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 aero-surfaces 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

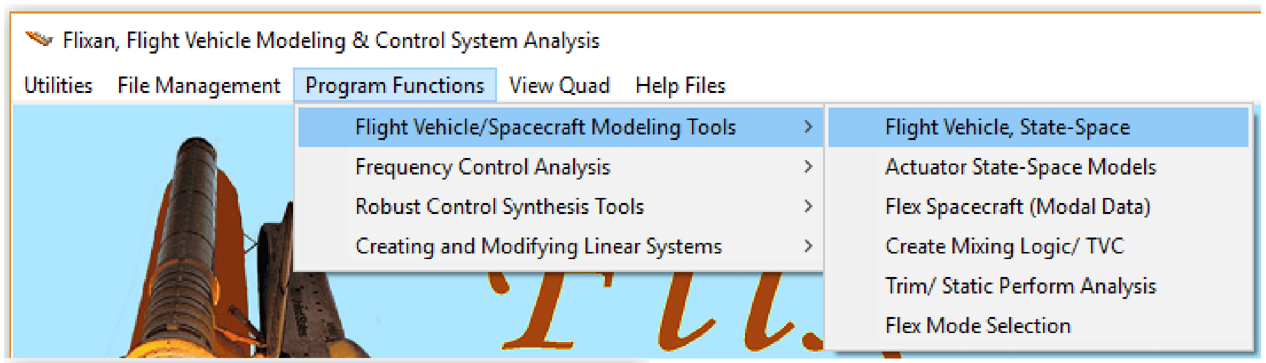
The aircraft data for generating the two rigid-body models are in file "*Fighter_RB.Inp*", which is located in folder "*Examples\Fighter Aircraft\Rigid Body Design*". Their titles are: "*Fighter Aircraft Design Model*" and "*Fighter Aircraft Rigid Body Model*". The flag lines below the titles indicate 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.

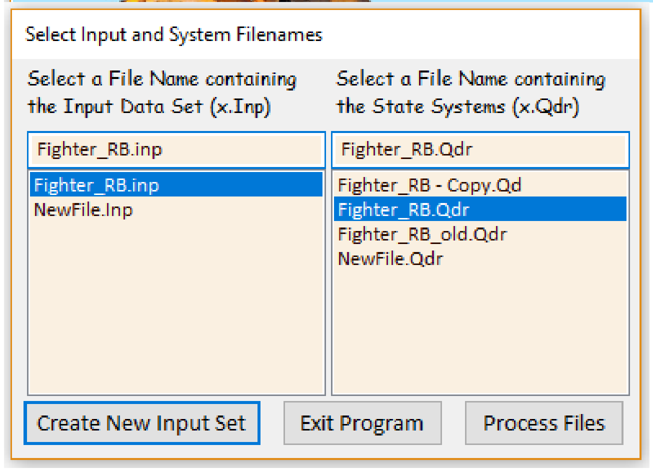
The two engines are defined to be gimbaling in a single direction (not two) and they are also throttling "*Single-Gimbal, Throttling*". The gimbaling direction for the two engines is defined to be in pitch, because the gimbaling direction is defined in the data by the maximum pitch and yaw deflections, 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° 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° in pitch and +12.7° in yaw, and for the right engine it should be +12.7° 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 ±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

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.


Flixan, Flight Vehicle Modeling & Control System Analysis
Utilities
File Management
Program Functions
View Quad
Help Files
Flight Vehicle/Spacecraft Modeling Tools
Frequency Control Analysis
Robust Control Synthesis Tools
Creating and Modifying Linear Systems
Flight Vehicle, State-Space
Actuator State-Space Models
Flex Spacecraft (Modal Data)
Create Mixing Logic/ TVC
Trim/ Static Perform Analysis
Flex Mode Selection


Select Input and System Filenames
Select a File Name containing the Input Data Set (x.Inp)
Select a File Name containing the State Systems (x.Qdr)
Create New Input Set
Exit Program
Process Files

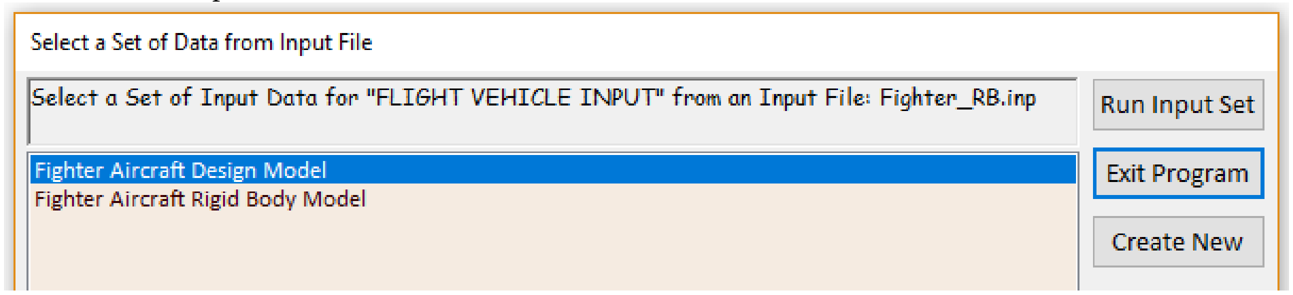
Fighter_RB.inp

Fighter_RB.inp
NewFile.Inp

Fighter_RB.Qdr

Fighter_RB - Copy.Qd
Fighter_RB.Qdr
Fighter_RB_old.Qdr
NewFile.Qdr

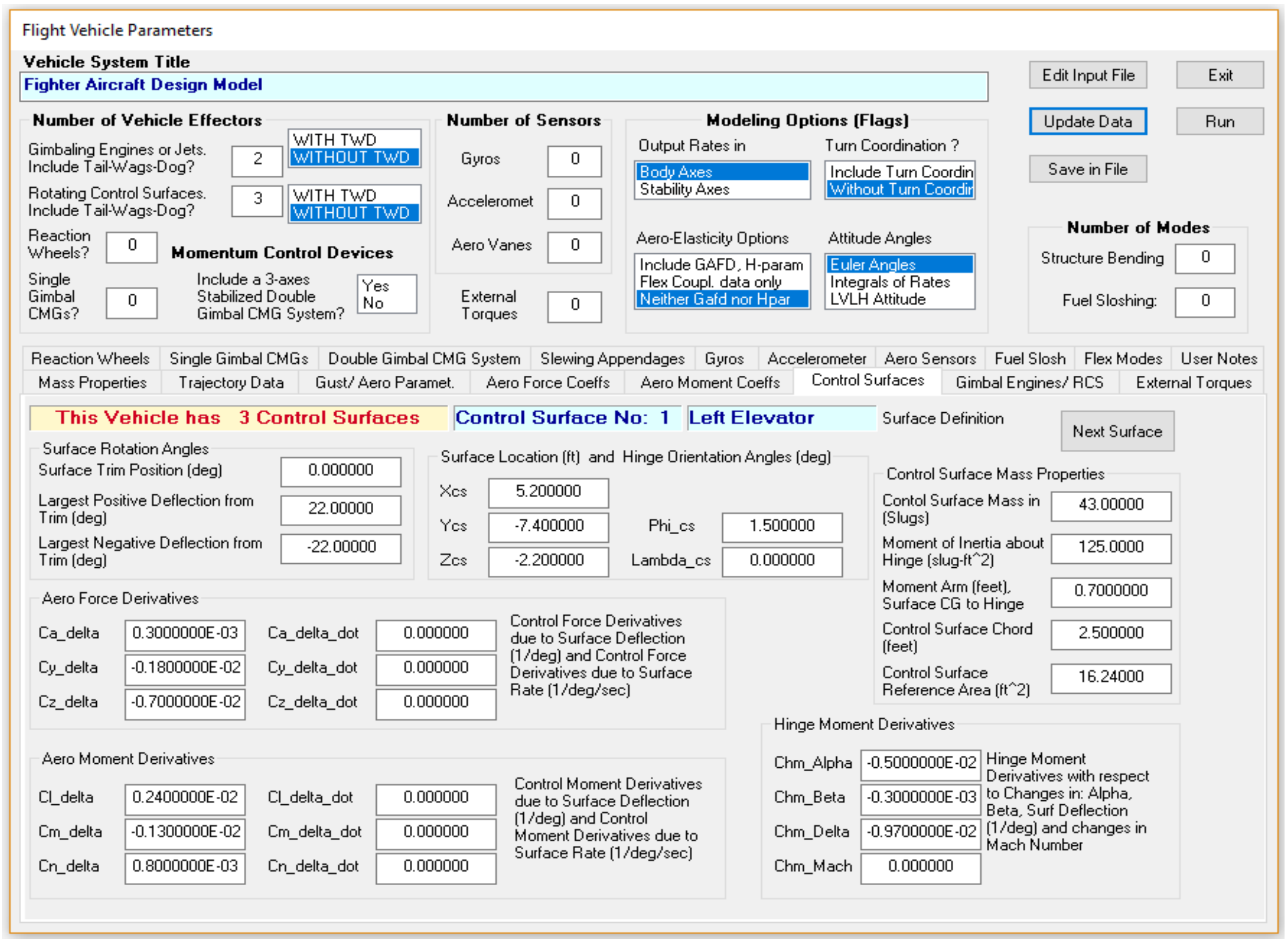
From the flight vehicle data selection menu select the first title "*Fighter Aircraft Design Model*" and click on "*Run Input Set*".



Select a Set of Input Data for "FLIGHT VEHICLE INPUT" from an Input File: Fighter_RB.inp

Fighter Aircraft Design Model
Fighter Aircraft Rigid Body Model

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	
Single Gimbal CMGs?	0	

Momentum Control Devices
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 Coordir]
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

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

Surface Rotation Angles

Surface Trim Position (deg)	0.000000
Largest Positive Deflection from Trim (deg)	22.00000
Largest Negative Deflection from Trim (deg)	-22.00000

Surface Location (ft) and Hinge Orientation Angles (deg)

Xcs	5.200000		
Ycs	-7.400000	Phi_cs	1.500000
Zcs	-2.200000	Lambda_cs	0.000000

Control Surface Mass Properties

Contol Surface Mass in (Slugs)	43.00000
Moment of Inertia about Hinge (slug-ft^2)	125.0000
Moment Arm (feet), Surface CG to Hinge	0.7000000
Control Surface Chord (feet)	2.500000
Control Surface Reference Area (ft^2)	16.24000

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

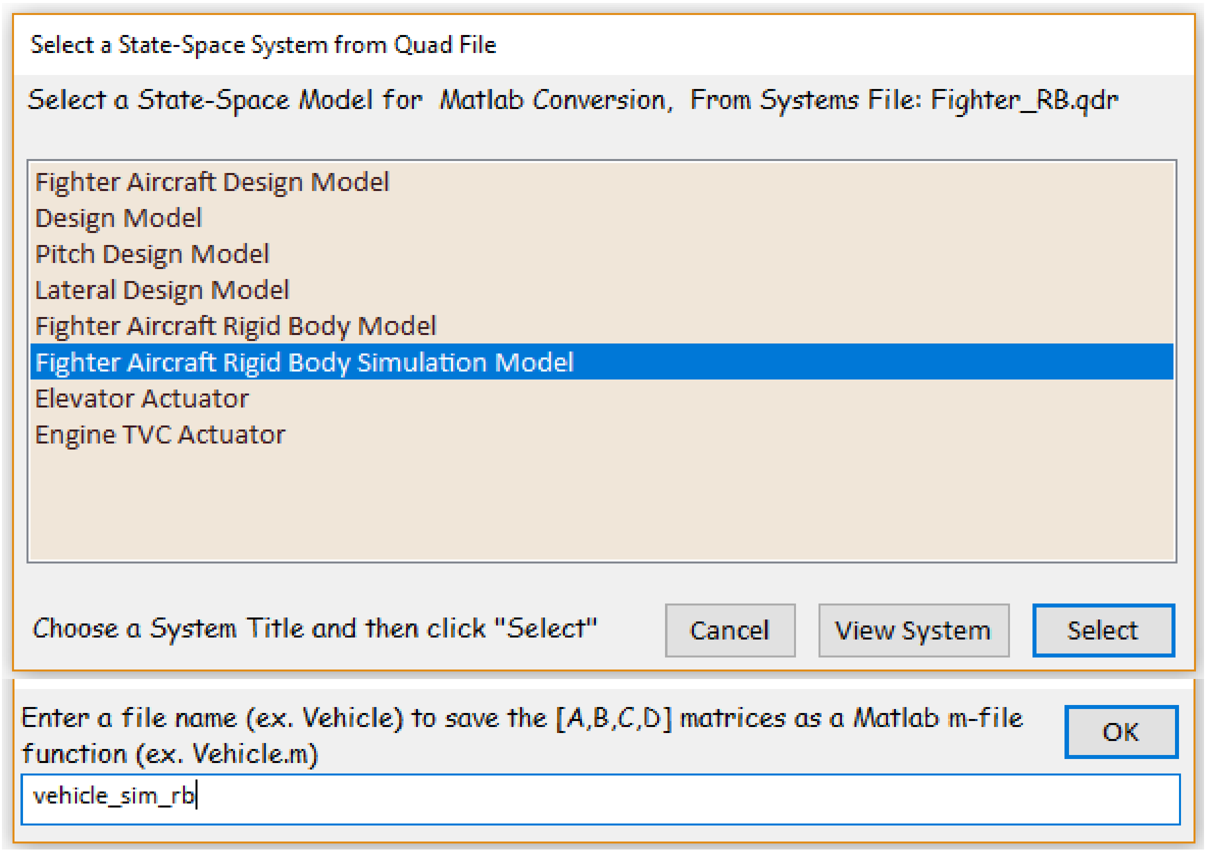
Hinge Moment Derivatives with respect to Changes in: Alpha, Beta, Surf Deflection (1/deg) and changes in Mach Number

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 Kmix4. 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.

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.


Select a State-Space System from Quad File

Select a State-Space Model for Matlab Conversion, From Systems File: Fighter_RB.qdr

- Fighter Aircraft Design Model
- Design Model
- Pitch Design Model
- Lateral Design Model
- Fighter Aircraft Rigid Body Model
- Fighter Aircraft Rigid Body Simulation Model
- Elevator Actuator
- Engine TVC Actuator

Choose a System Title and then click "Select"

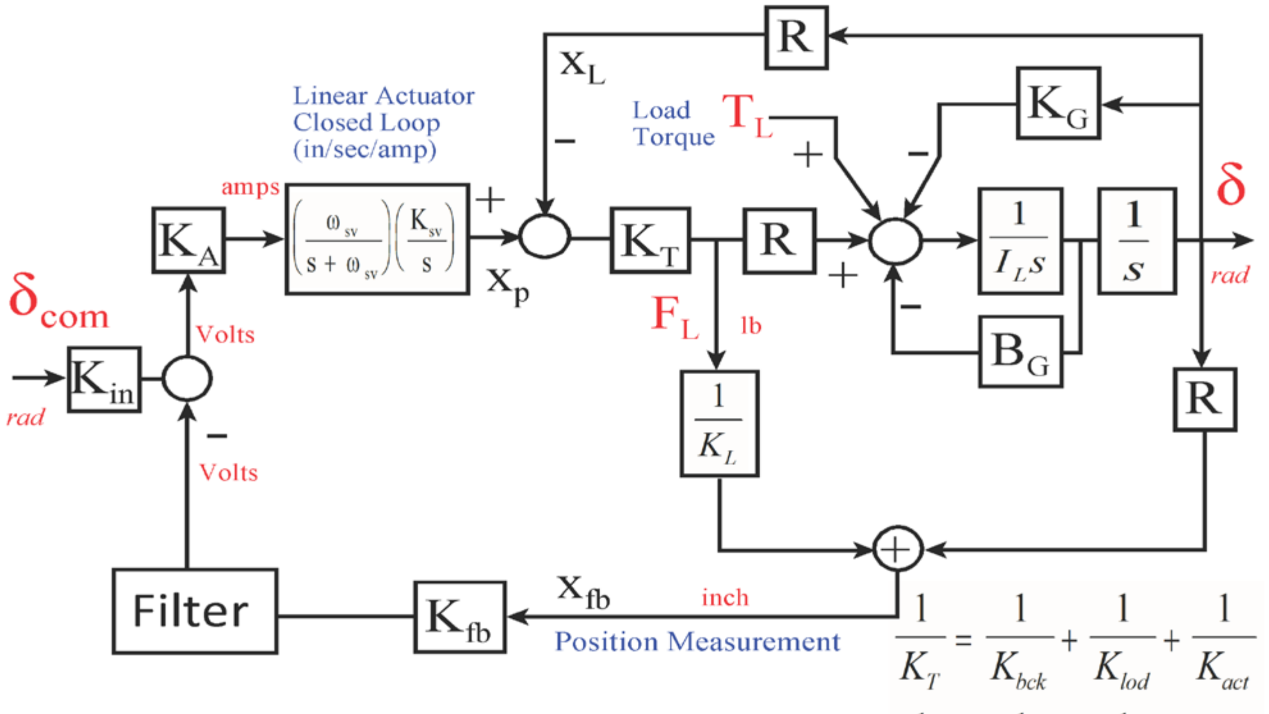
Enter a file name (ex. Vehicle) to save the [A,B,C,D] matrices as a Matlab m-file function (ex. Vehicle.m)

vehicle_sim_rb

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.



Simple Actuator Model

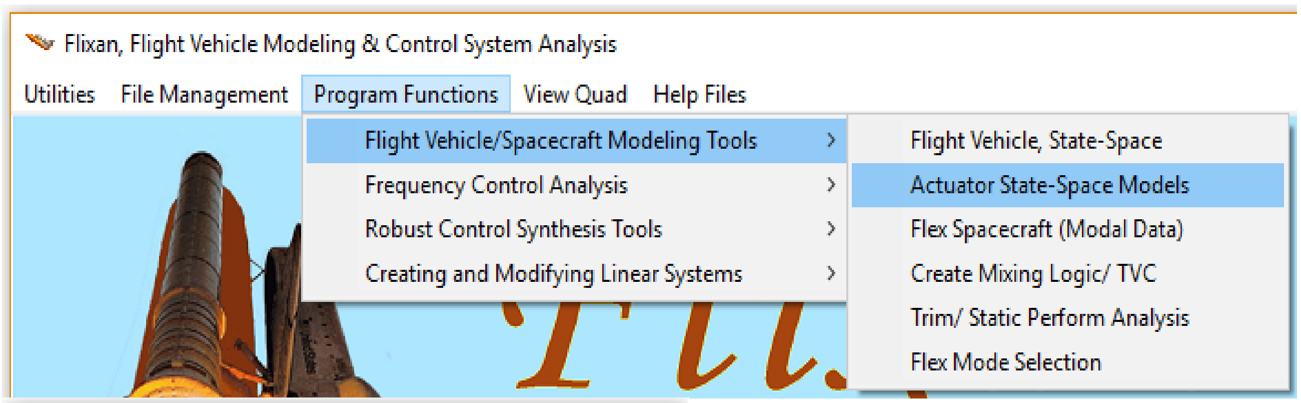
$$\frac{1}{K_T} = \frac{1}{K_{bck}} + \frac{1}{K_{lod}} + \frac{1}{K_{act}}$$

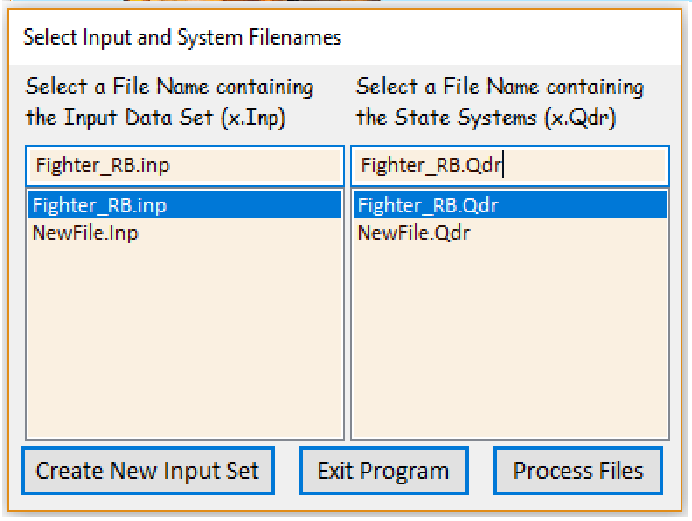
$$\frac{1}{K_L} = \frac{1}{K_{bck}} + \frac{1}{K_{lod}}$$

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.

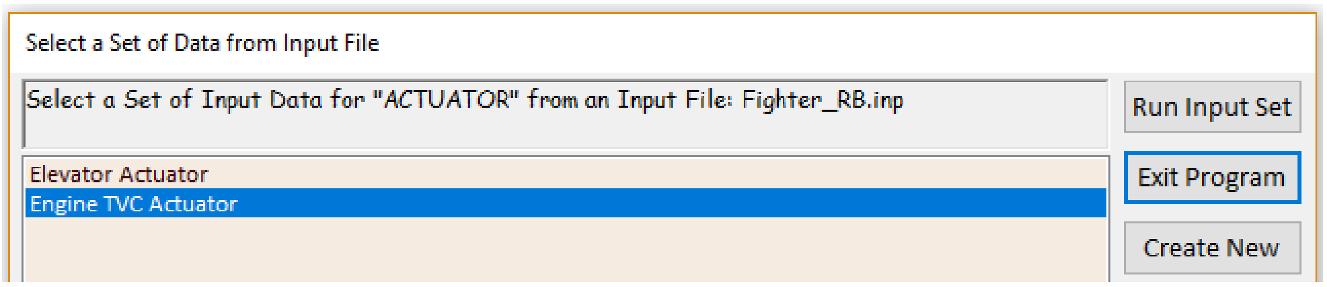
Running the Actuator Program

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.

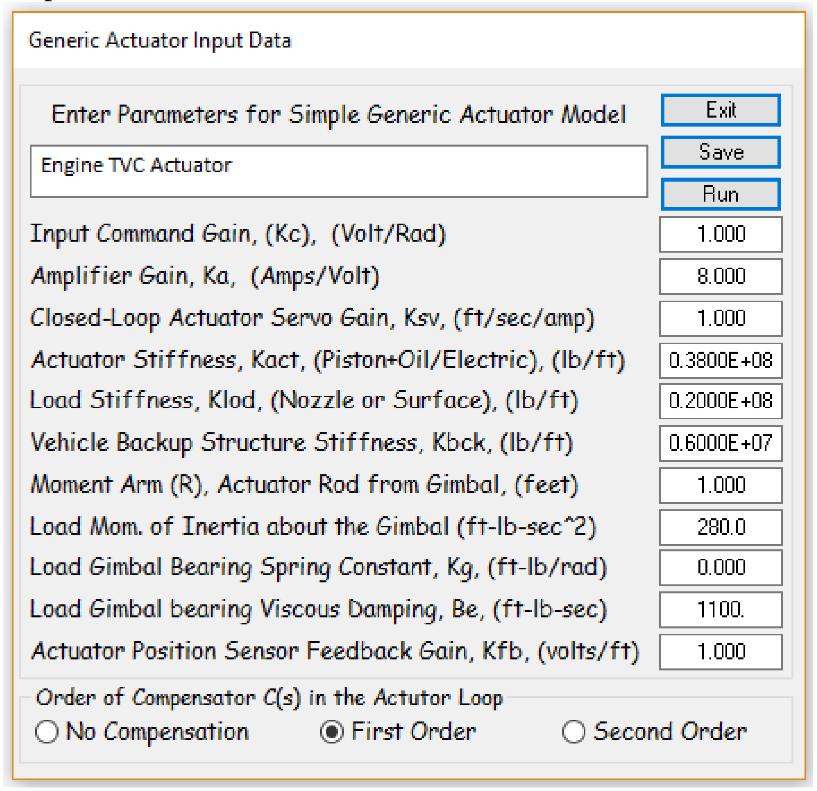

Flixan, Flight Vehicle Modeling & Control System Analysis
Utilities
File Management
Program Functions
View Quad
Help Files
Flight Vehicle/Spacecraft Modeling Tools
Frequency Control Analysis
Robust Control Synthesis Tools
Creating and Modifying Linear Systems
Flight Vehicle, State-Space
Actuator State-Space Models
Flex Spacecraft (Modal Data)
Create Mixing Logic/ TVC
Trim/ Static Perform Analysis
Flex Mode Selection


Select Input and System Filenames
Select a File Name containing the Input Data Set (x.Inp)
Select a File Name containing the State Systems (x.Qdr)
Fighter_RB.inp
Fighter_RB.Qdr
Fighter_RB.inp
NewFile.Inp
Fighter_RB.Qdr
NewFile.Qdr
Create New Input Set
Exit Program
Process Files

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.


Select a Set of Input Data for "ACTUATOR" from an Input File: Fighter_RB.inp

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

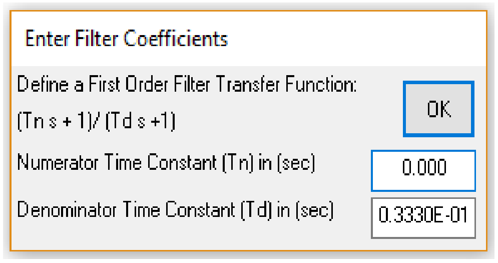
Enter Parameters for Simple Generic Actuator Model

Engine TVC Actuator

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 Actutor Loop

No Compensation [✓ First Order] Second Order



Enter Filter Coefficients

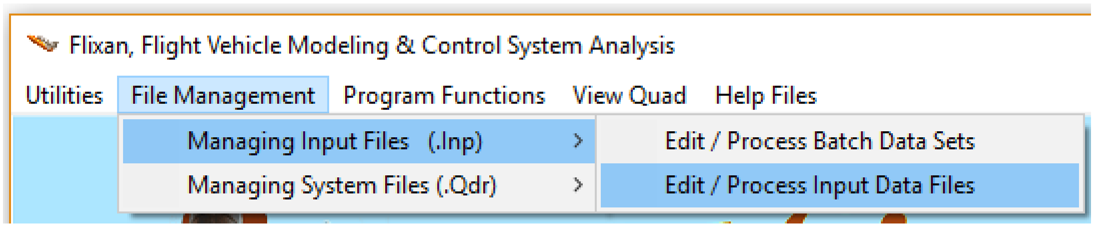
Define a First Order Filter Transfer Function:
(Tn s + 1)/ (Td s +1)

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*".

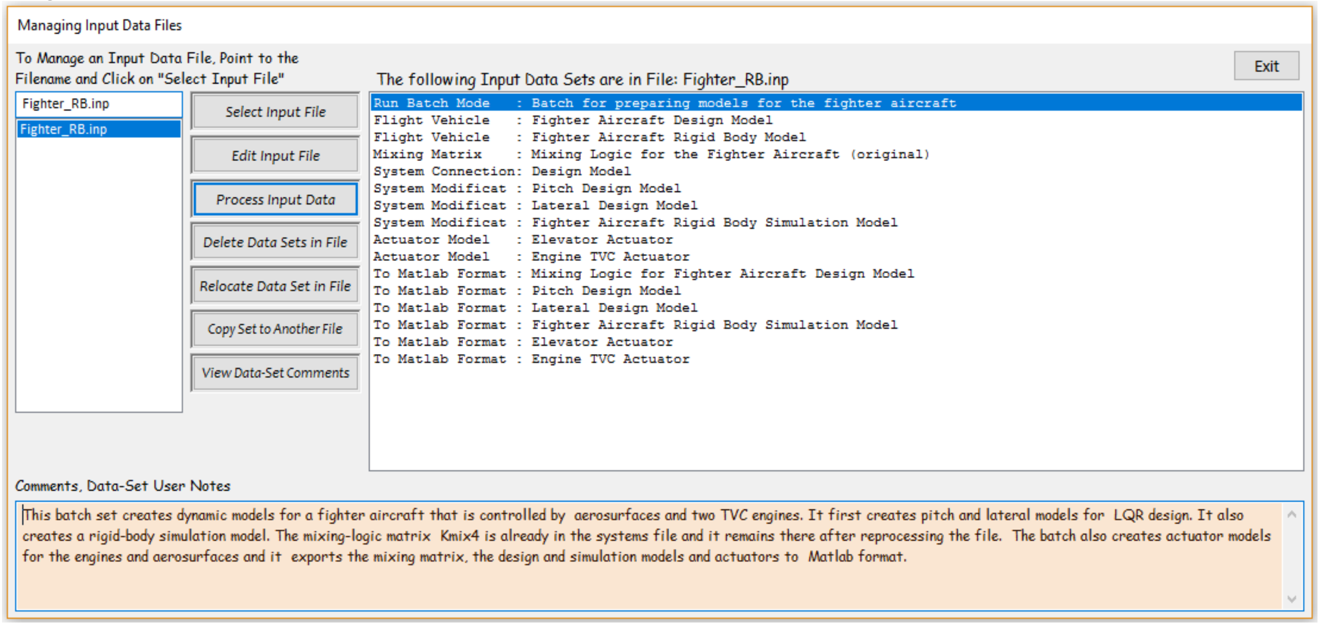
1.7 Batch Mode Processing

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.



Flixan, Flight Vehicle Modeling & Control System Analysis
Utilities
File Management
Program Functions
View Quad
Help Files
Managing Input Files (.Inp)
Managing System Files (.Qdr)
Edit / Process Batch Data Sets
Edit / Process Input Data Files

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*".


Managing Input Data Files
To Manage an Input Data File, Point to the Filename and Click on "Select Input File"
Fighter_RB.inp
Fighter_RB.inp
Select Input File
Edit Input File
Process Input Data
Delete Data Sets in File
Relocate Data Set in File
Copy Set to Another File
View Data-Set Comments
Exit
Comments, Data-Set User Notes

The following Input Data Sets are in File: Fighter_RB.inp

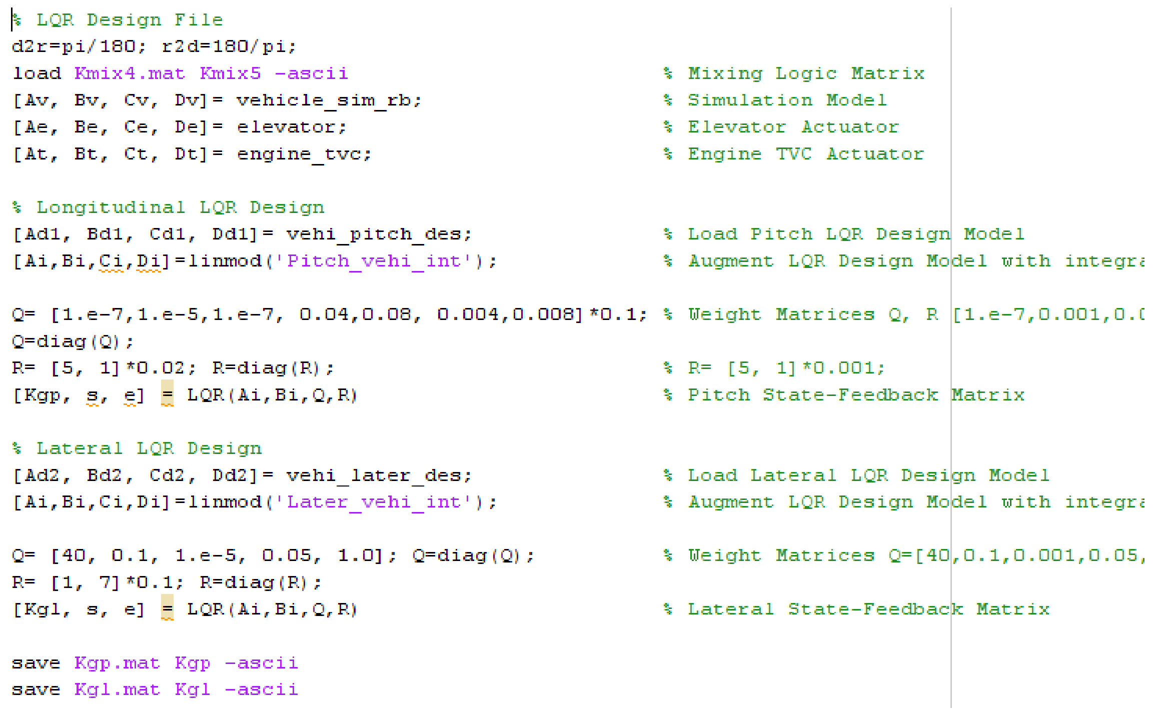
```
Run Batch Mode    : Batch for preparing models for the fighter aircraft
Flight Vehicle    : Fighter Aircraft Design Model
Flight Vehicle    : Fighter Aircraft Rigid Body Model
Mixing Matrix     : Mixing Logic for the Fighter Aircraft (original)
System Connection: Design Model
System Modificat : Pitch Design Model
System Modificat : Lateral Design Model
System Modificat : Fighter Aircraft Rigid Body Simulation Model
Actuator Model    : Elevator Actuator
Actuator Model    : Engine TVC Actuator
To Matlab Format : Mixing Logic for Fighter Aircraft Design Model
To Matlab Format : Pitch Design Model
To Matlab Format : Lateral Design Model
To Matlab Format : Fighter Aircraft Rigid Body Simulation Model
To Matlab Format : Elevator Actuator
To Matlab Format : Engine TVC Actuator
```

This batch set creates dynamic models for a fighter aircraft that is controlled by aerosurfaces and two TVC engines. It first creates pitch and lateral models for LQR design. It also creates a rigid-body simulation model. The mixing-logic matrix Kmix4 is already in the systems file and it remains there after reprocessing the file. The batch also creates actuator models for the engines and aerosurfaces and it exports the mixing matrix, the design and simulation models and actuators to Matlab format.

1.8 Control Design

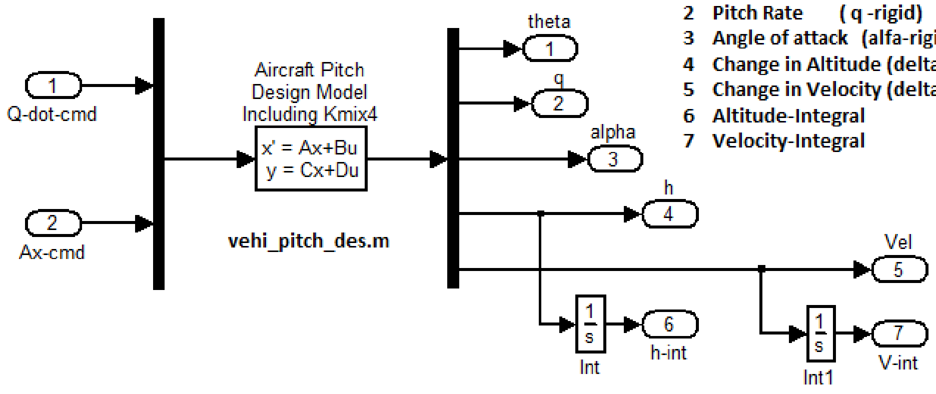
The control design and analysis in this example is performed using the Matlab program in subdirectory "*\Fighter Aircraft\Rigid Body Design\Mat_Rigid*". The actuator and vehicle pitch and lateral models are already converted to Matlab format and saved in this folder. The m-file "LQR_des.m" loads the 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 Kgp. 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 Kgl.



```
% 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_tvc;                            % 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 integra

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.
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 integra

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);
[Kgl, s, e] = LQR(Ai,Bi,Q,R)                             % Lateral State-Feedback Matrix

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

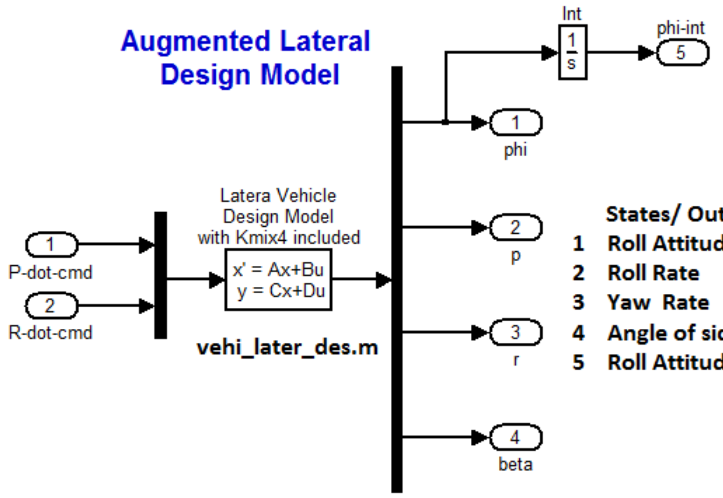
Augmented Pitch Design Model



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**

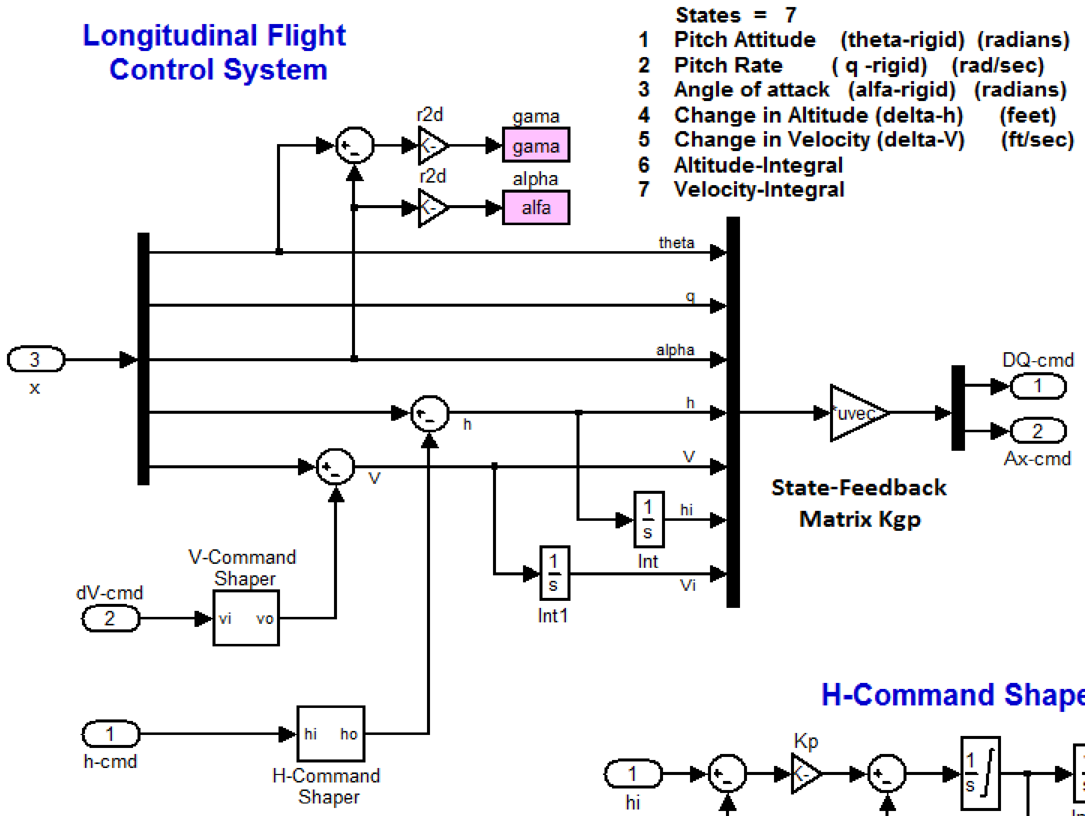
Augmented Lateral Design Model

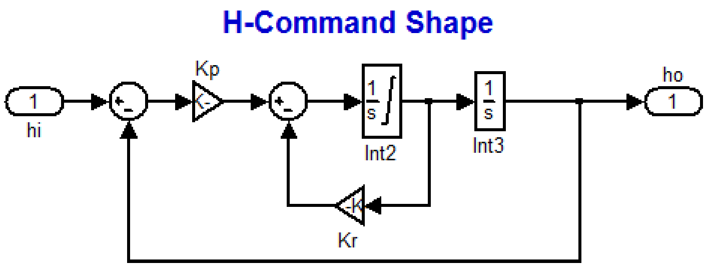


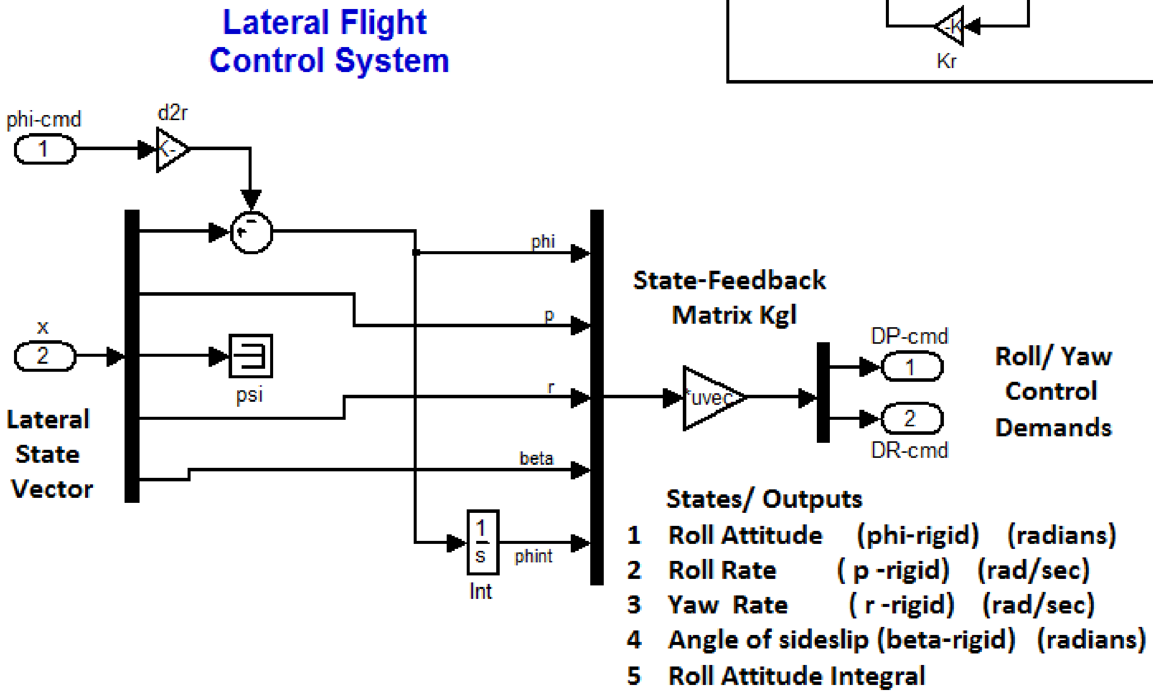
States/ Outputs:

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**

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.

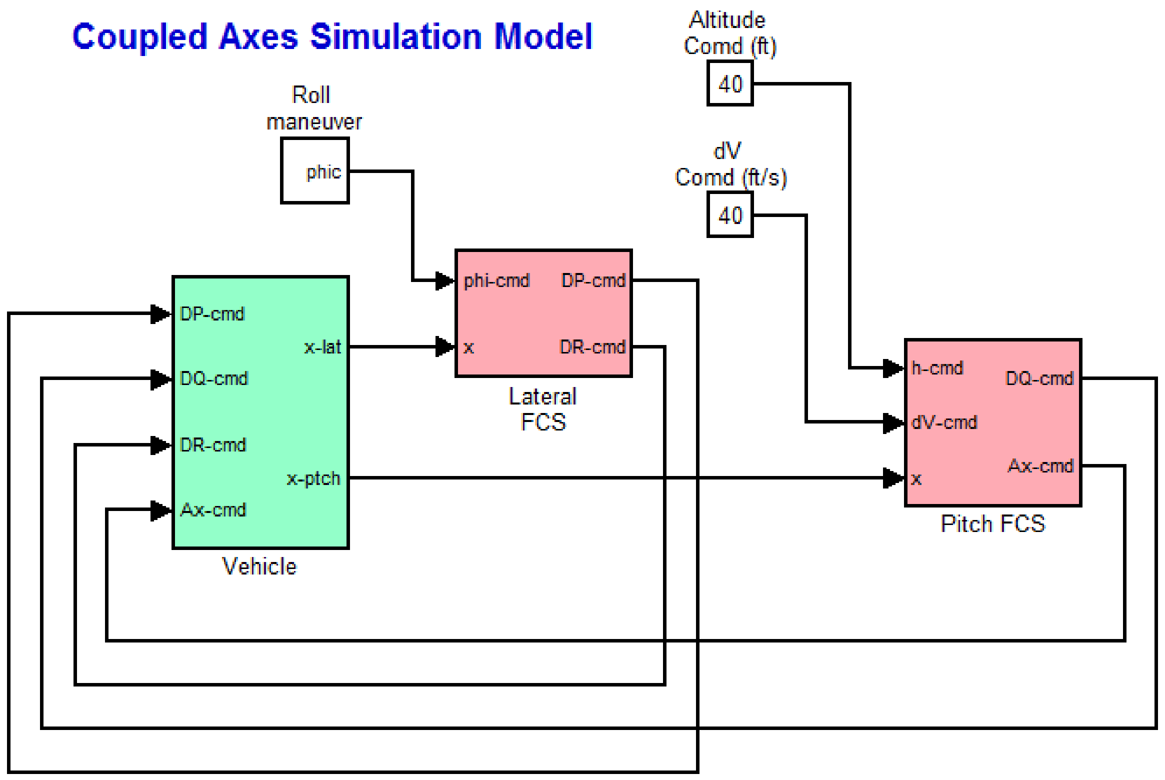

Longitudinal Flight Control System
States = 7
1 Pitch Attitude (theta-rigid) (radians)
2 Pitch Rate (q -rigid) (rad/sec)
3 Angle of attack (alfa-rigid) (radians)
4 Change in Altitude (delta-h) (feet)
5 Change in Velocity (delta-V) (ft/sec)
6 Altitude-Integral
7 Velocity-Integral
r2d
gama
gama
r2d
alpha
alfa
theta
q
alpha
h
V
hi
Vi
3
x
h
V
Int
Int1
uvec
DQ-cmd
1
2
Ax-cmd
State-Feedback Matrix Kgp
V-Command Shaper
dV-cmd
2
vi
vo
1
h-cmd
hi
ho
H-Command Shaper



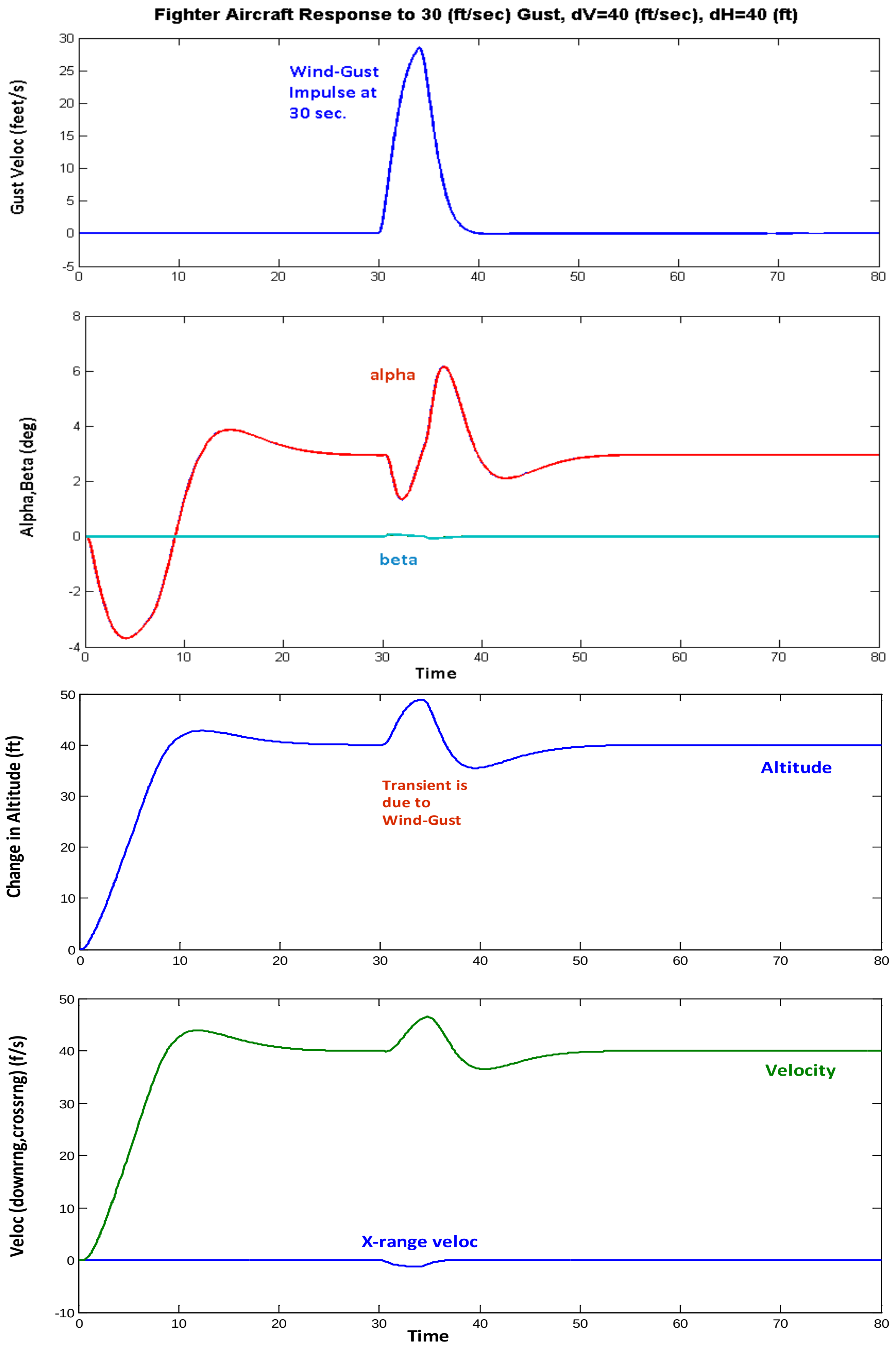

Lateral Flight Control System
phi-cmd
1
d2r
x
2
psi
Lateral State Vector
phi
p
r
beta
phint
Int
State-Feedback Matrix Kgl
uvec
DP-cmd
1
2
DR-cmd
Roll/ Yaw Control Demands
States/ Outputs
1 Roll Attitude (phi-rigid) (radians)
2 Roll Rate (p -rigid) (rad/sec)
3 Yaw Rate (r -rigid) (rad/sec)
4 Angle of sideslip (beta-rigid) (radians)
5 Roll Attitude Integral

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.



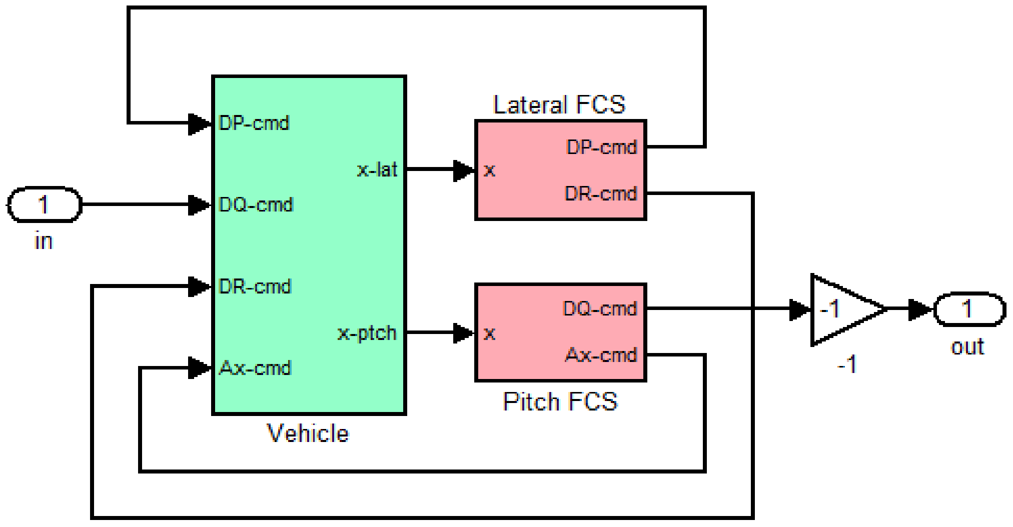
The vehicle dynamics block is shown in more details in the next figure. There are three rotational and one axial acceleration demands (DP, DQ, DR, Ax)$_{cmd}$ coming from the flight control systems. They are converted by the mixing logic matrix Kmix4 into 3 aerosurface deflection commands, 2 pitch gimbal 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.


Fighter Aircraft Response to 30 (ft/sec) Gust, dV=40 (ft/sec), dH=40 (ft)
Gust Veloc (feet/s)
Wind-Gust Impulse at 30 sec.
Alpha,Beta (deg)
alpha
beta
Time
Change in Altitude (ft)
Altitude
Transient is due to Wind-Gust
Veloc (downrng,crossrng) (f/s)
Velocity
X-range veloc
Time

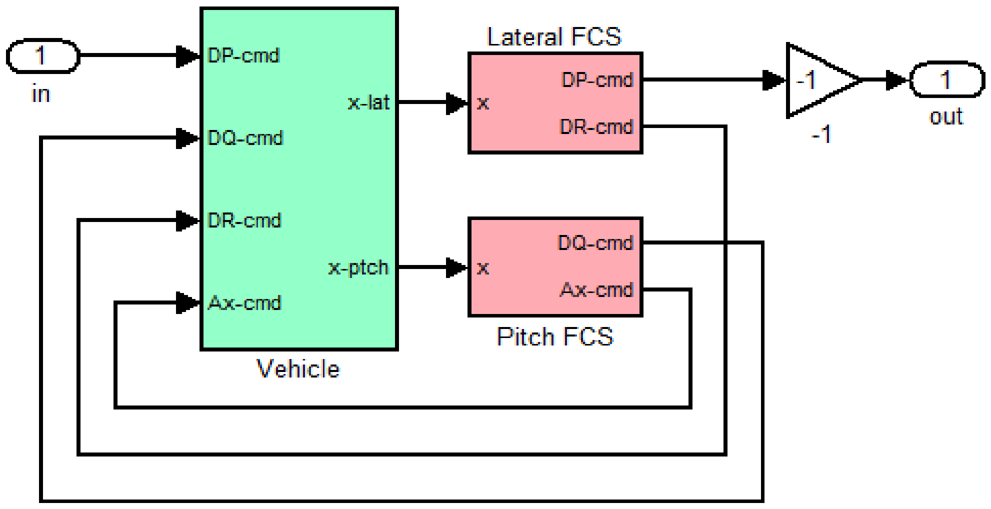
1.10 Stability Analysis

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

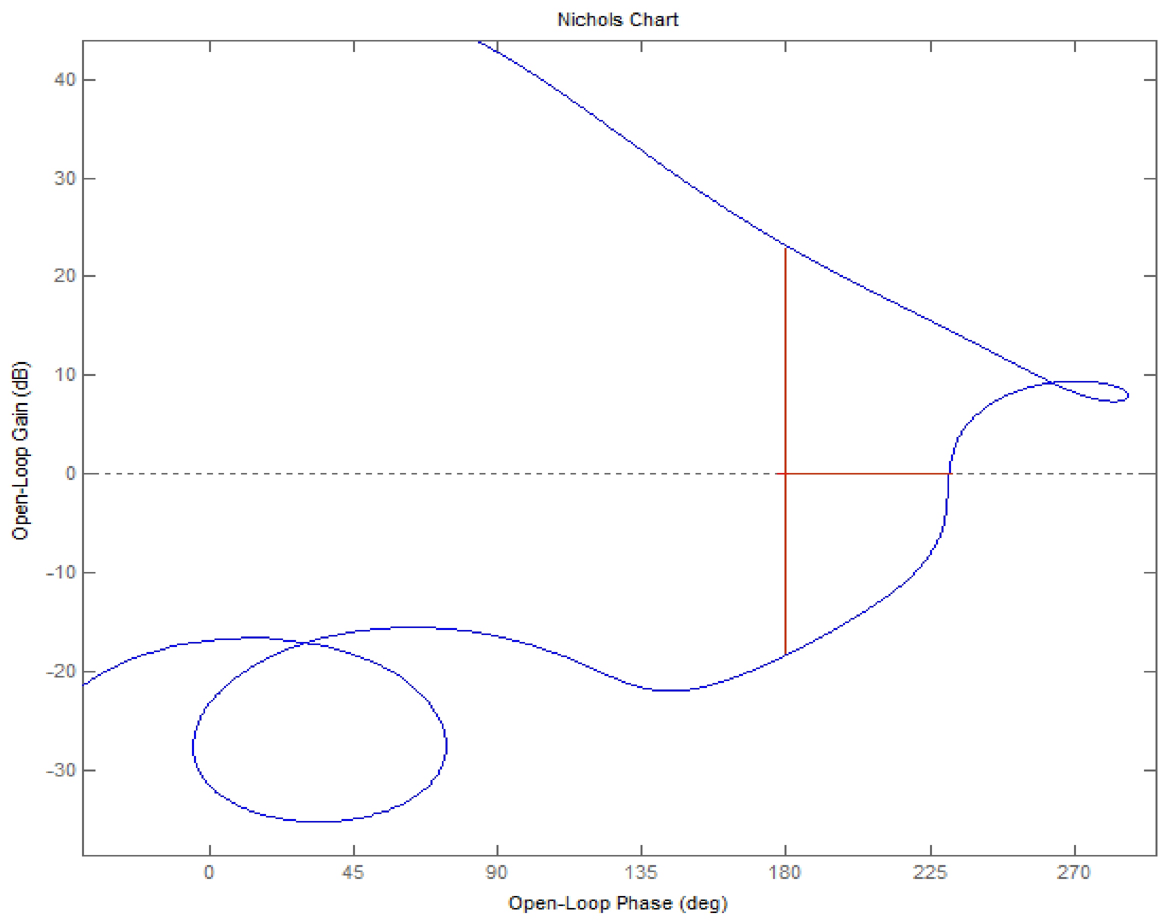


Pitch Loop Opened, Other Loops Closed

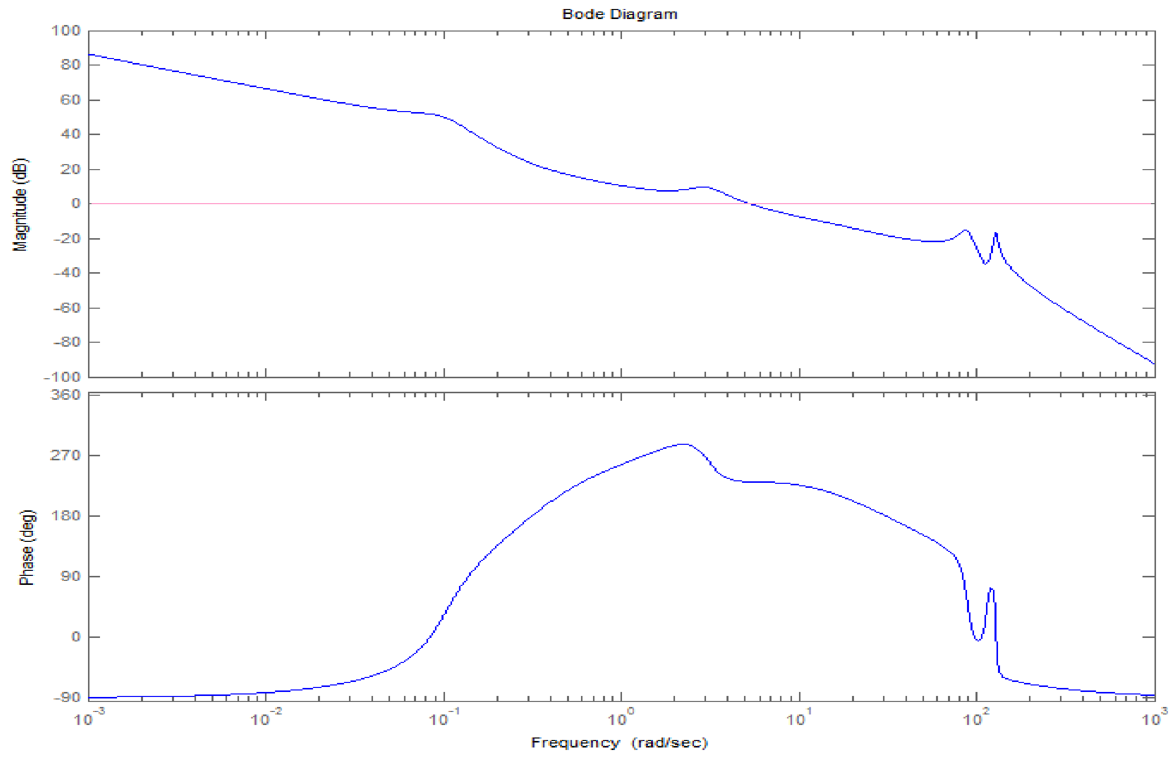


Roll Loop Opened, Other Loops Closed

Pitch Axis Stability



Pitch Axis Open-Loop Response



2.0 Flexible Vehicle Analysis

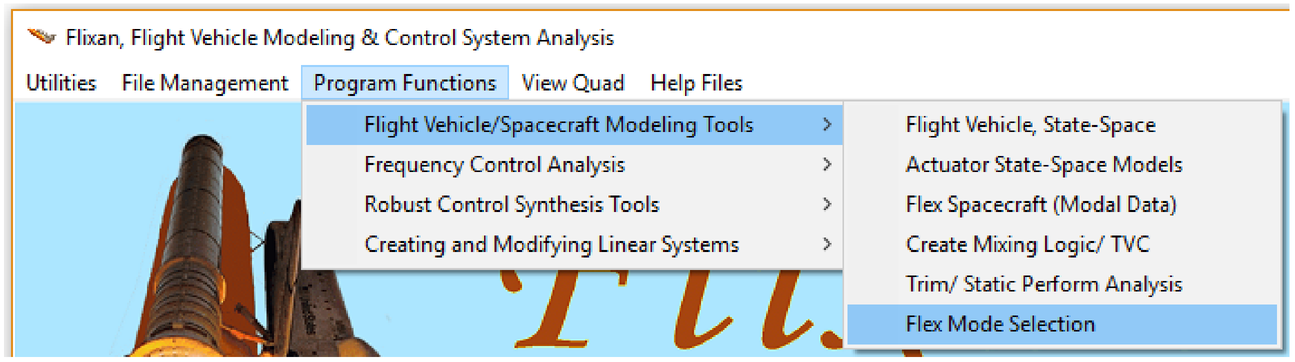
To analyze stability and performance of the flexible vehicle a more complex model is needed that will includes structural modes. The flight control system for the most part is similar to the rigid-body controller with some additional modifications in order to attenuate the flex modes. In the flex model we shall ignoring 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 to control the vehicle. The structure is excited by the angular accelerations of the TVC engines using the engine flex coupling coefficients $h_{ey(j,k)}$ (pitch only) and the angular accelerations of the control surfaces using the aerosurface flex coupling coefficients $h_{s(j,k)}$ as it is described in equation (2.3.4), ignoring the aero-elastic terms. The finite elements model must therefore include the flexibility of the aerosurfaces and engines, and it is calculated with the gimbals "locked" (no rotation dofs). The hinges and gimbals are released and coupled with the actuator models by the equations of motion which are described in section (2.3.1), and implemented in the vehicle modeling program. The inertial coupling coefficients are also used to calculate the moments at the pivots of the engines and aero surfaces due to local structural bending at, as described in equations (2.5-2 & 2.5-7). The purpose of this example is to demonstrate the flex modeling and analysis process. The analysis similar to the rigid-body and we shall compare the results with those obtained from the previous section.

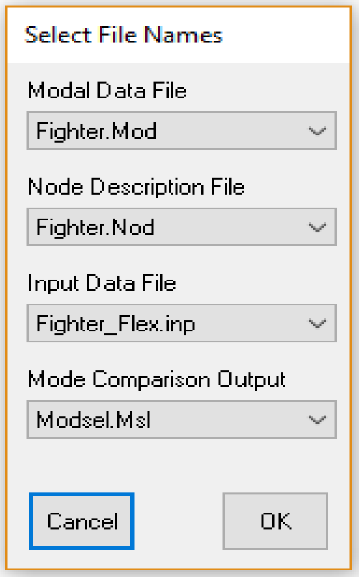
2.1 Input Data for the Flexible Aircraft

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 using the mode selection utility, and they have been converted from Nastran units and directions to units and directions compatible with the vehicle model. 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 h-parameters for the seven effectors are included in a separate "*Fighter.Gaf*" file. They are generated by the finite element modeling program and they are in units of moment of inertia (slug-ft^2). The flex coupling coefficients file should have an extension (.Gaf), be in the format shown, and it should 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 which are excited from the 3 aero-surfaces and the 2 engines in pitch. It does not include aero-elastic GAFD data as in other examples. The file includes the mode frequencies for the first 50 modes, followed by 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 permitted in this example and the yaw coupling coefficients are set to zero. In order 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. Otherwise, the program 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 and it will use the mode shapes and slopes at the hinges and gimbals. The vehicle modeling program will select the column data from the h-parameters file that correspond to the selected mode frequencies.

2.3 Mode Selection

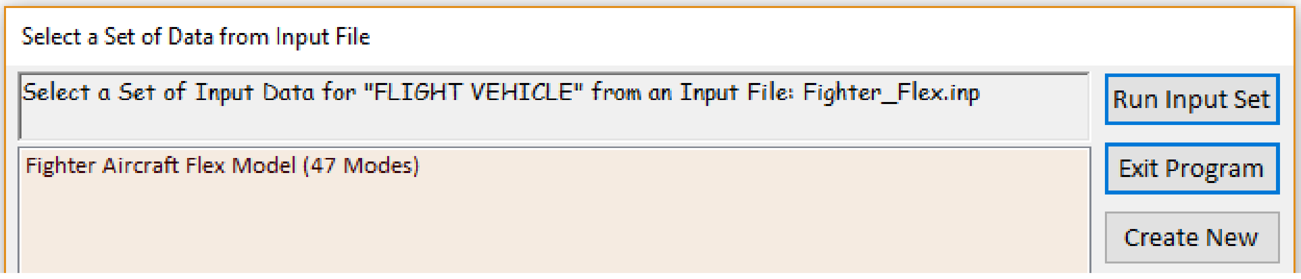
We will now go back to demonstrate the process of selecting the flex modes from the original Nastran generated modal data file "*Fighter.Mod*". The modal data file contains the aircraft mode shapes for the first 80 modes. The file "*Fighter.Gaf*", however, contains h-parameters data only for the first 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 is 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*".


Flixan, Flight Vehicle Modeling & Control System Analysis
Utilities
File Management
Program Functions
View Quad
Help Files
Flight Vehicle/Spacecraft Modeling Tools
Frequency Control Analysis
Robust Control Synthesis Tools
Creating and Modifying Linear Systems
Flight Vehicle, State-Space
Actuator State-Space Models
Flex Spacecraft (Modal Data)
Create Mixing Logic/ TVC
Trim/ Static Perform Analysis
Flex Mode Selection



Modal Data File: Fighter.Mod
Node Description File: Fighter.Nod
Input Data File: Fighter_Flex.inp
Mode Comparison Output: Modsel.Msl

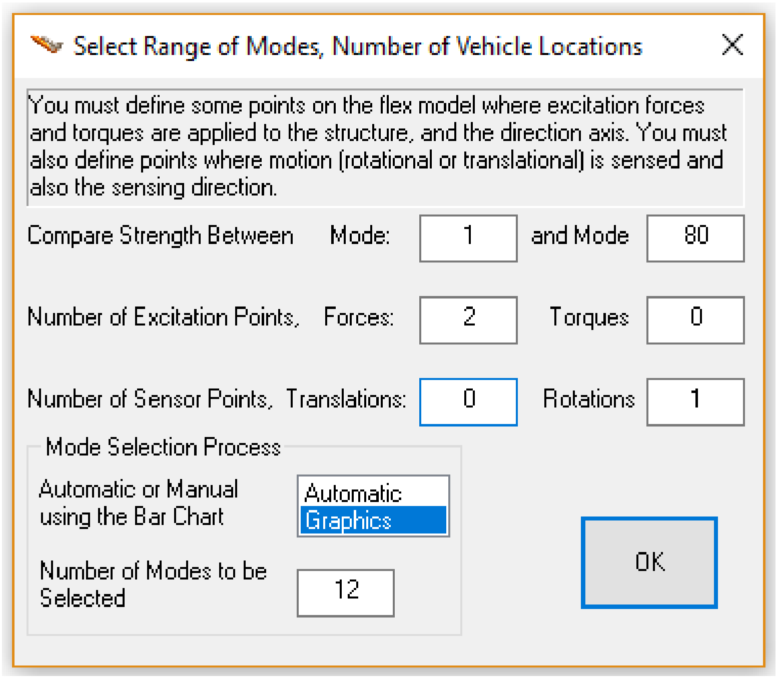
From the filename selection menu select the modal data file "*Fighter.Mod*", the input data file "*Fighter_Flex.Inp*", the nodes map file "*Fighter.Nod*", and the default output file "*Modsel.Msl*" where the mode comparison data will be saved, and click "*OK*". The nodes table is used in menus for selecting 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.



Select a Set of Input Data for "FLIGHT VEHICLE" from an Input File: Fighter_Flex.inp

Fighter Aircraft Flex Model (47 Modes)

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 will 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 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.


Select Range of Modes, Number of Vehicle Locations

You must define some points on the flex model where excitation forces and torques are applied to the structure, and the direction axis. You must also define points where motion (rotational or translational) is sensed and also the sensing direction.

Compare Strength Between Mode: 1 and Mode 80

Number of Excitation Points, Forces: 2 Torques 0

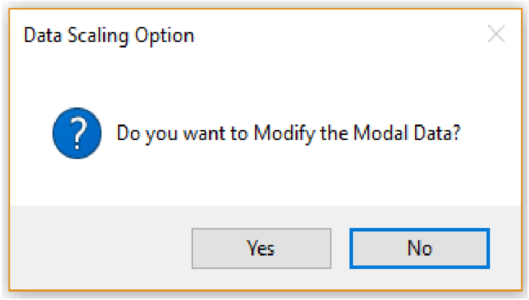
Number of Sensor Points, Translations: 0 Rotations 1

Mode Selection Process

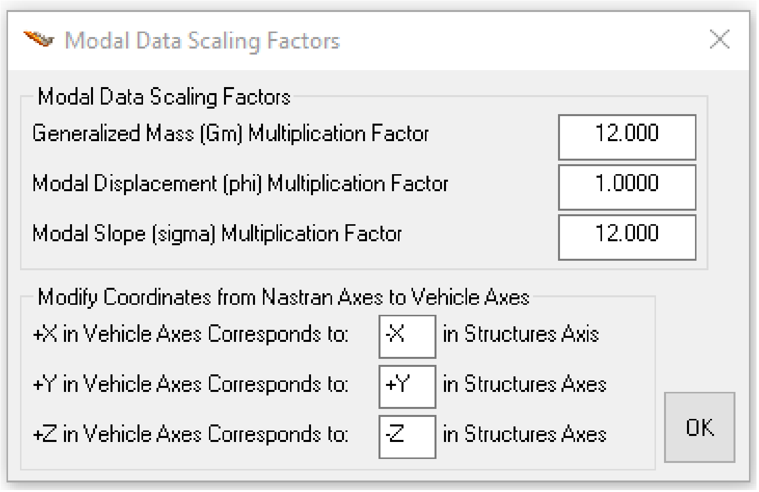
Automatic or Manual using the Bar Chart: Automatic / Graphics (Graphics selected)

Number of Modes to be Selected: 12

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).


Data Scaling Option

Do you want to Modify the Modal Data?


Modal Data Scaling Factors

Modal Data Scaling Factors

Generalized Mass (Gm) Multiplication Factor: 12.000

Modal Displacement (phi) Multiplication Factor: 1.0000

Modal Slope (sigma) Multiplication Factor: 12.000

Modify Coordinates from Nastran Axes to Vehicle Axes

+X in Vehicle Axes Corresponds to: -X in Structures Axis

+Y in Vehicle Axes Corresponds to: +Y in Structures Axes

+Z in Vehicle Axes Corresponds to: -Z in Structures Axes

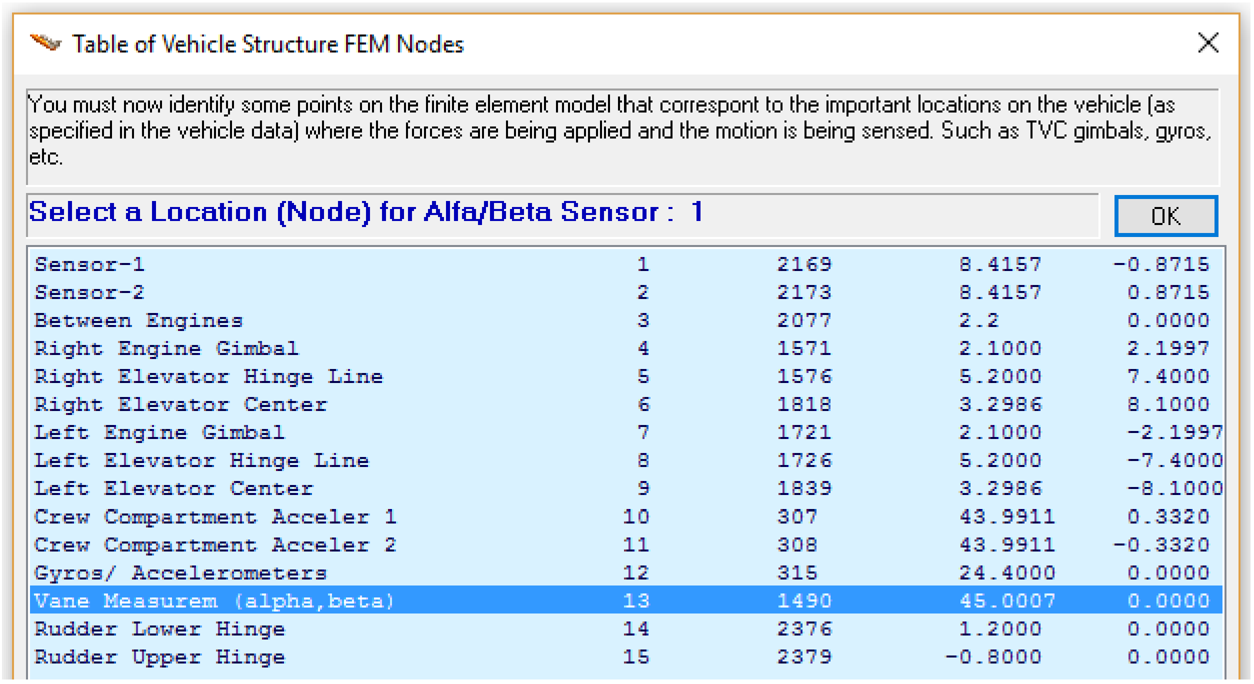


Table of Vehicle Structure FEM Nodes

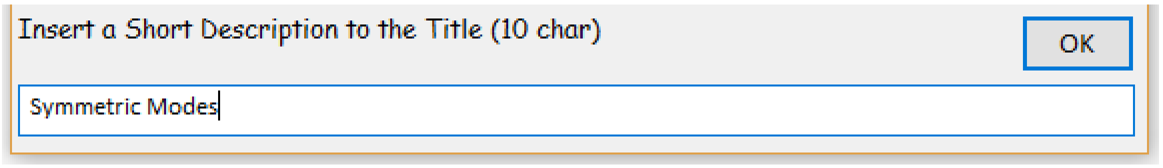
You must now identify some points on the finite element model that correspont to the important locations on the vehicle (as specified in the vehicle data) where the forces are being applied and the motion is being sensed. Such as TVC gimbals, gyros, etc.

Select a Location (Node) for Alfa/Beta Sensor : 1

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

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.

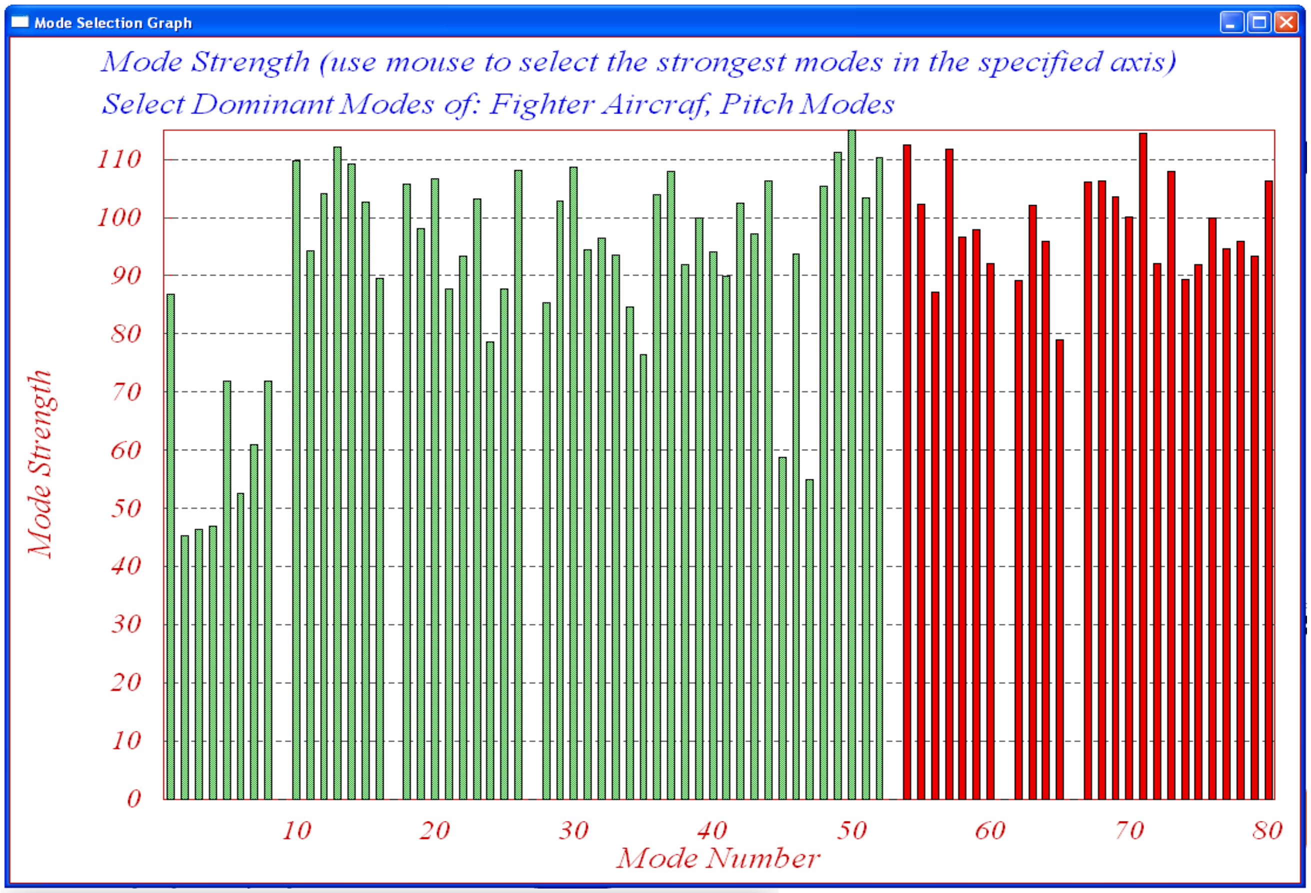
A short description of the selected modal data is required for documentation purposes. You may enter information, such as, conditions, excitation and sensor locations, directions, etc. It will be included in the title of the selected modes set, for example:

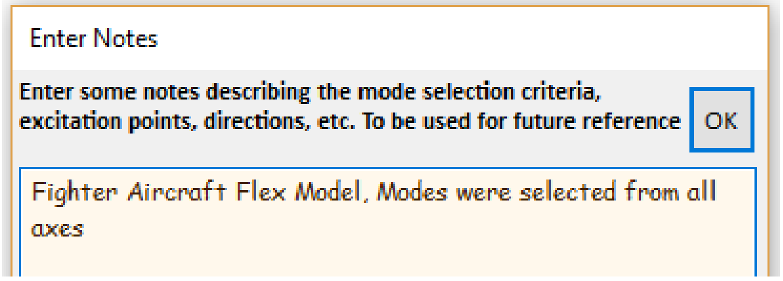


Insert a Short Description to the Title (10 char)

Symmetric Modes

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.




Enter Notes

Enter some notes describing the mode selection criteria, excitation points, directions, etc. To be used for future reference

OK

Fighter Aircraft Flex Model, Modes were selected from all axes

2.4 Modified Flex Aircraft Model

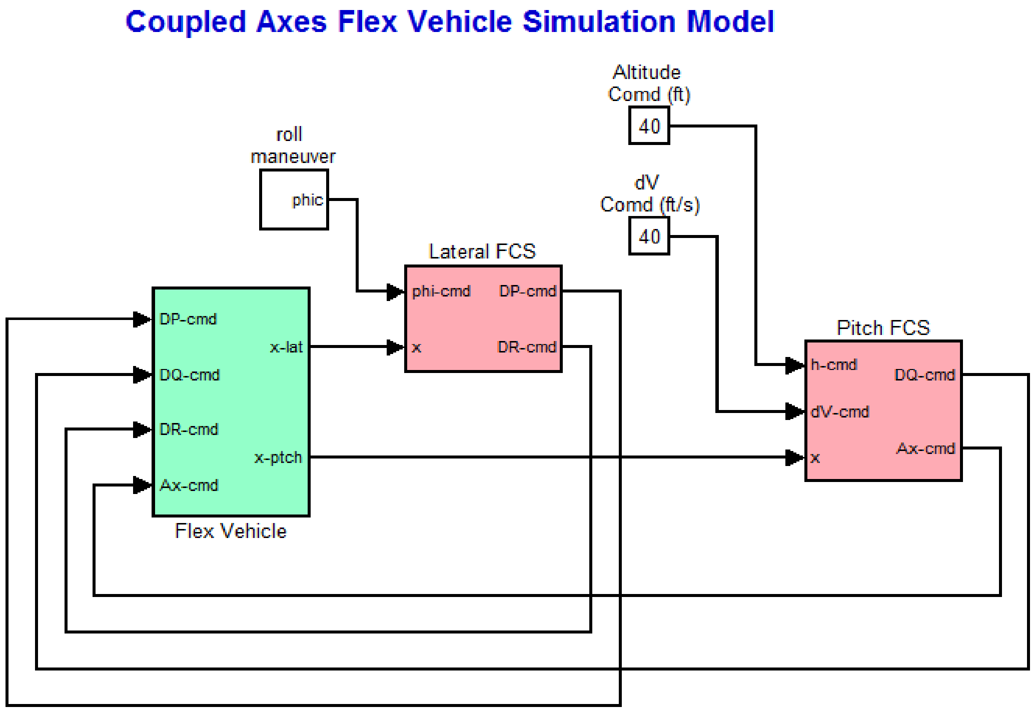
The flex vehicle model "*Fighter Aircraft Flex Model (47 Modes)*" is missing an important output that is needed for feedback in the control loop, the change in velocity (δV) relative to trim velocity V_0. This variable, however, is included in the states (state-10). We will use the Flixan system modification 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 from trim (V_0).

2.5 Actuator Models

The aerosurface and TVC actuator models used for flex analysis are the same type 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

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.



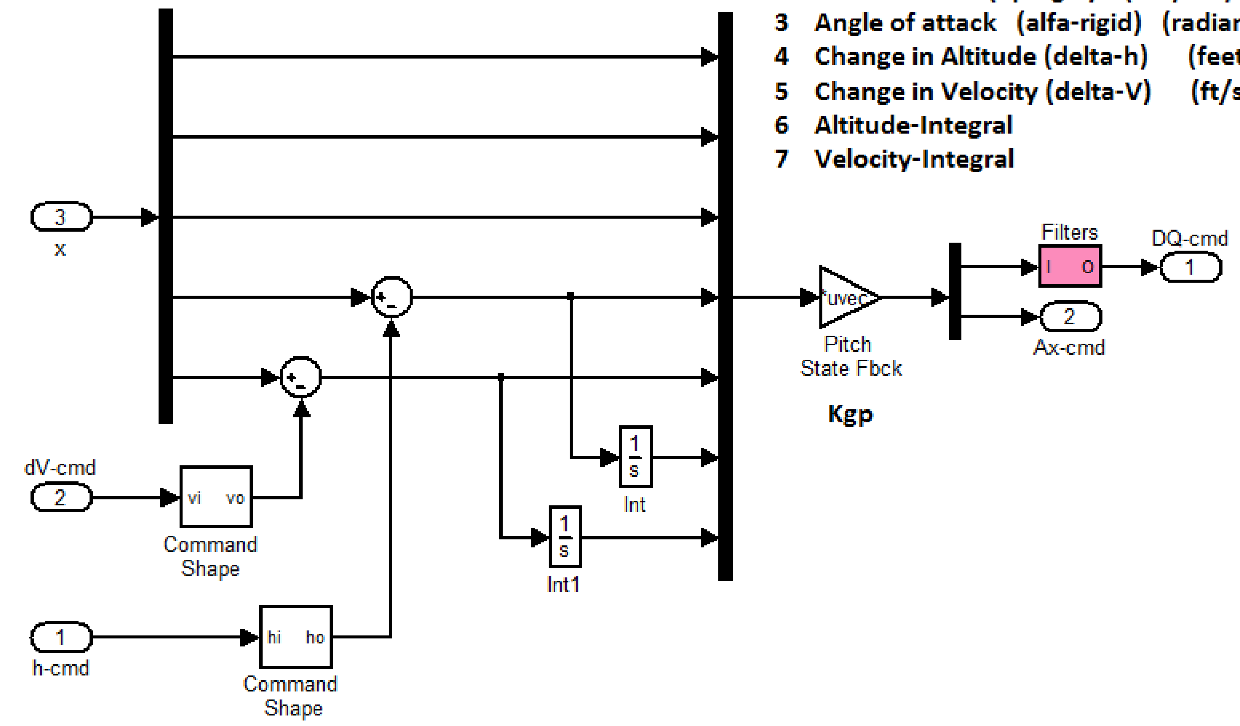
The three rotational and one axial acceleration flight control commands (DP, DQ, DR, Ax)$_{cmd}$ are converted by the mixing logic matrix Kmix4 into control surface deflection commands, engine gimbal deflection commands, and also thrust variation commands for the two 30,000 (lb) engines. The commands 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.

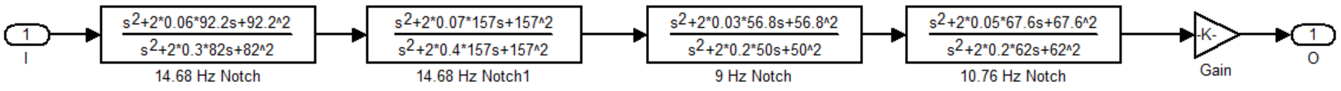
The pitch and lateral flight control systems were derived in the previous section using the LQR method and the rigid aircraft model. They are basically state-feedback controllers including notch and low-pass filters to attenuate structural flexibility in the pitch and yaw axes, as shown below. We will use the above Simulink model to repeat the two aircraft simulations that were demonstrated in the previous rigid-body analysis.

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		

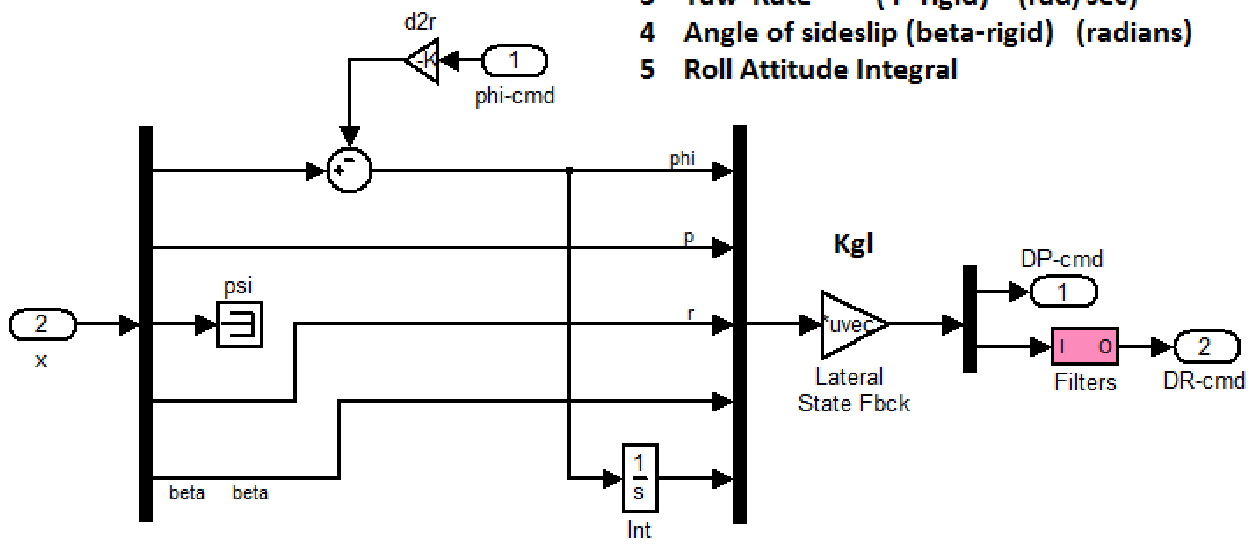



1
I
s^2+2*0.06*92.2s+92.2^2 / s^2+2*0.3*82s+82^2
14.68 Hz Notch
s^2+2*0.07*157s+157^2 / s^2+2*0.4*157s+157^2
14.68 Hz Notch1
s^2+2*0.03*56.8s+56.8^2 / s^2+2*0.2*50s+50^2
9 Hz Notch
s^2+2*0.05*67.6s+67.6^2 / s^2+2*0.2*62s+62^2
10.76 Hz Notch
-K-
Gain
1
O

Lateral Flight Control

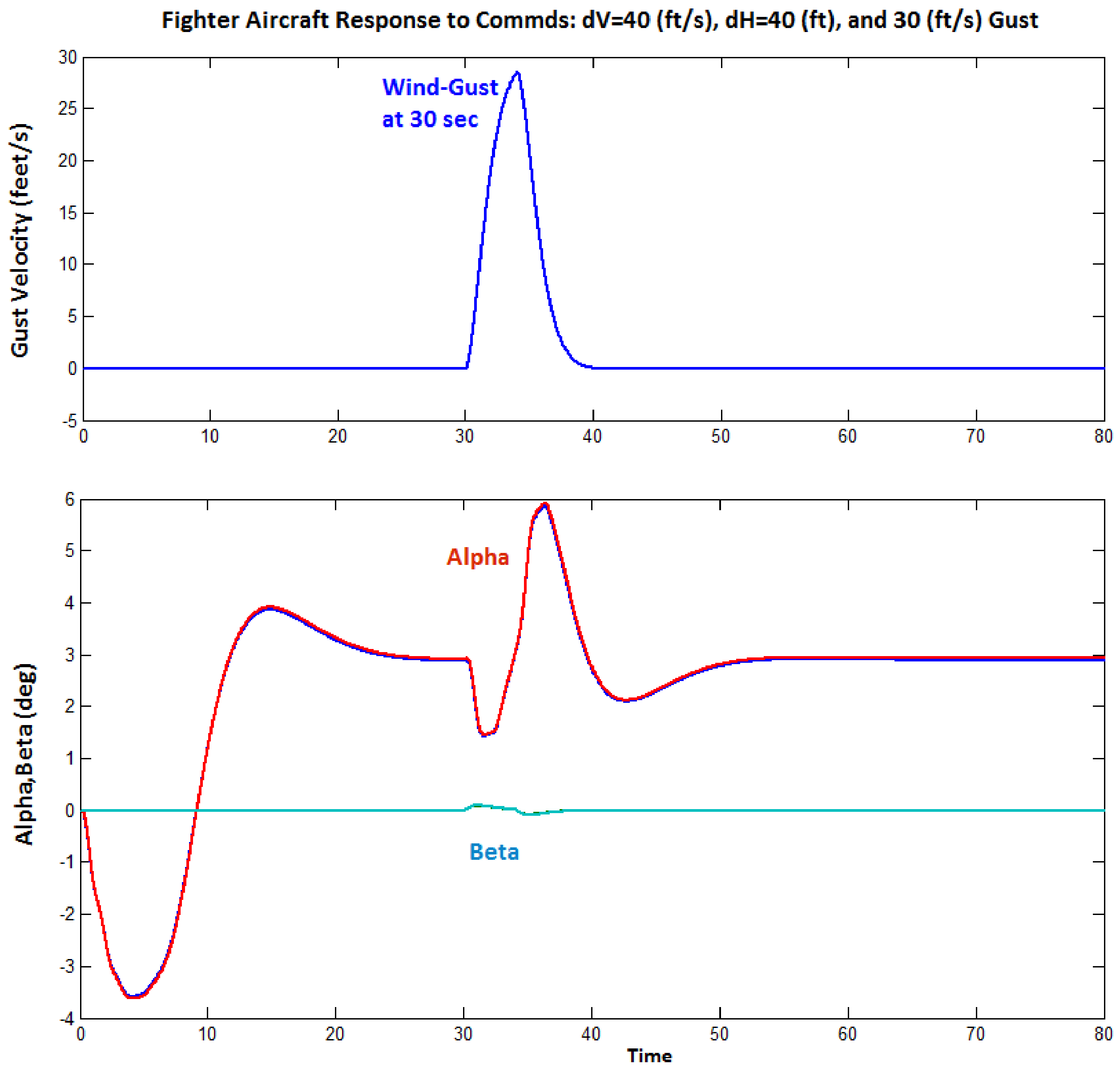
States/ Outputs

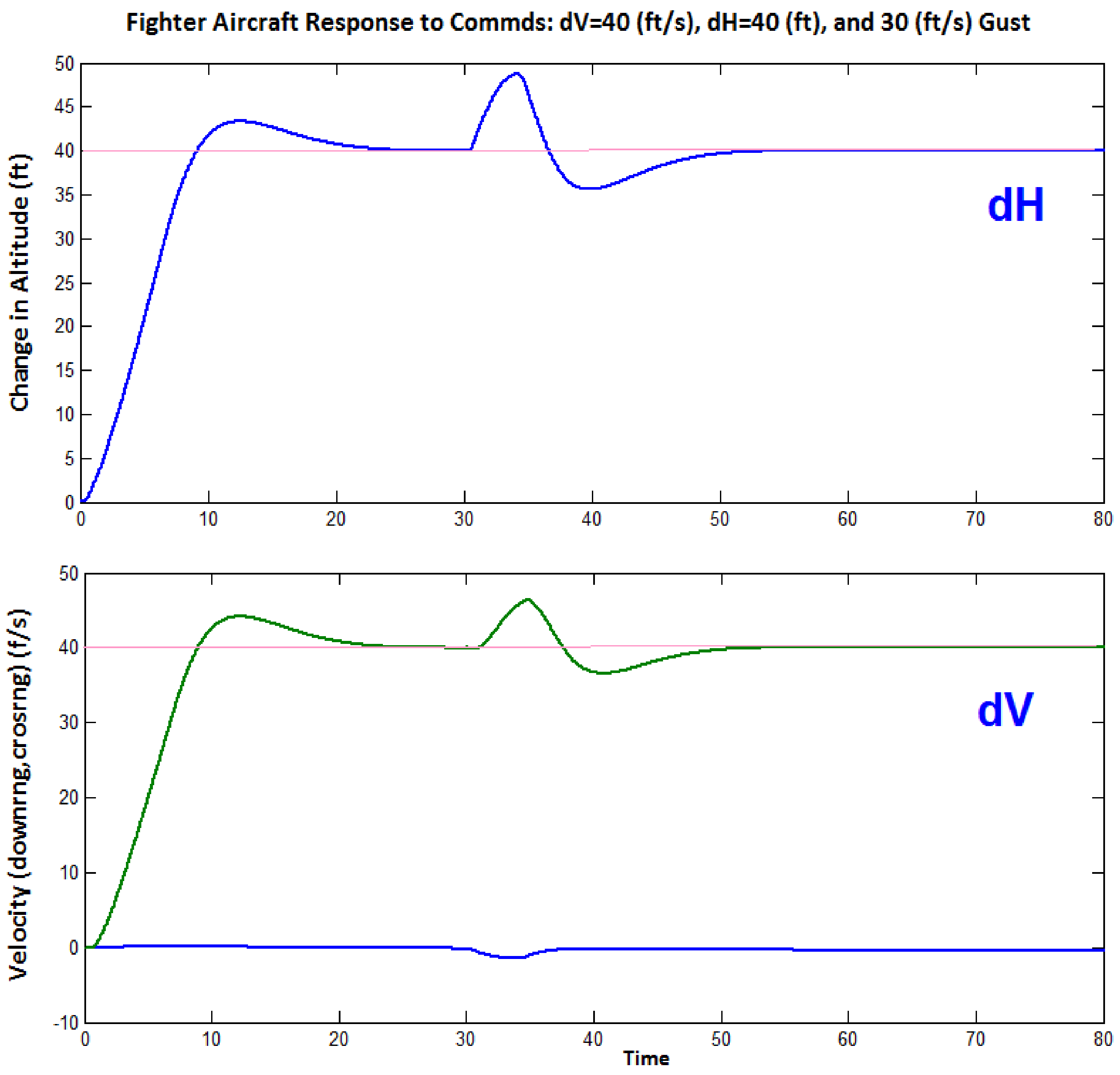
1	Roll Attitude	(phi-rigid)	(radians)
2	Roll Rate	(p -rigid)	(rad/sec)
3	Yaw Rate	(r -rigid)	(rad/sec)
4	Angle of sideslip	(beta-rigid)	(radians)
5	Roll Attitude Integral		



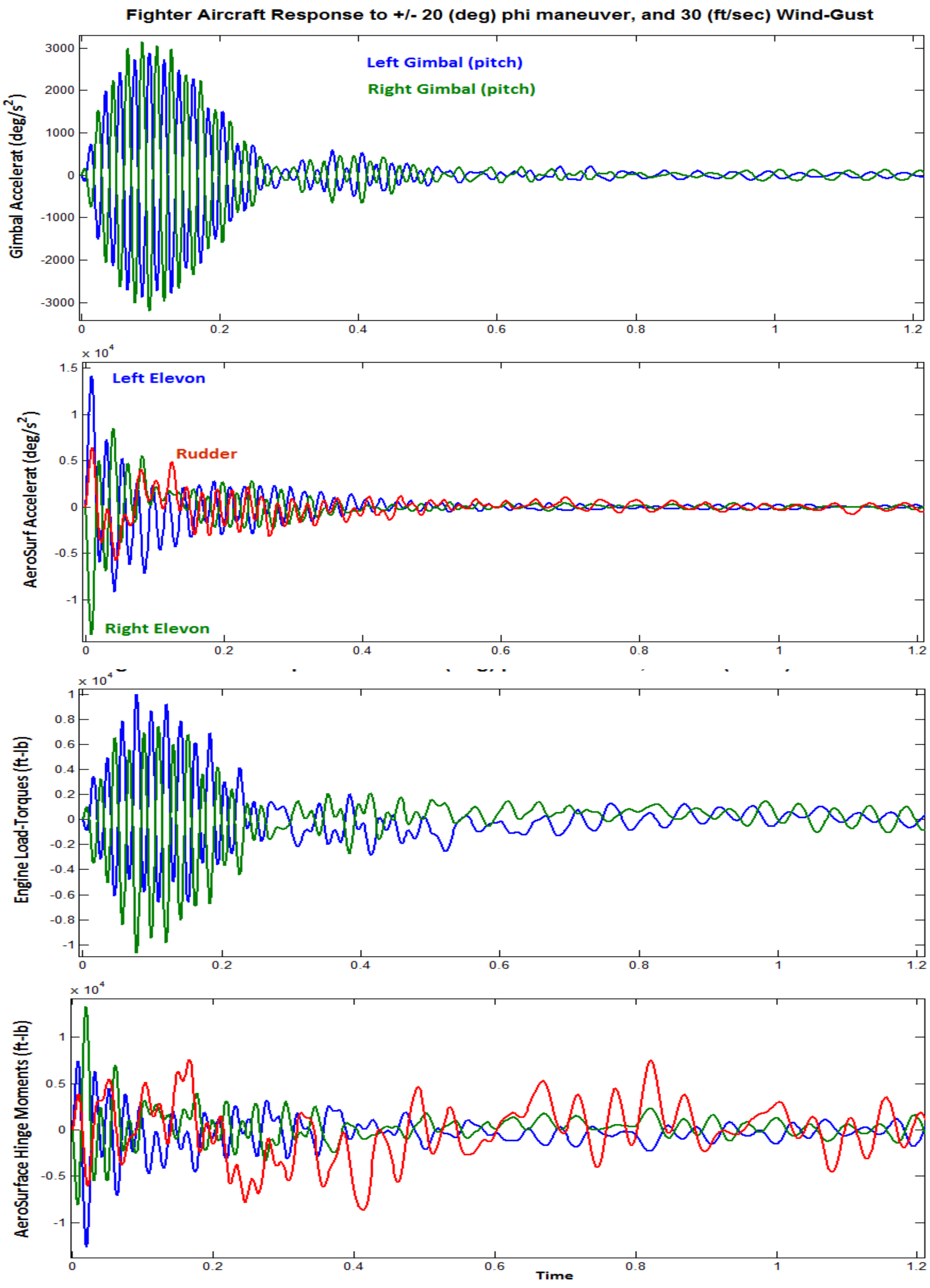
Pitch Maneuver

The first simulation performs a longitudinal maneuver where the vehicle is simultaneously commanded to increase its speed (δV=40 feet/sec), and also raise its altitude to (δH=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.





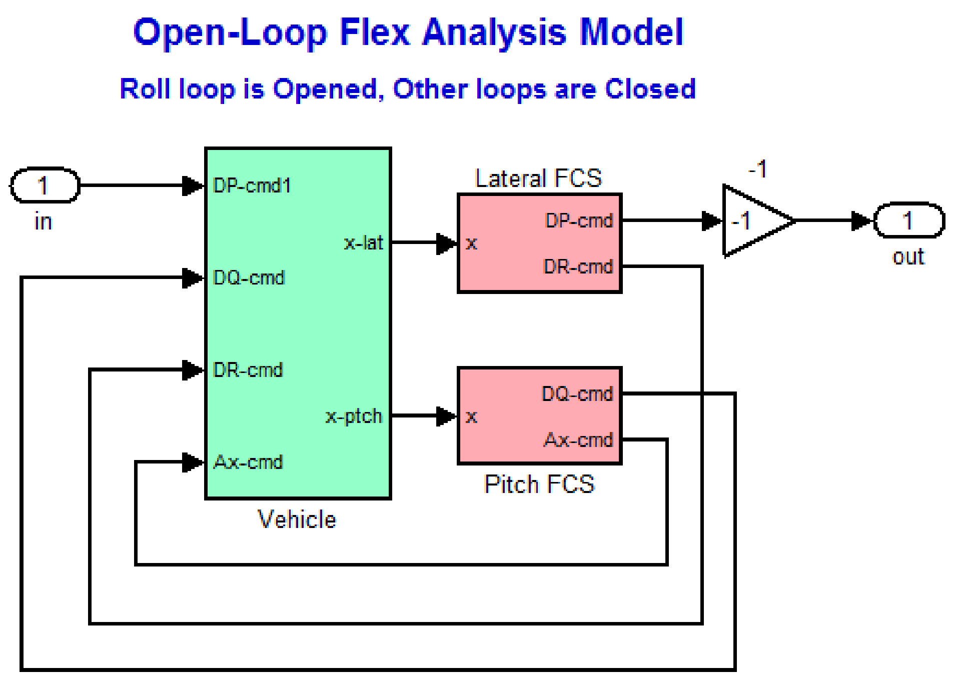
The next figure shows the aircraft body rates and attitude. The motion is mainly in pitch (green). The wind-gust also causes a small transient in the lateral direction (red and blue). The next plot shows the load-torques at the two engine gimbals, and the hinge moments at the three aerosurfaces which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

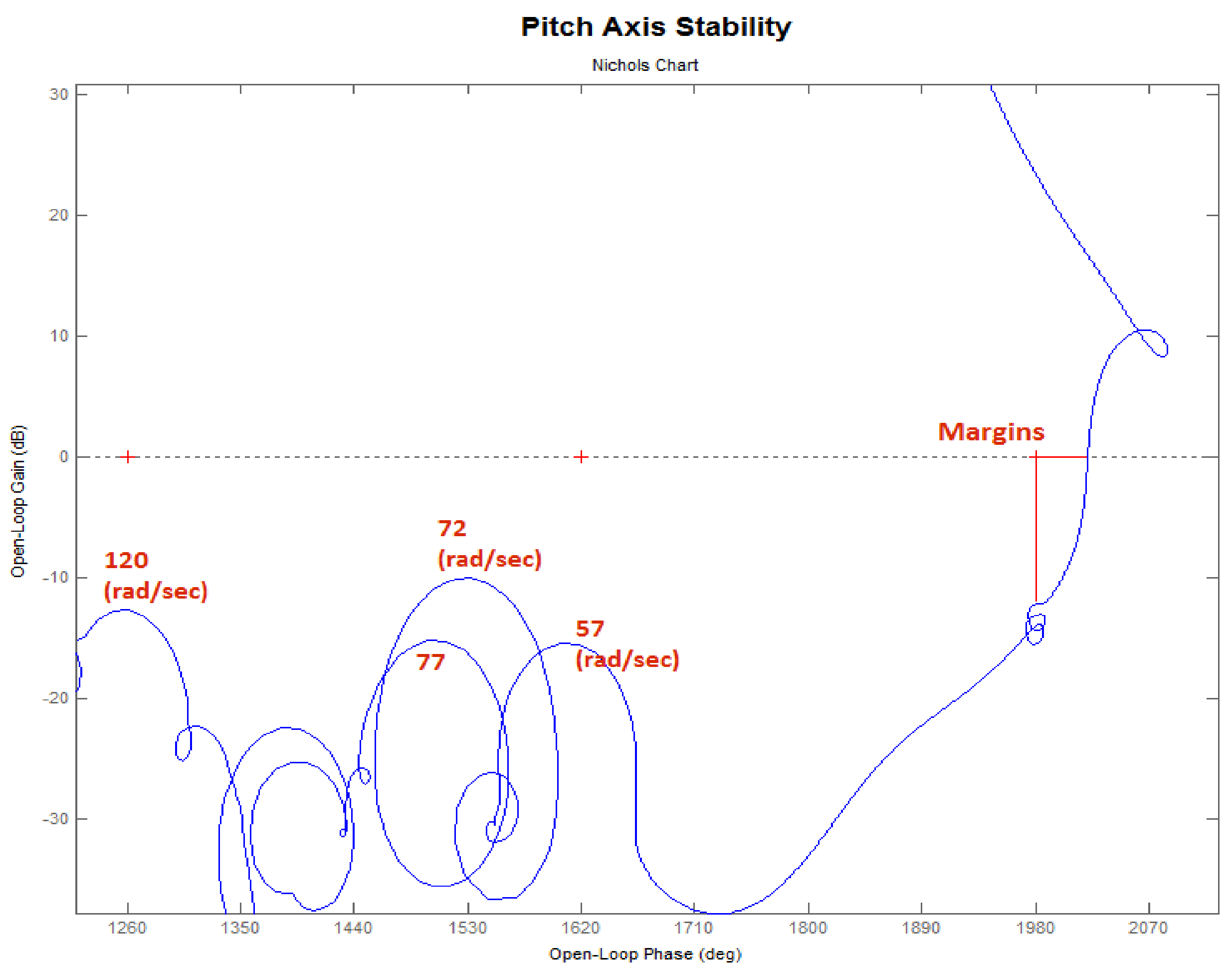

Fighter Aircraft Response to +/- 20 (deg) phi maneuver, and 30 (ft/sec) Wind-Gust
Left Gimbal (pitch)
Right Gimbal (pitch)
Gimbal Accelerat (deg/s^2)
Left Elevon
Rudder
Right Elevon
AeroSurf Accelerat (deg/s^2)
x 10^4
Engine Load-Torques (ft-lb)
x 10^4
AeroSurface Hinge Moments (ft-lb)
x 10^4
Time

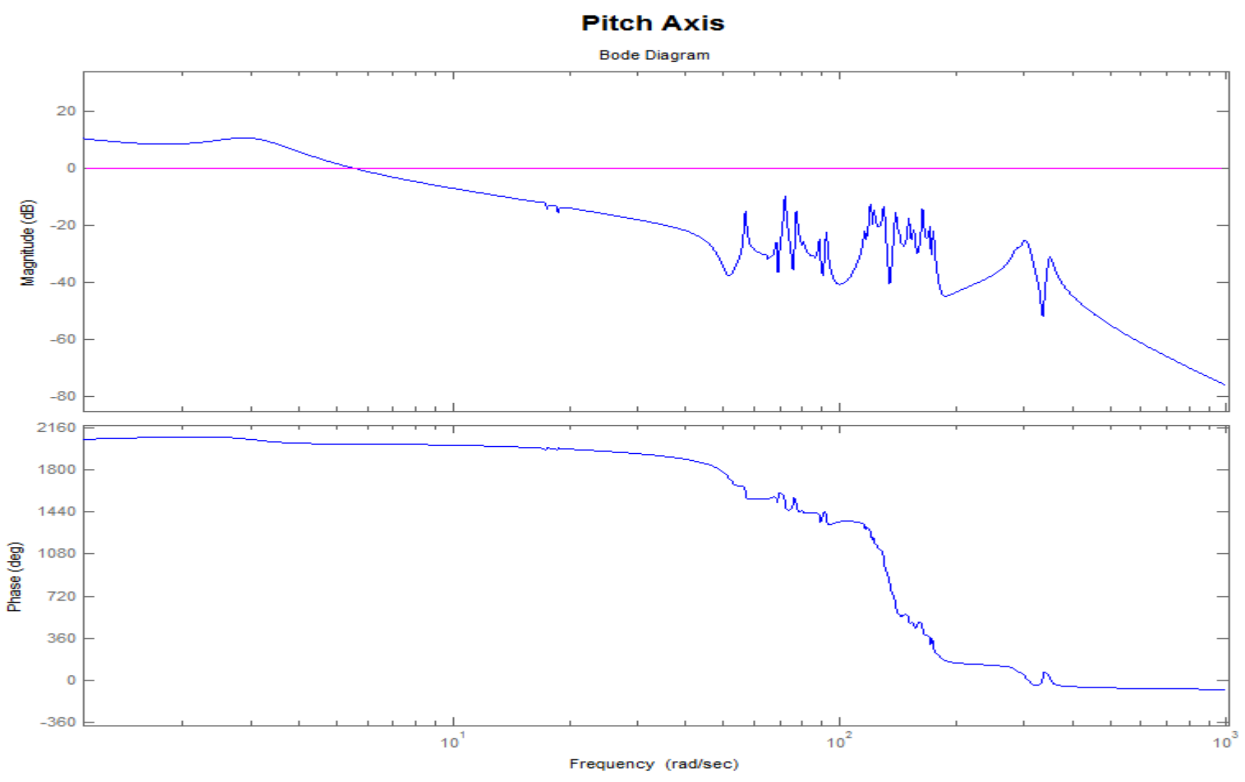
2.8 Stability Analysis

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)
```




Pitch Axis Stability
Nichols Chart
Open-Loop Gain (dB)
Open-Loop Phase (deg)
30
20
10
0
-10
-20
-30
1260
1350
1440
1530
1620
1710
1800
1890
1980
2070
Margins
72 (rad/sec)
120 (rad/sec)
77
57 (rad/sec)


Pitch Axis
Bode Diagram
Magnitude (dB)
Phase (deg)
Frequency (rad/sec)
20
0
-20
-40
-60
-80
2160
1800
1440
1080
720
360
0
-360
10^1
10^2
10^3