,Ci,Di]=linmod('Pitch_vehi_int'); % Augment LQR Design Model with integr:

Q= [1.e-7,1.e-5,1.e-7, 0.04,0.08, 0.004,0.008]*0.1; % Weight Matrices Q, R [1.e-7,0.001,0.0
Q=diag(Q);
R= [5, 1]*0.02; R=diag(R); % R= [5, 1]*0.001;
[Kgp, s, e] = LQR(Ai,Bi,Q,R) % Pitch State-Feedback Matrix

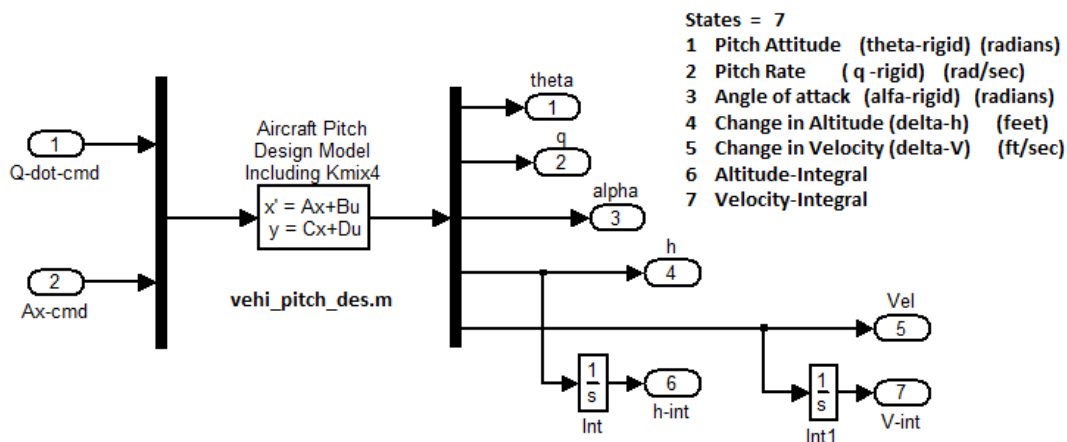
% Lateral LQR Design
[Ad2, Bd2, Cd2, Dd2]= vehi_later_des; % Load Lateral LQR Design Model
[Ai,Bi,Ci,Di]=linmod('Later_vehi_int'); % Augment LQR Design Model with integr:

Q= [40, 0.1, 1.e-5, 0.05, 1.0]; Q=diag(Q); % Weight Matrices Q=[40,0.1,0.001,0.05,
R= [1, 7]*0.1; R=diag(R); % Lateral State-Feedback Matrix
[Kgl, s, e] = LQR(Ai,Bi,Q,R)

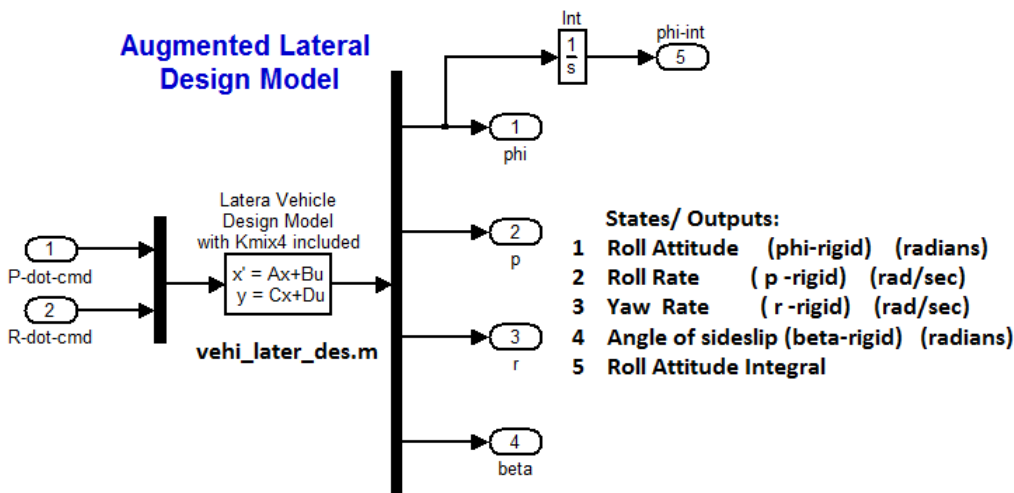
save Kgp.mat Kgp -ascii
save Kgl.mat Kgl -ascii

```

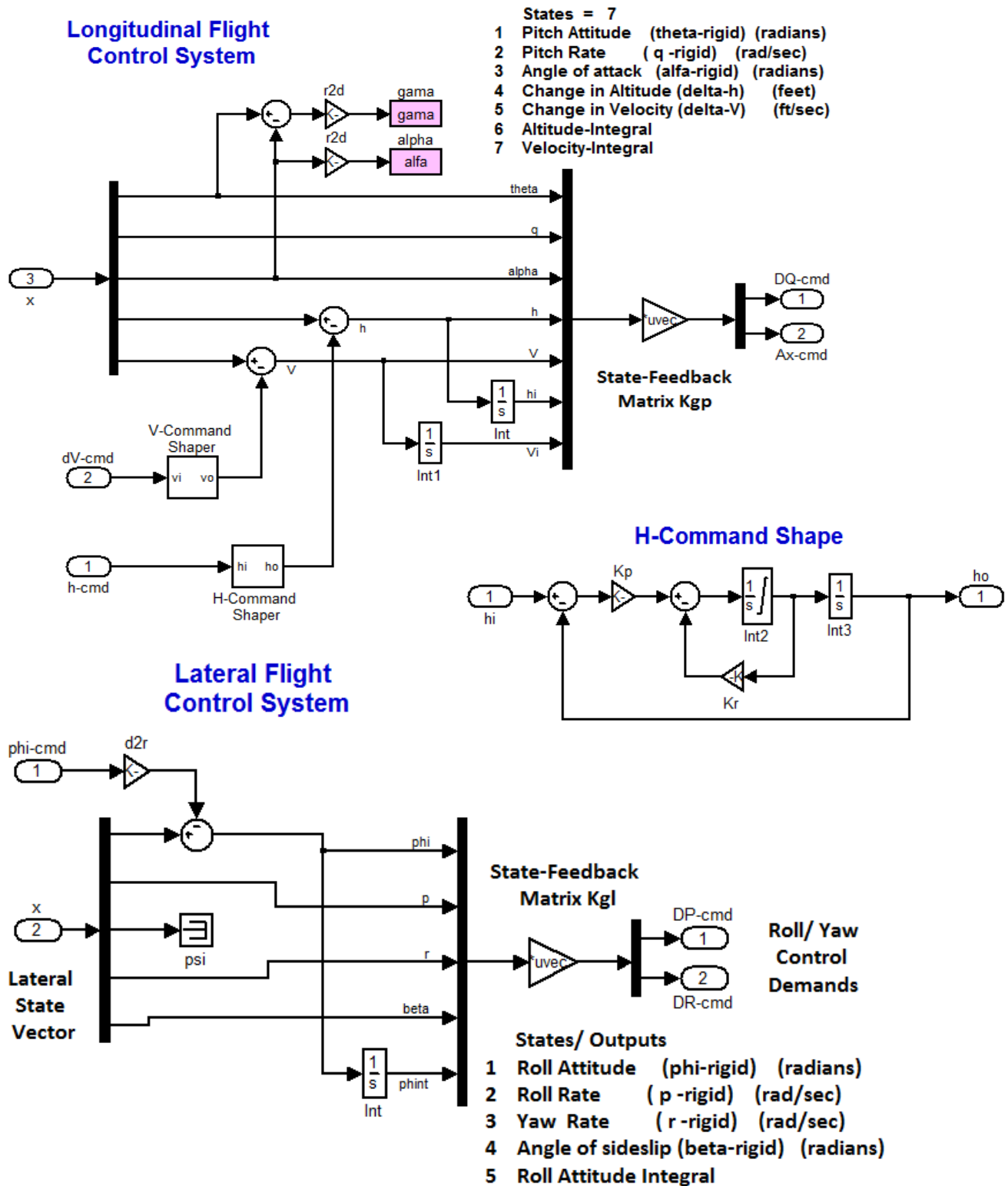
Augmented Pitch Design Model



Augmented Lateral Design Model

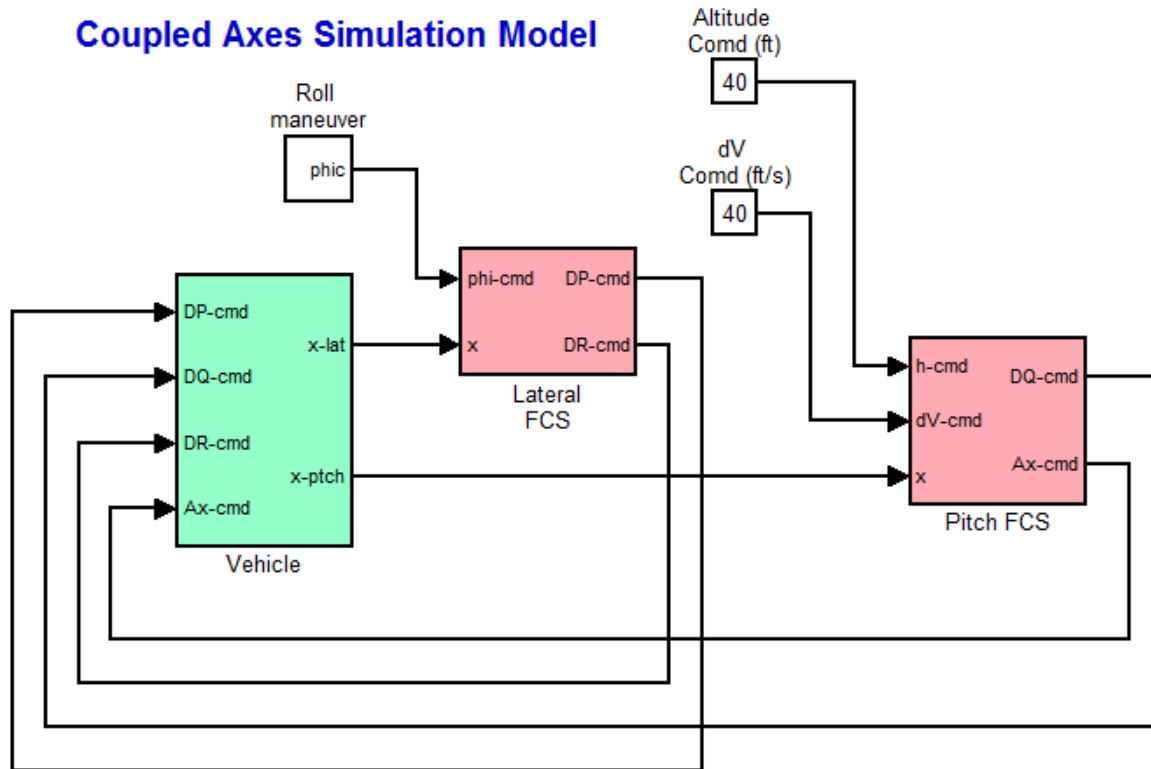


The longitudinal and lateral flight control systems are shown below and they include the previously designed state-feedback matrices. They convert the state-feedback error signals to acceleration flight control demands: (DQ and Ax) in pitch, and (DP and DR) in roll and yaw. In the longitudinal axes the system independently controls altitude and velocity. The delta-velocity and delta-altitude commands are shaped by command rate-limiting logic to prevent the controls from saturating.



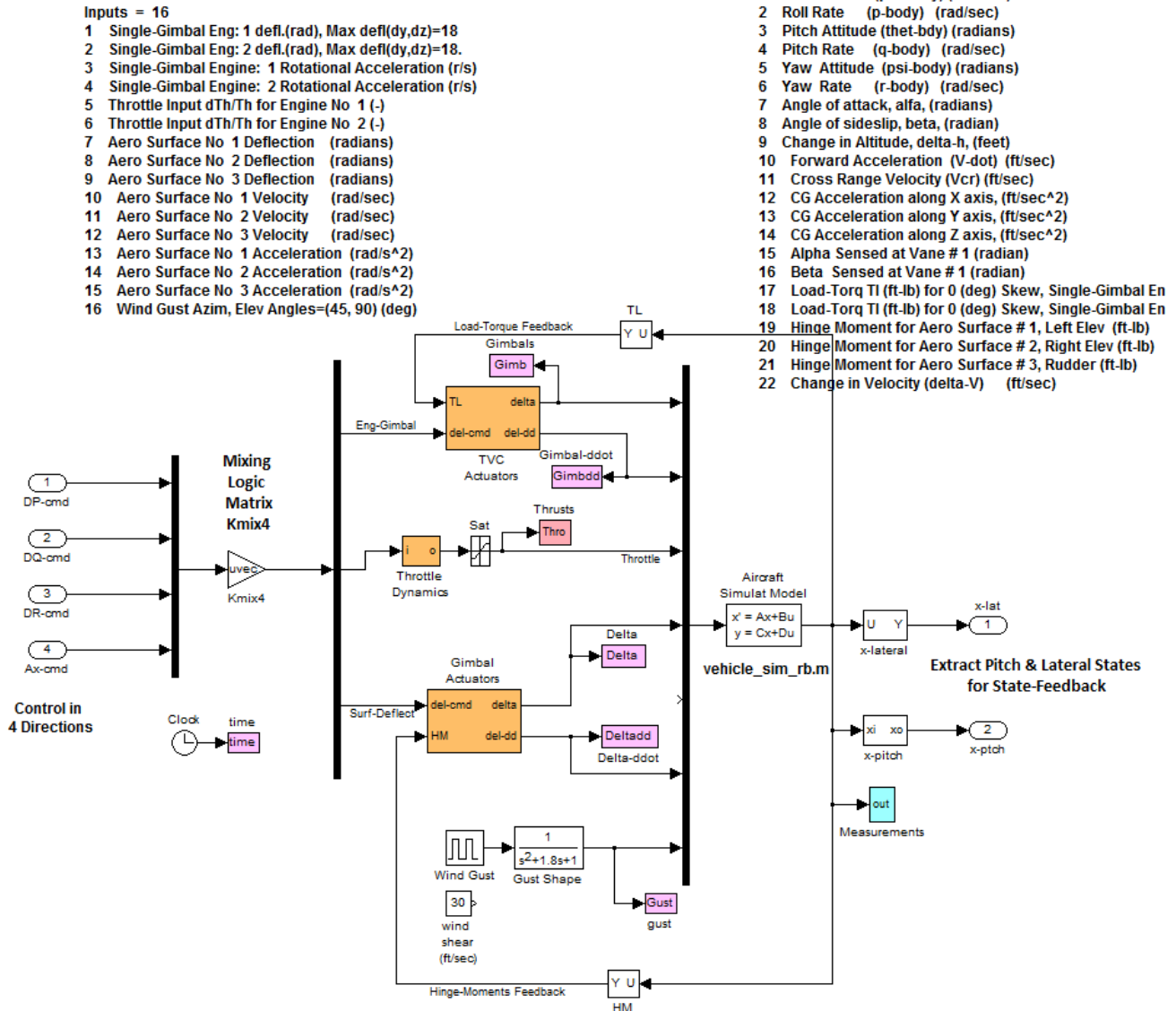
1.9 Simulation

The aircraft simulation is performed in Matlab using Simulink. The Simulink model is shown below and it is saved in file “*RigidBody_Sim.Mdl*”. The vehicle block includes the aircraft model, the actuator systems for the aerosurfaces, TVC, and throttle controls, the mixing-logic matrix, and the gust disturbance. The aircraft simulation system includes both, pitch and lateral dynamics and it is loaded into Matlab from file “*vehicle_sim_rb.m*”. The actuators and mixing logic matrix are also loaded by running the m-file “*run.m*”. In the longitudinal axes the commands are changes in the aircraft altitude and velocity relative to trim altitude and velocity values. In the lateral axes the command is roll attitude.



The vehicle dynamics block is shown in more details in the next figure. There are three rotational and one axial acceleration demands $(DP, DQ, DR, Ax)_{cmd}$ coming from the flight control systems. They are converted by the mixing logic matrix K_{mix4} into 3 aerosurface deflection commands, 2 pitch gimbal commands, and 2 thrust variation commands for the two 30,000 (lb) engines. The effector commands drive the actuators and become actual deflections and throttles which are inputs to the aircraft system. The actuator subsystems are loaded from files “*elevator.m*” and “*engine-tvc.m*”. There is also a wind-gust velocity disturbance that excites the aircraft system. The filter is used for shaping the gust impulse. The direction of the wind is defined in the vehicle data, and it is towards the vehicle, perpendicular to the x axis, and at 45° between the $+Z$ and the $+Y$ axes. The mechanical feedback loops are between the vehicle hinge moment outputs and the actuators. They capture the actuator torque-loading due to vehicle acceleration at the engines and surfaces.

Aircraft Model with Mixing Logic and Actuators

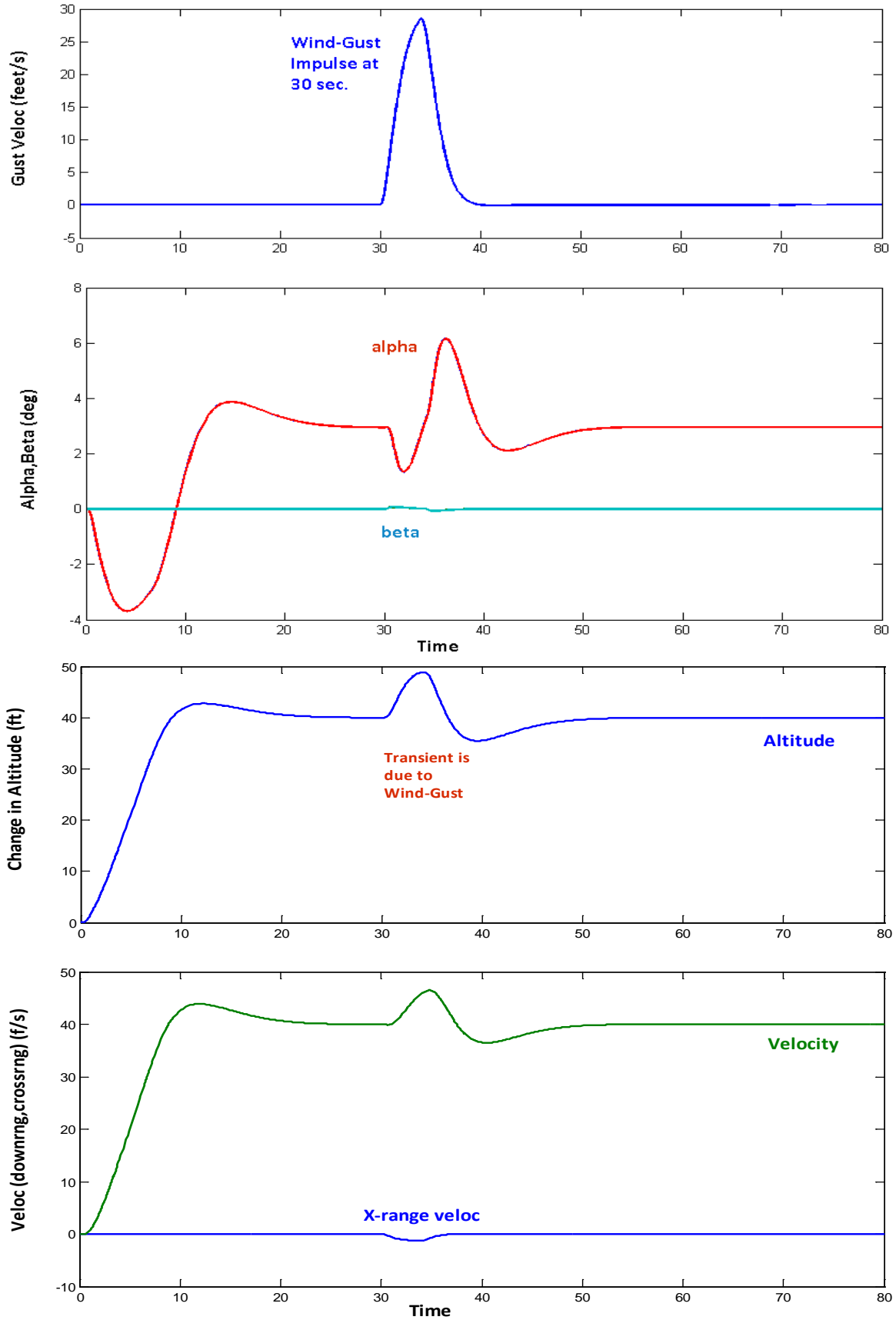


We will use this Simulink model to perform two simulations and analyze the response of the aircraft variables relative to trim conditions. The first simulation performs a longitudinal maneuver, where the vehicle is simultaneously commanded to increase in speed ($\delta V=40$ ft/sec) and to increase its altitude ($\delta H=40$ feet). In addition to the commands, a 30 (feet/sec) wind-gust disturbance is applied at time=30 seconds. The second simulation excites mainly the lateral directions. The vehicle is commanded to perform a roll attitude maneuver. When the simulation completes the Matlab script file “pl.m” is used to plot the Simulink data.

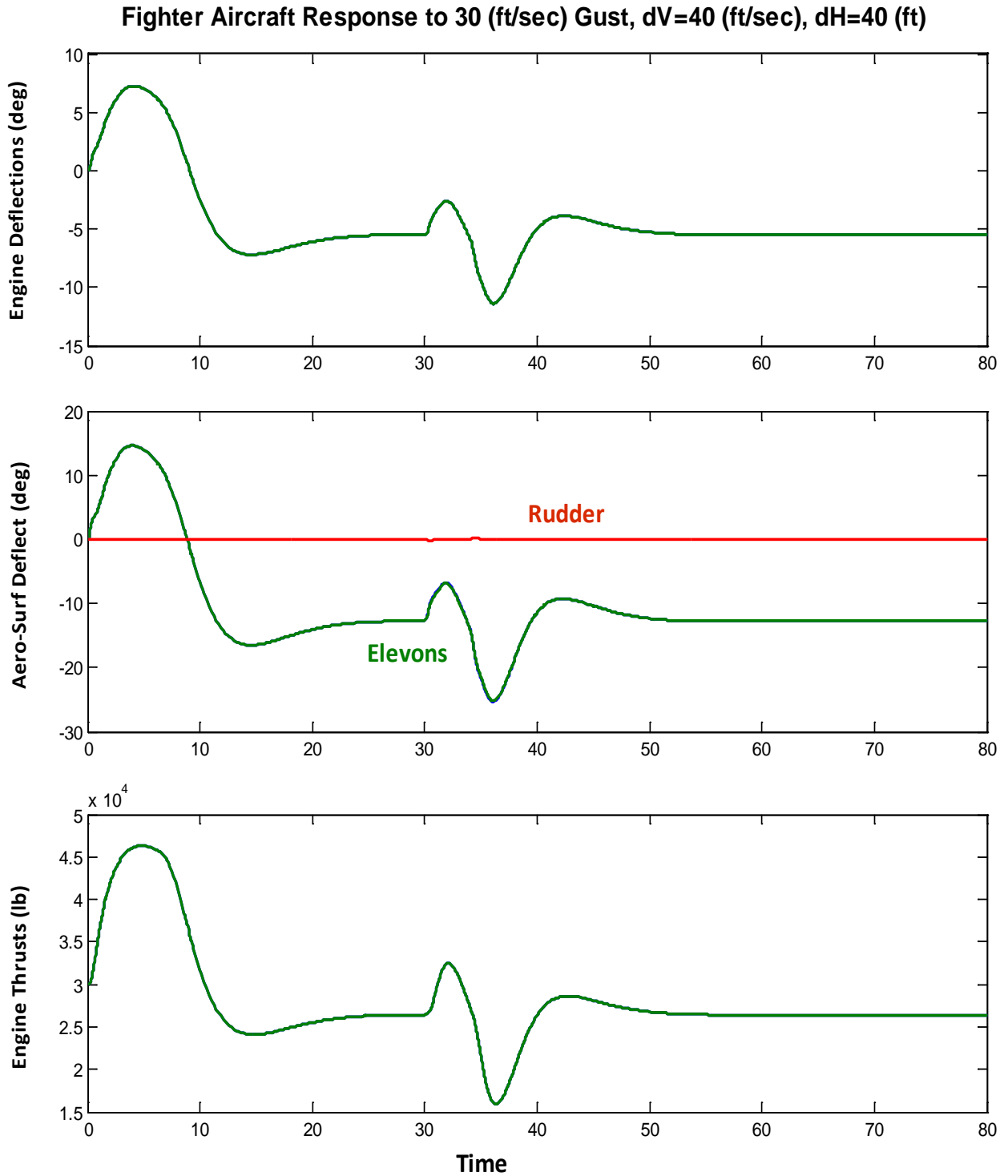
Altitude and Velocity Increase Maneuver

The figures below show the response of the aircraft variables during the longitudinal maneuver. The aircraft altitude and velocity adjust to track the input commands. The wind-gust at 30 sec causes some transient that eventually decay.

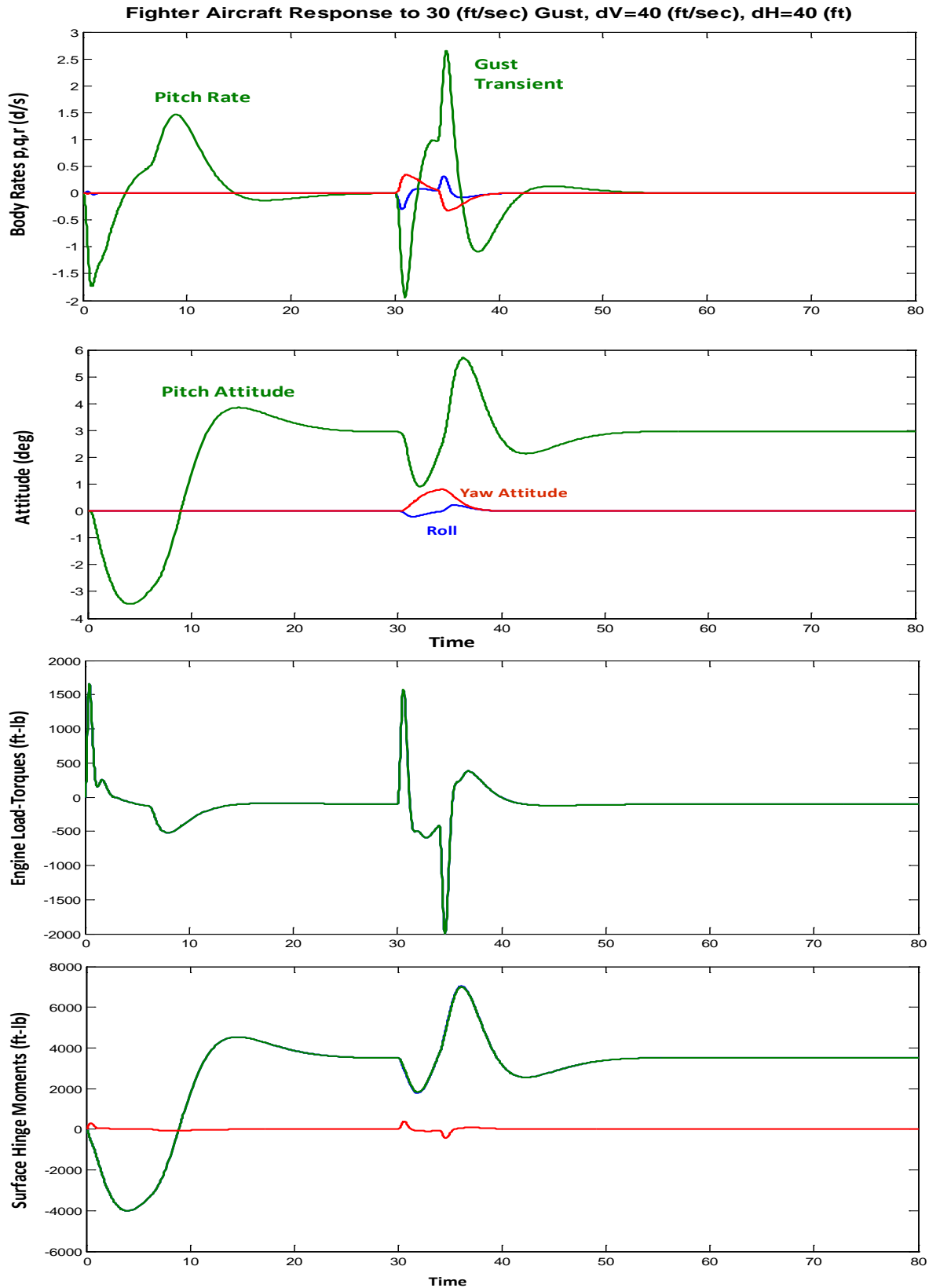
Fighter Aircraft Response to 30 (ft/sec) Gust, dV=40 (ft/sec), dH=40 (ft)



The figure below shows the deflections of the aircraft engines, the control surfaces, and the thrust variations required to perform the altitude and velocity change and also to react against the gust transient. The rudder deflection is almost zero. It also shows the engine thrusts which start at 30,000 (lb) each, they temporarily increase to 46,000 (lb) in order to raise the altitude and speed and they eventually reduce to 26,000 (lb). The wind-gust at 30 (sec) causes transients in all effectors.

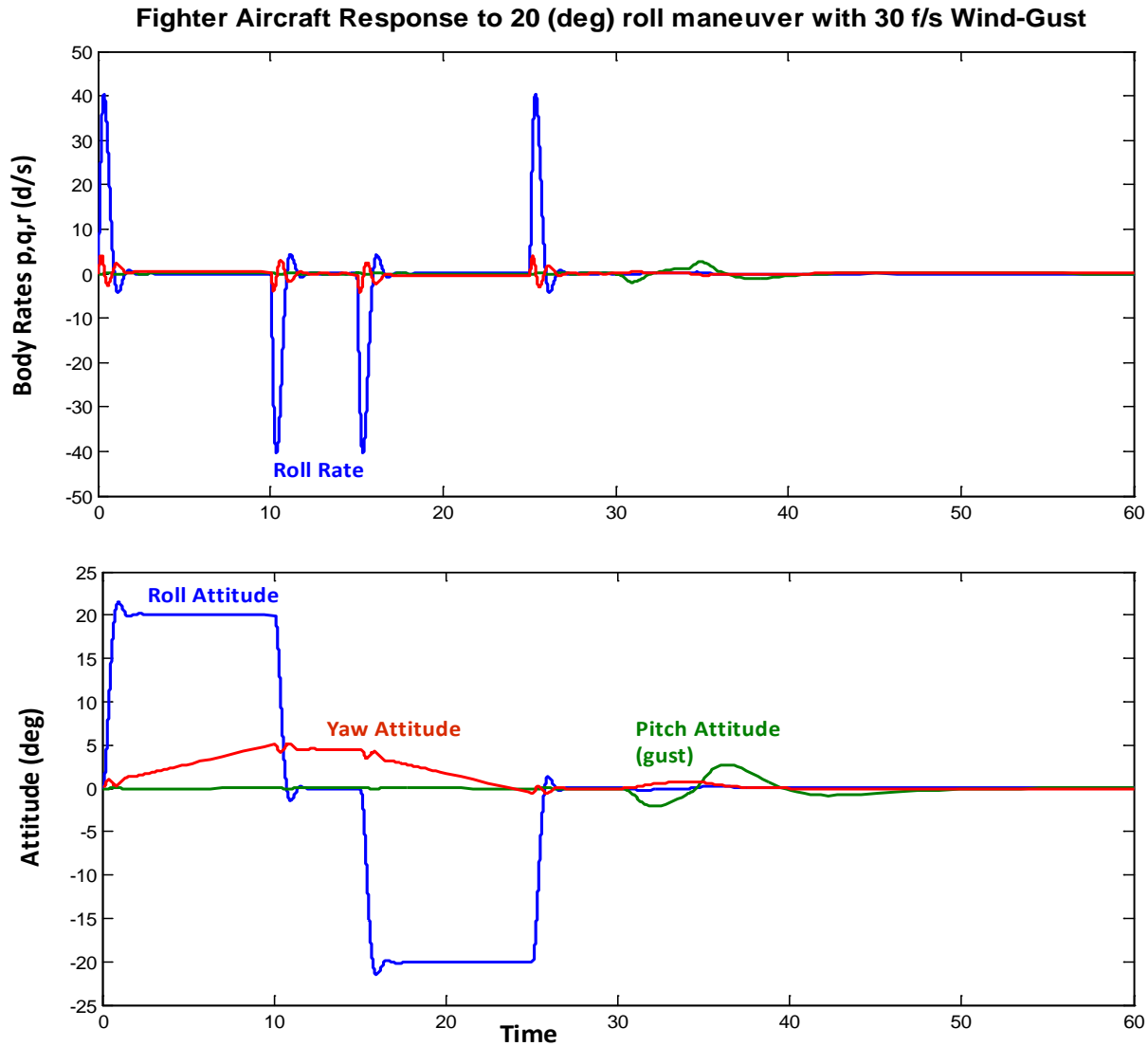


The next figure shows the aircraft attitude and rates. The motion is mainly in pitch direction (green). The wind-gust also causes a small transient in the lateral direction (red and blue). The load-torques at the two engine gimbals and the hinge moments at the three aerosurfaces are also shown. The hinge moments are mostly in the two elevons (green) due to the aerodynamic pressure.



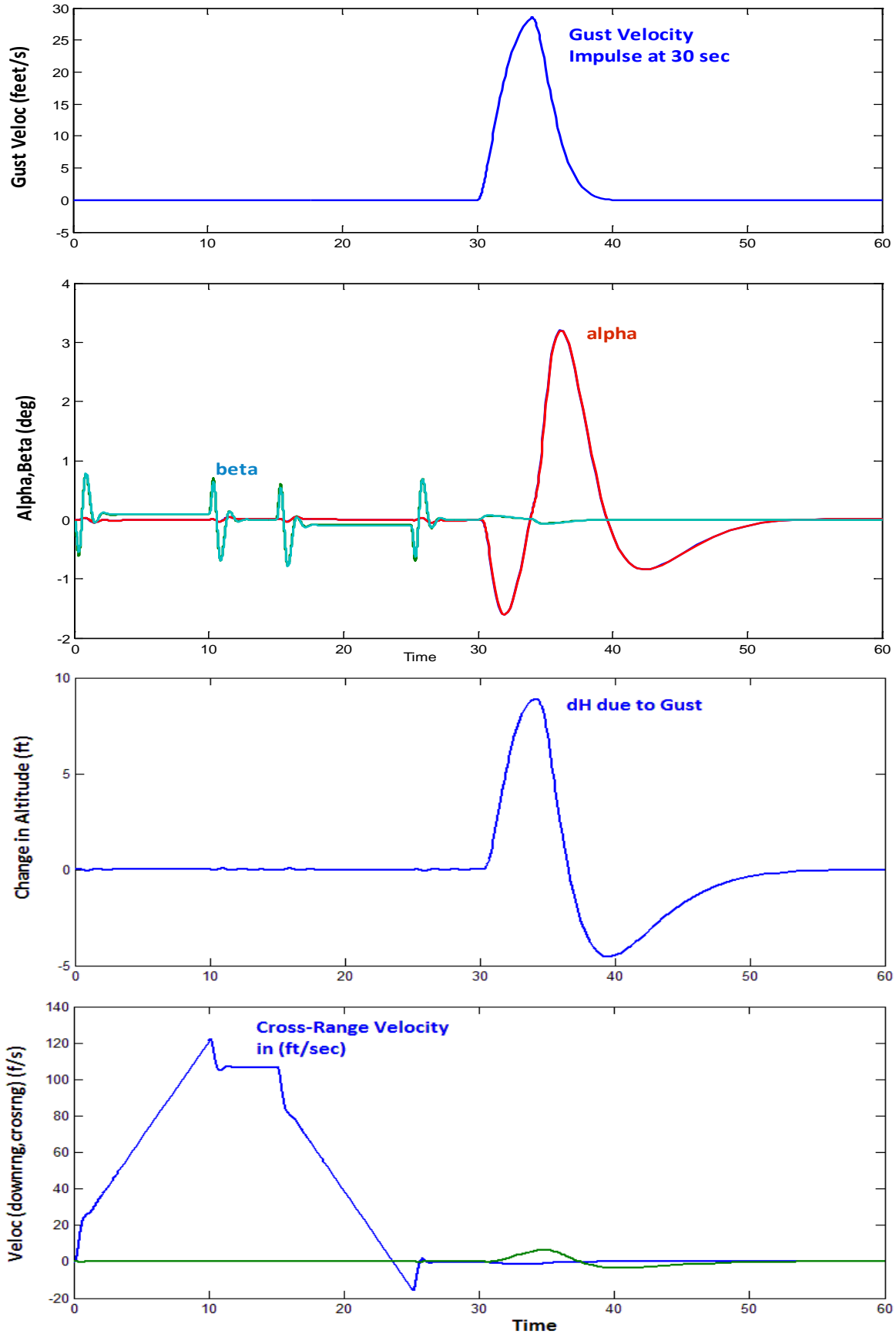
Roll Maneuver

The figures below show the aircraft response to a roll maneuver. This simulation demonstrates how the pilot may change the flight direction of the aircraft by performing roll maneuvers. The aircraft is commanded to rotate $+20^\circ$ in roll for 10 seconds, followed by a negative -20° roll rotation in the opposite direction for another 10 seconds. The wind-gust excitation is the same as in the previous example, occurring at 30 sec. The figure below shows how the aircraft attitude and rates respond to the roll maneuver.

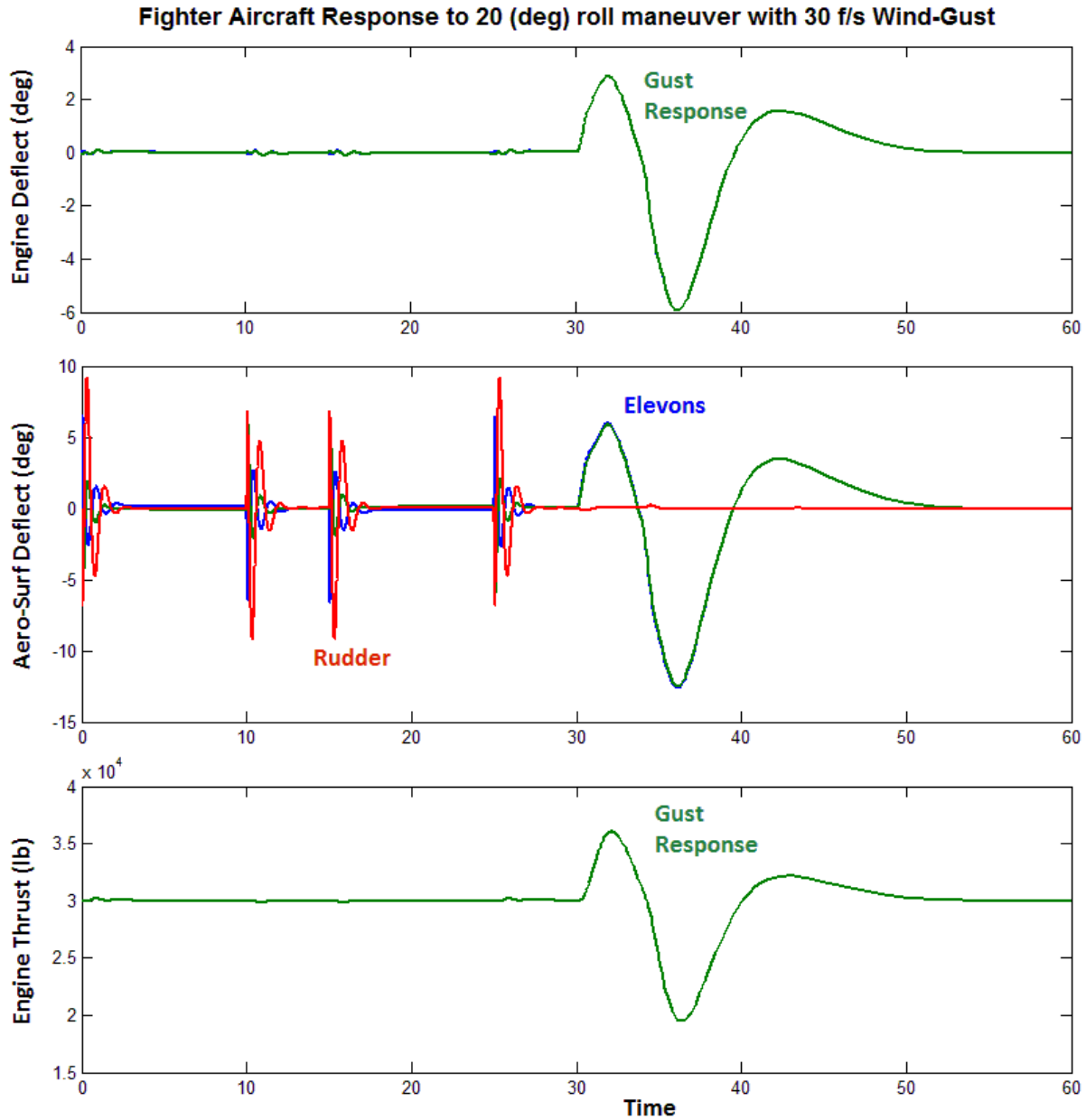


The following figures show the wind-gust velocity and the variations in the aircraft angles of attack (α) and sideslip (β). The sideslip is caused by the roll maneuver and alpha is due to the wind-gust. We also see the cross-range velocity (blue) due to roll. It changes the horizontal direction of the aircraft (not shown). The change in altitude (δH) is due to the wind-gust. The down-range speed (δV in green) changes slightly due to gust.

Fighter Aircraft Response to 20 (deg) roll maneuver with 30 f/s Wind-Gust

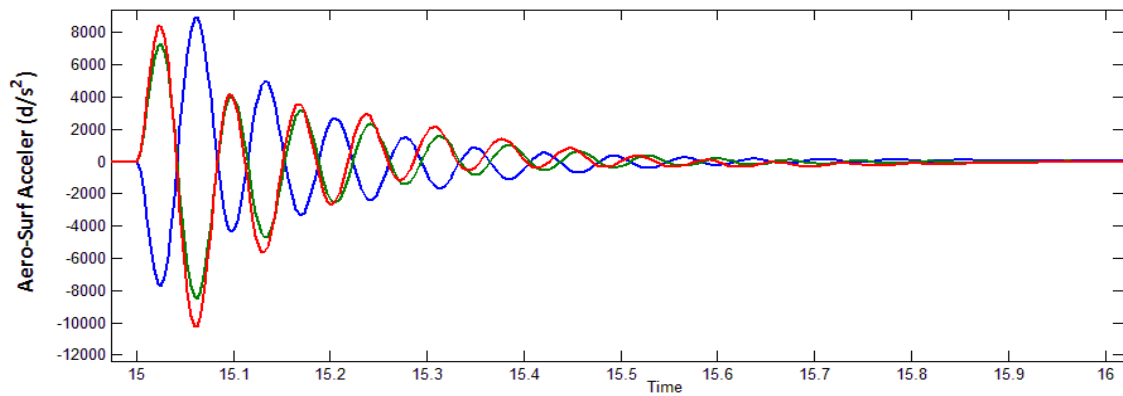
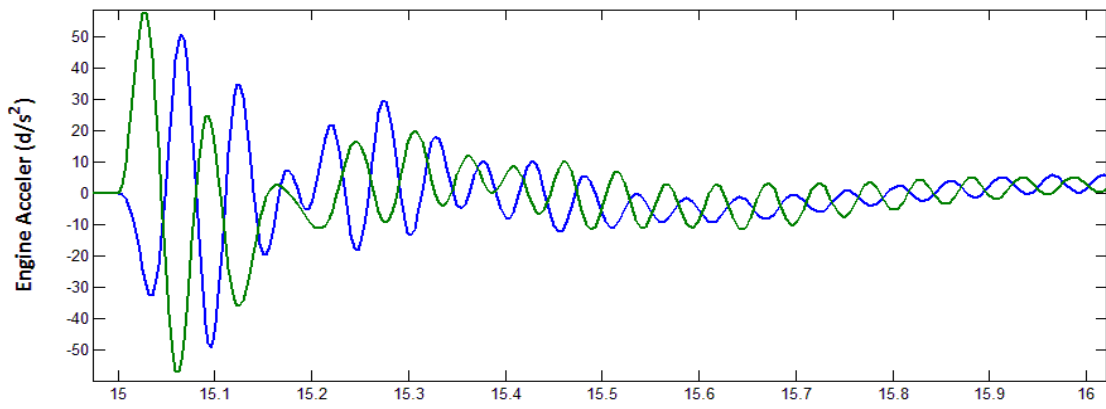
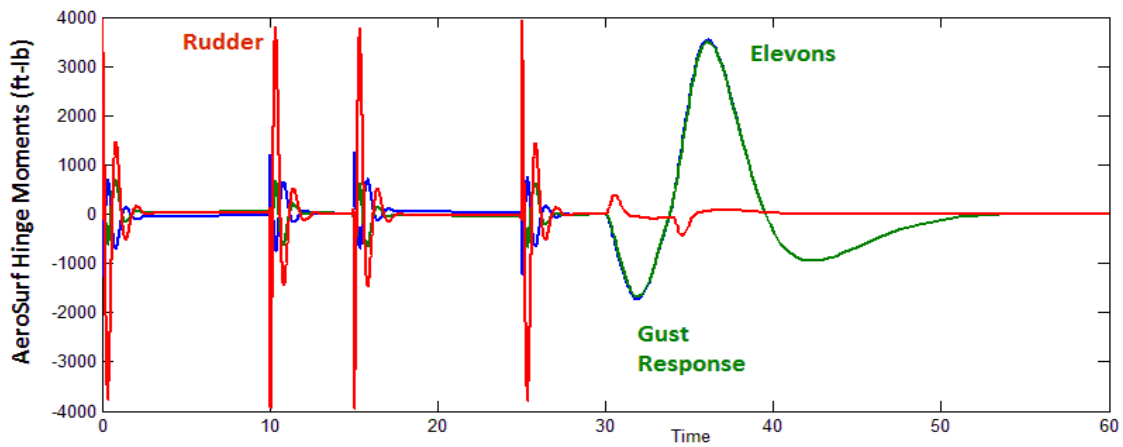
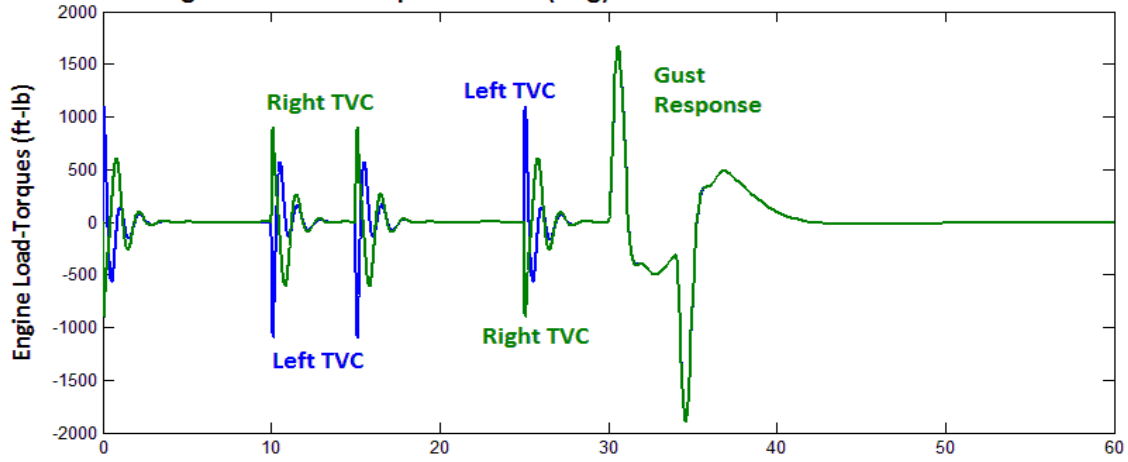


The next figure shows the aircraft controls. The aileron, differential left/right elevons (green and blue) and the rudder (red) are both used during the roll maneuver. The elevons and the engines are both used to counteract against the wind-gust. The gust also creates a transient on the engine thrusts which on the average they remain nominal at 30,000 (lb).



The next figure shows the load-torques at the two gimbaling engines and also the hinge moments at the three aerosurfaces. The left and right TVC and also the left and right elevons respond differentially during the roll maneuvers. The last figure at the bottom is not in the same scale as the top ones. It shows the accelerations of the two TVC engines and the three aerosurfaces in response to the roll command. It captures the differential action required for the roll maneuver during a short period of one second.

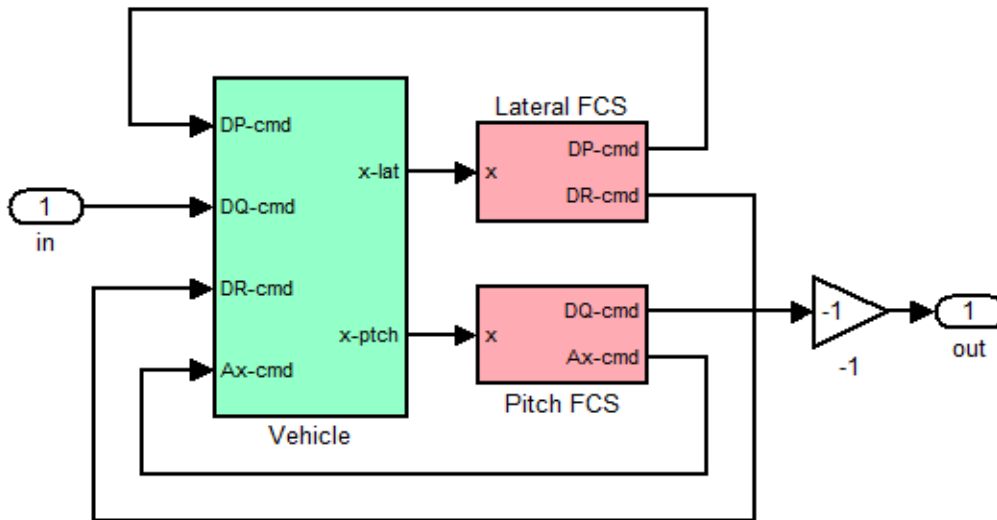
Fighter Aircraft Response to 20 (deg) roll maneuver with 30 f/s Wind-Gust



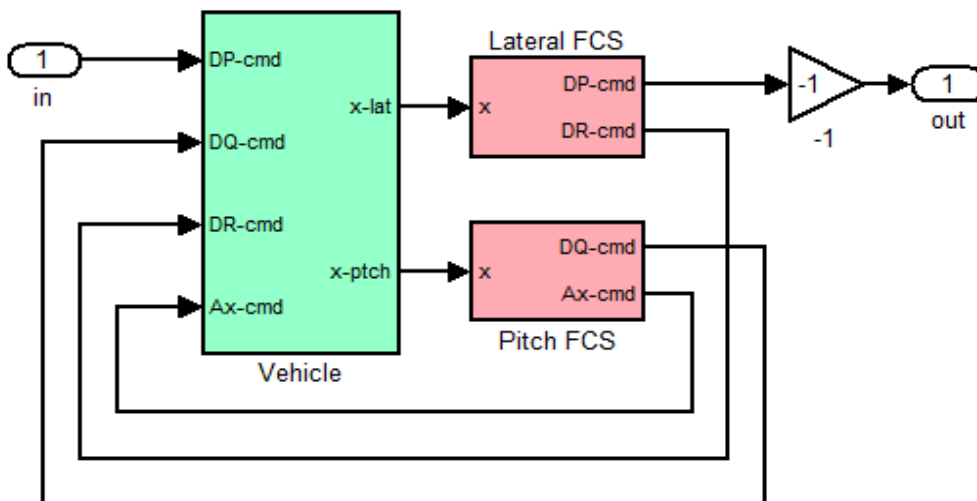
1.10 Stability Analysis

We will use Matlab to analyze the system stability in the frequency domain. The Matlab analysis is in subdirectory "...\\Fighter Aircraft\\Rigid Body Design\\Mat_Rigid" and is performed by the script m-file "run.m". It uses a similar Simulink model "Open_Loop.mdl" for the open-loop analysis, shown below, which consists of four control loops, three rotational and one translational. Only one loop is opened at a time when analyzing one of the control axes while the other three loops must remain closed, as shown below for the pitch axis stability analysis. The Simulink model must be modified by closing the pitch loop and opening another loop, roll for example, and rerunning the m-file. The stability analysis results in pitch and roll are shown in the figures below. They highlight the stability margins, which are sufficient. The stability margins for the remaining two loops, that is, yaw and the axial acceleration, can also be calculated similarly by modifying the Simulink block diagram and rerunning the Matlab m-file.

Open-Loop Analysis Model

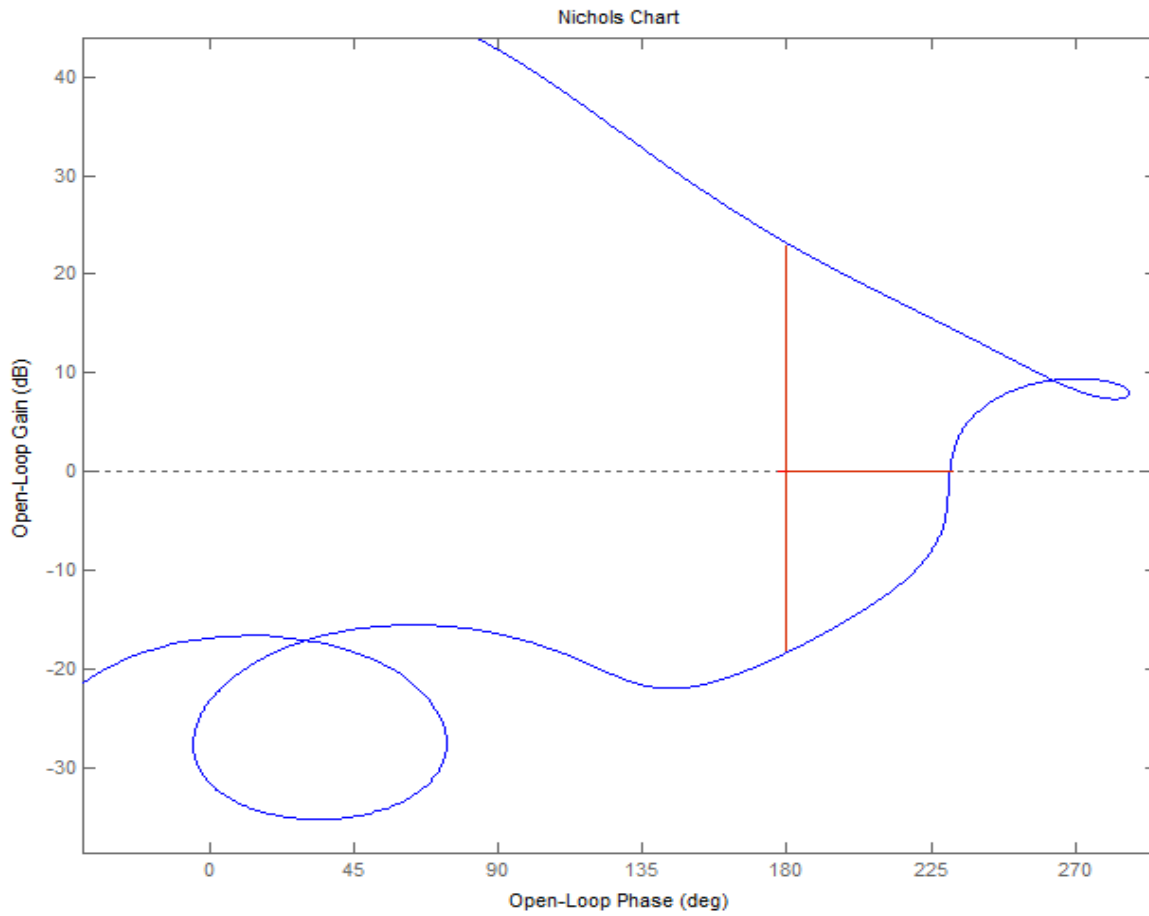


Pitch Loop Opened, Other Loops Closed

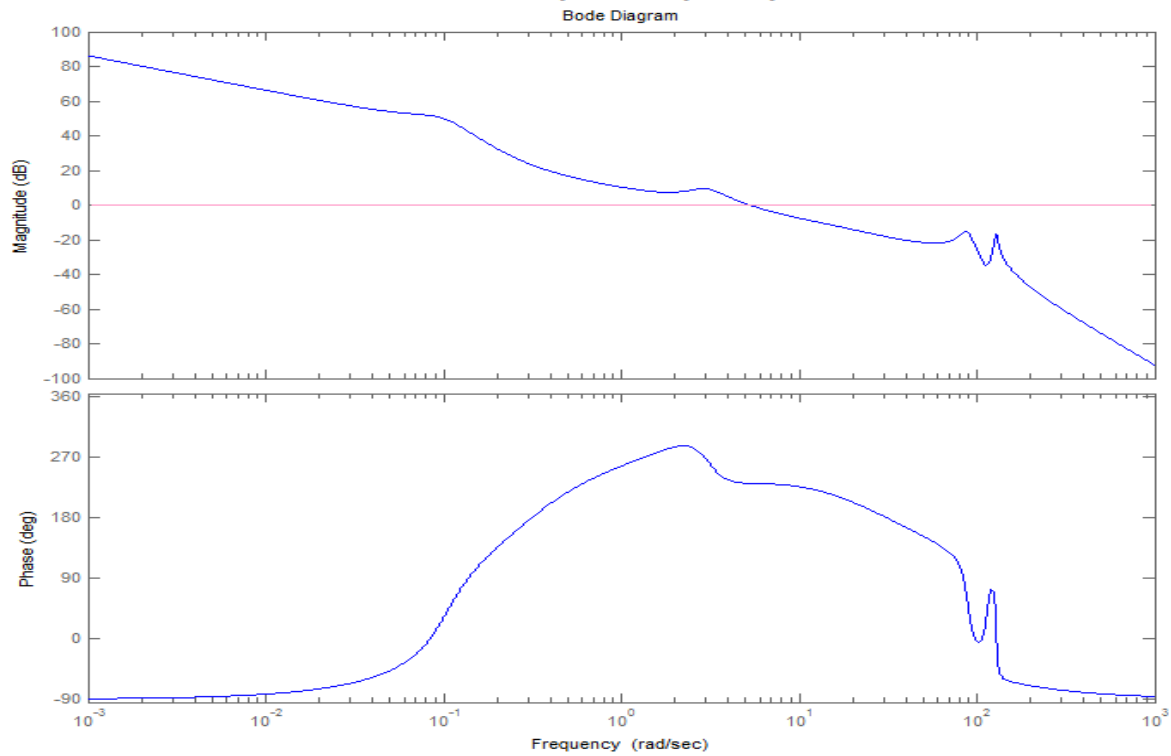


Roll Loop Opened, Other Loops Closed

Pitch Axis Stability

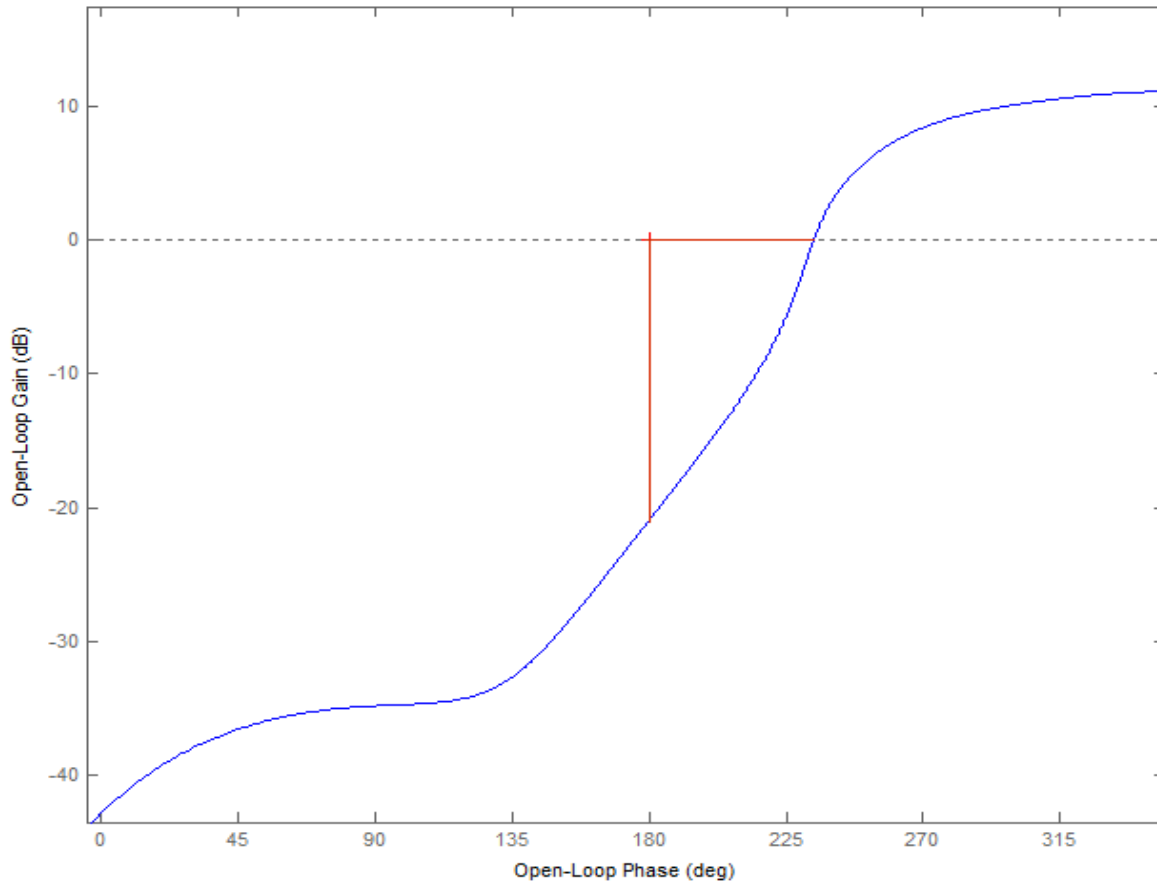


Pitch Axis Open-Loop Response



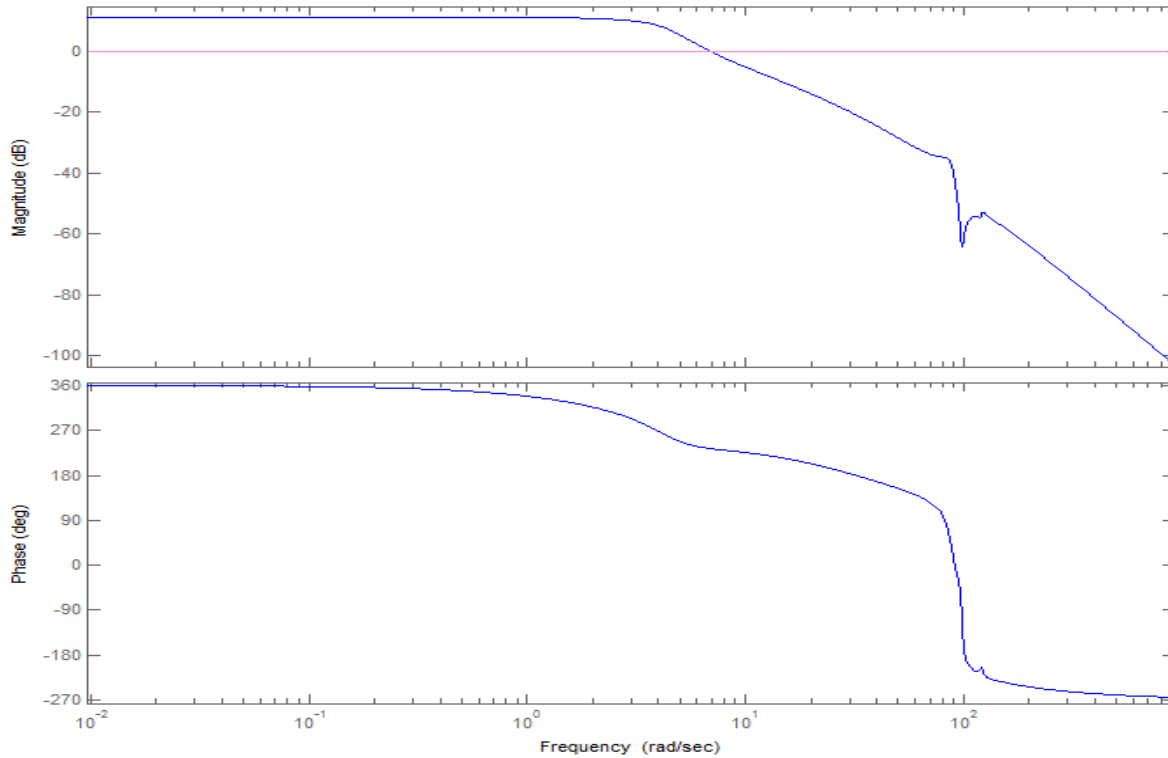
Roll Axis Stability

Nichols Chart



Roll Axis Open-Loop Response

Bode Diagram



2.0 Flexible Vehicle Analysis

To analyze stability and performance of the flexible vehicle a more complex model is needed that includes structural modes. The flight control system for the most part is similar to the state-feedback controller designed for the rigid-body, with some adjustments in order to attenuate the flex modes. In the flex model we shall ignore the aero-elastic effects and assume that the structure is excited mainly by the reaction torques of the engines and the control surfaces as they rotate about the pivots to control the vehicle. The vehicle flexibility is implemented by using inertial flex coupling coefficients, also known as h-parameters. They dynamically couple the angular accelerations of the aero-surfaces and gimbaling engines to excite the flex modes $\eta(j)$. The structure is excited by the angular accelerations of the TVC engines $\ddot{\delta}_{ye}$ via the engine coupling coefficients $h_{ey(j,k)}$ (pitch only), and also the angular accelerations of the aerosurfaces $\ddot{\delta}_s$ via the aerosurface coupling coefficients $h_{s(j,k)}$ as it is described in equation (2.7.4), ignoring the aero-elastic terms. The aircraft finite elements model must, therefore, include the flexibility of the aerosurfaces and engines, and the FEM is calculated with the gimbals “locked”. The hinges and gimbals are released in the equations of motion by the actuator rotations and the h-parameters as it is described in section 2.8 and in the actuators section 4.1, and implemented in the vehicle modeling program. The actuator model in this case does not include local structural stiffnesses since they are already included in the aircraft structure. The inertial coupling coefficients are also used to calculate the moments at the pivots of the engines T_{LY} and the aero surfaces M_{hs} due to local structural bending of the j^{th} mode ($\ddot{\eta}_j$), as it is described in equations (2.10.2 & 2.10.8). The purpose of this example is to demonstrate the flex modeling and analysis process. The analysis is similar to the rigid-body and we will compare the results with those obtained from the previous section.

2.1 Input Data for the Flexible Aircraft

The aircraft data used for generating this flexible vehicle model are in file “*Fighter-Flex.Inp*”, which is in directory “...*\Examples\Fighter Aircraft\Flex Analysis*”. The title of the flexible aircraft data set is: “*Fighter Aircraft Flex Model (47 Modes)*”. It is similar to the rigid-body data except that it includes flexibility. This aircraft dataset uses 47 flex modes which have already been pre-selected from the modal data file “*Fighter.Mod*” by the mode selection utility and have been converted from Nastran units and directions to units and directions compatible with the vehicle model. The h-parameters for the seven effectors are included in a separate “*Fighter.Gaf*” file. They are generated by the finite element modeling program and are in units of moment of inertia (slug-ft²). The flex coupling coefficients file should be in the format shown and have an extension (.Gaf). It does not include aero-elastic GAFD data as shown in other examples. It must be located in the project folder together with the other Flixan files. In this example the file “*Fighter.Gaf*” contains only flex coupling coefficients for the first 50 modes. It includes the mode frequencies and the h-parameters for the three aero-surfaces and the two engines in both pitch and yaw. The effectors are seven because it assumes that the engines are also gimbaling in yaw, although yaw gimbaling is not applicable in this case and the yaw coefficients are zero. To activate the inertial flex coupling option you must include the key words “**Flex Coupling Coeff**” in the flags line which is located near the top of the aircraft dataset, below the title and comment lines. The vehicle modeling program will select the column data from the h-parameters file that correspond to the selected mode frequencies. Otherwise, it will ignore the flex coupling coefficients and will excite vehicle flexibility at the hinges by the reaction forces and torques assuming that the effectors are rigid bodies rotating about the hinges. It will use the mode shapes and slopes at the hinges and gimbals.

```

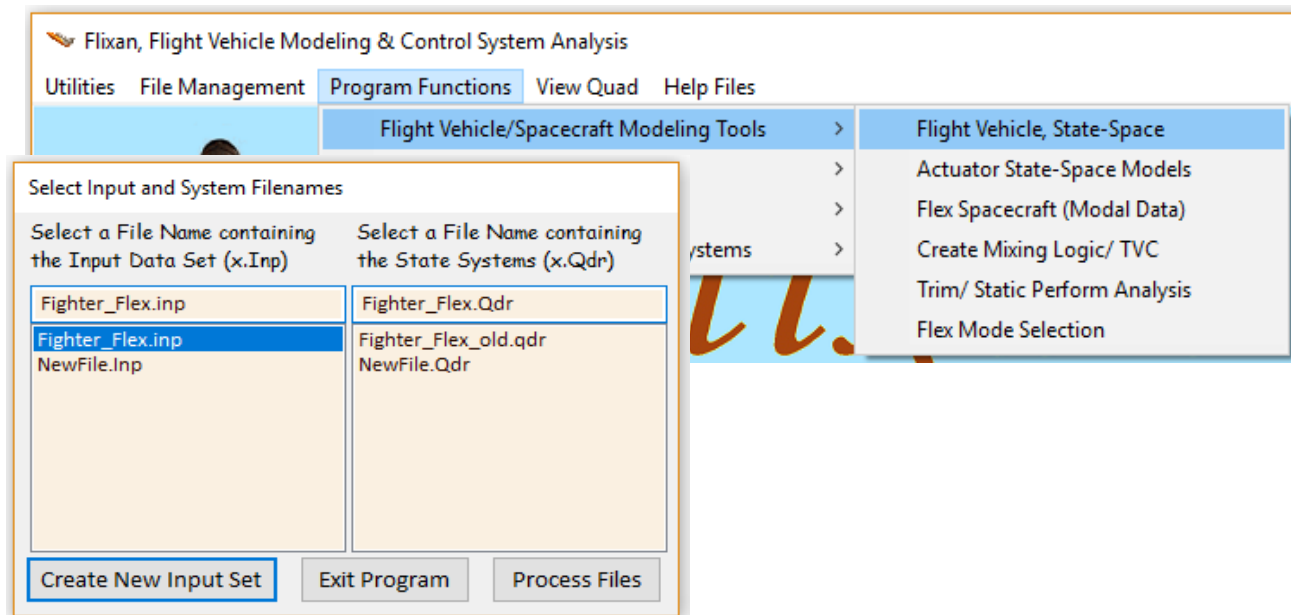
INERTIAL COUPLING PARAMETERS (H-param) in (lb-sec^2-ft)
Fighter Aircraft
! This file contains the Inertial Flex Coupling coefficients for the three control
! surfaces and the two engines of a fighter airplane. They are used
! to couple the effector motion with structure flex modes.
!
Number of: Flexible Modes, Control Surfaces, Engines=          50,   3,   2
Mode Frequency in (rad/sec)                                17.4380          17.8834          17.9020          17.9035          18.6519
Inertial Coupling (H-param) for Surface 1, Left Elevator, Modes (j=1,2,Nmode), ..... 1.5795E+00      -1.3297E+00      -1.1332E+00      2.1523E-01      -2.9742E-01
Inertial Coupling (H-param) for Surface 2, Right Elevator, Modes (j=1,2,Nmode), ..... 1.6310E+00      1.2482E+00      9.7070E-01      4.2701E-03      -5.3859E-01
Inertial Coupling (H-param) for Surface 3, Rudder, Modes (j=1,2,Nmode), ..... 5.2182E-02      -3.4909E+00      -7.9029E-01      -1.9075E-02      7.5942E-02
Inertial Coupling (H-param) for Left Engine # 1, (Pitch), Modes (j=1,2,Nmode), ..... 2.2931E+00      -1.1191E+00      -1.0033E+00      2.4997E-01      -4.8948E-01
Inertial Coupling (H-param) for Left Engine # 1, (Yaw), Modes (j=1,2,Nmode), ..... 0.0000E+00      0.0000E+00      0.0000E+00      0.0000E+00      0.0000E+00
Inertial Coupling (H-param) for Right Engine # 2, (Pitch), Modes (j=1,2,Nmode), ..... 2.2607E+00      9.7017E-01      7.4000E-01      6.9556E-02      -6.8222E-01
Inertial Coupling (H-param) for Left Engine # 2, (Yaw), Modes (j=1,2,Nmode), ..... 0.0000E+00      0.0000E+00      0.0000E+00      0.0000E+00      0.0000E+00

```

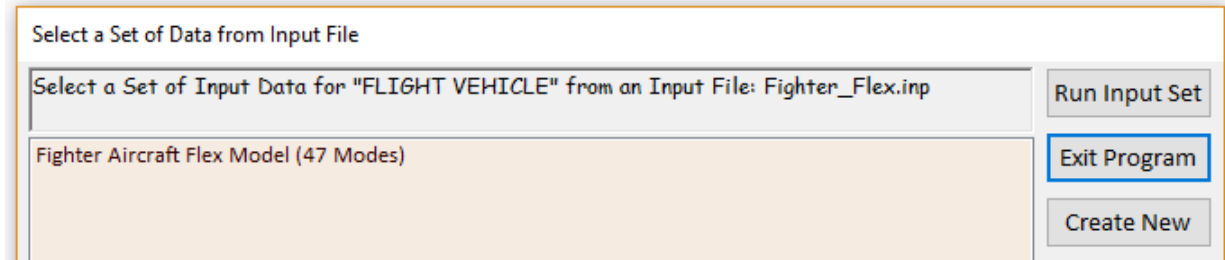
Near the bottom of the input data file “*Fighter_Flex.Inp*”, a set of selected mode frequencies and shapes is included at important vehicle locations. The title of the selected modes is “*Fighter Aircraft Flex Model, Modes from all axes*”. This title is also included in the last line of the vehicle data set, below the line “*Number of Modes = 47*”. This is how the program locates the selected modal data, since there may be more than one set of selected modes in the input data file. The selected set of modes was extracted from a bigger finite elements model file “*Fighter.Mod*” by a mode selection process. The modes preparation process also requires a nodes map file “*Fighter.Nod*” that contains a table of vehicle locations and the node numbers which are included in the modal data file. The nodes file is also saved in the “*Flex Analysis*” folder. It is used in the mode selection process to create menus for selecting vehicle locations that correspond to excitation and sensors locations. We will describe the mode selection and data preparation process in section 2.3, but for now let us assume that the aircraft dataset and the selected modes are already saved in the input data file “*Fighter_Flex.Inp*” and run the vehicle modeling program to create the flexible vehicle state-space model for analysis and simulations. We must also make sure that the h-parameters file “*Fighter.Gaf*” is also in the same folder.

2.2 Create the Flexible Vehicle Model

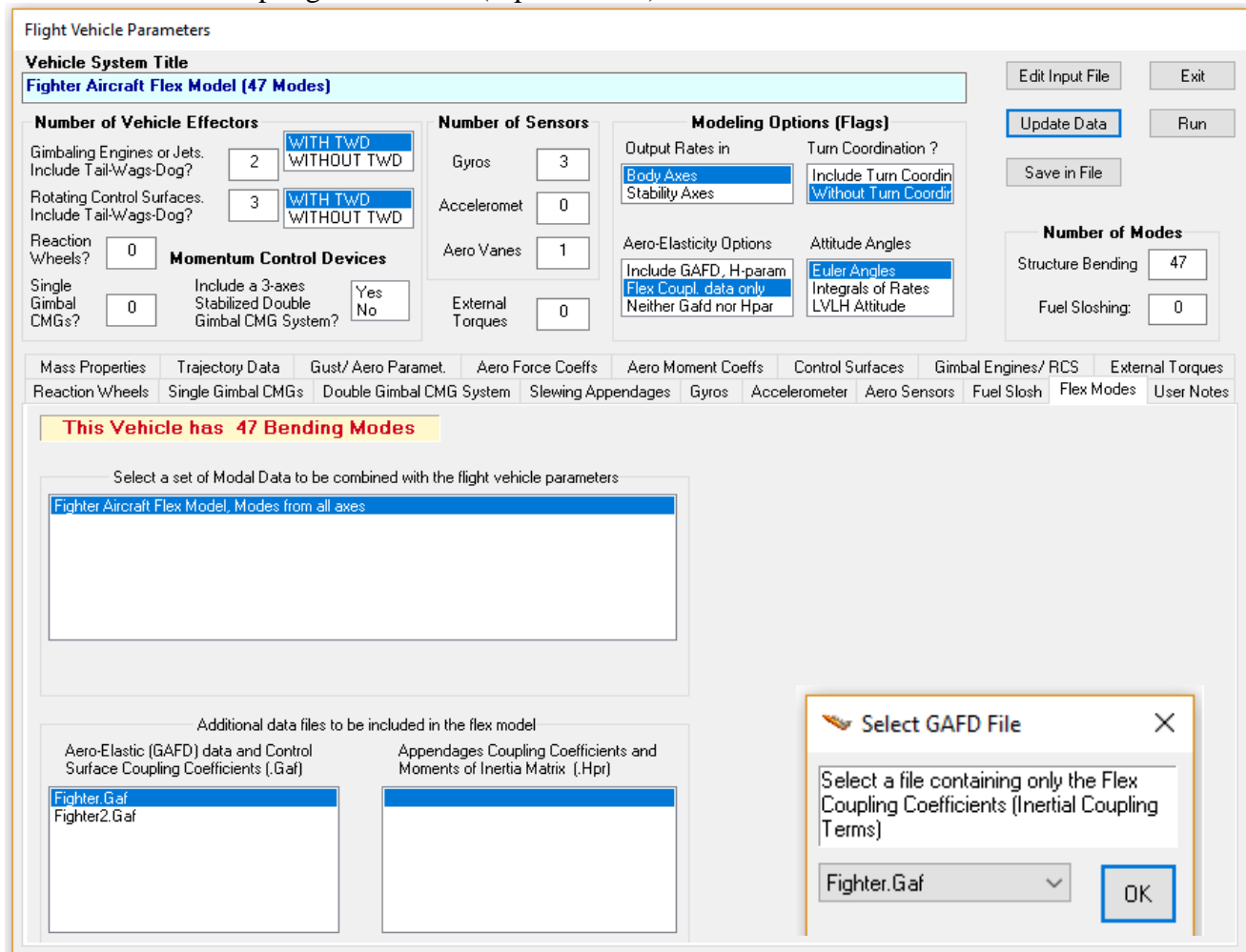
After describing the data files we are now ready to process the data by running the vehicle modeling program. Start the Flixan program, and select the directory: “\Flixan\ Flight\ Examples\Fighter Aircraft\ Flex Analysis”. From the Flixan main menu select “*Program Functions*”, “*Flight Vehicle/Spacecraft Modeling Tools*”, and select “*Flight Vehicle State-Space*”. From the filename selection menu select the input data file “*Fighter_Flex.Inp*”, the output systems file “*Fighter_Flex.Qdr*”, and click on “*Process Files*”.



From the flight vehicle data selection menu select the only title “*Fighter Aircraft Flex Model (47 Modes)*” and click on “*Run Input Set*”.



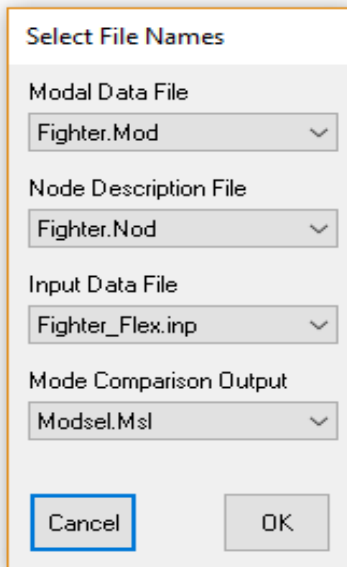
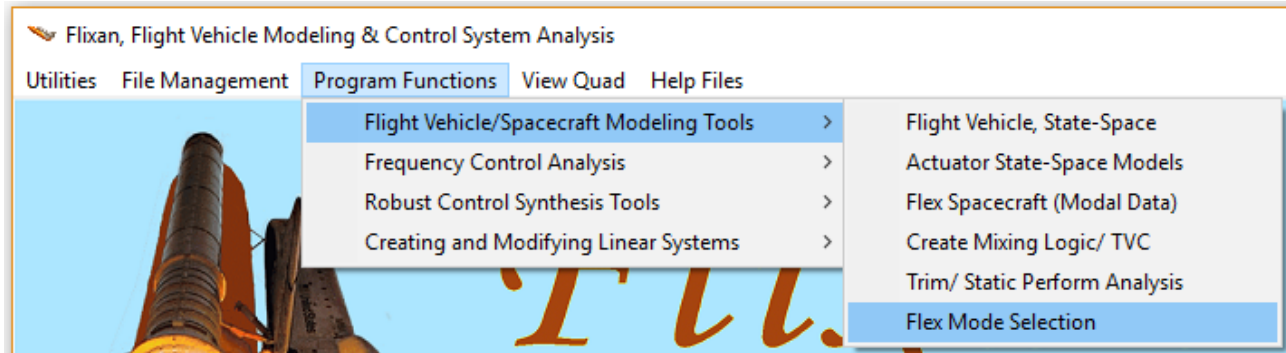
The following dialog comes up that has multiple tabs showing the various vehicle parameters. You may click on the various tabs to view the vehicle data in groups. Notice that the tail-wags-dog options are now turned on for both engines and 3 aerosurfaces. Also in the flag menus, the “Flex-Coupling data” flag is selected that requires the flex coupling coefficients file. Click on “Run” to process the vehicle data. Finally, using the small dialog below, select the file “*Fighter.Gaf*” which contains the flex coupling coefficients (h-parameters) and click “OK”.



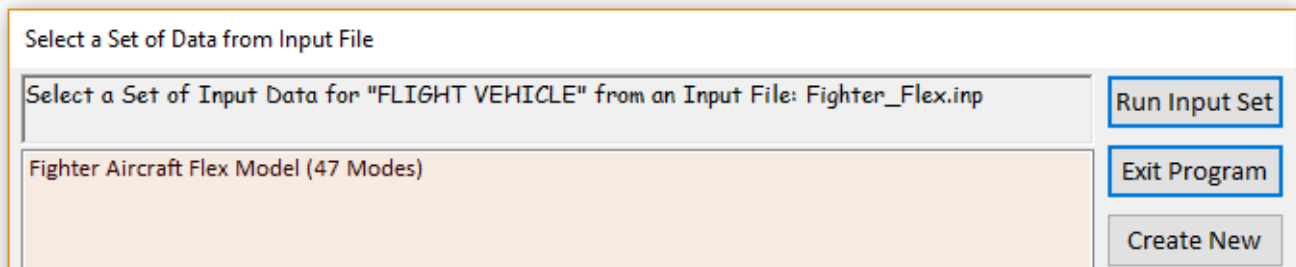
The program calculates the state-space model of the flexible aircraft using the 47 selected modes and saves it in the systems file “*Fighter_Flex.Qdr*”. Its title is “*Fighter Aircraft Flex Model (47 Modes)*” and the definitions of the system states, inputs, and outputs are also included below the system matrices

2.3 Mode Selection

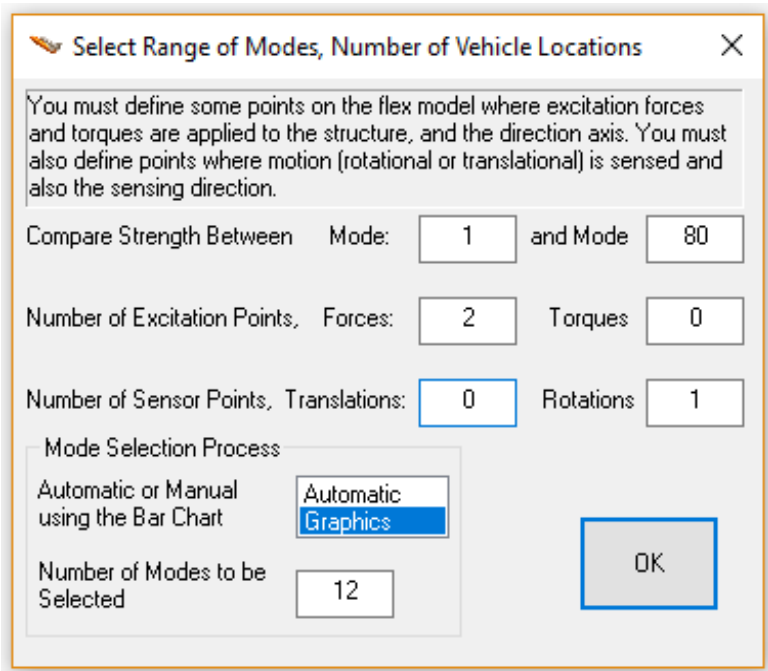
We will now go back to demonstrate the process of selecting the flex modes from the original Nastran generated modal data file “*Fighter.Mod*”. This file includes the aircraft mode frequencies and shapes for the first 80 modes. The file “*Fighter.Gaf*”, however, contains h-parameters data only for the first 50 modes. So it makes no sense to select any modes above mode 50. The selected modes for the corresponding vehicle locations will be rescaled, and saved in file “*Fighter_Flex.Inp*” as a set of selected modes with a title “*Fighter Aircraft Flex Model, Modes from all axes*”. The number of the selected modes in the dataset is 49 but we will only include the first 47 in the dynamic model because the last one 47th corresponds to mode 50. Since our h-parameters file is limited to the first 50 modes we try to select as many modes as possible between modes 1 and 50 and reject only three very weak ones. We will use the mode selection program to select dominant modes in all directions. Start the Flixan program and go to directory “...\\Examples\\Fighter Aircraft\\Flex Analysis”. Go to “Program Functions”, “Flight Vehicle/ Spacecraft Modeling”, and select “Flex Mode Selection”.



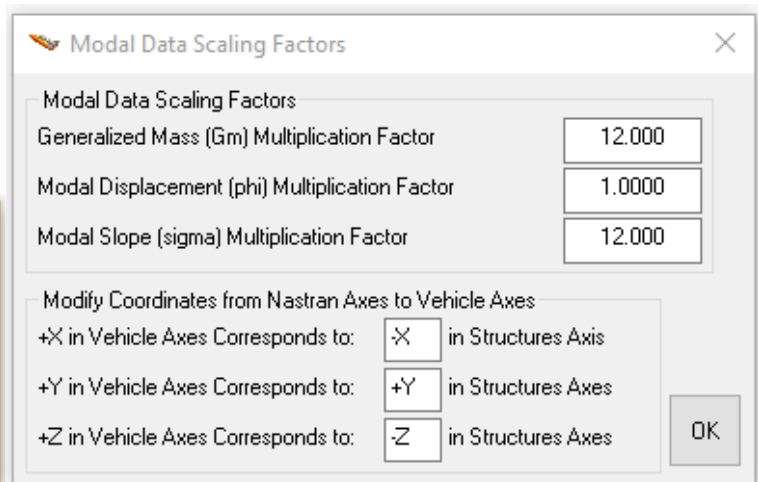
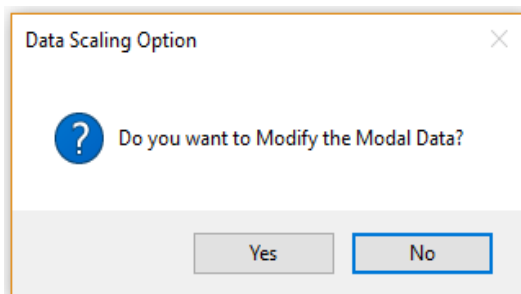
From the filename selection menu select the modal data file “*Fighter.Mod*”, the input data file “*Fighter_Flex.Inp*”, the nodes map file “*Fighter.Nod*”, and the default output file “*Modsel.Msl*” where the mode comparison data will be saved, and click “OK”. The nodes table is used in menus for selecting vehicle locations that correspond to force or torque excitation points and also for sensors locations. From the vehicle data selection menu below, select the title of the aircraft input data “*Fighter Aircraft Flex Model (47 Modes)*”, and click on “Run Input Set”. The mode selection program uses the aircraft data in order to match structural nodes in the finite elements model with important vehicle locations such as sensors, engine gimbals, and aero-surfaces.



Use the following dialog to specify a range of modes for modal strength comparison (1 to 80). You must also define the number of excitation points where we shall apply forces and torques, and the number of sensor points where we will be measuring translations and rotations for flex mode comparison. These could be any points (nodes) in the structural model and not necessarily nodes that correspond to the locations defined in the vehicle data file. In this case we specify 2 force excitation points and one rotational measurement point. The mode strengths will be calculated based on their contributions in the pitch direction but there will be plenty of modes chosen that contribute in all three directions.



The modal data are in Nastran units which are different from units used in the flight vehicle data. Also the body axes in the Nastran model are different from the directions of the body axes in our Flixan model. In the Nastran model the x-axis direction is towards the back of the vehicle and the z-axis is up. We must, therefore, scale the modal data and change the directions in some of the axes. In the next menu you must click on “Yes” to modify the data and in the next dialog you must enter the following scaling factors: Multiply the modal mass by 12 to convert it from “snails” to “slugs”. The mode shapes (ϕ) do not change. The slopes (σ) must also be scaled up by 12 because they must be changed from (rad/inch) to (rad/ft). The X and Z directions change signs (+X in flight control axes corresponds to -X in the Nastran model).



In the following menus we must select the locations and directions of the two force excitations 1 & 2 that were specified in the previous dialog. The 2 force points are at the hinges of the left and right elevators (nodes 8 & 5) and both forces are applied in the +z direction. The menu below shows how to select the first force at the left elevator. After selecting, click “OK”, and do the same for the right elevator force at node 5.

Table of Vehicle Structure FEM Nodes

In mode selection, in order to calculate the relative mode strength of a number of modes in a specified direction you must define some node points in the Nastran model where the excitation forces or torques will be applied and also the forcing directions.

Similarly, you must also define the sensor points (translations or rotations) and the sensing directions.

Select a Location (Node) for Force Excitation : 1

Sensor-1	1	2169	8.4157	-0.871
Sensor-2	2	2173	8.4157	0.871
Between Engines	3	2077	2.2	0.000
Right Engine Gimbal	4	1571	2.1000	2.199
Right Elevator Hinge Line	5	1576	5.2000	7.400
Right Elevator Center	6	1818	3.2986	8.100
Left Engine Gimbal	7	1721	2.1000	-2.199
Left Elevator Hinge Line	8	1726	5.2000	-7.400
Left Elevator Center	9	1839	3.2986	-8.100
Crew Compartment Acceler 1	10	307	43.9911	0.332
Crew Compartment Acceler 2	11	308	43.9911	-0.332
Gyros/ Accelerometers	12	315	24.4000	0.000
Vane Measurem (alpha,beta)	13	1490	45.0007	0.000
Rudder Lower Hinge	14	2376	1.2000	0.000
Rudder Upper Hinge	15	2379	-0.8000	0.000

OK

Cancel

Axis

Along X
Along Y
Along Z

Direction

+ (positive)
- (negative)

We must also specify the location and the measuring direction for the rotational sensor that was specified earlier that will be used in mode comparison and selection. We select the gyros and accelerometers node #12 measuring in the +pitch direction, and click “OK”.

Table of Vehicle Structure FEM Nodes

In mode selection, in order to calculate the relative mode strength of a number of modes in a specified direction you must define some node points in the Nastran model where the excitation forces or torques will be applied and also the forcing directions.

Similarly, you must also define the sensor points (translations or rotations) and the sensing directions.

Select a Location (Node) for Rotational Sensor: 1

Sensor-1	1	2169	8.4157	-0.871
Sensor-2	2	2173	8.4157	0.871
Between Engines	3	2077	2.2	0.000
Right Engine Gimbal	4	1571	2.1000	2.199
Right Elevator Hinge Line	5	1576	5.2000	7.400
Right Elevator Center	6	1818	3.2986	8.100
Left Engine Gimbal	7	1721	2.1000	-2.199
Left Elevator Hinge Line	8	1726	5.2000	-7.400
Left Elevator Center	9	1839	3.2986	-8.100
Crew Compartment Acceler 1	10	307	43.9911	0.332
Crew Compartment Acceler 2	11	308	43.9911	-0.332
Gyros/ Accelerometers	12	315	24.4000	0.000
Vane Measurem (alpha,beta)	13	1490	45.0007	0.000
Rudder Lower Hinge	14	2376	1.2000	0.000
Rudder Upper Hinge	15	2379	-0.8000	0.000

OK

Cancel

Axis

Roll
Pitch
Yaw

Direction

+ (positive)
- (negative)

Now that we have specified the excitation and measurement points for mode comparison, we must also identify which structural nodes correspond to important aircraft locations. The locations are defined in the vehicle input dataset, that is, aerosurface hinges, engine gimbals, and sensors. In the next three menus the user is asked to select locations for the 3 aerosurfaces, (left and right elevators, and rudder). The menu below shows the selection of the left elevator hinge (node #8) which corresponds to aero-surface #1 in the input data. Click “OK” and repeat for the other two control surfaces: the right elevator hinge (node #5) which corresponds to aerosurface #2, and the Rudder Upper Hinge (node #15) which corresponds to aerosurface #3. The rudder has two structural nodes allocated because it is a long aerosurface.

Table of Vehicle Structure FEM Nodes [X]

You must now identify some points on the finite element model that correspond to the important locations on the vehicle (as specified in the vehicle data) where the forces are being applied and the motion is being sensed. Such as TVC gimbals, gyros, etc.

Select a Location (Node) for Control Surface : 1 [OK]

Sensor-1	1	2169	8.4157	-0.8715
Sensor-2	2	2173	8.4157	0.8715
Between Engines	3	2077	2.2	0.0000
Right Engine Gimbal	4	1571	2.1000	2.1997
Right Elevator Hinge Line	5	1576	5.2000	7.4000
Right Elevator Center	6	1818	3.2986	8.1000
Left Engine Gimbal	7	1721	2.1000	-2.1997
Left Elevator Hinge Line	8	1726	5.2000	-7.4000
Left Elevator Center	9	1839	3.2986	-8.1000
Crew Compartment Acceler 1	10	307	43.9911	0.3320
Crew Compartment Acceler 2	11	308	43.9911	-0.3320
Gyros/ Accelerometers	12	315	24.4000	0.0000
Vane Measurem (alpha,beta)	13	1490	45.0007	0.0000
Rudder Lower Hinge	14	2376	1.2000	0.0000
Rudder Upper Hinge	15	2379	-0.8000	0.0000

Table of Vehicle Structure FEM Nodes [X]

You must now identify some points on the finite element model that correspond to the important locations on the vehicle (as specified in the vehicle data) where the forces are being applied and the motion is being sensed. Such as TVC gimbals, gyros, etc.

Select a Location (Node) for Control Surface : 3 [OK]

Sensor-1	1	2169	8.4157	-0.8715
Sensor-2	2	2173	8.4157	0.8715
Between Engines	3	2077	2.2	0.0000
Right Engine Gimbal	4	1571	2.1000	2.1997
Right Elevator Hinge Line	5	1576	5.2000	7.4000
Right Elevator Center	6	1818	3.2986	8.1000
Left Engine Gimbal	7	1721	2.1000	-2.1997
Left Elevator Hinge Line	8	1726	5.2000	-7.4000
Left Elevator Center	9	1839	3.2986	-8.1000
Crew Compartment Acceler 1	10	307	43.9911	0.3320
Crew Compartment Acceler 2	11	308	43.9911	-0.3320
Gyros/ Accelerometers	12	315	24.4000	0.0000
Vane Measurem (alpha,beta)	13	1490	45.0007	0.0000
Rudder Lower Hinge	14	2376	1.2000	0.0000
Rudder Upper Hinge	15	2379	-0.8000	0.0000

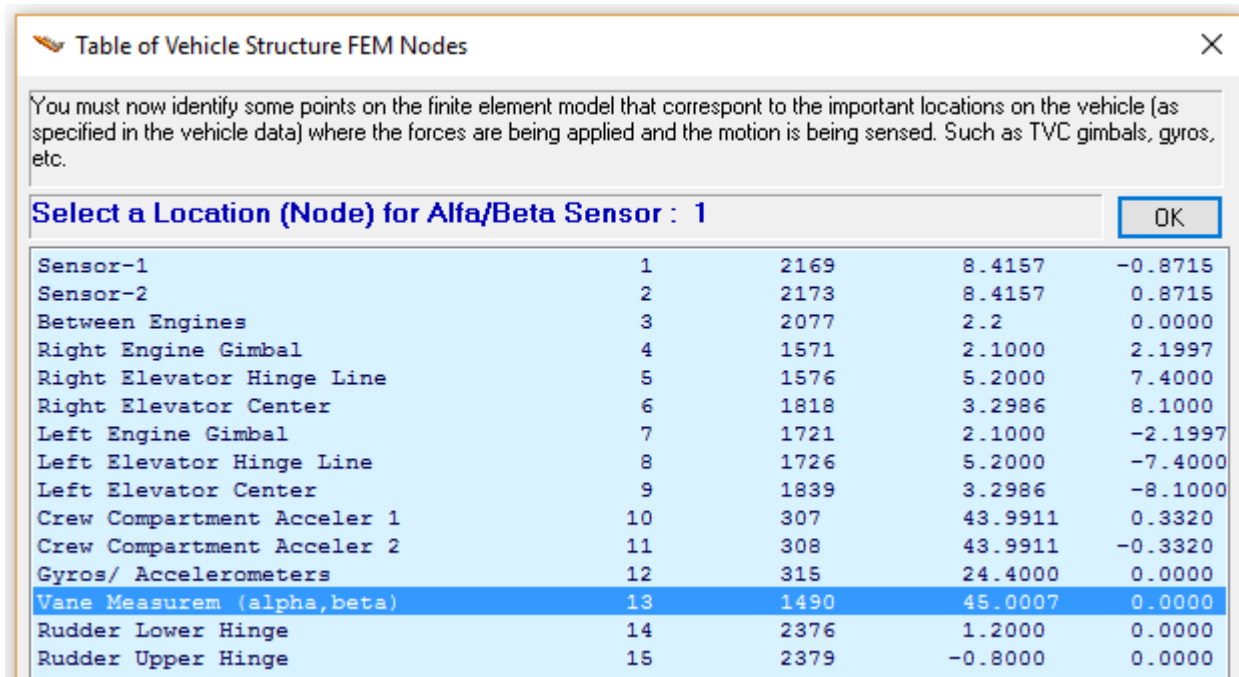
In the next two menus we must select nodes (#7 and #4) for the two gimbaling engines (left and right engines). Chose the left engine gimbal (node #7), that corresponds to the left engine #1 in the vehicle data, and then chose the right engine gimbal (node #4) for the right engine #2.

Node Name	ID	X	Y	Z
Sensor-1	1	2169	8.4157	-0.8715
Sensor-2	2	2173	8.4157	0.8715
Between Engines	3	2077	2.2	0.0000
Right Engine Gimbal	4	1571	2.1000	2.1997
Right Elevator Hinge Line	5	1576	5.2000	7.4000
Right Elevator Center	6	1818	3.2986	8.1000
Left Engine Gimbal	7	1721	2.1000	-2.1997
Left Elevator Hinge Line	8	1726	5.2000	-7.4000
Left Elevator Center	9	1839	3.2986	-8.1000
Crew Compartment Acceler 1	10	307	43.9911	0.3320
Crew Compartment Acceler 2	11	308	43.9911	-0.3320
Gyros/ Accelerometers	12	315	24.4000	0.0000
Vane Measurem (alpha,beta)	13	1490	45.0007	0.0000
Rudder Lower Hinge	14	2376	1.2000	0.0000
Rudder Upper Hinge	15	2379	-0.8000	0.0000

We must also select a node for the gyros/ accelerometer locations (node #12) three times, for the three (roll, pitch, and yaw) rate gyros specified in the vehicle data.

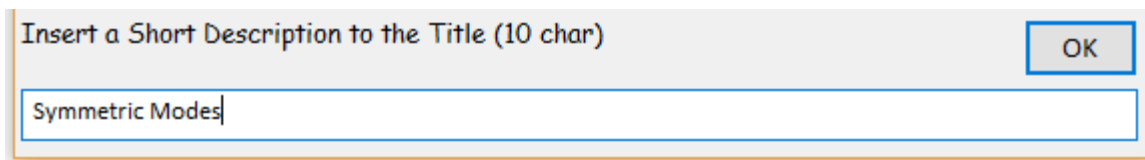
Node Name	ID	X	Y	Z
Sensor-1	1	2169	8.4157	-0.8715
Sensor-2	2	2173	8.4157	0.8715
Between Engines	3	2077	2.2	0.0000
Right Engine Gimbal	4	1571	2.1000	2.1997
Right Elevator Hinge Line	5	1576	5.2000	7.4000
Right Elevator Center	6	1818	3.2986	8.1000
Left Engine Gimbal	7	1721	2.1000	-2.1997
Left Elevator Hinge Line	8	1726	5.2000	-7.4000
Left Elevator Center	9	1839	3.2986	-8.1000
Crew Compartment Acceler 1	10	307	43.9911	0.3320
Crew Compartment Acceler 2	11	308	43.9911	-0.3320
Gyros/ Accelerometers	12	315	24.4000	0.0000
Vane Measurem (alpha,beta)	13	1490	45.0007	0.0000
Rudder Lower Hinge	14	2376	1.2000	0.0000
Rudder Upper Hinge	15	2379	-0.8000	0.0000

We also select (node #13) for (alpha, beta) air data sensor which is also specified in the vehicle model, and finally a disturbance point that will not be used in the model.

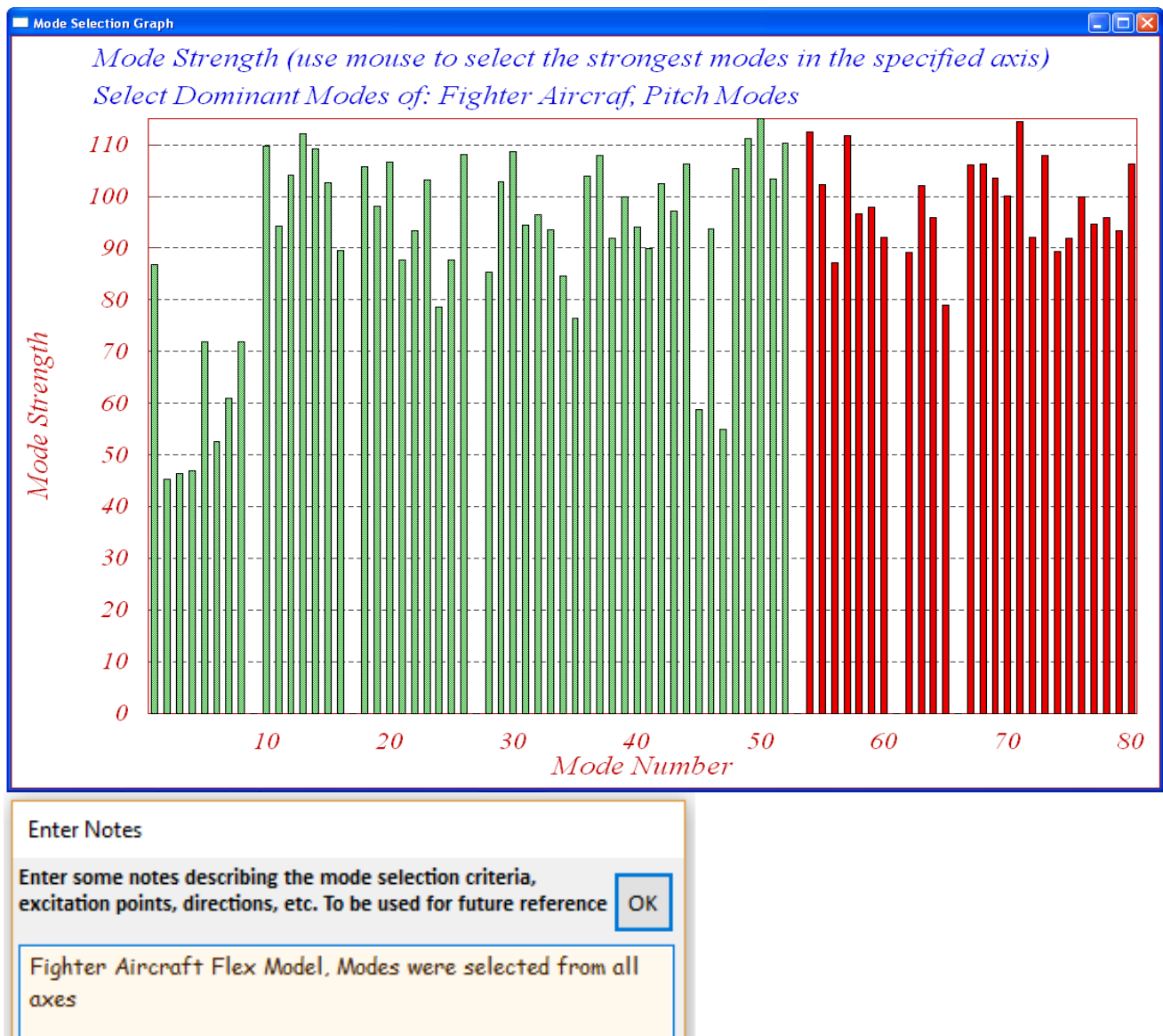


The mode selection program reads the mode shapes and slopes at the selected locations and performs a mode strength comparison calculated between the excitation points and directions to the sensor points and directions. When the comparison is complete the program saves the relative mode strength of each mode in file “Modsel.Dat” and displays a mode strength comparison bar chart, shown below for the pitch axis. The bar chart plots the relative mode strength of each mode versus the mode number and all bars are initially red before selection. The mode selection is performed interactively by the user by clicking with the mouse on the 49 modes to be selected from the bar-chart. Mode numbers (9, 17, & 27) were not included because they are weak. Mode numbers greater than 52 were also excluded. When we click with the mouse on the modes to be selected, the color of the mode bars change from red to green after selection. Press “Enter” when the mode selection is complete.

A short description of the selected modal data is required for documentation purposes. You may enter information, such as, conditions, excitation and sensor locations, directions, etc. It will be included in the title of the selected modes set, for example:



The selected modes are finally saved in file “Fighter_Flex.Inp”. The default title of the modal set was modified to: “Fighter Aircraft Flex Model, Modes from all axes”. The last two modes (51 and 52) are not included in the flex state-space model because we have flex coupling coefficients only for the first 50 modes. The user may finally enter some comments in the dialog below that will be included in the selected modal data set, below the title.



2.4 Modified Flex Aircraft Model

The flex vehicle model “*Fighter Aircraft Flex Model (47 Modes)*” is missing an important output that will be needed for feedback in the velocity control loop. It is the change in velocity δV relative to trim velocity V_0 . This variable, however, is included in the states (state-10). We will use the Flixan system modification program, as before, to modify the aircraft model and include state-10 in the output vector. The system modification dataset is in file “*Fighter_Flex.inp*” and its title is “*Fighter Aircraft Flex Simulation Model (47 Modes)*”. The modified aircraft system will be saved in the systems file “*Fighter_Flex.Qdr*” and it will include one additional output #25 representing the aircraft velocity variation δV from trim V_0 .

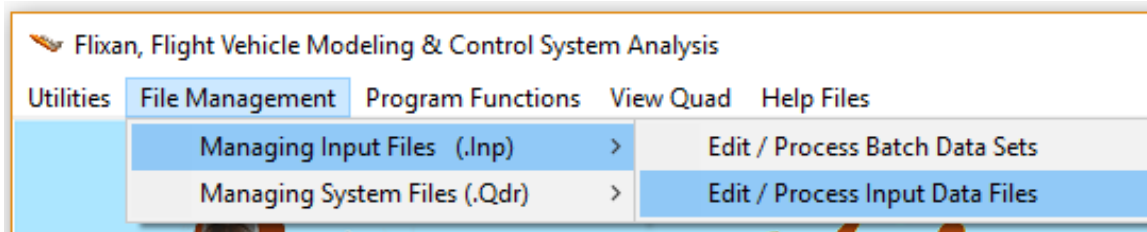
2.5 Actuator Models

The aerosurface and TVC actuator models used for flex analysis are the same model types as the actuators used in the rigid-body analysis but they have different stiffnesses. The backup and load stiffnesses are increased by a factor of 10 at least because those stiffnesses are now included in the

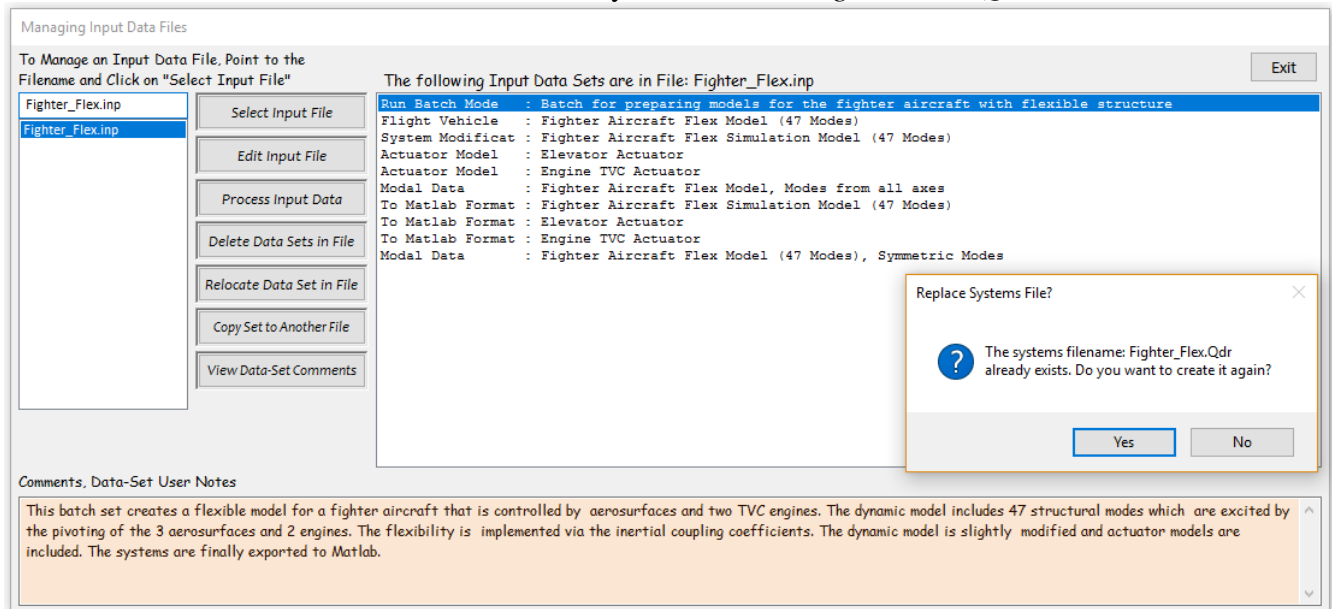
finite elements model. In the rigid-body analysis the backup and load stiffnesses were included in the actuator subsystems in order to capture flexibility at the actuator/ vehicle attachments. The actuator datasets for the aerosurfaces and the TVC engines are already saved in file “*Fighter_Flex.Inp*”. Their titles are “*Elevator Actuator*” and “*Engine TVC Actuator*”.

2.6 Processing the Dataset in Batch Mode and Exporting the Models to Matlab

The datasets in file “*Fighter_Flex.Inp*” can be processed interactively as it was shown in section 1. In this section, however, we will process the file in batch mode by executing the batch set which is located at the top of the file. The title of the batch set is “*Batch for preparing models for the fighter aircraft with flexible structure*”. It will process the aircraft data, modify the model, generate the actuators and save the systems in file “*Fighter_Flex.Qdr*”. It will also convert and export those systems from the systems file and be transferred to the Matlab analysis folder “*\Fighter Aircraft\Flex Analysis\Mat_Flex*”. The batch can be processed from the File Manager utility by selecting “*Edit/ Process Input Data Files*”, as shown below.



The input file manager dialog comes up, and from the menu on the left side select the input file “*Fighter_Flex.Inp*” and click on “*Select Input File*”. The menu on the right shows the titles of the datasets that are saved in this input file. Select the first one which is the batch set and click on “*Process Input Data*”. Click on “*Yes*” to overwrite the previous data. The batch will then process the datasets which are in this file and save the systems in file “*Fighter_Flex.Qdr*”.

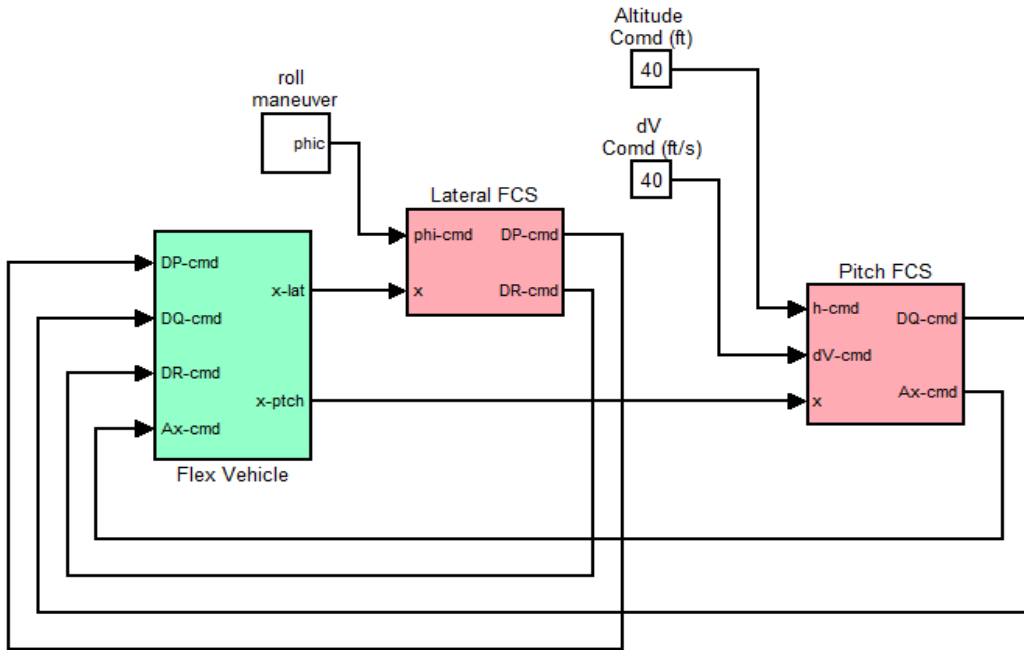


It will also save the modified aircraft model as a Matlab function m-file “*vehicle_flex.m*” and the two actuators in m-files “*elevator.m*” and “*engine_tvc.m*”. These systems will be saved in the Matlab analysis subdirectory “*Mat_Flex*” and they will be loaded into Matlab for analysis.

2.7 Flex Simulation

The flexible aircraft simulation is similar to the rigid-body and it is performed in Matlab subdirectory “*Mat_Flex*”. The Simulink model is in file “*Flex_Sim.mdl*” shown below. The flex vehicle dynamics block is similar to the rigid-body block. It includes the flexible aircraft state-space system “*Vehicle_flex.m*” that contains 47 flex modes as already described. The vehicle model, actuators, and mixing logic matrix are loaded into the Matlab workspace by running the m-file “*run.m*”. The simulation consists of both, pitch and lateral dynamics and four control loops which are closed via the mixing-logic matrix. In the longitudinal control system the commands are changes in the aircraft altitude and velocity relative trim altitude and velocity. In the lateral axes the command is roll attitude.

Coupled Axes Flex Vehicle Simulation Model



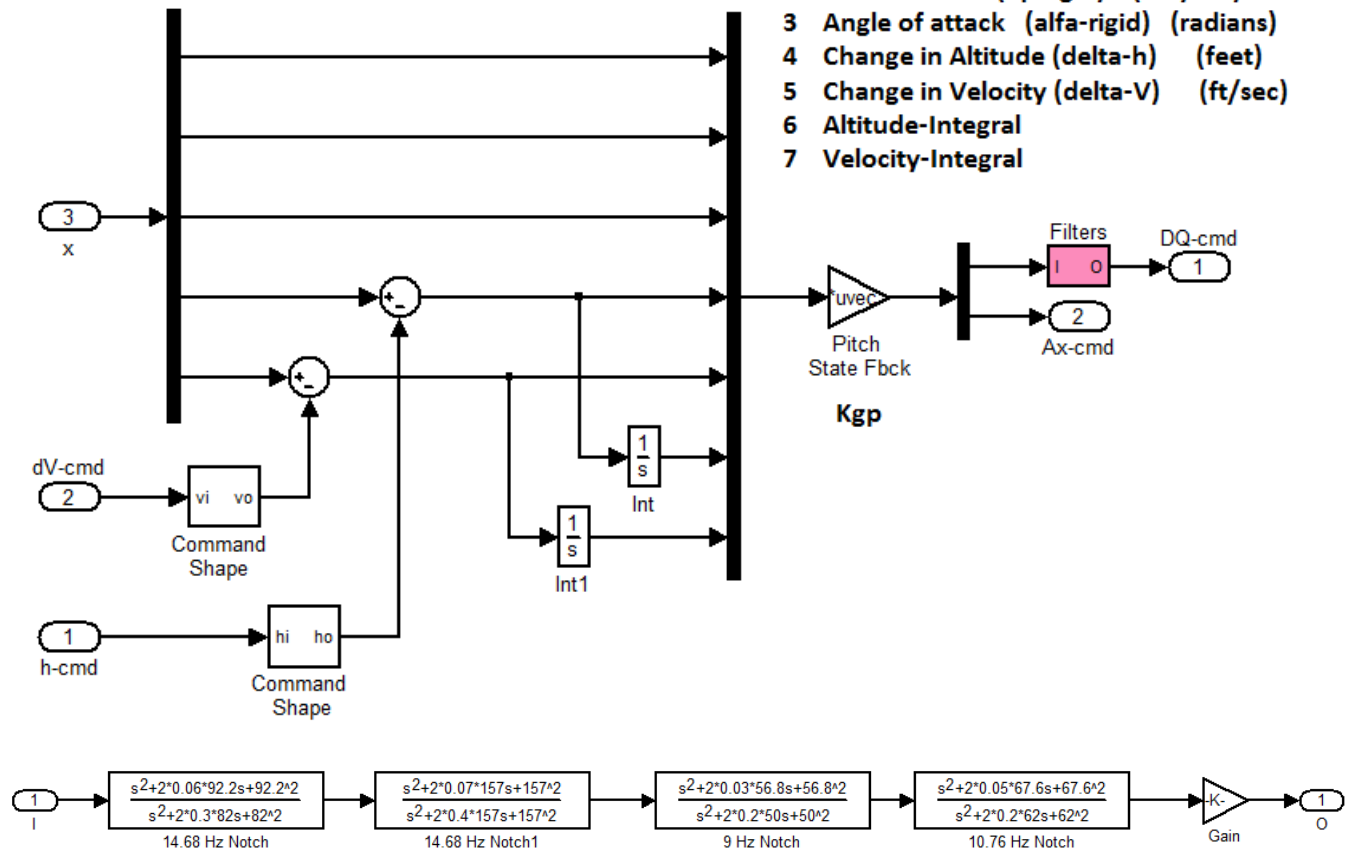
The three rotational and one axial acceleration flight control commands $(DP, DQ, DR, Ax)_{cmd}$ are converted by the mixing logic matrix K_{mix4} into control surface deflection commands, engine gimbal deflection commands, and also thrust variation commands for the two 30,000 (lb) engines. The commands drive the actuators and become deflections and throttle inputs that control the aircraft. The simulation model includes a wind-gust velocity disturbance that excites the aircraft dynamics. The wind direction is defined in the vehicle data, and it is towards the vehicle, perpendicular to the x axis, and at 45° between the $+Z$ and the $+Y$ axes. The mechanical feedback loops between the vehicle hinge moment outputs and the actuators represent the actuator loading due to vehicle acceleration at the engines and the control surfaces.

The pitch and lateral state-feedback matrices K_{gp} and K_{gl} were derived in the previous section using the LQR method and the rigid aircraft models. The flight control systems will now include notch and low-pass filters to attenuate structural flexibility in the pitch and yaw axes, as shown below. We will use the above Simulink model to repeat the two aircraft simulations that were presented in the previous rigid-body analysis, including flexibility.

Pitch Flight Control System

States = 7

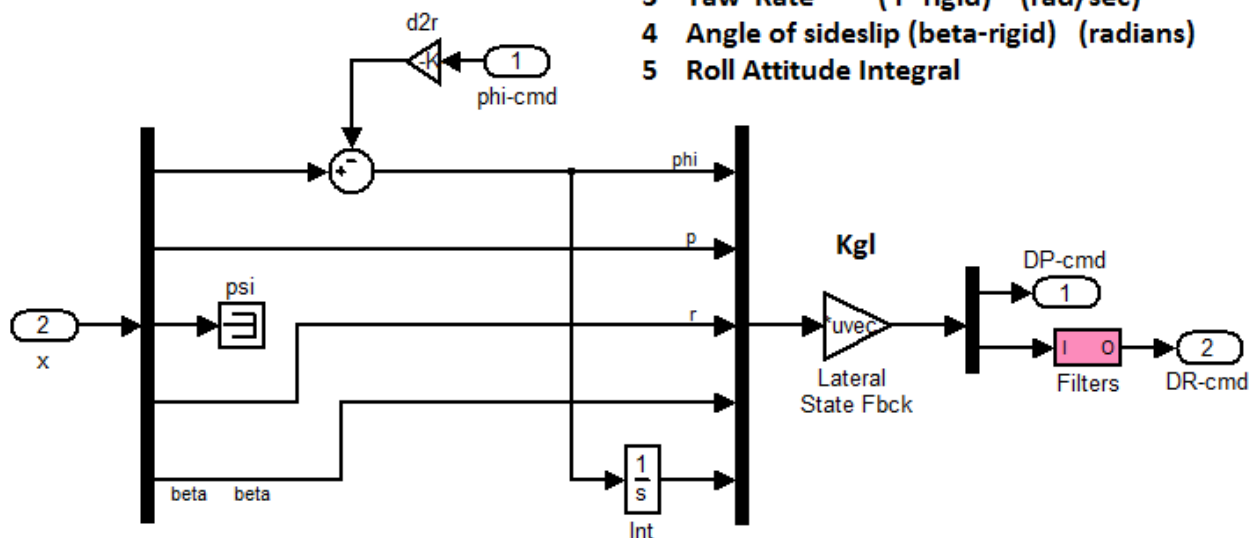
- 1 Pitch Attitude (theta-rigid) (radians)
- 2 Pitch Rate (q-rigid) (rad/sec)
- 3 Angle of attack (alfa-rigid) (radians)
- 4 Change in Altitude (delta-h) (feet)
- 5 Change in Velocity (delta-V) (ft/sec)
- 6 Altitude-Integral
- 7 Velocity-Integral



Lateral Flight Control

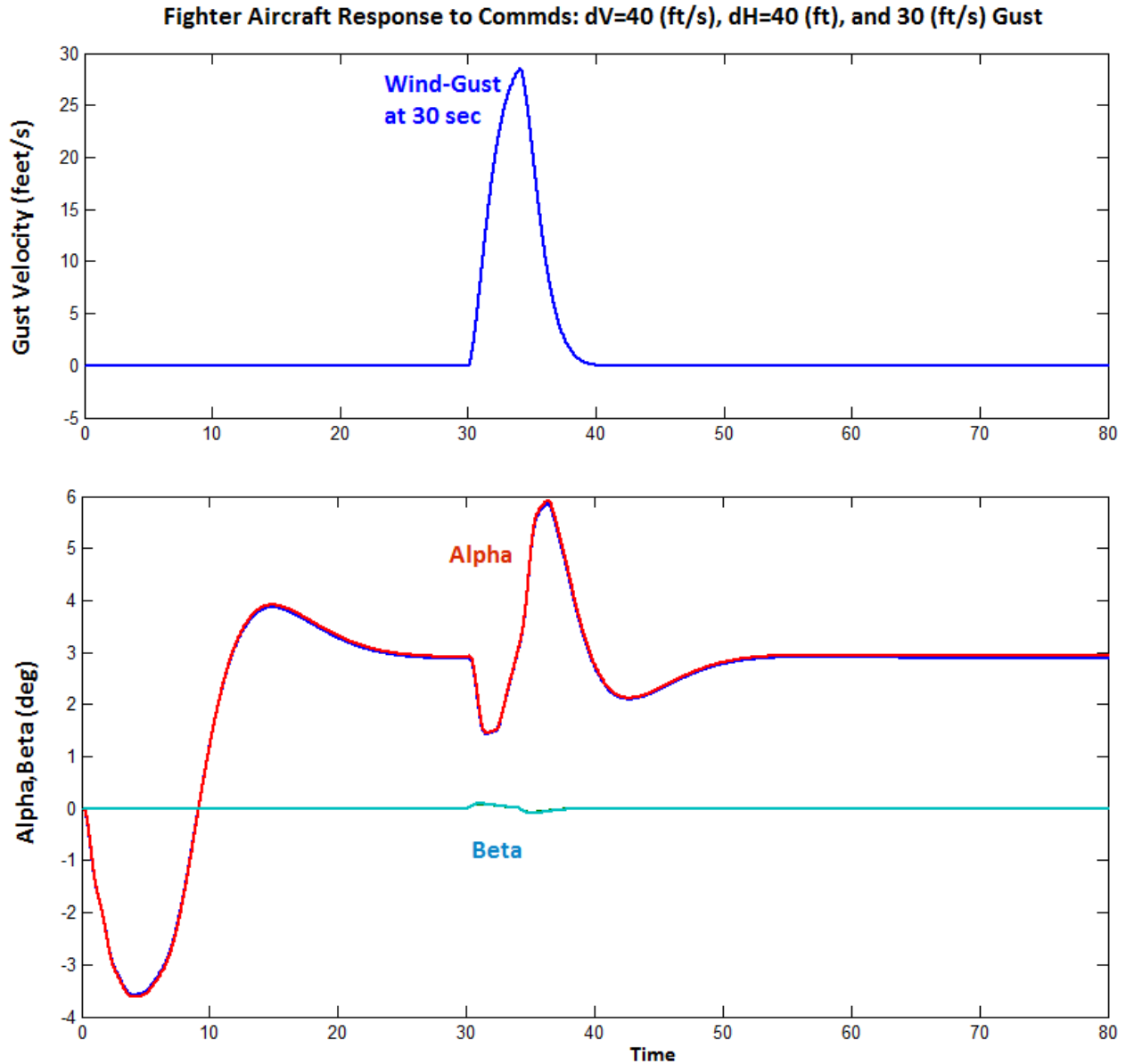
States/ Outputs

- 1 Roll Attitude (phi-rigid) (radians)
- 2 Roll Rate (p-rigid) (rad/sec)
- 3 Yaw Rate (r-rigid) (rad/sec)
- 4 Angle of sideslip (beta-rigid) (radians)
- 5 Roll Attitude Integral

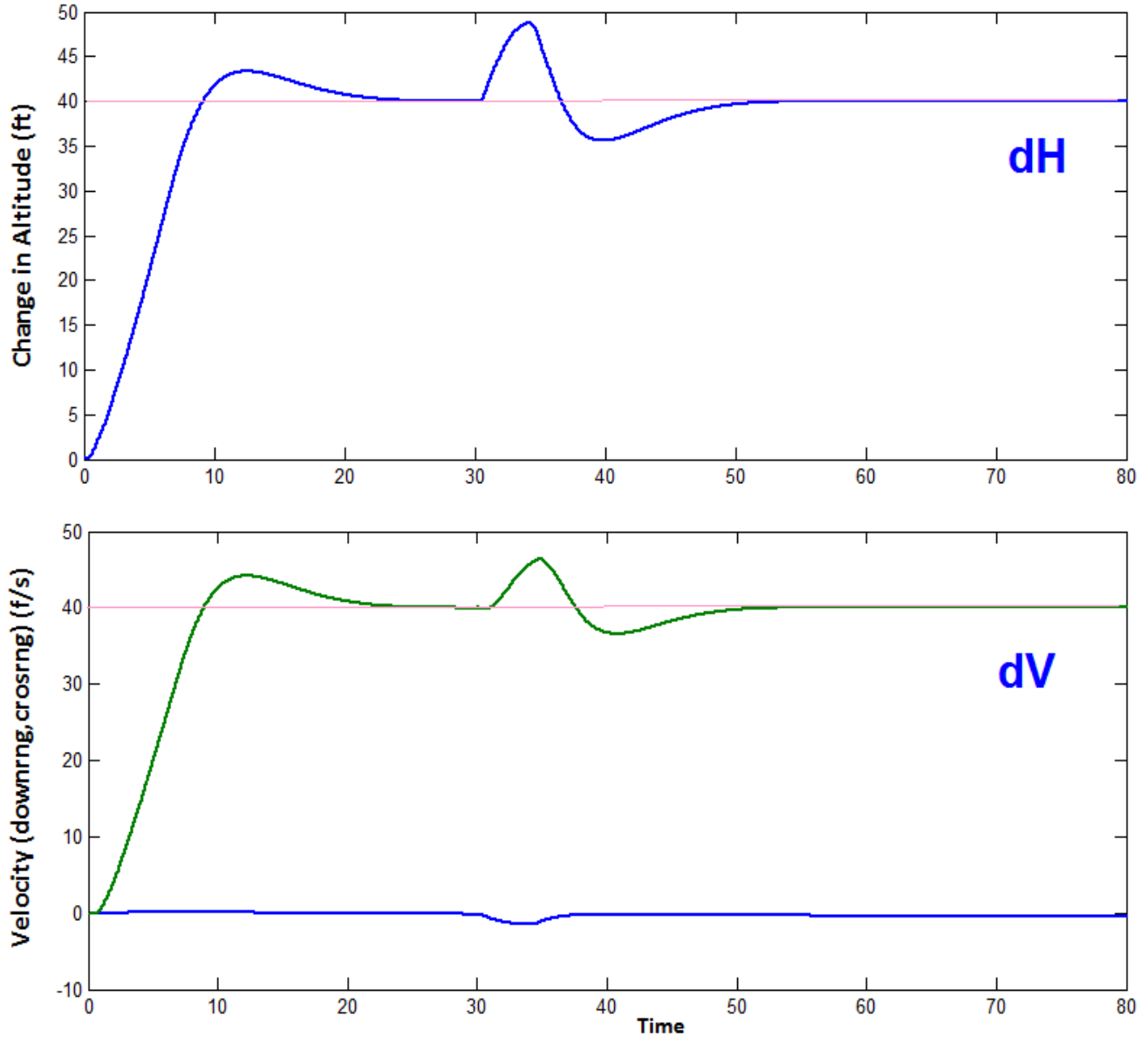


Pitch Maneuver

The first simulation executes a longitudinal maneuver where the vehicle is simultaneously commanded to increase its speed ($\delta V=40$ feet/sec), and also raise its altitude to ($\delta H=40$ feet). A 30 (feet/sec) wind-gust disturbance is also applied at 30 seconds. When the simulation is complete a Matlab script file "Pl.m" is used to plot the data. The figures show the vehicle responses during the maneuver. The aircraft altitude and velocity respond to the input commands. The wind-gust causes a disturbance transient which eventually decays.

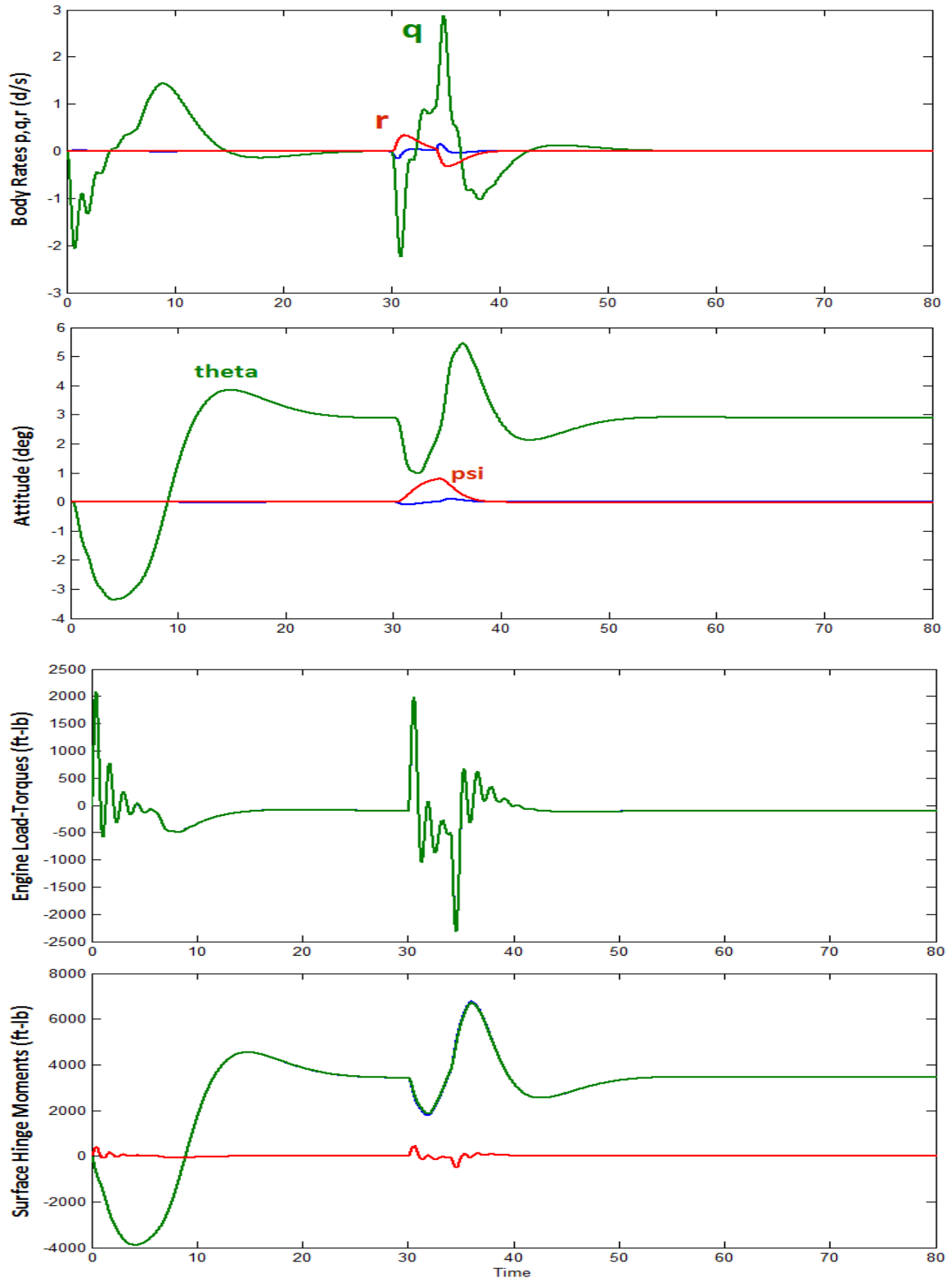


Fighter Aircraft Response to Commds: $dV=40$ (ft/s), $dH=40$ (ft), and 30 (ft/s) Gust

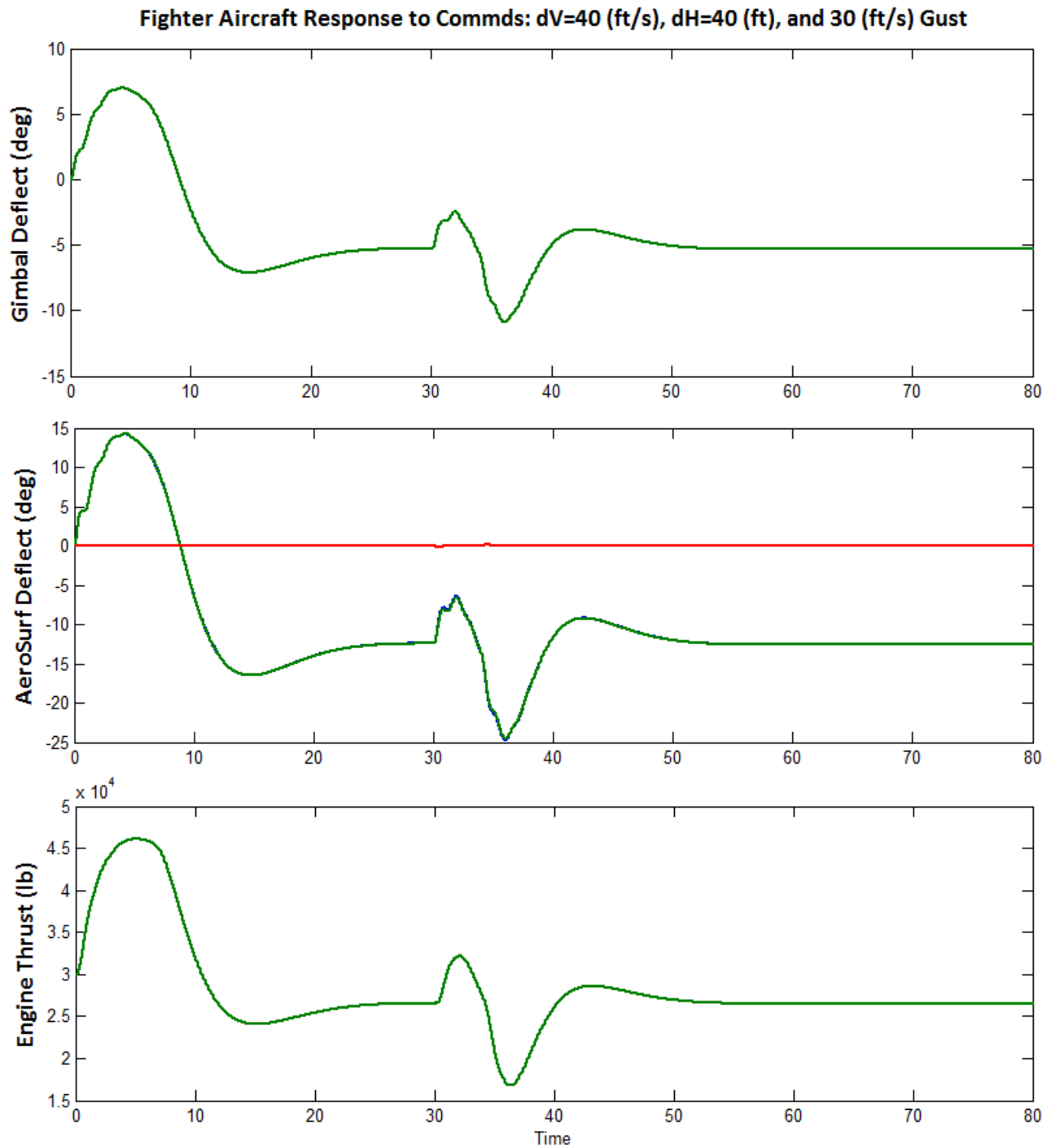


The next figure shows the aircraft body rates and attitude. The motion is mainly in pitch (green). The wind-gust also causes a small transient in the lateral direction (red and blue). The next plot shows the load-torques at the two engine gimbals, and the hinge moments at the three aerosurfaces which are mostly due to the aerodynamic pressure.

Fighter Aircraft Response to Commds: $dV=40$ (ft/s), $dH=40$ (ft), and 30 (ft/s) Gust



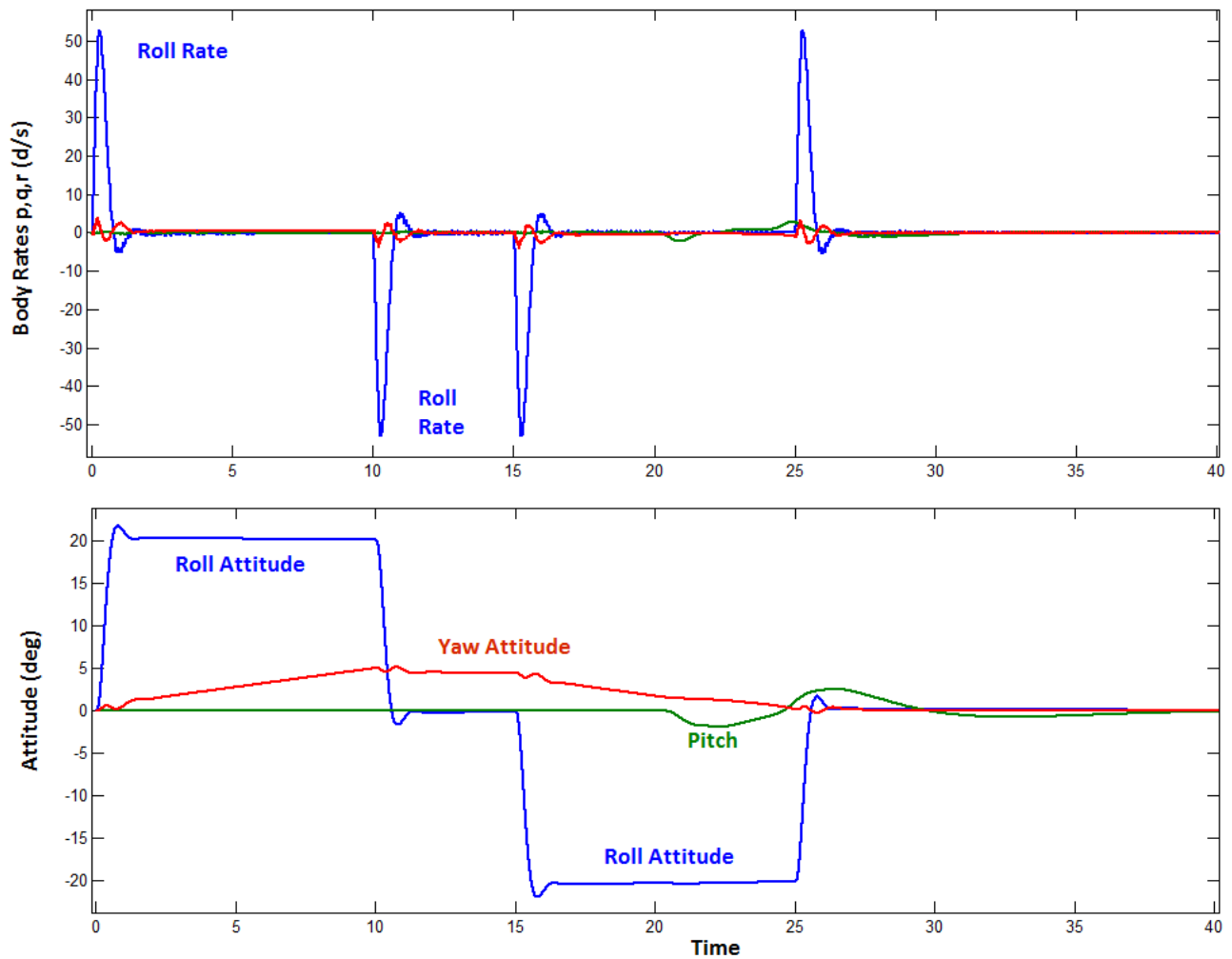
The figure below shows the pitch deflections of the aircraft engines, the aerosurface deflections, and the thrust variations which are required to perform the altitude and velocity change maneuver, and also to attenuate the transient caused by the gust. The rudder deflection (red) is almost zero. It shows also the engine thrusts which start at 30,000 (lb) are temporarily increased to 46,000 (lb) in order to raise the altitude and speed, and they eventually drop to 26,000 (lb). The wind-gust at 30 seconds causes transients in all effectors as they respond to the wind disturbance. The results are almost identical to the rigid-body simulation results.



Lateral Maneuver

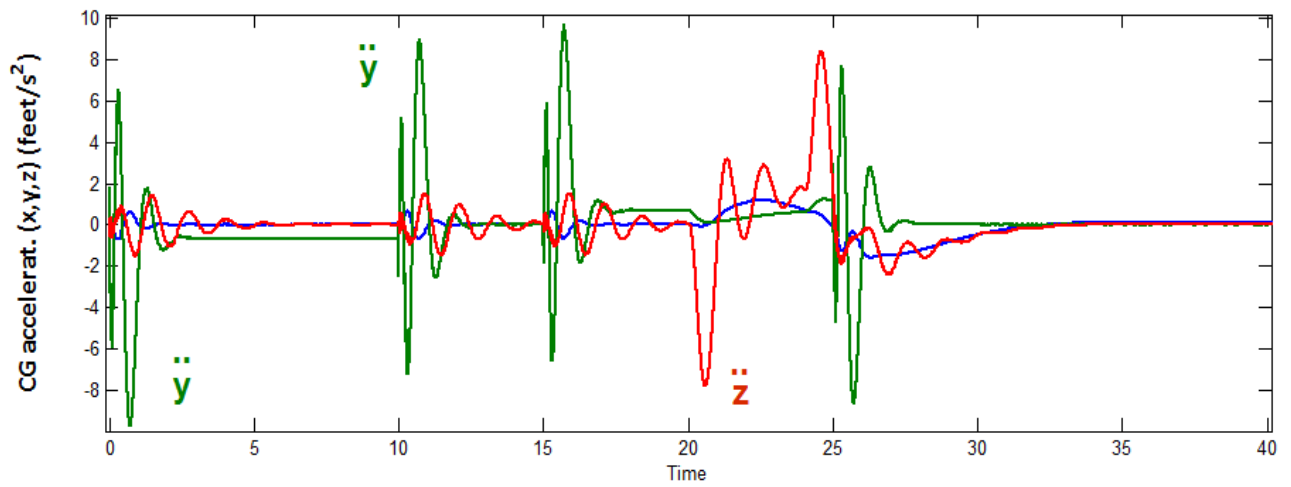
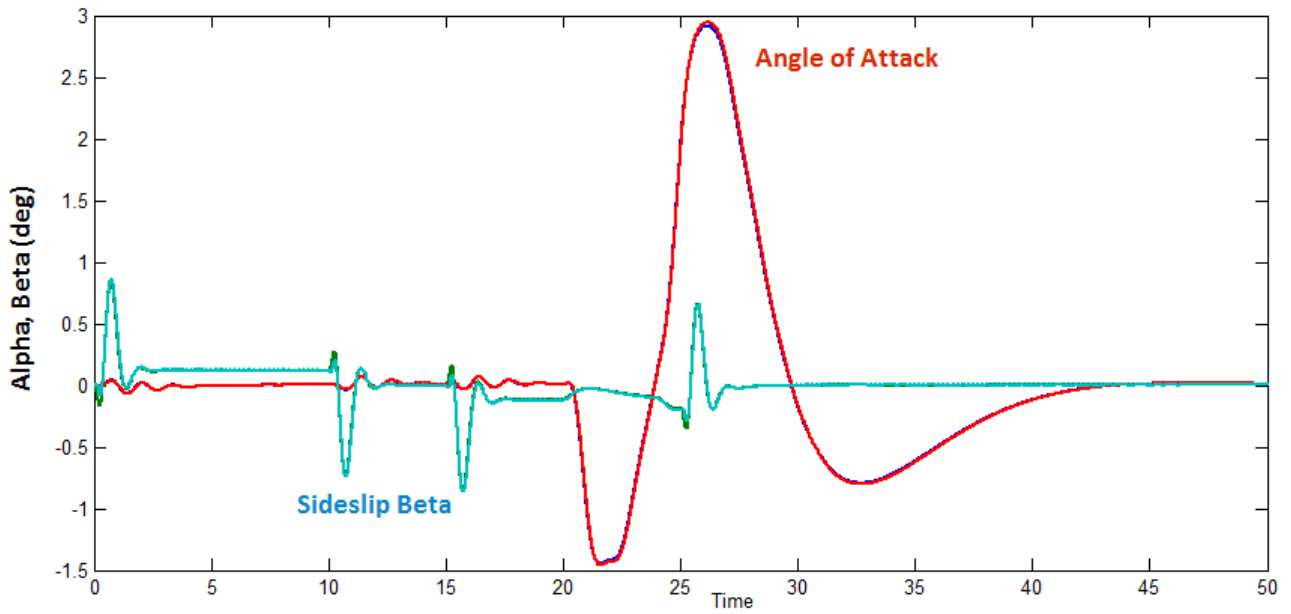
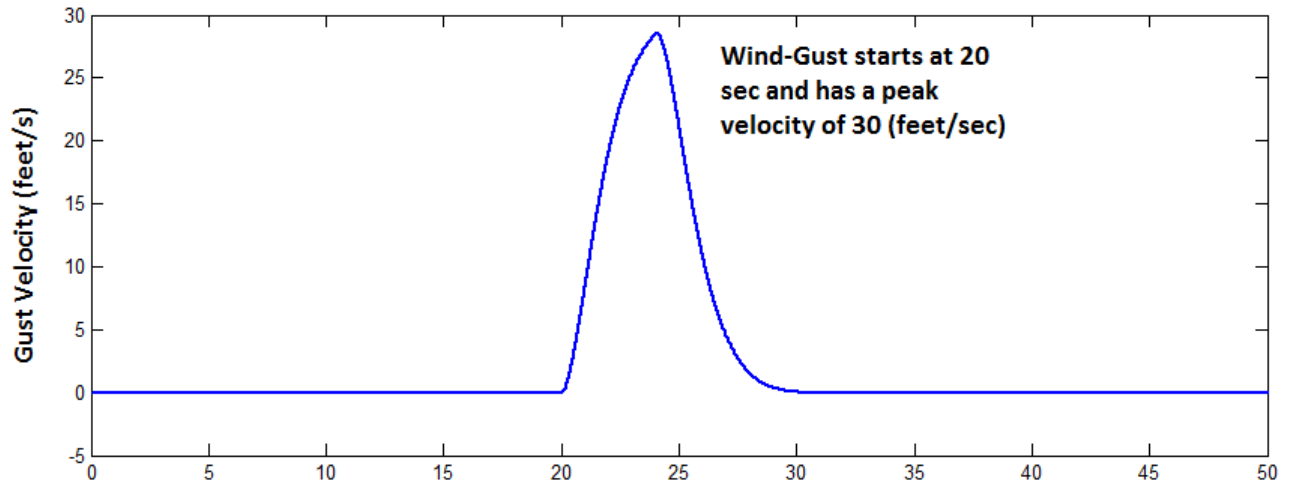
The second simulation excites mainly the lateral dynamics. The aircraft is commanded to rotate $+20^\circ$ in roll for 10 seconds, followed by a negative -20° roll in the opposite direction for another 10 seconds. The wind-gust excitation is the same as in our previous example and it is applied at 20 sec. This simulation is intended to demonstrate how the pilot changes the flight direction of the aircraft by rolling the vehicle. The figures show the simulation results from the lateral maneuver. The figure below shows how the aircraft attitude and rates as they respond to the roll commands. The results are similar to the rigid-body results. Structural flexibility can be observed in the rates.

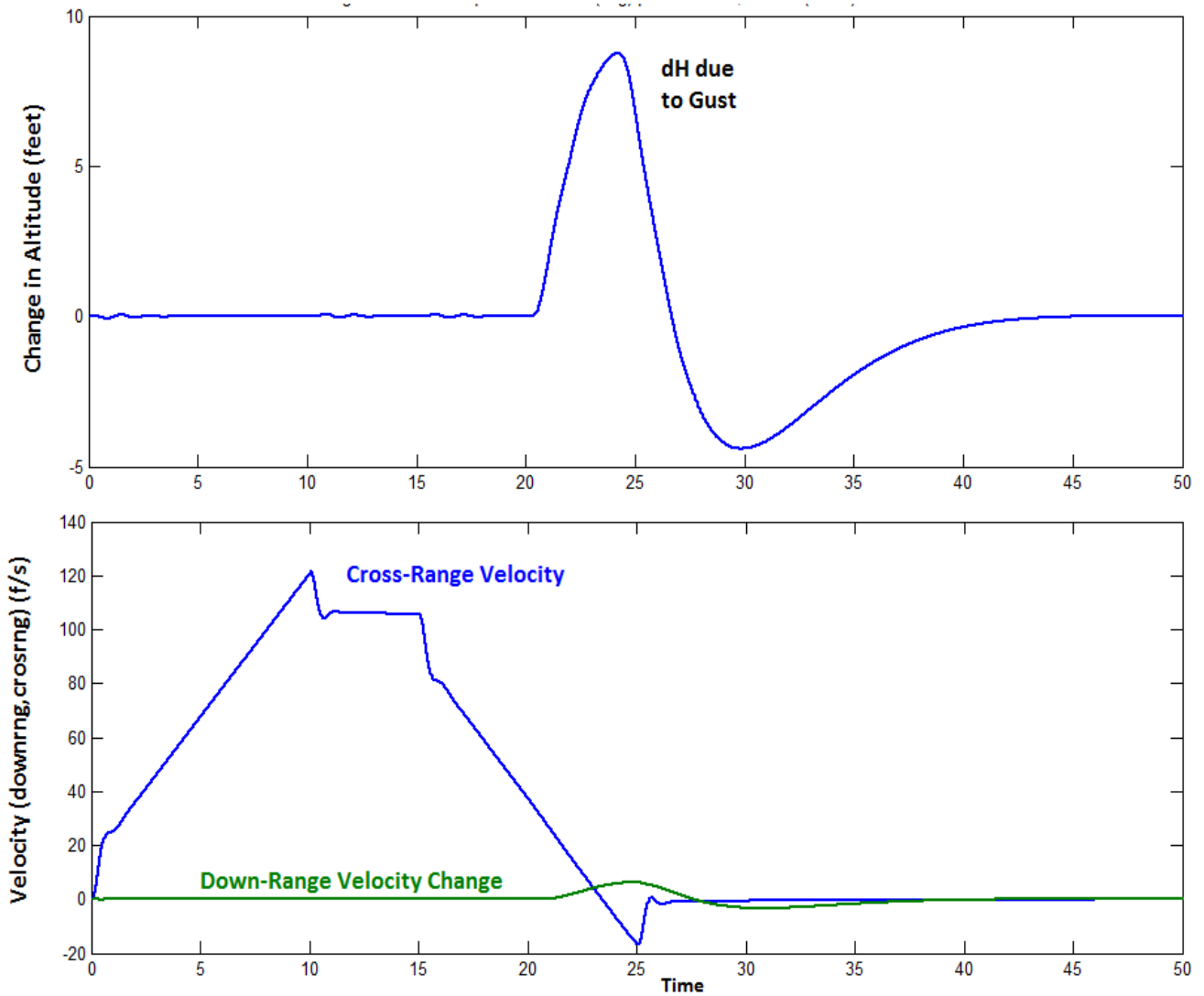
Fighter Aircraft Response to +/- 20 (deg) phi maneuver, and 30 (ft/sec) Wind-Gust



The next figures show the wind-gust velocity which is applied at 20 seconds and the variations in the aircraft angles of attack and sideslip (α , β). There is a small sideslip due to roll, and alpha is excited after 20 sec by the wind-gust disturbance. We also show the translational accelerations which are mainly in the y direction due to rolling. There is also acceleration in the $-z$ direction (up) due to the gust. We also see the variation in the cross-range velocity (blue) due to the roll maneuver which causes a change in the aircraft horizontal direction. The change in altitude δH is caused by the upward gust. The aircraft eventually corrects the altitude transient. The down-range speed δV in (green) does not change much.

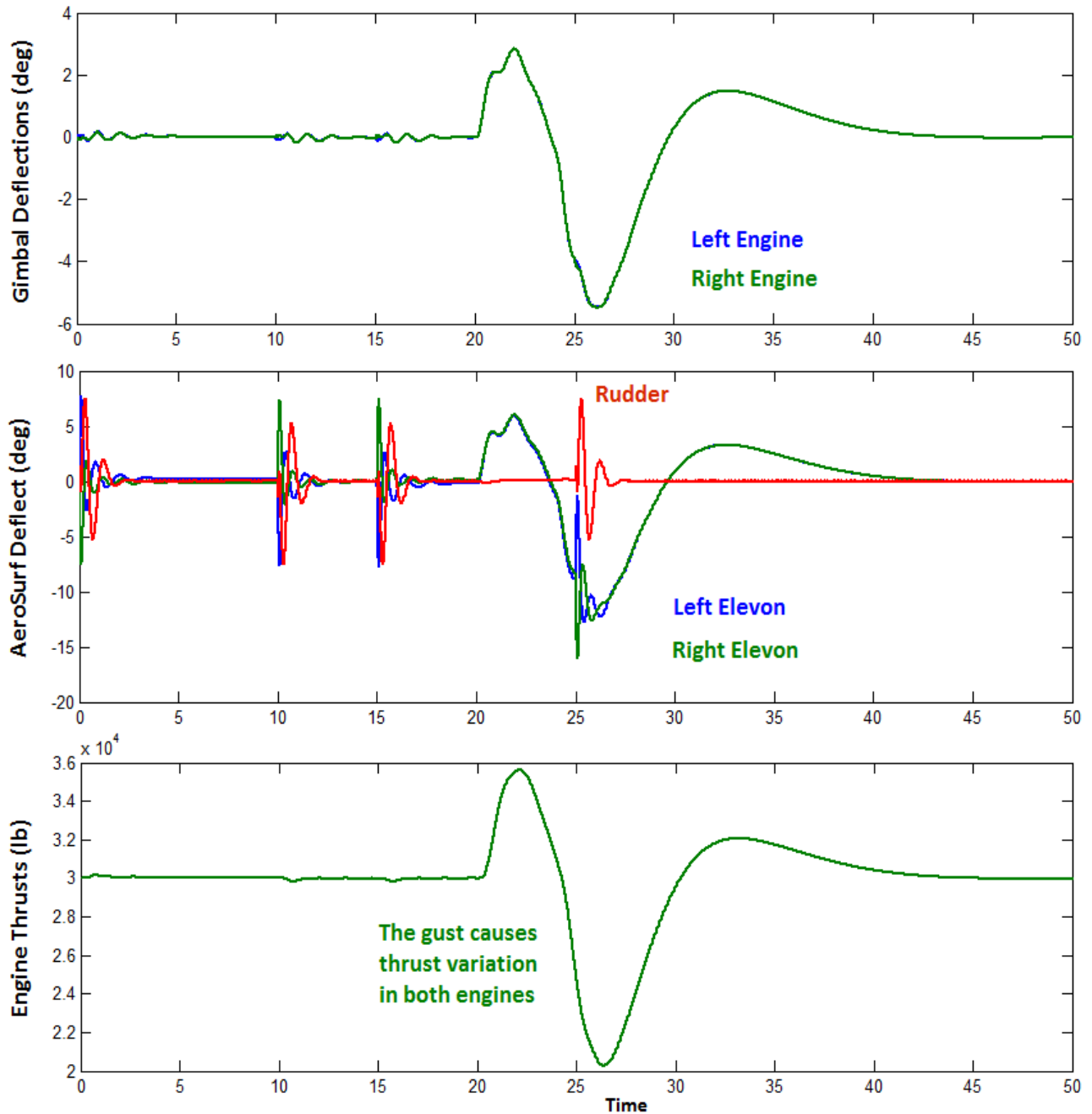
Fighter Aircraft Response to +/- 20 (deg) phi maneuver, and 30 (ft/sec) Wind-Gust





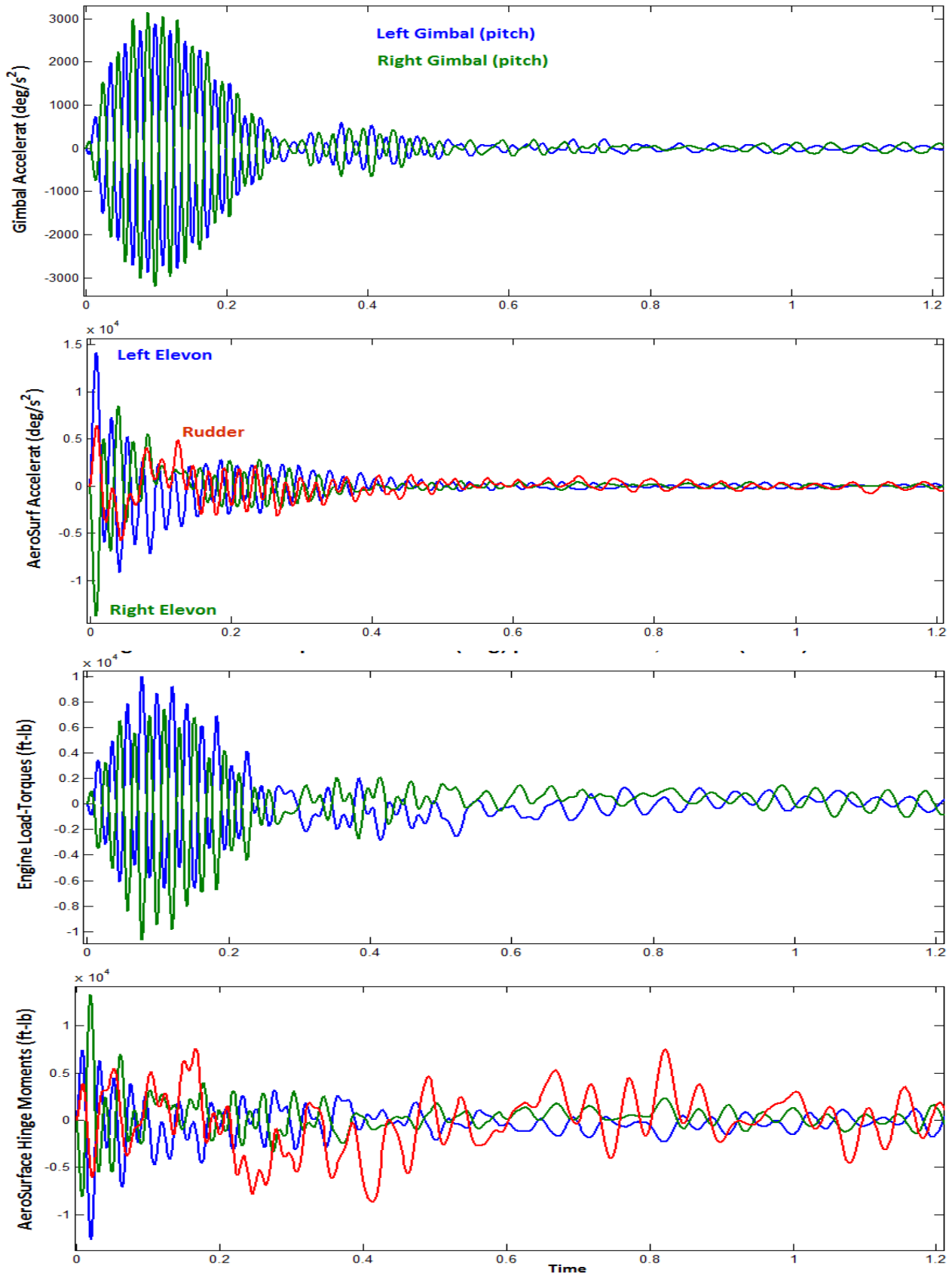
The figure below shows the aircraft controls. The aileron (differential elevons) and the rudder are both used during the roll maneuver. The elevons and the engine throttles are used to control the vehicle response to the wind disturbance. The gust creates a transient in the engine thrusts which eventually return to nominal at 30,000 (lb) each.

Fighter Aircraft Response to +/- 20 (deg) phi maneuver, and 30 (ft/sec) Wind-Gust



The last two figures show the pitch gimbal accelerations and the load-torques at the two TVC engines. It also shows the accelerations and hinge moments at the three aerosurfaces. The left and right elevators are blue and green respectively, and the rudder is shown in red. The duration of the plots is only 1.2 seconds to demonstrate that there is a significant amount of structural excitation at the engine gimbals and at the control surfaces.

Fighter Aircraft Response to +/- 20 (deg) phi maneuver, and 30 (ft/sec) Wind-Gust



2.8 Stability Analysis

The stability analysis is performed in the frequency domain by executing the Matlab file “run.m”. It uses the Simulink model “*Open_Loop.Mdl*”, shown below, which is similar to the model used for rigid-body analysis, to calculate the frequency response between the opened input and output. It consists of four control loops, three rotational and one translational. The flex vehicle and the flight control subsystems are the same as the ones used in the simulation model “*Flex_Sim.mdl*”. Only one loop is opened at a time while the other three loops must be closed, as shown for roll analysis below. This Simulink model can be modified by closing the roll loop, opening another loop, and rerunning the script file that will plot the Bode and Nichols charts. The stability analysis results in roll and pitch are shown in the figures below. They all have sufficient phase and gain margins. Some of the flex modes are attenuated with filters.

```

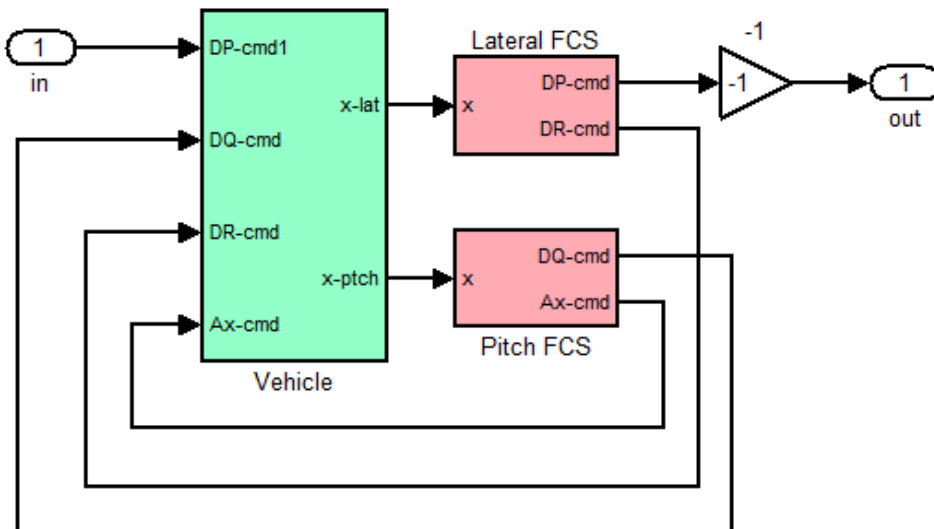
% Coupled Axes Frequency Response Analysis
d2r=pi/180; r2d=180/pi;
load Kmix4.mat Kmix4 -ascii           % Mixing Logic Matrix
load Kgp.mat Kgp -ascii              % Pitch Gains
load Kgl.mat Kgl -ascii              % Lateral Gains
[Av, Bv, Cv, Dv]= vehicle_flex;      % Flex Simulation Model
[Ae, Be, Ce, De]= elevator;         % Elevator Actuator
[At, Bt, Ct, Dt]= engine_tvc;       % Engine TVC Actuator

[Ao,Bo,Co,Do]=linmod('Open_Loop');  % Frequency Response Mo
w=logspace(-3, 3, 10000);
sys= SS(Ao,Bo,Co,Do);
figure(1); Nichols(sys,w)
figure(2); Bode(sys,w)

```

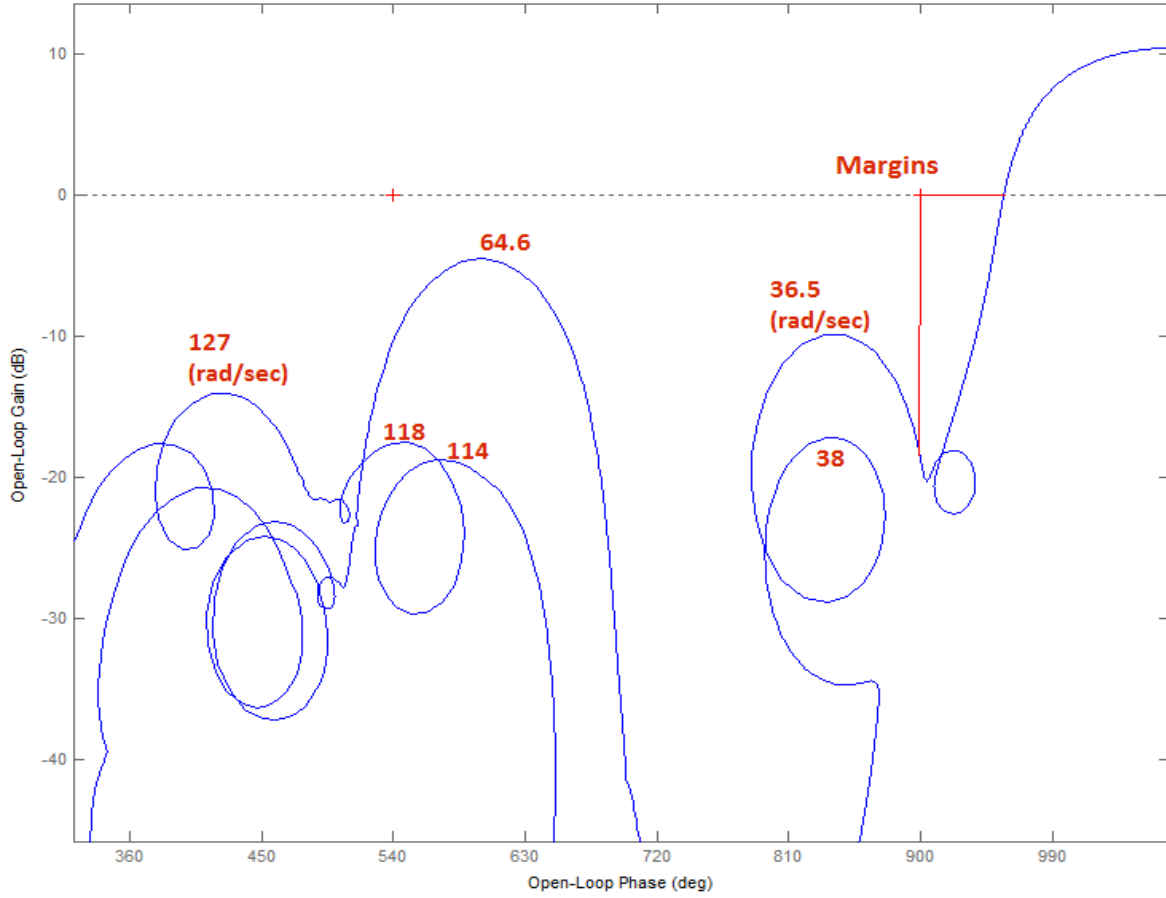
Open-Loop Flex Analysis Model

Roll loop is Opened, Other loops are Closed



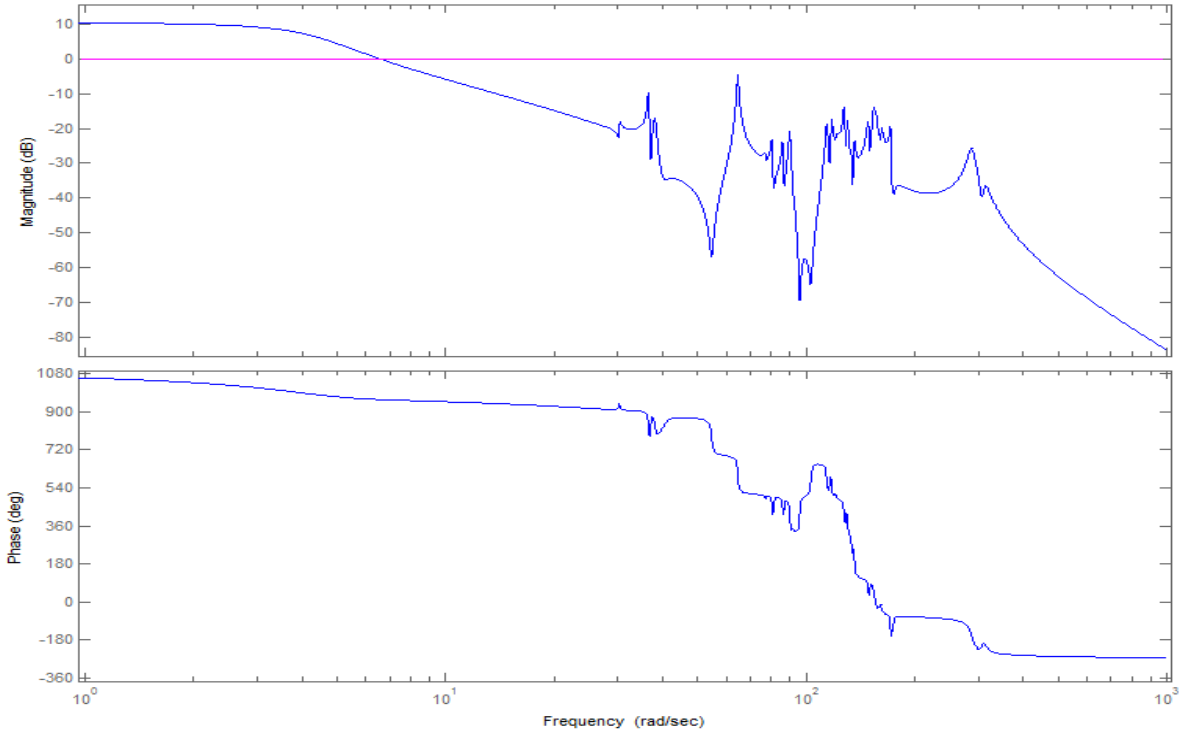
Roll Axis Stability

Nichols Chart

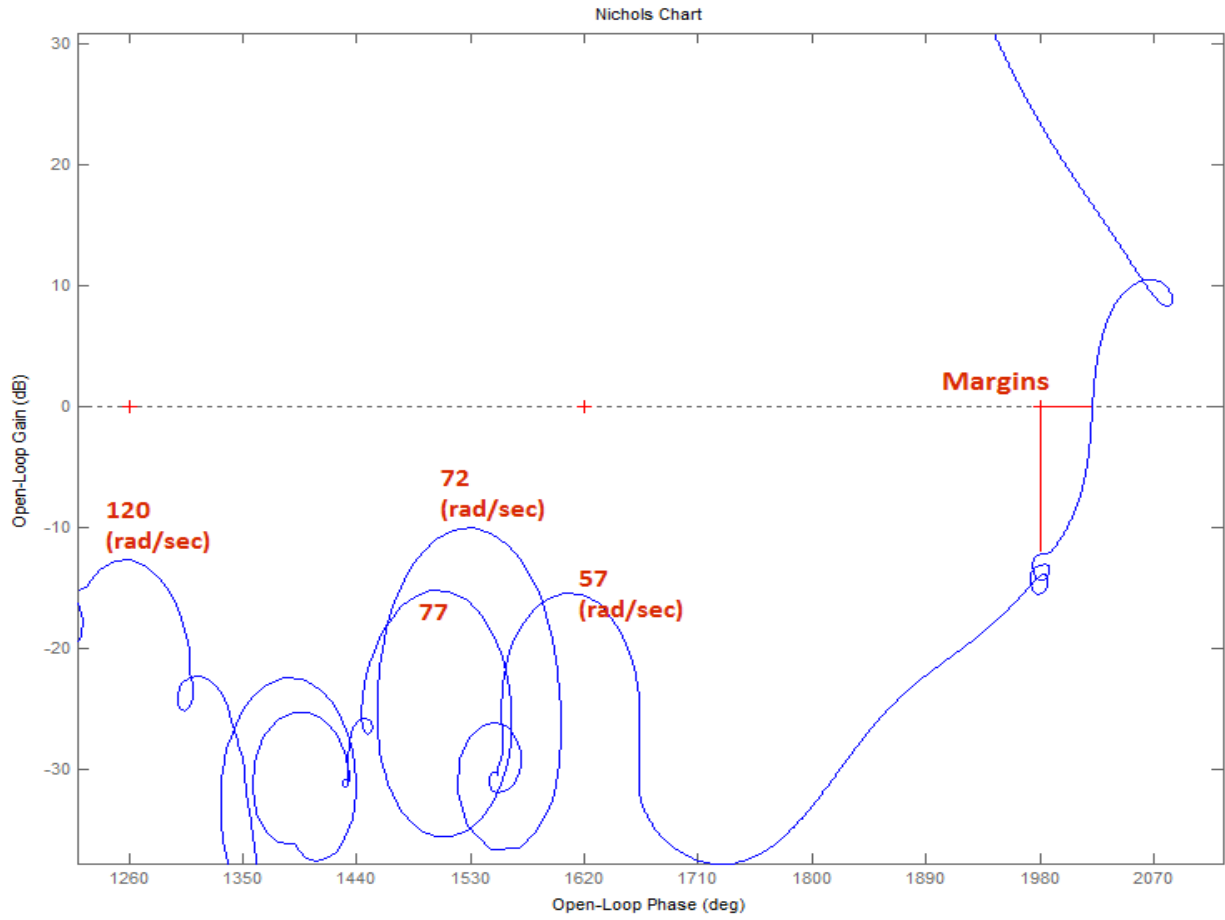


Roll Axis Bode

Bode Diagram



Pitch Axis Stability



Pitch Axis

