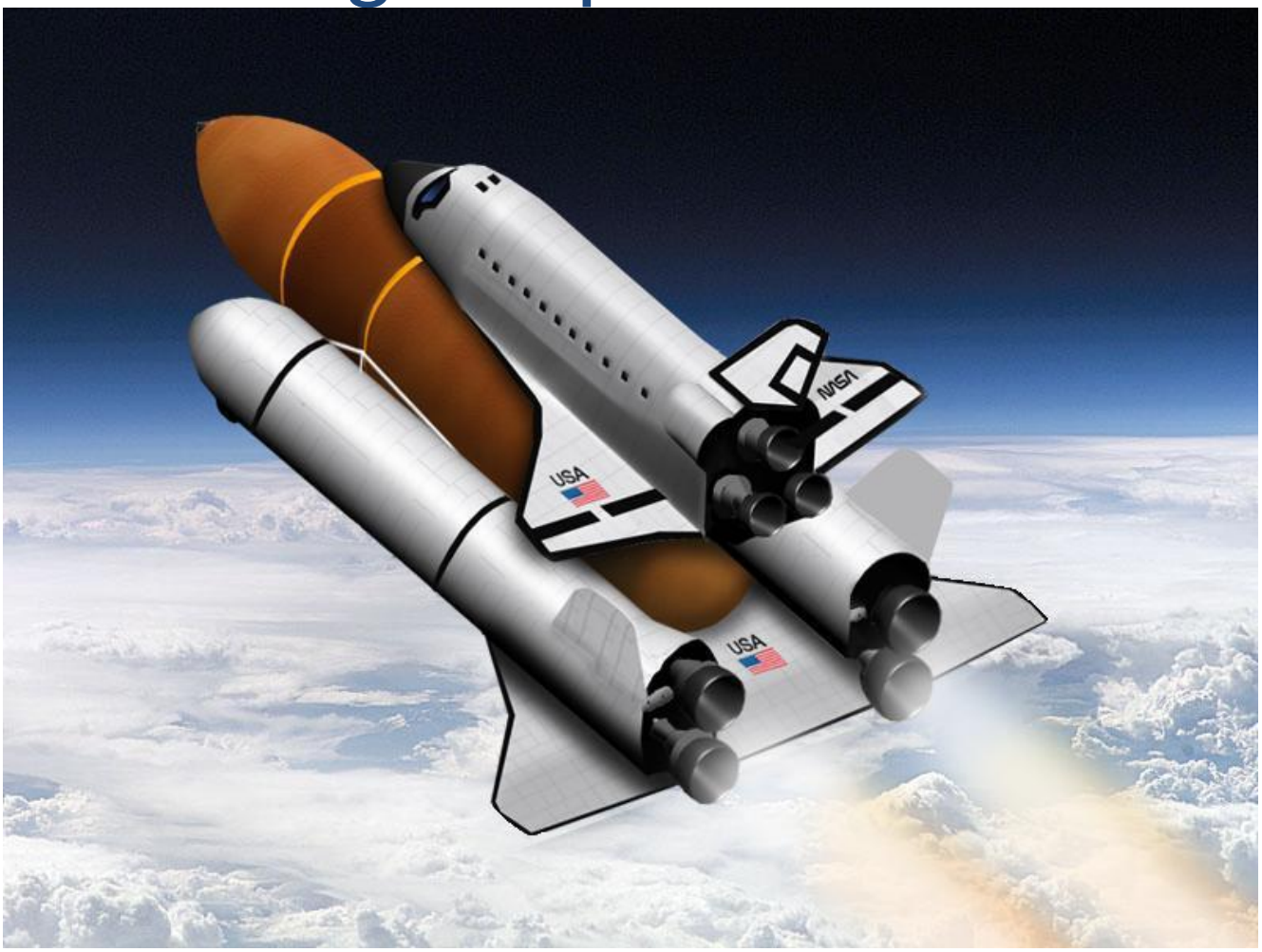
Shuttle Vehicle with Reusable First Stage Liquid Booster



In this example we have a Space Shuttle vehicle type during first stage that is attached to an airplane like single booster with multiple engines. After first stage separation the booster has enough propellant to return to the launch site and lands like an airplane. This example is intended to demonstrate the design and analysis of a thruster engine steering logic that uses thrust vectoring and differential throttling together, to control the vehicle. It also demonstrates the capability of the TVC to reconfigure itself in the event of an engine failure. The Shuttle vehicle is attached to the external tank and the tank is attached to the first stage booster aircraft. The booster is similar to the picture above except that it uses six engines, three on each side. The three Shuttle main engines are also thrusting during first stage. We will analyze only one flight condition, at max dynamic pressure, because it is the most critical for stability and structural loading. The methodology, however, can be applied to other flight conditions.

At Max-Q the nominal thrust of each booster engine is 961,000 (lb), but they are capable of varying their thrusts above and below nominal thrust, that is ±45%. The thrust can, therefore, vary from a lowest value of 480,000 (lb) to a maximum thrust of 1.4 million (lb). In this example we will

demonstrate how to take advantage of the variable engine thrust capability for steering the vehicle by means of combining differential-throttling and thrust vector control (TVC). Differential-throttling is more effective in the yaw axis because of the long y-moment arm. The example also demonstrates how we can take advantage of the multiple engine capability to design a mixing-logic algorithm that can be modified in the event of an engine failure. This adjustment is independent of the flight control law which remains unchanged. When an engine thrust drops to zero it no longer responds to the gimbaling or throttling demands from the mixing logic, and the mixing-logic gains are automatically adjusted to accommodate the new thruster configuration in order to provide the torques required by the flight control system. The control system is not affected by the failure. In the event of an off-centered engine failure the thrusts of the remaining booster engines are adjusted to balance the thrust on both sides of the vehicle in order to avoid sideslip loads. The remaining two engines thrust on the failed side are increased and the thrusts of the three engines on the other side are reduced to equalize the thrusts on both sides.

In the analysis that follows we will use the Flixan program to create a 9 engine flight vehicle model that uses TVC for all 9 engines, with an additional throttle control in the 6 liquid booster engines. We will also derive a mixing logic matrix that combines TVC and differential throttling for vehicle steering. We will also create a vehicle model that simulates an engine failure by reducing one of the liquid booster engines thrust on the right side to zero and derive a mixing-logic matrix for the failed engine configuration. We will also introduce multiple propellant sloshing tanks in the dynamic model using spring-mass models for each tank. Then we will combine the TVC with the vehicle models, the actuators, and the flight control system, and will develop analysis and simulation models for analyzing the system stability using classical frequency domain analysis methods in Matlab. The analysis is repeated using both: nominal vehicle and engine out configurations. We will show that the stability margins and vehicle performance are not affected by the configuration change because the mixing logic adjustment is capable of compensating against the engine failure.

1 Analysis Files

The Flixan modeling files for this example are in directory "*…\Examples\ Multi-Engine First-Stage Liquid Booster*". This folder contains an input data file "*Multi-Eng-RLB.Inp*" that includes the launch vehicle data without engine failures. The same input file includes also actuator data, the flight control system, and also system interconnection instructions. There is also a similar file "*Eng-Out-RLB.Inp*" that includes input data for the vehicle with the failed engine. On the top of the input files there are batch instruction datasets for batch mode execution. Structural flexibility is not introduced in this analysis because the emphasis here is in mixing-logic design and multiple sloshing tanks. The directory folder also includes two system files generated by Flixan. They contain the state-space systems for the two vehicle configurations. The file "*Multi-Eng-RLB.Qdr*" contains the systems and matrices for the nominal vehicle, and file "*Eng-Out-RLB.Qdr*" contains the systems of the engine-out vehicle. The engine mixing logic matrices and the interconnection systems are also saved in the system files.

The Matlab analysis and Simulations are performed in two separate subdirectories "… *\Multi-Engine First-Stage Liquid Booster\Mat_Frequ*", and "*Mat_Sim*". The first one is used for frequency domain analysis and the second one is for time domain simulations. The subfolders include the nominal and engine-out vehicle systems that were exported as Matlab function files, "*vehicle.m*", and "*vehicle_eo.m*". They also include the mixing logic matrices for the nominal and the engine-out vehicles, "*ThVC.Mat*" and "*ThVC_eo.Mat*", that were created by the mixing logic program and were

exported as Matlab matrices from the system files. They also include the pitch and lateral flight control systems (FCS) combined as a single continuous system *"fcs.m"*. It was exported into Matlab from the systems file "*Multi-Eng-RLB*.Qdr". The gimbal TVC actuators for the 9 engines in file "*comb_actuator.m*". The combined open-loop system (vehicle, actuators, TVC and FCS) for the nominal vehicle is "*openloop.m*", and for the engine-out vehicle is "*openloop_eo.m*". The combined closed-loop systems are "*closedloop.m*" and "*closedloop_eo.m*". These systems were combined together using Flixan and were converted to Matlab format for analysis. The subdirectory also includes a Matlab file "run.m" for loading the systems and matrices to Matlab and performing frequency response analysis. There are also Simulink models for time-domain simulations and frequency domain analysis. The files "pl.m" and "pl2.m" are used for plotting the simulations data.

2 Flight Vehicle Models Description

The input data for the vehicle model with all 9 engines running is in file "*Multi-Eng-RLB.Inp*". The title of the vehicle dataset is "*Throttable Multi-Engine Launch Vehicle at Max-Q, T=72 sec*". The vehicle modeling program processes this input dataset to create the vehicle state-space model at MaxQ. The flags below the title and comments are set for body rates outputs and Euler angle attitudes. The top part of data consists of mass properties, trajectory data, and aero coefficients. The vehicle parameters were obtained from a trajectory point mass simulation during maximum dynamic pressure of 684 (psf). Below the aero data we have the 9 engines. Three Space Shuttle main engines that provide 471,000 (lb) of thrust each and six liquid booster engines that provide 961,000 (lb) of nominal thrust. All 9 engines are defined as "Gimbaling" which means that they can gimbal in the pitch and yaw directions. The three SSME thrusts cannot be varied because the nominal and maximum thrusts are the same value, but they can gimbal to a maximum of ±10 degrees in pitch and yaw. The six liquid booster engines are also defined as "Throttling", which means that they can also vary their thrust ±45% from nominal. That is, from a minimum of 522,000 (lb), to a maximum of 1.4 million (lb). The amount of thrust variation above nominal is defined in the data by the nominal thrust and maximum thrust, which are not the same in the liquid boosters. The thrust difference defines the percentage of thrust variation that can be used to control the vehicle, in addition to gimbaling. It is assumed that the variation below nominal is the same percentage. The six booster engines can also gimbal to a maximum deflection of ±5 degrees in pitch but only ±1 degree in yaw because yaw can easily be controlled by differential throttling.

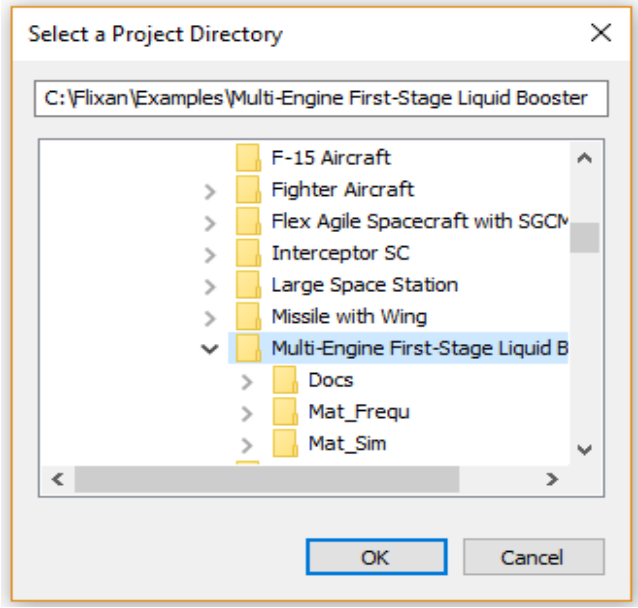
The vehicle is defined to have 6 sloshing propellant tanks. The slosh data consist of: the slosh mass, the slosh frequencies (1g) in two directions perpendicular to the acceleration vector, the slosh damping coefficients in two directions, and the location of the slosh mass (un-deflected). Note that, the slosh frequencies must be defined at 1g acceleration. The slosh frequency is automatically scaled proportionally to the square-root of the vehicle acceleration. Flexibility was not included in the analysis because it is very similar to other examples. The emphasis is rather on modeling and analyzing propellant sloshing with multiple tanks, combining TVC with differential throttling, and implementing/ analyzing engine failures.

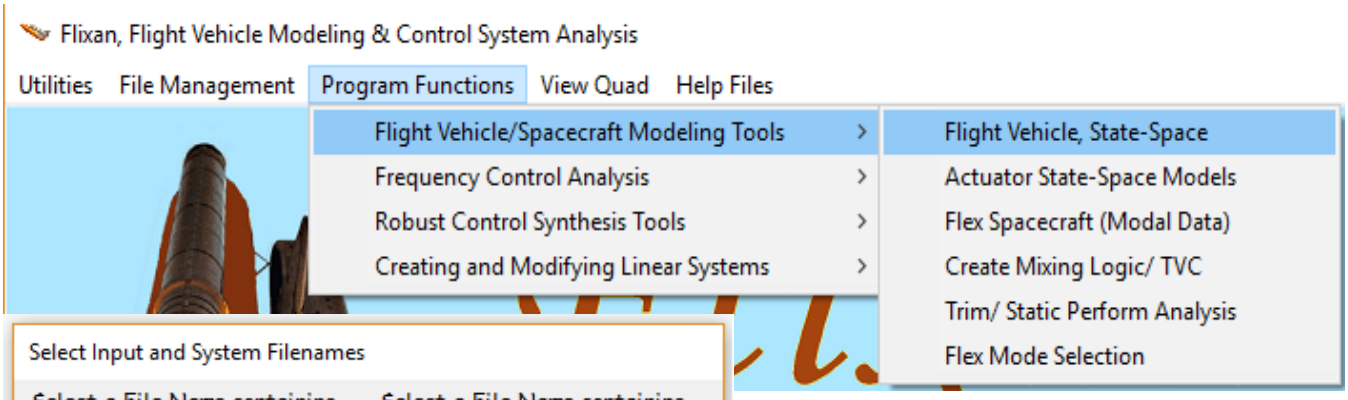
The inputs to the state-space dynamic model are: pitch and yaw TVC deflections in radians. There are also 6 throttle control inputs for the 6 booster engines. A throttle input does not represent thrust variation but it is defined as the ratio of thrust variation divided by the nominal engine thrust. In the simulations, therefore, a throttle input of +1 is equivalent to an input thrust equal to twice the nominal thrust and a throttle input of -1 is equivalent to zero thrust. In this example, however, the throttle control inputs to the dynamic model must not exceed ±0.45 since the booster engines thrust can only

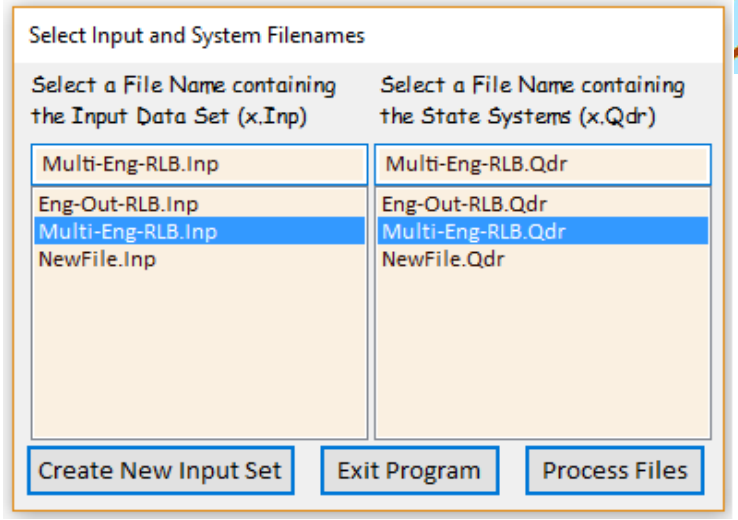
vary up to a maximum of ±45% from nominal thrust. There are attitude and rate gyro outputs for flight control and two accelerometer outputs: a normal accelerometer Nz and a lateral accelerometer Ny, which are used by the FCS in the implementation of the load-relief system. The tail-wag-dog option was introduced in all 9 engines. This generates additional gimbal acceleration inputs (rad/sec) in the vehicle model in order to implement the TWD dynamics and requires additional interconnections between the actuator gimbal acceleration outputs and the vehicle model inputs. The TWD option also introduces pitch and yaw load-torque outputs at the TVC gimbals. These outputs torques from the dynamic model are used to implement the load-torque mechanical feedback in the actuator models.

3 Generating the Launch Vehicle Model

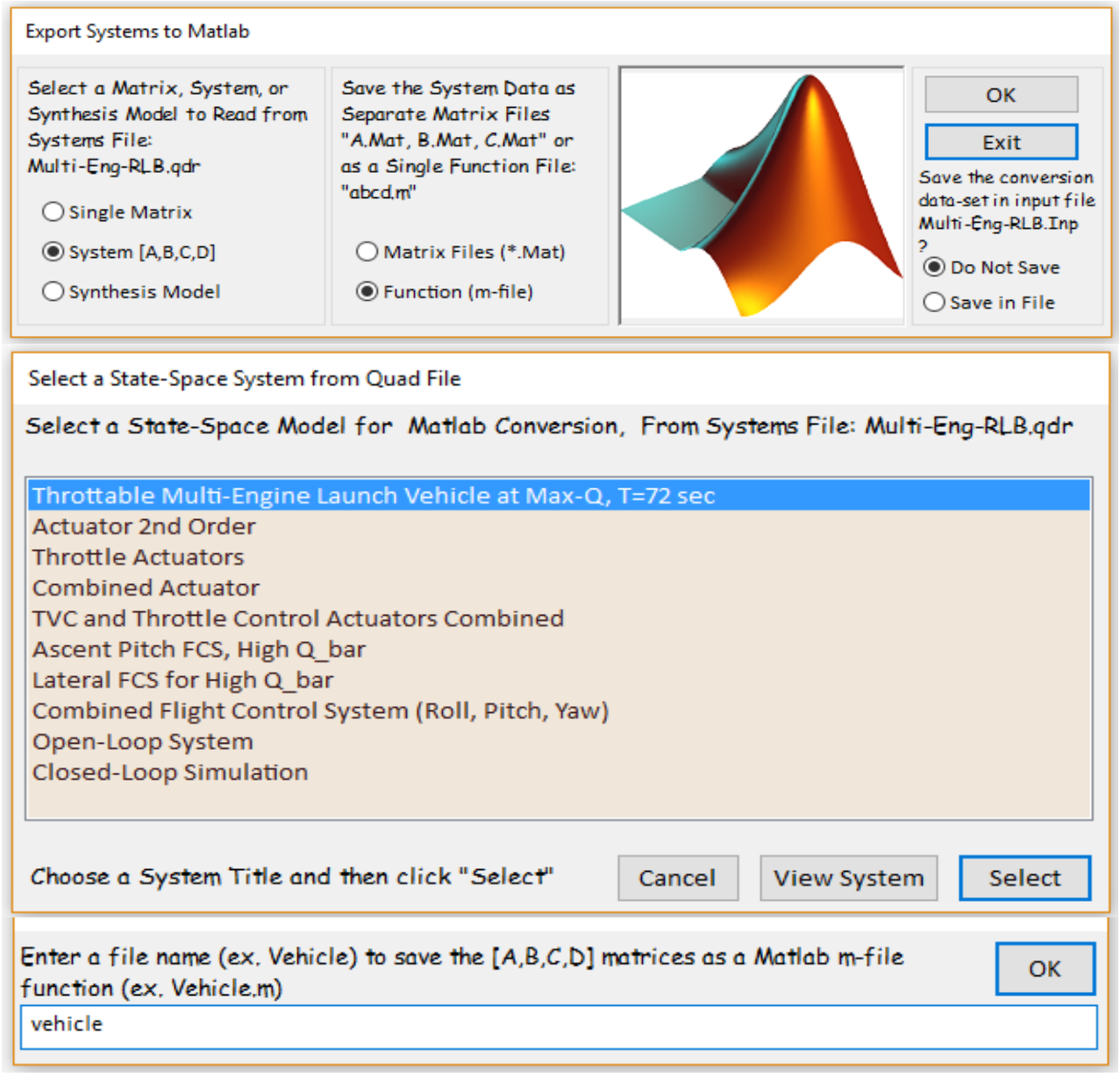
Our first step is to use the flight vehicle modeling program to process the launch vehicle dataset from the input file and generate the vehicle state-space model that will be saved in the systems file. Start the Flixan program and select the project directory *"Flixan\Examples\Multi-Engine First-Stage Liquid Booster"*. From the Flixan main menu select "*Program Functions*", then "*Flight Vehicle/ Spacecraft Modeling*", and then "*Vehicle State-Space*". From the filenames selection menu select the input and system files: "*Multi-Eng-RLB.Inp*" and "*Multi-Eng-RLB.Qdr*" respectively and click on "*Process Files*".







In the next menu specify a "*System (A, B, C, D)*" to be converted as a "Function m-file". The next menu shows the state-space systems which are included in file "*Multi-Eng-RLB.Qdr*". Select the first system "*Throttable Multi-Engine Launch Vehicle at Max-Q, T=72 sec*" to be converted to Matlab format. In the next dialog type the name of the Matlab function m-file, "vehicle" in order to create a state-space function m-file "*vehicle.m*" for the nominal vehicle, and click "OK. The vehicle quadruple matrices will be loaded in the Matlab workspace by executing the initialization m-file "run.m" and the vehicle subsystem is embedded in Simulink/ Vehicle subsystem as a continuous state-space system, shown in Figure 2.1.



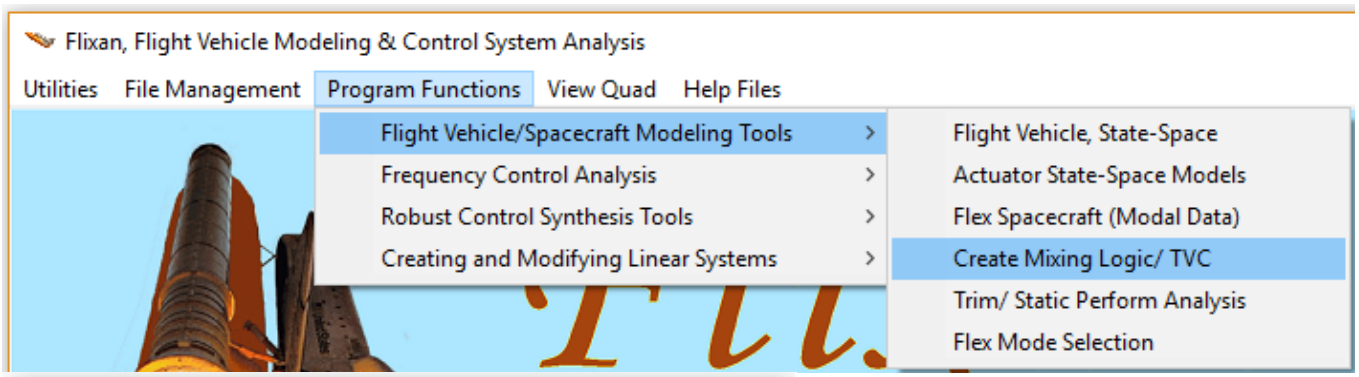
5 Mixing Logic Design (Combination of TVC with Engine Throttling)

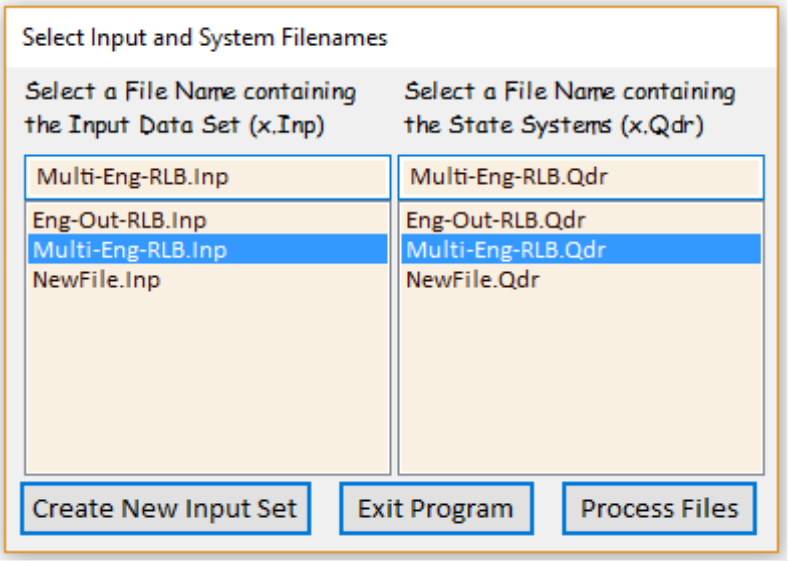
In this example we want to emphasize the design of a steering algorithm that efficiently combines the available effectors to control a launch vehicle by using both: TVC and differential throttling. It can also reconfigure itself in the event of an engine failure without a modification in the flight control law and without a significant impact on vehicle performance and stability. The mixing logic matrix converts the flight control system roll, pitch, and yaw acceleration demands to gimbal deflections and thrust variations in order to achieve the required accelerations for flight control. From the controls analysis point of view, the Mixing-Logic matrix post-multiplies the vehicle dynamic model and attempts to diagonalize the open-loop system. In fact, if you ignore the aerodynamic forces, it transforms the system (acceleration/ acceleration command) to an identity matrix. The advantage of generating this Vehicle/ Mixing Logic open-loop combination system is that it is robust to engine failures, assuming, of course, that the vehicle has sufficient engines, or in other words, sufficient control degrees-of-freedom, to span all directions without saturating the controls, and also, that the failure is detected early enough to reconfigure the mixing logic before the instability diverges.

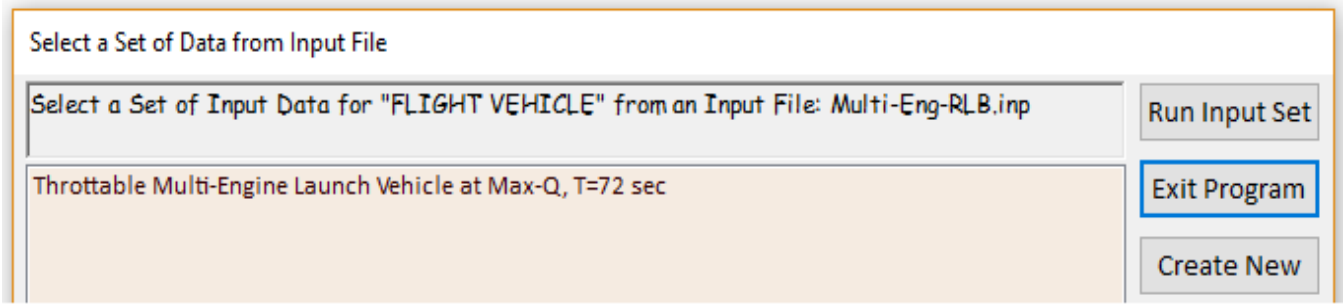
The mixing logic algorithm in this example uses a combination of gimbal deflections and thrust variations. The nominal vehicle has nine engines. All engines can gimbal in pitch and yaw but their maximum gimbal angles are different. The three Shuttle main engines have larger gimbal deflection angles (10^{o}) than the six booster engines, but they have lower thrust 470,000 (lb) and do not have any differential throttling capability. The 6 liquid booster engines can gimbal 5^{o} in pitch but only 1^{o} in yaw. They have a nominal thrust of 961,000 (lb) and their thrusts can vary ±45% to provide additional steering capability. All these parameters are taken into consideration by the mixing-logic algorithm which optimizes the mixing-logic matrix at a fixed vehicle configuration.

The mixing-logic matrix is generated by the Mixing-Logic program, which is included in Flixan program. The program reads the flight vehicle and engine parameters which are defined in the flight vehicle input data in file "*Multi-Eng-RLB.Inp*", the maximum gimbal deflections for each engine, the engines thrust and thrust variations, and calculates the mixing logic matrix that when inserted in front of the vehicle it transforms the plant almost to an identity matrix at the particular flight condition. This vehicle has 24 control inputs: 9 pitch gimbal deflections, 9 yaw gimbal deflections, and 6 throttle control inputs for the booster engines. It has additional inputs for the implementation of tail-wag-dog dynamics, and also an input for the wind-gust disturbance, but only 24 of the vehicle inputs are used for steering and only those inputs are connected to the mixing logic outputs via the 24 actuators. In the vehicle model the throttle control inputs are defined to be the ratios of thrust variation divided by the nominal thrust for each throttling engine. In the simulations, the throttle inputs should not exceed ±0.45 because the maximum thrust is limited to 1.4 million (lb). The size of the mixing logic matrix is (24x3). It has 24 outputs which are connected to the 24 vehicle control inputs via the 24 actuators. The 3 inputs to the mixing logic matrix are: roll, pitch, and yaw vehicle acceleration demands which are outputs from the flight control system.

To execute the mixing logic program for the nominal vehicle configuration you must first select the project directory "*Examples\Multi-Engine First-Stage Liquid Booster*". From the Flixan main menu select "*Program Functions*", then "*Flight Vehicle/ Spacecraft Modeling*", and then "*Create Mixing Logic/ TVC*". From the filenames selection menu select the input and system files: "*Multi-Eng-RLB.Inp*" and "*Multi-Eng-RLB.Qdr*" respectively and click on "*Process Files*". From the "*Flight Vehicles*" input data selection menu chose the only vehicle set "*Throttling Multi-Engine Launch Vehicle at Max-Q, T=72 sec*" and click "*Run Input*".

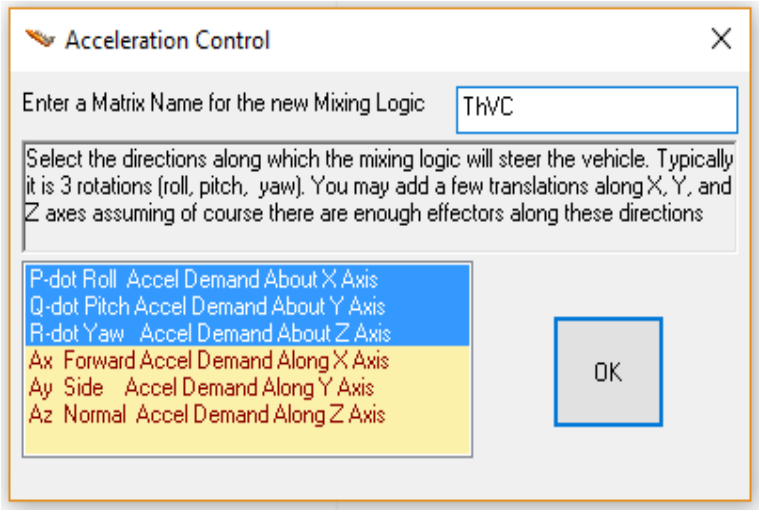






The following dialog defines the vehicle acceleration degrees-of-freedom that will be controlled by the mixing-logic matrix. Select the default three steering directions, roll, pitch, and yaw rotational accelerations. You must also enter the name of the mixing logic matrix "*ThVC*", and click "OK".

Then answer "Yes" to save the mixing-logic matrix ThVC in file "*Multi-Eng-RLB.Qdr*".



6 Modeling the Engine Thrust Failure

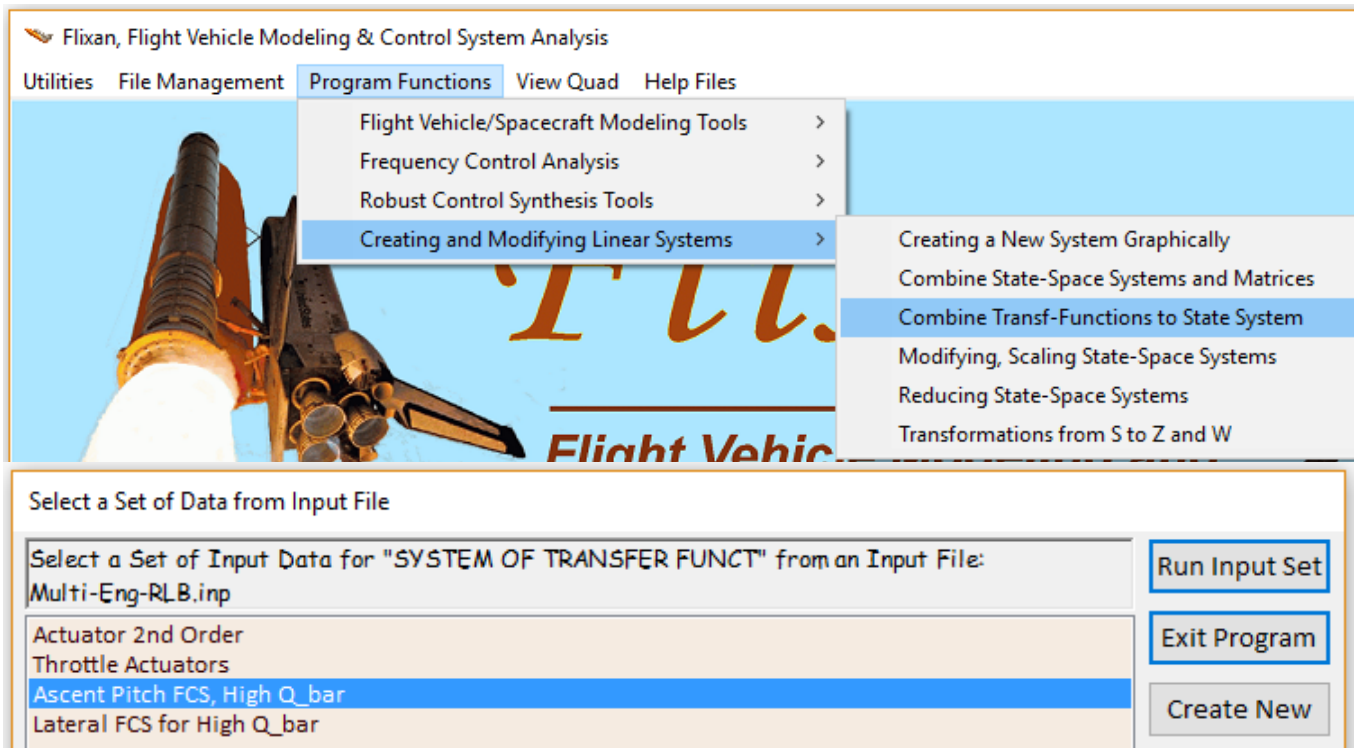
The same modeling procedure is repeated for the engine-out vehicle setup which has the thrust in one of the engines set to zero. The modified vehicle input data is in file "*Eng-Out RLB.Inp*", and its title is "*Throttable Multi-Engine Launch Vehicle at Max-Q, T=72 sec, Engine-9 Out*". The only difference between the two configurations is in the engine data. This vehicle has the last engine #9 set to zero, which is one of the right booster engines, to simulate thrust failure. This model will be used to demonstrate the capability of the adjustable mixing logic to reconfigure the mixing logic in the event of an engine failure. For implementation reasons (i.e. to maintain the same output dimension in both systems) the engine #9 was not omitted from the failed engine model but its thrust was set to a very small value. The nominal thrusts of the liquid booster engines were also adjusted in order to maintain thrust balance between the left and right sides. The nominal thrust of the remaining two engines on the failed (right) side was increased to 1.2 million (lb) with a maximum thrust still 1.4 million (lb). The nominal thrust of the three engines on the left side was reduced to 0.8 million (lb) with a maximum thrust still 1.4 million (lb). The flight vehicle modeling program was also used to generate the engine-out vehicle state-space model which is saved in a separate systems file "*Eng-Out RLB.Qdr*" under the title: "*Throttable Multi-Engine Launch Vehicle at Max-Q, T=72 sec, Engine-9 Out*". The name of the engine-out vehicle system is "*vehicle-eo.m*" and it was also exported to the Matlab subdirectories to be used in linear analysis. The modified dataset produces a different vehicle model that requires a significantly different mixing logic matrix in comparison with the nominal vehicle mixing logic.

Repeat the same mixing logic generation process for the engine-out vehicle configuration. There are two files in the project folder for the failed engine vehicle which are very similar to the nominal engine files: A vehicle input data file "*Eng-Out RLB.Inp*" and a system file "*Eng-Out RLB.Qdr*". The vehicle data in the failed engine case are modified to represent an engine #9 thrust failure on the right side of the liquid booster. The thrust on engine #9 was set to almost zero. It was not set completely to zero for implementation reasons in order to avoid taking the failed engine completely out of the model and not having to change the dimension of the new mixing matrix. Notice, that in the failed engine case the nominal thrusts of the remaining booster engines were also changed to preserve the thrust balance between left and right. The mixing logic matrix of the vehicle with the failed engine is saved in file "*Eng-Out RLB.Qdr*" below the vehicle state-space model. The name of the new matrix is "*ThVC_eo*".

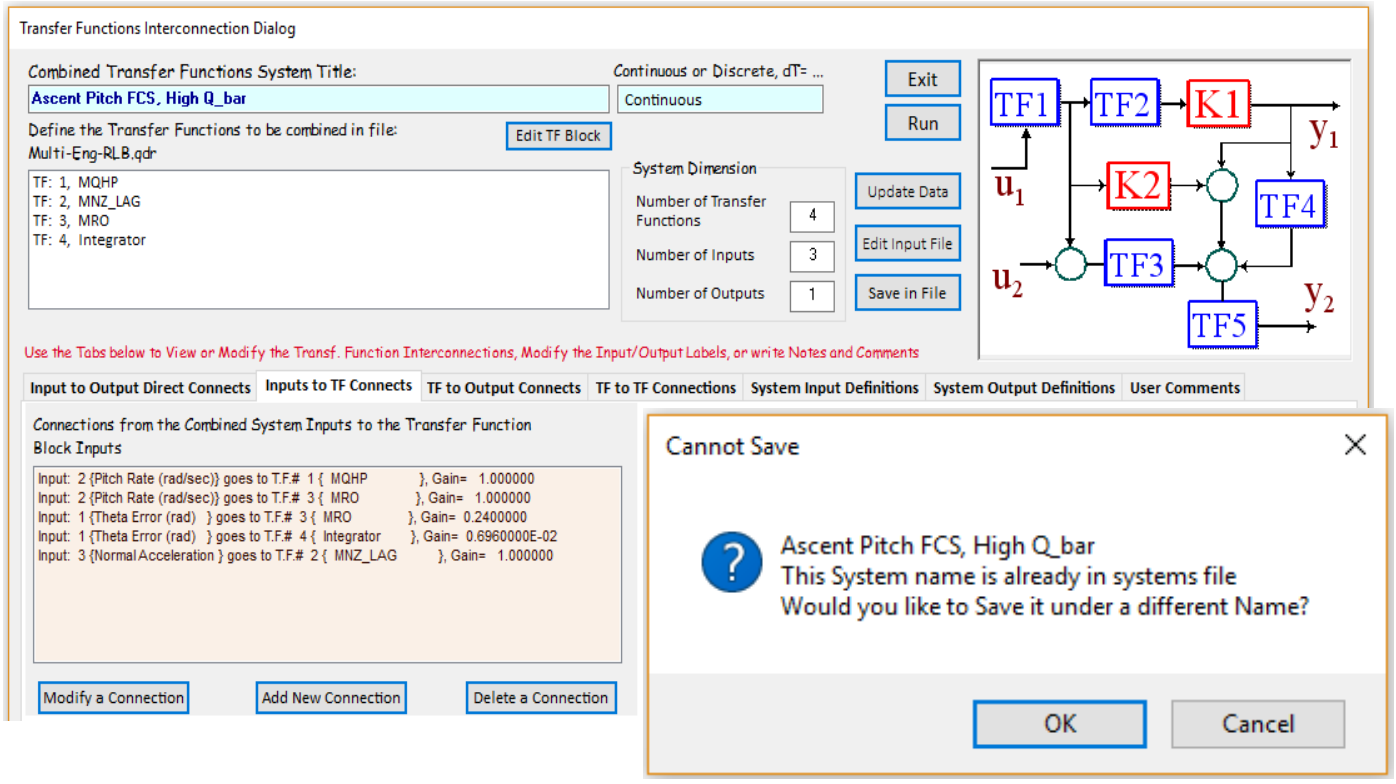
7 Actuator Models (Gimbaling and Throttling)

In this analysis we are using 24 actuators, one for each control input: nine pitch TVC gimbal actuators, nine yaw TVC actuators, and six throttle control actuators. The gimbal actuators are second order systems of 3.5 Hz bandwidth, see Figure 8.3. The inputs are deflection commands in (rad) coming from the mixing-logic. Each actuator has two outputs that drive the corresponding vehicle inputs: gimbal deflection in (rad), and gimbal acceleration in (rad/sec^2). The gimbal accelerations are used for the implementation of the tail-wag-dog dynamics.

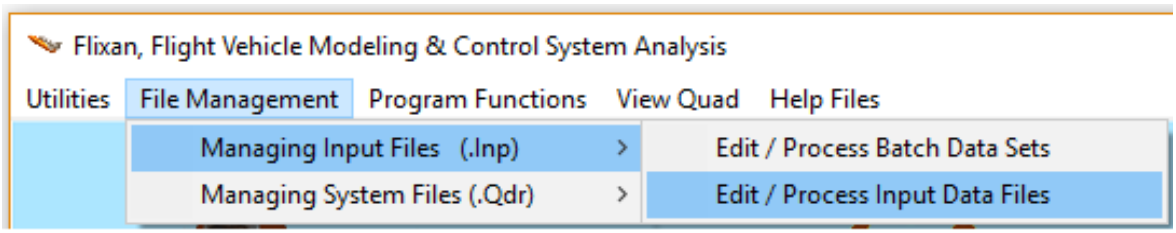
The TVC actuator models are created from transfer-functions data using the Flixan transfer-functions utility program. The transfer function data for a single gimbaling engine is in file "*Multi-Eng-RLB.Inp*" and its title is "*Actuator 2nd Order*". The Flixan transfer functions interconnection utility is used to create the second order actuator state-space system which has one input and two outputs and it is saved in systems file "*Multi-Eng-RLB.Qdr*" under the same title "*Actuator 2nd Order*". Nine of those second order gimbal actuators state-space systems are combined together using the Flixan systems combination program to create a group of 9 actuators that: with their first outputs drive the 9 pitch



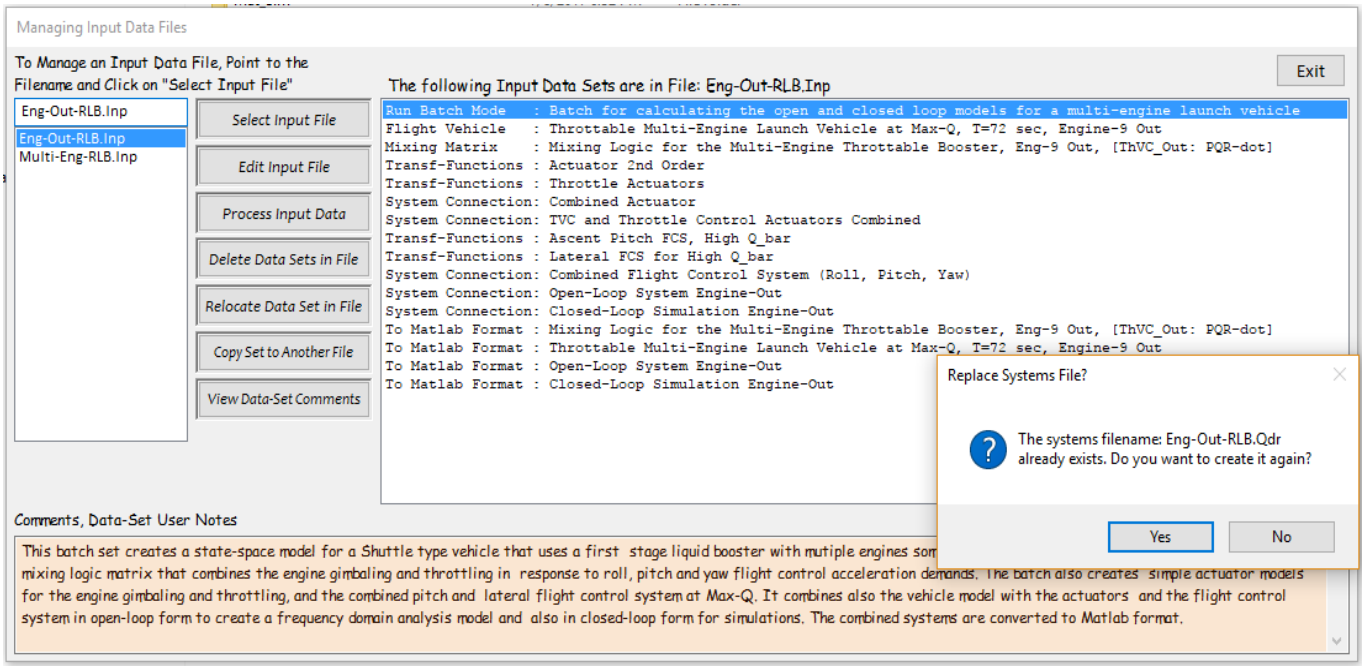
The Transfer Functions Interconnection dialog comes up that shows the four TFs and their interconnections in tabs. Click on "Run" to process the dataset, and then click "*OK*" in the next dialog to save the pitch FCS in the systems file. Repeat the same process to generate the lateral FCS.



The systems and matrices are also converted to Matlab (.m) and (.mat) file formats to be used in Matlab/ Simulink analysis. To run the batch dataset, start the Flixan program and select the project directory "*Multi-Engine First-Stage Liquid Booster*". From the Flixan menu select: "*File Management*", "*Manage Input Files*", and then "*Edit/ Process Input Data*".



The input file management utility dialog appears and from the input file selection menu on the left, select the input data file "*Multi-Eng-RLB.Inp*" and click "*Select Input File*" button. The menu on the right shows the titles of all datasets included in that file and the corresponding programs that will process the datasets. Select the top option which is the batch set and click "*Execute/ View Input Data*". In the next dialog click "Yes" to indicate that it is ok to replace the already existing systems in file "*Multi-Eng-RLB.Qdr*".



The batch set executes and will process the remaining datasets in batch mode as shown below. It saves the files in the root directory "*Multi-Engine First-Stage Liquid Booster*". From there you may transfer the files in the subdirectories for further analysis using Matlab. There is a similar batch set in the engine-out file "*Eng-Out-RLB.inp*" that generates similar files for the engine-out vehicle configuration.

10 Frequency Domain Analysis

Stability analysis is performed in folder "*Multi-Engine First-Stage Liquid Booster\ Mat_Frequ*" using the Open-Loop Simulink model "*Open_Loop.Mdl*", shown in Figure 10.1 below. This model consists of three blocks: (a) the actuators with the mixing logic matrix ThVC, (b) the flight vehicle state-space model, which is loaded from file vehicle.m, and contains the pitch and lateral vehicle dynamics, and (c) the combined pitch and lateral flight control state-system, loaded from file fcs.m. For frequency response analysis the loop is opened at the FCS/ TVC interface, one axis at a time, with the other two axes closed. For example, when evaluating roll axis stability, the roll loop is opened at the FCS output (DP_tvc), and the pitch and yaw loops (DQ_tvc and DR_tvc) are closed, as shown in Figure 10.1. To analyze stability in pitch and yaw, this Simulink model must be modified accordingly by opening one loop and closing the other two.

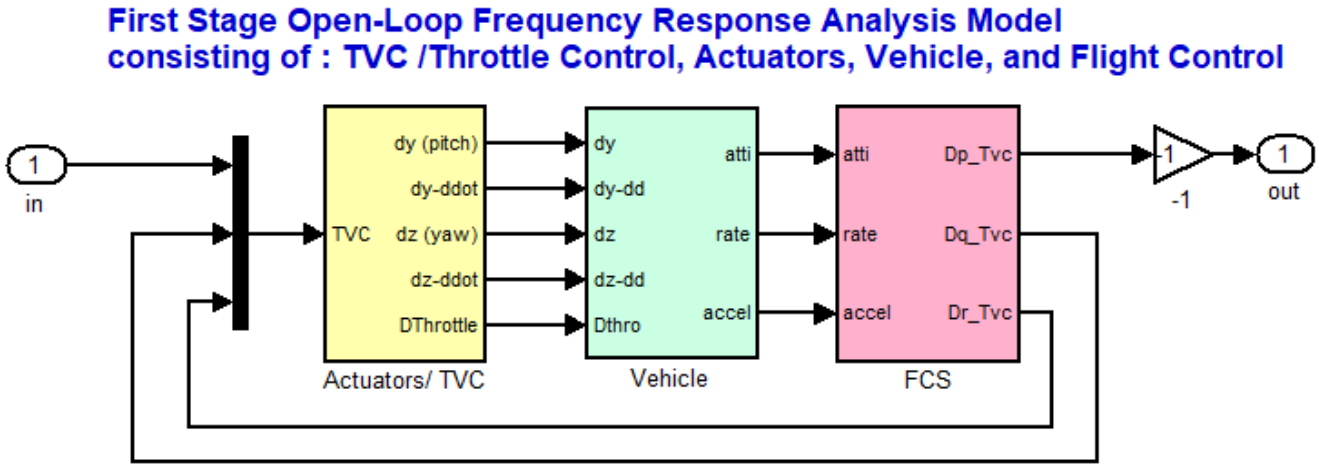


Figure 10.1 Stability Analysis Model "*Open_Loop.Mdl*" for Frequency Response Analysis

Figure 10.2 shows the actuator subsystem that contains 24 actuators models for the engine gimbaling and throttle controls, 9 pitch gimbals, 9 yaw gimbals, and 6 throttle control actuators, as already described. The block includes the mixing logic matrix ThVC that converts the (roll, pitch, and yaw) flight control demands to gimbal and throttle actuator commands. The actuator gimbal outputs control the pitch and yaw engine gimbal deflections that drive the flight vehicle model. The gimbal accelerations are also needed in the vehicle model for the implementation of the tail-wag-dog dynamics. The last 6 outputs in the actuator subsystem are lower bandwidth throttle control actuators for the liquid booster engines. They are used for regulating the engine thrusts in the vehicle model and the implementation of the differential throttling feature as commanded by the mixing-logic.

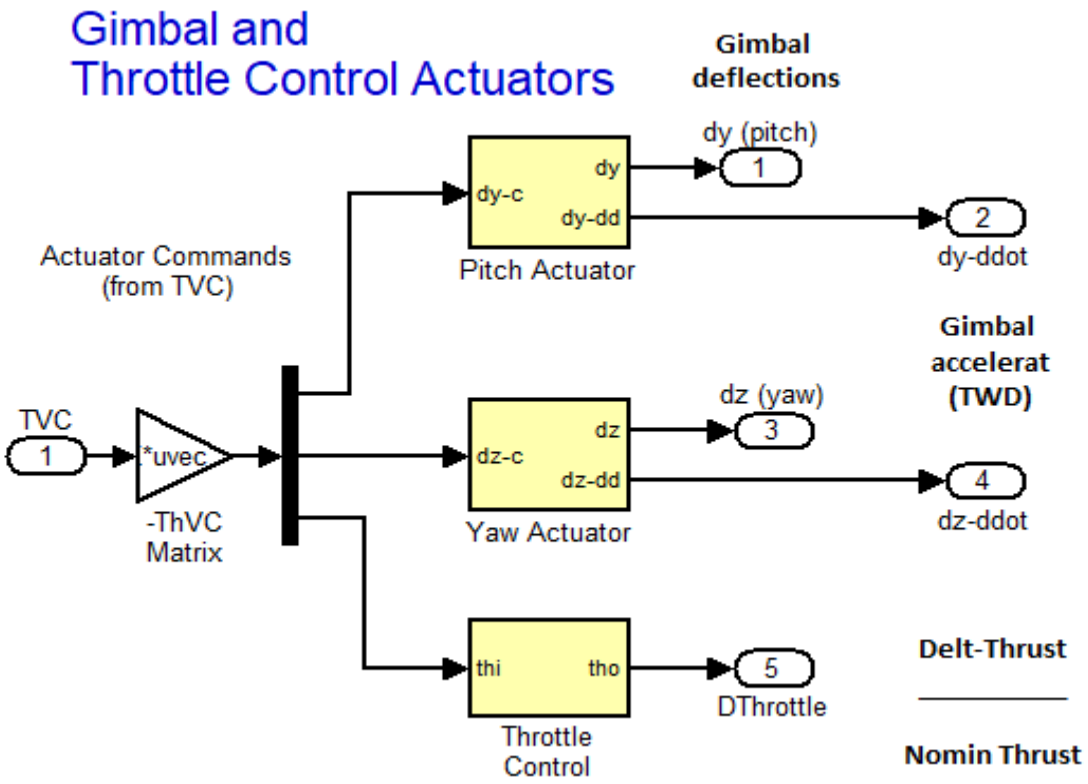


Figure 10.2 Actuator Subsystem for the Engine Gimbaling and Throttling including the Mixing Logic Matrix

A Simulink model similar to Figure 10.1 was constructed for the failed engine vehicle model. Actually the same "*Open_Loop.Mdl*" Simulink file was used, but the nominal vehicle was replaced with the engine-out vehicle "*vehicle_eo.m*", and the engine-out mixing logic matrix was replaced with "ThVC-eo.Mat". The reason for analyzing the system stability using both vehicle configurations and mixing logic matrices is to: (a) make sure that stability margins are sufficient in all 3 axes, and (b) demonstrate that the mixing-logic adjustment is capable of handling the engine thrust failure without any substantial impact on system stability. The performance analysis is confirmed by time-domain simulations.

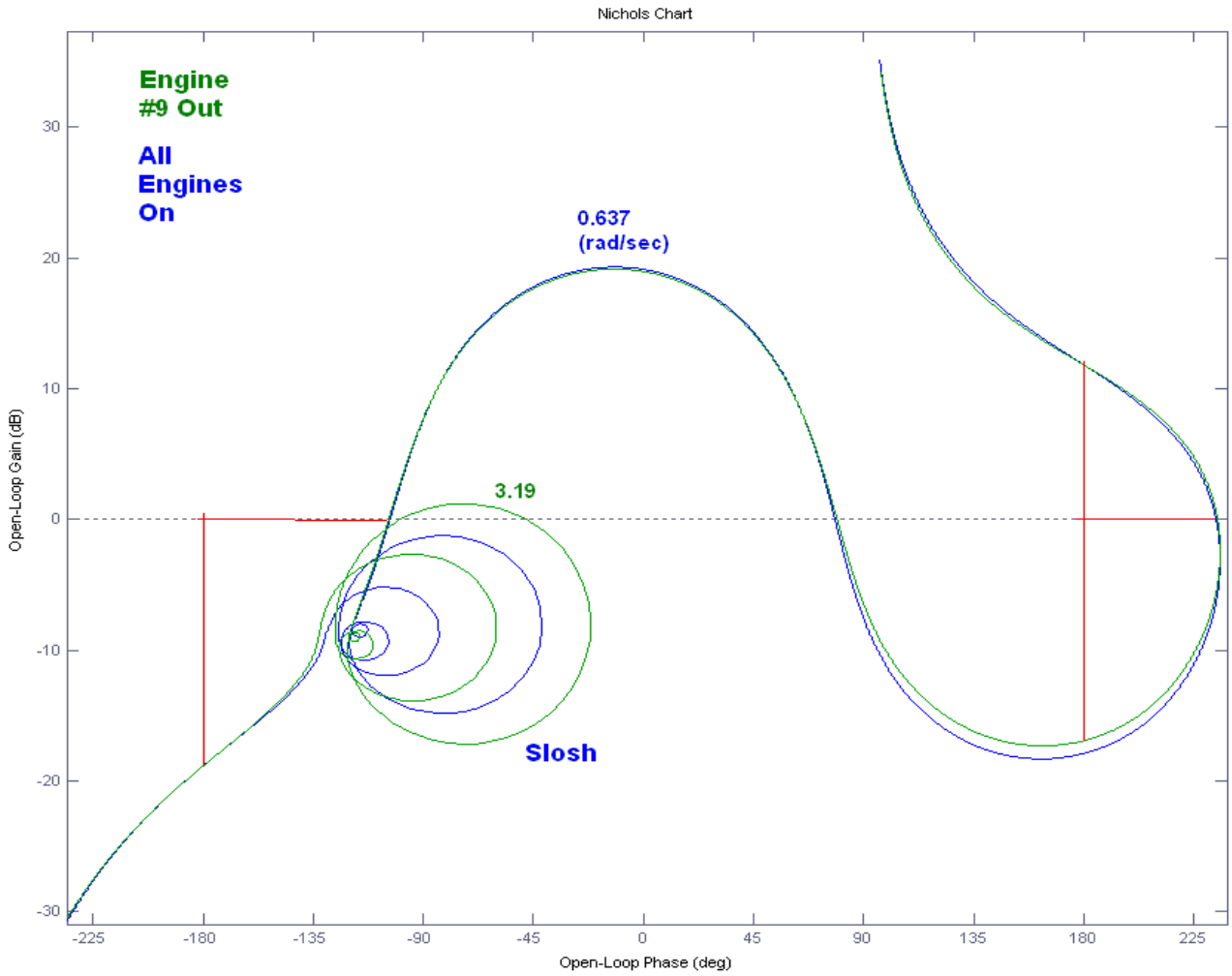
The stability analysis is performed by executing the Matlab script file "*run.m*". This file loads the two vehicle systems, the actuator, and the flight control systems into Matlab. It also loads the mixing logic matrices plus additional open-loop and closed-loop systems created using Flixan. It uses the Simulink models "*Open-Loop.Mdl*" or "*Open-Loop2.Mdl*" to calculate the frequency response and plot the Nichols. The following Nichols plots demonstrate that the stability margins are sufficient in all three axes. In addition, the differences between the 9-engine (green) and 8-engine (blue) loci are minimal in all three axes which show that the engine failure has no impact on vehicle stability after the TVC modification and, therefore, the engine failure is transparent to the flight control loop, assuming of course that the loss in thrust is detected early enough to reconfigure the mixing logic.

```
% File: run.m
% Multi-Engine Throttleable Booster System
r2d=180/pi; d2r=pi/180;
LBThrust=0.96097e+6;                    % Nominal F1 Engine Thrust
thc=8;                                  % Throttle Control Bandwidth
[Ave, Bve, Cve, Dve]= vehicle;          % Load the Vehicle Model from Flixan
%[Ave, Bve, Cve, Dve]= vehicle_eo;      % Engine Out Vehicle Model from Flixan
[Aac, Bac, Cac, Dac]= comb_actuator;    % Load the 9 combined actuators
[Afc, Bfc, Cfc, Dfc]= fcs;              % Load Pitch/Lateral Flight Control
[Aol, Bol, Col, Dol]= openloop;         % Combined Open-Loop Nominal (Actuat+Vehicle+FCS)
%[Aol, Bol, Col, Dol]= openloop_eo;     % Combined Open-Loop Eng-out (Actuat+Vehicle+FCS)
[Acl, Bcl, Ccl, Dcl]= closedloop;       % Closed-Loop Simulation Model
load ThVC.mat ThVC -ascii               % TVC plus Throttle Control Matrix (Nominal)
load ThVC_eo.mat ThVC_eo -ascii         % TVC plus Throttle Control Matrix (Eng-out)

[Ao,Bo,Co,Do]=linmod('Open_Loop');      % Combine State-Space system for ...
%[Ao,Bo,Co,Do]=linmod('Open_Loop2');     % Pre-Combined State-Space system for ...
sys= SS(Ao,Bo,Co,Do);                   % frequency response analysis

[Ac,Bc,Cc,Dc]=linmod('Closed_Loop');    % Pre-Combined State-Space system for ...
sysc=SS(Ac,Bc,Cc,Dc);                   % frequency response analysis
w=logspace(-4,2.5,25000);
figure(2); Nichols(sys,w)
%figure(1); Nichols(sys,w, syseo,w)
figure(3); Sigma(sysc,w)
```

Pitch Axis Open-Loop Stability



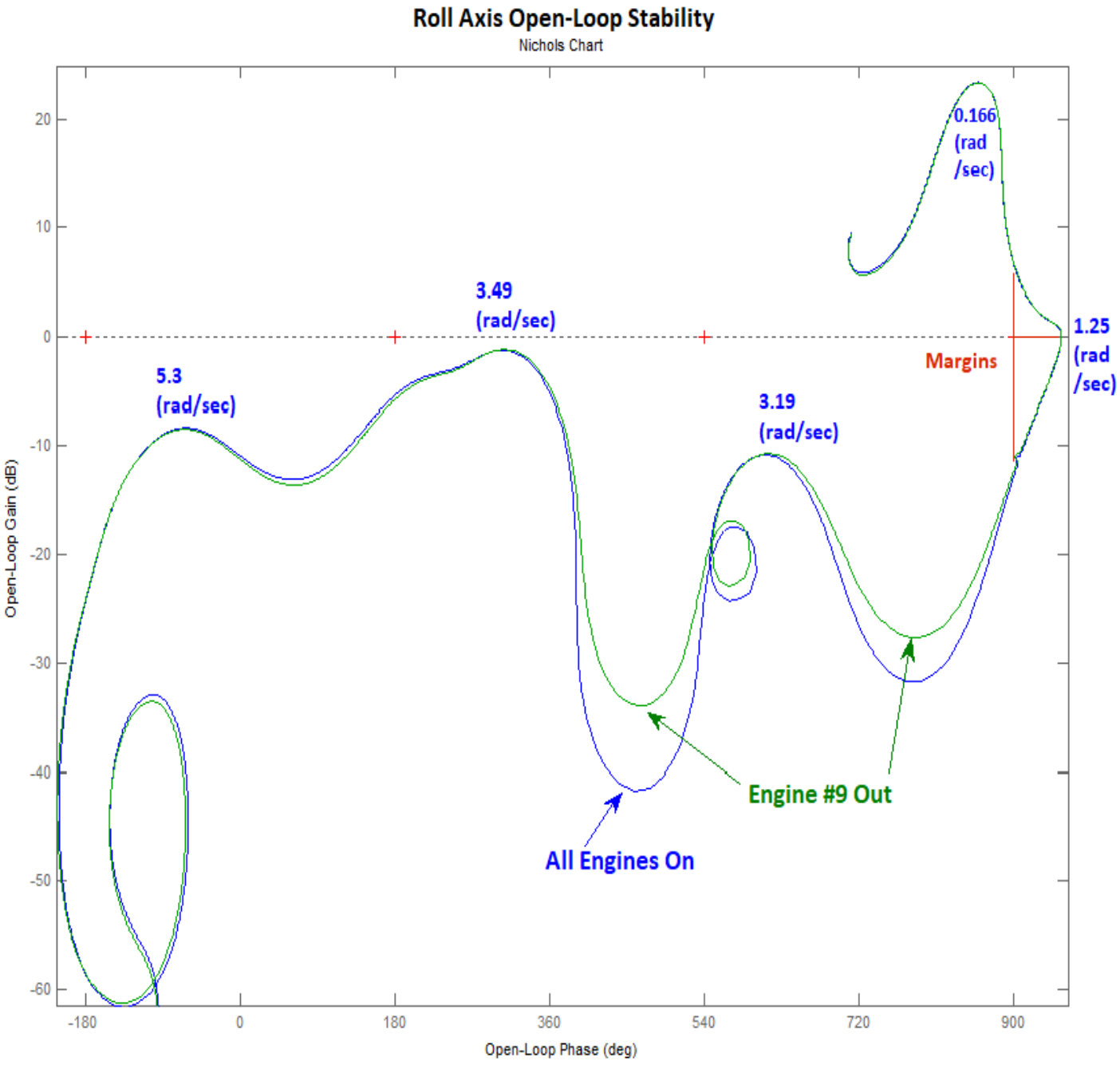


Figure 10.3 Stability Analysis, Nominal versus Engine-Out Models Comparison (Roll, Pitch and Yaw)

Figure 11.2 shows the closed-loop system sensitivity in the frequency domain between a wind-gust input and the angles of attack and sideslip together as a vector. The wind the direction is specified relative to the vehicle in the input data. It shows that the system sensitivity to gusts is weak at 0.67 (rad/sec), mainly in the lateral direction. This weakness in sensitivity is caused by the load-relief Ny-feedback and vertical stabilizer, and it is noticeable in the Shuttle vehicle in the High-Q region.

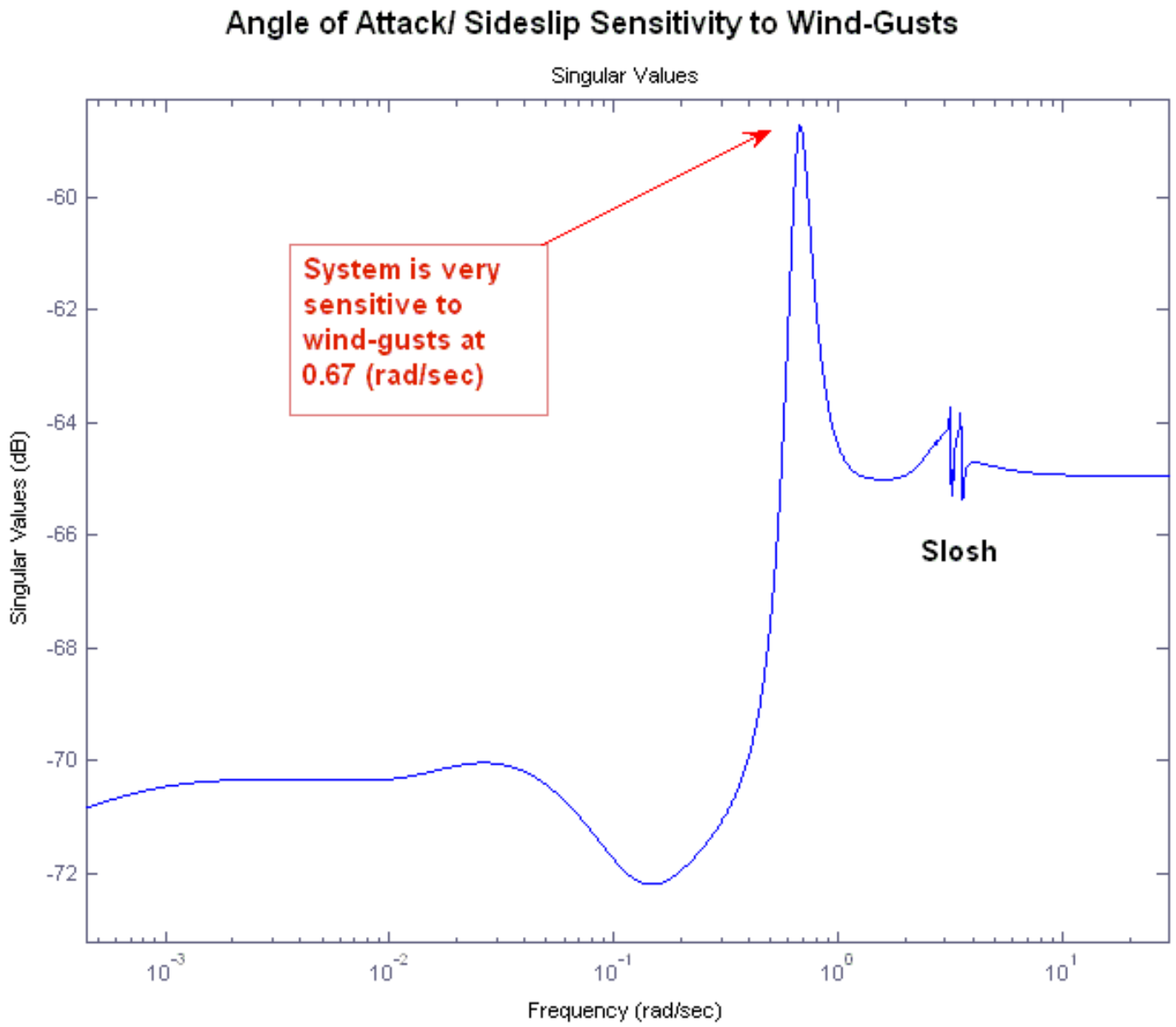


Figure 11.2 Closed-Loop System Sensitivity Observed in the Angles of Attack and Sideslip due to Wind-Gust Excitations. It uses System "*Closed_Loop.Mdl*" and Shows Weak sensitivity at 0.67 (rad/sec).

Figure 11.3 is a similar sensitivity frequency response plot of the closed-loop system attitude response per attitude command. It uses the same closed-loop system of Figure 11.1. The closed-loop system bandwidth is approximately 0.1 Hz. During high dynamic pressure the ascent system command following performance is somewhat compromised because of the load-relief feedback which has a higher priority. Load-relief feedback from the normal and lateral accelerometers is used to reduce aerodynamic loads due to wind-shear and gusts.

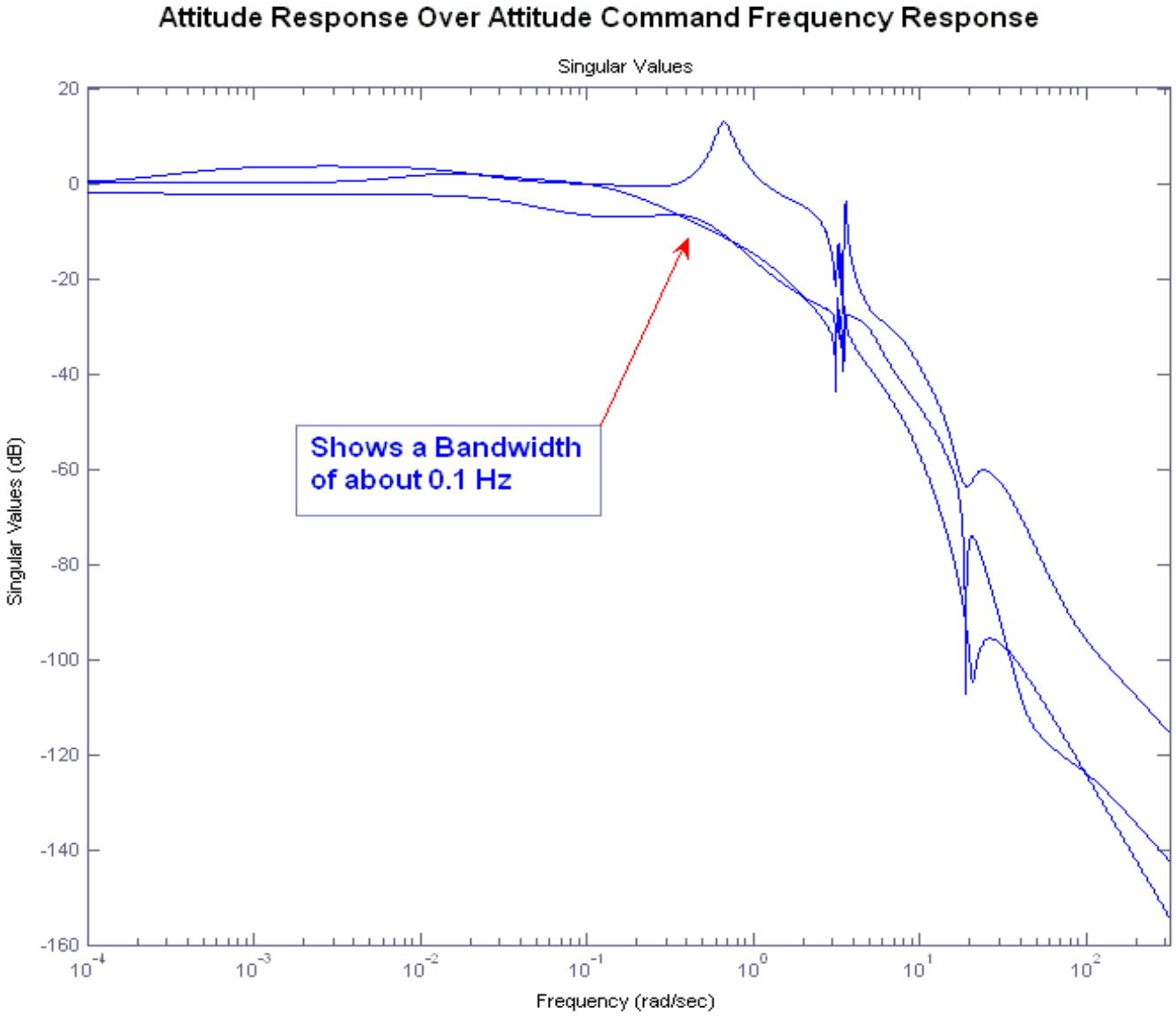


Figure 11.3 SV Frequency Response of the Attitude Command-Following System *"Closed_Loop.Mdl"*. Shows the Closed-Loop System Bandwidth

12 Simulation Models

There are two Simulink models "*CL_Sim_Nom.Mdl*" and "*CL_Sim_EOut.Mdl*", used for simulating the closed-loop system response to attitude commands and to wind gust disturbances. They are similar and are located in the Matlab subdirectory "*Flight\ Examples\ Multi-Engine First-Stage Liquid Booster\Mat_Sim*". The first model uses the nominal vehicle state-space system from file "*vehicle.m*" with all 9 engines fully operating and the mixing-logic matrix ThVC. The second Simulink model uses the state-space system from file "*vehicle_eo.m*" that operates only with 8 engines. The nominal model "*CL_Sim_Nom.Mdl*", is shown in figure 12.11 below. It consists of 3 subsystems which are combined together in Simulink, (a) the 9 pitch, 9 yaw gimbal and 6 throttle control actuators (yellow), (b) the flight vehicle model (green) which includes pitch and lateral dynamics, and (c) the FCS/TVC block (magenta) that includes the flight control state-space system and the mixing logic matrix ThVC.

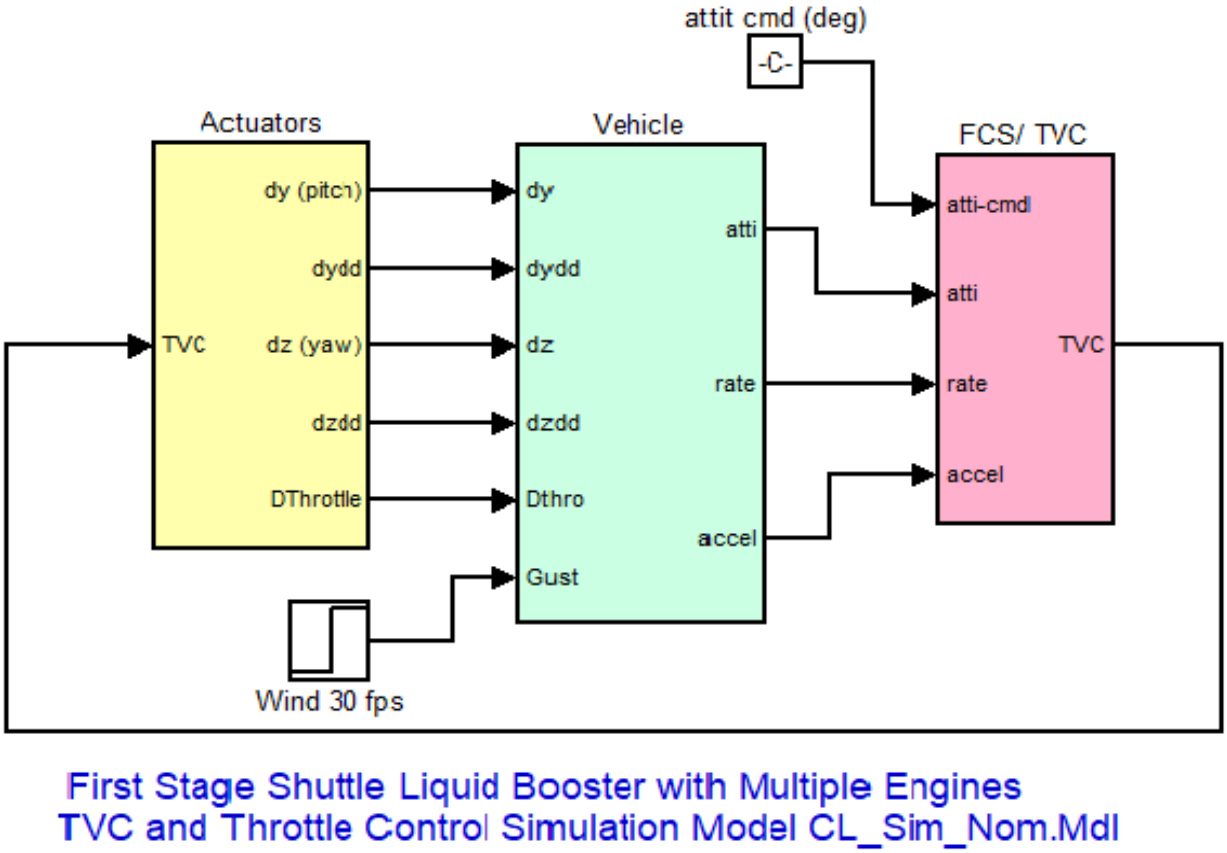


Figure 12.1 Simulation model "*CL_Sim_Nom.mdl*" for Shuttle with Multi-Engine Liquid Booster. Uses a Combination of TVC and Throttle Control

Figure 12.2 shows the actuators subsystem consisting of 3 sub-blocks, the 9 pitch gimbal actuators, the 9 yaw gimbal actuators, and the 6 throttle control actuators. The gimbal actuator is shown in detail in Figure 12.3. Its outputs are deflections in (rad) and gimbal accelerations for the TWD implementation in (rad/sec^2). The 6 throttle actuators shown in figure 12.4 control the thrust variation. Zero throttle command to an engine is equivalent to nominal engine thrust. δThrottle= +0.45 is equivalent to 45% thrust increase and δThrottle= -0.45 is equivalent to 45% thrust reduction command. The gimbal and throttle commands are generated by the mixing logic matrix.

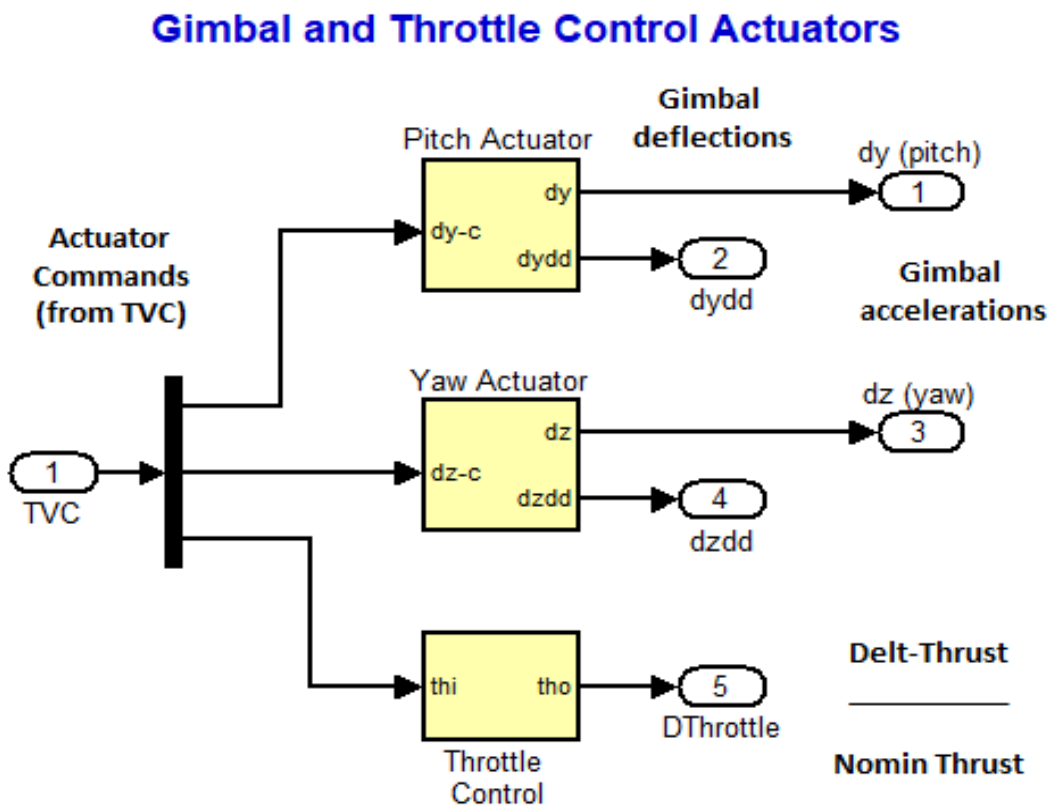


Figure 12.2, Gimbal and Throttle Control Actuators Subsystem

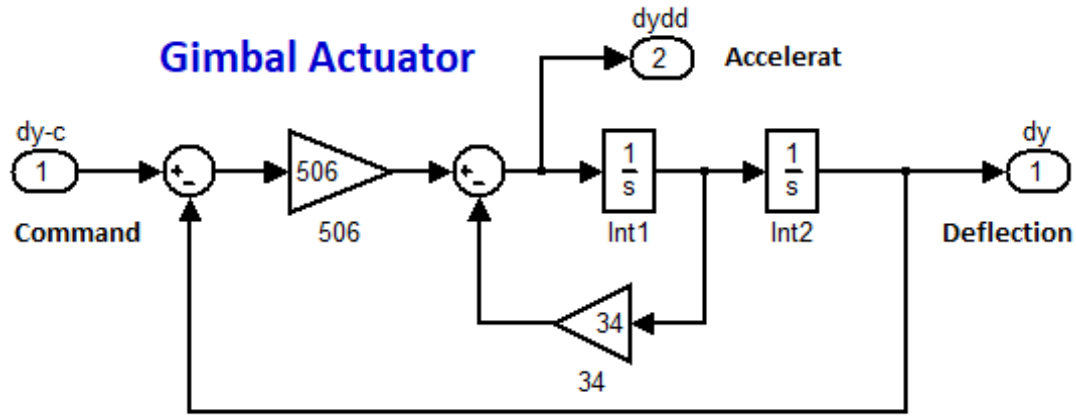


Figure 12.3 Second Order Gimbal Actuator Model

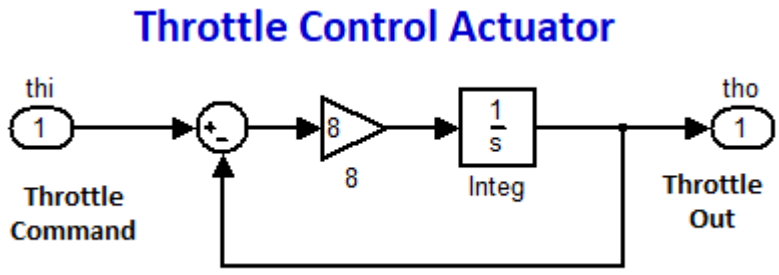
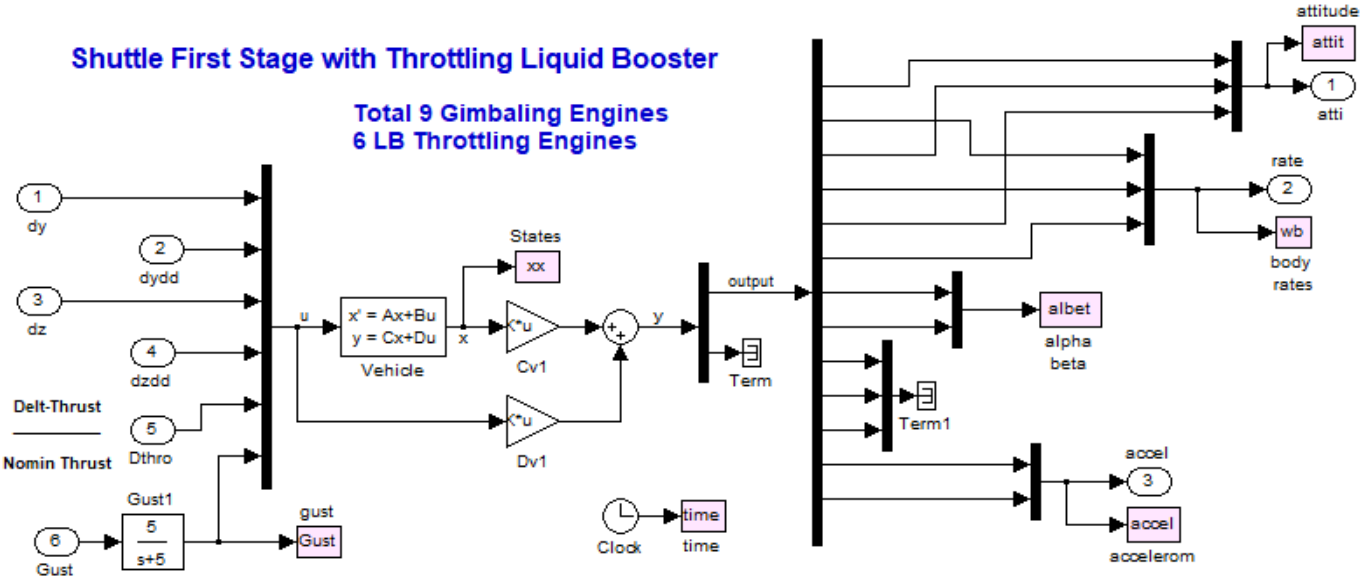


Figure 12.4 Throttle Control Actuator

The launch vehicle model is in the second block of Figure 12.1 and it is shown in detail in Figure 12.5. The title of the vehicle state-space system is "*Throttable Multi-Engine Launch Vehicle at Max-Q, T=72 sec*" which was saved in the m-file "*vehicle.m*".




```
% Inputs = 43
%   1   Engine No  1 Pitch Deflect. (rad), Dymax= 10.0000 deg
%   2   Engine No  2 Pitch Deflect. (rad), Dymax= 10.0000 deg
%   3   Engine No  3 Pitch Deflect. (rad), Dymax= 10.0000 deg
%   4   Engine No  4 Pitch Deflect. (rad), Dymax=  5.0000 deg
%   5   Engine No  5 Pitch Deflect. (rad), Dymax=  5.0000 deg
%   6   Engine No  6 Pitch Deflect. (rad), Dymax=  5.0000 deg
%   7   Engine No  7 Pitch Deflect. (rad), Dymax=  5.0000 deg
%   8   Engine No  8 Pitch Deflect. (rad), Dymax=  5.0000 deg
%   9   Engine No  9 Pitch Deflect. (rad), Dymax=  5.0000 deg
%  10   Engine No  1 Pitch Acceleration  (rad/sec^2)
%  11   Engine No  2 Pitch Acceleration  (rad/sec^2)
%  12   Engine No  3 Pitch Acceleration  (rad/sec^2)
%  13   Engine No  4 Pitch Acceleration  (rad/sec^2)
%  14   Engine No  5 Pitch Acceleration  (rad/sec^2)
%  15   Engine No  6 Pitch Acceleration  (rad/sec^2)
%  16   Engine No  7 Pitch Acceleration  (rad/sec^2)
%  17   Engine No  8 Pitch Acceleration  (rad/sec^2)
%  18   Engine No  9 Pitch Acceleration  (rad/sec^2)
%  19   Engine No  1 Yaw Deflect. (rad), Dzmax= 10.0000 deg
%  20   Engine No  2 Yaw Deflect. (rad), Dzmax= 10.0000 deg
%  21   Engine No  3 Yaw Deflect. (rad), Dzmax= 10.0000 deg
%  22   Engine No  4 Yaw Deflect. (rad), Dzmax=  1.0000 deg
%  23   Engine No  5 Yaw Deflect. (rad), Dzmax=  1.0000 deg
%  24   Engine No  6 Yaw Deflect. (rad), Dzmax=  1.0000 deg
%  25   Engine No  7 Yaw Deflect. (rad), Dzmax=  1.0000 deg
%  26   Engine No  8 Yaw Deflect. (rad), Dzmax=  1.0000 deg
%  27   Engine No  9 Yaw Deflect. (rad), Dzmax=  1.0000 deg
%  28   Engine No  1 Yaw Acceleration  (rad/sec^2)
%  29   Engine No  2 Yaw Acceleration  (rad/sec^2)
%  30   Engine No  3 Yaw Acceleration  (rad/sec^2)
%  31   Engine No  4 Yaw Acceleration  (rad/sec^2)
%  32   Engine No  5 Yaw Acceleration  (rad/sec^2)
%  33   Engine No  6 Yaw Acceleration  (rad/sec^2)
%  34   Engine No  7 Yaw Acceleration  (rad/sec^2)
%  35   Engine No  8 Yaw Acceleration  (rad/sec^2)
%  36   Engine No  9 Yaw Acceleration  (rad/sec^2)
%  37   Throttle Input dTh/Th for Engine No  4 (-)
%  38   Throttle Input dTh/Th for Engine No  5 (-)
%  39   Throttle Input dTh/Th for Engine No  6 (-)
%  40   Throttle Input dTh/Th for Engine No  7 (-)
%  41   Throttle Input dTh/Th for Engine No  8 (-)
%  42   Throttle Input dTh/Th for Engine No  9 (-)
%  43   Wind Gust Azim, Elev Angles=(  45.0  53.9) (deg)
```

```
% Outputs = 31
%   1   Roll Attitude  (phi-body) (radians)
%   2   Roll Rate      (p-body)   (rad/sec)
%   3   Pitch Attitude (thet-bdy) (radians)
%   4   Pitch Rate     (q-body)   (rad/sec)
%   5   Yaw  Attitude  (psi-body) (radians)
%   6   Yaw  Rate      (r-body)   (rad/sec)
%   7   Angle of attack, alfa, (radians)
%   8   Angle of sideslip, beta, (radian)
%   9   Change in Altitude, delta-h, (feet)
%  10   Forward Acceleration  (V-dot)  (ft/sec)
%  11   Cross Range Velocity (Vcr) (ft/sec)
%  12   Accelerom #  1, (along Z), (ft/sec^2) Translat. Acc
%  13   Accelerom #  2, (along Y), (ft/sec^2) Translat. Acc
%  14   Ptch Load-Torque Tly for Engine: 1,(ft-lb)
%  15   Ptch Load-Torque Tly for Engine: 2,(ft-lb)
%  16   Ptch Load-Torque Tly for Engine: 3,(ft-lb)
%  17   Ptch Load-Torque Tly for Engine: 4,(ft-lb)
%  18   Ptch Load-Torque Tly for Engine: 5,(ft-lb)
%  19   Ptch Load-Torque Tly for Engine: 6,(ft-lb)
%  20   Ptch Load-Torque Tly for Engine: 7,(ft-lb)
%  21   Ptch Load-Torque Tly for Engine: 8,(ft-lb)
%  22   Ptch Load-Torque Tly for Engine: 9,(ft-lb)
%  23   Yaw  Load-Torque Tlz for Engine: 1,(ft-lb)
%  24   Yaw  Load-Torque Tlz for Engine: 2,(ft-lb)
%  25   Yaw  Load-Torque Tlz for Engine: 3,(ft-lb)
%  26   Yaw  Load-Torque Tlz for Engine: 4,(ft-lb)
%  27   Yaw  Load-Torque Tlz for Engine: 5,(ft-lb)
%  28   Yaw  Load-Torque Tlz for Engine: 6,(ft-lb)
%  29   Yaw  Load-Torque Tlz for Engine: 7,(ft-lb)
%  30   Yaw  Load-Torque Tlz for Engine: 8,(ft-lb)
%  31   Yaw  Load-Torque Tlz for Engine: 9,(ft-lb)
```

Figure 12.5 Launch Vehicle Dynamics Subsystem Using State-Space System File "*Vehicle.m*"

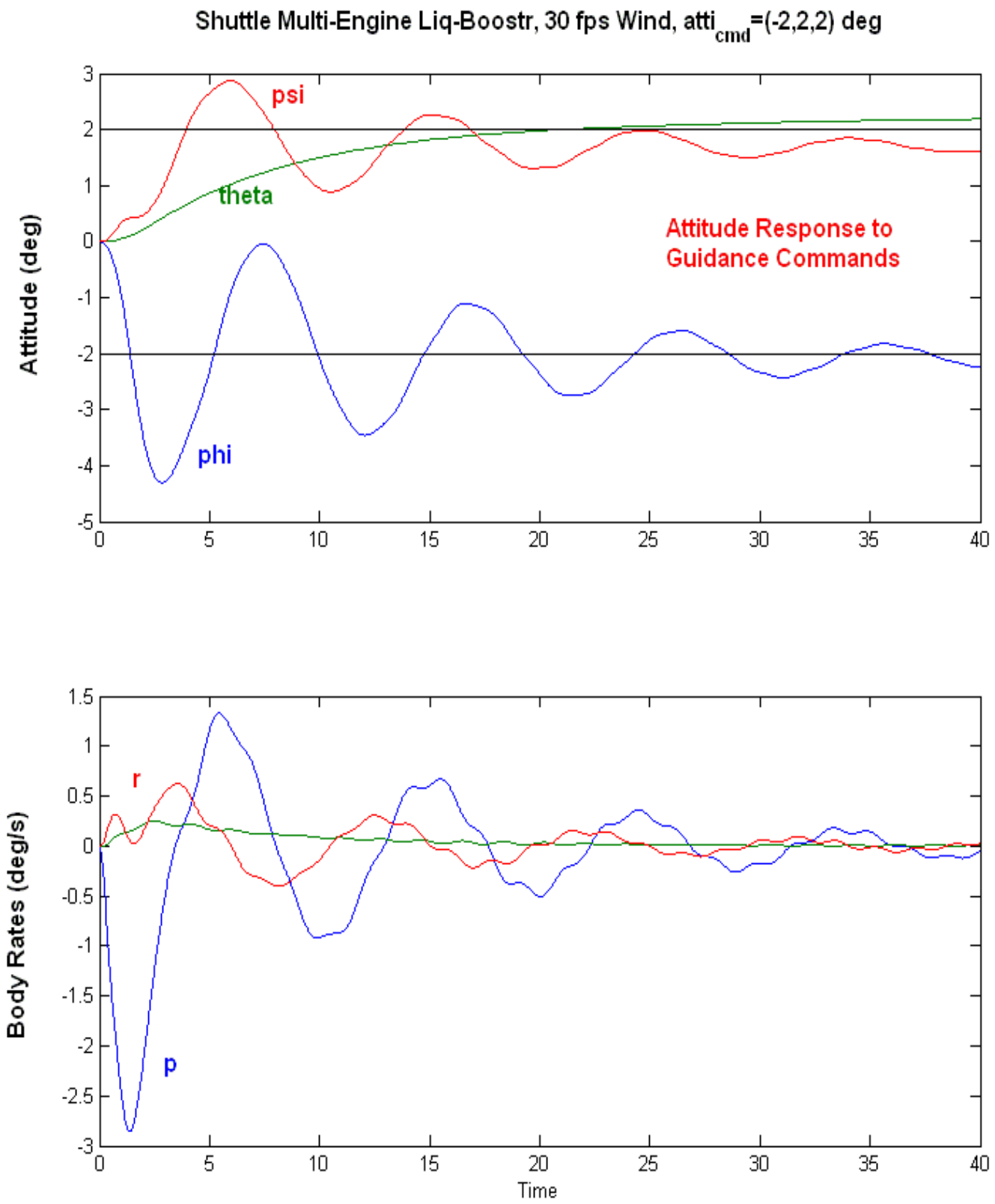


Figure 13.1a Attitude and Rate Responses to Simultaneously Applied Attitude Commands and Wind Shear Disturbance

13.2 Engine-Out Simulation

The engine-out model "*CL_Sim_EOut.Mdl*" in subdirectory "*Examples\Multi-Engine First-Stage Liquid Booster\Mat_Sim*" is very similar to Figure 12.1, but a little more complex because it is initialized from the nominal model of Figure 12.1. We run the "*CL_Sim_Nom.Mdl*" model that includes all 9 engines for 8 seconds and save the terminal conditions (inputs, states, outputs). The terminal conditions are then used to initialize the engine-out simulation model "*CL_Sim_EOut.Mdl*" which uses the vehicle model with the failed engine "*Throttable Multi-Engine Launch Vehicle at Max-Q, T=72 sec, Engine-9 Out*", which was exported to m-file "*vehicle_eo.m*" for Matlab analysis.

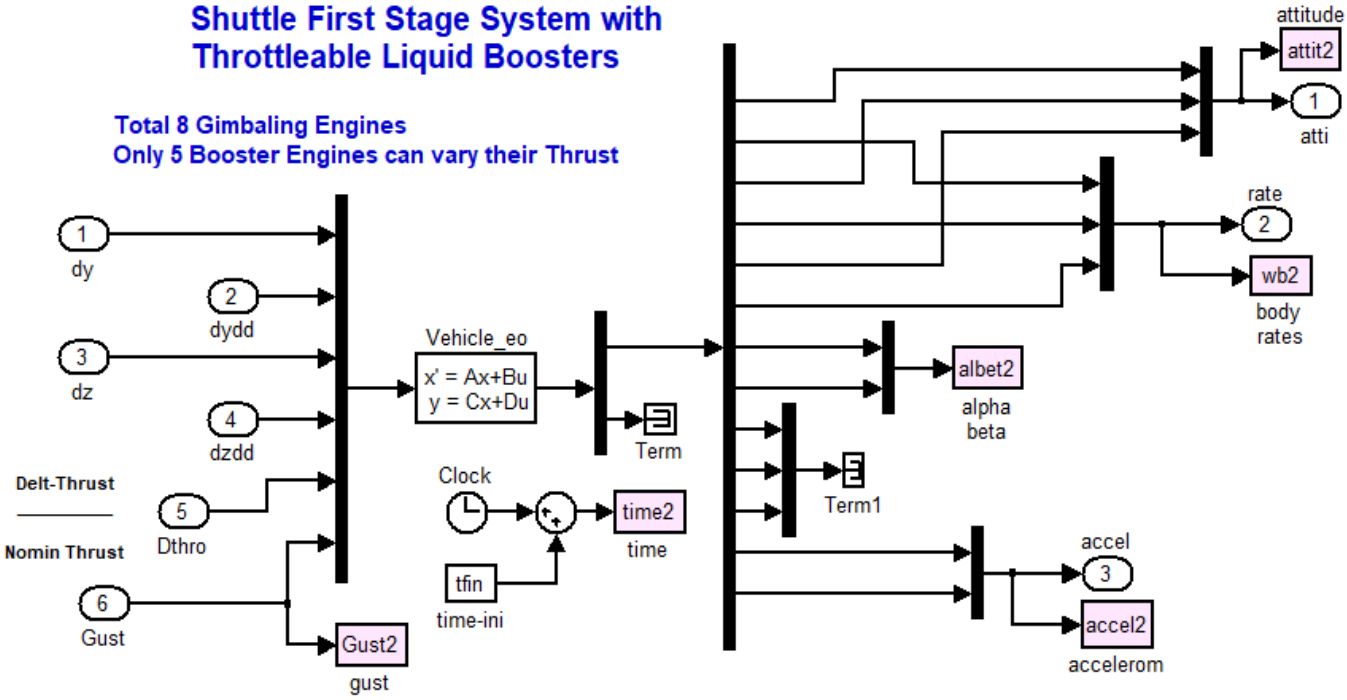


Figure 13.2 Vehicle Subsystem in "*CL_Sim_EOut.Mdl*" uses state-space system from "*vehicle_eo.m*"

In this simulation we start with the nominal vehicle simulation like before, and we fail engine #9 eight seconds later, while the vehicle is performing an attitude change maneuver and in the presence of a steady wind disturbance. Eight seconds later we simulate the engine failure by switching vehicle models and replacing it with the failed engine system. The nominal mixing logic matrix "*ThVC.Mat*" is also replaced at time= 8 sec with the engine-out mixing logic matrix "*ThVC_eo.Mat*" that was calculated earlier using the mixing logic program. Both mixing-logic matrices are the same size but the second one was calculated under the assumption that the engine-9 thrust is reduced to zero.

We are actually running the two simulation models sequentially. They both receive the same inputs, they use identical actuators and flight controls, but they have different vehicle dynamics and mixing-logic matrices. "*CL_Sim_Nom.Mdl*" performs the nominal vehicle simulation from zero to 8 seconds, when the failure occurs, and it saves the terminal states, inputs, and outputs of the vehicle system. "*CL_Sim_EOut.Mdl*" initializes its state-space variables from the previous model, performs the engine-out simulation from 8 to 40 seconds using the engine-out mixing logic. Figure 13.3 shows the combined systems response to the excitations. The 30 (feet/sec) wind-shear step was smoothed by a low-pass filter.

The Shuttle vehicle response is intentionally sluggish and oscillatory in the roll and yaw axes during max dynamic pressure condition because of the load-relief feedback. The attitude response to commands is also slow. At Max-Q the control emphasis is not command following, but rather to reduce the body rates and the (α and β) transients due to the wind-shear. One might argue that the lateral oscillation should also be further dampened. This fix was demonstrated in the Shuttle Ascent example by applying H-infinity control.

At the 8 seconds transition point between the nominal and the failed-engine vehicles, the 8 engine vehicle state-vector is initialized from terminal state-vector of the 9 engine vehicle to achieve a smooth transition in the vehicle response. The main noticeable difference is in the controls, that is, the pitch and yaw gimbal deflections and the throttle commands in the actuators. This significant change after 8 seconds is not caused by the flight control system output since it does not change between models, but it is caused by the switching of the mixing logic matrix. Notice, that the mixing-logic of the failed engine system causes bigger gimbal deflections and throttling of the remaining working engines after switching. They are compensating for the failed engine #9 which is no longer capable to produce any torques on the vehicle. Notice, also that the tail-wag-dog modeling was taken out in this simulation in order to simplify the engine gimbal initialization after the transition by using simple first order actuator models instead of the more complex second order models used previously.

To run the double model simulation you must first execute the script file "run.m" to load the two vehicles state-space systems, their corresponding mixing logic matrices, and the flight control system into Matlab. It also initializes the engine-out vehicle model in "*CL_Sim_EOut*.Mdl". Then, open the 9-engine Simulink model "*CL_Sim_Nom*.Mdl" and run it for 8 seconds to calculate the simulation data prior to the engine failure. Check the vehicle state-vector and gimbal angles at the end of the run and use these values to initialize and execute "run.m". The initialization values in "*CL_Sim_EOut*.Mdl" should be equal to the terminal values of "*CL_Sim_Nom*.Mdl". Now open and run "*CL_Sim_EOut*.Mdl" for 32 seconds to generate the simulation data beginning at 8 sec, after the engine failure. Finally, run the plotting script file "pl2.m", which combines the data from the two simulation models and plots them in Matlab as shown in Figure 13.3 below.

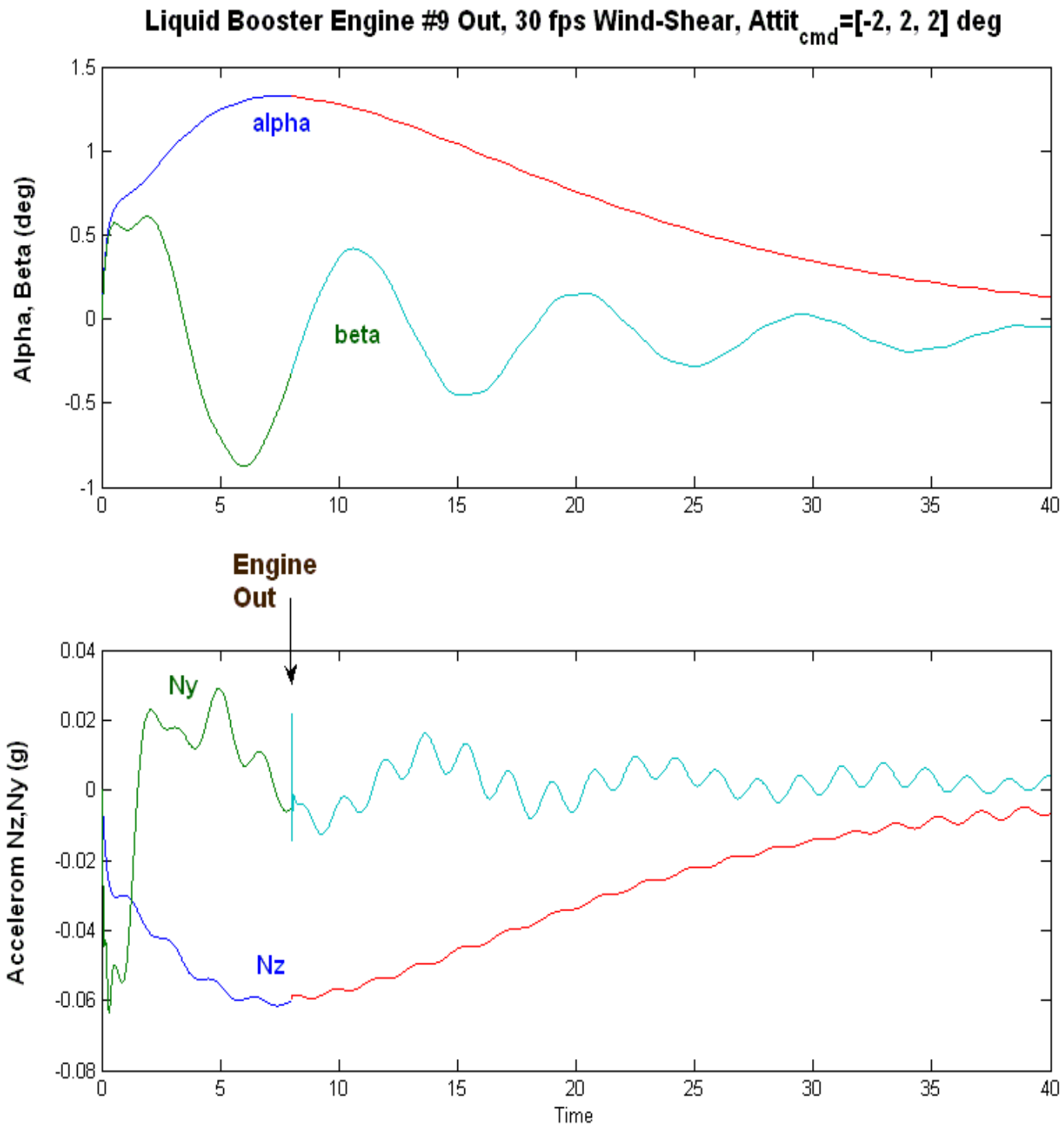


Figure 13.3a Angle of Attack/ Sideslip and Accelerometer Responses. Fuel sloshing is visible in the accelerometers, but it is stable.

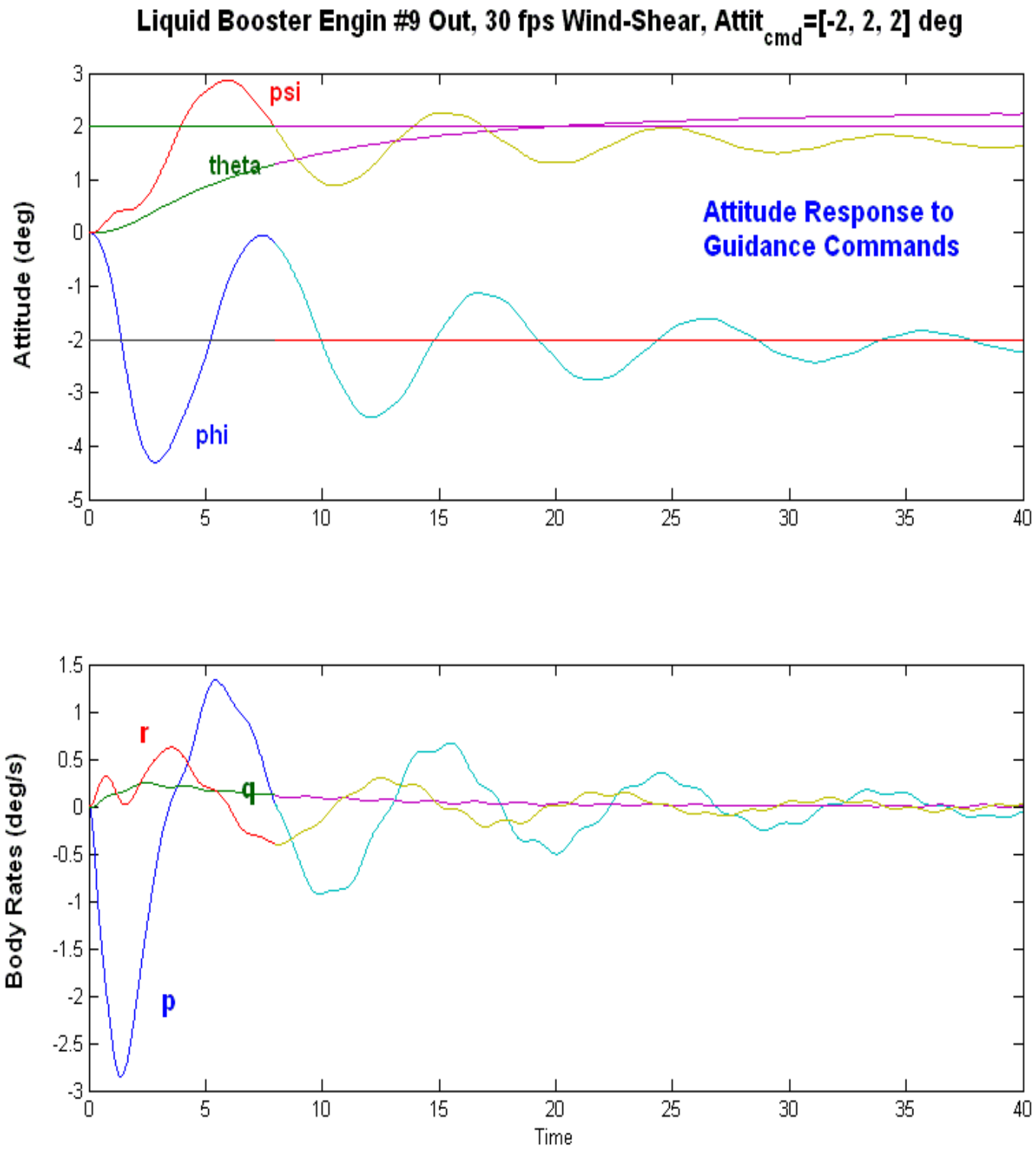


Figure 13.3b Attitude and Rate Response to a simultaneous Attitude Command and Wind Shear