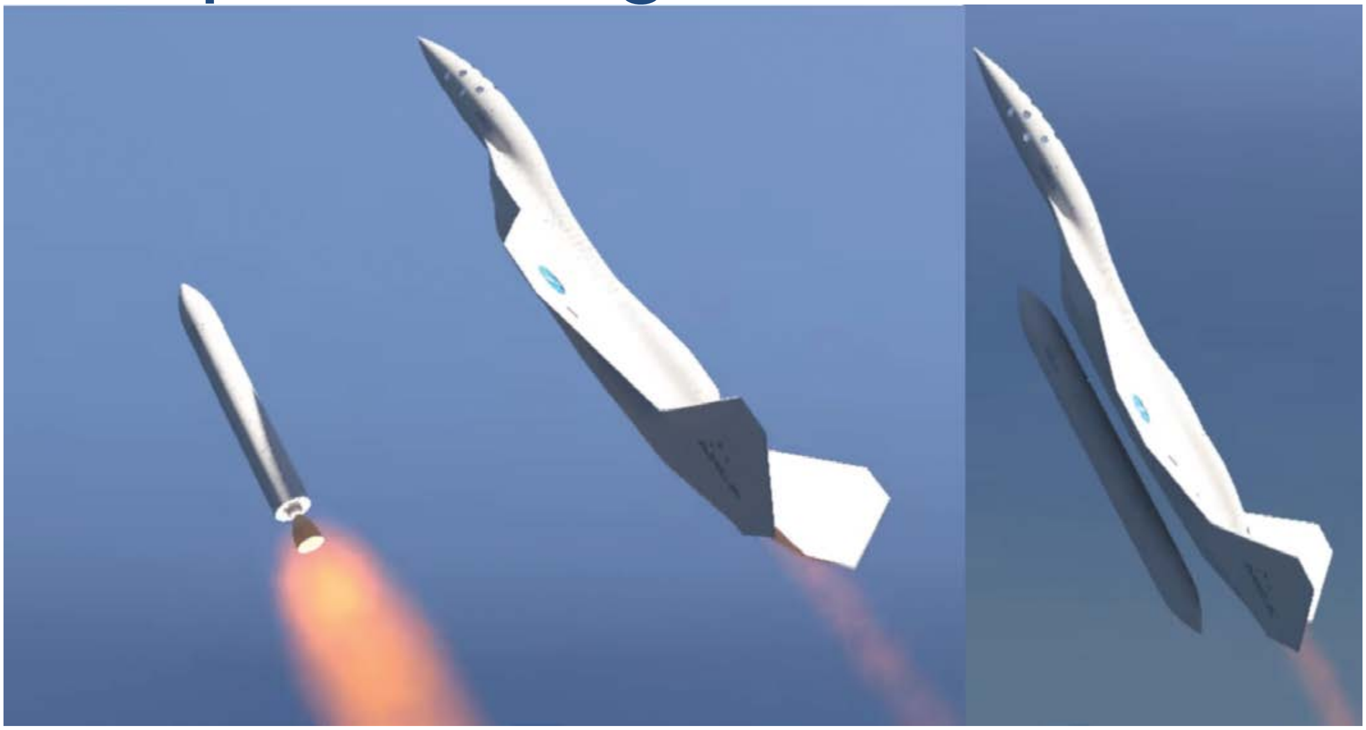
Air Launched Booster to Orbit Complete Design and Simulation



The purpose of this example is to familiarize the student with the entire methodology of designing, analyzing, and simulating the flight control system for a special type of launch vehicle. This vehicle is a little more complex than a typical launch vehicle that takes off from the ground because it is released from an aircraft that makes the take-off a little more dynamic, with adverse transients to be considered. Some of the advantages of an air launched booster are: reduced weight and cost because the first stage is a reusable aircraft, the engine size and thrust is reduced, plus the structural loading on the launch vehicle due to aerodynamics is reduced because the air density and dynamic pressure is lower at high altitudes.

The carrier aircraft is also special because it is capable of climbing up to 43,000 feet and releasing the booster in an upward direction, that is, with a positive flight-path angle γ, as shown in the picture above. This maneuver requires an aircraft with a rocket that ignites at about 30,000 feet and helps the aircraft climb to higher altitude in order to release the vehicle in an upward direction, at low speed, just before stalling. Then the aircraft rolls on the side to avoid impact, and the launch vehicle engines ignite 5 seconds after separation for safety in case of an explosion. A high altitude initial condition in an upward direction and at low dynamic pressure, are the key ingredients for optimizing the trajectory that will maximize the payload weight to orbit capability.

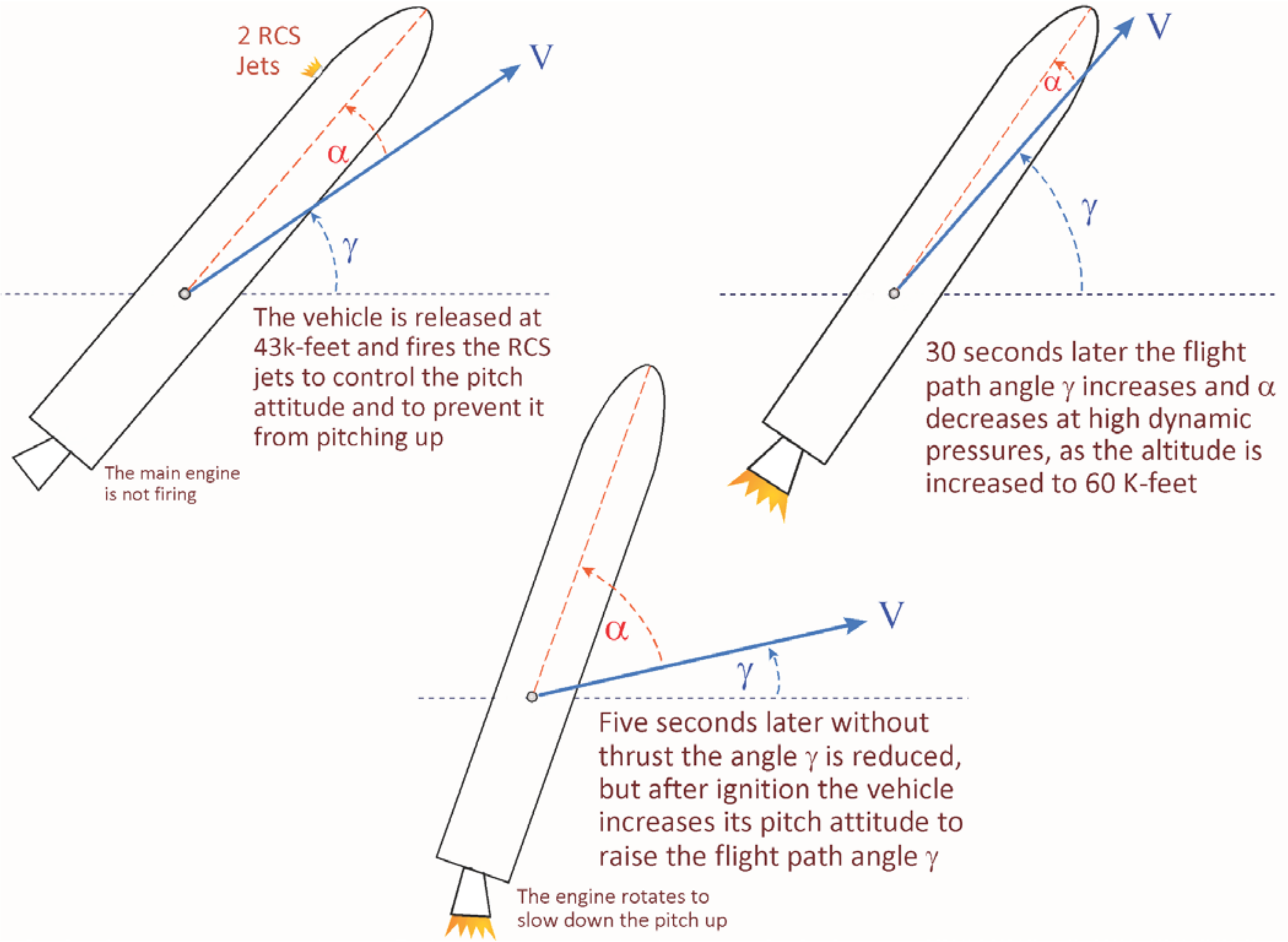
The launch vehicle is similar to the vehicle shown in this picture. It consists of three stages and delivers a spacecraft into an earth orbit. The two liquid boosters on the left and right sides are firing during first stage and they have two TVC engines that gimbal in pitch and yaw. They also throttle to control the acceleration. The middle booster has only one TVC engine that fires during second stage after the first stage separation. The third or upper stage places the spacecraft into circular orbit. It has a TVC engine that ignites after separation of the second stage. The vehicle also has two 800 (lb) reaction control thrusters for roll control during second and third stages, since one TVC engine is not capable for controlling roll. The two thrusters are located near the front of the vehicle and they are capable of firing in the ±Z body axis. They are also used immediately after separation from the aircraft to control pitch attitude during the 5 seconds before TVC ignition. This is because the launch vehicle is aerodynamically unstable and it would, otherwise, pitch up and impact the aircraft if left uncontrollable at high angles of attack.

We will perform a complete analysis of this vehicle beginning with: static analysis, preliminary control design, create a 6-DOF simulation from release to orbit insertion, detailed linear analysis, and control design. The study consists of the following tasks:

- Generate a preliminary trajectory that is initialized from the aircraft release conditions and places the spacecraft to circular orbit around the earth. This trajectory is generated by a point-mass trajectory optimization program and maximizes the payload weight to fuel ratio.
- Obtain aerodynamic data during first and second stages, mass properties in different flight conditions, and engine data such as thrusts, locations, etc. Propellant sloshing parameters, such as, slosh masses, frequencies, damping, and also structural data will be needed later in the detailed control analysis.
- Use the trajectory and other data, to perform static analysis, and to analyze the vehicle capability to trim along the optimized trajectory. Check the engine trim angles to determine the engine mounting positions. Analyze the vehicle static stability to make sure that it is neither too stable nor unstable. Check the controllability to make sure that there is sufficient control authority to maneuver and to overcome aero disturbances, CG shifts, etc.
- Design a preliminary flight control system for the rigid vehicle using Matlab. Use Flixan to generate vehicle models at different flight conditions along the first, second, and third stages of the trajectory. Then use Matlab and the Flixan rigid models to design state-feedback control gains to control the vehicle and to track the guidance commands. Create a table of gains versus flight time in an Excel file and export it into the 6-DOF simulation.
- Use Matlab/ Simulink to create a 6-DOF simulation for the entire trajectory, from release to orbit insertion. The simulation includes off-the-shelf models from the Aerospace blockset for the environment, gravity, atmosphere, aerodynamics, etc. It also includes look-up tables of thrusts, mass-properties, and the tables of the control system gains versus time. The guidance in the simulation is open-loop. The control system is tracking the optimal flight path angle versus time generated from the trajectory program.
- The simulation results are used to create more efficient and advanced vehicle models that also include propellant sloshing and structural flexibility. This vehicle includes liquid propellant tanks, and sloshing due to vehicle motion generates disturbances that interact with the rigid body motion that may cause instabilities. Structural flexibility also interacts with the control system, limits the control system bandwidth and may cause oscillatory instabilities. The Flixan models are upgraded to include sloshing and flexibility and vehicle systems are created for detailed control analysis.
- The detailed vehicle models are then used to update the control system designs in different flight conditions. Estimators and filters are included and the gains are adjusted in order to accommodate flexibility and sloshing. Control system analysis and simulations are performed at fixed time-slices, such as, MaxQ to analyze the stability margins and verify a satisfactory response to commands and wind gusts.

1. Static Analysis

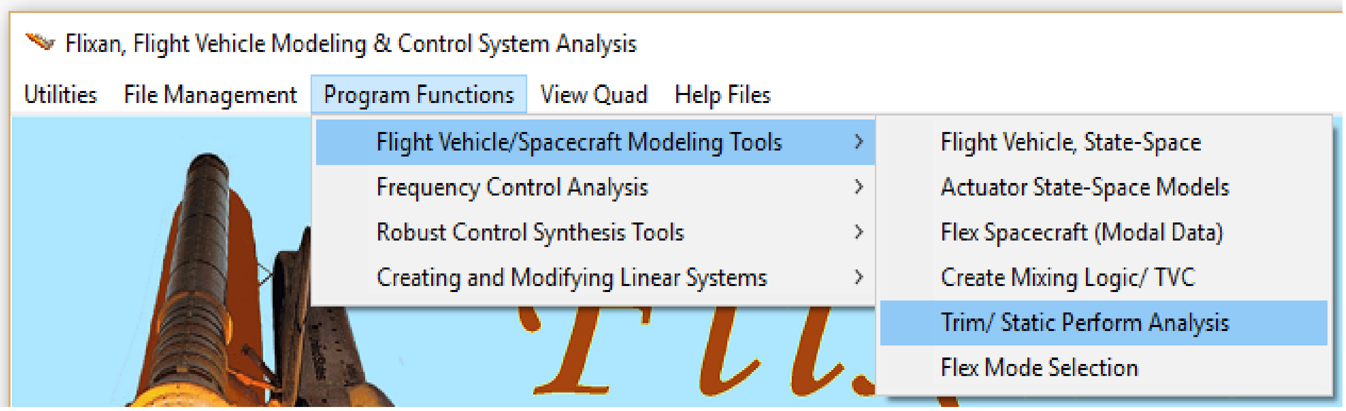
Before we begin the control design, analysis, and simulations we must first analyze the vehicle static performance and stability. It will be meaningless to design control laws and simulations if the vehicle does not meet the basic controllability and performance criteria. We will look at parameters such as static stability, controllability, disturbances, etc. For this type of analysis we will need some preliminary data, such as, vehicle mass properties, engine data, aerodynamic coefficients, and an initial trajectory. The static analysis is separated in first and second stages.

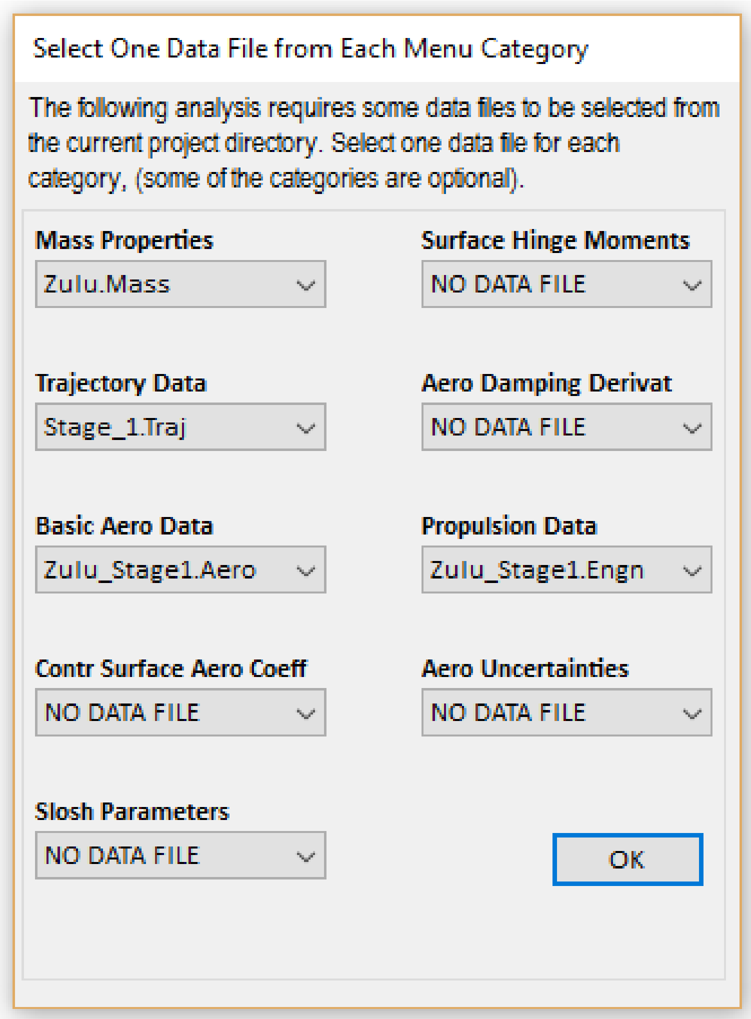


1.1 First Stage Static Analysis

The first stage begins 5 seconds after separation from the aircraft when the two booster engines ignite. The vehicle is released from the aircraft at an altitude of 43,000 feet with a positive $\gamma=45^o$. The two engines ignite 5 seconds later for safety, and since the vehicle is unstable in pitch, it fires two reaction control thrusters to control the pitch attitude during the period when the TVC is inactive, to prevent it from pitching up and striking against the aircraft. In the mean-time during the free-drift, gamma is reduced to 20^o. As soon as the engine ignites the vehicle pitches up to an attitude $\theta=86^o$ and increases the angle of attack to 60° in order to raise γ. The change in γ does not happen instantly but it takes a few seconds. This attitude maneuver is possible because the dynamic pressure is still low, at 40 psf. Eventually 30 seconds after release and at an altitude of 58,000 feet, gamma increases to 55° and alpha is reduced to 2° because the dynamic pressure is high. This maneuver achieves the necessary initial flight path angle γ that maximizes the payload weight to orbit. A point-mass trajectory optimization program was used to calculate the preliminary trajectory that places the satellite to a circular orbit. This trajectory is separated in 3 segments for first, second and third stages and it will be used in the static analysis, and to command the 6-DOF simulation.

The files for the first stage analysis are located in folder "*Examples\Twin Booster Rocket\1-Static Analysis\Stage-1*". The mass properties are in file "*Zulu.Mass*" and they are calculated as a function of vehicle mass which decreases as the fuel is depleted. The basic aerodynamic coefficients are in file "*Zulu_Stage1.Aero*", which contains aero coefficients at 16 Mach numbers, 3 betas, and 24 values of the angle of attack. The file also includes the reference lengths and the location of the moment reference center. The engines file "*Zulu_Stage1.Engn*" includes parameters for the two TVC engines, such as the average engine thrusts 380,000 (lb), gimbal locations, pitch and yaw mounting angles, 5° maximum deflections relative to the mounting positions, and a throttling parameter. Although in this vehicle we are not directly controlling the acceleration, we do allow, however, the throttle to vary during trimming in order to match the acceleration defined in the trajectory. The thrust variation in the trajectory is both intentional and due to the atmospheric pressure. So we allow the engines to throttle up and down during trimming to ±99% from their nominal thrust (by setting the throttle parameters to 0.99) in order to match the acceleration. The trajectory data during first stage is in file "*Stage_1.Traj*". It contains parameters versus time that will be used to trim the launch vehicle and calculate its performance in different times. Trimming is used to calculate the engine deflections required to balance the moments and forces along the 4 directions which are important in this application. Trimming also allows us to determine how much CG variation or yawing moment due to engine thrust mismatch the system is able to tolerate along the trajectory. We must first start the Trim program, as shown below, select the data files, and select to plot the trajectory parameters using the Trim main menu.





Select One Data File from Each Menu Category

The following analysis requires some data files to be selected from the current project directory. Select one data file for each category, (some of the categories are optional).

Mass Properties: Zulu.Mass

Surface Hinge Moments: NO DATA FILE

Trajectory Data: Stage_1.Traj

Aero Damping Derivat: NO DATA FILE

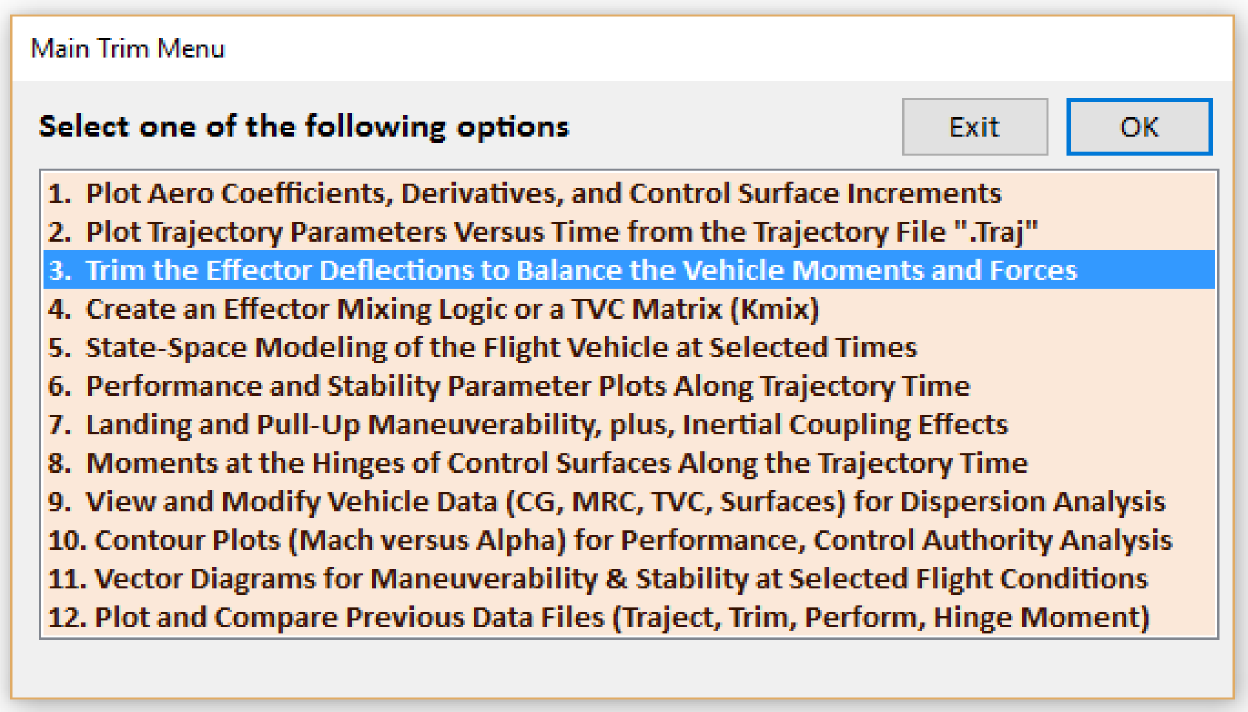
Basic Aero Data: Zulu_Stage1.Aero

Propulsion Data: Zulu_Stage1.Engn

Contr Surface Aero Coeff: NO DATA FILE

Aero Uncertainties: NO DATA FILE

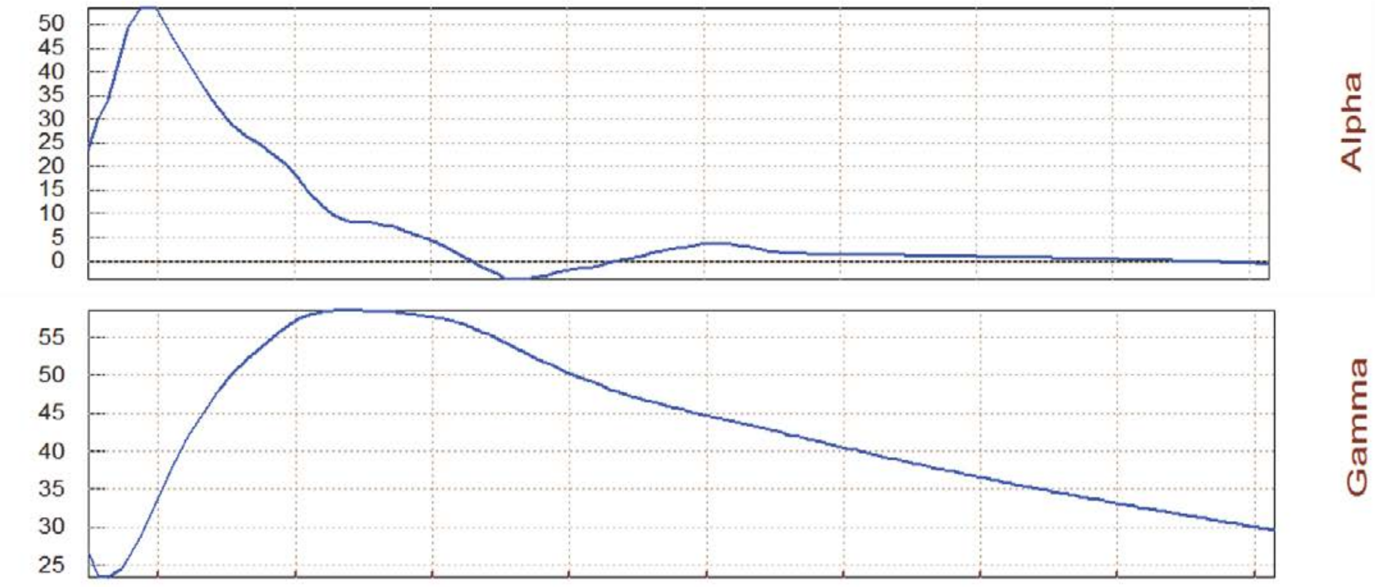
Slosh Parameters: NO DATA FILE



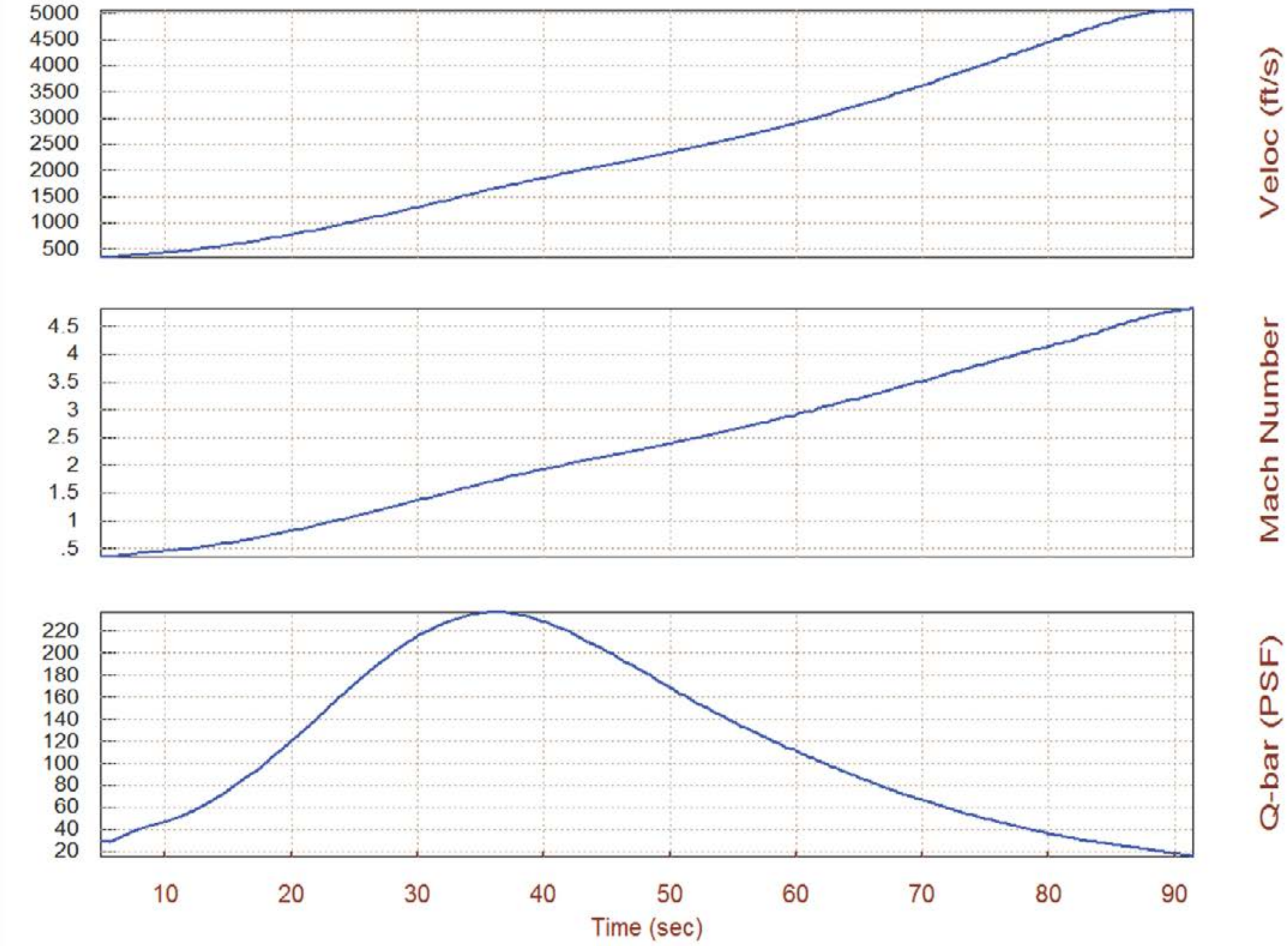
Main Trim Menu

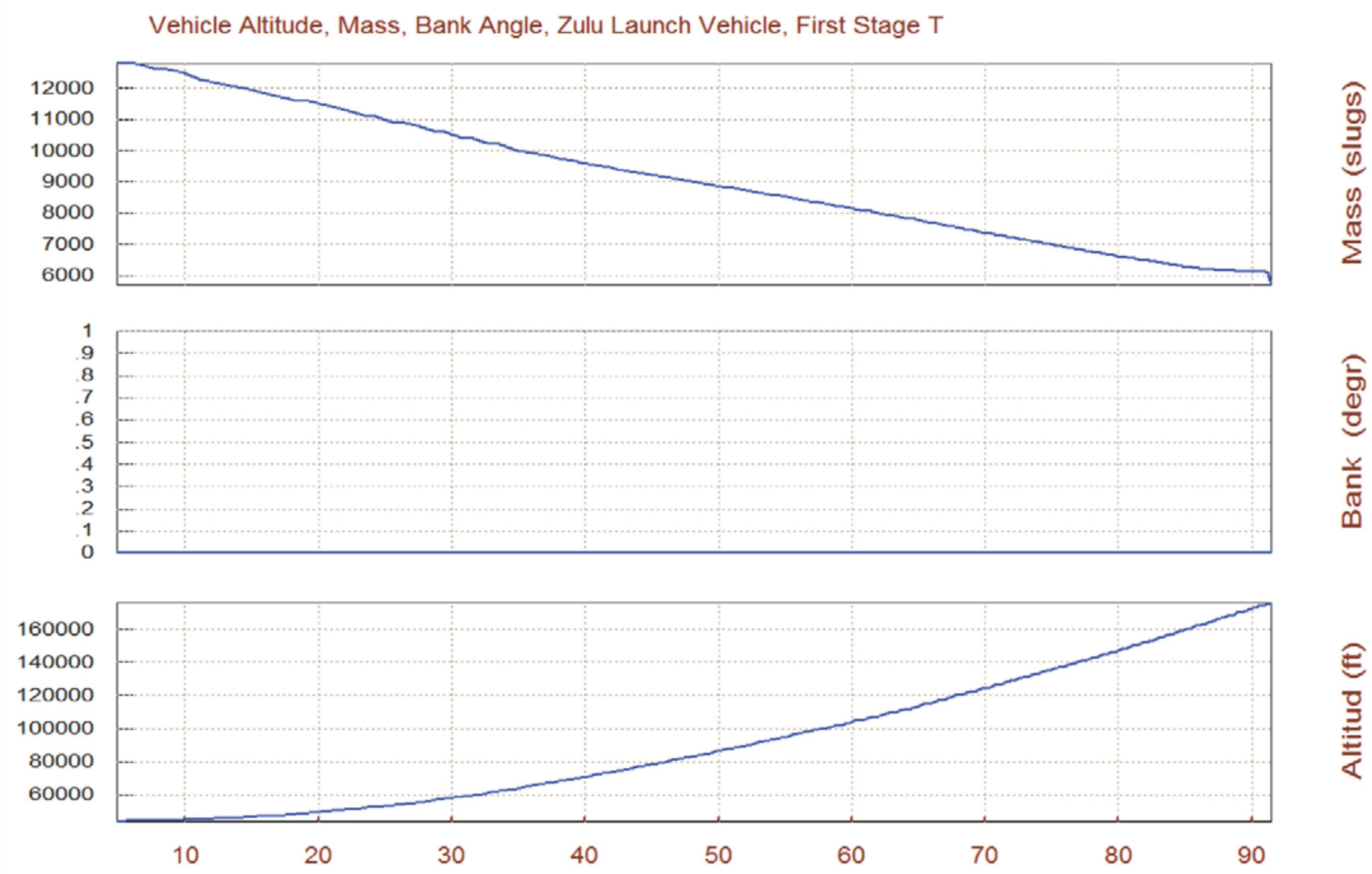
Select one of the following options

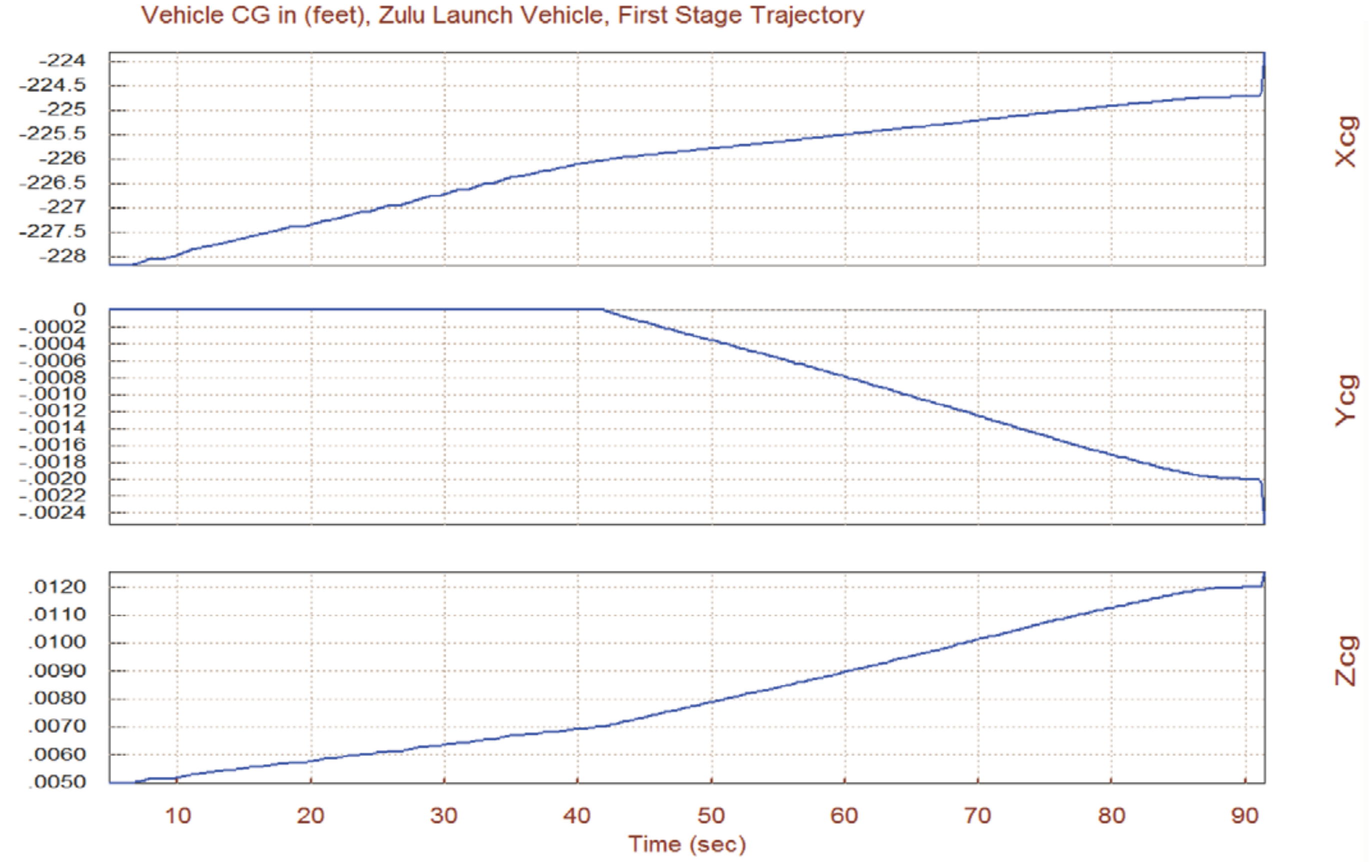
1. Plot Aero Coefficients, Derivatives, and Control Surface Increments
2. Plot Trajectory Parameters Versus Time from the Trajectory File ".Traj"
3. Trim the Effector Deflections to Balance the Vehicle Moments and Forces
4. Create an Effector Mixing Logic or a TVC Matrix (Kmix)
5. State-Space Modeling of the Flight Vehicle at Selected Times
6. Performance and Stability Parameter Plots Along Trajectory Time
7. Landing and Pull-Up Maneuverability, plus, Inertial Coupling Effects
8. Moments at the Hinges of Control Surfaces Along the Trajectory Time
9. View and Modify Vehicle Data (CG, MRC, TVC, Surfaces) for Dispersion Analysis
10. Contour Plots (Mach versus Alpha) for Performance, Control Authority Analysis
11. Vector Diagrams for Maneuverability & Stability at Selected Flight Conditions
12. Plot and Compare Previous Data Files (Traject, Trim, Perform, Hinge Moment)

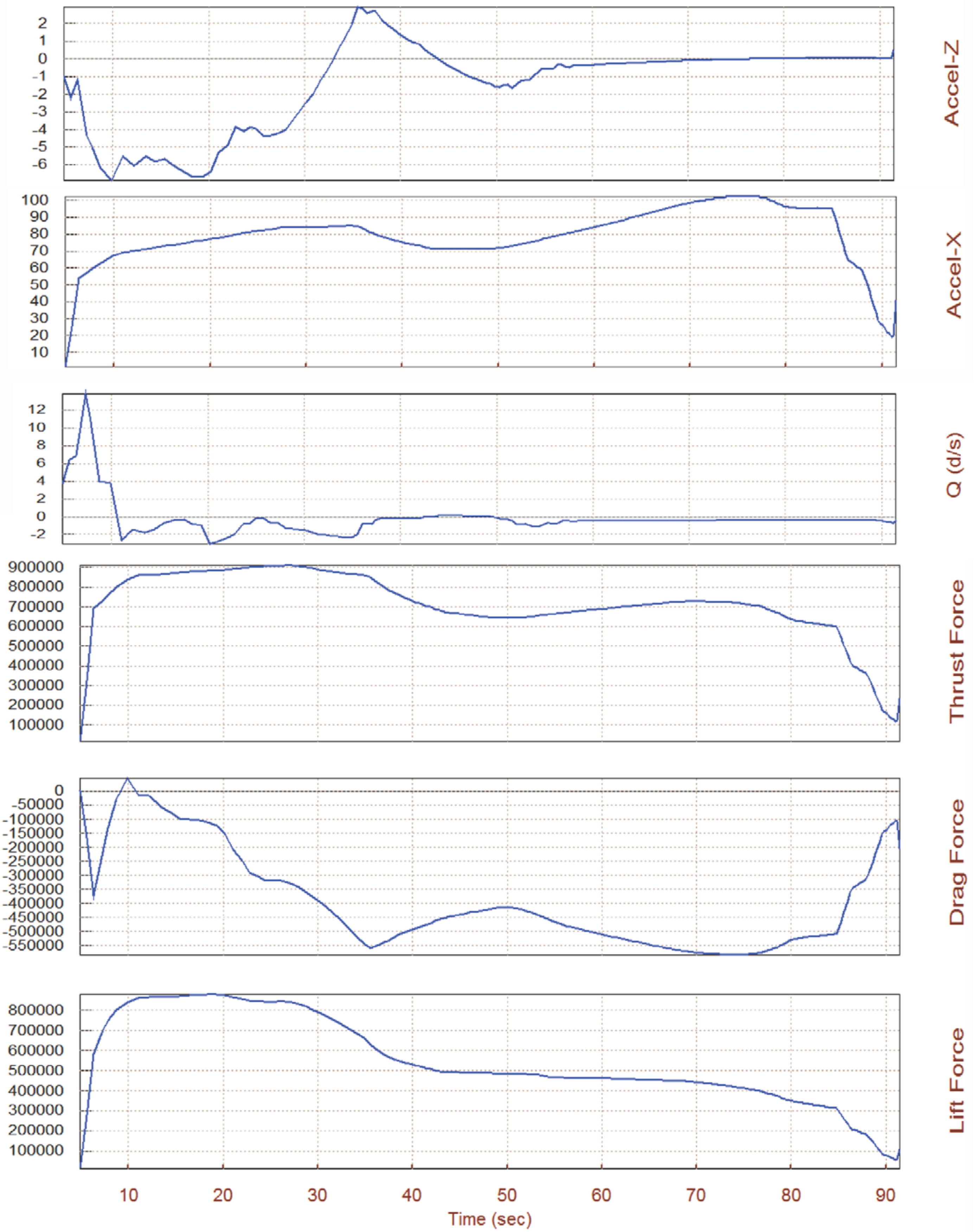


Velocity, Dynamic Pressure, Zulu Launch Vehicle, First Stage Trajectory




Vehicle Altitude, Mass, Bank Angle, Zulu Launch Vehicle, First Stage T
12000
11000
10000
9000
8000
7000
6000
Mass (slugs)
1
.9
.8
.7
.6
.5
.4
.3
.2
.1
0
Bank (degr)
160000
140000
120000
100000
80000
60000
Altitud (ft)
10
20
30
40
50
60
70
80
90


Vehicle CG in (feet), Zulu Launch Vehicle, First Stage Trajectory
-224
-224.5
-225
-225.5
-226
-226.5
-227
-227.5
-228
Xcg
0
-.0002
-.0004
-.0006
-.0008
-.0010
-.0012
-.0014
-.0016
-.0018
-.0020
-.0022
-.0024
Ycg
.0120
.0110
.0100
.0090
.0080
.0070
.0060
.0050
Zcg
10
20
30
40
50
60
70
80
90
Time (sec)



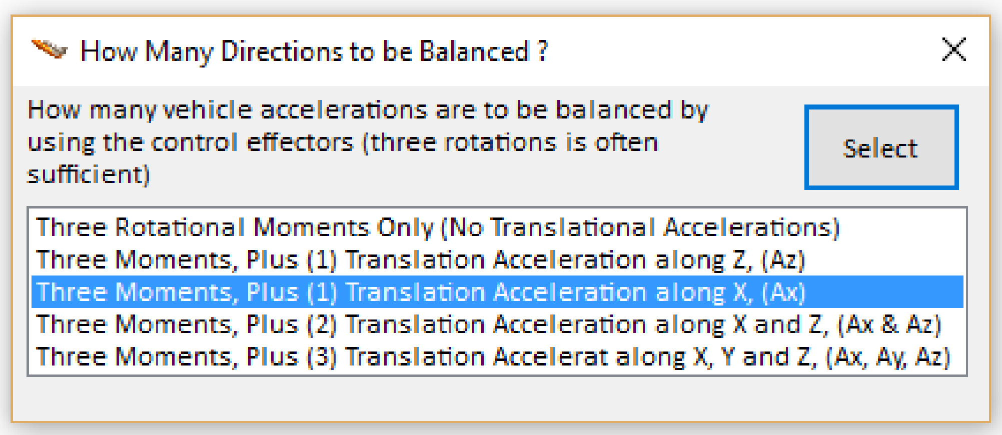
2
1
0
-1
-2
-3
-4
-5
-6
Accel-Z
100
90
80
70
60
50
40
30
20
10
Accel-X
12
10
8
6
4
2
0
-2
Q (d/s)
900000
800000
700000
600000
500000
400000
300000
200000
100000
Thrust Force
0
-50000
-100000
-150000
-200000
-250000
-300000
-350000
-400000
-450000
-500000
-550000
Drag Force
800000
700000
600000
500000
400000
300000
200000
100000
Lift Force
10
20
30
40
50
60
70
80
90
Time (sec)

The vehicle is released with a positive flight path angle γ=45° at 43,000 (feet) from an aircraft that is moving upwards assisted with rocket engines. The angle of attack at release time is: α=5°. The two side booster engines ignite at approximately 5 seconds later to avoid damaging the aircraft. This vehicle is statically unstable and by the time the engines ignite α increases to 25°. So the analysis of the first stage begins at engine ignition since we can't trim or analyze performance without thrust control.

Notice, that as soon as the two engines ignite, the angle of attack and pitch attitude is increased to the maximum in order to raise the flight-path angle γ while the dynamic pressure is reasonably low. The combined thrust is variable and it reaches to 900,000 (lb). The vertical force is positive mainly due to the upward component of thrust and the drag force is negative, meaning that the horizontal force is positive due to thrusting. The altitude reaches 180,000 (feet) at the end of first stage, and the dynamic pressure reaches MaxQ= 230 (psf) at t=35 sec. The CG travel is calculated from the mass properties as a function of the vehicle mass.

Trimming

The next step is to trim the TVC engines and to balance the moments and acceleration. Trimming is selected from the third option of the main menu. You must first define the directions along which to trim the vehicle. We select the three moments plus the x-axis acceleration. The x-axis acceleration is included in order to adjust the vehicle thrust to match the trajectory acceleration. This process calculates the engine deflections and the thrust variations required to match the trajectory moments and forces.

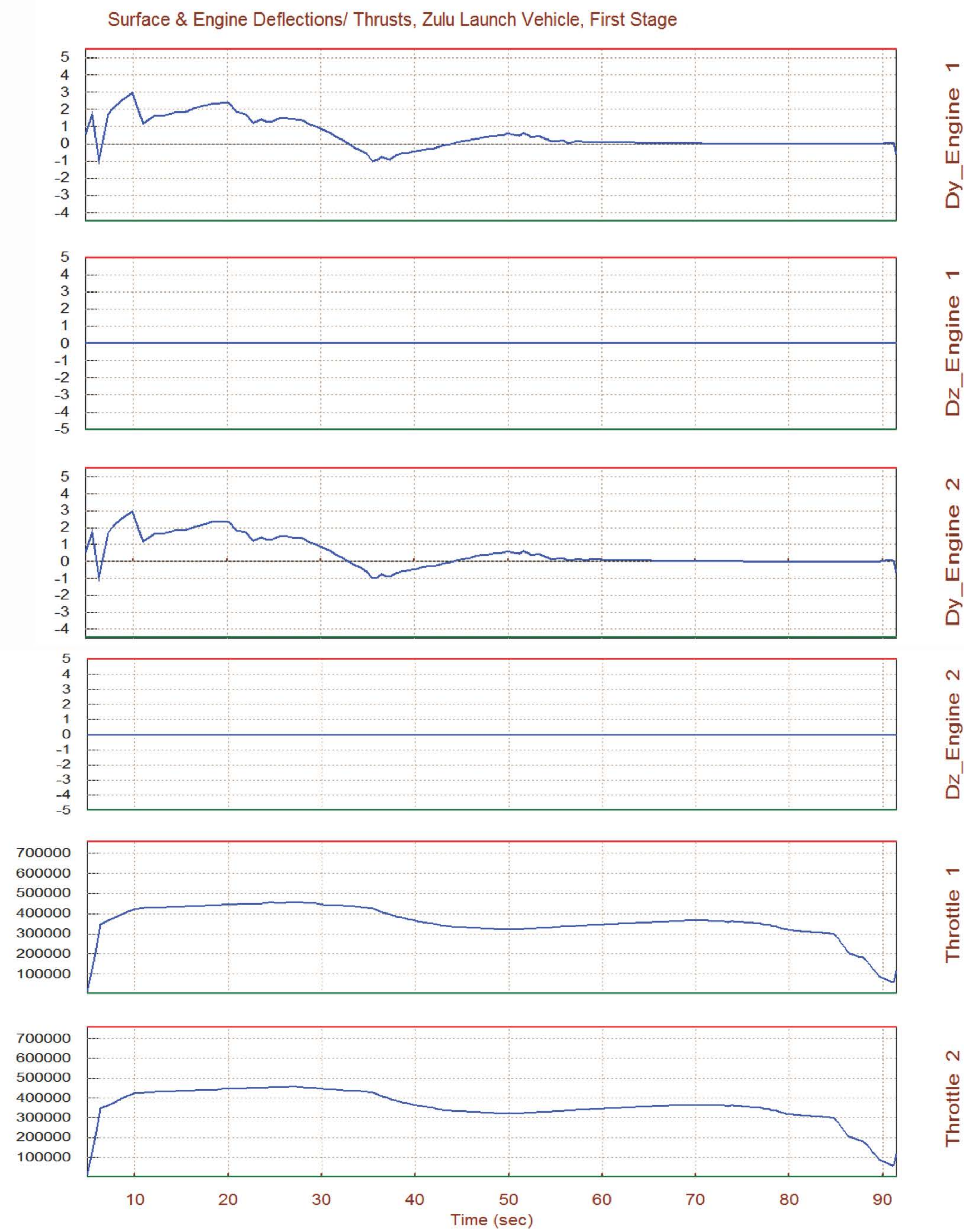


How Many Directions to be Balanced ?

How many vehicle accelerations are to be balanced by using the control effectors (three rotations is often sufficient)

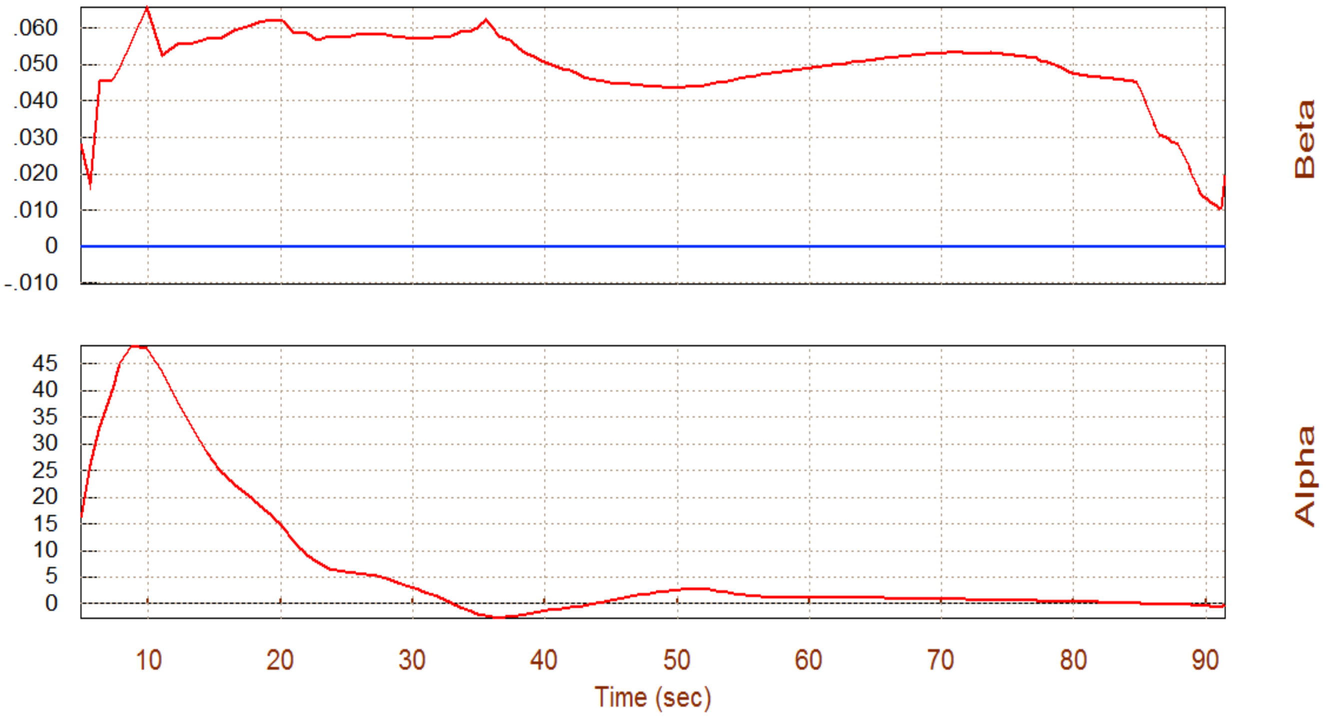
Three Rotational Moments Only (No Translational Accelerations)
Three Moments, Plus (1) Translation Acceleration along Z, (Az)
Three Moments, Plus (1) Translation Acceleration along X, (Ax)
Three Moments, Plus (2) Translation Acceleration along X and Z, (Ax & Az)
Three Moments, Plus (3) Translation Accelerat along X, Y and Z, (Ax, Ay, Az)

The figure below shows the engine gimbal angles and thrusts as a function of time. The deflections are mainly in pitch. The yaw deflections are almost zero. The engine thrusts vary in order to match the trajectory accelerations. The results demonstrate that we have sufficient control authority to trim, i.e. to balance the vehicle moments and forces with the TVC moments and forces.


Surface & Engine Deflections/ Thrusts, Zulu Launch Vehicle, First Stage
Dy_Engine 1
Dz_Engine 1
Dy_Engine 2
Dz_Engine 2
Throttle 1
Throttle 2
5
4
3
2
1
0
-1
-2
-3
-4
-5
700000
600000
500000
400000
300000
200000
100000
10
20
30
40
50
60
70
80
90
Time (sec)

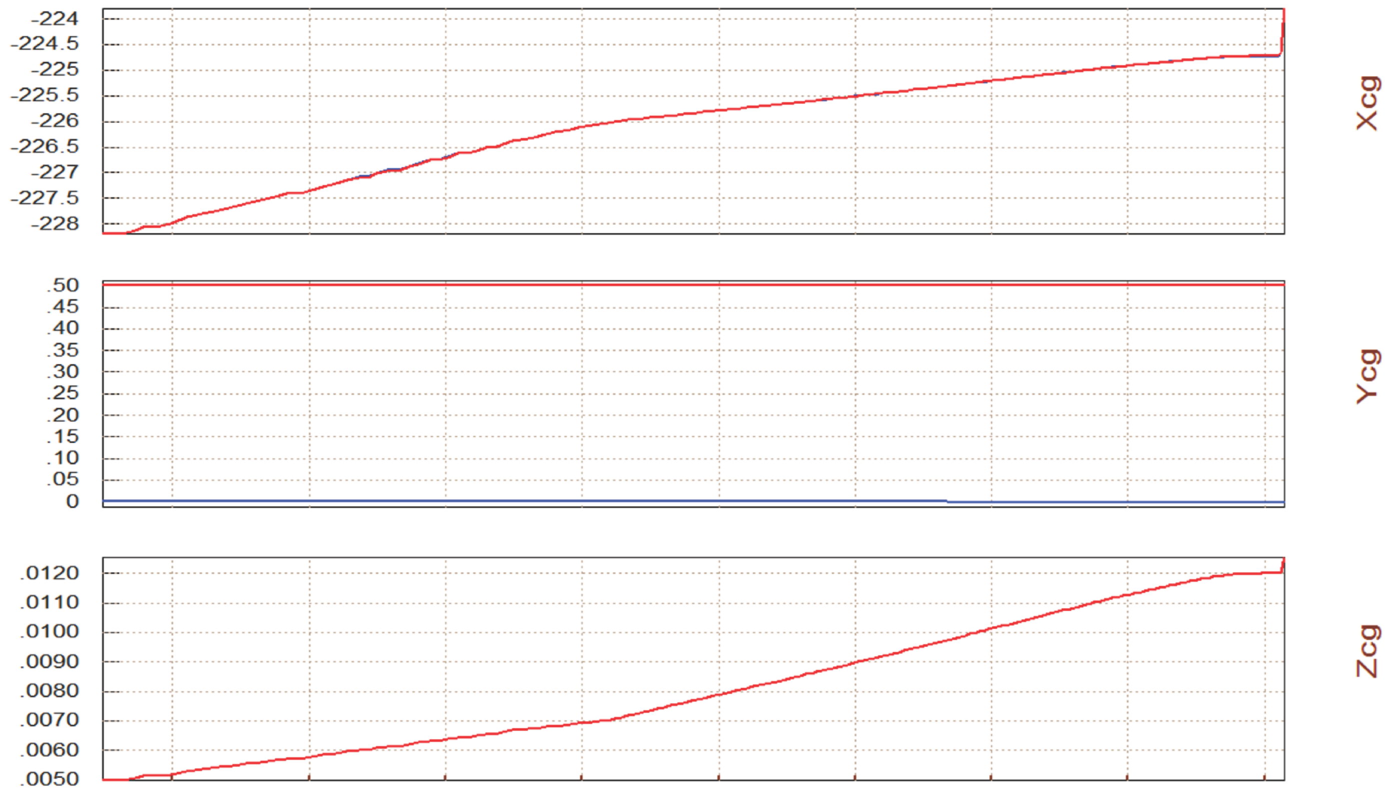
Trimming is not only used along the nominal vehicle trajectory but it can also be used to analyze trajectory dispersions. We do this by modifying the trajectory and retrimming. For example, we may want to check the effects due to CG variations, alpha and beta dispersions due to winds, or the effects of disturbance forces and torques on the trimming results, and to make sure that we still have sufficient TVC authority to trim. In the following example we used the trajectory modification option to modify the nominal trajectory and include Y_{CG} dispersion of 0.5 (feet) and also a constant yawing moment of 1000 (ft-lb) along the entire first stage trajectory representing a thrust mismatch. The modified trajectory was saved in file "*Stage_1c.Traj*" and it is shown against the original trajectory.

We may now retrim using the modified trajectory and compare the trim results against the original trajectory. We used Option-12 from the main menu to overlay the two trajectories and the two trim results on the same plots and compare results. The first thing we notice is that the beta angle from the modified trajectory (red) is no longer zero. The vehicle has a sideslip due to the engines yawing to balance the disturbance moments caused by the thrust mismatch and the Y_{CG} offset.

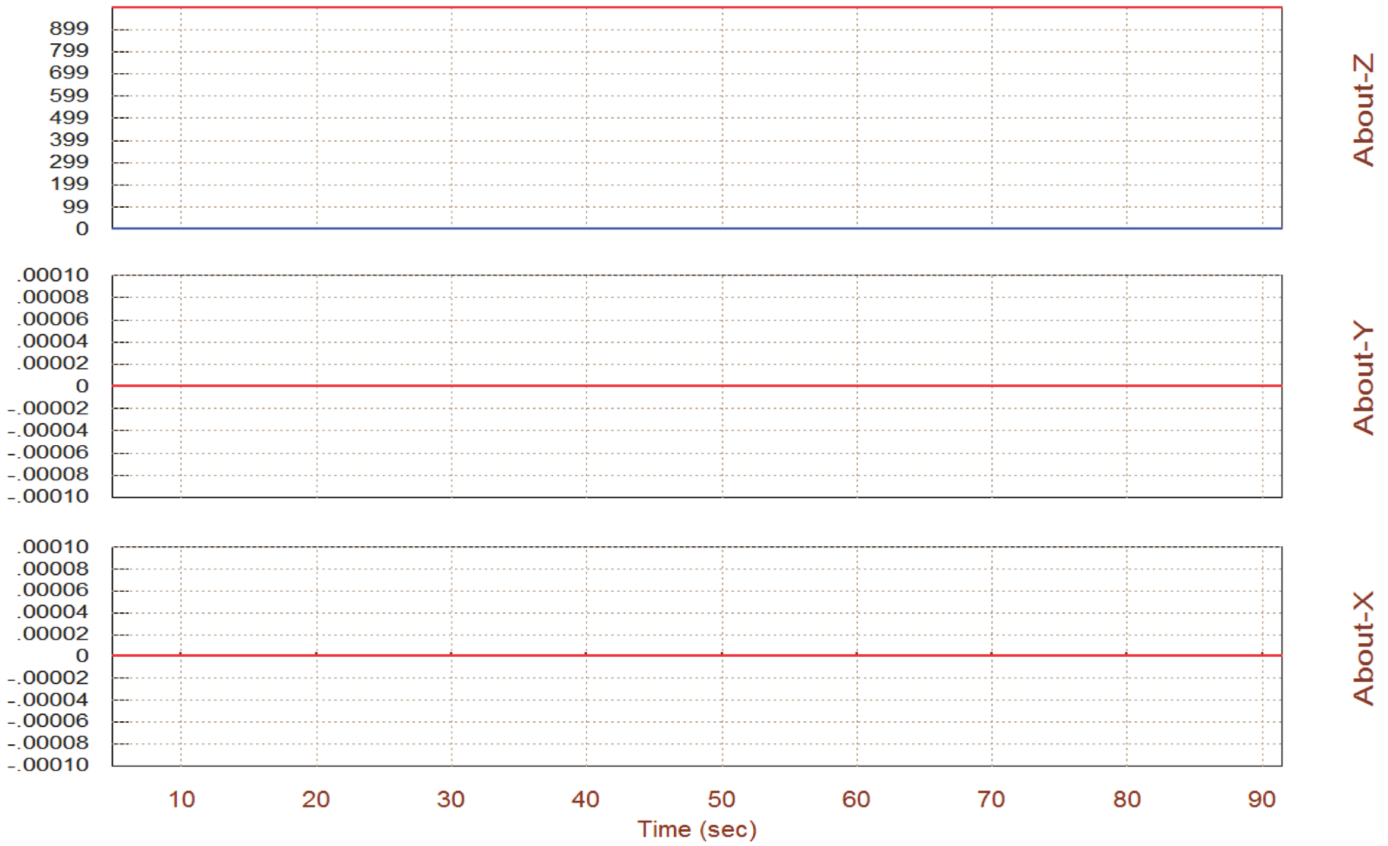


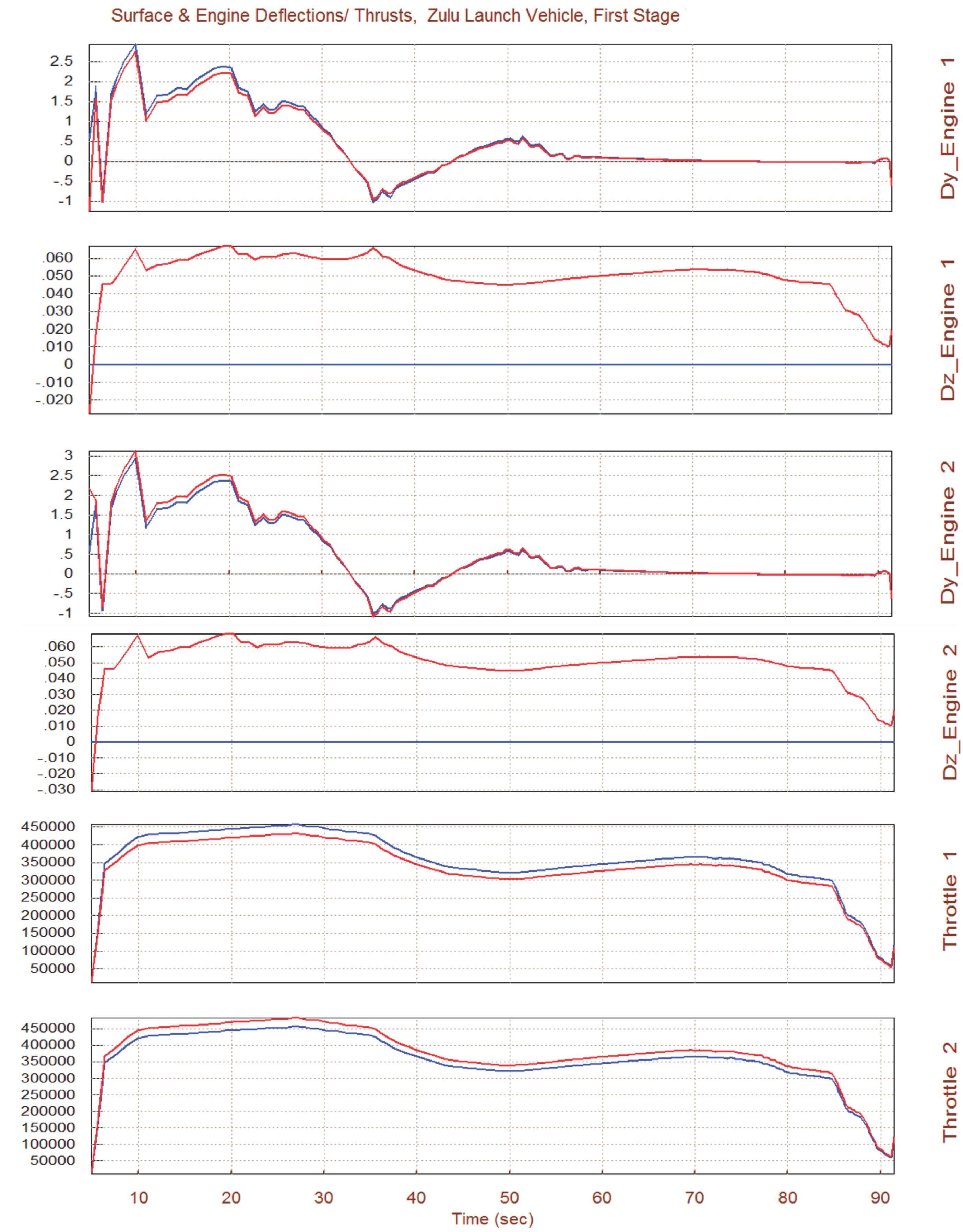
The modified trim requires +yaw deflections in both engines to balance the positive disturbance torque. The pitch engine deflections (red) are slightly different from the original non-dispersed trim (blue) in order to balance the rolling moment due to beta. The thrusts are also slightly different producing a negative yaw torque to counteract the disturbance moment.

Vehicle CG in (feet), Zulu Launch Vehicle, First Stage Trajectory



External Disturbance Torques (ft-lb), Zulu Launch Vehicle, First Stage




Surface & Engine Deflections/ Thrusts, Zulu Launch Vehicle, First Stage
Dy_Engine 1
2.5
2
1.5
1
.5
0
-.5
-1
Dz_Engine 1
.060
.050
.040
.030
.020
.010
0
-.010
-.020
Dy_Engine 2
3
2.5
2
1.5
1
.5
0
-.5
-1
Dz_Engine 2
.060
.050
.040
.030
.020
.010
0
-.010
-.020
-.030
Throttle 1
450000
400000
350000
300000
250000
200000
150000
100000
50000
Throttle 2
450000
400000
350000
300000
250000
200000
150000
100000
50000
10
20
30
40
50
60
70
80
90
Time (sec)

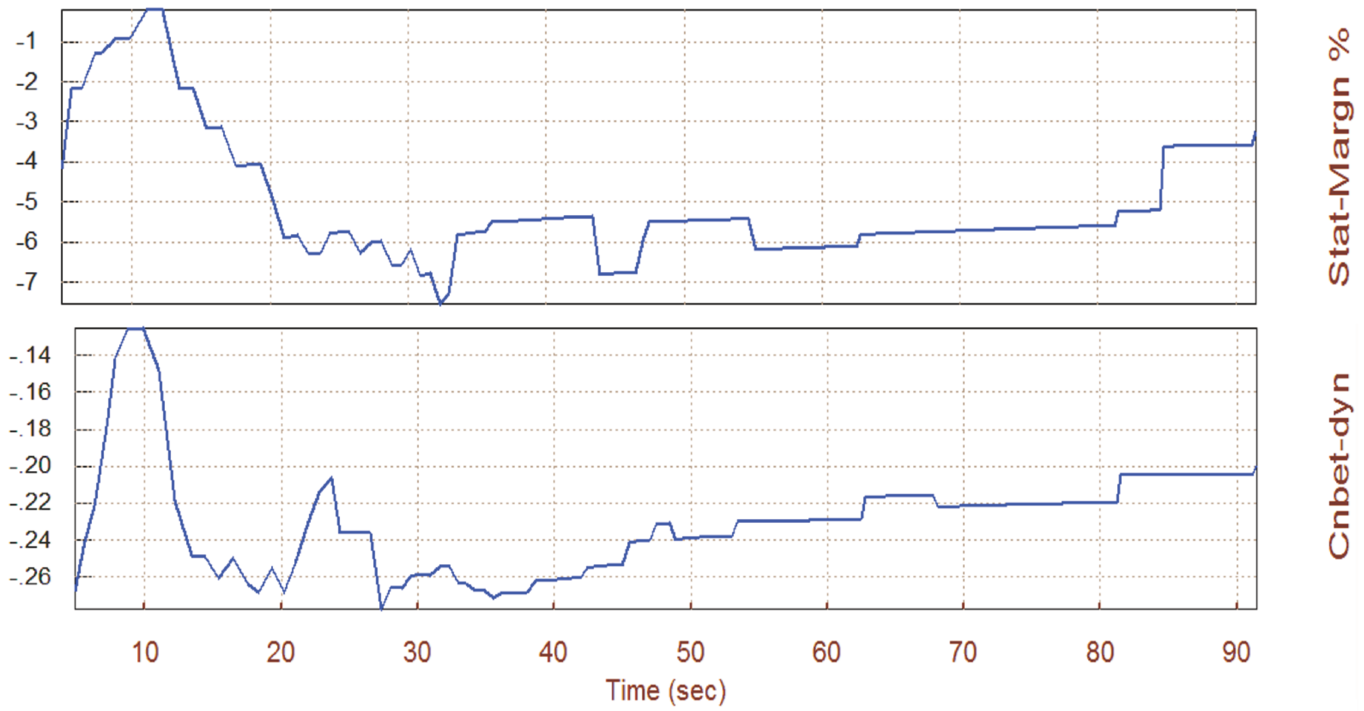
Static Performance Analysis during First Stage

The static performance parameters along the trajectory are obtained by selecting Option-6 from the main menu. The performance parameters calculation requires a mixing logic matrix that will combine the TVC effectors (2 pitch, 2 yaw, and 2 throttle) to achieve the 4 accelerations demanded in roll, pitch, yaw, and axial accelerations. In this case we don't have a pre-selected matrix and we shall allow the Trim program to calculate the TVC matrix from the vehicle data.

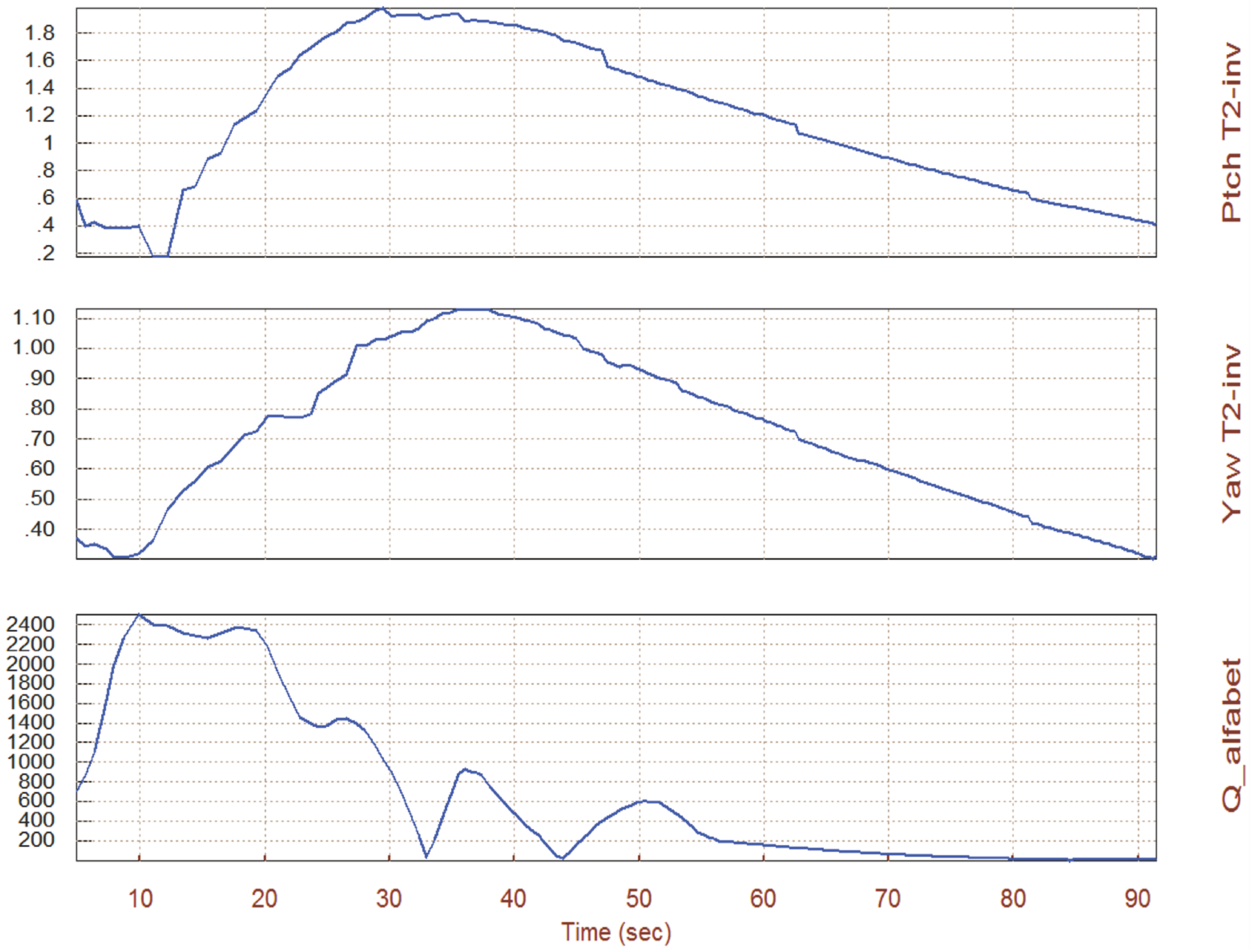
The following plots show the performance parameters results. The static margin and $C_{n\beta}$-dynamic parameters indicate that the vehicle is statically unstable like most launch vehicles. The T2-inverse parameter in pitch and yaw indicate similar static instability. The pitch axis is more unstable with a time to double amplitude slightly more than 0.5 sec, and the yaw time to double amplitude is approximately 0.9 seconds. The amount of instability is acceptable in both directions, and those parameters will define the actuator bandwidth and maximum rate. The Q-alpha parameter defines the aerodynamic loading on the vehicle structure and it should be maintained below 3,500 (psf-deg), otherwise structural damage may occur. The LCDP defines dynamic roll controllability and it is excellent in this vehicle because it is close to one. The rotational control authority in roll, pitch, and yaw is also very good demonstrating that the effector system is capable of tolerating $\pm 5^o$ alpha and beta dispersions. That is, the control effort against the variations is less than 1 (or half for margin). The pitch controllability against ±50 (feet/sec) air-speed variations is also very good (very little effort is required). The next four plots show the maximum accelerations that can be attained in the four control directions by maximizing the control demands in the positive (blue) and in the negative (green) directions from trim. That is, roll, pitch, and yaw in (rad/sec^2) by gimbaling the TVC, and axial acceleration through thrust variations in (feet/sec^2).

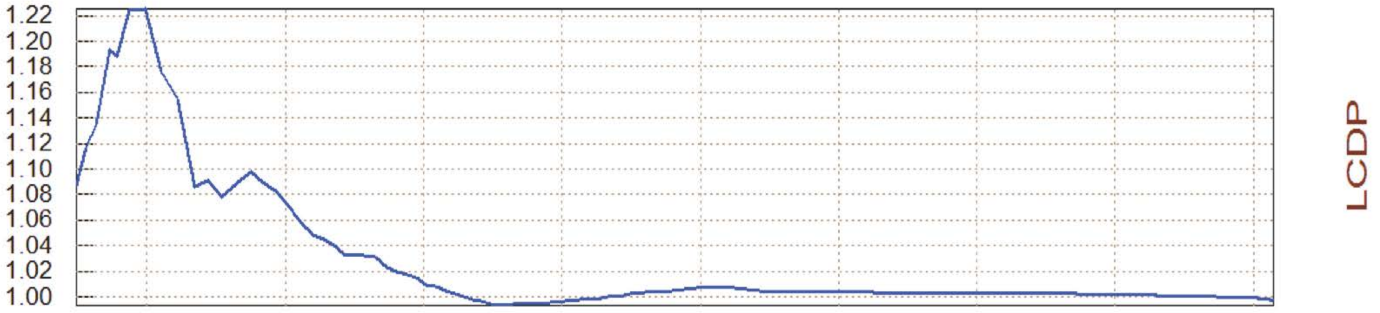
Contour Plots

The performance results are also presented in contour plots which are a more elegant way of picturing the parameters in the entire Mach versus alpha range. The contour plots help us to identify unacceptable regions in the entire Mach versus alpha range and to properly shape the trajectory in order to avoid those regions. In this example we do not have any bad regions.

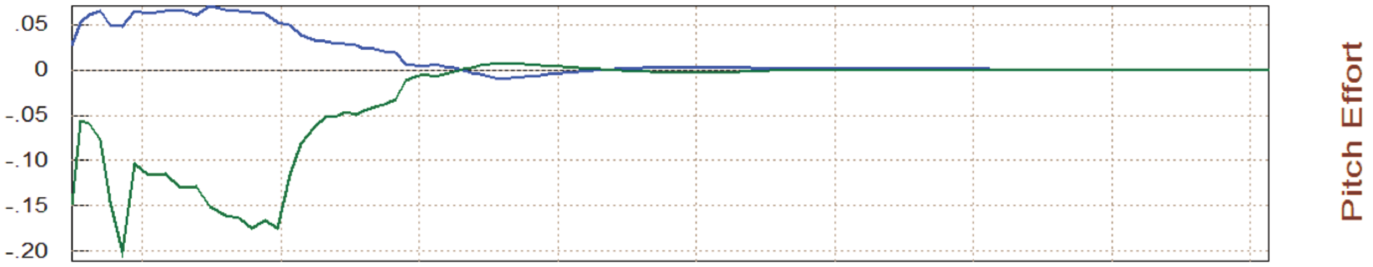


Short-Period (w)/ Time-to-Double-Ampl-Inverse (/sec), Q_alpha_beta (deg-lb/ft^2)

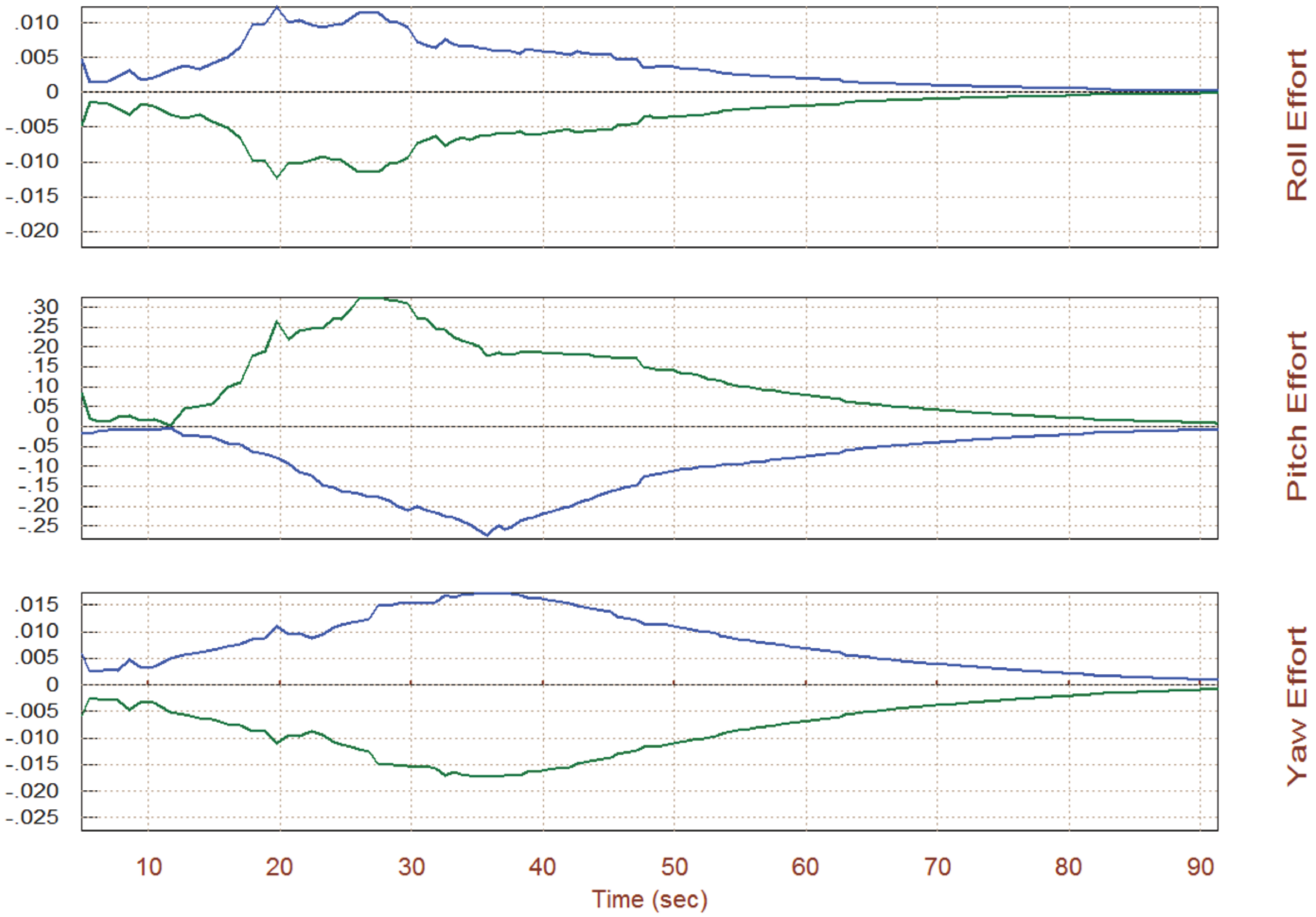


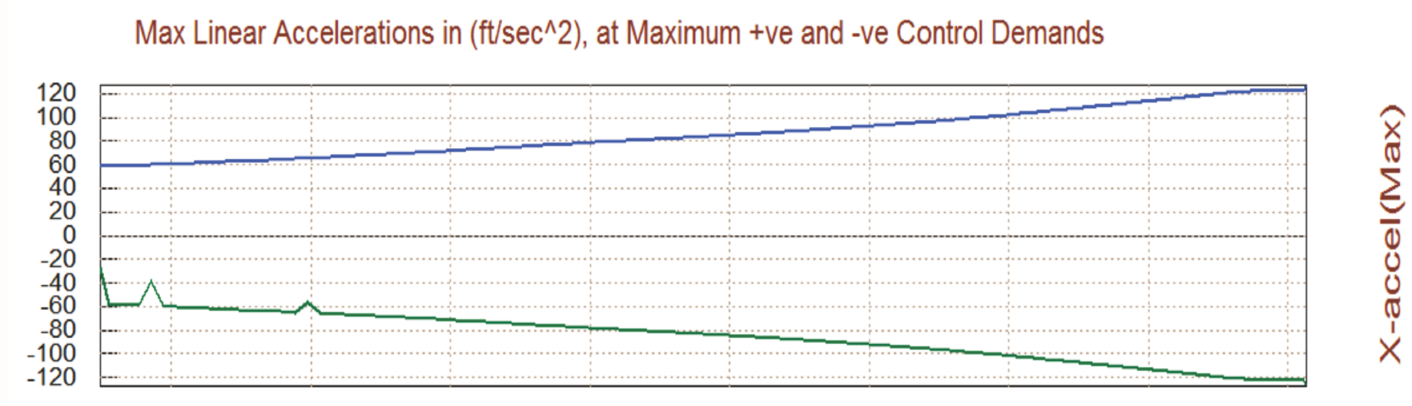


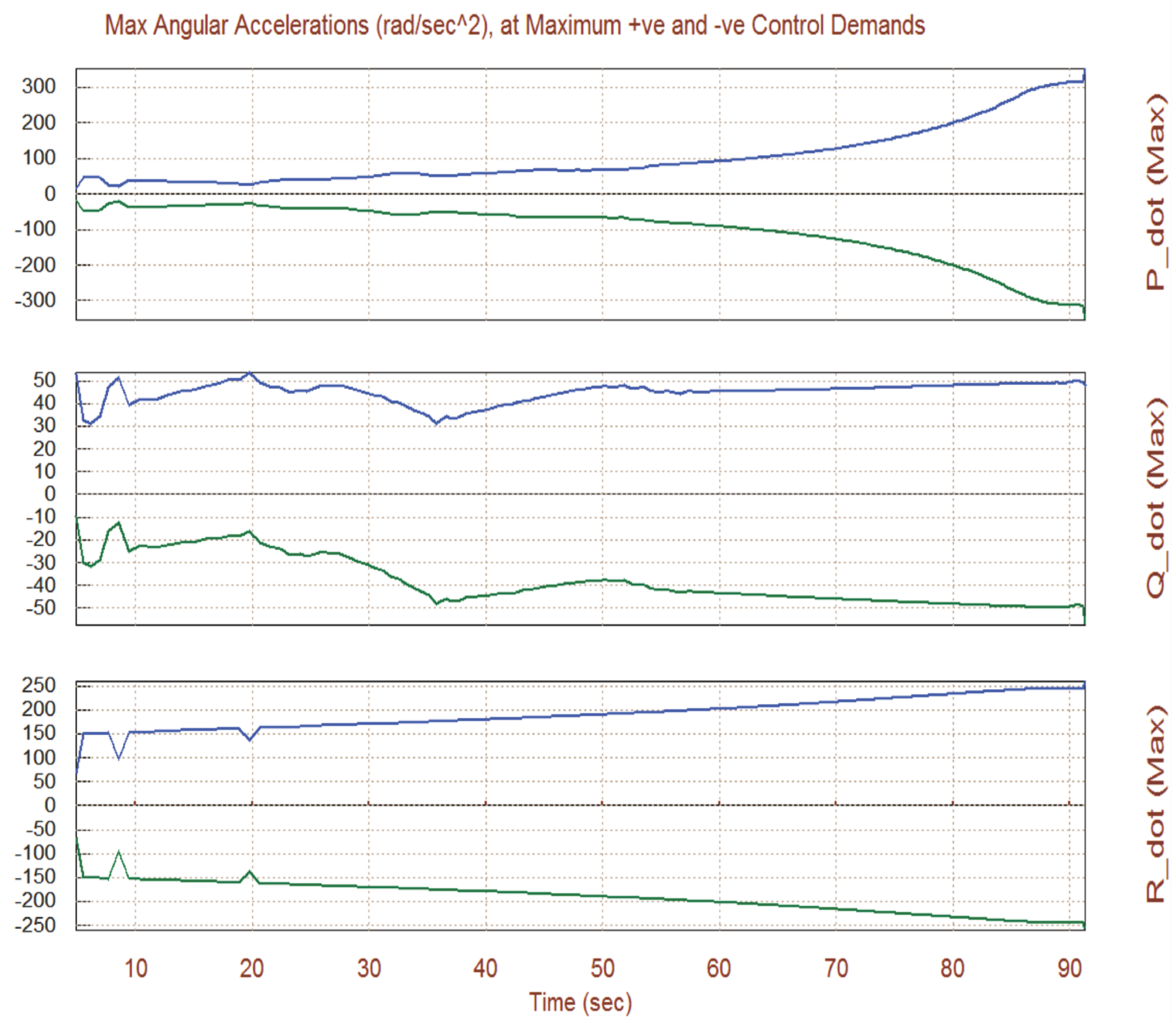
Rotational Control Authority |dQ/dQmax|<1 Against +Vmax & -Vmax Veloc Variations

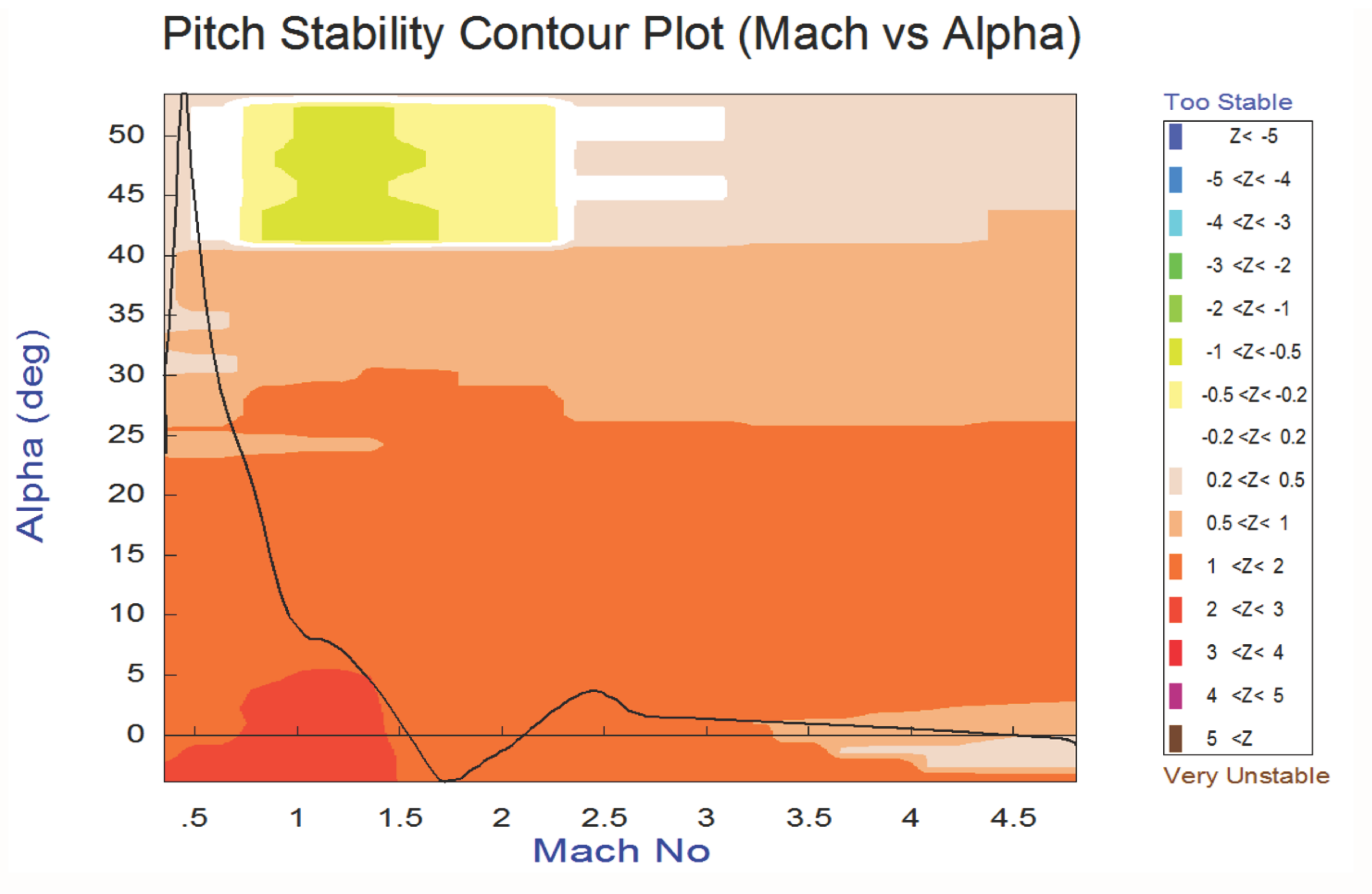


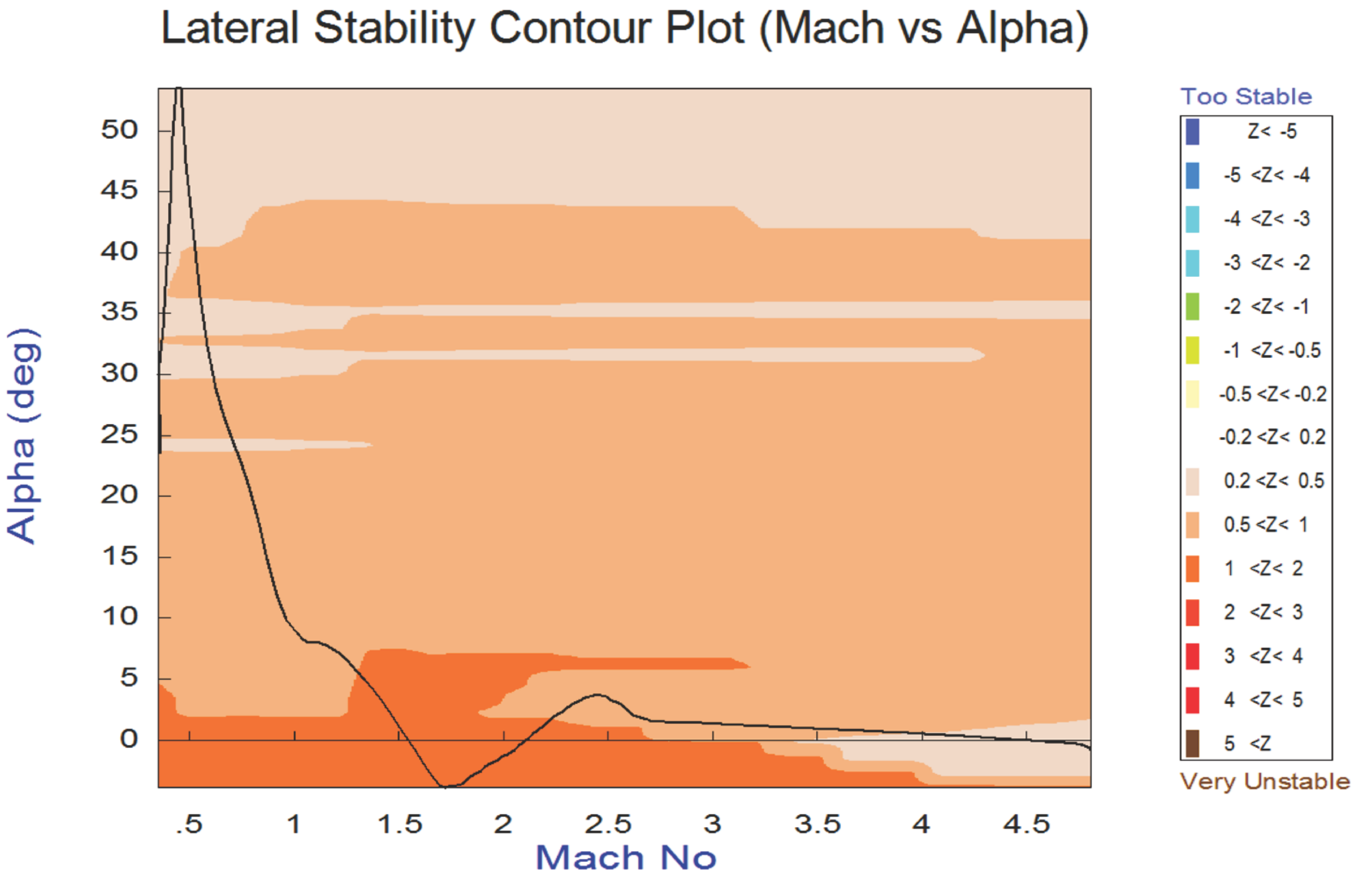
Rotation Control Authority |dQ/dQmax|<1 for 5 (deg) of Alpha & Beta Variation

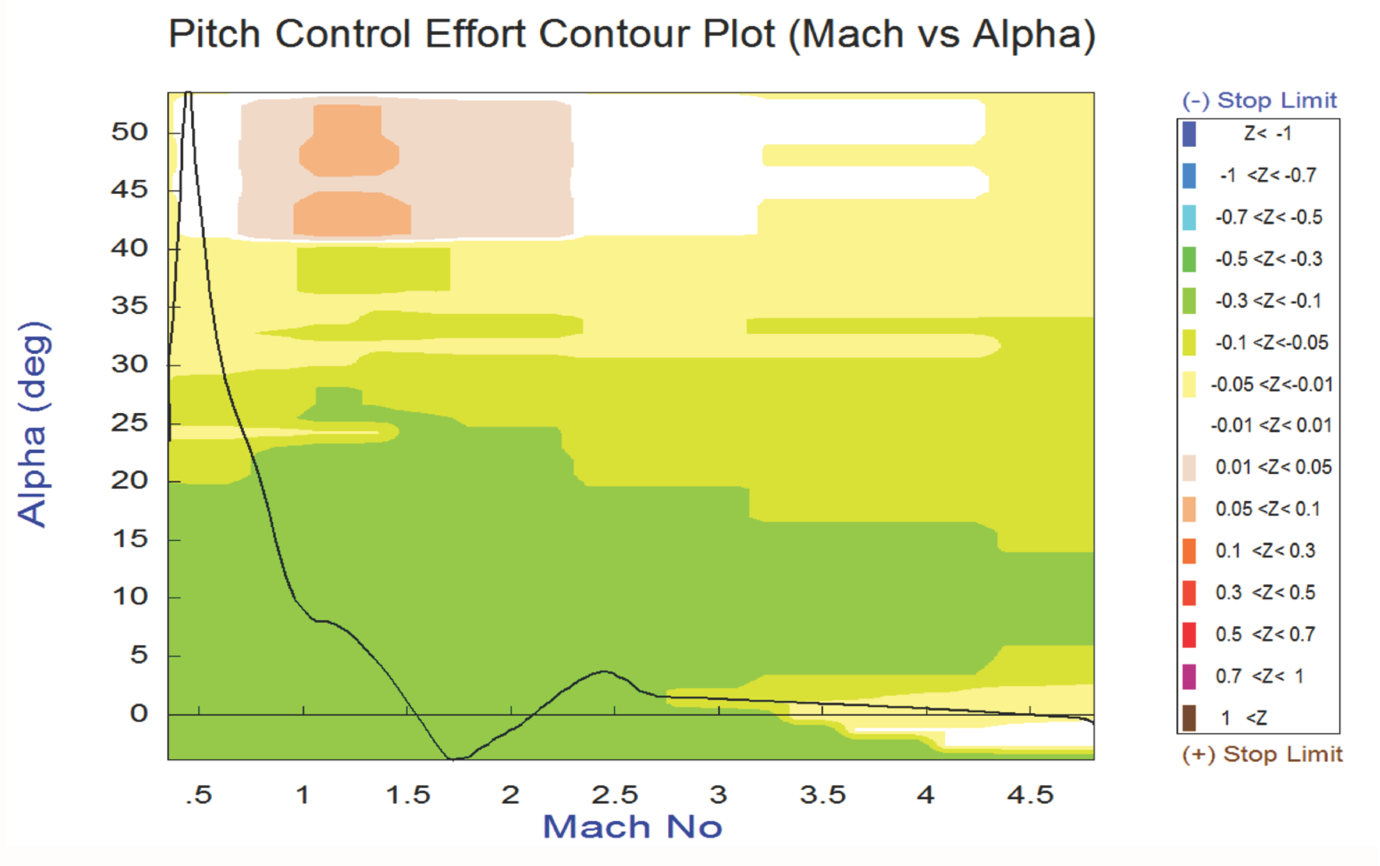


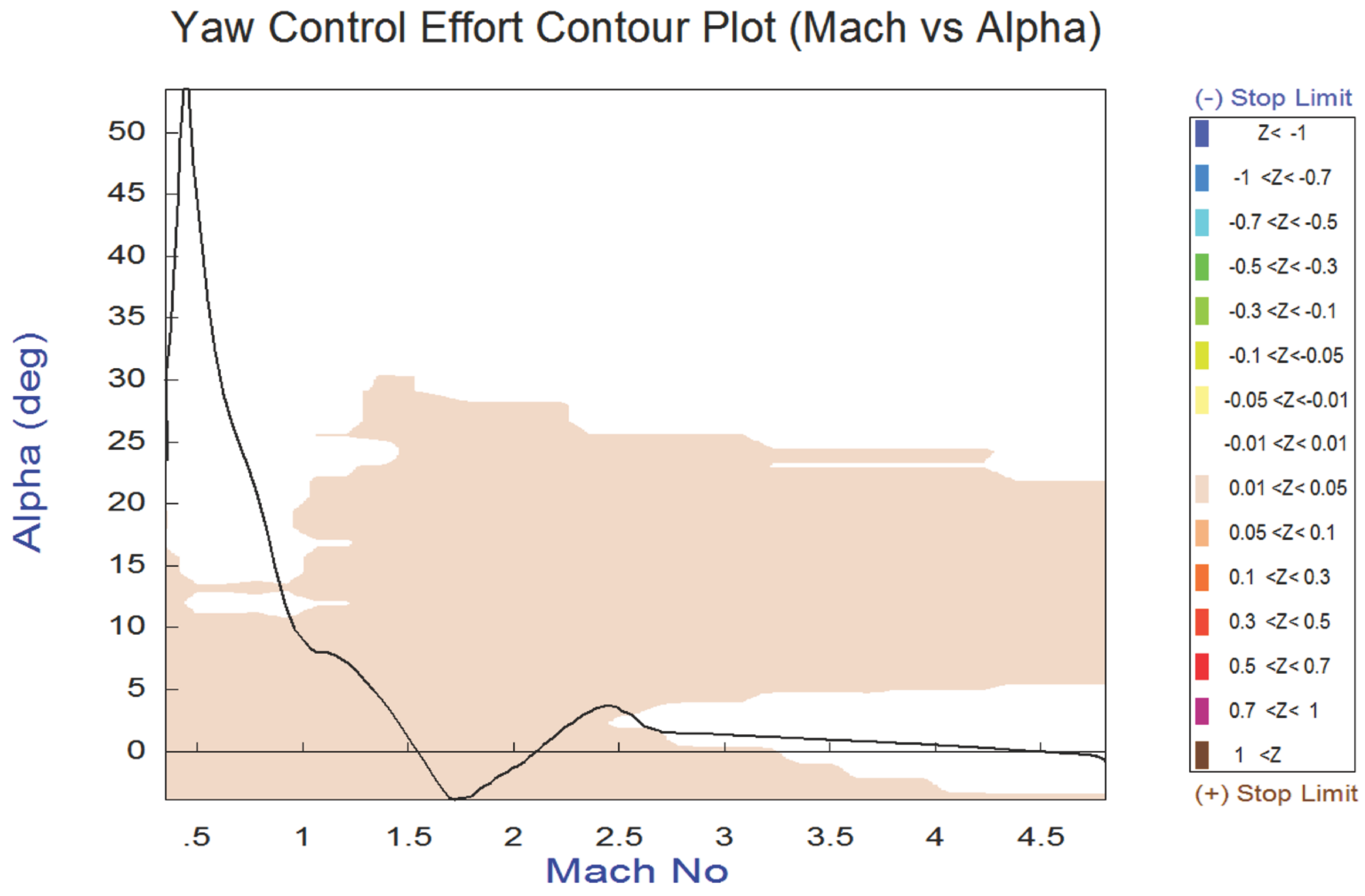

Max Linear Accelerations in (ft/sec^2), at Maximum +ve and -ve Control Demands
120
100
80
60
40
20
0
-20
-40
-60
-80
-100
-120
X-accel(Max)


Max Angular Accelerations (rad/sec^2), at Maximum +ve and -ve Control Demands
300
200
100
0
-100
-200
-300
P_dot (Max)
50
40
30
20
10
0
-10
-20
-30
-40
-50
Q_dot (Max)
250
200
150
100
50
0
-50
-100
-150
-200
-250
R_dot (Max)
10
20
30
40
50
60
70
80
90
Time (sec)

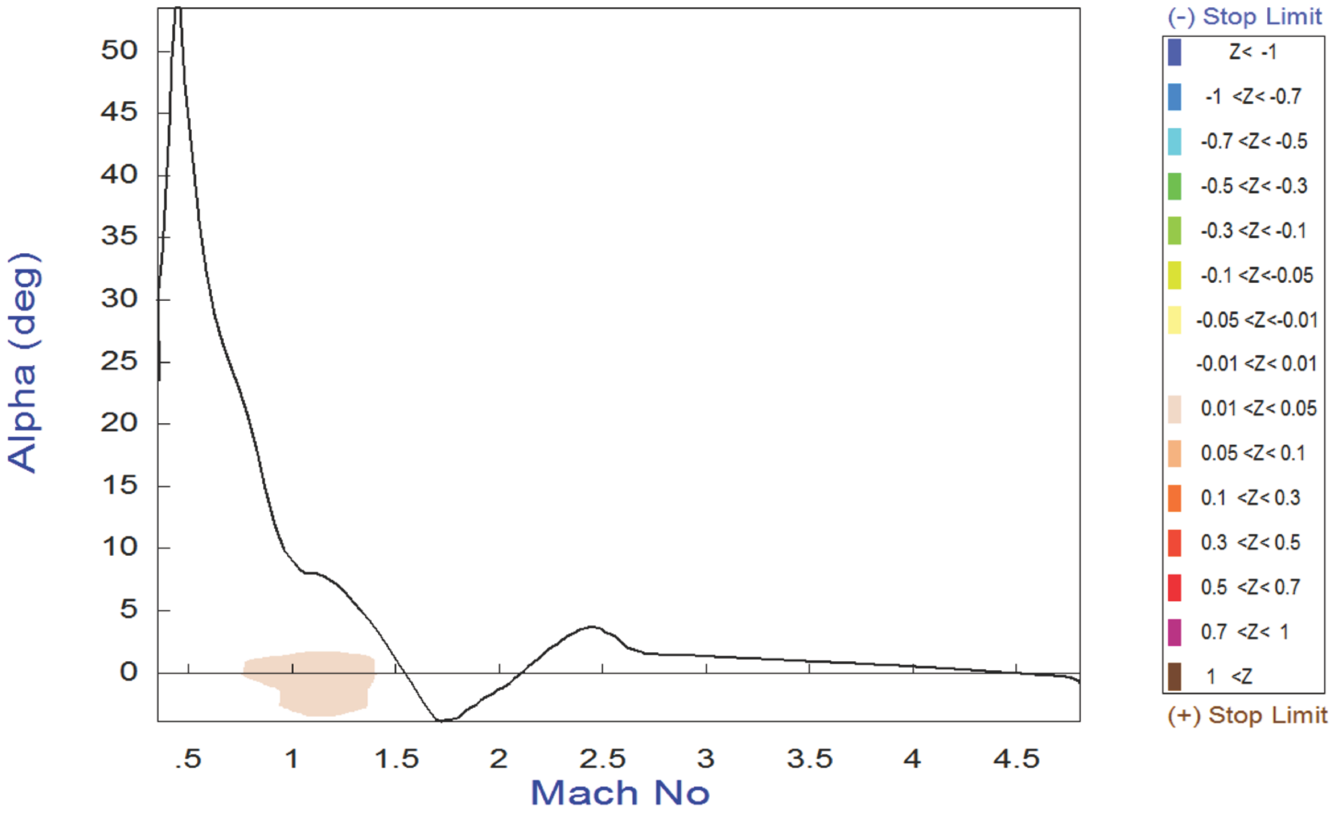

Pitch Stability Contour Plot (Mach vs Alpha)
Alpha (deg)
50
45
40
35
30
25
20
15
10
5
0
.5
1
1.5
2
2.5
3
3.5
4
4.5
Mach No
Too Stable
Z< -5
-5 <Z< -4
-4 <Z< -3
-3 <Z< -2
-2 <Z< -1
-1 <Z< -0.5
-0.5 <Z< -0.2
-0.2 <Z< 0.2
0.2 <Z< 0.5
0.5 <Z< 1
1 <Z< 2
2 <Z< 3
3 <Z< 4
4 <Z< 5
5 <Z
Very Unstable


Lateral Stability Contour Plot (Mach vs Alpha)
Alpha (deg)
50
45
40
35
30
25
20
15
10
5
0
.5
1
1.5
2
2.5
3
3.5
4
4.5
Mach No
Too Stable
Z< -5
-5 <Z< -4
-4 <Z< -3
-3 <Z< -2
-2 <Z< -1
-1 <Z< -0.5
-0.5 <Z< -0.2
-0.2 <Z< 0.2
0.2 <Z< 0.5
0.5 <Z< 1
1 <Z< 2
2 <Z< 3
3 <Z< 4
4 <Z< 5
5 <Z
Very Unstable

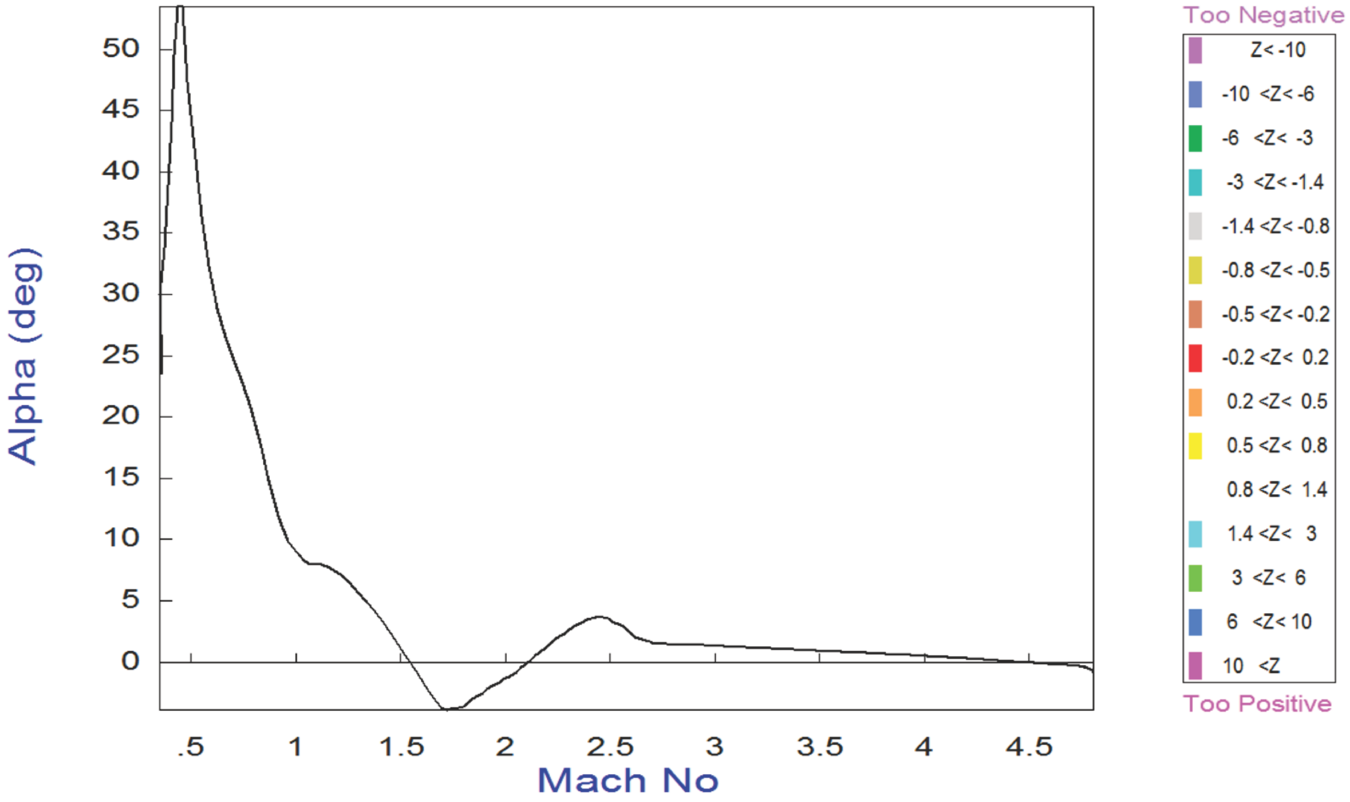

Pitch Control Effort Contour Plot (Mach vs Alpha)
Alpha (deg)
50
45
40
35
30
25
20
15
10
5
0
.5
1
1.5
2
2.5
3
3.5
4
4.5
Mach No
(-) Stop Limit
Z< -1
-1 <Z< -0.7
-0.7 <Z< -0.5
-0.5 <Z< -0.3
-0.3 <Z< -0.1
-0.1 <Z<-0.05
-0.05 <Z<-0.01
-0.01 <Z< 0.01
0.01 <Z< 0.05
0.05 <Z< 0.1
0.1 <Z< 0.3
0.3 <Z< 0.5
0.5 <Z< 0.7
0.7 <Z< 1
1 <Z
(+) Stop Limit


Yaw Control Effort Contour Plot (Mach vs Alpha)
Alpha (deg)
50
45
40
35
30
25
20
15
10
5
0
.5
1
1.5
2
2.5
3
3.5
4
4.5
Mach No
(-) Stop Limit
Z< -1
-1 <Z< -0.7
-0.7 <Z< -0.5
-0.5 <Z< -0.3
-0.3 <Z< -0.1
-0.1 <Z<-0.05
-0.05 <Z<-0.01
-0.01 <Z< 0.01
0.01 <Z< 0.05
0.05 <Z< 0.1
0.1 <Z< 0.3
0.3 <Z< 0.5
0.5 <Z< 0.7
0.7 <Z< 1
1 <Z
(+) Stop Limit

Roll Control Effort Contour Plot (Mach vs Alpha)



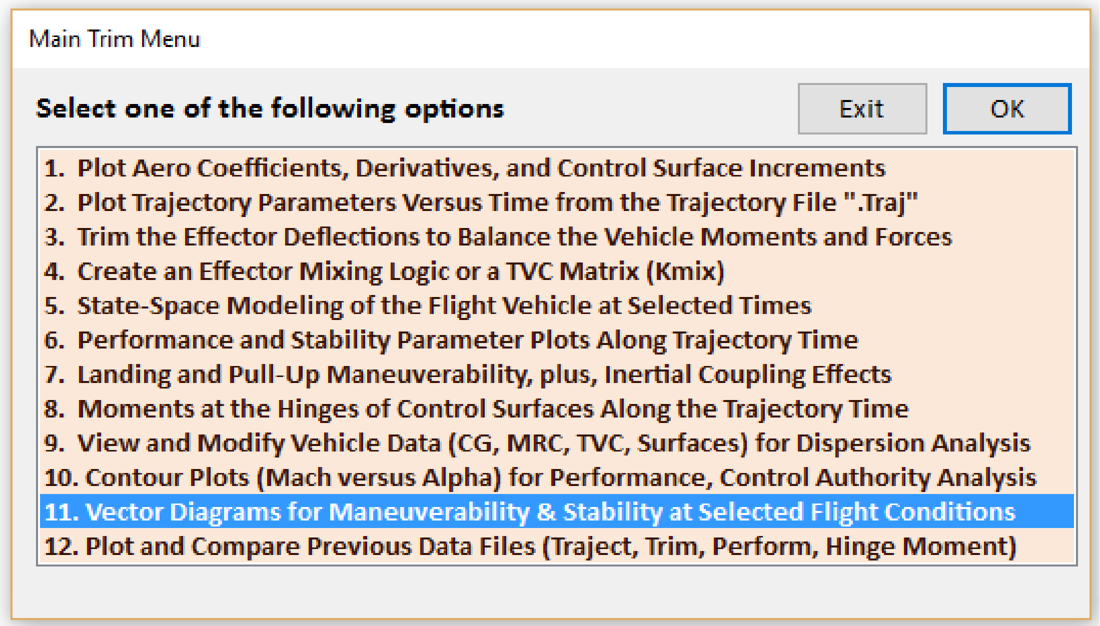
Roll Departure (LCDP) Contour Plot (Mach vs Alpha)

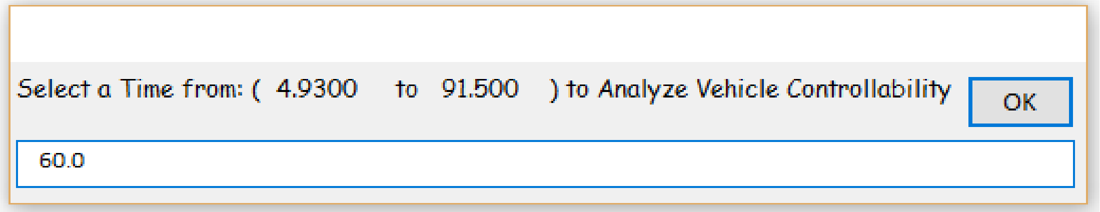


Vector Diagrams

Controllability at specific flight conditions is also analyzed by means of vector diagrams. They present in vector form the maximum moments or maximum accelerations generated from the TVC in this case, against the moments or accelerations generated due to alpha and beta dispersions from trim. They tell us if we have sufficient controllability and in the proper direction to counteract against the dispersion disturbance. The acceleration partials demonstrate the efficiency of our TVC mixing logic. That is, if the acceleration per acceleration demand is of the proper magnitude and direction. Ideally they should be unit vectors pointing in the corresponding directions, although this is not always necessary.

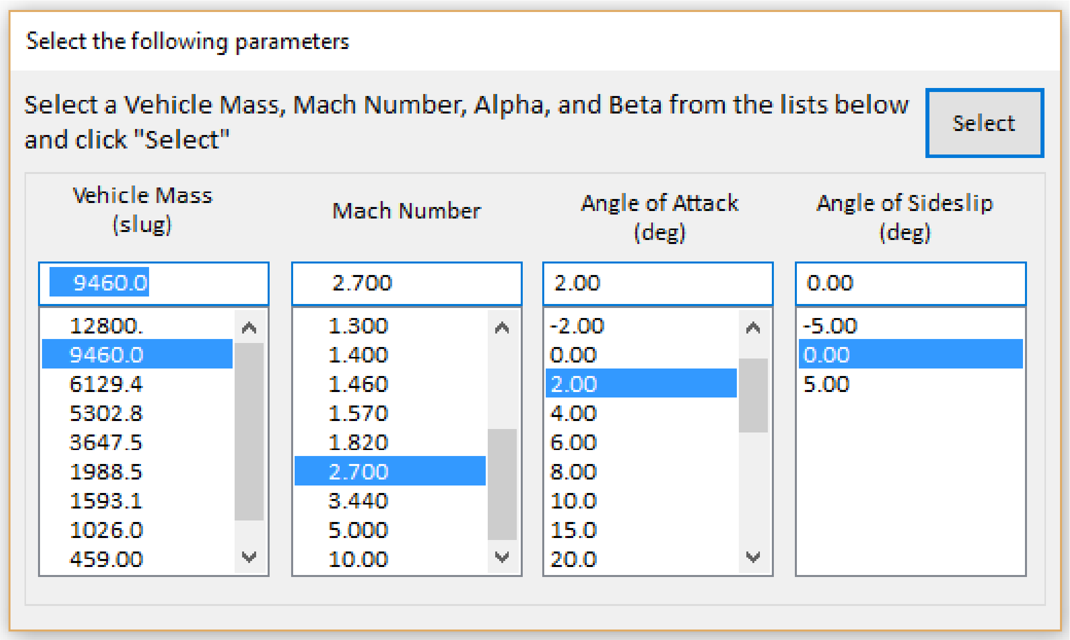
The vector diagrams are created by selecting Option-11 from the main menu. We must first select the flight condition to be analyzed by entering the trajectory time. The menus in the next dialog confirm the flight condition in terms of vehicle mass, Mach number, alpha, and beta. The user may change these default values, if he wishes. In the next dialog we must enter the maximum alpha and beta dispersion angles from trim, and also the maximum airspeed variation from V_0. They are used to define the disturbance moments or accelerations. Finally the program requires a mixing-logic matrix in order to calculate the control moments. You may select a pre-calculated matrix from the systems file or in this case, since we do not have a matrix, we shall allow the program to calculate it by selecting the middle option in the next dialog. The results demonstrate that we have excellent controllability in this flight condition in all three rotations and also in the axial acceleration. The disturbances are small in comparison with the controls.





Select a Time from: (4.9300 to 91.500) to Analyze Vehicle Controllability

60.0


Select the following parameters

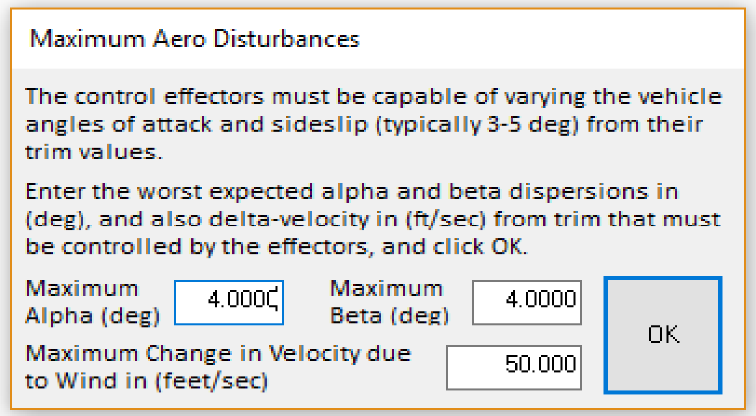
Select a Vehicle Mass, Mach Number, Alpha, and Beta from the lists below and click "Select"

Vehicle Mass (slug): 9460.0
12800.
9460.0
6129.4
5302.8
3647.5
1988.5
1593.1
1026.0
459.00

Mach Number: 2.700
1.300
1.400
1.460
1.570
1.820
2.700
3.440
5.000
10.00

Angle of Attack (deg): 2.00
-2.00
0.00
2.00
4.00
6.00
8.00
10.0
15.0
20.0

Angle of Sideslip (deg): 0.00
-5.00
0.00
5.00


Maximum Aero Disturbances

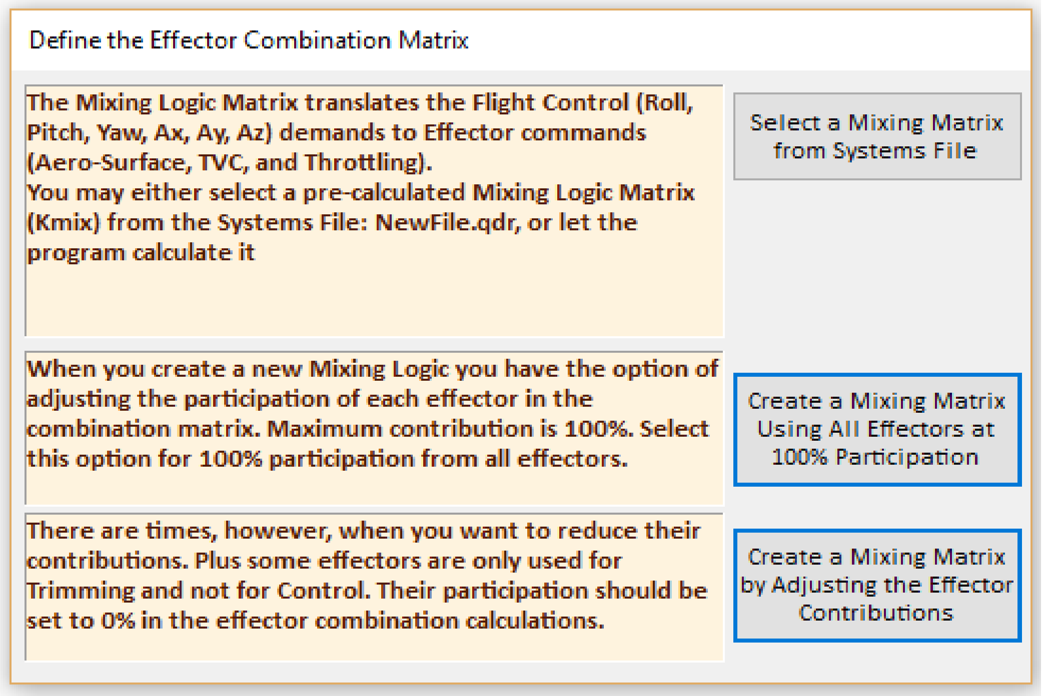
The control effectors must be capable of varying the vehicle angles of attack and sideslip (typically 3-5 deg) from their trim values.

Enter the worst expected alpha and beta dispersions in (deg), and also delta-velocity in (ft/sec) from trim that must be controlled by the effectors, and click OK.

Maximum Alpha (deg): 4.0000

Maximum Beta (deg): 4.0000

Maximum Change in Velocity due to Wind in (feet/sec): 50.000

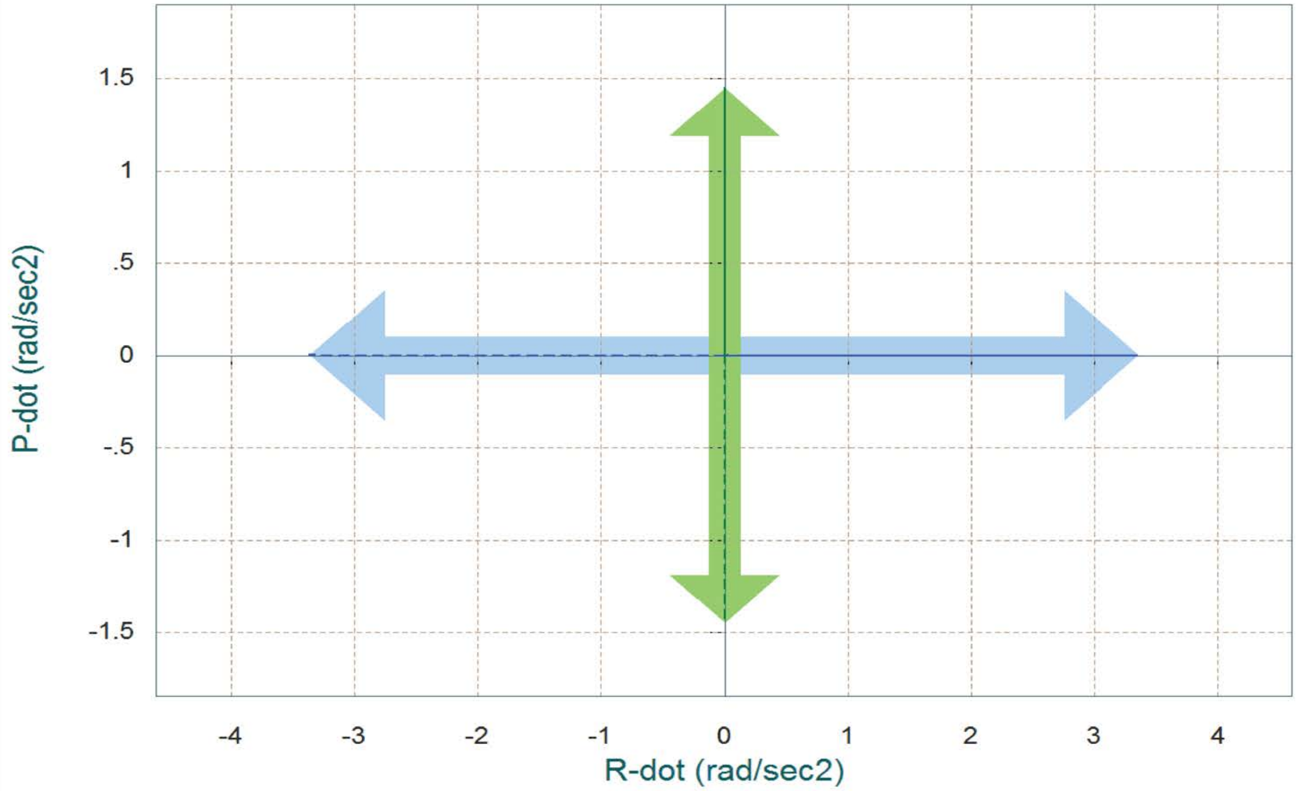

Define the Effector Combination Matrix

The Mixing Logic Matrix translates the Flight Control (Roll, Pitch, Yaw, Ax, Ay, Az) demands to Effector commands (Aero-Surface, TVC, and Throttling).
You may either select a pre-calculated Mixing Logic Matrix (Kmix) from the Systems File: NewFile.qdr, or let the program calculate it

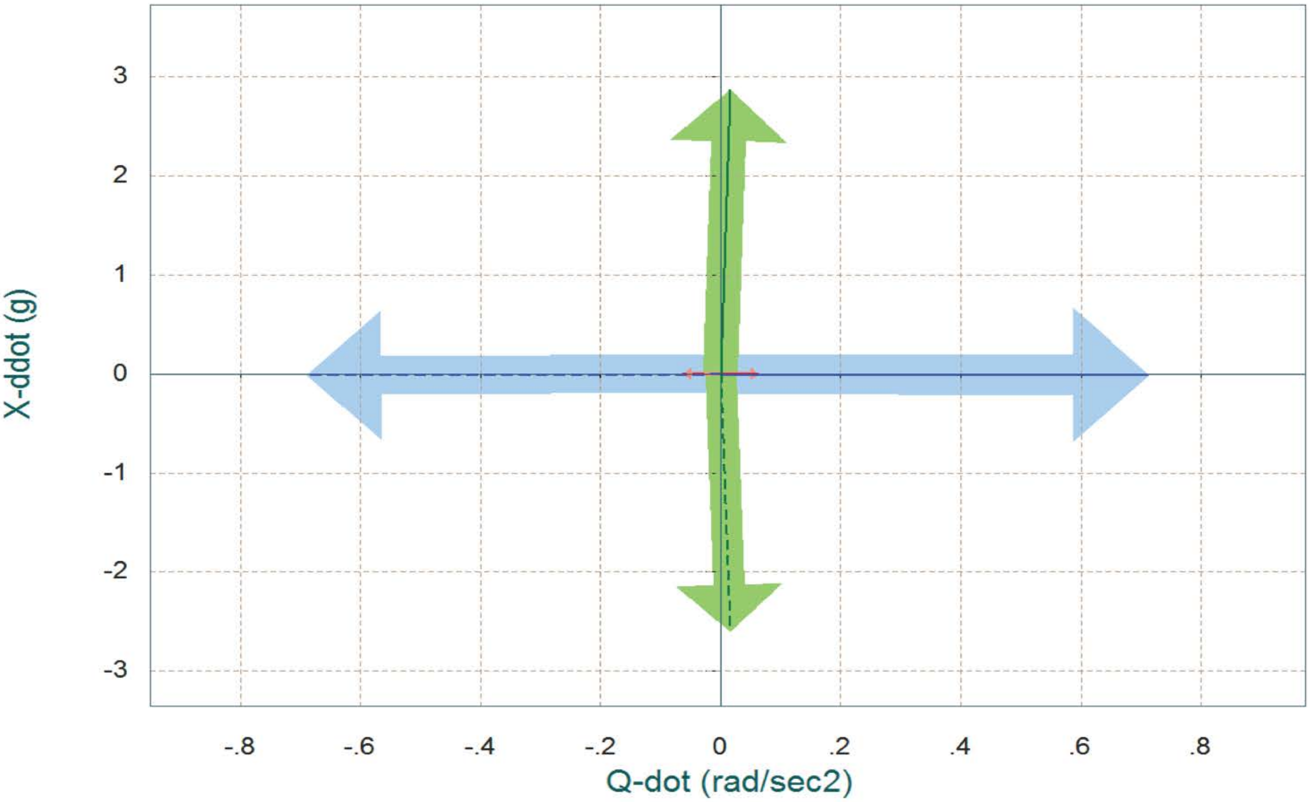
When you create a new Mixing Logic you have the option of adjusting the participation of each effector in the combination matrix. Maximum contribution is 100%. Select this option for 100% participation from all effectors.

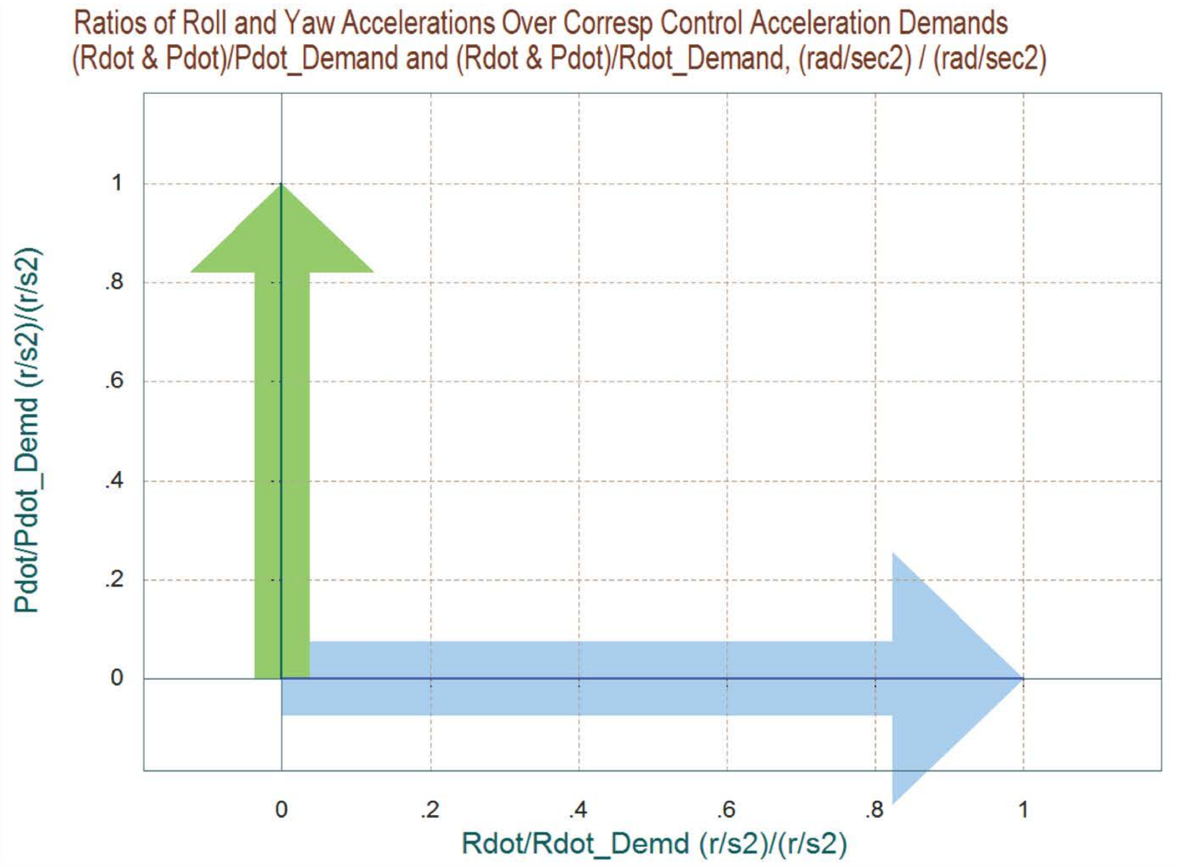
There are times, however, when you want to reduce their contributions. Plus some effectors are only used for Trimming and not for Control. Their participation should be set to 0% in the effector combination calculations.

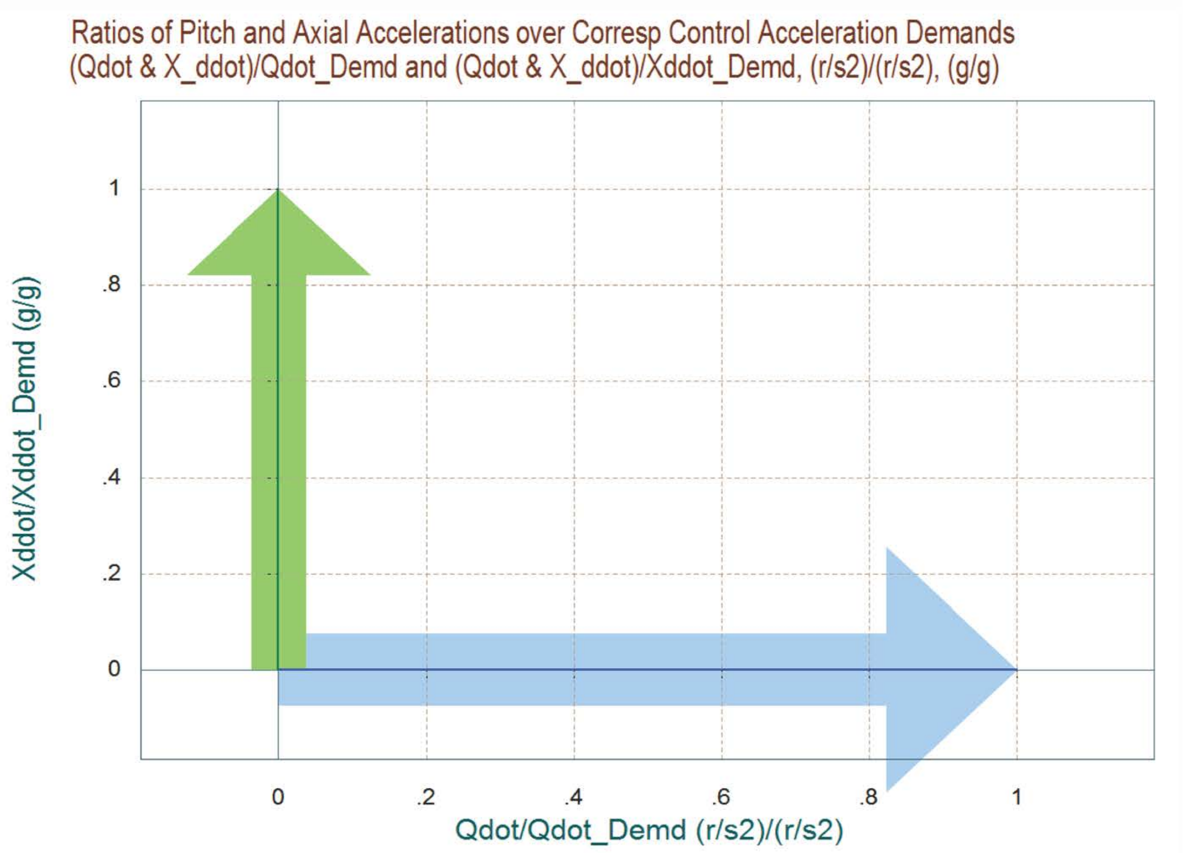
Comparison Between Maximum Control Accelerations and Max Accels due to Beta (red)
Roll & Yaw Accelerations due to Maximum Roll/ Yaw Control and due to Max Beta

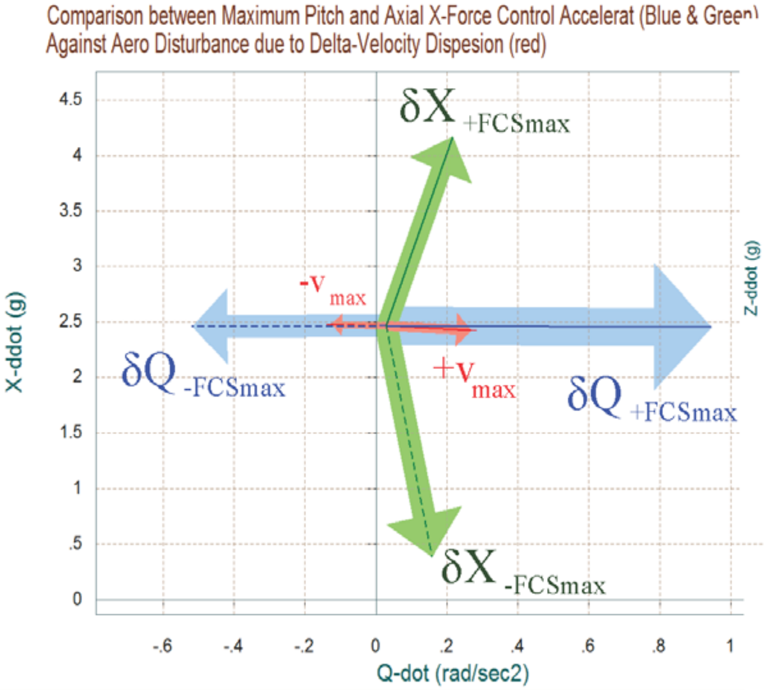


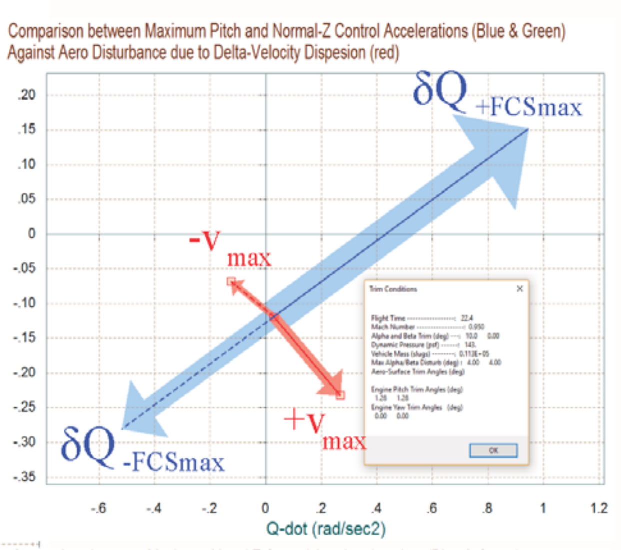
Comparison Between Maximum Control Accelerations and Max Accels due to Alpha (red)
Pitch Accelerat and Axial X-Accelerations due to Max Control and due to Max Alpha




Ratios of Roll and Yaw Accelerations Over Corresp Control Acceleration Demands
(Rdot & Pdot)/Pdot_Demand and (Rdot & Pdot)/Rdot_Demand, (rad/sec2) / (rad/sec2)
Pdot/Pdot_Demd (r/s2)/(r/s2)
1
.8
.6
.4
.2
0
0
.2
.4
.6
.8
1
Rdot/Rdot_Demd (r/s2)/(r/s2)


Ratios of Pitch and Axial Accelerations over Corresp Control Acceleration Demands
(Qdot & X_ddot)/Qdot_Demd and (Qdot & X_ddot)/Xddot_Demd, (r/s2)/(r/s2), (g/g)
Xddot/Xddot_Demd (g/g)
1
.8
.6
.4
.2
0
0
.2
.4
.6
.8
1
Qdot/Qdot_Demd (r/s2)/(r/s2)

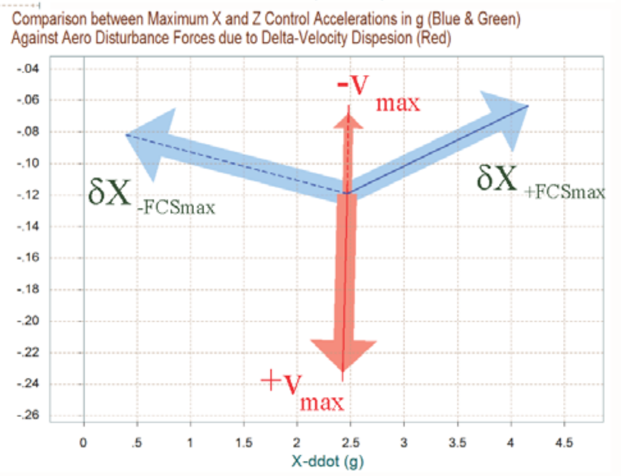




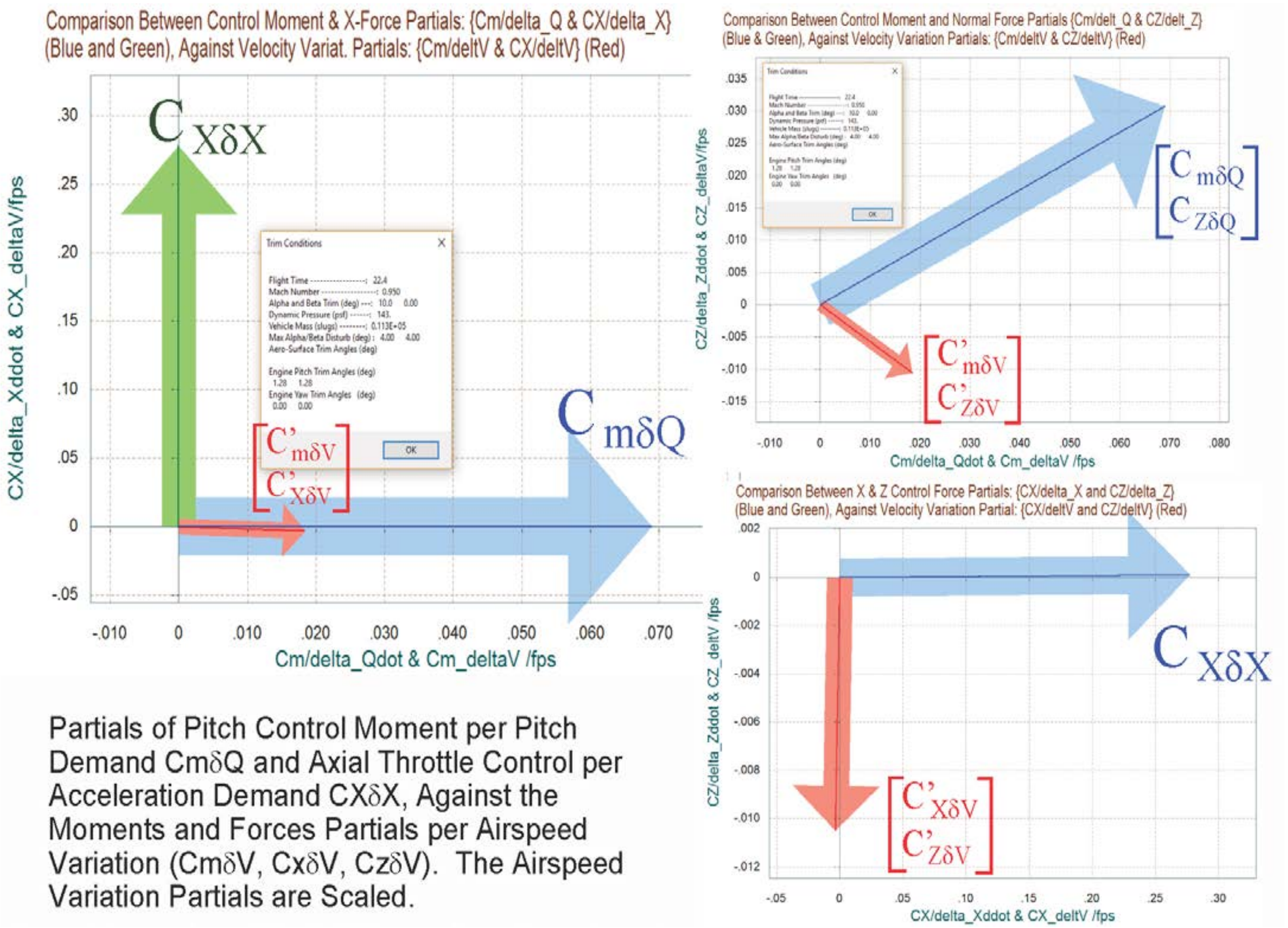
Flight Time 22.4
Mach Number 0.950
Alpha and Beta Trim (deg) ... 10.0 0.00
Dynamic Pressure (psf) ...... 143.
Vehicle Mass (slugs) .... 0.112E+05
Max Alpha/Beta Disturb (deg) : 4.00 4.00
Aero-Surface Trim Angles (deg)

Engine Pitch Trim Angles (deg)
1.28 1.28
Engine Yaw Trim Angles (deg)
0.00 0.00

Accelerations in (g) and in (rad/sec^2) produced by Maximizing the Controls from Trim Conditions. The Blue Vectors show the maximum: Q-dot, X, and Z accelerations produced by maximizing the pitch control $\delta Q_{\pm FCS\_Max}$ from Trim. The Green Vectors show the maximum accelerations produced by throttling up and down from nominal thrust $\delta X_{\pm FCS\_Max}$. The red vectors show the acceleration variations generated by the airspeed variations $\pm V_{max}$ from V_0.



Similar to the α_{max} and β_{max} dispersions we can also compare the maximum control capability of the effector system against disturbances generated by airspeed variations relative to trim speed V_0. In the figure above Flixan calculates the maximum accelerations generated by the maximum airspeed variation $\pm v_{max}$ and compares it against the maximum pitch, axial, and normal accelerations generated by the controls. This launch vehicle has controls in two directions: a pitch TVC control and axial acceleration control by varying the engine thrusts. In the flight condition shown it is trimmed at 2.5 g acceleration due to the main engine thrusting and it can change its acceleration from $\delta X_{-FCSMax}$=0.4 g to $\delta X_{+FCSMax}$=4.2 g by throttling the engines (green vectors). It can also change its pitch acceleration through gimbaling the TVC engines from $\delta Q_{-FCSMax}$ to $\delta Q_{+FCSMax}$ (blue vectors). The above vector diagram compares the control accelerations against the accelerations generated by the wind dispersion (red vectors). This vehicle is statically unstable and it is flying with a positive α_0. An increase in the airspeed due to wind $+v_{max}$, therefore, produces a positive pitching moment, a negative z-acceleration (upwards), and a negative x-acceleration (more drag). In this situation, the pitch and axial controls are more powerful than those generated from the $\pm v_{max}$ dispersions due to the wind and they can, therefore, be counteracted. Note that, there is a considerable wind dispersion effect in the z-acceleration, but we don't care about the z-direction since we are not controlling this axis.



Flight Time ------------------: 22.4
Mach Number ------------------: 0.950
Alpha and Beta Trim (deg) ---: 10.0 0.00
Dynamic Pressure (psf) -------: 143.
Vehicle Mass (slugs) ---------: 0.113E+05
Max Alpha/Beta Disturb (deg) : 4.00 4.00
Aero-Surface Trim Angles (deg)

Engine Pitch Trim Angles (deg)
1.28 1.28
Engine Yaw Trim Angles (deg)
0.00 0.00

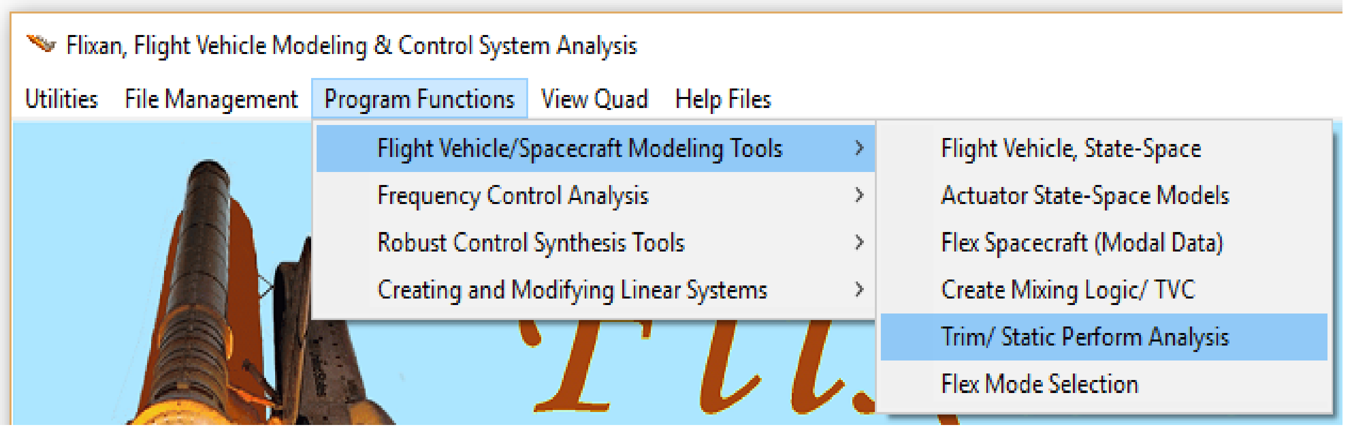
Partials of Pitch Control Moment per Pitch Demand CmδQ and Axial Throttle Control per Acceleration Demand CXδX, Against the Moments and Forces Partials per Airspeed Variation (CmδV, CxδV, CzδV). The Airspeed Variation Partials are Scaled.

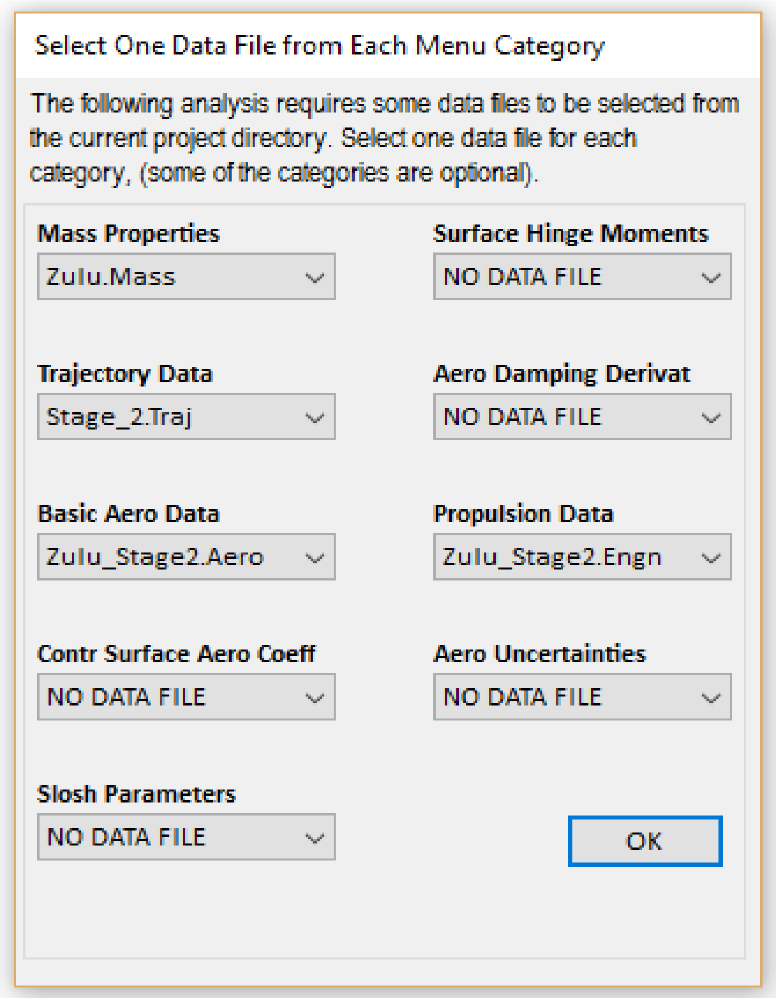
The vector diagrams above compare the pitch and axial control partials (blue and green vectors) of a launch vehicle, that is statically unstable and at a positive angle of attack α_0, against the moment and force partials generated due to airspeed variation (red vectors). The red vectors are scaled as described in Section 8.3. The vector diagram illustrates that a pitch control demand variation δQ_{FCS} (blue vector) generates mainly a pitching moment $C_{m\delta Q}$, as expected, and it does not couple in the axial direction. It generates also a positive z-force $C_{Z\delta Q}$. That's because the engines pivot negative (up) to generate the pitch moment and they also generate the +z force. Similarly, the axial control demand variation δX_{FCS} (green vector) generates an axial force $C_{X\delta X}$ by throttling the engines and it does not couple in the pitch direction, as expected. The scaled red vector is the moment and force partials per airspeed variation. An increase in airspeed causes: a positive pitching moment, a negative z-force (up), and a reduction in x-force (more drag). The control partials are sufficiently greater in magnitude than the velocity partials, as they should be, in order to compensate against wind-speed variations

1.2 Static Analysis during Second Stage

After the two side boosters separate at t=91.5 seconds, the middle second stage booster ignites and the middle engine is used to control the vehicle in pitch and yaw. The files for the second stage static analysis are in folder "*Examples**Twin Booster Rocket**1-Static Analysis**Stage-2*". The mass properties are the same in file "*Zulu.Mass*". The aerodynamic coefficients are in file "*Zulu_Stage2.Aero*", which contains aero coefficients at 16 Mach numbers, 3 betas, and 15 values of angle of attack. The file also includes the reference lengths and the location of the moment reference center. In addition to the TVC the vehicle uses two reaction control thrusters that fire in the ±Z direction for roll control. They can provide a maximum thrust of ±800 pounds. The TVC engines and RCS jets data are located in file "*Zulu_Stage2.Engn*". It includes thrusts, locations, 5^0 maximum deflections for the TVC, and ±100% throttling for the two jets. The TVC engine is also allowed to throttle ±99% relative to an average 380,000 (lb) thrust during trimming in order to match the acceleration in the preliminary second stage trajectory file: "*Stage_2.Traj*" that was created from a point-mass trajectory optimization program.

By trimming along the second stage trajectory we calculate the engine deflections necessary to balance the 3 moments and the axial acceleration. When there is no sideslip or Y_{CG} offset, the yaw disturbances are zero and the engine deflection is mainly in pitch and the main engine thrust is varied by the Trim program in order to match the acceleration. When there is a shift in the Y_{CG}, however, the engine also rotates in yaw to counteract the yawing moment due to the Y_{CG} shift, and the reaction control thrusters also fire to balance the rolling moment due to beta. Let us start the Trim program, select the data files as shown, and from the Trim main menu select to plot the trajectory parameters.



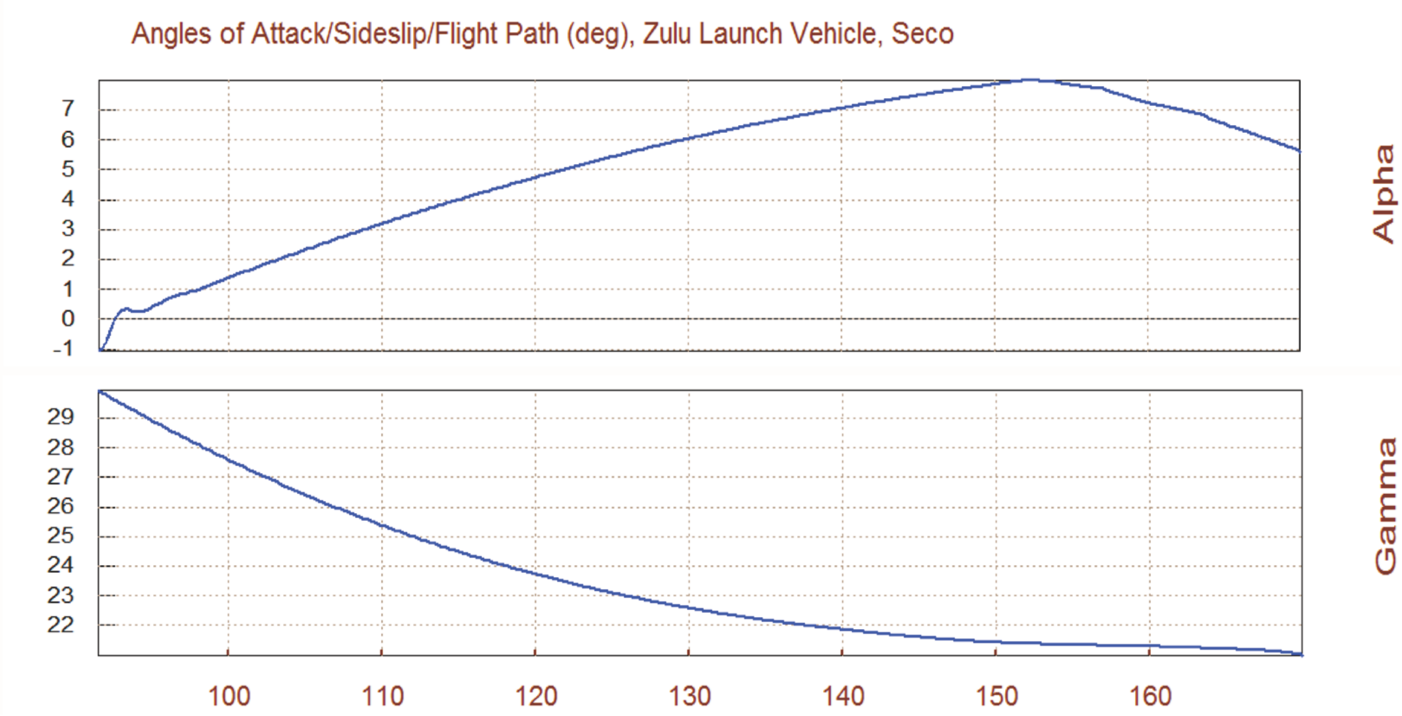


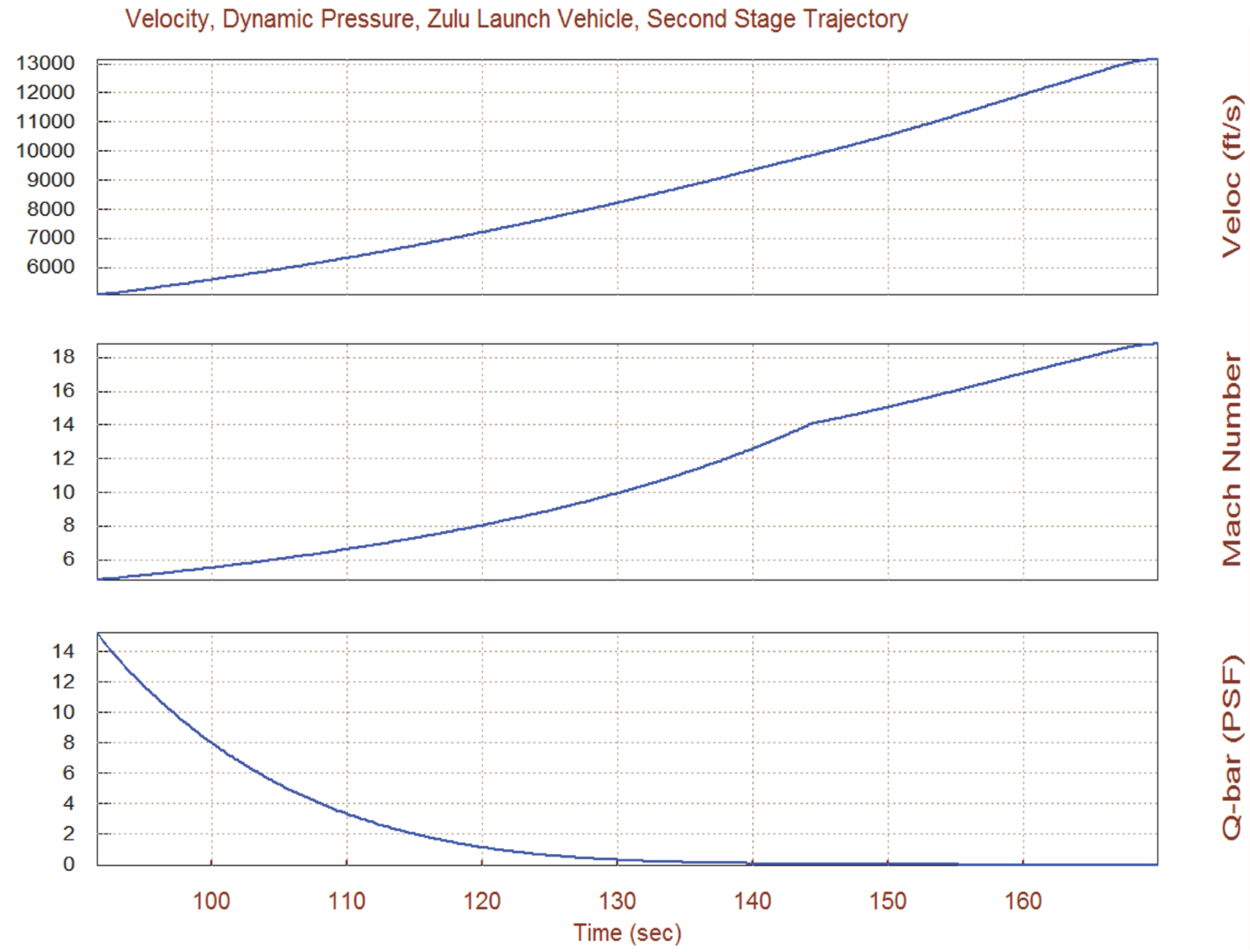
Select One Data File from Each Menu Category

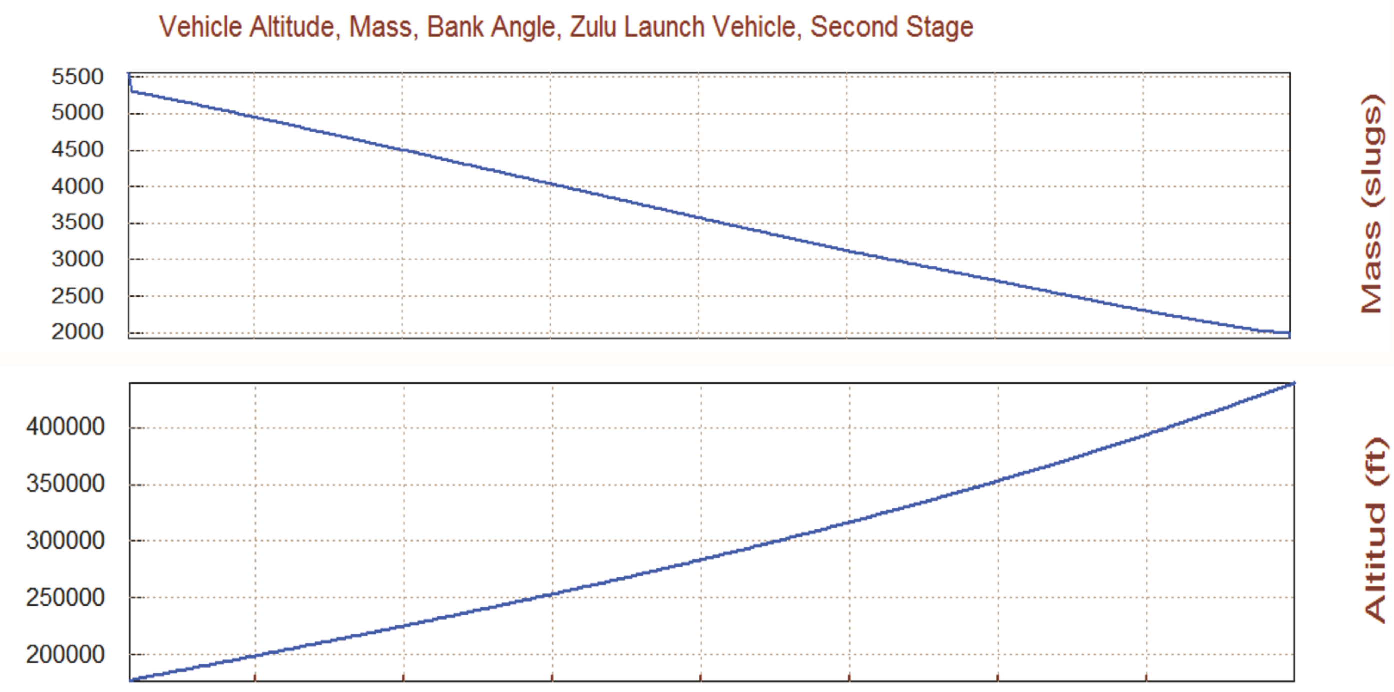
The following analysis requires some data files to be selected from the current project directory. Select one data file for each category, (some of the categories are optional).

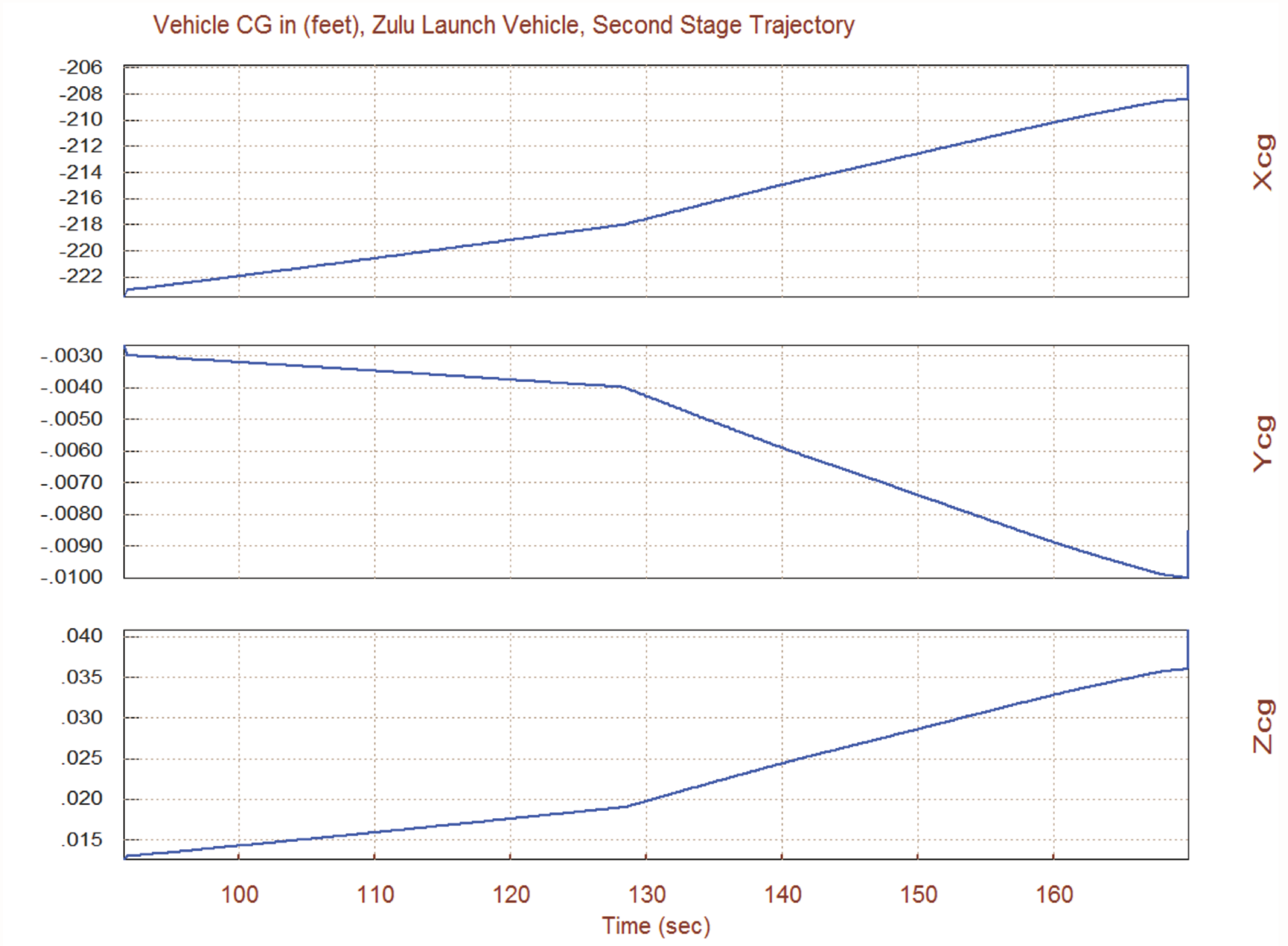
Mass Properties: Zulu.Mass
Surface Hinge Moments: NO DATA FILE
Trajectory Data: Stage_2.Traj
Aero Damping Derivat: NO DATA FILE
Basic Aero Data: Zulu_Stage2.Aero
Propulsion Data: Zulu_Stage2.Engn
Contr Surface Aero Coeff: NO DATA FILE
Aero Uncertainties: NO DATA FILE
Slosh Parameters: NO DATA FILE

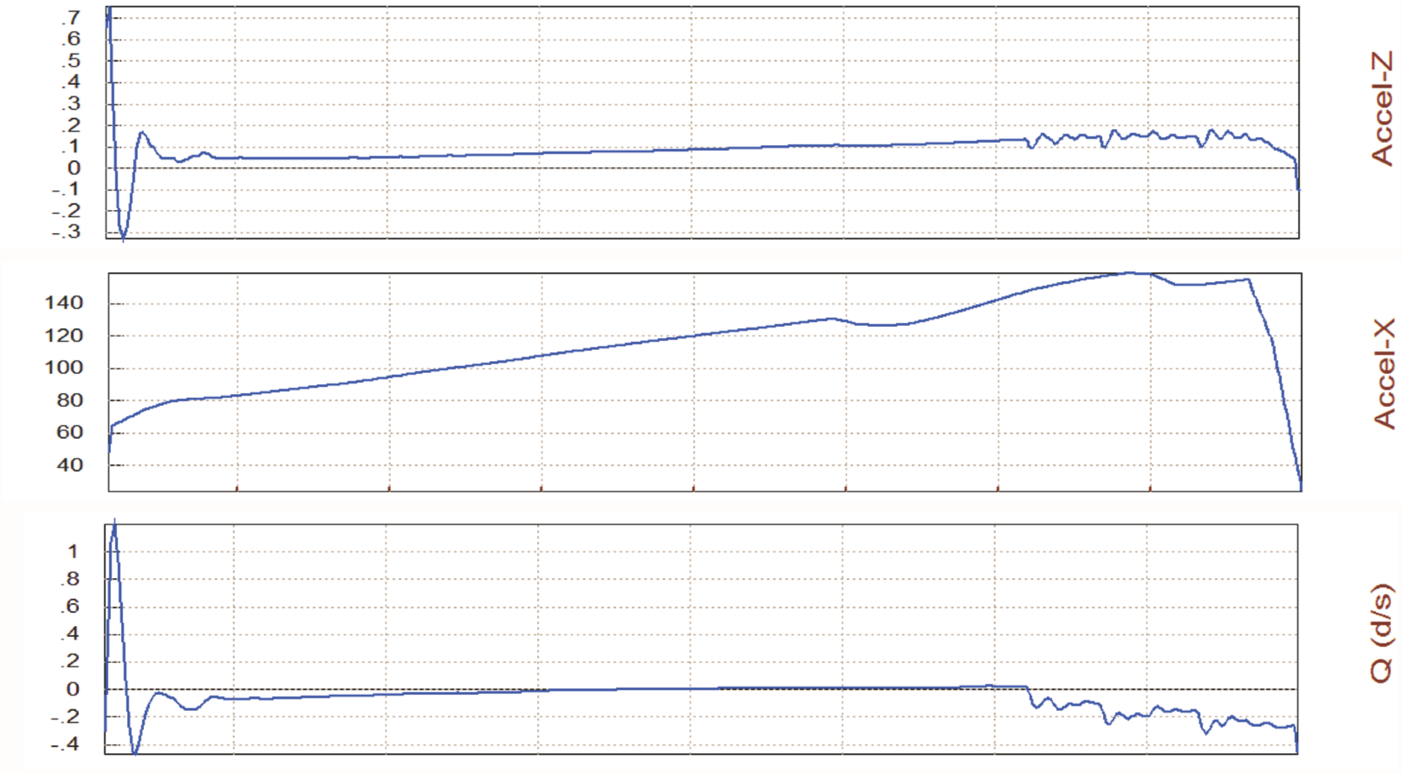
The trajectory shows the vehicle mass is steadily decreasing after staging while the altitude increases to 450,000 (feet) and the speed to 13,000 (feet/sec) at the end of the second stage, where the middle second stage booster separates and the third stage ignites. The X_{CG} is moving forward and the Z_{CG} is shifting in the +Z direction as the fuel is depleting. The dynamic pressure is much lower during second stage. The flight path angle gamma is steadily decreasing but it is still positive as the vehicle continues to ascend. The angle of attack is mostly positive. It is small when the dynamic pressure is high in order to reduce drag and Qalpha loading. It increases to 8^o when the dynamic pressure is very small. The thrust exceeds 400,000 pounds and the axial acceleration reaches 5-gs near the end of the second stage as the vehicle becomes lighter. The lift force is due to the vertical thrust component and the negative drag force represents the forward/ horizontal component of thrust.

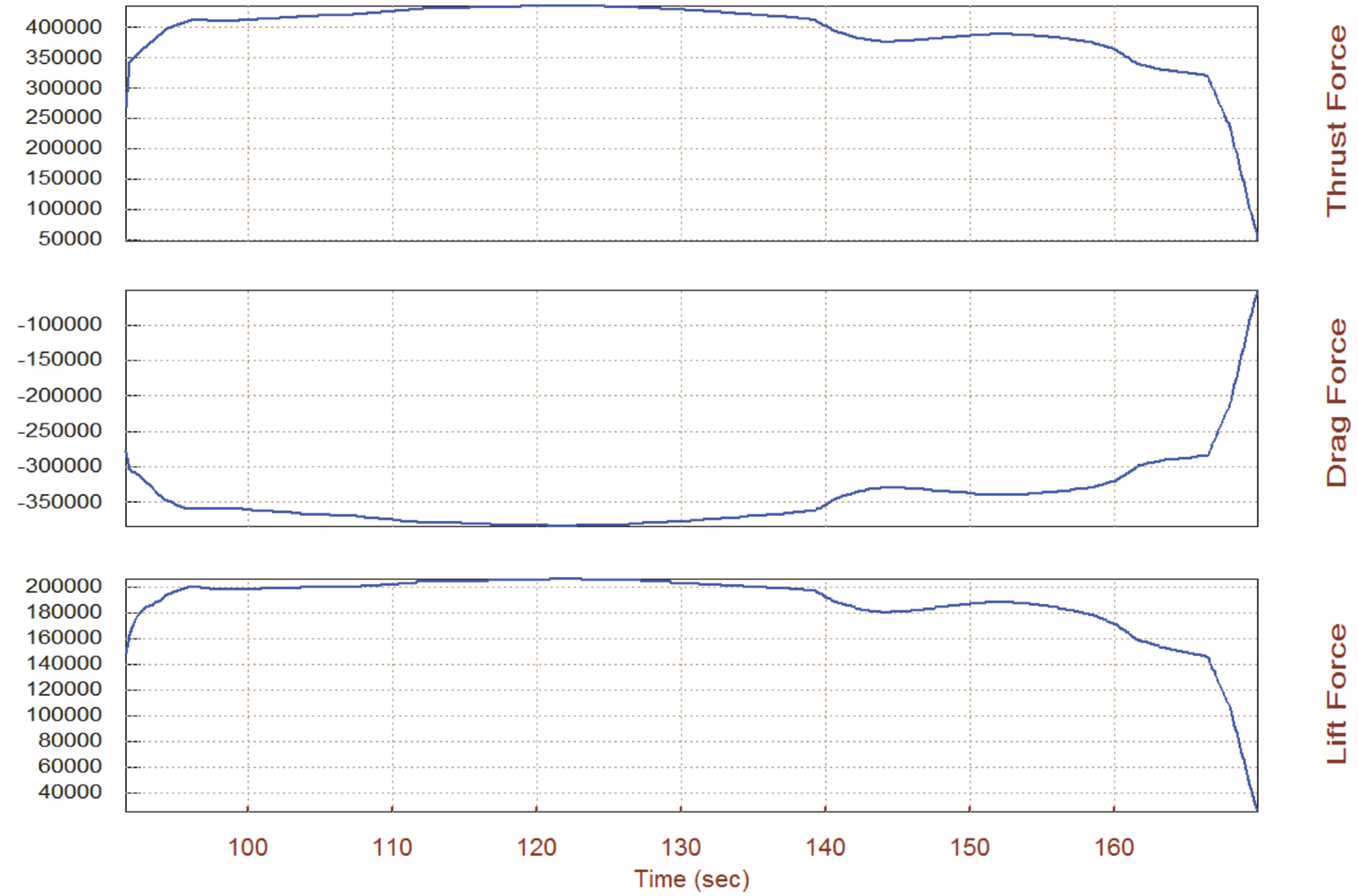

Angles of Attack/Sideslip/Flight Path (deg), Zulu Launch Vehicle, Seco
7
6
5
4
3
2
1
0
-1
Alpha
29
28
27
26
25
24
23
22
Gamma
100
110
120
130
140
150
160


Velocity, Dynamic Pressure, Zulu Launch Vehicle, Second Stage Trajectory
13000
12000
11000
10000
9000
8000
7000
6000
Veloc (ft/s)
18
16
14
12
10
8
6
Mach Number
14
12
10
8
6
4
2
0
Q-bar (PSF)
100
110
120
130
140
150
160
Time (sec)


Vehicle Altitude, Mass, Bank Angle, Zulu Launch Vehicle, Second Stage
5500
5000
4500
4000
3500
3000
2500
2000
Mass (slugs)
400000
350000
300000
250000
200000
Altitud (ft)

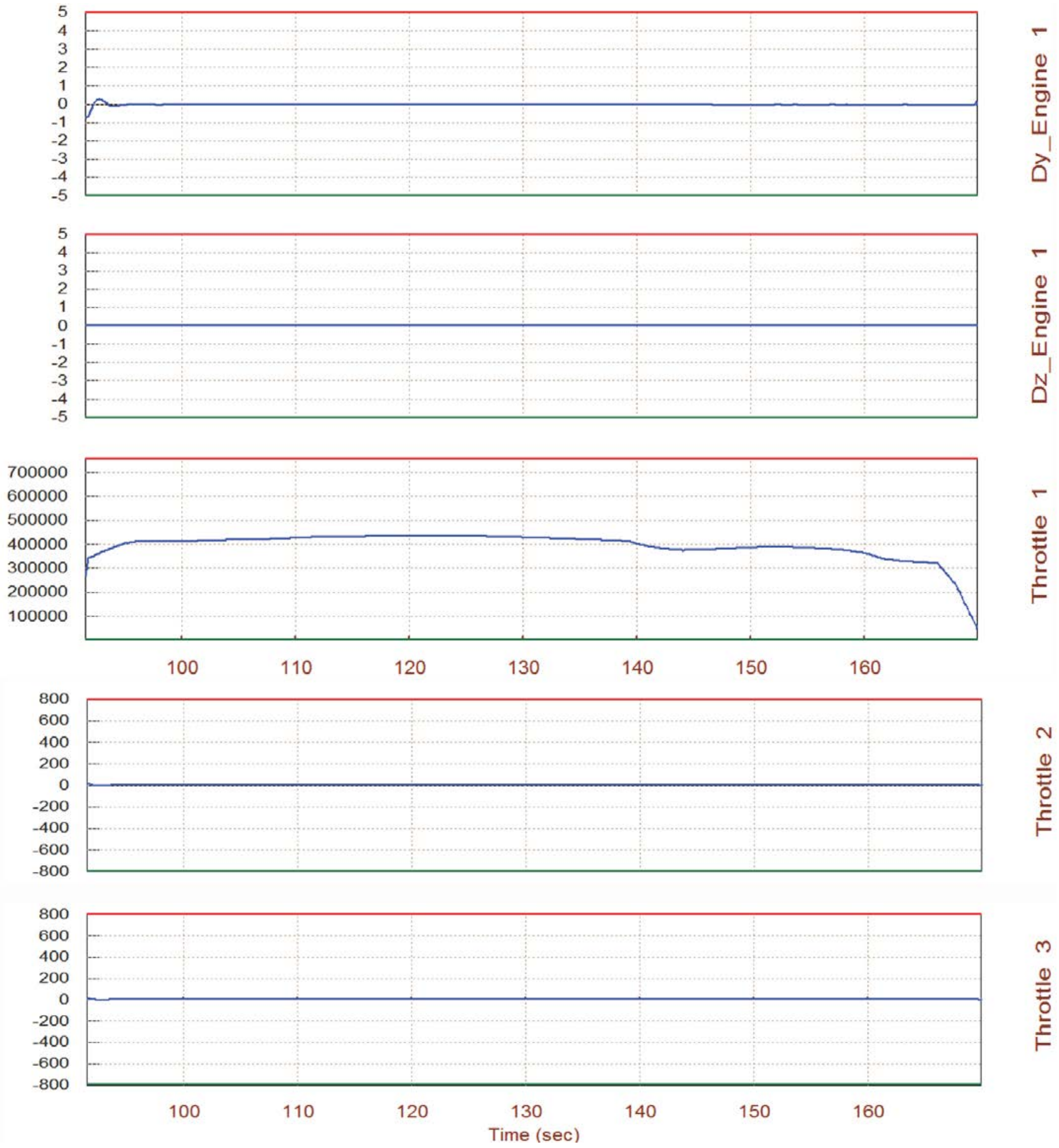

Vehicle CG in (feet), Zulu Launch Vehicle, Second Stage Trajectory
-206
-208
-210
-212
-214
-216
-218
-220
-222
Xcg
-.0030
-.0040
-.0050
-.0060
-.0070
-.0080
-.0090
-.0100
Ycg
.040
.035
.030
.025
.020
.015
Zcg
100
110
120
130
140
150
160
Time (sec)


Sensed Acceleration in (ft/sec^2), Zulu Launch Vehicle, Second Stage Trajec

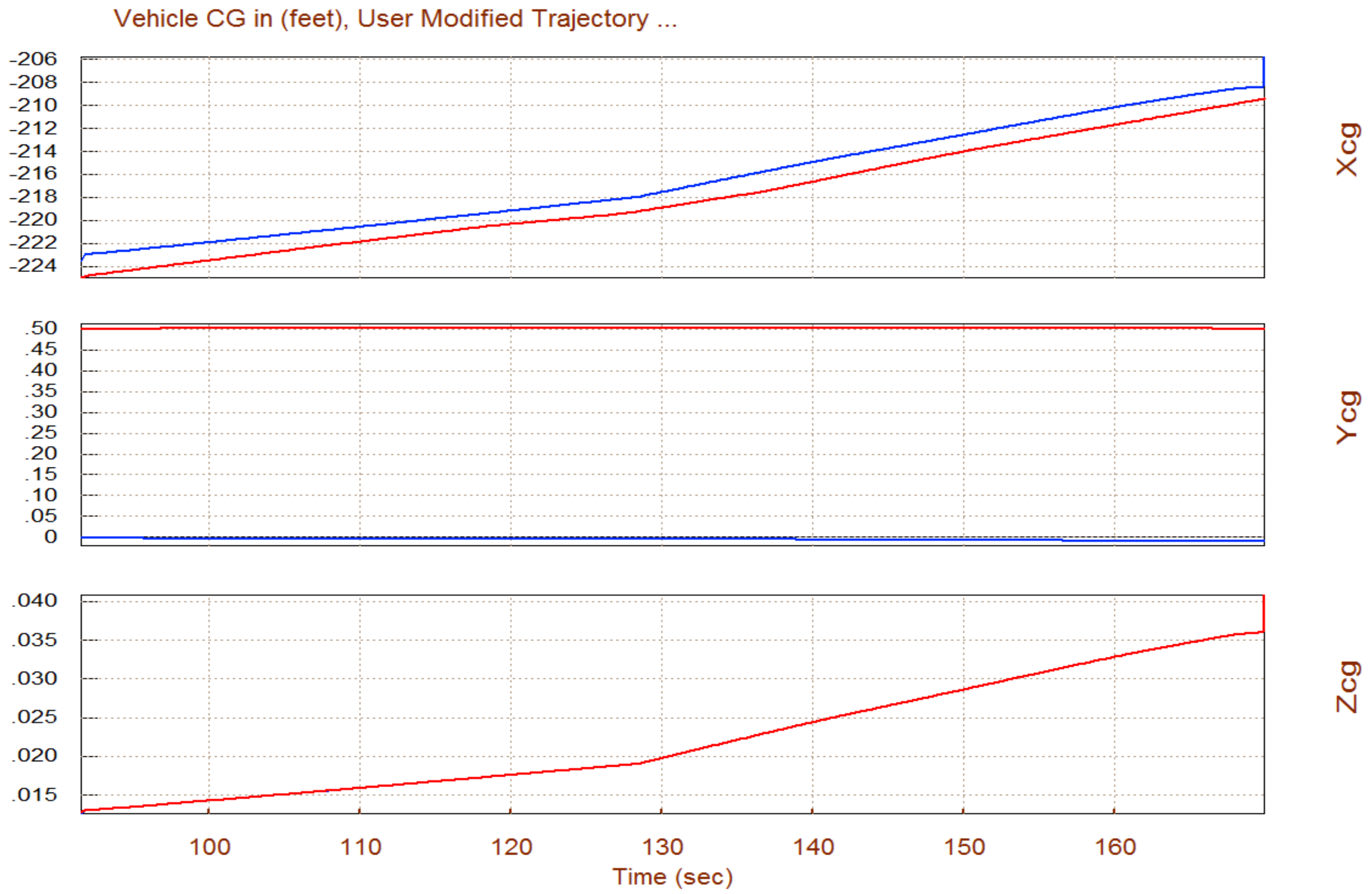

Aero Lift/Drag Forces, Eng. Thrust in (lb), Zulu Launch Vehicle, Second Stag

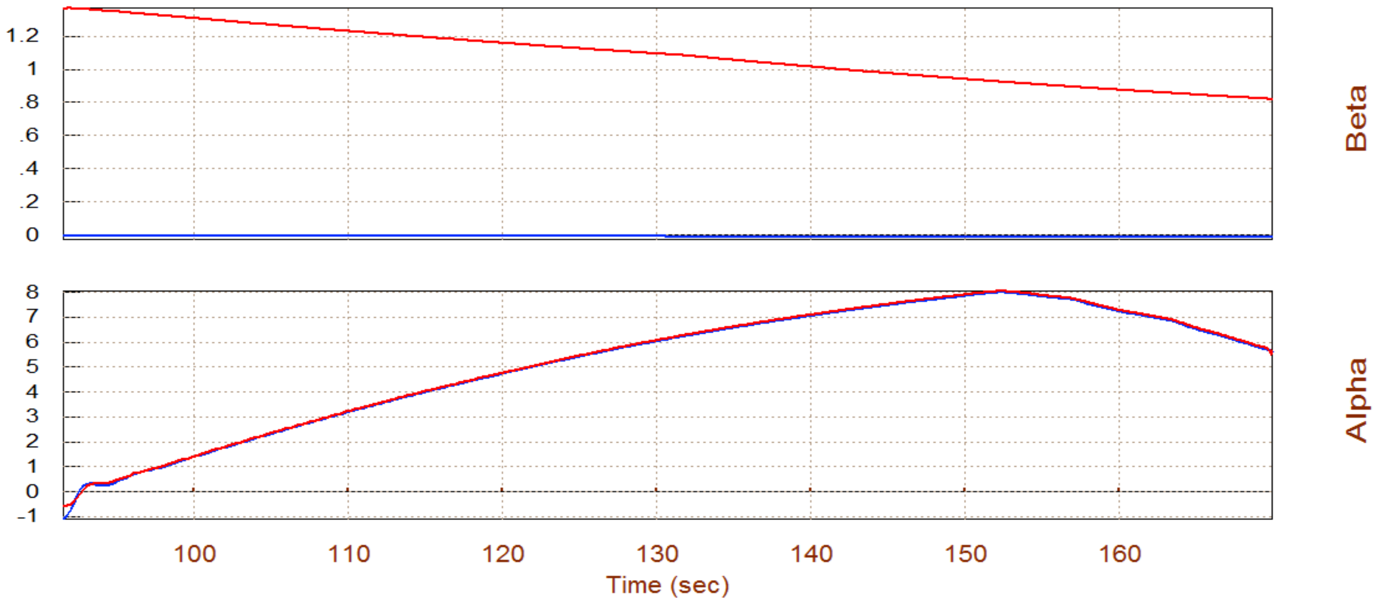
Trimming

The next figure shows the effector trim results along the nominal 2nd stage trajectory. There is a small pitch engine deflection and thrust variation to balance the longitudinal accelerations. The yaw TVC deflection is zero because the yaw disturbance is almost zero, and therefore, there is no thruster firing either since there is no excitation in roll, as well.

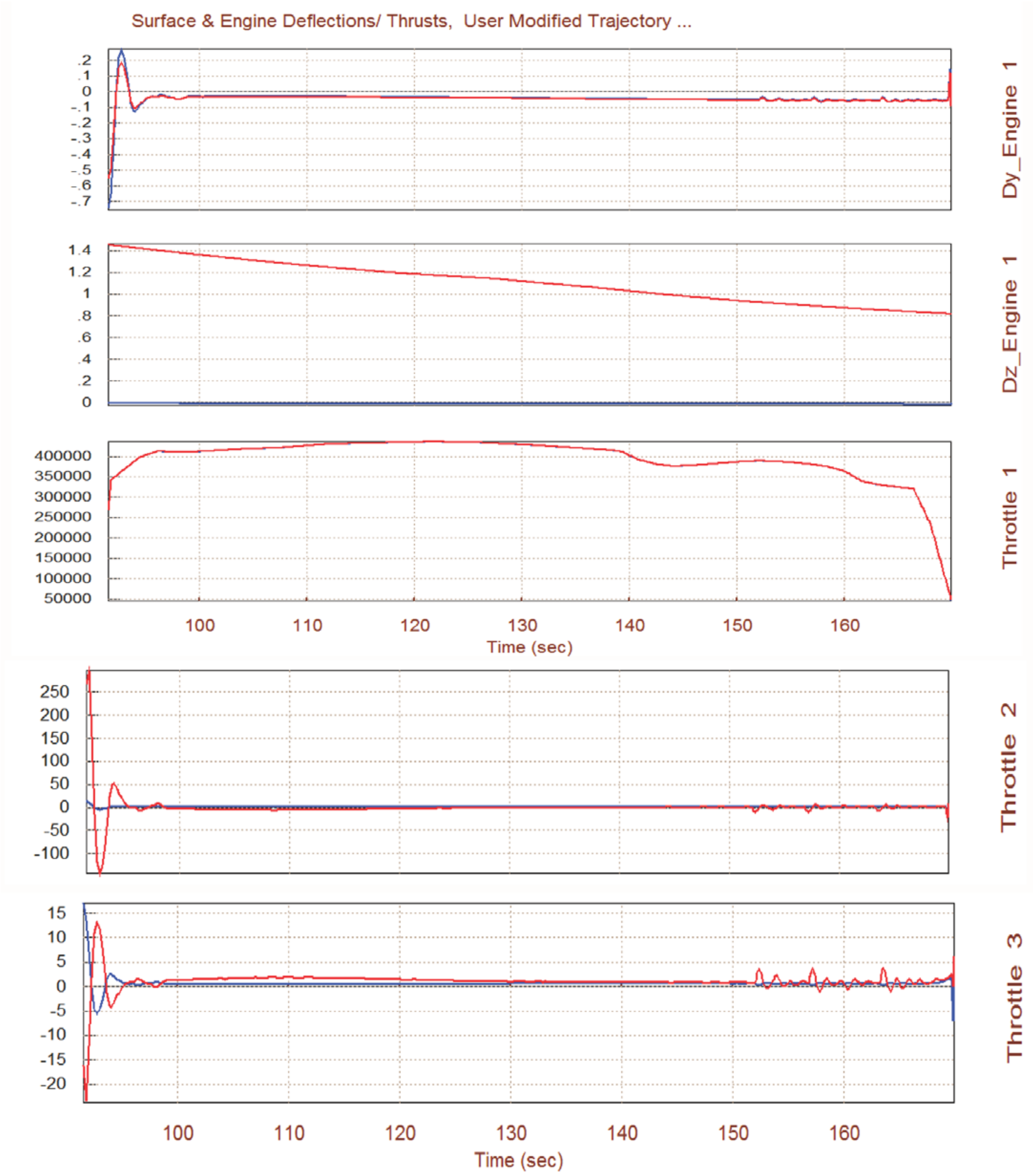


The trim results change, however, when we modify the trajectory to include CG biases. We modified the trajectory file to “Stage_2c.Traj” which includes dispersions in the X_{CG} and Y_{CG}. The X_{CG} is shifted towards the back and the Y_{CG}=0.5 feet to the right, as shown below. We retrim using the modified trajectory and compare the trim results against the original trajectory. The first thing we notice is that the vehicle trims with a positive beta angle in order to balance the moment due to the +Y_{CG} shift.





The effector results also show a significant amount of yaw TVC deflection (red) that is required to balance the yaw moment caused by the Y_{CG} shift. The original trim is the blue curve. There is also a small variation in the pitch TVC deflection due to the X_{CG} shift. The two reaction control thrusters are also firing differentially to balance the rolling moment caused by the lateral asymmetry.



Static Performance Analysis during Second Stage

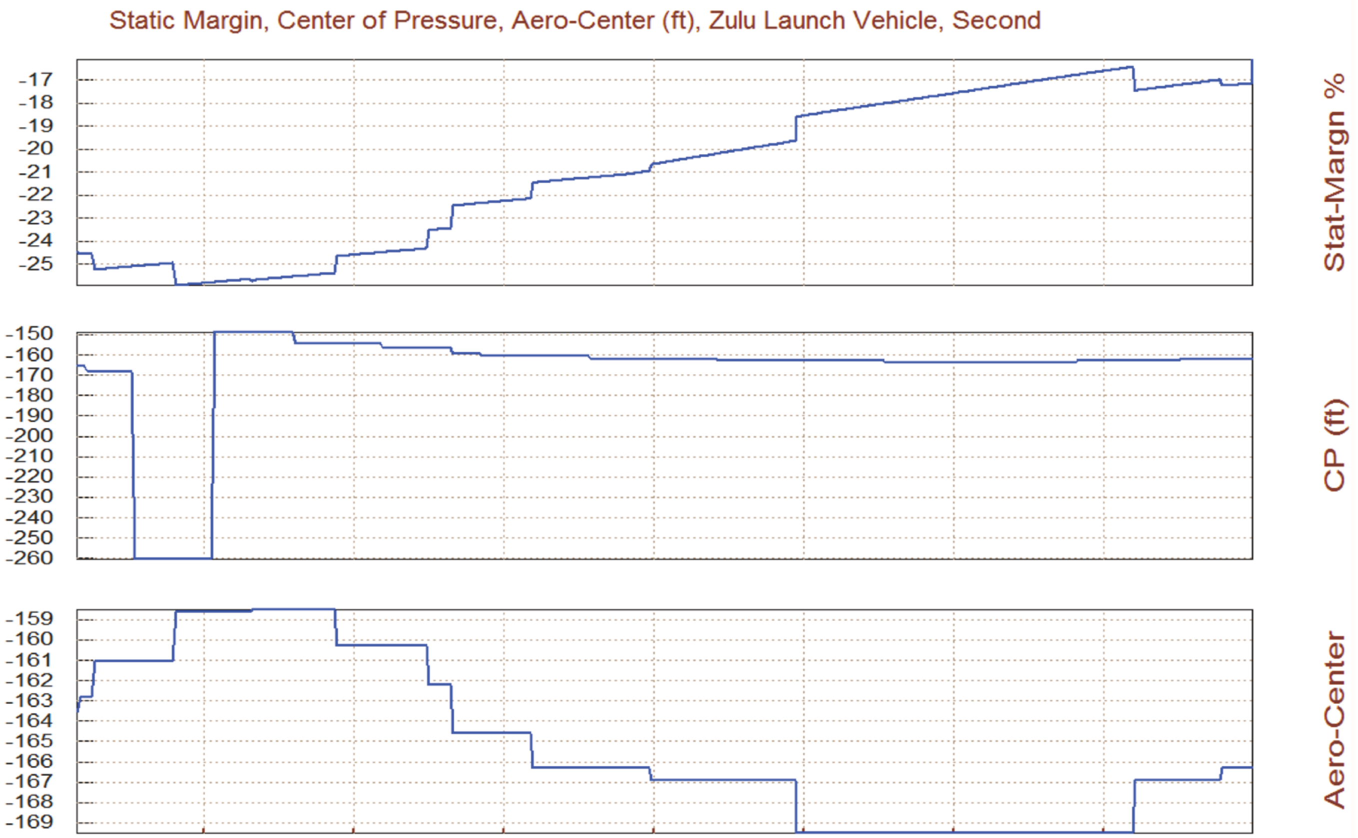
The following static performance parameters plots along second stage are obtained using Option-6. The performance parameters calculation requires a mixing logic matrix that will combine the five effectors (2 TVC, 1 engine throttle, and 2 RCS thrusters) in order to achieve the four demanded accelerations (roll, pitch, yaw, and axial). A precalculated mixing logic matrix Kmix3 is already saved in file “Mixing_Matrix.Qdr” and it must be selected prior to the analysis.

The T2-inverse parameters show that the vehicle is statically unstable in both pitch and yaw, but the time-to-double amplitude is sufficiently long, about 2 seconds. The $Cn\beta$-dynamic parameter is negative, which is also a sign of lateral instability. The LCDP defines dynamic roll controllability and it is excellent (close to one). The Qalpha parameter is small during second stage. The control authority in roll, pitch, and yaw is very good because the control effort magnitude against $\pm 5^o$ of angle of attack and sideslip dispersions is much smaller than one in both directions. The pitch and yaw T2-inverse parameter is also shown in contour plots against the entire Mach versus alpha range.

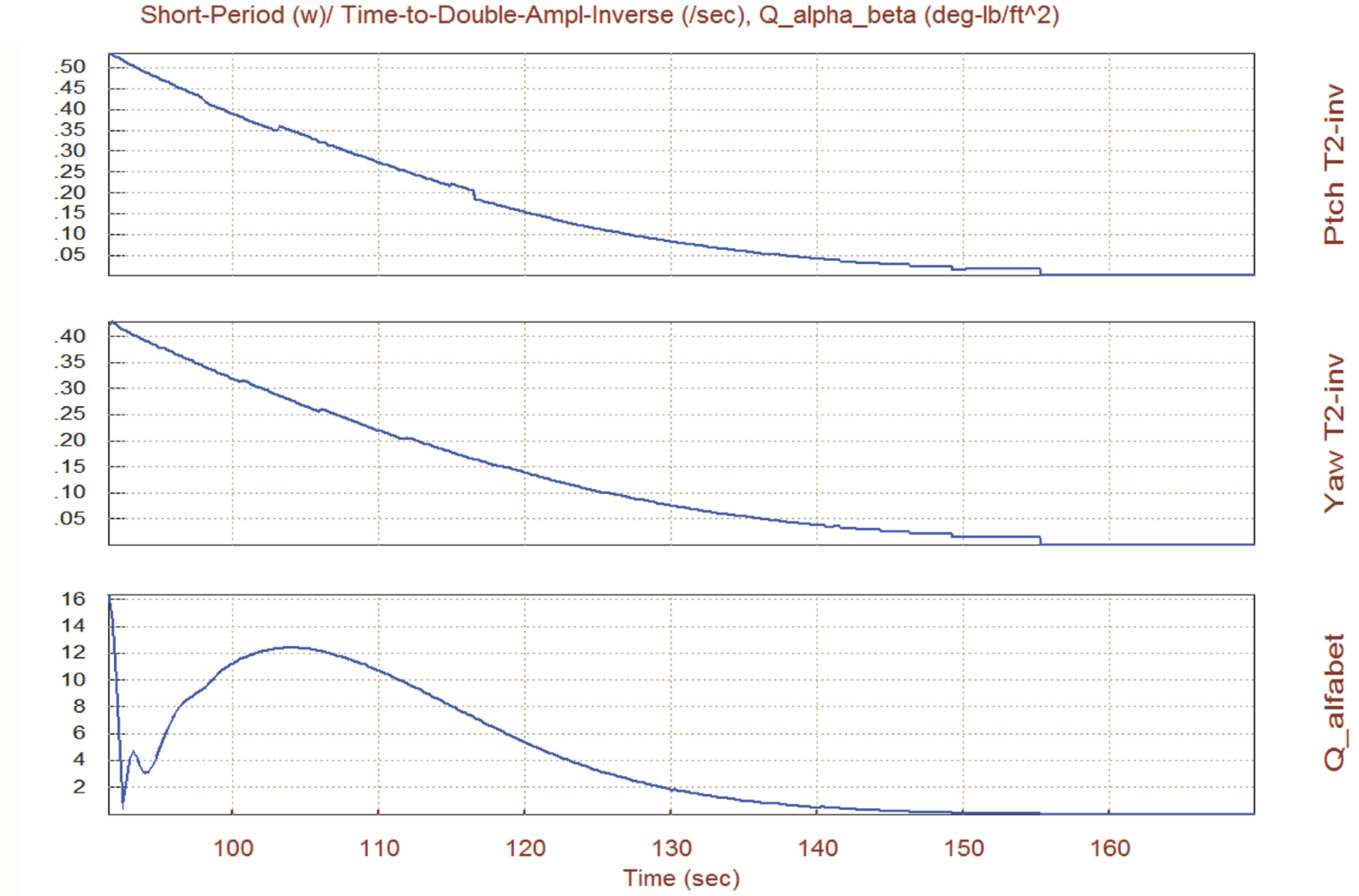
Vector Diagrams

The two vector diagrams show the maximum roll and yaw torques generated by the effectors in the positive and in the negative directions. The roll and yaw demands are combined by the precalculated mixing logic matrix that calculates the TVC deflection and throttling commands, and it must be selected from file “*Mixing_Matrix.Qdr*”. The two vector diagrams correspond to a flight time= 100 sec, and their difference is the Y_{CG} location. The top diagram is obtained using the nominal trajectory “*Stage_2.Traj*” where the Y_{CG} is almost zero. The vehicle trims with zero yaw engine deflection ($\delta z=0$) and the trim moments are zero. The vectors show the ±maximum normalized roll and yaw moments at ±peak roll and yaw demands.

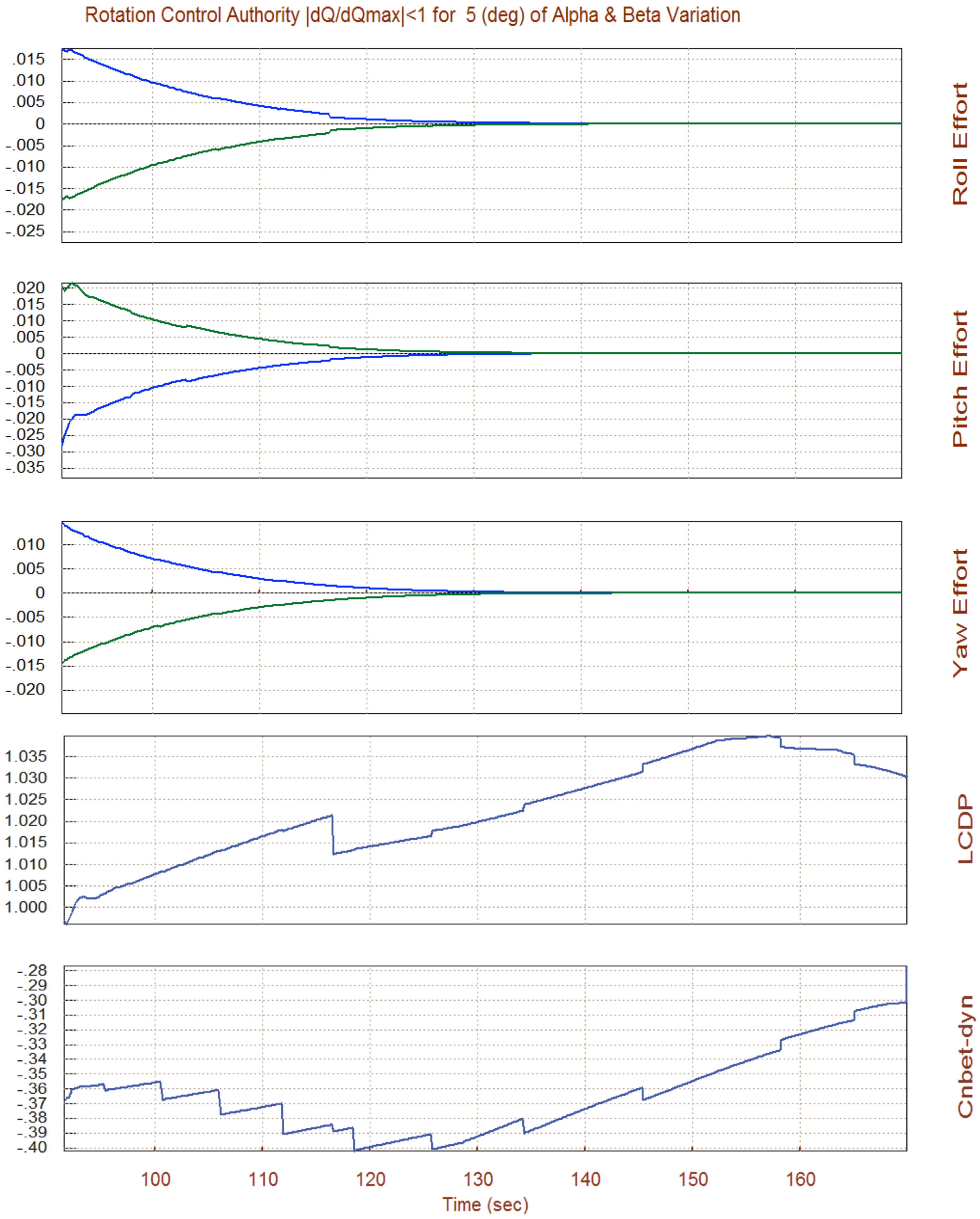
The bottom diagram is obtained from the trajectory “*Stage_2b.Traj*” that has an offset in the Y_{CG} equal to 0.5 (feet). The trajectory modifications are easily made graphically using the Flixan trajectory modification option from the menu above the plots. They are saved and can be renamed. The Y_{CG} shift causes the TVC engine to rotate in yaw $\delta z=+1.4^o$ in order to balance the yawing moment. This causes the vehicle to trim with a negative yaw control moment. Since the engine is already biased positively it has less space to deflect in the positive yaw direction before it hits the $+5^o$ limit and it has, therefore, less control capability in the negative yaw direction from trim. It has a lot more space to deflect in the negative δz direction before it reaches the -5^o limit and it can provide more control capability from trim in the positive yaw direction. Similar analysis must be repeated at different flight conditions with other types of trajectory modifications in order to ensure that the vehicle is capable to withstand different types of static dispersions. We must also evaluate stability and control authority against disturbances, winds, etc. After the static analysis is complete we are now ready to begin the dynamic analysis.

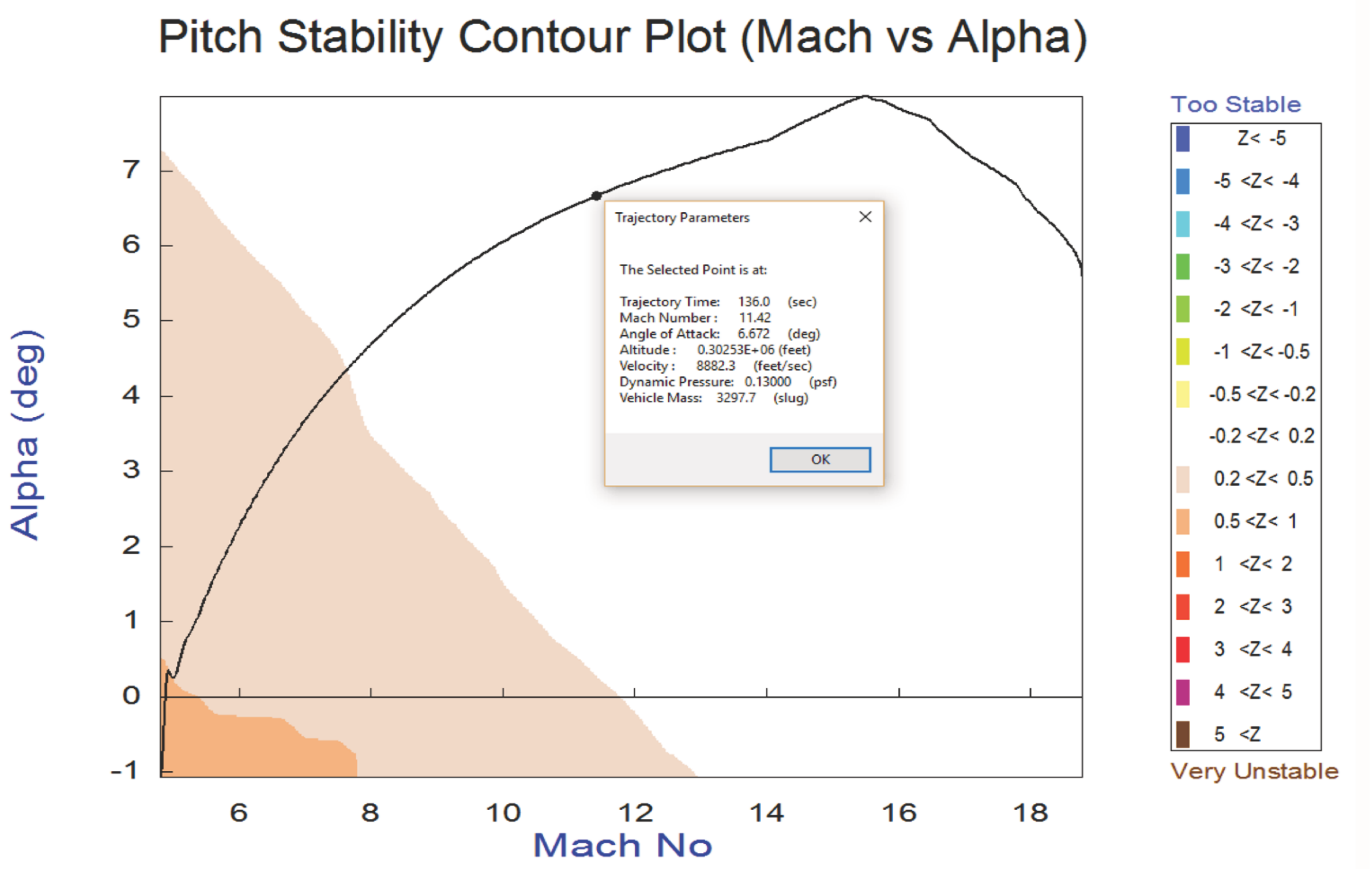


Static Margin, Center of Pressure, Aero-Center (ft), Zulu Launch Vehicle, Second



Short-Period (w)/ Time-to-Double-Ampl-Inverse (/sec), Q_alpha_beta (deg-lb/ft^2)


Rotation Control Authority |dQ/dQmax|<1 for 5 (deg) of Alpha & Beta Variation

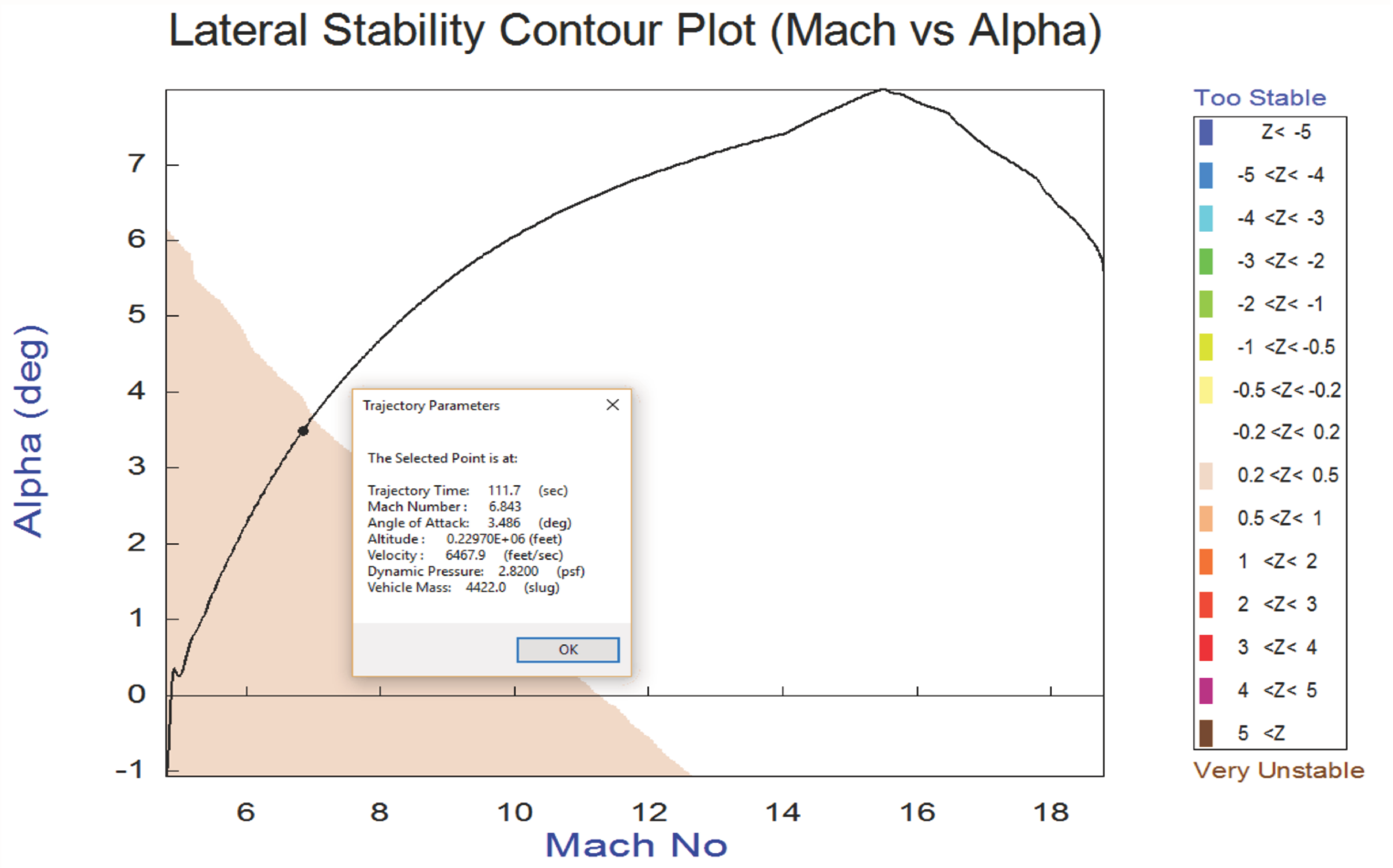


Pitch Stability Contour Plot (Mach vs Alpha)

Trajectory Parameters

The Selected Point is at:

Trajectory Time: 136.0 (sec)
Mach Number : 11.42
Angle of Attack: 6.672 (deg)
Altitude : 0.30253E+06 (feet)
Velocity : 8882.3 (feet/sec)
Dynamic Pressure: 0.13000 (psf)
Vehicle Mass: 3297.7 (slug)

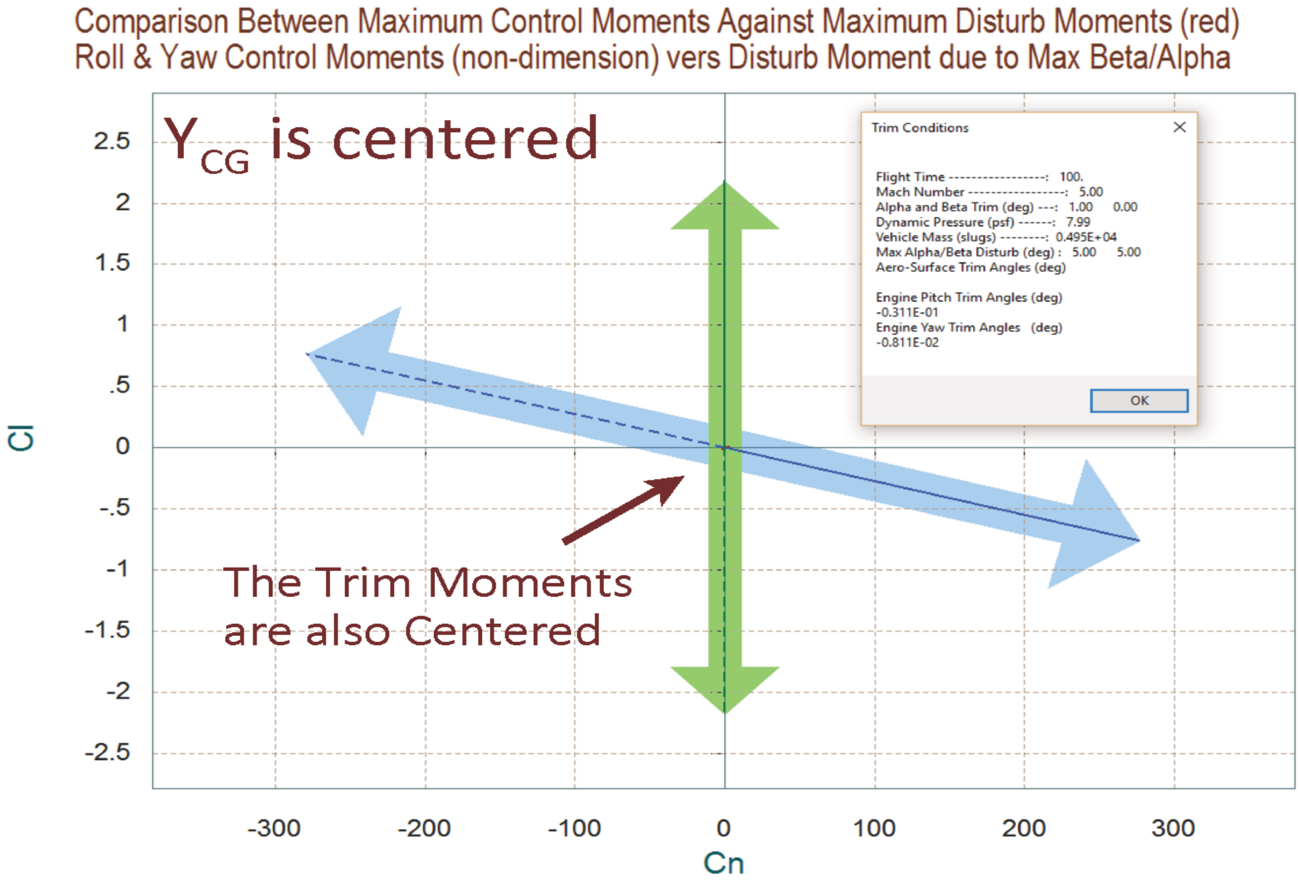


Lateral Stability Contour Plot (Mach vs Alpha)

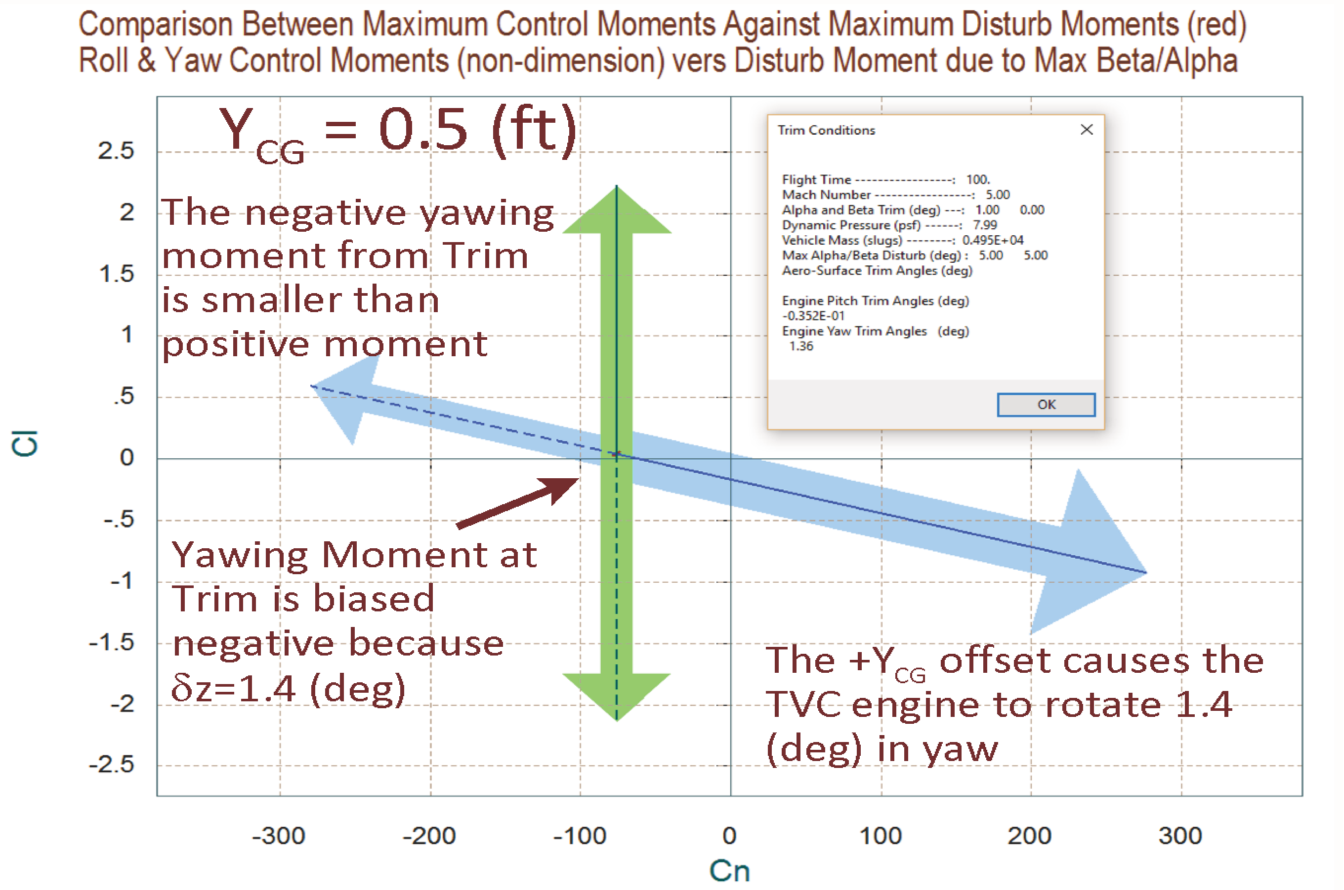
Trajectory Parameters

The Selected Point is at:

Trajectory Time: 111.7 (sec)
Mach Number : 6.843
Angle of Attack: 3.486 (deg)
Altitude : 0.22970E+06 (feet)
Velocity : 6467.9 (feet/sec)
Dynamic Pressure: 2.8200 (psf)
Vehicle Mass: 4422.0 (slug)



```
Flight Time -----------------:   100.
Mach Number -----------------:   5.00
Alpha and Beta Trim (deg) ---:   1.00     0.00
Dynamic Pressure (psf) ------:   7.99
Vehicle Mass (slugs) --------:  0.495E+04
Max Alpha/Beta Disturb (deg) :   5.00     5.00
Aero-Surface Trim Angles (deg)

Engine Pitch Trim Angles (deg)
-0.311E-01
Engine Yaw Trim Angles   (deg)
-0.811E-02
```



```
Flight Time -----------------:   100.
Mach Number -----------------:   5.00
Alpha and Beta Trim (deg) ---:   1.00     0.00
Dynamic Pressure (psf) ------:   7.99
Vehicle Mass (slugs) --------:  0.495E+04
Max Alpha/Beta Disturb (deg) :   5.00     5.00
Aero-Surface Trim Angles (deg)

Engine Pitch Trim Angles (deg)
-0.352E-01
Engine Yaw Trim Angles   (deg)
  1.36
```

2. Flight Control Design

Our next step is to design a simple and preliminary flight control system that will be used in the 6-dof simulation. It will be updated later as the design progresses. The control system consists of state-feedback gains and it is designed by the LQR method using state-space models of the launch vehicle in different flight conditions. The gains are then interpolated in-between flight conditions. The LQR method requires simple design models of the vehicle at each flight time and the control gains are synthesized around those models. The vehicle models are generated by the Flixan program using the Static Analysis/ Trim option. The program creates separate vehicle input data files at different flight times from release, and the files (together with the Matlab scripts) are placed in different folders under "FCS Design". The structures of the vehicle models are different between first and second stages. During first stage the launch vehicle is controlled by the two side boosters, but in second stage it is controlled by the middle booster engine and the two reaction control thrusters for roll. The gains from each flight condition folder are calculated from Matlab scripts that use the Flixan systems and Simulink models. Then, they are collected in an Excel file and exported to the 6-dof simulation.

2.1 Flight Control System Architecture

The pitch flight control system is shown in Figure 2.1. It uses state-feedback from the pitch attitude θ, pitch rate q, angle of attack α, and the integral of the flight-path angle γ. It calculates the pitch control demand that is converted to pitch engine deflections (δy) by the mixing-logic matrix. There are two guidance commands: pitch attitude and flight-path angle and they are coordinated. Otherwise, it is impossible to achieve independent tracking of θ and γ simultaneously. The reason for applying two commands is because in some instances, such as, immediately after engine ignition, it is important to control only attitude in order to achieve a successful pitch up maneuver and to increase gamma before the dynamic pressure becomes high in which case it's hard to raise γ. The gains from the various states are adjusted to either emphasize attitude control or γ-tracking. At high dynamic pressures the alpha gain is increased in order to minimize the angle of attack dispersions and, therefore, structural loading. These adjustments are made by varying the Q and R penalties in the LQR design.

The lateral control system is shown in Figure 2.2. This is also a state-feedback from the roll and yaw attitude (ϕ and ψ), the body rates (p and r), and the sideslip angle β. The state-feedback outputs are roll and yaw control demands which are converted to pitch and yaw engine deflections (δy and δz) by the mixing-logic matrix. Figure 2.3 shows the lateral guidance system. The aircraft releases the vehicle in a direction that will achieve the proper orbit inclination. Flight direction errors may occur, however, due to cross-wind disturbances. The directional error activates the roll and yaw guidance commands that correct the flight direction and place the vehicle in the proper inclination angle.

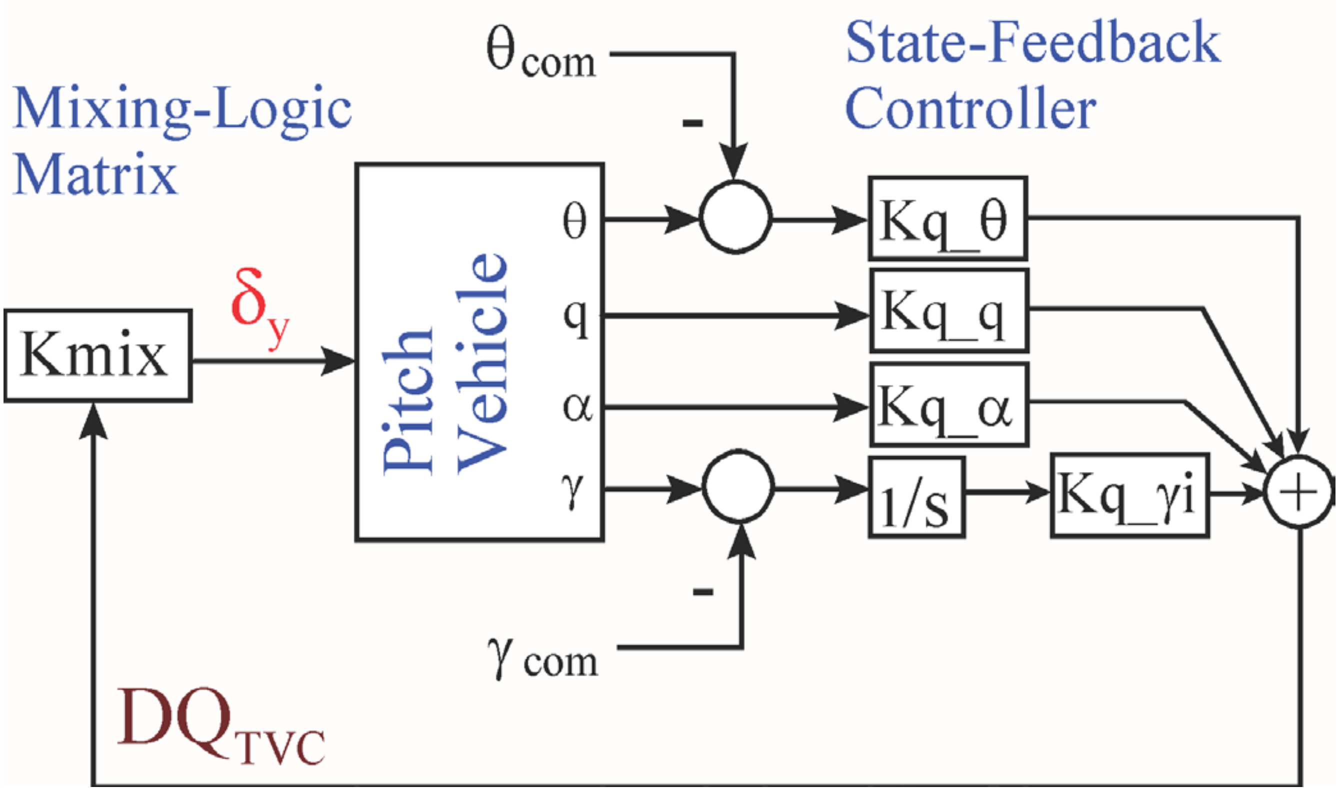


Figure 2.1 Pitch Control System uses State-Feedback to Generate the Pitch Control Demand DQ_{TVC}

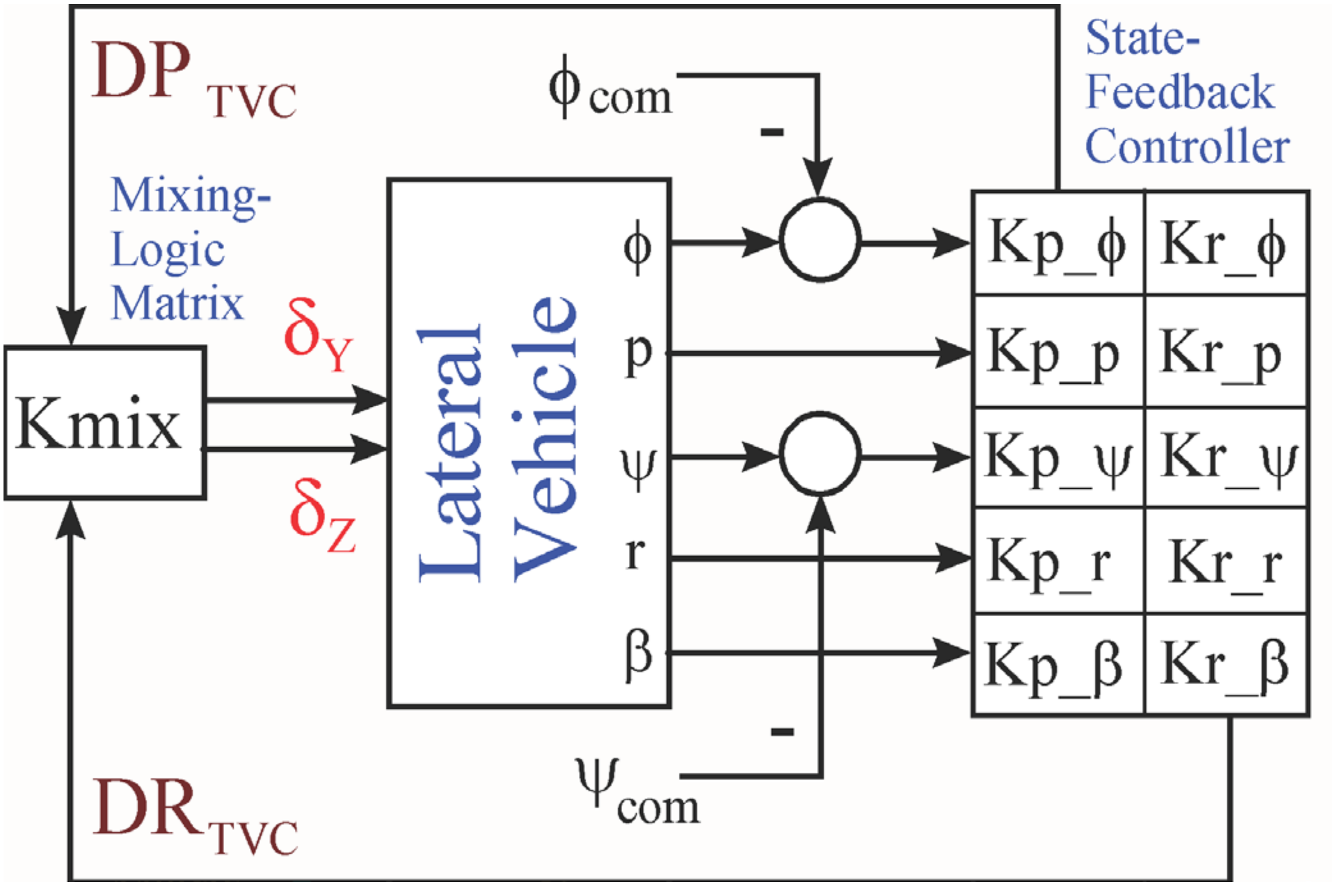


Figure 2.2 Lateral Control System uses State-Feedback to Generate the Roll and Yaw Control Demands

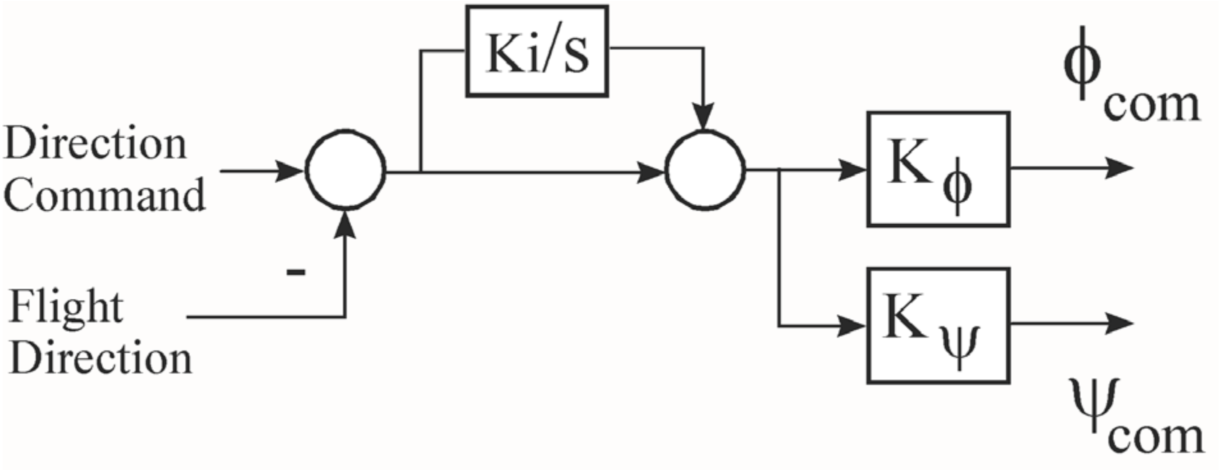
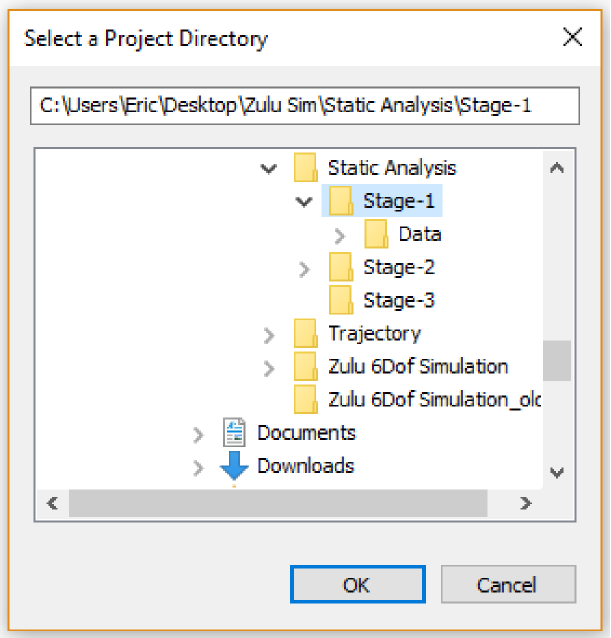
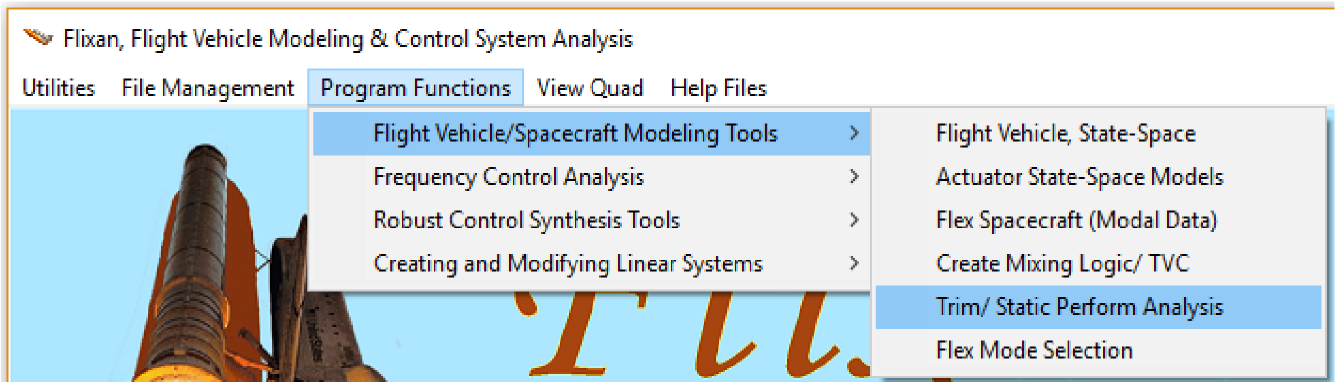


Figure 2.3 Lateral Guidance Generates the Roll and Yaw Attitude Commands

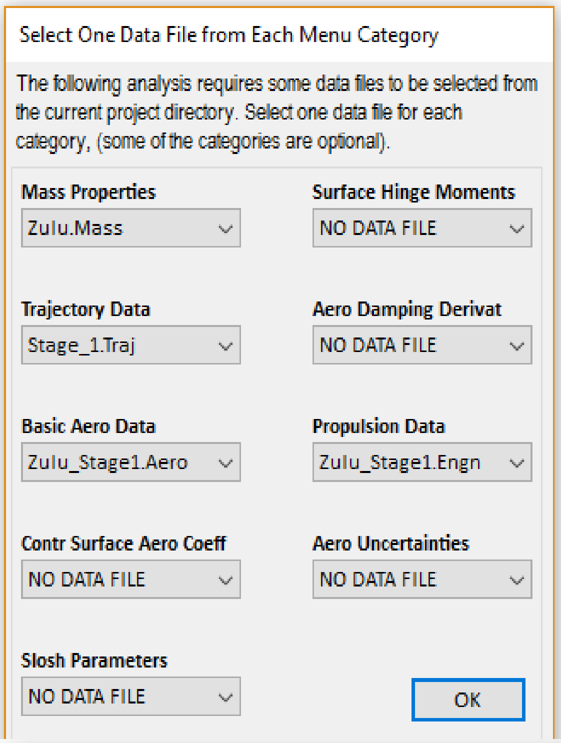
2.2 Creating the Vehicle Design Models

The input data file (.Inp) contains the vehicle data for a specific flight condition. This input file will be processed by the Flixan vehicle modeling program to generate the vehicle state-space models that will be used in the control design. This input file can either be created manually by modifying a previous file or it can be created using a Flixan option that is available in the Trim application and it can be selected as shown in the following first stage example. You must first start the Flixan program and select the directory that includes the first stage trajectory, mass, and aero data. Then, from the main menu select the "Trim/ Static Analysis Program".


C:\Users\Eric\Desktop\Zulu Sim\Static Analysis\Stage-1



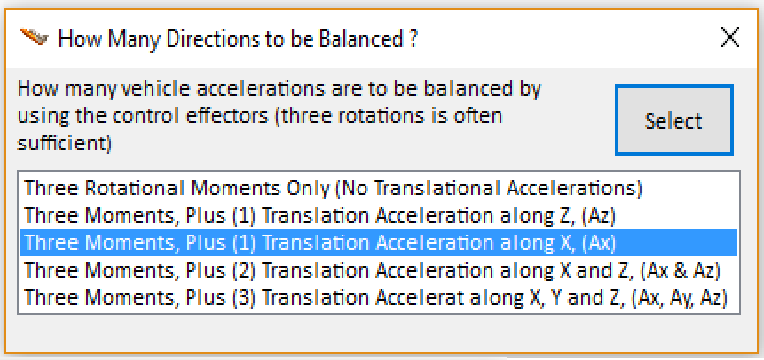
Select the data files as it was shown in the previous static analysis, and from the Trim main menu select the 3rd option to trim the vehicle along the trajectory, as before. That is because the trim file (.Trim) is needed for the creation of the vehicle input data. Make sure that you select 3 rotations plus the axial acceleration, as before.



Select One Data File from Each Menu Category

The following analysis requires some data files to be selected from the current project directory. Select one data file for each category, (some of the categories are optional).

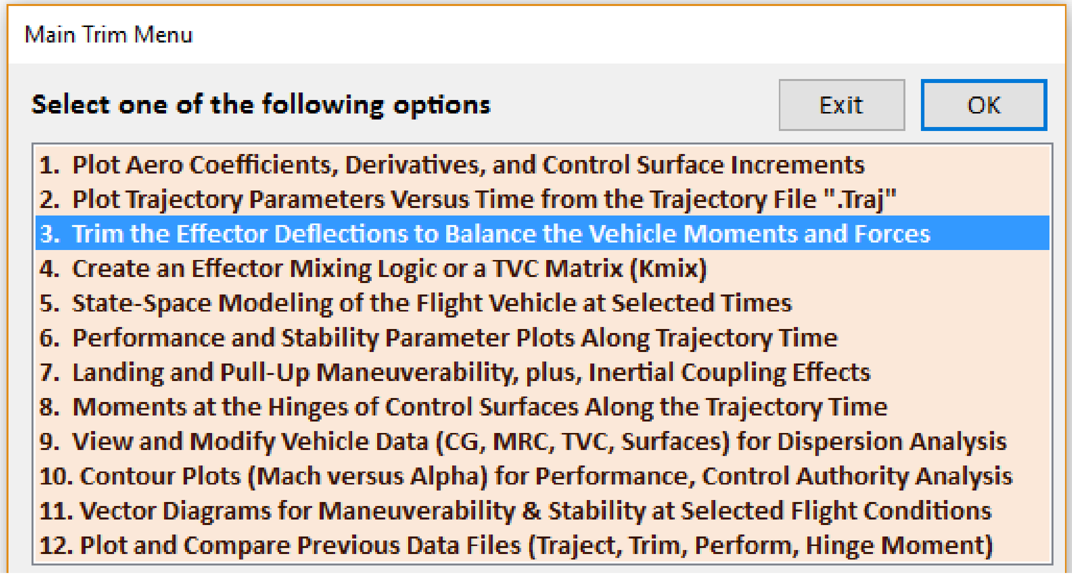
Mass Properties: Zulu.Mass	Surface Hinge Moments: NO DATA FILE
Trajectory Data: Stage_1.Traj	Aero Damping Derivat: NO DATA FILE
Basic Aero Data: Zulu_Stage1.Aero	Propulsion Data: Zulu_Stage1.Engn
Contr Surface Aero Coeff: NO DATA FILE	Aero Uncertainties: NO DATA FILE
Slosh Parameters: NO DATA FILE	



How Many Directions to be Balanced ?

How many vehicle accelerations are to be balanced by using the control effectors (three rotations is often sufficient)

Three Rotational Moments Only (No Translational Accelerations)
Three Moments, Plus (1) Translation Acceleration along Z, (Az)
Three Moments, Plus (1) Translation Acceleration along X, (Ax)
Three Moments, Plus (2) Translation Acceleration along X and Z, (Ax & Az)
Three Moments, Plus (3) Translation Accelerat along X, Y and Z, (Ax, Ay, Az)

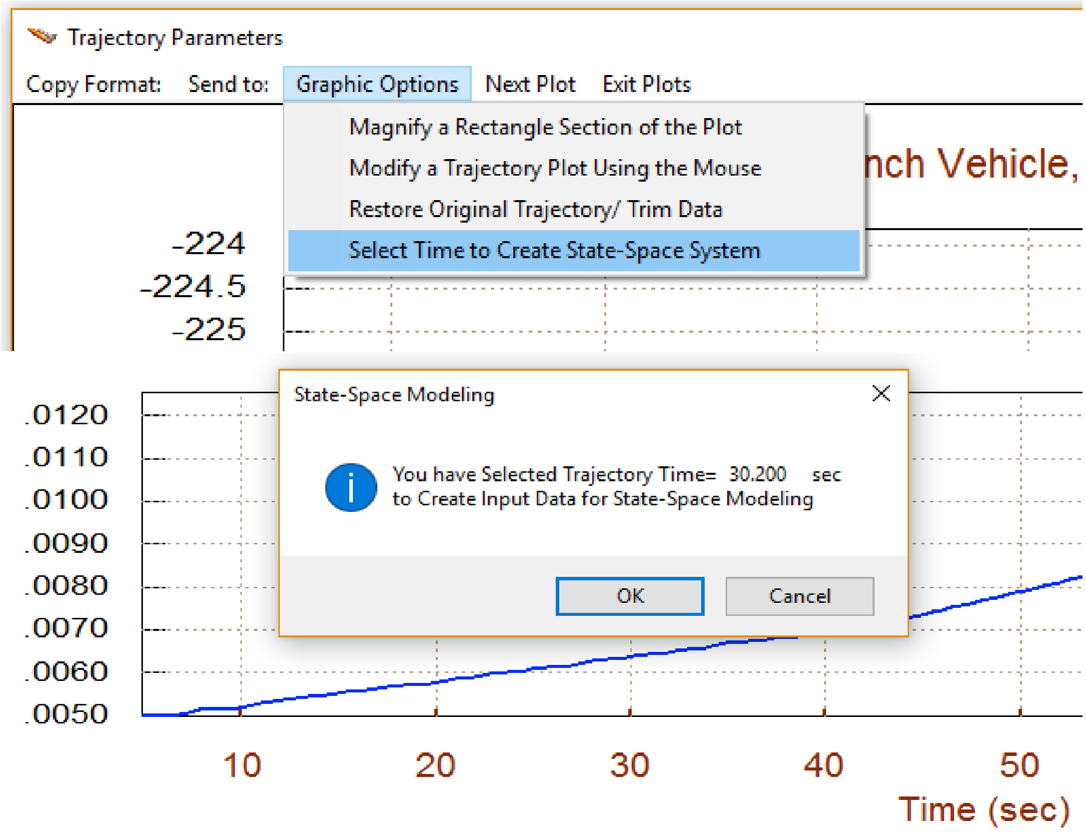


Main Trim Menu

Select one of the following options

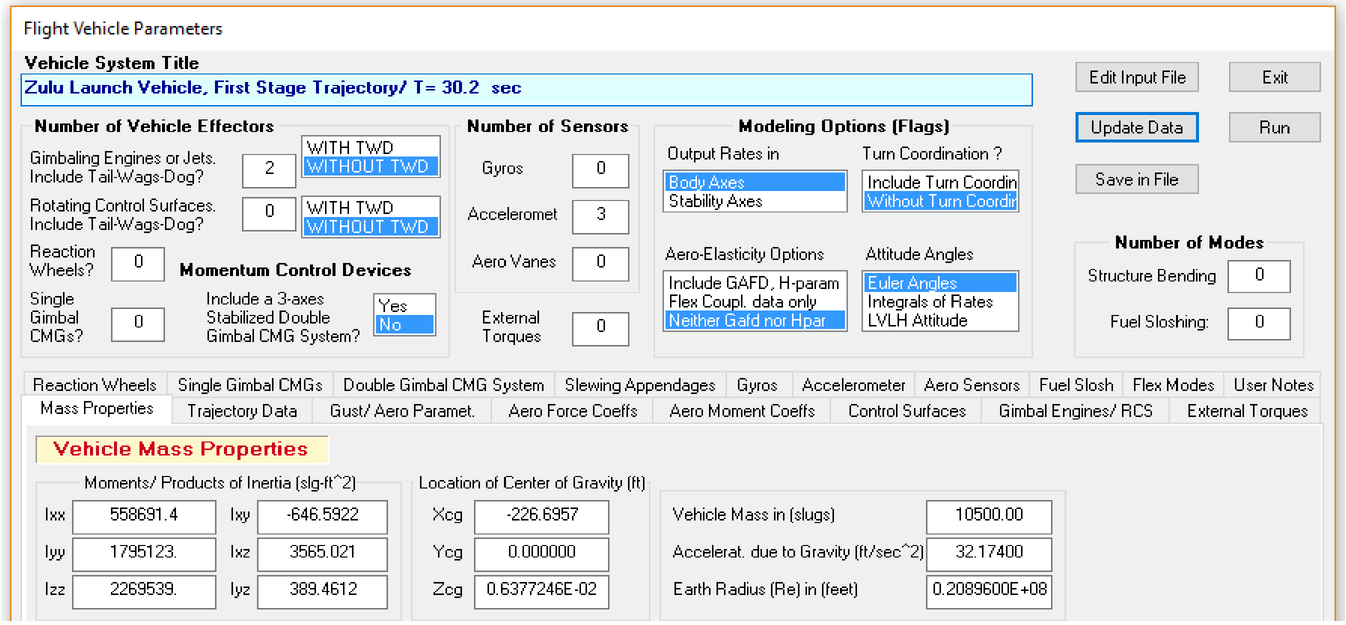
1. Plot Aero Coefficients, Derivatives, and Control Surface Increments
2. Plot Trajectory Parameters Versus Time from the Trajectory File ".Traj"
3. Trim the Effector Deflections to Balance the Vehicle Moments and Forces
4. Create an Effector Mixing Logic or a TVC Matrix (Kmix)
5. State-Space Modeling of the Flight Vehicle at Selected Times
6. Performance and Stability Parameter Plots Along Trajectory Time
7. Landing and Pull-Up Maneuverability, plus, Inertial Coupling Effects
8. Moments at the Hinges of Control Surfaces Along the Trajectory Time
9. View and Modify Vehicle Data (CG, MRC, TVC, Surfaces) for Dispersion Analysis
10. Contour Plots (Mach versus Alpha) for Performance, Control Authority Analysis
11. Vector Diagrams for Maneuverability & Stability at Selected Flight Conditions
12. Plot and Compare Previous Data Files (Traject, Trim, Perform, Hinge Moment)

After trimming return to the main menu and select the 2nd option to plot the first stage trajectory. The trajectory plot includes a menu bar at the top. Go to the "*Graphic Options*" and click on "*Select Time to Create a State-Space System*", as shown.



You have Selected Trajectory Time= 30.200 sec
to Create Input Data for State-Space Modeling

Using the mouse select a time along the horizontal scale, 30.2 seconds in this demo, to create the linear model. The program confirms the time that you have selected and you must click OK, otherwise "Cancel" to change the flight time. The vehicle modeling program presents the following dialog with tabs, from where the user may check the vehicle data. Click on "*Save in File*" to save the vehicle data in the input file.



Vehicle System Title
Zulu Launch Vehicle, First Stage Trajectory/ T= 30.2 sec

Number of Vehicle Effectors
Gimbaling Engines or Jets. Include Tail-Wags-Dog? 2 WITH TWD / WITHOUT TWD (selected)
Rotating Control Surfaces. Include Tail-Wags-Dog? 0 WITH TWD / WITHOUT TWD (selected)
Reaction Wheels? 0
Single Gimbal CMGs? 0

Momentum Control Devices
Include a 3-axes Stabilized Double Gimbal CMG System? Yes / No (selected)

Number of Sensors
Gyros 0
Acceleromet 3
Aero Vanes 0
External Torques 0

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

Number of Modes
Structure Bending 0
Fuel Sloshing: 0

Vehicle Mass Properties

Moments/ Products of Inertia (slg-ft^2)

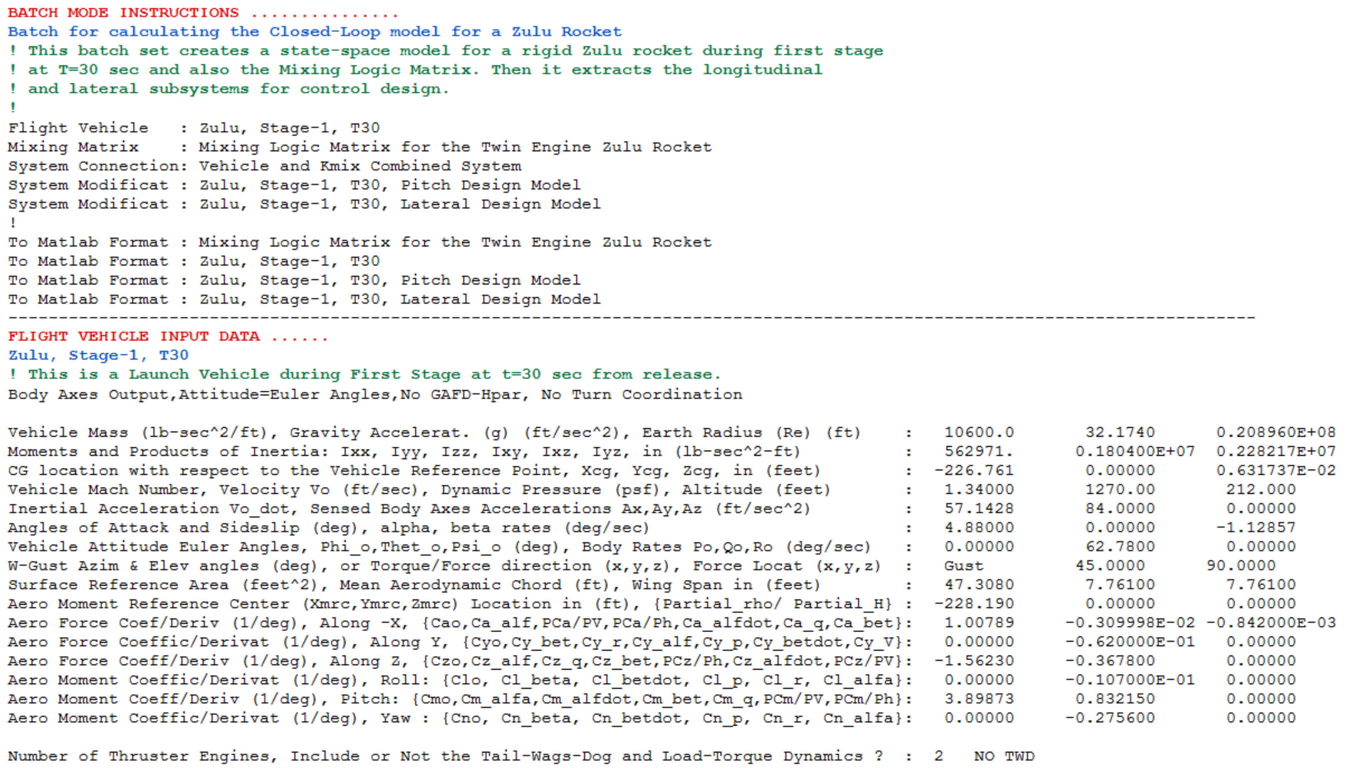
Ixx	558691.4	Ixy	-646.5922
Iyy	1795123.	Ixz	3565.021
Izz	2269539.	Iyz	389.4612

Location of Center of Gravity (ft)
Xcg -226.6957
Ycg 0.000000
Zcg 0.6377246E-02

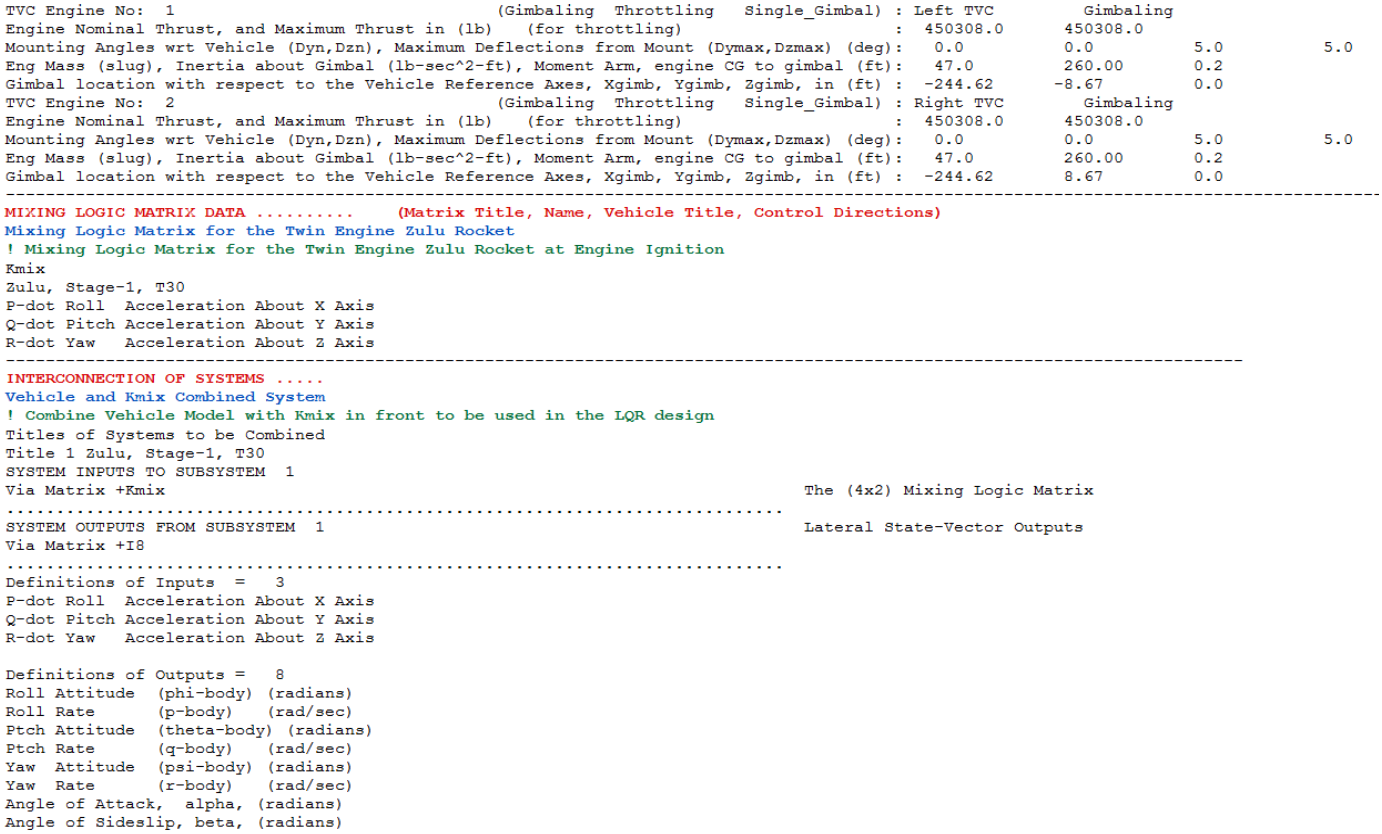
Vehicle Mass in (slugs) 10500.00
Accelerat. due to Gravity (ft/sec^2) 32.17400
Earth Radius (Re) in (feet) 0.2089600E+08

The input file for this flight condition is “*Zulu_T30.Inp*” and it is shown below. In addition to the vehicle data, the input file also includes other datasets for Flixan applications that perform other functions related to this vehicle modeling project. There are datasets that extract separate pitch and lateral design models, generate a mixing-logic matrix for this flight condition, perform system interconnections, and generate systems that can be loaded into Matlab. It also includes a batch that speeds up the processing of the input file.

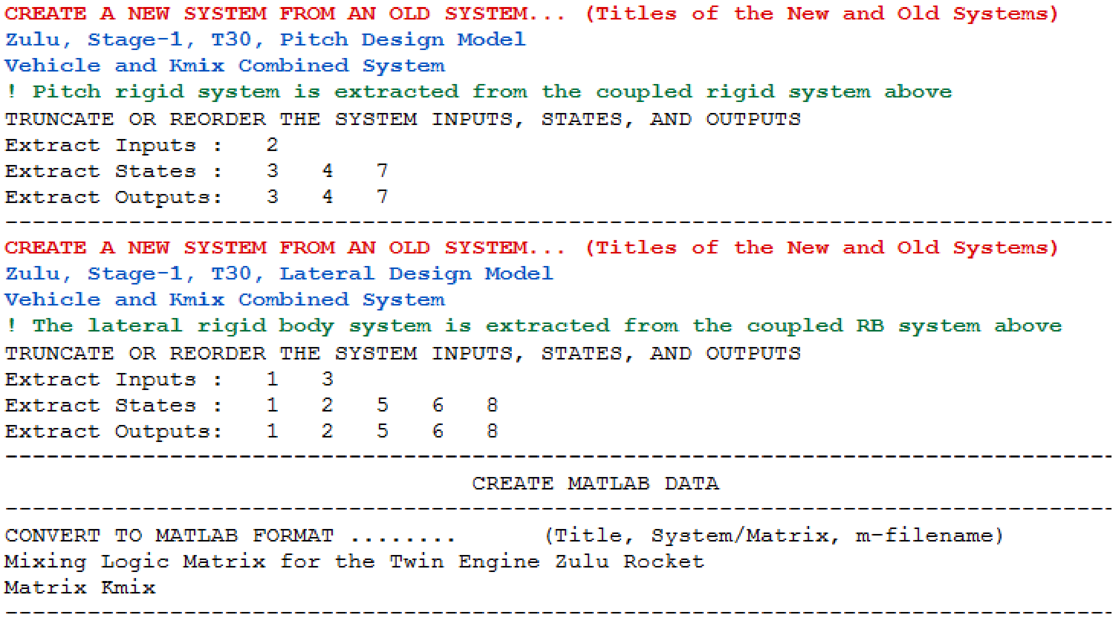
The pitch vehicle system includes the states: (θ, q, and α) and it is saved in m-file: “vehi_pitch_des.m”. The lateral system states are: (ϕ, p, ψ, r, and β) and it is saved in m-file: “vehi_later_des.m”. The design models include the mixing-logic matrix “Kmix.Mat” that converts the roll, pitch, and yaw FCS acceleration demands to TVC pitch and yaw engine deflections. Notice, the inputs to the mixing logic matrix in the control design do not include the axial acceleration but only the rotational demands because the engine throttling to control the acceleration is performed open-loop ant it is, therefore, not included in the flight control system. The rigid vehicle model “*Zulu, Stage-1, T30*” is also saved as an m-file: “*vehicle_rb.m*”. It will be used later in Matlab to evaluate and adjust the gains by analyzing the closed-loop system performance.



```
BATCH MODE INSTRUCTIONS ...............
Batch for calculating the Closed-Loop model for a Zulu Rocket
! This batch set creates a state-space model for a rigid Zulu rocket during first stage
! at T=30 sec and also the Mixing Logic Matrix. Then it extracts the longitudinal
! and lateral subsystems for control design.
!
Flight Vehicle    : Zulu, Stage-1, T30
Mixing Matrix     : Mixing Logic Matrix for the Twin Engine Zulu Rocket
System Connection: Vehicle and Kmix Combined System
System Modificat : Zulu, Stage-1, T30, Pitch Design Model
System Modificat : Zulu, Stage-1, T30, Lateral Design Model
!
To Matlab Format : Mixing Logic Matrix for the Twin Engine Zulu Rocket
To Matlab Format : Zulu, Stage-1, T30
To Matlab Format : Zulu, Stage-1, T30, Pitch Design Model
To Matlab Format : Zulu, Stage-1, T30, Lateral Design Model
----------------------------------------------------------------------------------------------------------------------------
FLIGHT VEHICLE INPUT DATA ......
Zulu, Stage-1, T30
! This is a Launch Vehicle during First Stage at t=30 sec from release.
Body Axes Output,Attitude=Euler Angles,No GAFD-Hpar, No Turn Coordination

Vehicle Mass (lb-sec^2/ft), Gravity Accelerat. (g) (ft/sec^2), Earth Radius (Re) (ft)    :   10600.0       32.1740      0.208960E+08
Moments and Products of Inertia: Ixx, Iyy, Izz, Ixy, Ixz, Iyz, in (lb-sec^2-ft)          :   562971.      0.180400E+07  0.228217E+07
CG location with respect to the Vehicle Reference Point, Xcg, Ycg, Zcg, in (feet)        :  -226.761       0.00000      0.631737E-02
Vehicle Mach Number, Velocity Vo (ft/sec), Dynamic Pressure (psf), Altitude (feet)       :   1.34000       1270.00       212.000
Inertial Acceleration Vo_dot, Sensed Body Axes Accelerations Ax,Ay,Az (ft/sec^2)         :   57.1428       84.0000       0.00000
Angles of Attack and Sideslip (deg), alpha, beta rates (deg/sec)                         :   4.88000       0.00000      -1.12857
Vehicle Attitude Euler Angles, Phi_o,Thet_o,Psi_o (deg), Body Rates Po,Qo,Ro (deg/sec)   :   0.00000       62.7800       0.00000
W-Gust Azim & Elev angles (deg), or Torque/Force direction (x,y,z), Force Locat (x,y,z)  :   Gust         45.0000      90.0000
Surface Reference Area (feet^2), Mean Aerodynamic Chord (ft), Wing Span in (feet)        :   47.3080       7.76100       7.76100
Aero Moment Reference Center (Xmrc,Ymrc,Zmrc) Location in (ft), {Partial_rho/ Partial_H} :  -228.190       0.00000       0.00000
Aero Force Coef/Deriv (1/deg), Along -X, {Cao,Ca_alf,PCa/PV,PCa/Ph,Ca_alfdot,Ca_q,Ca_bet}:   1.00789     -0.309998E-02 -0.842000E-03
Aero Force Coeffic/Derivat (1/deg), Along Y, {Cyo,Cy_bet,Cy_r,Cy_alf,Cy_p,Cy_betdot,Cy_V}:   0.00000     -0.620000E-01   0.00000
Aero Force Coeff/Deriv (1/deg), Along Z, {Czo,Cz_alf,Cz_q,Cz_bet,PCz/Ph,Cz_alfdot,PCz/PV}:  -1.56230      -0.367800       0.00000
Aero Moment Coeffic/Derivat (1/deg), Roll: {Clo, Cl_beta, Cl_betdot, Cl_p, Cl_r, Cl_alfa}:   0.00000     -0.107000E-01   0.00000
Aero Moment Coeff/Deriv (1/deg), Pitch: {Cmo,Cm_alfa,Cm_alfdot,Cm_bet,Cm_q,PCm/PV,PCm/Ph}:   3.89873       0.832150       0.00000
Aero Moment Coeffic/Derivat (1/deg), Yaw : {Cno, Cn_beta, Cn_betdot, Cn_p, Cn_r, Cn_alfa}:   0.00000      -0.275600       0.00000

Number of Thruster Engines, Include or Not the Tail-Wags-Dog and Load-Torque Dynamics ?  :  2   NO TWD
```

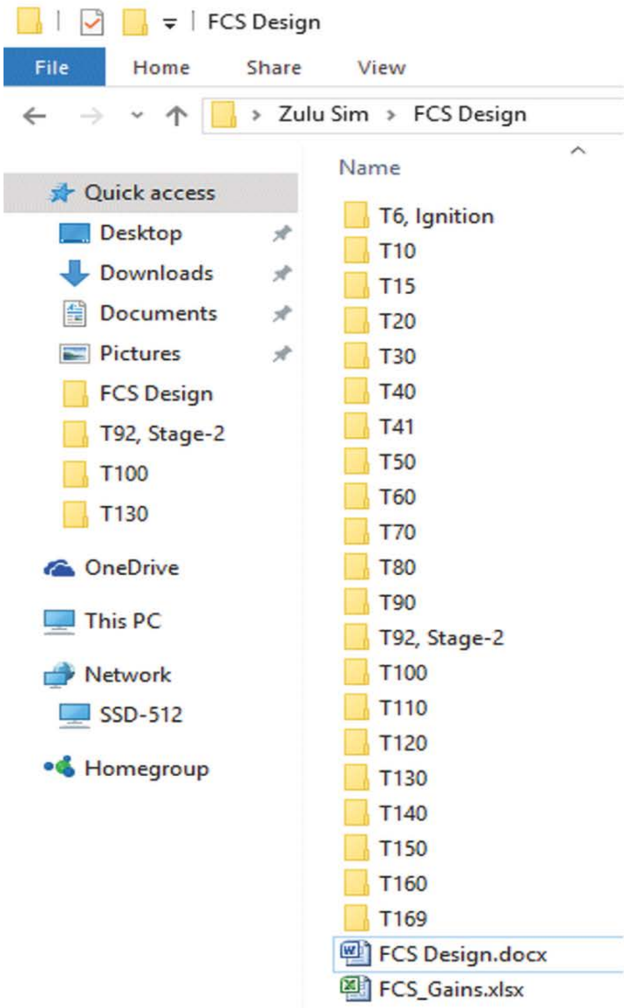


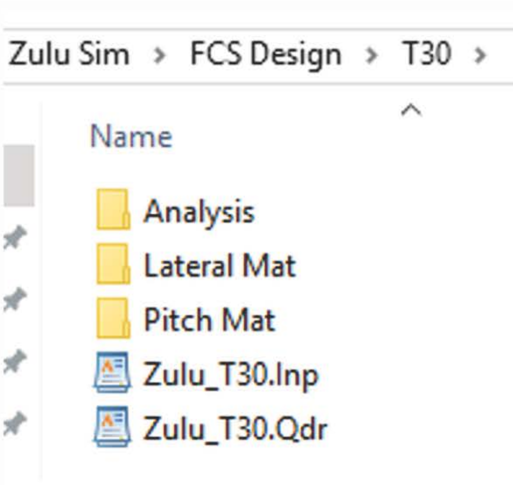
```
TVC Engine No:  1                                 (Gimbaling  Throttling   Single_Gimbal) : Left TVC        Gimbaling
Engine Nominal Thrust, and Maximum Thrust in (lb)    (for throttling)                  :  450308.0      450308.0
Mounting Angles wrt Vehicle (Dyn,Dzn), Maximum Deflections from Mount (Dymax,Dzmax) (deg):   0.0          0.0           5.0           5.0
Eng Mass (slug), Inertia about Gimbal (lb-sec^2-ft), Moment Arm, engine CG to gimbal (ft):   47.0         260.00        0.2
Gimbal location with respect to the Vehicle Reference Axes, Xgimb, Ygimb, Zgimb, in (ft) :  -244.62      -8.67         0.0
TVC Engine No:  2                                 (Gimbaling  Throttling   Single_Gimbal) : Right TVC       Gimbaling
Engine Nominal Thrust, and Maximum Thrust in (lb)    (for throttling)                  :  450308.0      450308.0
Mounting Angles wrt Vehicle (Dyn,Dzn), Maximum Deflections from Mount (Dymax,Dzmax) (deg):   0.0          0.0           5.0           5.0
Eng Mass (slug), Inertia about Gimbal (lb-sec^2-ft), Moment Arm, engine CG to gimbal (ft):   47.0         260.00        0.2
Gimbal location with respect to the Vehicle Reference Axes, Xgimb, Ygimb, Zgimb, in (ft) :  -244.62       8.67         0.0
------------------------------------------------------------------------------------------------------------------------------------------
MIXING LOGIC MATRIX DATA ..........     (Matrix Title, Name, Vehicle Title, Control Directions)
Mixing Logic Matrix for the Twin Engine Zulu Rocket
! Mixing Logic Matrix for the Twin Engine Zulu Rocket at Engine Ignition
Kmix
Zulu, Stage-1, T30
P-dot Roll  Acceleration About X Axis
Q-dot Pitch Acceleration About Y Axis
R-dot Yaw   Acceleration About Z Axis
------------------------------------------------------------------------------------------------------------------------
INTERCONNECTION OF SYSTEMS .....
Vehicle and Kmix Combined System
! Combine Vehicle Model with Kmix in front to be used in the LQR design
Titles of Systems to be Combined
Title 1 Zulu, Stage-1, T30
SYSTEM INPUTS TO SUBSYSTEM  1
Via Matrix +Kmix                                                            The (4x2) Mixing Logic Matrix
...........................................................................
SYSTEM OUTPUTS FROM SUBSYSTEM  1                                            Lateral State-Vector Outputs
Via Matrix +I8
...........................................................................
Definitions of Inputs  =   3
P-dot Roll  Acceleration About X Axis
Q-dot Pitch Acceleration About Y Axis
R-dot Yaw   Acceleration About Z Axis

Definitions of Outputs =   8
Roll Attitude  (phi-body)  (radians)
Roll Rate      (p-body)    (rad/sec)
Ptch Attitude  (theta-body)  (radians)
Ptch Rate      (q-body)    (rad/sec)
Yaw  Attitude  (psi-body)  (radians)
Yaw  Rate      (r-body)    (rad/sec)
Angle of Attack,  alpha,  (radians)
Angle of Sideslip, beta,  (radians)
------------------------------------------------------------------------------------------------------------------------------------------
```

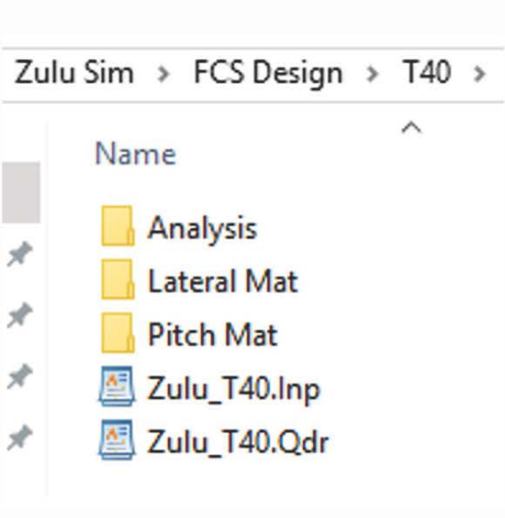


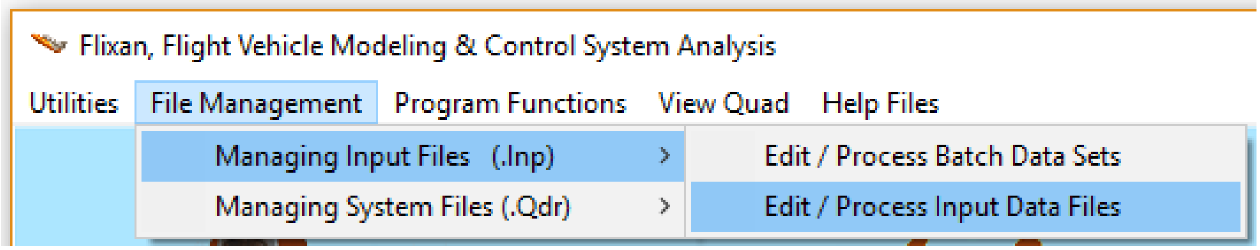
```
CREATE A NEW SYSTEM FROM AN OLD SYSTEM... (Titles of the New and Old Systems)
Zulu, Stage-1, T30, Pitch Design Model
Vehicle and Kmix Combined System
! Pitch rigid system is extracted from the coupled rigid system above
TRUNCATE OR REORDER THE SYSTEM INPUTS, STATES, AND OUTPUTS
Extract Inputs :    2
Extract States :    3    4    7
Extract Outputs:    3    4    7
--------------------------------------------------------------------------
CREATE A NEW SYSTEM FROM AN OLD SYSTEM... (Titles of the New and Old Systems)
Zulu, Stage-1, T30, Lateral Design Model
Vehicle and Kmix Combined System
! The lateral rigid body system is extracted from the coupled RB system above
TRUNCATE OR REORDER THE SYSTEM INPUTS, STATES, AND OUTPUTS
Extract Inputs :    1    3
Extract States :    1    2    5    6    8
Extract Outputs:    1    2    5    6    8
--------------------------------------------------------------------------
                              CREATE MATLAB DATA
--------------------------------------------------------------------------
CONVERT TO MATLAB FORMAT ........      (Title, System/Matrix, m-filename)
Mixing Logic Matrix for the Twin Engine Zulu Rocket
Matrix Kmix
--------------------------------------------------------------------------
```

The data files for each flight condition are placed in separate folders under “FCS Design”, as shown below. To process the data file in batch mode go to “*File Management*”, “*Managing Input Files*”, and “*Edit/ Process Input Data Files*”.

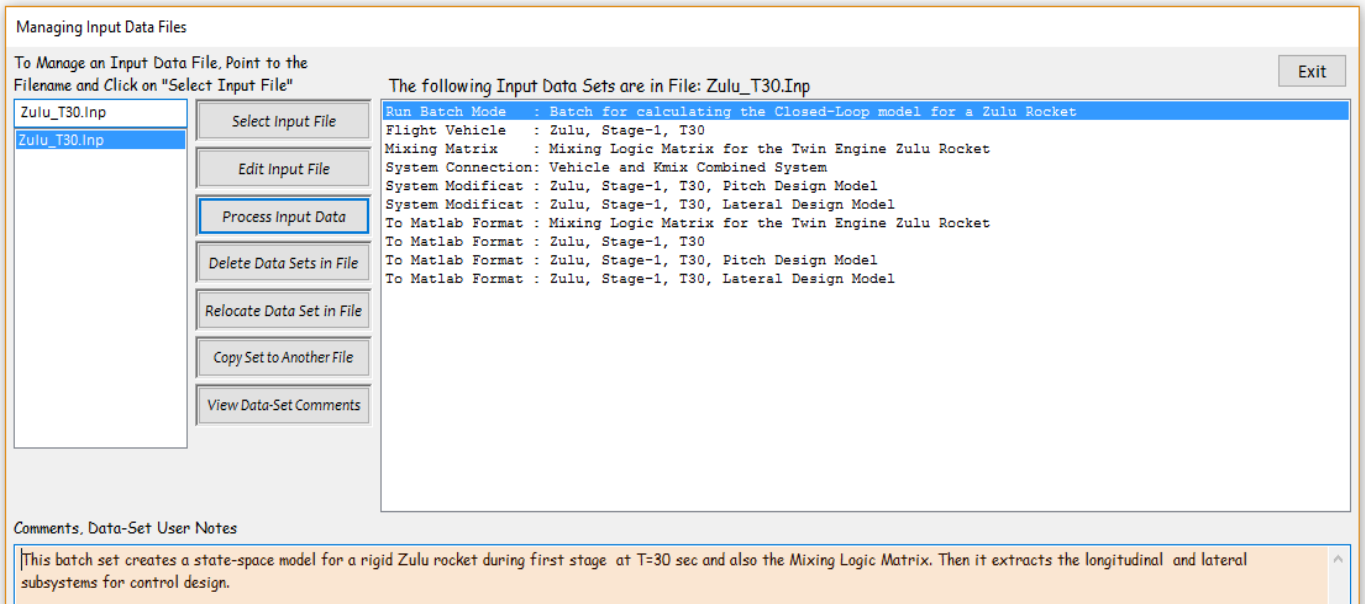








The input file manager dialog comes up. From the left menu select the input file "*Zulu_T30.Inp*" and the right menu will show the titles of the datasets that are saved in this file. Select the batch set, which is at the top, and click on "*Process Input Data*". The program will process the data and generate the m-files for the Matlab analysis.



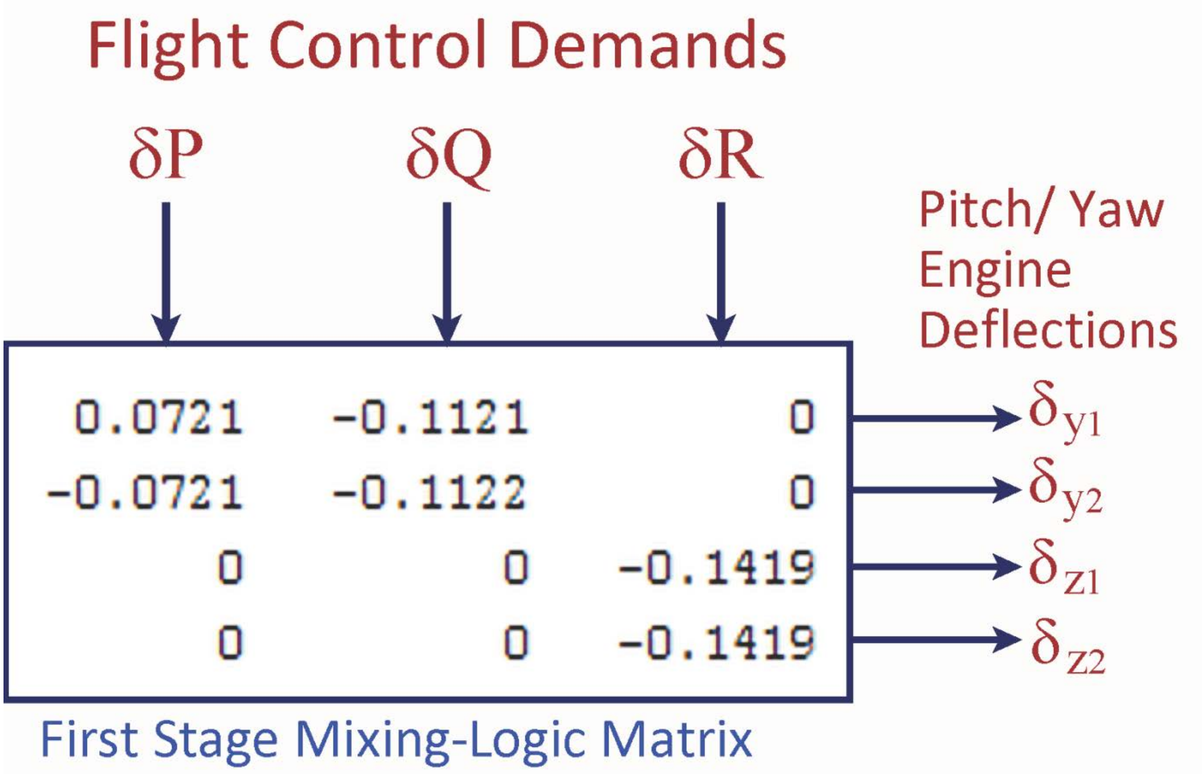
The following Input Data Sets are in File: Zulu_T30.Inp

```
Run Batch Mode    : Batch for calculating the Closed-Loop model for a Zulu Rocket
Flight Vehicle    : Zulu, Stage-1, T30
Mixing Matrix     : Mixing Logic Matrix for the Twin Engine Zulu Rocket
System Connection: Vehicle and Kmix Combined System
System Modificat : Zulu, Stage-1, T30, Pitch Design Model
System Modificat : Zulu, Stage-1, T30, Lateral Design Model
To Matlab Format : Mixing Logic Matrix for the Twin Engine Zulu Rocket
To Matlab Format : Zulu, Stage-1, T30
To Matlab Format : Zulu, Stage-1, T30, Pitch Design Model
To Matlab Format : Zulu, Stage-1, T30, Lateral Design Model
```

This batch set creates a state-space model for a rigid Zulu rocket during first stage at T=30 sec and also the Mixing Logic Matrix. Then it extracts the longitudinal and lateral subsystems for control design.

2.3 Simple “Time-Slice” Design during First Stage

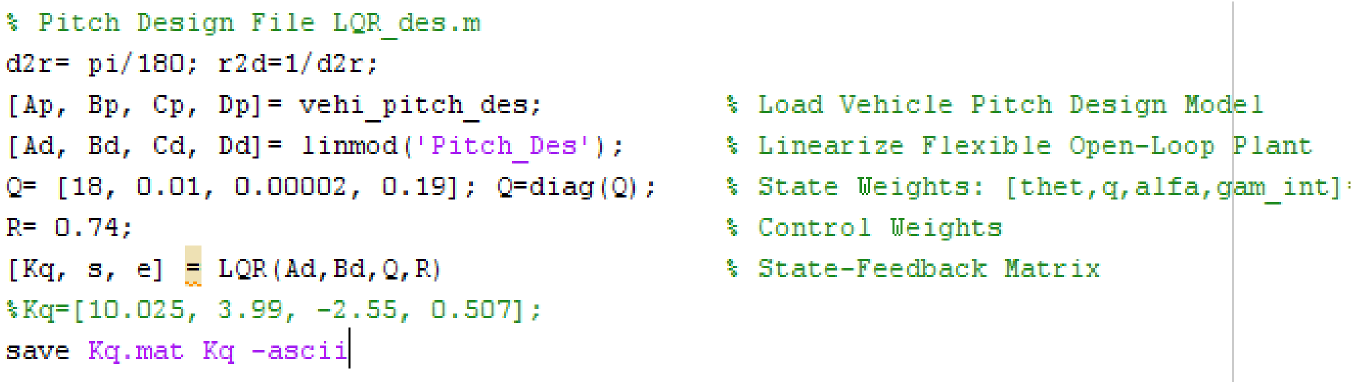
During first stage the vehicle uses the two side boosters for roll, pitch, and yaw control. The middle booster is not firing. The mixing logic is a (4x3) matrix that converts the 3 demands from the flight control system to pitch and yaw TVC deflections (δyi and δzi). A roll demand rotates the two engines differentially. A +pitch demand rotates the engines in the –pitch direction, and +yaw demand rotates the engines in the –yaw direction.



After processing the input file, move the m-file: “*vehi_pitch_des.m*” to the pitch design folder “Pitch_Mat”, and the m-file “*vehi_later_des.m*” to the lateral design folder “Lateral_Mat”. The mixing-logic matrix and the rigid vehicle m-file “*vehicle_rb.m*” go to the analysis folder.

2.3.1 Pitch Control Design

The pitch control gain is created in subdirectory “Pitch_Mat” by processing the file “*LQR_des.m*” in Matlab. This script loads the pitch design model and uses the LQR function to generate the state-feedback gain matrix Kq.mat. The original design model consists of states: (θ, q, and α). An additional state γ-integral is included, as shown in Figure 2.4, where: $\gamma=\theta-\alpha$, to improve tracking of the flight-path angle γ.



```
% Pitch Design File LQR_des.m
d2r= pi/180; r2d=1/d2r;
[Ap, Bp, Cp, Dp]= vehi_pitch_des;           % Load Vehicle Pitch Design Model
[Ad, Bd, Cd, Dd]= linmod('Pitch_Des');      % Linearize Flexible Open-Loop Plant
Q= [18, 0.01, 0.00002, 0.19]; Q=diag(Q);    % State Weights: [thet,q,alfa,gam_int]
R= 0.74;                                    % Control Weights
[Kq, s, e] = LQR(Ad,Bd,Q,R)                 % State-Feedback Matrix
%Kq=[10.025, 3.99, -2.55, 0.507];
save Kq.mat Kq -ascii
```

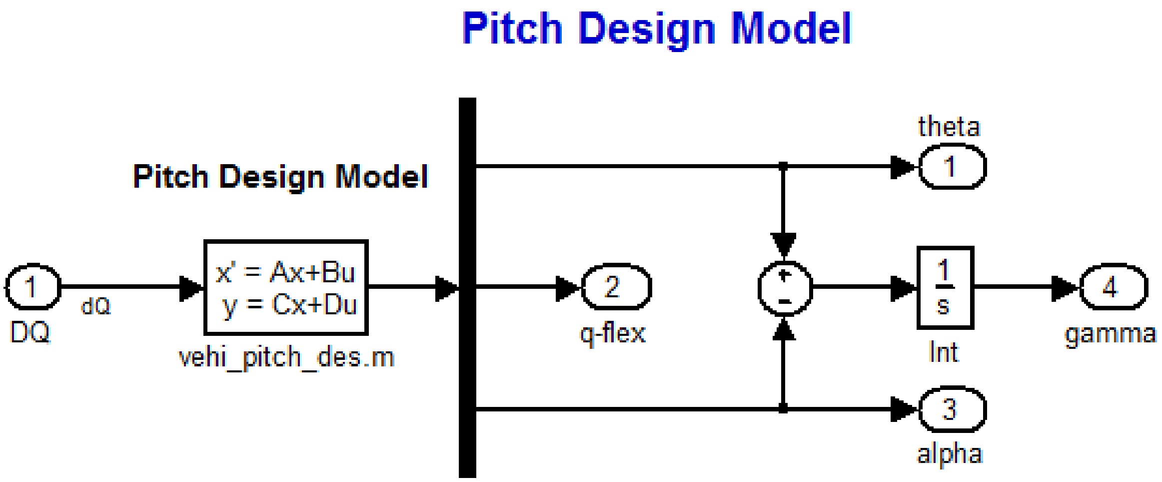
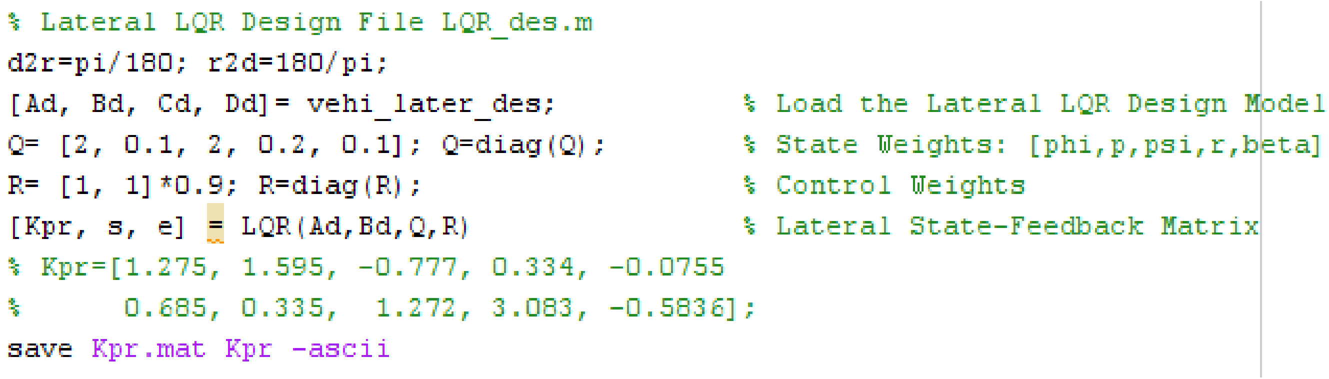


Figure 2.4 The Pitch State-Space System is Augmented to Include Gamma Integral

2.3.2 Lateral Control Design

The lateral LQR design is similar and the files are located in subdirectory “Lateral_Mat”. The script below loads the lateral vehicle design model and generates the lateral state-feedback matrix Kpr.mat which calculates the roll and yaw control demands as a function of the lateral states: (ϕ, p, ψ, r, and β).



```
% Lateral LQR Design File LQR_des.m
d2r=pi/180; r2d=180/pi;
[Ad, Bd, Cd, Dd]= vehi_later_des;           % Load the Lateral LQR Design Model
Q= [2, 0.1, 2, 0.2, 0.1]; Q=diag(Q);        % State Weights: [phi,p,psi,r,beta]
R= [1, 1]*0.9; R=diag(R);                   % Control Weights
[Kpr, s, e] = LQR(Ad,Bd,Q,R)                % Lateral State-Feedback Matrix
% Kpr=[1.275, 1.595, -0.777, 0.334, -0.0755
%      0.685, 0.335,  1.272, 3.083, -0.5836];
save Kpr.mat Kpr -ascii
```

2.3.3 First Stage “Time-Slice” Control Analysis

Before exporting the state-feedback gains in the 6-dof simulation we must analyze the vehicle performance in a simple rigid-body simulation and find out, how efficiently is the vehicle responding to the pitch and lateral guidance commands? This is a “time-slice” analysis at a fixed flight condition. The Simulink file is “*Sim.Mdl*” in the subdirectory “*Analysis*”, shown in Figure 2.5, is used to evaluate the control design. It helps us adjust the weight matrices (Q and R) in the LQR process in order to achieve an acceptable response time, engine deflection amplitudes, and a satisfactory steady-state error. It uses the vehicle model from file “*vehicle_rb.m*”, the mixing logic matrix Kmix, and the LQR derived state-feedback gains Kq and Kpr.

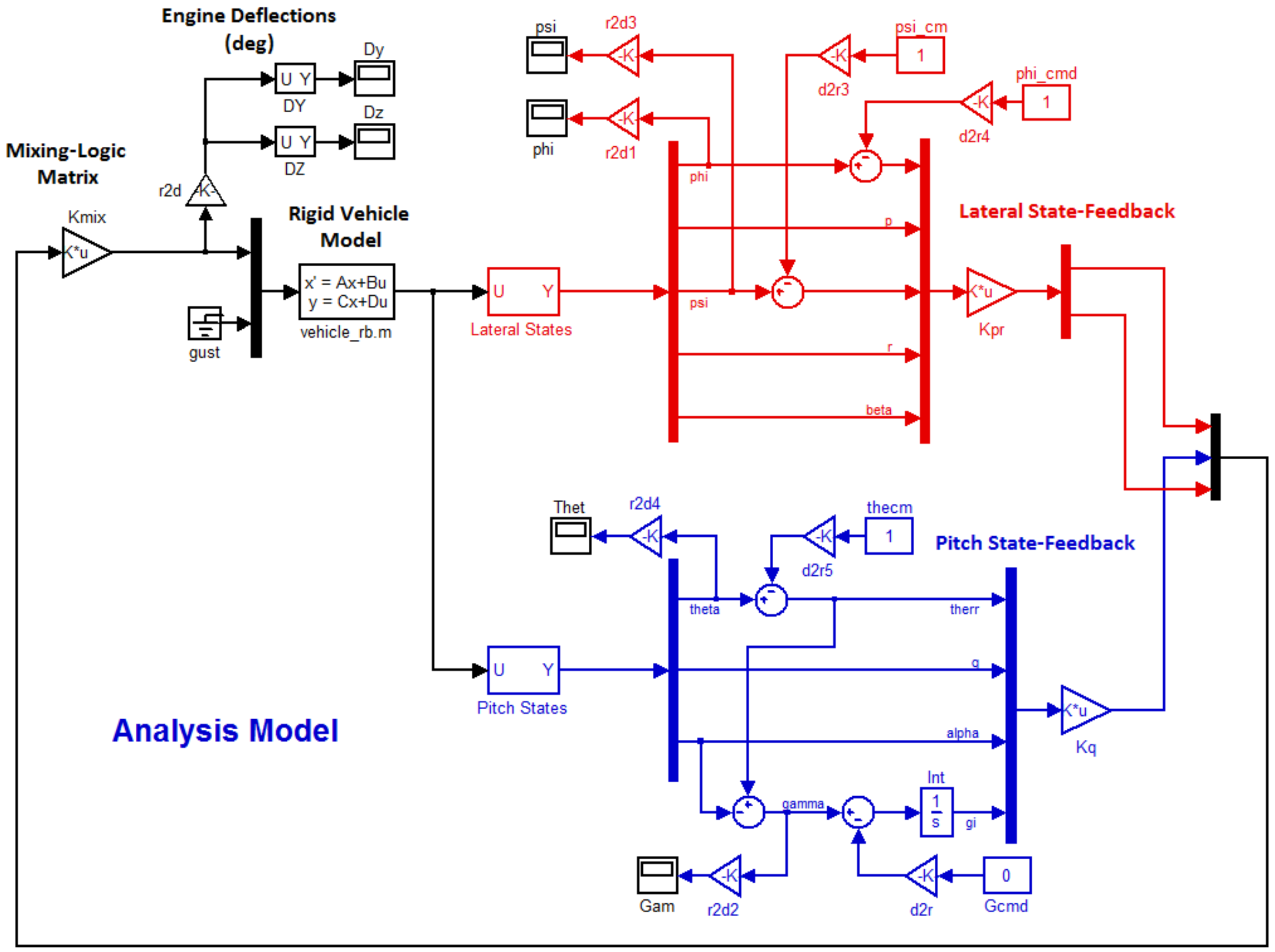


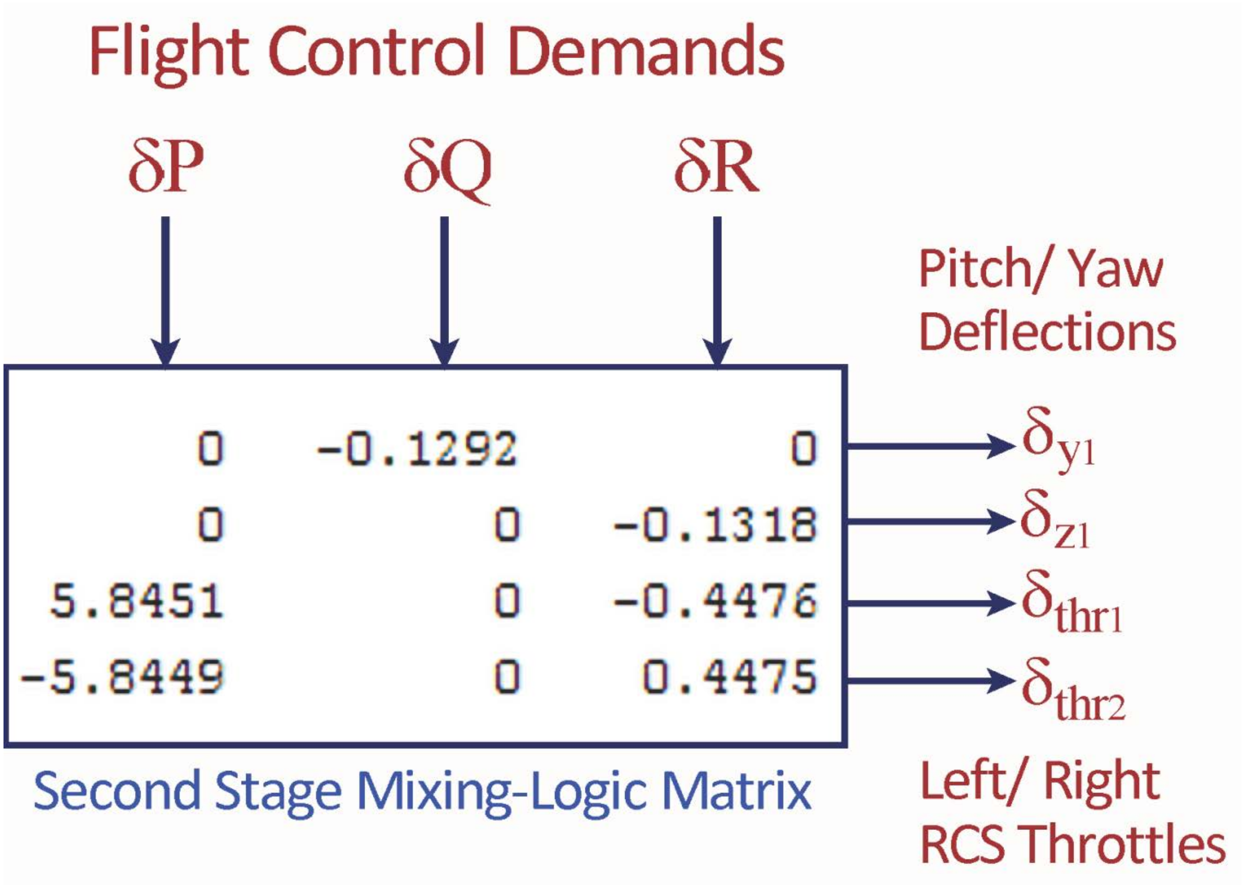
Figure 2.5 Simulink Model "*Sim.Mdl*" used for analyzing the System's Response to Commands during First Stage

Notice that in the first five seconds after ignition (before t=10 sec), and while the dynamic pressure is still low, our design requirement is not to track γ_{com} but to control only the pitch attitude command θ_{com}. The pitch design model in this flight condition, therefore, is simplified to include only states: (θ and q) and the state-feedback vector consists of gains only from those two states. The gains from (α and γ-integral) are set to zero.

2.4 Simple "Time-Slice" Design during Second Stage

The vehicle dynamic model used during second stage is different. It has only one TVC engine for pitch and yaw control. The two side boosters are ejected and only the middle engine is firing. For roll control it uses a pair of back-to-back RCS jets that fire in the ±Z direction. The mixing logic is a (4x3) matrix that blends the engine with the two jets for 3-axes control. The 3 inputs are roll, pitch, and yaw demands from the flight control system and the outputs are: pitch and yaw TVC engine deflections ($\delta y1$ and $\delta z1$) and throttle commands for the two jets.

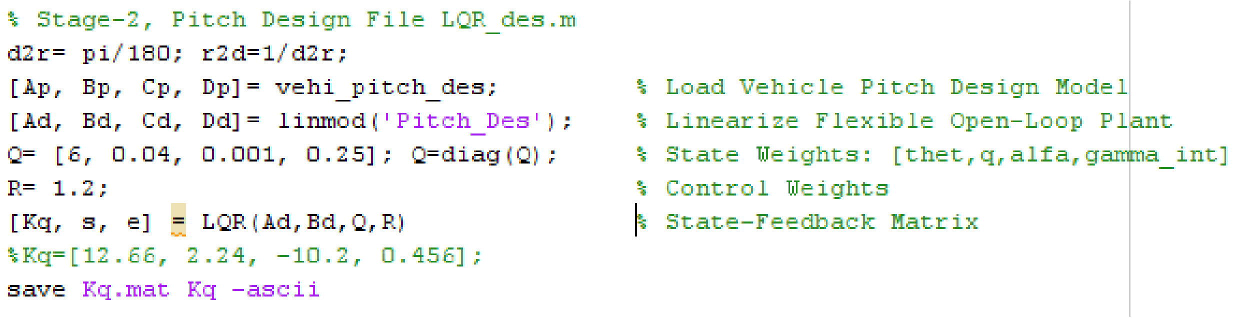
Note that the jet thrusts are included in the Flixan vehicle model and the system expects throttles instead of thrusts. The matrix outputs, therefore, must be throttles that do not exceed ±1 which correspond to ±800 (lb). The mixing logic matrix for the T100 case is shown below. A roll demand fires the two jets differentially. A +pitch demand deflects the TVC engine in the –pitch direction. A +yaw demand deflects the TVC engine in the –yaw direction and also fires the jets differentially to counteract the induced roll.



The vehicle input file: "*Zulu_T100.Inp*" for flight time=100 sec is in folder "*Twin Booster Rocket\2-FCS Design\T100*". After processing this file in Flixan, move the m-file: "*vehi_pitch_des.m*" to the pitch design subfolder "*Pitch_Mat*", and the m-file "*vehi_later_des.m*" to the lateral design subfolder "*Lateral_Mat*". The mixing-logic matrix Kmix and the vehicle m-file "*vehicle_rb.m*" must go in the analysis subfolder.

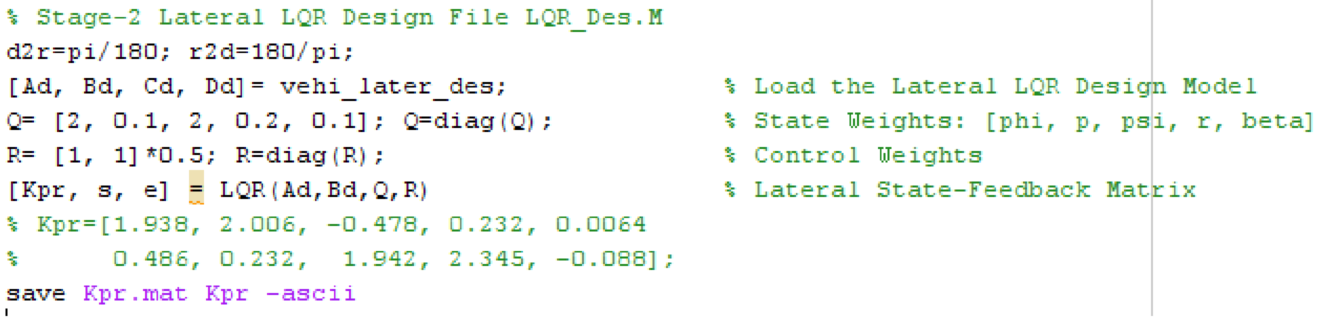
2.4.1 Second Stage Control Design

The pitch control gain is created in subdirectory "*Pitch_Mat*" by executing the Matlab file "*LQR_des.m*", shown below at time=100 seconds. The script loads the pitch design model which includes the Kmix matrix and uses the LQR function to generate the state-feedback matrix Kq.mat. The original design model consists of states: (θ, q, and α). An additional state γ-integral is included, where: γ=θ–α, to improve tracking of the flight-path angle γ.



```
% Stage-2, Pitch Design File LQR_des.m
d2r= pi/180; r2d=1/d2r;
[Ap, Bp, Cp, Dp]= vehi_pitch_des;          % Load Vehicle Pitch Design Model
[Ad, Bd, Cd, Dd]= linmod('Pitch_Des');     % Linearize Flexible Open-Loop Plant
Q= [6, 0.04, 0.001, 0.25]; Q=diag(Q);      % State Weights: [thet,q,alfa,gamma_int]
R= 1.2;                                    % Control Weights
[Kq, s, e] = LQR(Ad,Bd,Q,R)                % State-Feedback Matrix
%Kq=[12.66, 2.24, -10.2, 0.456];
save Kq.mat Kq -ascii
```

The lateral LQR design is similar and the files are located in subdirectory "*Lateral_Mat*". The script "*LQR_des.m*" loads the lateral vehicle design model and uses the LQR function to generate the lateral state-feedback matrix Kpr.mat which calculates the roll and yaw control demands as a function of the lateral states: (ϕ, p, ψ, r, and β).



```
% Stage-2 Lateral LQR Design File LQR_Des.M
d2r=pi/180; r2d=180/pi;
[Ad, Bd, Cd, Dd]= vehi_later_des;             % Load the Lateral LQR Design Model
Q= [2, 0.1, 2, 0.2, 0.1]; Q=diag(Q);          % State Weights: [phi, p, psi, r, beta]
R= [1, 1]*0.5; R=diag(R);                     % Control Weights
[Kpr, s, e] = LQR(Ad,Bd,Q,R)                  % Lateral State-Feedback Matrix
% Kpr=[1.938, 2.006, -0.478, 0.232, 0.0064
%       0.486, 0.232,  1.942, 2.345, -0.088];
save Kpr.mat Kpr -ascii
```

2.4.2 Simple "Time-Slice" Analysis during Second Stage

We must also analyze the vehicle performance in a simple rigid-body simulation to check the system's response to the pitch and lateral guidance commands. The differences between the first and second stage analysis models are in the TVC, RCS thrusters, and the mixing-logic matrix. Otherwise, the overall process is very similar. We use the Simulink file "*Sim.Mdl*" in subdirectory "*Twin Booster Rocket\2-FCS Design\T100\Analysis*", shown in Figure 2.6, to analyze the system's response to guidance step commands at t=100 sec. It includes the vehicle model from file "*vehicle_rb.m*", the mixing logic matrix Kmix, and the pitch and lateral state-feedback gains Kq and Kpr respectively. This simulation helps us adjust the weight matrices (Q and R) in the LQR design process in order to achieve an acceptable response time, TVC deflection and thruster magnitudes, and a satisfactory steady-state error.

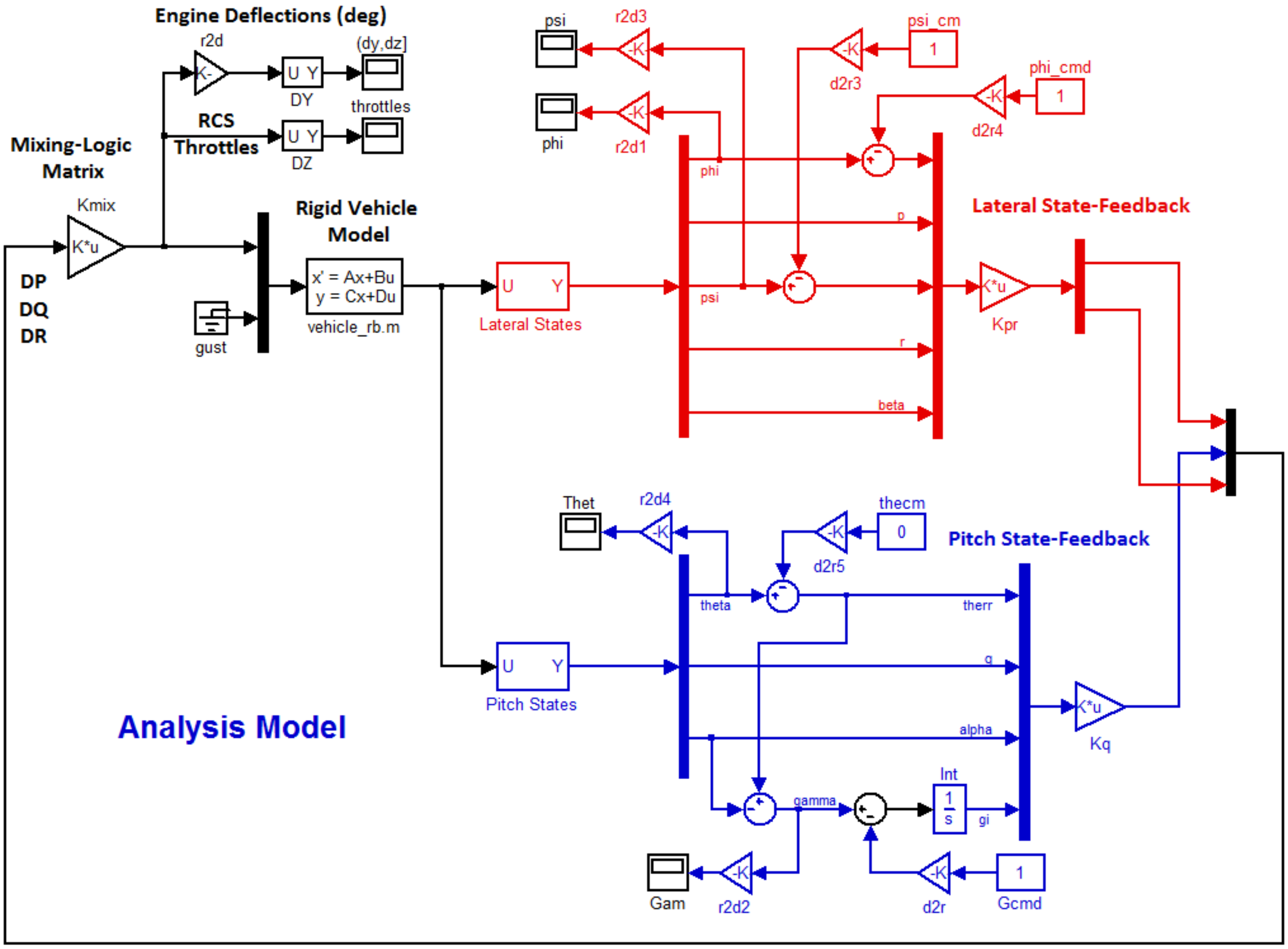
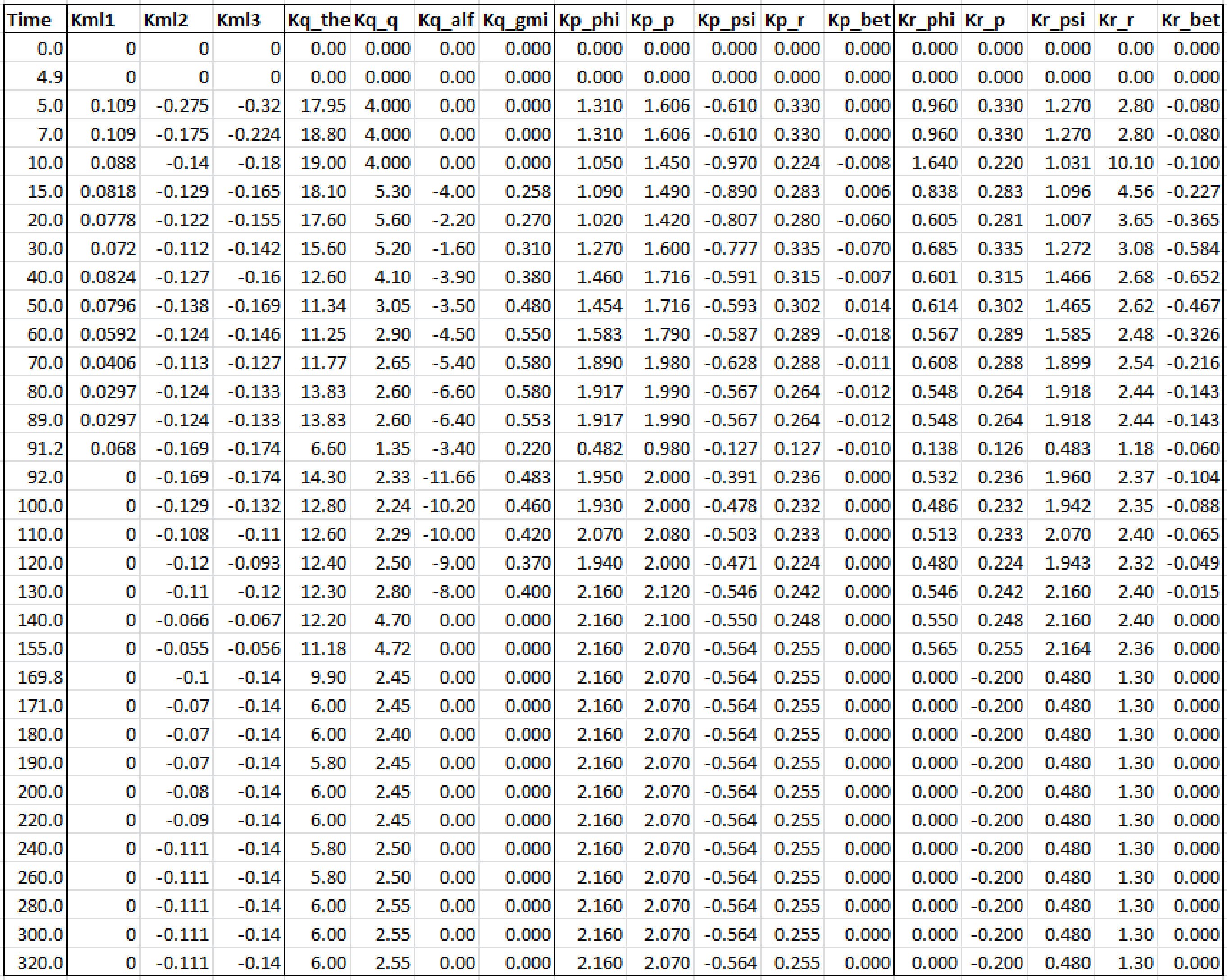


Figure 2.6 Simulink Model "Sim.Mdl" for analyzing the System's Response to Commands during Second Stage

2.5 Creating a Look-Up Table in Excel

When the design at each flight condition is complete, the gains for each time-slice are placed in an Excel table from where they will be exported into the Matlab simulation. The first three columns after time are the essential roll, pitch, and yaw gains from the mixing-logic matrix. Then we have 4 columns for the pitch state-feedback gains, 5 columns for roll, and 5 columns for the yaw state-feedback gains. The roll control using the reaction control thrusters in the second and third stages was implemented with fixed gains because they do not vary as much.



Time	Kml1	Kml2	Kml3	Kq_the	Kq_q	Kq_alf	Kq_gmi	Kp_phi	Kp_p	Kp_psi	Kp_r	Kp_bet	Kr_phi	Kr_p	Kr_psi	Kr_r	Kr_bet
0.0	0	0	0	0.00	0.000	0.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00	0.000
4.9	0	0	0	0.00	0.000	0.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.00	0.000
5.0	0.109	-0.275	-0.32	17.95	4.000	0.00	0.000	1.310	1.606	-0.610	0.330	0.000	0.960	0.330	1.270	2.80	-0.080
7.0	0.109	-0.175	-0.224	18.80	4.000	0.00	0.000	1.310	1.606	-0.610	0.330	0.000	0.960	0.330	1.270	2.80	-0.080
10.0	0.088	-0.14	-0.18	19.00	4.000	0.00	0.000	1.050	1.450	-0.970	0.224	-0.008	1.640	0.220	1.031	10.10	-0.100
15.0	0.0818	-0.129	-0.165	18.10	5.30	-4.00	0.258	1.090	1.490	-0.890	0.283	0.006	0.838	0.283	1.096	4.56	-0.227
20.0	0.0778	-0.122	-0.155	17.60	5.60	-2.20	0.270	1.020	1.420	-0.807	0.280	-0.060	0.605	0.281	1.007	3.65	-0.365
30.0	0.072	-0.112	-0.142	15.60	5.20	-1.60	0.310	1.270	1.600	-0.777	0.335	-0.070	0.685	0.335	1.272	3.08	-0.584
40.0	0.0824	-0.127	-0.16	12.60	4.10	-3.90	0.380	1.460	1.716	-0.591	0.315	-0.007	0.601	0.315	1.466	2.68	-0.652
50.0	0.0796	-0.138	-0.169	11.34	3.05	-3.50	0.480	1.454	1.716	-0.593	0.302	0.014	0.614	0.302	1.465	2.62	-0.467
60.0	0.0592	-0.124	-0.146	11.25	2.90	-4.50	0.550	1.583	1.790	-0.587	0.289	-0.018	0.567	0.289	1.585	2.48	-0.326
70.0	0.0406	-0.113	-0.127	11.77	2.65	-5.40	0.580	1.890	1.980	-0.628	0.288	-0.011	0.608	0.288	1.899	2.54	-0.216
80.0	0.0297	-0.124	-0.133	13.83	2.60	-6.60	0.580	1.917	1.990	-0.567	0.264	-0.012	0.548	0.264	1.918	2.44	-0.143
89.0	0.0297	-0.124	-0.133	13.83	2.60	-6.40	0.553	1.917	1.990	-0.567	0.264	-0.012	0.548	0.264	1.918	2.44	-0.143
91.2	0.068	-0.169	-0.174	6.60	1.35	-3.40	0.220	0.482	0.980	-0.127	0.127	-0.010	0.138	0.126	0.483	1.18	-0.060
92.0	0	-0.169	-0.174	14.30	2.33	-11.66	0.483	1.950	2.000	-0.391	0.236	0.000	0.532	0.236	1.960	2.37	-0.104
100.0	0	-0.129	-0.132	12.80	2.24	-10.20	0.460	1.930	2.000	-0.478	0.232	0.000	0.486	0.232	1.942	2.35	-0.088
110.0	0	-0.108	-0.11	12.60	2.29	-10.00	0.420	2.070	2.080	-0.503	0.233	0.000	0.513	0.233	2.070	2.40	-0.065
120.0	0	-0.12	-0.093	12.40	2.50	-9.00	0.370	1.940	2.000	-0.471	0.224	0.000	0.480	0.224	1.943	2.32	-0.049
130.0	0	-0.11	-0.12	12.30	2.80	-8.00	0.400	2.160	2.120	-0.546	0.242	0.000	0.546	0.242	2.160	2.40	-0.015
140.0	0	-0.066	-0.067	12.20	4.70	0.00	0.000	2.160	2.100	-0.550	0.248	0.000	0.550	0.248	2.160	2.40	0.000
155.0	0	-0.055	-0.056	11.18	4.72	0.00	0.000	2.160	2.070	-0.564	0.255	0.000	0.565	0.255	2.164	2.36	0.000
169.8	0	-0.1	-0.14	9.90	2.45	0.00	0.000	2.160	2.070	-0.564	0.255	0.000	0.000	-0.200	0.480	1.30	0.000
171.0	0	-0.07	-0.14	6.00	2.45	0.00	0.000	2.160	2.070	-0.564	0.255	0.000	0.000	-0.200	0.480	1.30	0.000
180.0	0	-0.07	-0.14	6.00	2.40	0.00	0.000	2.160	2.070	-0.564	0.255	0.000	0.000	-0.200	0.480	1.30	0.000
190.0	0	-0.07	-0.14	5.80	2.45	0.00	0.000	2.160	2.070	-0.564	0.255	0.000	0.000	-0.200	0.480	1.30	0.000
200.0	0	-0.08	-0.14	6.00	2.45	0.00	0.000	2.160	2.070	-0.564	0.255	0.000	0.000	-0.200	0.480	1.30	0.000
220.0	0	-0.09	-0.14	6.00	2.45	0.00	0.000	2.160	2.070	-0.564	0.255	0.000	0.000	-0.200	0.480	1.30	0.000
240.0	0	-0.111	-0.14	5.80	2.50	0.00	0.000	2.160	2.070	-0.564	0.255	0.000	0.000	-0.200	0.480	1.30	0.000
260.0	0	-0.111	-0.14	5.80	2.50	0.00	0.000	2.160	2.070	-0.564	0.255	0.000	0.000	-0.200	0.480	1.30	0.000
280.0	0	-0.111	-0.14	6.00	2.55	0.00	0.000	2.160	2.070	-0.564	0.255	0.000	0.000	-0.200	0.480	1.30	0.000
300.0	0	-0.111	-0.14	6.00	2.55	0.00	0.000	2.160	2.070	-0.564	0.255	0.000	0.000	-0.200	0.480	1.30	0.000
320.0	0	-0.111	-0.14	6.00	2.55	0.00	0.000	2.160	2.070	-0.564	0.255	0.000	0.000	-0.200	0.480	1.30	0.000

3. Launch Vehicle 6-DOF Simulation

The Simulink model was developed in Matlab/ Simulink. It is in file “*Zulu_6dof_Sim.Mdl*” which is located in the subdirectory “*Flixan\Examples\Twin Booster Rocket\3-Zulu 6Dof Simulation*”. It is initialized by the following m-file “Init.m”, and the script “pl.m” is used to plot the data. The initialization file also loads the mass properties, thrusts, aero data, and gains versus time which are loaded into tables.

```
% Initialization for the Zulu Non-Linear 6-dof Simulation
clear all
d2r=pi/180; r2d=180/pi;
h0=43031; x0=0; a0=4.73*d2r;                           % Initial Altitude h0,Alpha
decim=10;                                              % Plots Resolution
gm0=45.034*d2r; V0=436.7;                              % Initial Gamma, speed: V0=436.7;
Euler_0=[0, 49.766, 0]*d2r;                            % Initial Attitude (rad)
pqr_0=[0, 0.0048,0];                                   % Initial Rates (rad/sec)
Uvw_0=[cos(a0),0,sin(a0)]*V0;                          % X,Y,Z Veloc Comp Body (ft/sec)
x_cg= -228.19; y_cg=0.0;  z_cg=0.005;                  % Init CG
cg=[x_cg, y_cg, z_cg];
x_ref=-228.19; y_ref=0; z_ref=0;                       % MRC
mrc=[x_ref, y_ref, z_ref];
accel=[-225, 0, 0];                                    % Accelerometer Location
Sref=47.308; ch=7.761; sp=7.761;
Nmch=16; Nalf=24; Nbet=3;
Tstg=91.2;                                             % First  Stage Separation Time
Tst2=170.0;                                            % Second Stage Separation Time

% Load the Commands from Trajectory
load Commands -ascii
timin= Commands(:,1);                                  % Time In
thecm= Commands(:,2);                                  % Theta Command (deg)
gamcm= Commands(:,3);                                  % Gamma Command (deg)
thrs1= Commands(:,4);                                  % Engine 1 Thrust (lb)
thrs2= Commands(:,5);                                  % Engine 2 Thrust (lb)
thrs3= Commands(:,6);                                  % Engine 3 Thrust (lb)
mssin= Commands(:,7);                                  % Vehicle Mass

ee=[0,0,0,0,0,0,0,0,0,1;
    0,0,0,0,0,0,0,0,1,0;
    0,0,0,0,0,0,0,1,0,0;
    0,0,0,0,0,0,1,0,0,0;
    0,0,0,0,0,1,0,0,0,0;
    0,0,0,0,1,0,0,0,0,0;
    0,0,0,1,0,0,0,0,0,0;
    0,0,1,0,0,0,0,0,0,0;
    0,1,0,0,0,0,0,0,0,0;
    1,0,0,0,0,0,0,0,0,0];
```

```
% Load the Mass Properties
load Zulu_Mass -ascii
msspr= ee*Zulu_Mass(:,1);                                % Vehi Mass from Mass Props
Ixxpr= ee*Zulu_Mass(:,2);                                % Ixx from Mass Props
Iyypr= ee*Zulu_Mass(:,3);                                % Iyy from Mass Props
Izzpr= ee*Zulu_Mass(:,4);                                % Izz from Mass Props
Ixypr= ee*Zulu_Mass(:,5);                                % Ixy from Mass Props
Ixzpr= ee*Zulu_Mass(:,6);                                % Ixz from Mass Props
Iyzpr= ee*Zulu_Mass(:,7);                                % Iyz from Mass Props
Xcgpr= ee*Zulu_Mass(:,8);                                % X_cg from Mass Props
Ycgpr= ee*Zulu_Mass(:,9);                                % Y_cg from Mass Props
Zcgpr= ee*Zulu_Mass(:,10);                               % Z_cg from Mass Props

% Engine Gimbals
tvc1=[-244.6, -8.7, 0.0];                                % Stage-1 TVC Locations
tvc2=[-244.6, +8.7, 0.0];
tvc3=[-244.6,  0.0, 0.0];                                % Stage-2 TVC Locations
tvc4=[-200.0, 0.0, 0.06];                                % Stage-3 TVC Locations
rcs1=[-120.0, -4.0, -0.0];                               % Left  RCS Location
rcs2=[-120.0, +4.0, -0.0];                               % Right RCS Location

% Load Aero Data for the Base Vehicle
[CX, CY, CZ, Cl, Cm, Cn] = Stage1_Aero_Coeff;
[CX2,CY2,CZ2,Cl2,Cm2,Cn2]= Stage2_Aero_Coeff;
Machs=[0.35, 0.6, 0.8, 0.85, 0.9, 0.95, 1.1,1.3,1.4,1.46,1.57,1.82,2.7,3.44,5,10];
Betas=[-5, 0, 5];
Alfas=[-20,-4,-2,0,2,4,6,8,10,15,20,25,30,35,40,50,60,70,80,90,100,120,140,160];
Alfa2=[-4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

% Load FCS Gains
load FCS_Gains -ascii
timgn  = FCS_Gains(:,1);                                  % Gains Time In
Kml1   = FCS_Gains(:,2);                                  % Roll Mixing Logic Gain
Kml2   = FCS_Gains(:,3);                                  % Ptch Mixing Logic Gain
Kml3   = FCS_Gains(:,4);                                  % Yaw  Mixing Logic Gain

Kq_the = FCS_Gains(:,5);                                  % Pitch Kq_theta
Kq_q   = FCS_Gains(:,6);                                  % Pitch Kq_q (pitch rate)
Kq_alf = FCS_Gains(:,7);                                  % Pitch Kq_alpha
Kq_gmi = FCS_Gains(:,8);                                  % Pitch Kq_gamma_integral

Kp_phi = FCS_Gains(:,9);                                  % Roll Kp_phi (roll attitude)
Kp_p   = FCS_Gains(:,10);                                 % Roll Kp_p (roll rate)
Kp_psi = FCS_Gains(:,11);                                 % Roll Kp_psi (yaw attitude)
Kp_r   = FCS_Gains(:,12);                                 % Roll Kp_r (yaw rate)
Kp_bet = FCS_Gains(:,13);                                 % Roll Kp_beta

Kr_phi = FCS_Gains(:,14);                                 % Yaw Kr_phi (roll attitude)
Kr_p   = FCS_Gains(:,15);                                 % Yaw Kr_p (roll rate)
Kr_psi = FCS_Gains(:,16);                                 % Yaw Kr_psi (yaw attitude)
Kr_r   = FCS_Gains(:,17);                                 % Yaw Kr_r (yaw rate)
Kr_bet = FCS_Gains(:,18);                                 % Yaw Kr_beta

% Load Estimator Gains
load Estim_Gains -ascii
timest = Estim_Gains(:,1);                                % Gains Time
Cz_alf = Estim_Gains(:,2);                                % Cz_alpha
Cz0    = Estim_Gains(:,3);                                %
Cy_bet = Estim_Gains(:,4);                                % Cy_beta
```

3.1. 6-DOF Simulation Model

Figures (3.1 to 3.20) show the Simulink blocks and their functions are described in detail. The simulation includes all three stages from aircraft release to orbit insertion, which takes 315 seconds. The engines and the flight controls are switched in and out and the mass-properties and gains are scheduled from look-up tables. The top level diagram in Figure 3.1 consists of the launch vehicle dynamics (green block), the environment models (cyan block), a purple block that blends the aerodynamics and calculates alpha, beta, the Mach number, dynamic pressure, etc. There is also a light blue block that includes the aerodynamic coefficients for calculating the aerodynamic forces and moments. The orange block includes the TVC engines, the reaction control thrusters and calculates the total forces and torques applied to the vehicle, including aero. The yellow blocks include the TVC and RCS flight controls. They calculate engine deflections and jet thrust variations as a function of commands and the vehicle motion. A lot of the interconnections are internal and they are not shown in this block diagram for simplicity. Some of the blocks are off-the-shelf items from the Simulink/ Aerospace blockset.

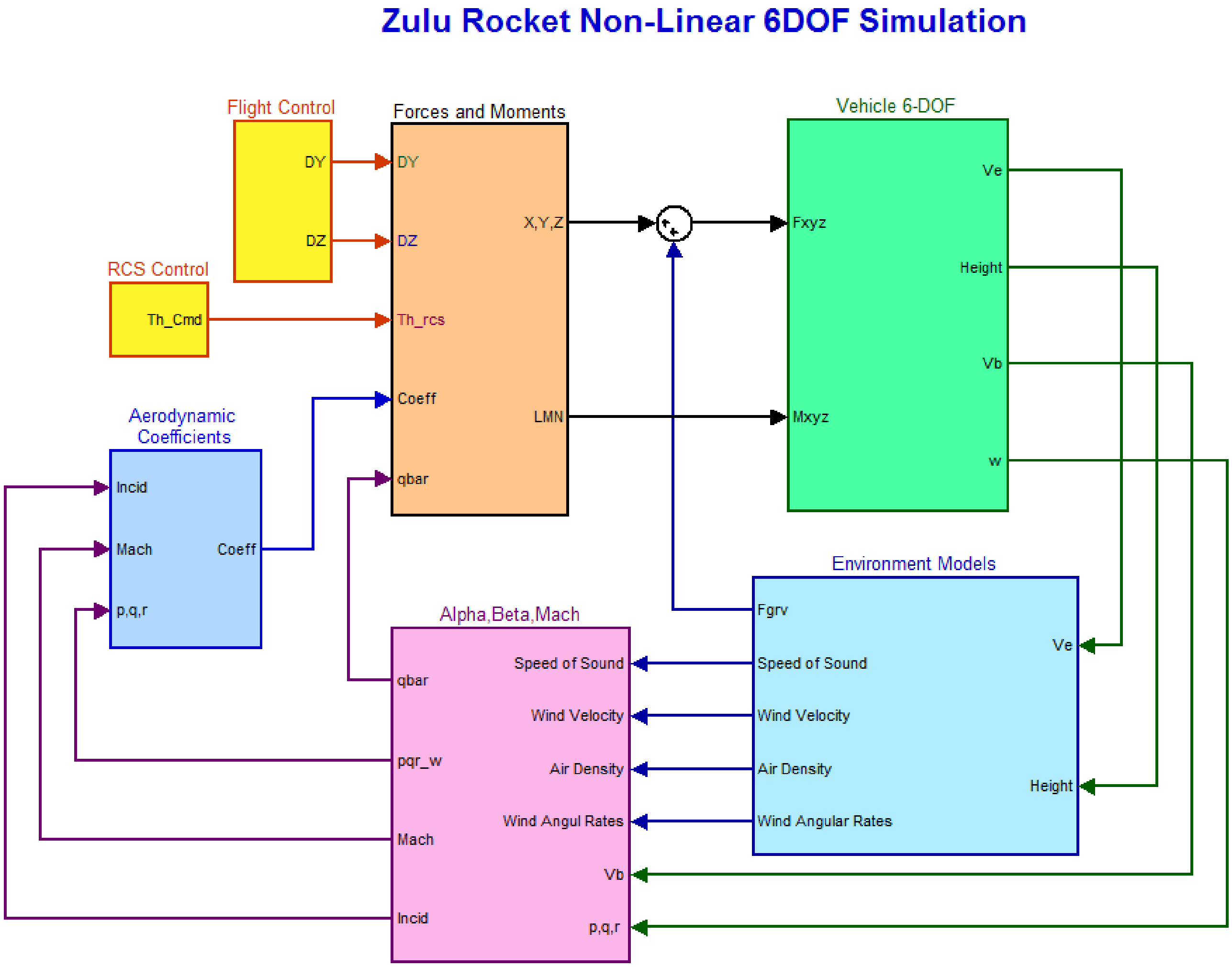


Figure 3.1 Top Level Block Diagram of the Simulink Model "*Zulu_6dof_Sim.Mdl*" of the Zulu Air-Launched Vehicle

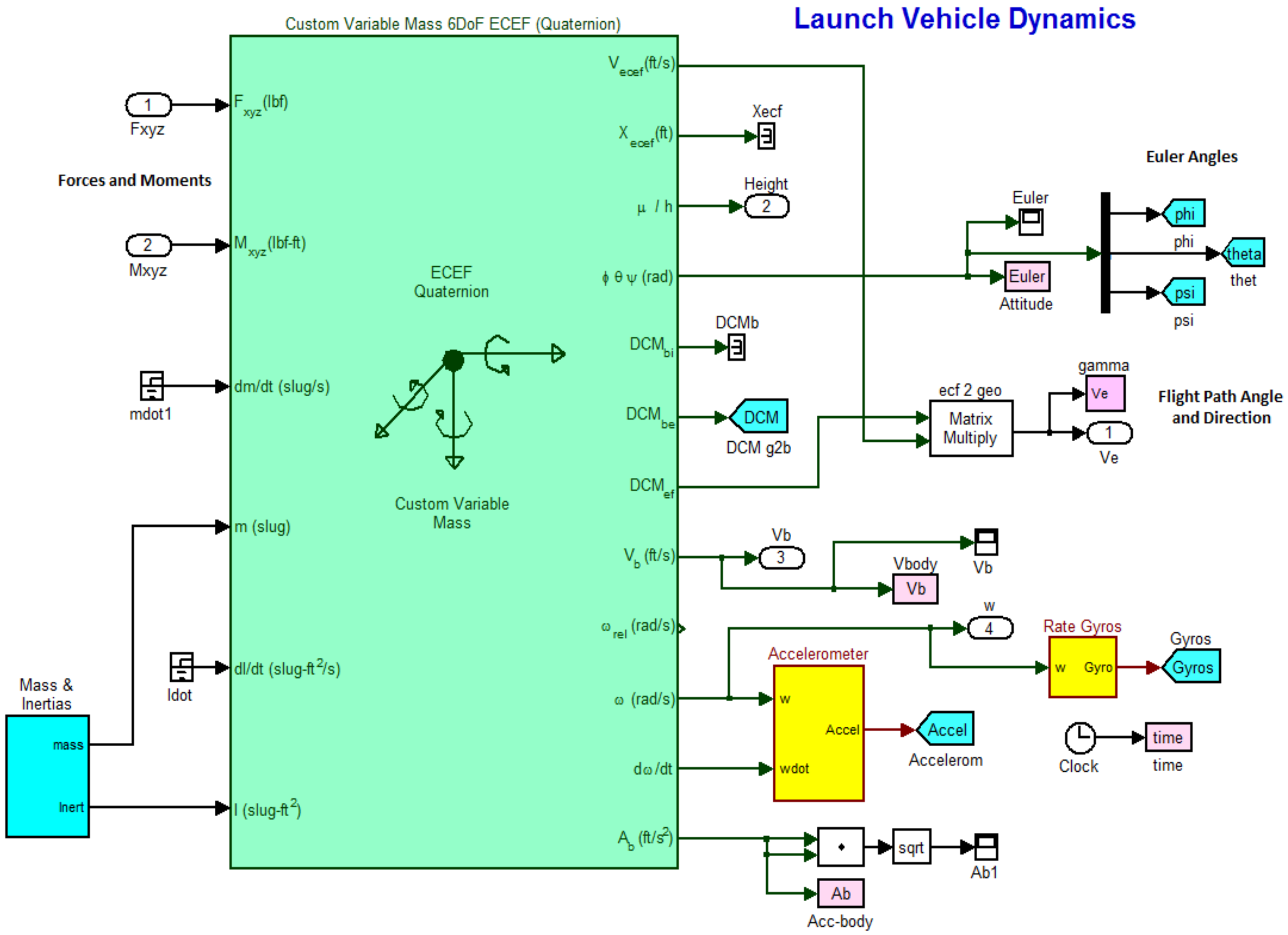


Figure 3.2 Vehicle Dynamics Block, Mass Properties Calculations, and Sensor Blocks

Figure 3.2 includes the launch vehicle dynamics which is excited by the external forces and torques. The standard Aerospace Blockset Simulink model “Custom Variable Mass 6DoF ECEF (Quaternion)” is used, which is based on an Earth-centered Earth-fixed (ECEF) coordinate frame (X_{ECEF}, Y_{ECEF}, Z_{ECEF}) about an Earth-centered inertial (ECI) reference frame (X_{ECI} , Y_{ECI} , Z_{ECI}). The origin of the ECEF coordinate frame is the center of the Earth. The applied forces [Fx Fy Fz] are in the body frame. The vehicle model is initialized at the release point, latitude, longitude, and vehicle altitude. Also at the release point velocity, Euler angles orientation, and body rates. In addition to the external forces and torques the model receives the vehicle mass and the moments of inertia tensor from the mass-properties block. Figure 3.3 shows the mass-properties block in detail. The mass is calculated as a function of time, and the moment of inertia tensor is calculated as a function of mass from the mass-properties look-up table which is loaded from file “*Zulu_Mass.Mat*”. The mass-properties subsystem also calculates the engine moment arms. That is the distances between the thrusters (TVC and RCS) and the vehicle CG.

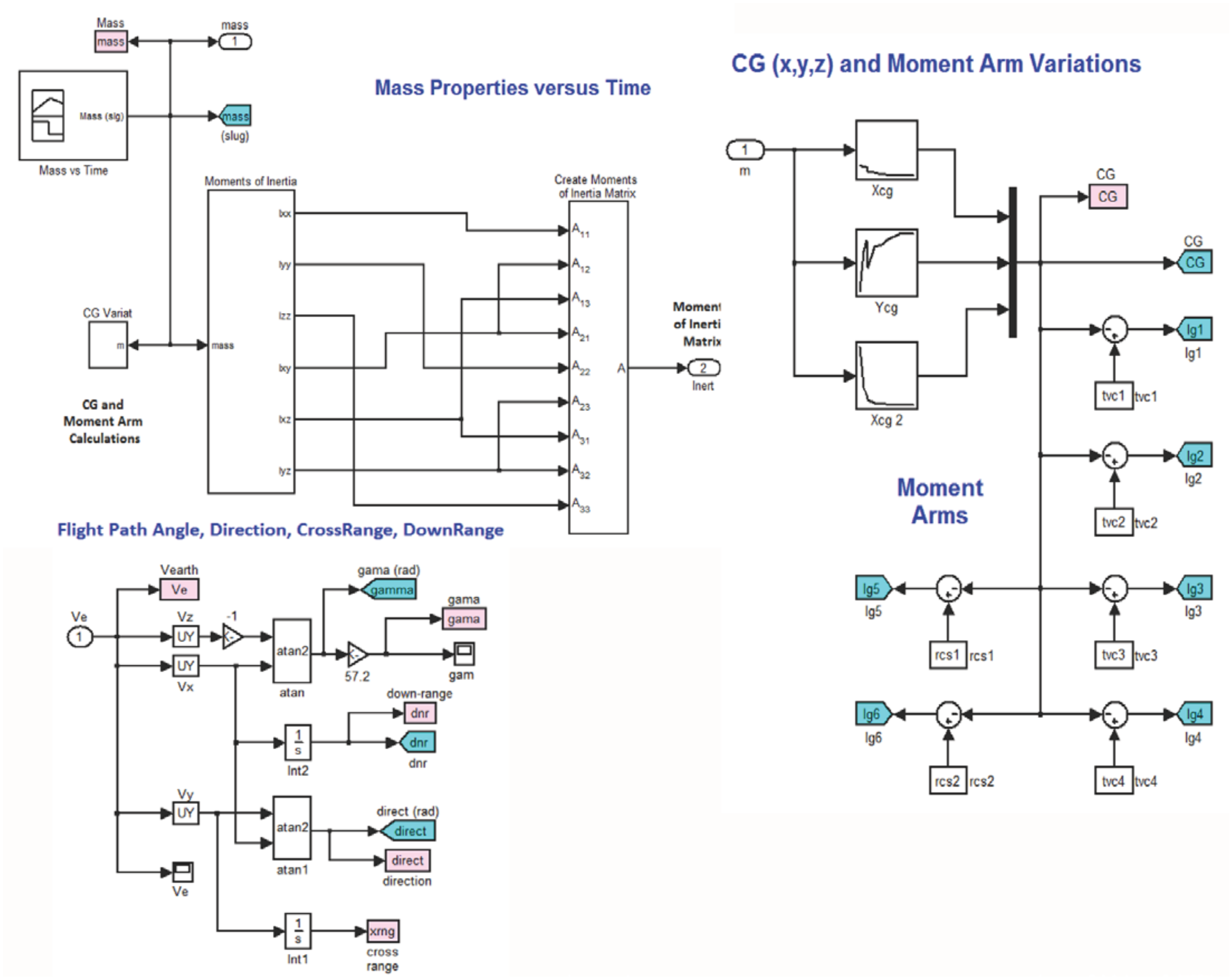


Figure 3.3 Mass-Properties Block that Calculates the Vehicle Mass and Moments of Inertia Tensor. It Includes a Subsystem that calculates the Flight-Path angle and Flight Direction.

The gamma and flight direction angles are calculated from the vehicle velocity in the geodetic axes. The direction cosine matrix (DCM_{ef}) transforms the velocity from the ECEF axes to the geodetic axes from where we calculate flight path and direction angles, and the downrange and crossrange distances from the release point.

The accelerometer model is shown in Figure 3.4. The accelerometer is not located at the CG and, therefore, in addition to vehicle acceleration at the CG it measures also contributions due to the vehicle rotational acceleration multiplied by its distance from the CG.

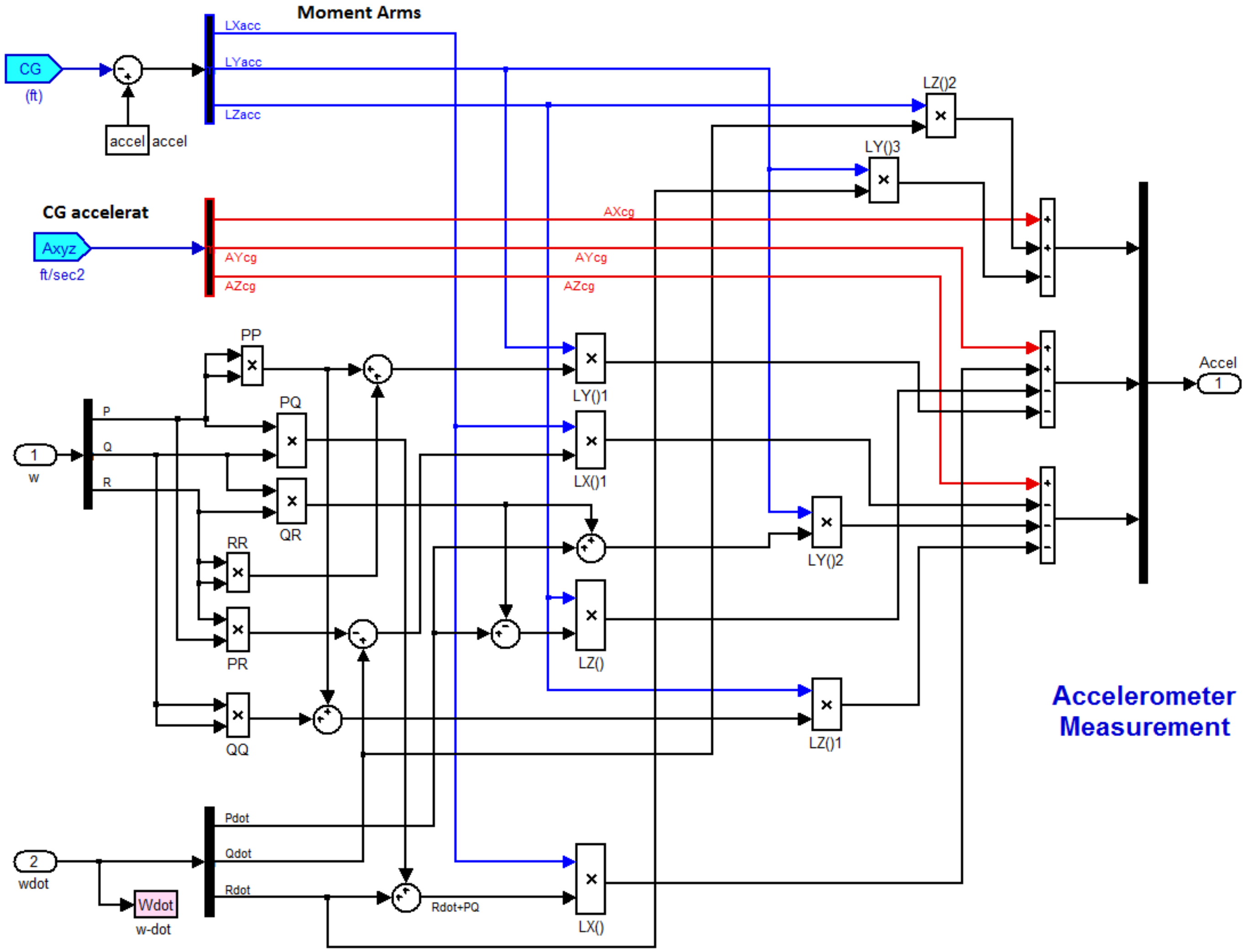


Figure 3.4 Accelerometer measurement consists of CG acceleration, plus contributions due to vehicle rotation

Figure 3.5 shows the Environment and Wind Disturbances subsystems. The atmospheric model calculates the air density and the speed of sound as a function of altitude. These parameters are needed in the calculations of Mach number, dynamic pressure, and aerodynamic forces. There is also a gravity model that calculates g as a function of altitude, longitude, and latitude. It calculates also the gravity forces in the body frame after converting them from the geo frame. The gravity forces are combined with the control forces and are applied to the vehicle. The wind disturbance model consists of three components: wind-shear as a function of altitude, turbulence in both velocity and rates as a function of altitude and speed, and wind-gust pulses. They are additional velocities and rates due to wind that combine with the vehicle velocity and rates and are used to calculate the aerodynamic forces on the vehicle.

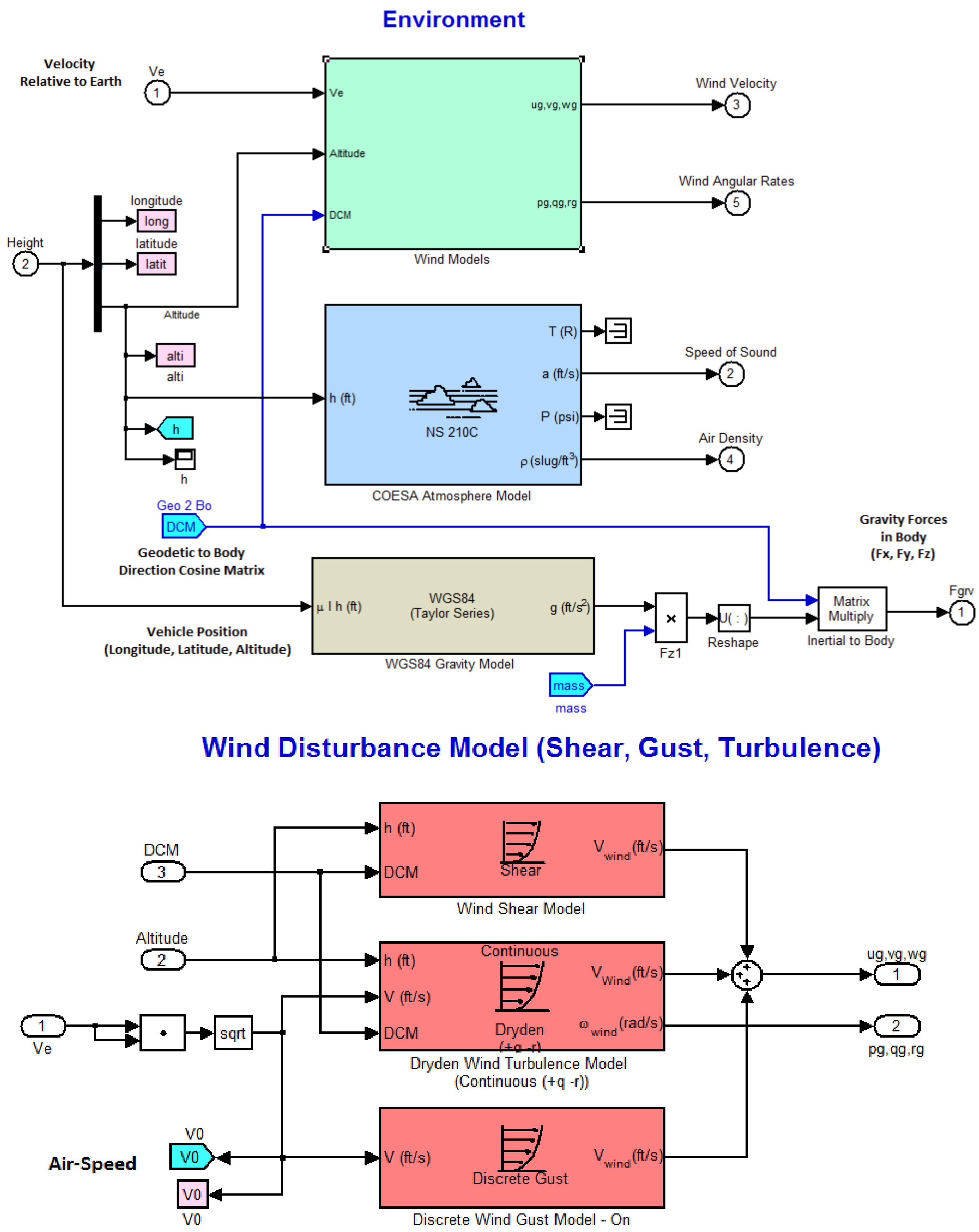


Figure 3.5 Environment and Wind Disturbances Subsystems

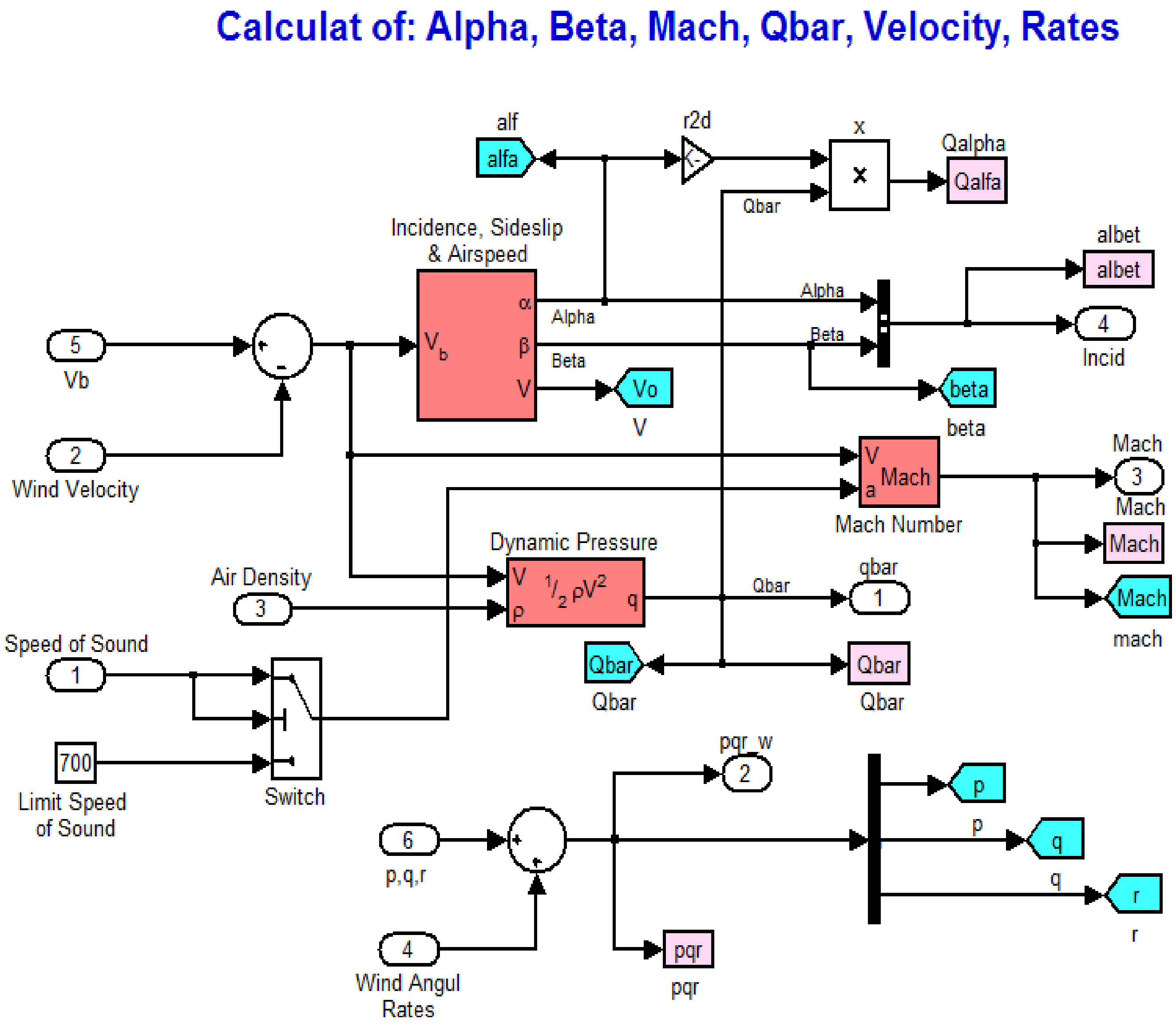


Figure 3.6 Aerodynamic Parameters Calculation

Figure 3.6 is the purple block in Figure 3.1. It calculates the angles of attack and sideslip and the vehicle speed relative to the wind. It also calculates the Mach number, the dynamic pressure Qbar, and Qalpha which is a parameter that measures the vehicle structural loading. It should typically be less than 3,500 (psf-deg). It also combines the vehicle rates with the wind angular rates from the turbulence model. These rates are combined with the damping derivatives to calculate the damping forces in the aerodynamic coefficients (light blue block in Fig 3.1).

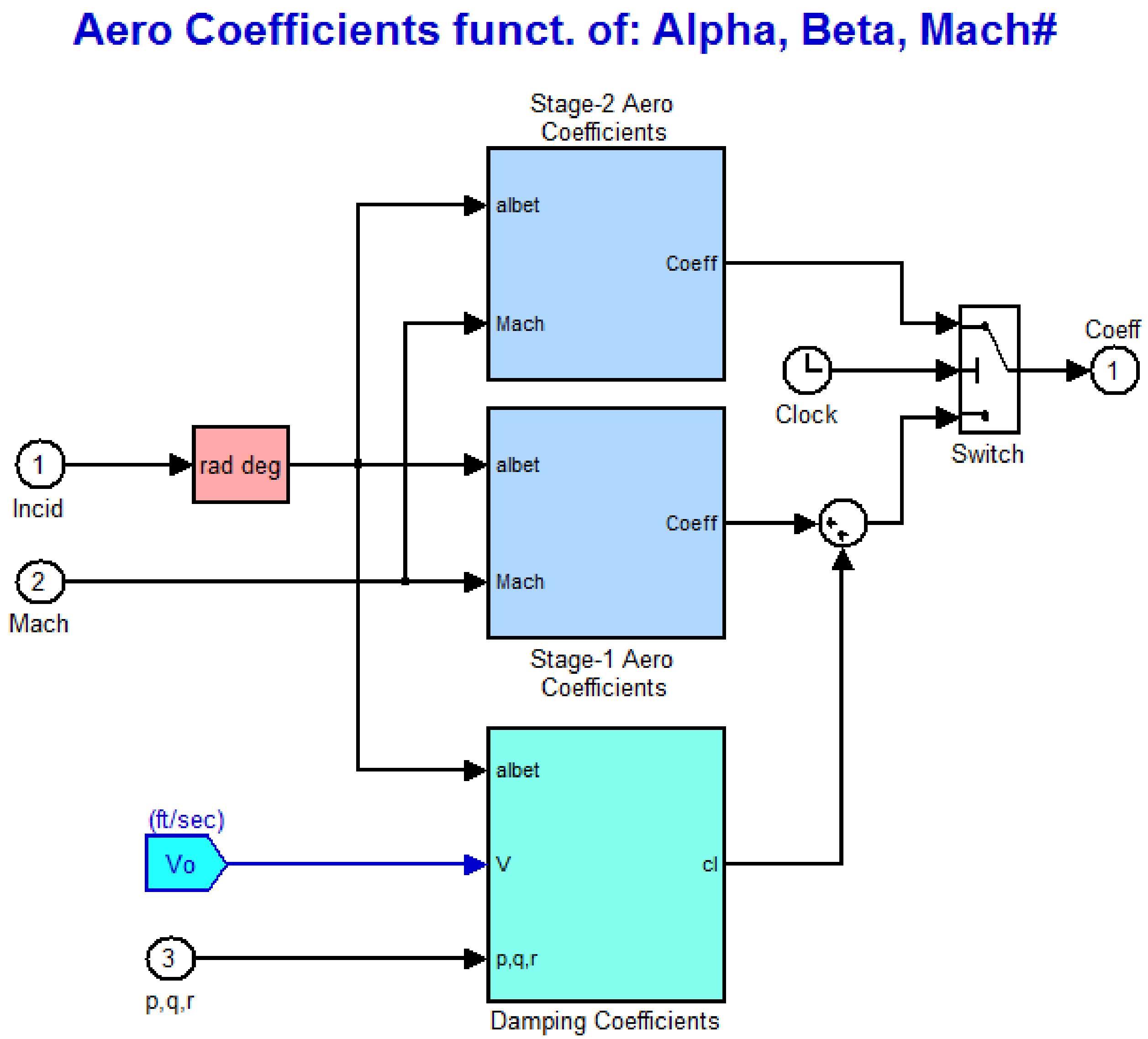


Figure 3.7 Aerodynamic Coefficients Block which Includes First and Second Stage Aero Coefficients Subsystems, and also First Stage Damping Derivatives. The switch changes the coefficients after staging which occurs 91.2 seconds after release.

The aerodynamic coefficients block in Figure 3.7 calculates the six aerodynamic coefficients (CA, CY, CZ, Cl, Cm, and Cn) that generate the aerodynamic forces and moments on the vehicle. The first and second stage coefficients are calculated from separate look-up tables and they are switched after first stage separation. There is also a block that includes the damping forces and moments due to vehicle rates p_w, q_w, and r_w relative to the wind.

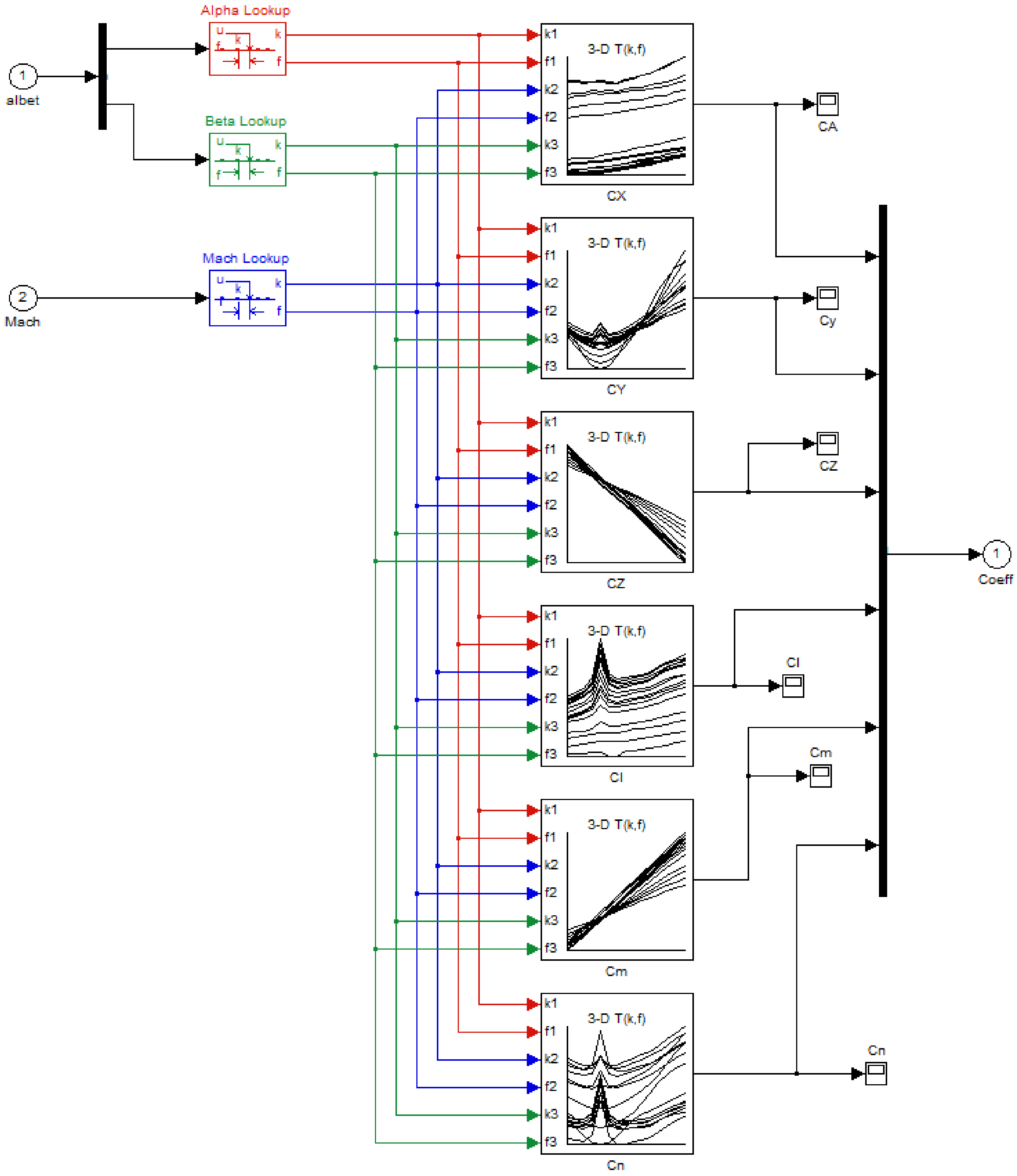


Figure 3.8 First Stage Aerodynamic Coefficients are Functions of Mach Number, Alpha, and Beta. They are loaded from file “Stage1_Aero_Coeff.m”. Second Stage Aerodynamic Coefficients are loaded from file “Stage2_Aero_Coeff.m”.

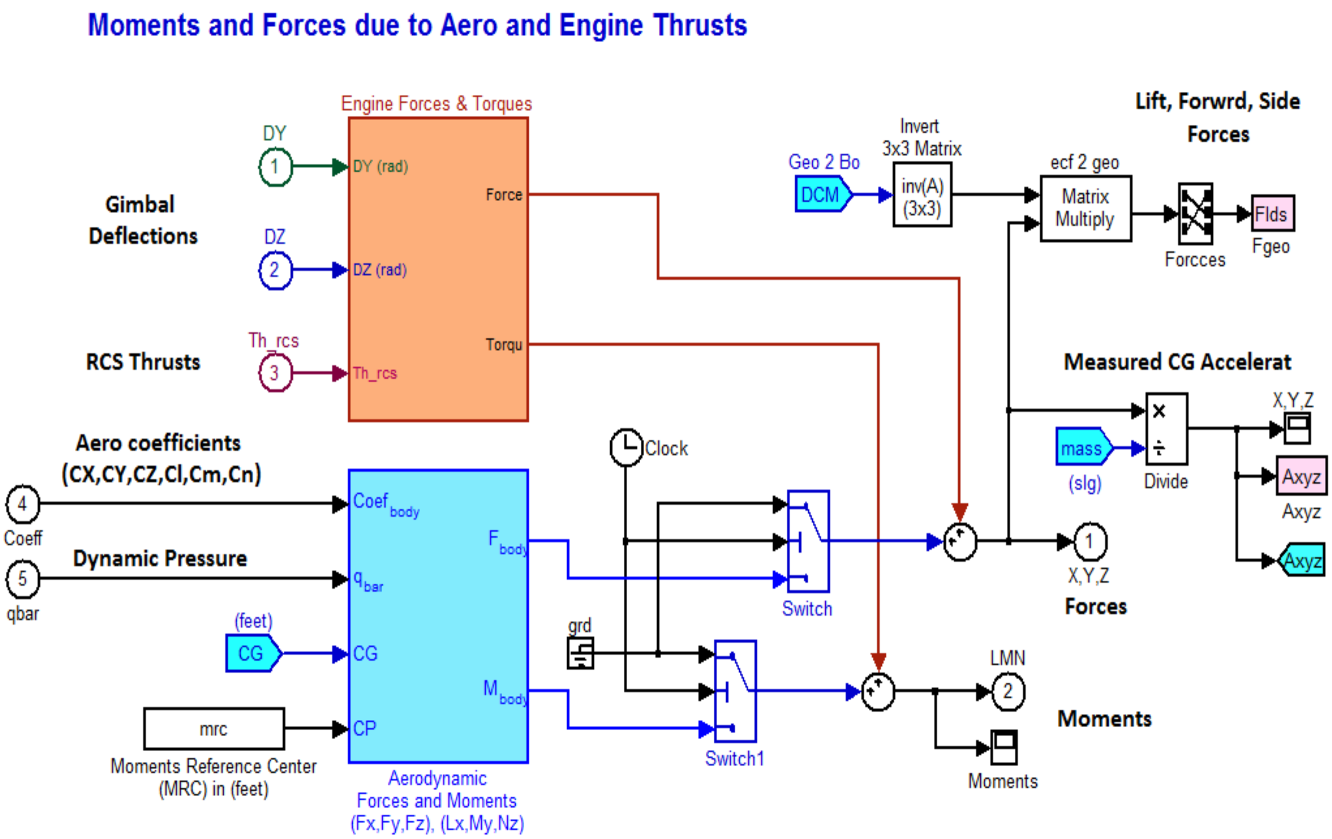


Figure 3.9 Aerodynamic and TVC Force and Moment Subsystems

Figure 3.8 shows the first stage aerodynamic coefficients interpolation subsystem. The 6 coefficients are loaded from file “Stage1_Aero_Coeff.m”. They are 3-dimensional arrays which are functions of: Mach, alpha, and beta. The second stage subsystem is very similar and the coefficients are loaded from file “Stage2_Aero_Coeff.m”.

The orange block in Figure 3.9 calculates the propulsion forces and torques in body axes. They are due to engine gimbaling and RCS thruster firing. This block is shown in detail in Figures (3.10 and 3.11). The second blue block below it calculates the aerodynamic forces and torques from the aero coefficients. They are also functions of the dynamic pressure, reference lengths, and reference area. It also requires the locations of the CG and the MRC. The total forces are divided by the vehicle mass to calculate the accelerations in body coordinates. The forces are also transformed from body to geo axes to calculate the vertical and horizontal force components.

Figure 3.10 shows the propulsion block in detail. The block receives the gimbal deflections and the RCS thrusts from the flight control blocks and calculates the propulsion forces and torques in the body axes. The TVC and RCS moment arms (lgi) are used to calculate the torques. The forces from all engines and jets are summed to calculate the total propulsion force. The individual engine torques are obtained by multiplying the forces with the engine moment arms. They are also combined to calculate the total propulsion torque. The engine thrusts for the first, second, and third stages are shown in Figure 3.12. They are generated from a look-up table as a function of time. Notice that engine #3 comprises of the thrust for both: second and third stages. Figure 3.11 shows the Figure 3.10 subsystems that generate the TVC forces as a function of engine thrust and deflection angles, and also the jet forces as a function of thrust variation and their fixed orientation angles.

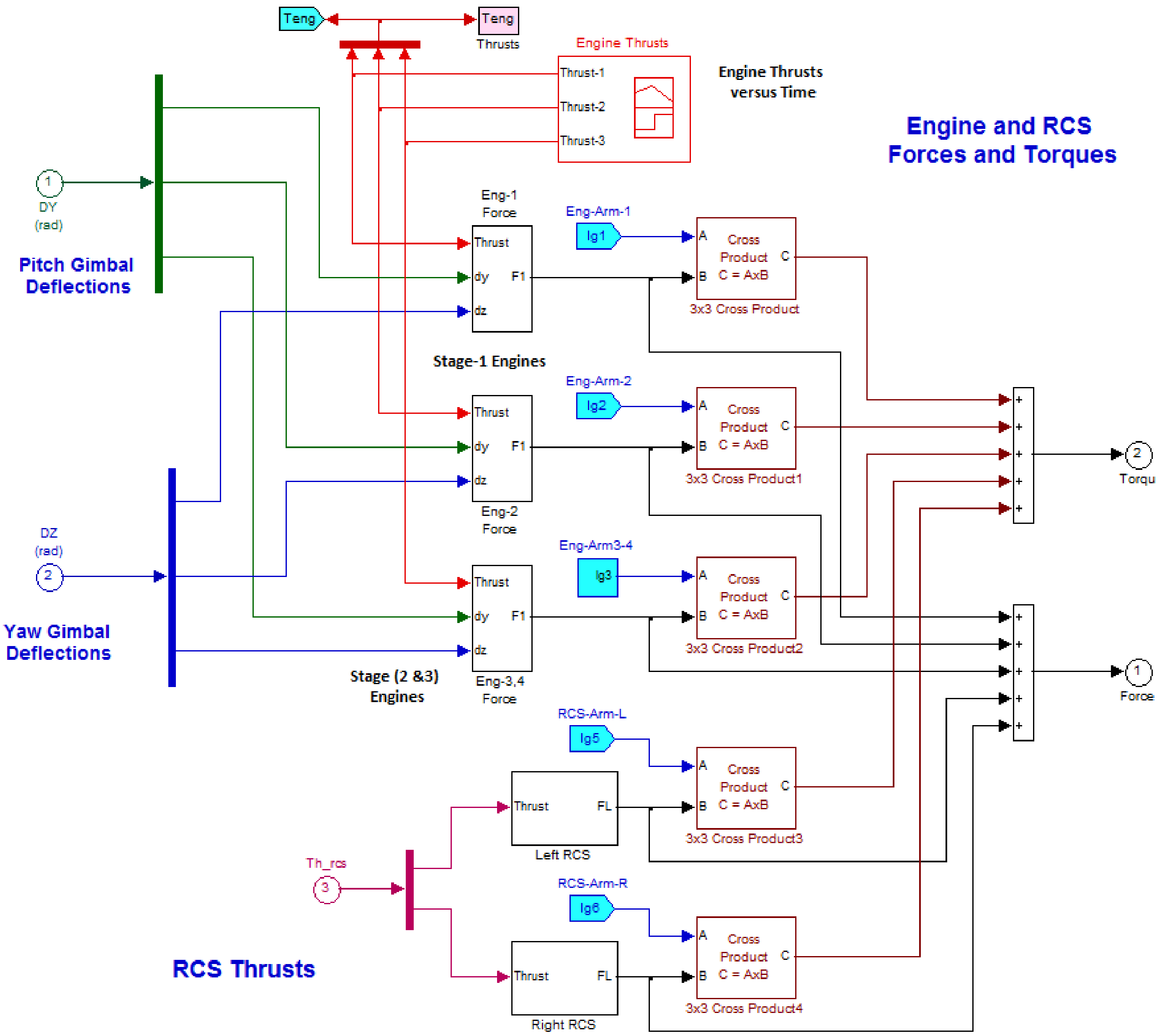


Figure 3.10 Engine Forces and Torques Subsystem

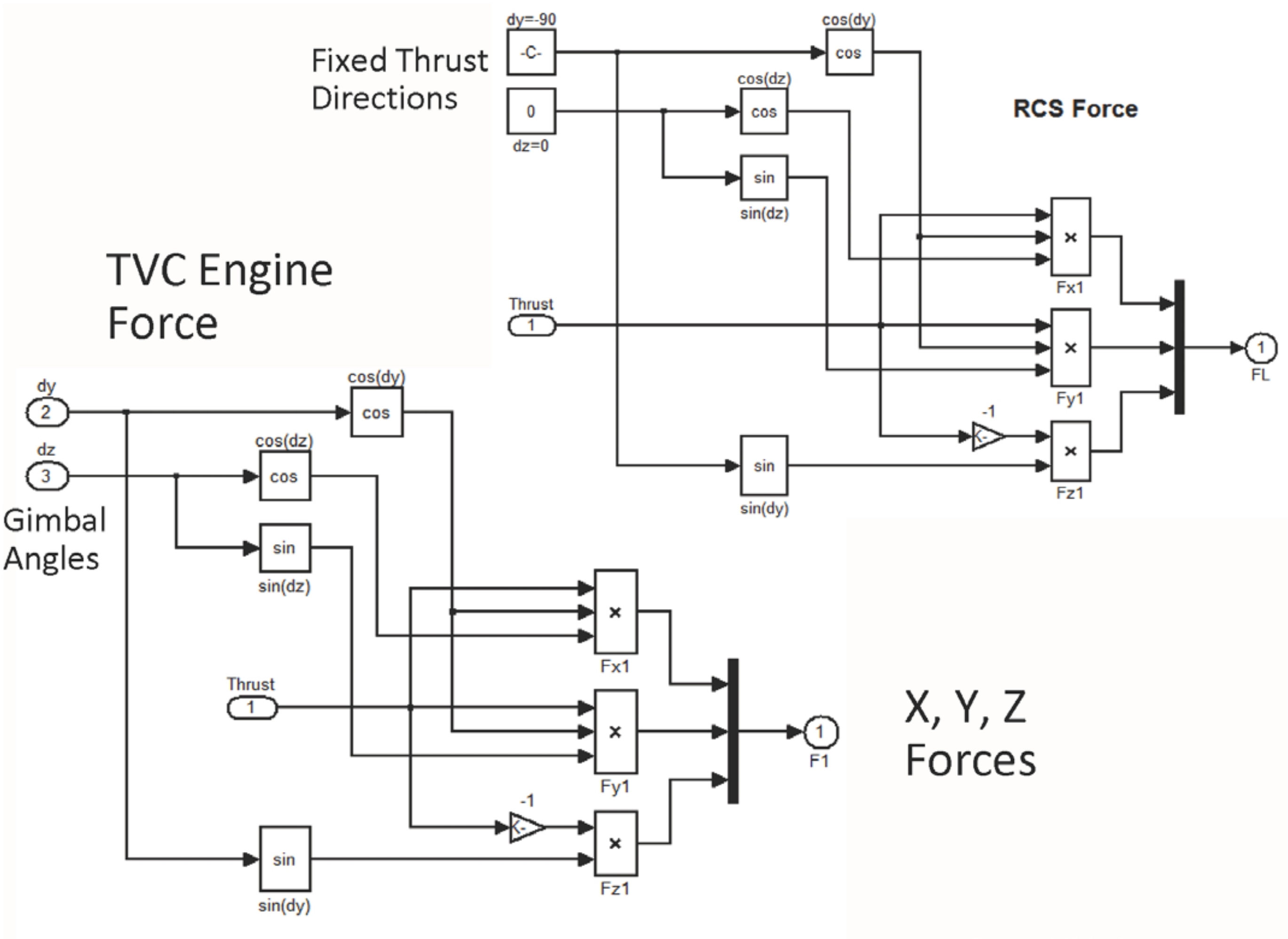


Figure 3.11 TVC and RCS X,Y, and Z Forces as a Function of Gimbal Angles and Thruster Orientation

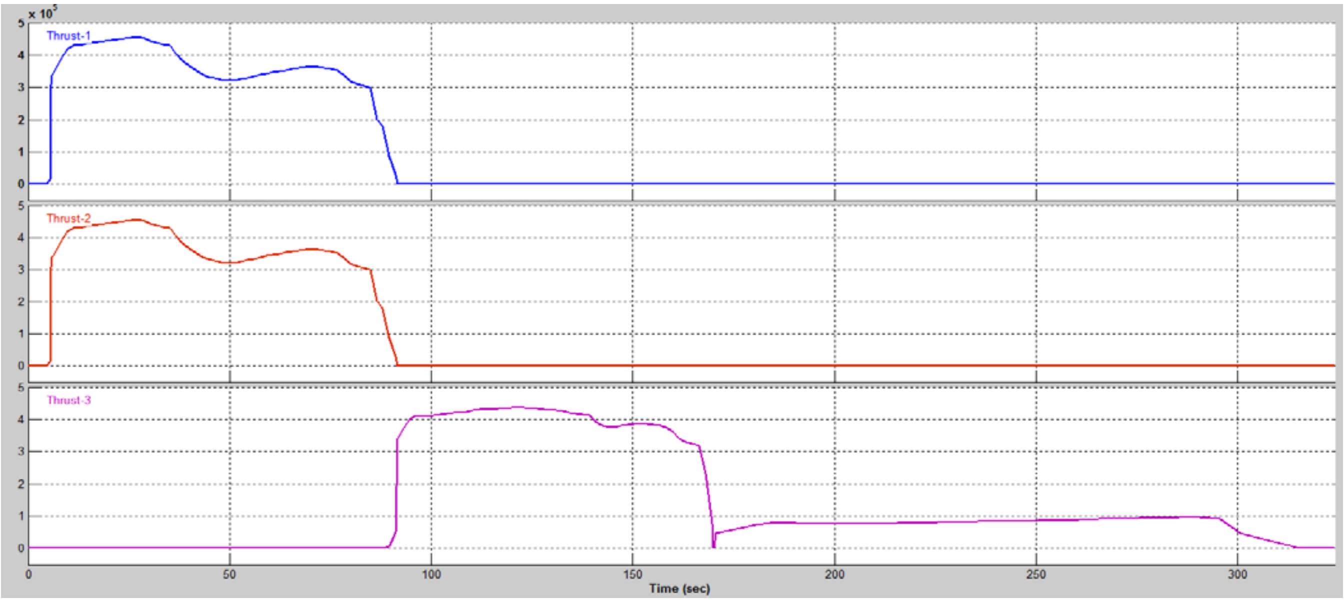


Figure 3.12 Engine Thrusts for the First, Second, and Third Stages, as a Function of Time

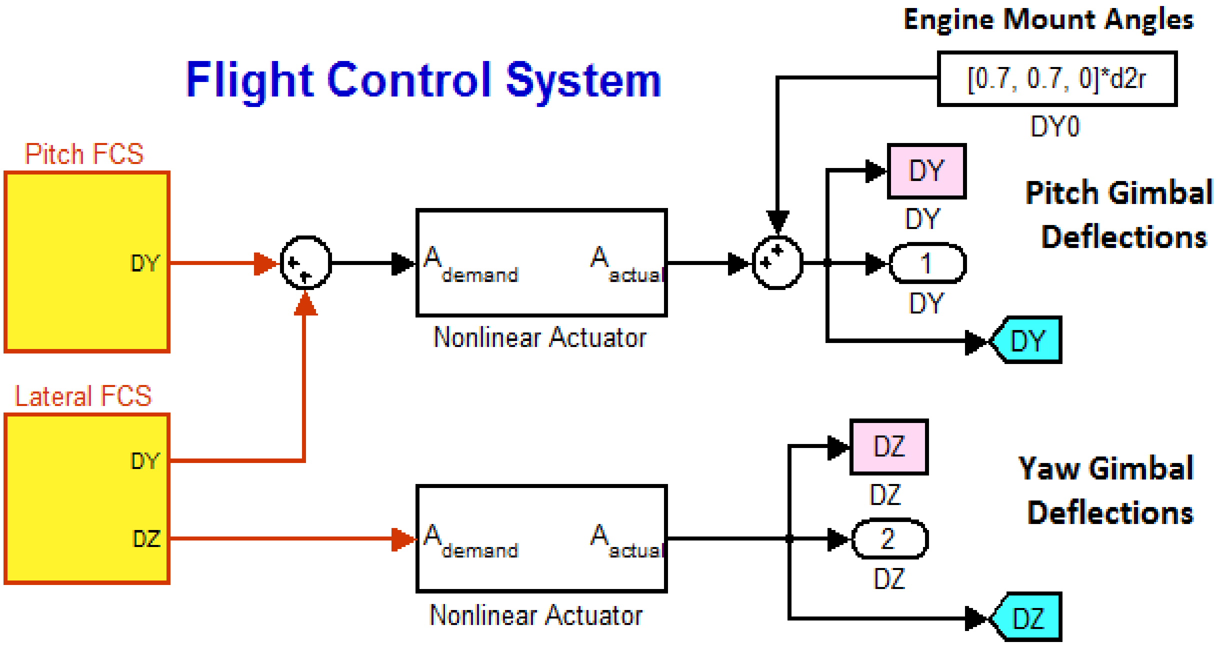


Figure 3.13 Pitch and Lateral Flight Control Systems for TVC, Including the TVC Actuators

Figure 3.13 shows the TVC flight control system from a top level observation. During first stage it consists of the Pitch FCS that deflects the engines only in pitch, and the Lateral FCS that deflects the engines in both: pitch and yaw directions. During second and third stages the FCS requires the RCS jets for roll control. Figure 3.13 also shows the non-linear TVC actuators. They have $\pm 5^o$ amplitude limits from the mounting positions in both pitch and yaw. They also have a 25 (deg/sec) rate limit. The engines are mounted with a bias angle of 0.7^o in pitch.

Figure 3.14 shows the longitudinal flight control system. It consists of feedback from the states (θ, q, α, and γ-integral). The gains were designed using linear Flixan models generated at fixed flight conditions, as it was described in the FCS design section. They are loaded from file "*FCS_Gains.Mat*", scheduled and interpolated from a look-up table as a function of time, see Figure 3.15. The guidance commands consist of coordinated pitch attitude and gamma commands (θ_{com} and γ_{com}) as a function of time which are loaded from file "*Commands.Mat*", see Figure 3.15. The gains are tuned to emphasize either attitude control or gamma control in different phases. Initially they emphasize pitch control in order to perform the pitch up maneuver, and also later towards the end of second and third stages where the dynamic pressure is almost zero. At other times they attempt to track both theta and gamma which are coordinated, otherwise it wouldn't be possible to track them independently. The gimbal deflections are a linear combination of states multiplied with the corresponding gains. It is assumed that most of the states are measurable, except alpha which is estimated from other variables, as we shall see in Figure 3.18.

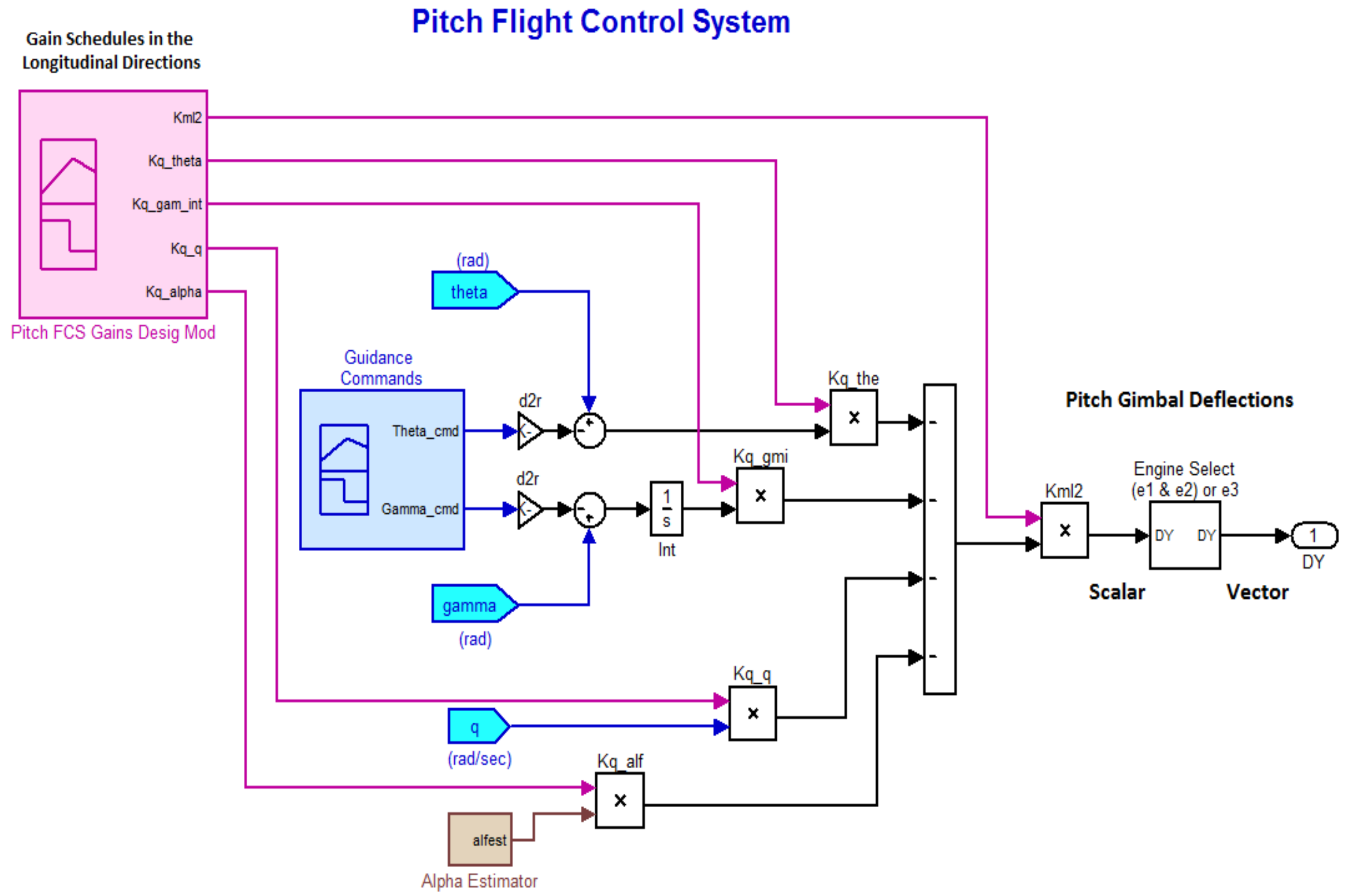


Figure 3.14 Pitch Flight Control System Includes State-Feedback, Guidance Commands, Gain Scheduling and an Angle of Attack Estimator

Figure 3.16 shows the lateral FCS, consisting of a (2x5) roll and yaw state-feedback matrix from (ϕ, p, ψ, r, and β). The gains are loaded from file "*FCS_Gains.Mat*" and they were designed using linear Flixan models generated at fixed flight conditions. They are scheduled and interpolated from look-up tables, shown in Figure 3.17.

It is assumed that most of the states are measurable, except for the angles of attack and sideslip. The angle of attack and sideslip estimators are shown in Figure 3.18. The estimators are filters that basically solve the normal and lateral force equations. Figure 3.18a implements the normal force equation which is a function of the aerodynamic force, the TVC force, and the vehicle mass times Z-acceleration, and solves for alpha. The look-up table consists of aerodynamic the parameters $Cz\alpha$ and Cz0 versus time which are loaded from file "*Estim_Gains.Mat*". They are needed for calculating the normal aero force as a function of Qbar. The TVC force is calculated from the gimbal angles and the engine thrusts. The mass times acceleration term also needs a correction from the pitch rate gyro because the accelerometer is not located at the CG and it measures also rotations. Similarly, Figure 3.18b implements the lateral force equation and solves for beta using a $Cy\beta$ coefficient look-up table.

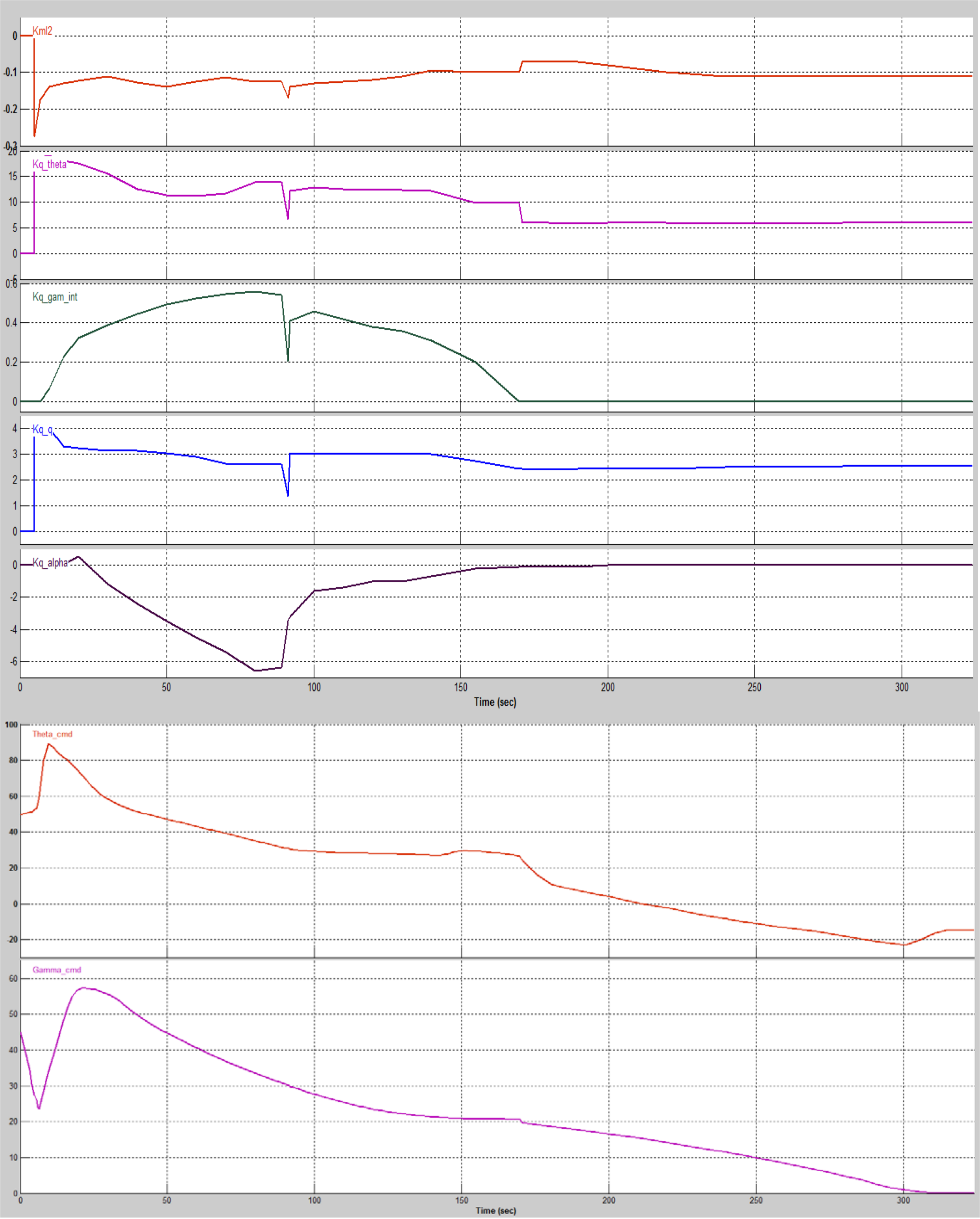


Figure 3.15 Pitch State-Feedback Gains Including Coordinated Guidance Commands: (θ_{com} and γ_{com})

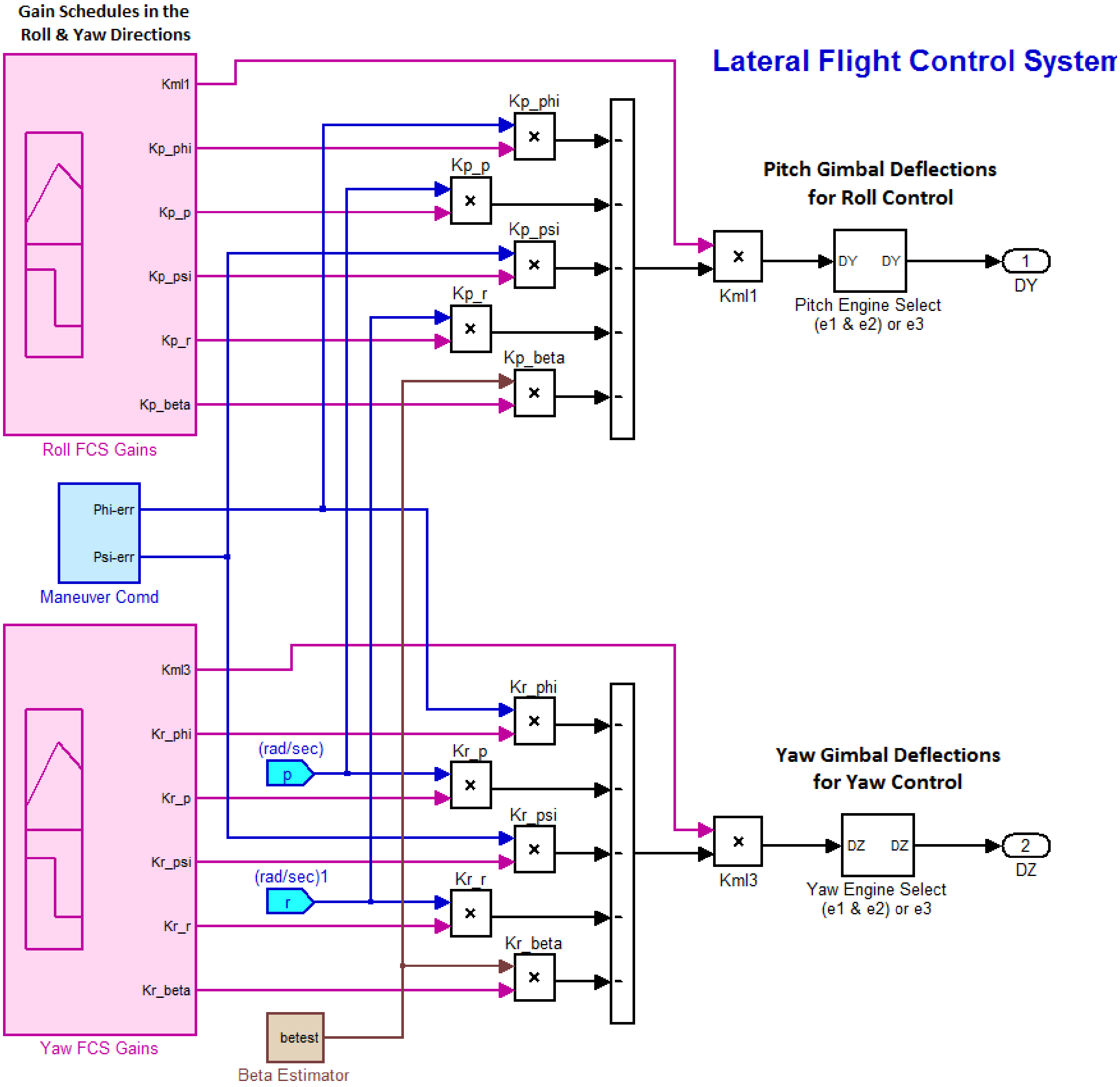


Figure 3.16 Lateral Flight Control System Includes a (2x5) State-Feedback Matrix, Directional Guidance, Gain Scheduling and an Angle of Sideslip Estimator

The lateral guidance subsystem (blue block) generates the roll and yaw commands and it is shown expanded in Figure 3.19. It performs small corrections in the flight direction to counteract wind-shear, aircraft initialization errors, etc. in order to place the satellite in the proper inclination angle. It corrects the vehicle direction by commanding small roll and yaw attitude adjustments as a function of directional error.

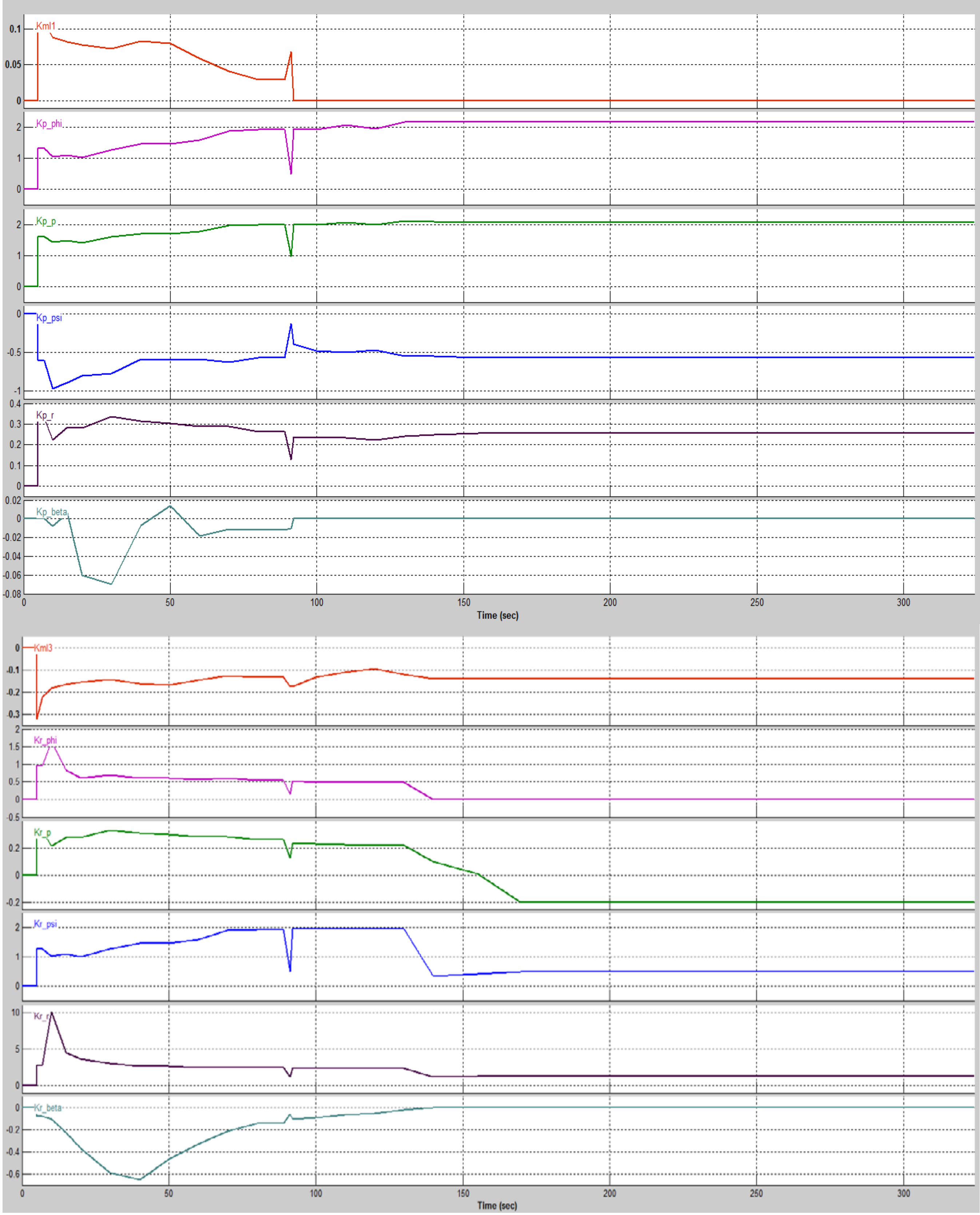


Figure 3.17 Roll and Yaw State-Feedback Gains Scheduled as a Function of Time

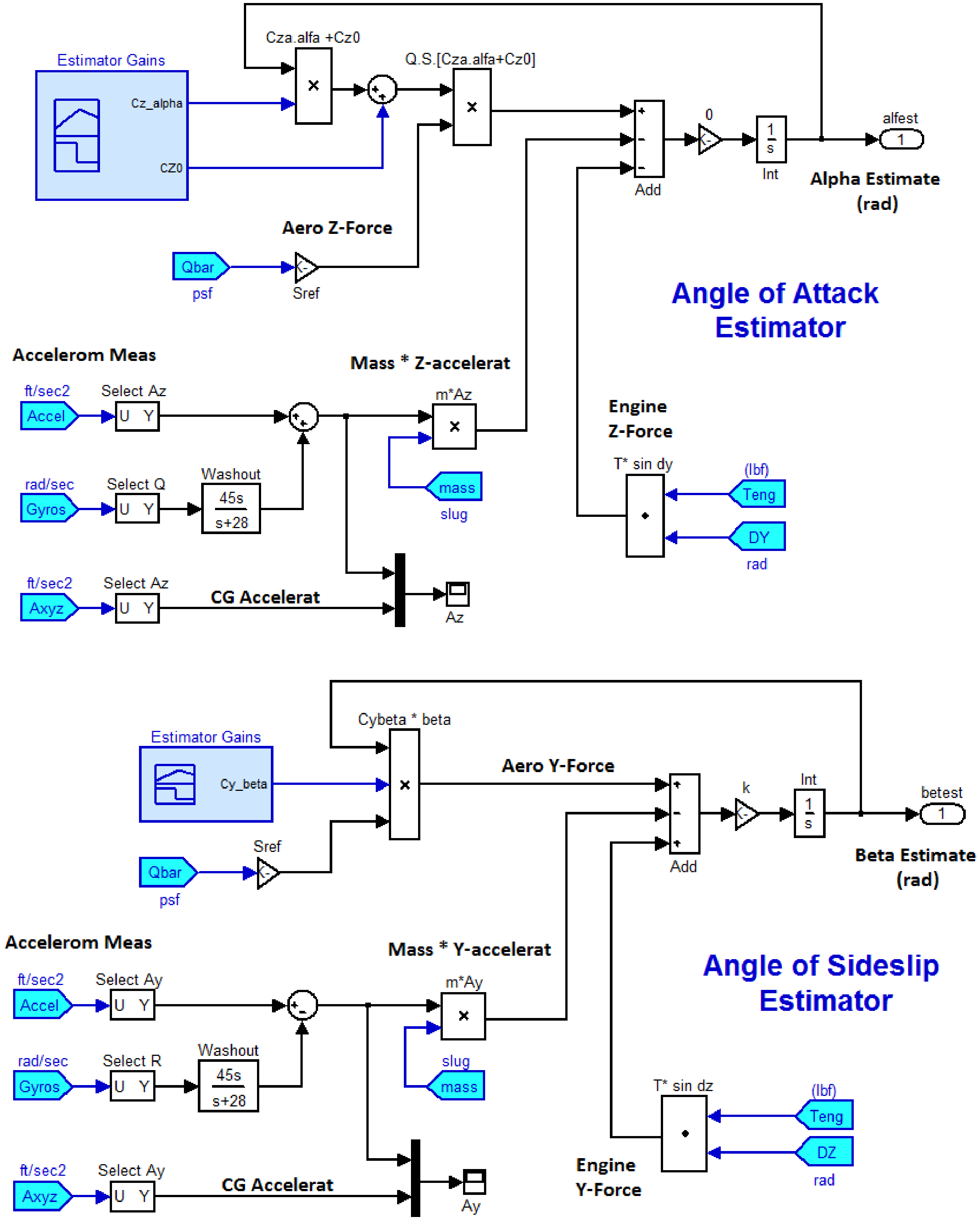


Figure 3.18 Angle of Attack and Sideslip Estimators

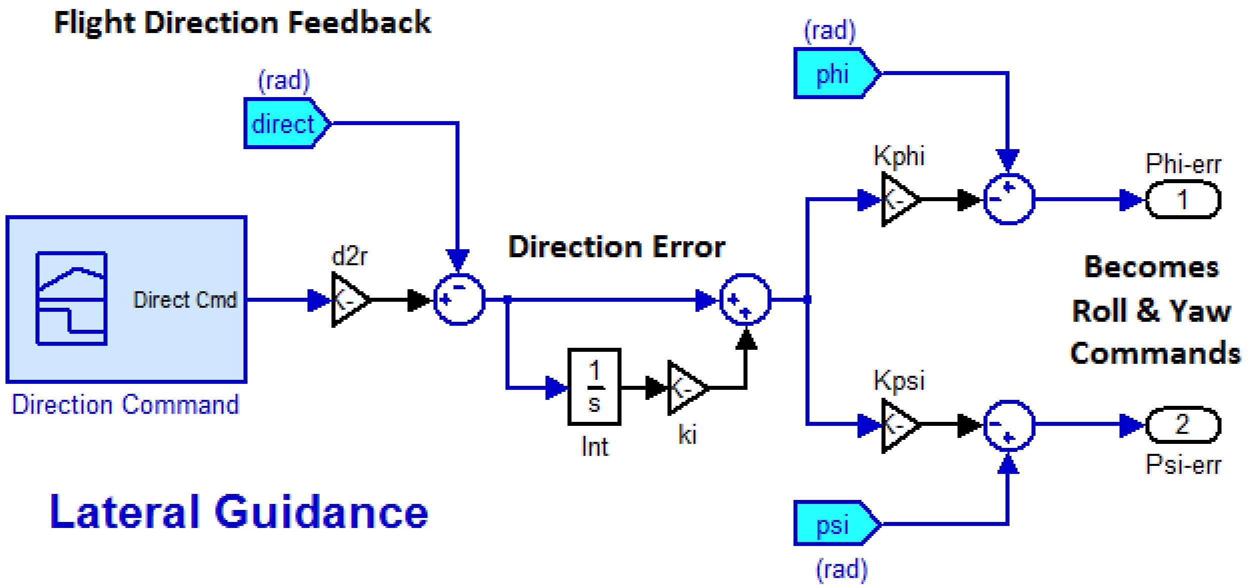


Figure 3.19 Lateral Guidance System Generates Roll and Yaw Commands to Correct Flight Direction Errors

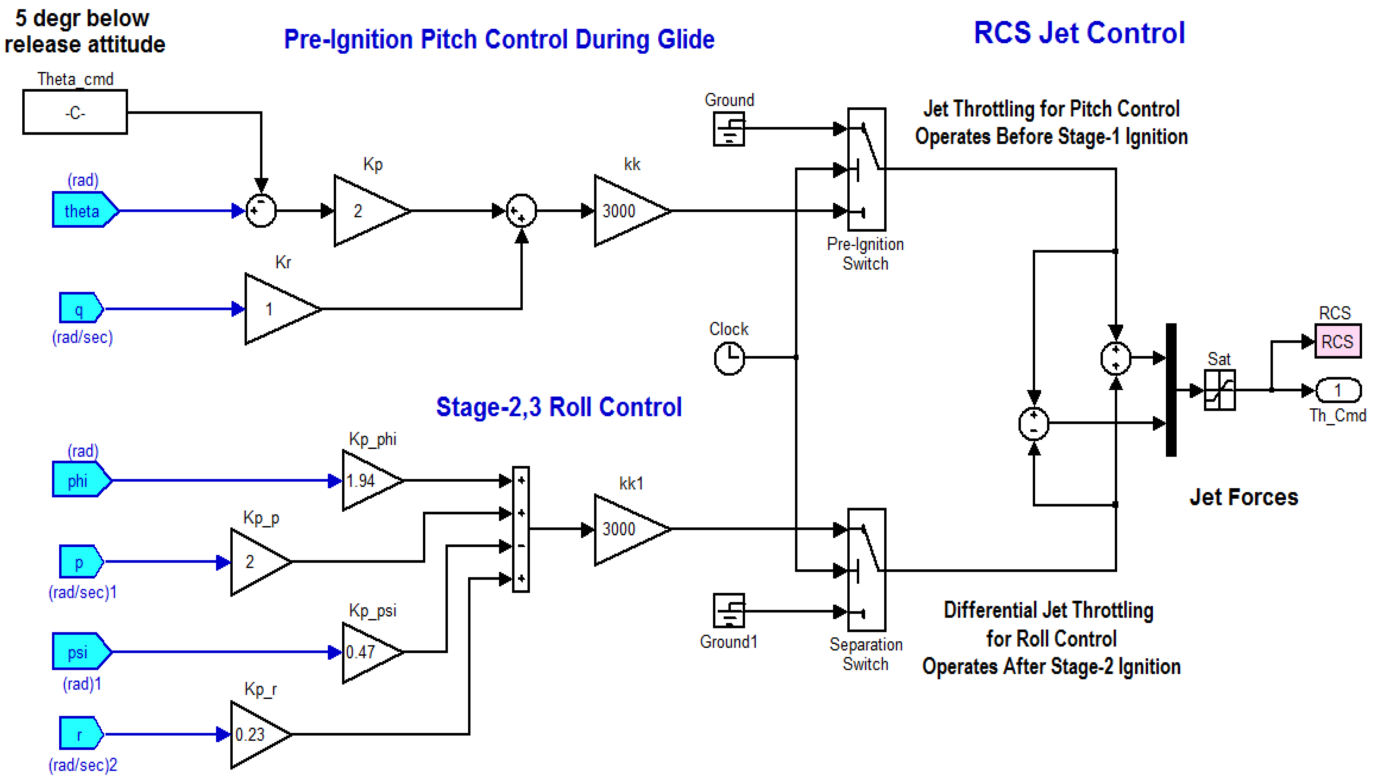


Figure 3.20 Reaction Control System for the Two Fixed Thrusters

Figure 3.20 shows the Reaction Control System for the two jets. It is used in two separate phases: (a) prior to stage-1 engine ignition to control pitch attitude in the 5 seconds period between release and engine ignition, and (b) during second and third stages to control roll attitude. It receives a negative pitch command immediately after separation from the aircraft to prevent it from colliding with the aircraft because the vehicle is unstable and it would otherwise pitch up and strike the aircraft. For simplicity it is assumed that the jets can provide a continuous thrust and they are not bang-bang.

3.2 Simulation Results

Figures (3.21 to 3.31) show the simulation results from release to orbit insertion. The motion is mostly in the pitch direction because the carrier aircraft releases it in a Northward direction required to achieve the proper inclination angle for a polar orbit in this example. There is, however, a lesser lateral motion caused by a small change in the flight direction and also due reaction against cross-wind.

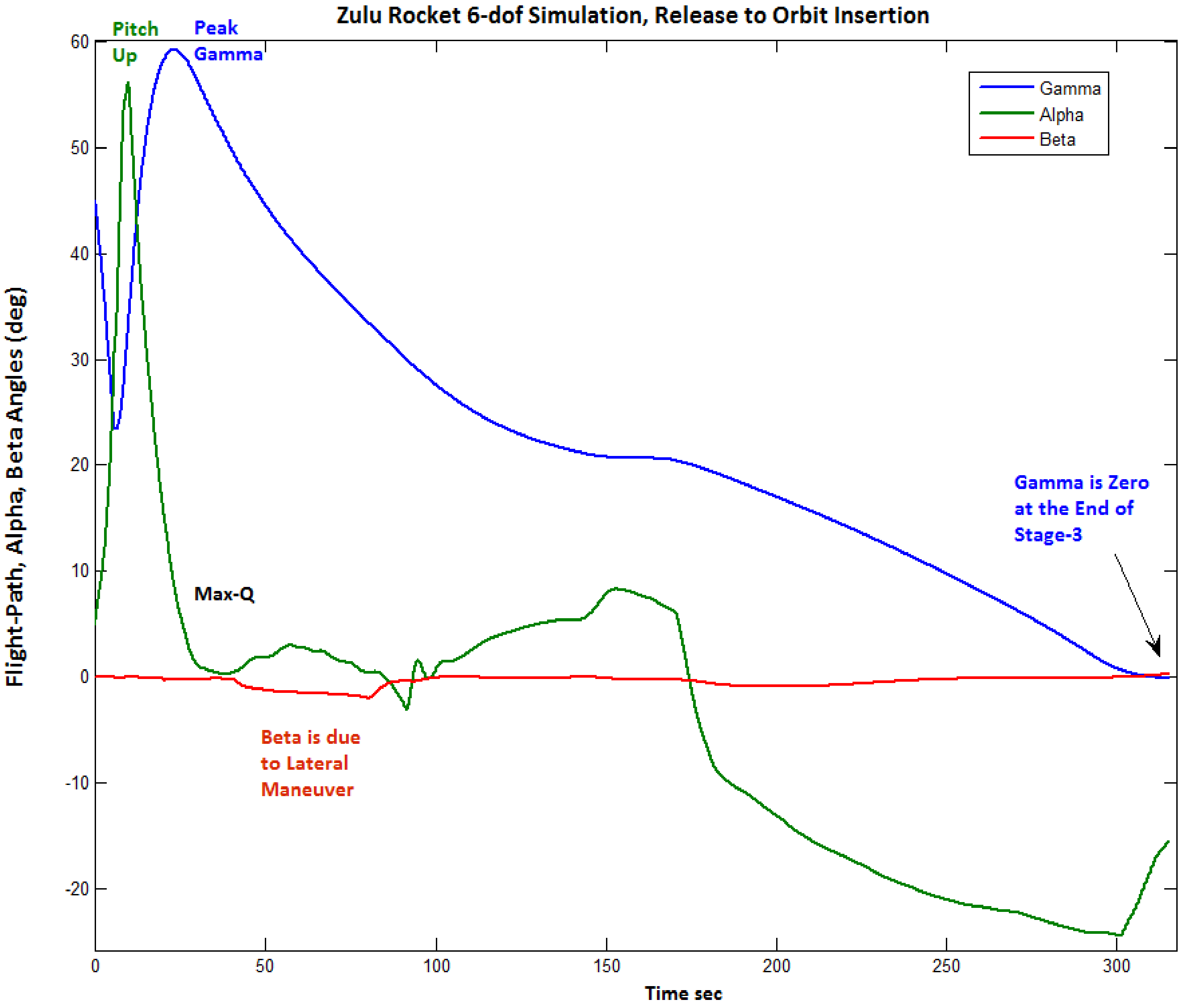


Figure 3.21 Flight Path Angle, Angle of Attack, and Angle of Sideslip

Figure 3.21 shows the flight-path angle (blue line). It begins at 45^0 at release and it drops to 24^0 during the 5 sec period without thrust. The angle of attack (green line) and pitch attitude are increased after ignition in order to raise gamma fast and before Max-Q. Gamma reaches its peak value of 59^0 and then it decreases monotonically all the way to zero which occurs at orbit insertion. The angle of attack is reduced to zero at Max-Q in order to minimize structural loading. The sideslip angle (red) is very close to zero. It deviates slightly when the vehicle reacts against wind-shear disturbances and also during the maneuver to change the flight direction.

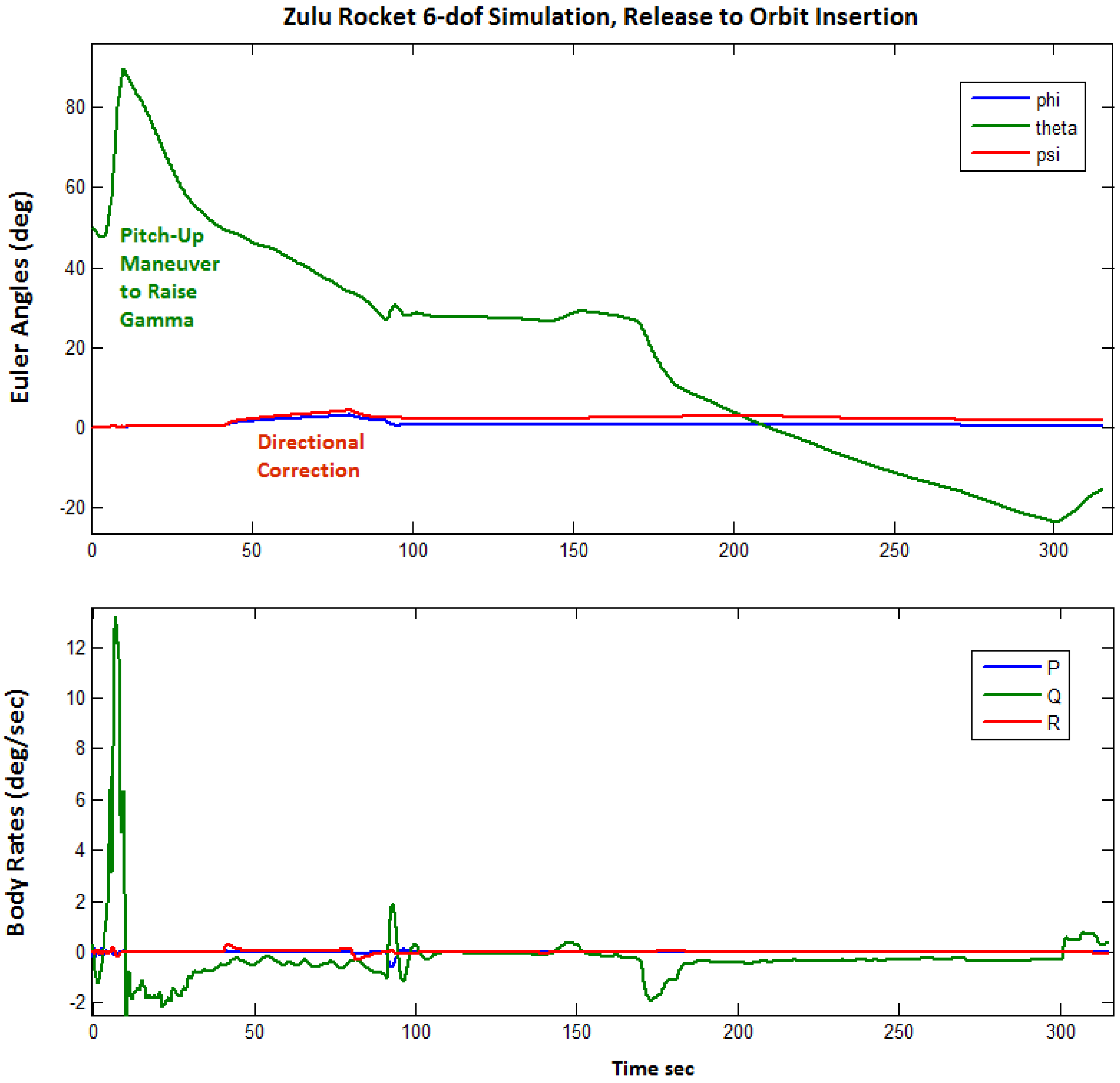


Figure 3.22 Vehicle Euler Angles and Body Rates

Figure 3.22 shows the vehicle attitude and body rates against time. The pitch attitude is commanded to rotate at almost 90^0 immediately after ignition in order to increase gamma. This pitch maneuver is fast and it is accomplished by the TVC which is assisted by the vehicle static instability, because it naturally diverges fast when alpha is high. The roll and yaw attitude and rates are very small. They deviate slightly after 40 sec to execute the 2^0 directional adjustment maneuver.

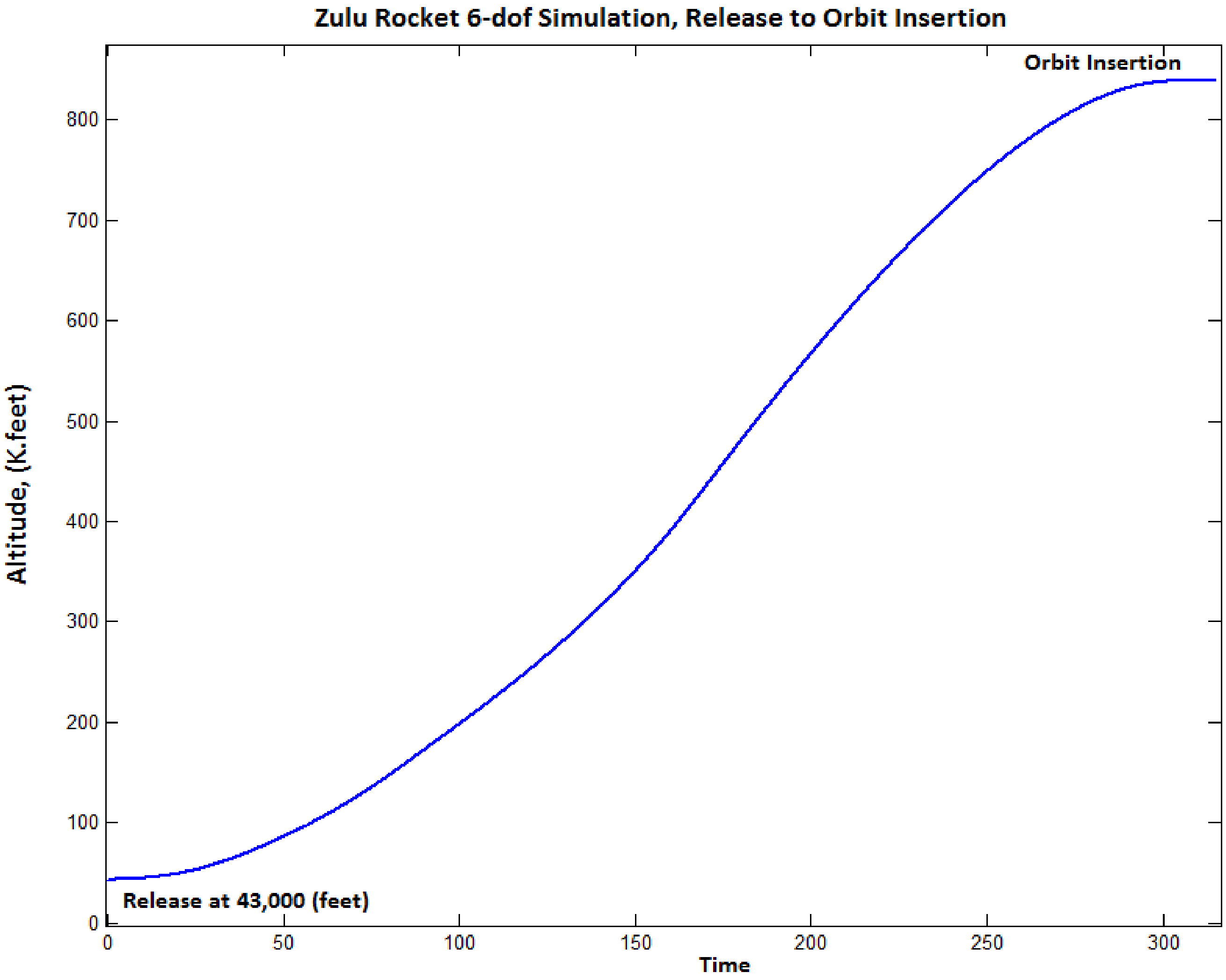


Figure 3.23 Altitude versus Time from Release to Orbit Insertion

Figure 3.23 shows the vehicle altitude against time during the 3 stages, from release to orbit insertion. The peak altitude is 840,000 feet and it flattens because gamma approaches zero. Figure 3.24 also shows the altitude plotted against downrange position in nautical miles. The 3rd stage motion is parallel to the ground at the engine cut-off.

Figure 3.25 shows the air speed variation versus time and how the accelerations vary with staging. The accelerations increase towards the end of each stage as the vehicle becomes lighter. Notice that the thrust is not zero between the first and second stages because the second stage ignites a little prior to stage-1 burn-out. This overlap is shown in Figure 3.26 that plots the engine thrusts versus time in all 3 stages. Notice that the thrusts of engines 1 and 2 are the same so the total thrust is doubled during first stage. The vehicle has only one engine during second stage and RCS for roll control. The third stage thrust is small because the vehicle is lighter and it can achieve greater acceleration with smaller thrust. The RCS jets are shown firing together during the first 5 seconds free-drift to control the pitch attitude. They are also firing differentially after stage-1 separation to control roll due to the separation disturbance.

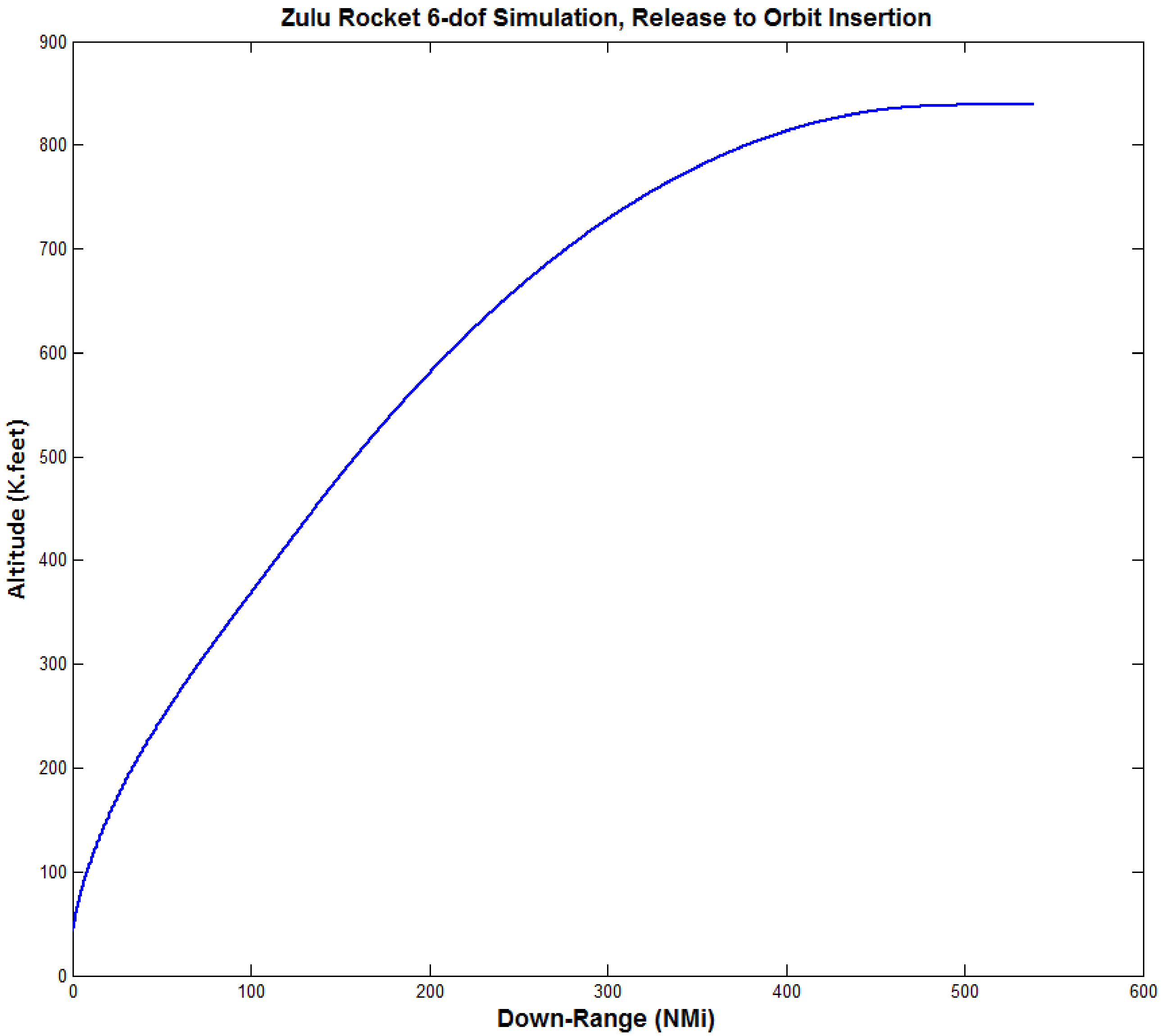


Figure 3.24 Altitude versus Down-Range Distance in Nautical Miles

Figure 3.27 shows the pitch and yaw gimbal deflections during first, second, and third stages. The pitch gimbals begin at 0.7^o which is their mounting positions. The first stage pitch gimbal angles for engines #1 and #2 (shown in green) are equal. They are ejected after staging and the green line during stage-2 in the plot is meaningless. Engine #3 is firing during second stage (shown in red). The fourth engine (also shown in red) is firing during third stage after stage-2 separates. In this simulation engine #3 plays the role of the third and fourth TVC engines for stages 2 and 3. Its thrust is loaded from the same source that includes both stages. The location of the gimbal, however, is modified after the second staging.

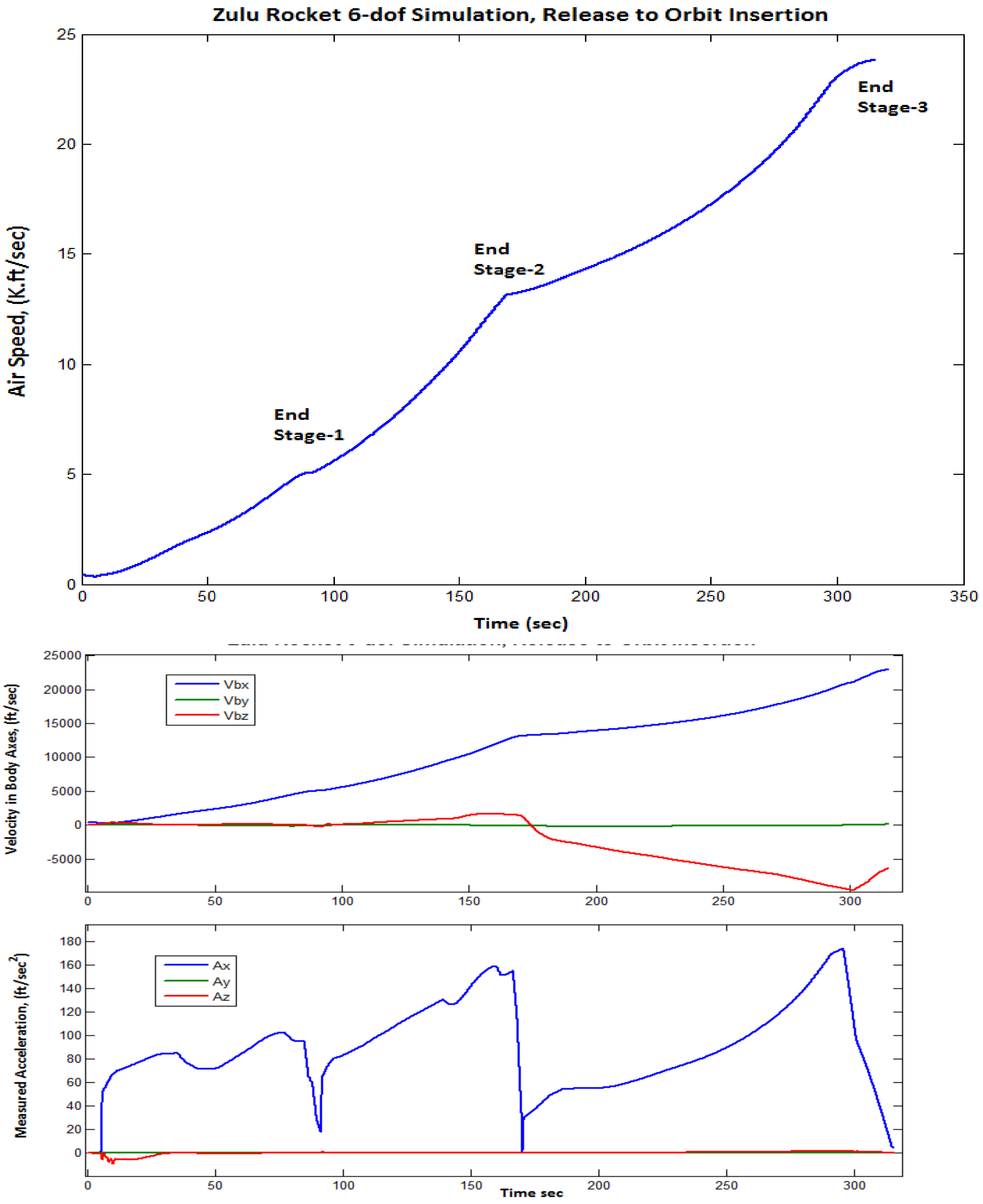


Figure 3.25 Air Speed and Accelerations during the 3 Stages

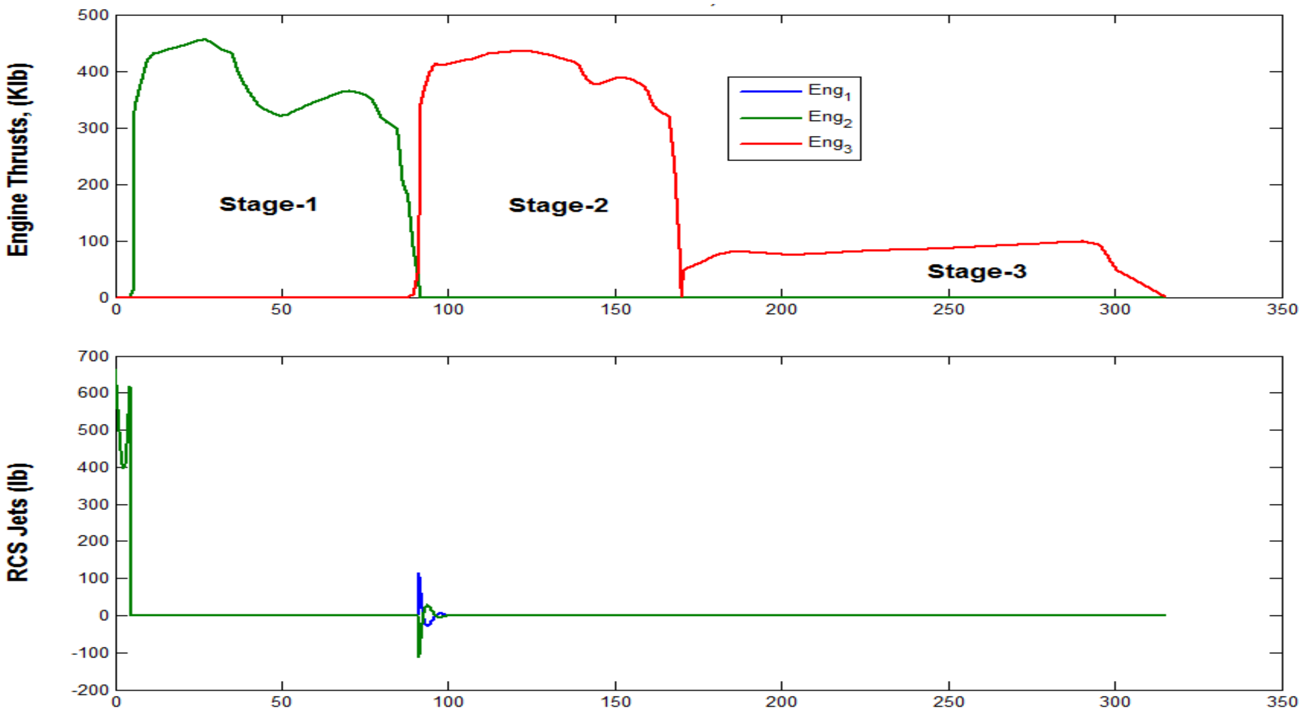


Figure 3.26 TVC Engines and RCS Thrusts versus Time

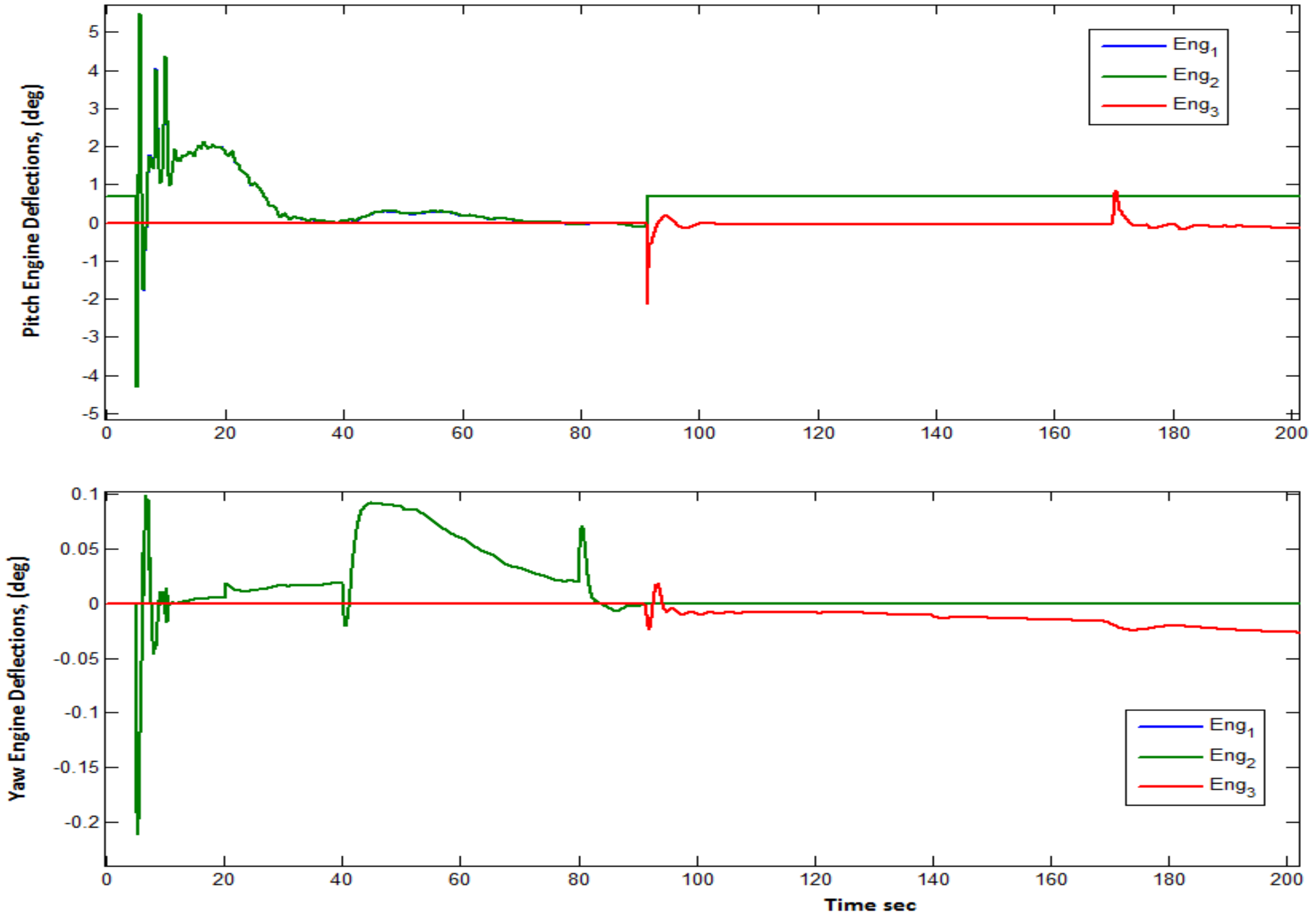


Figure 3.27 TVC Gimbal Deflections during First, Second and Third Stages

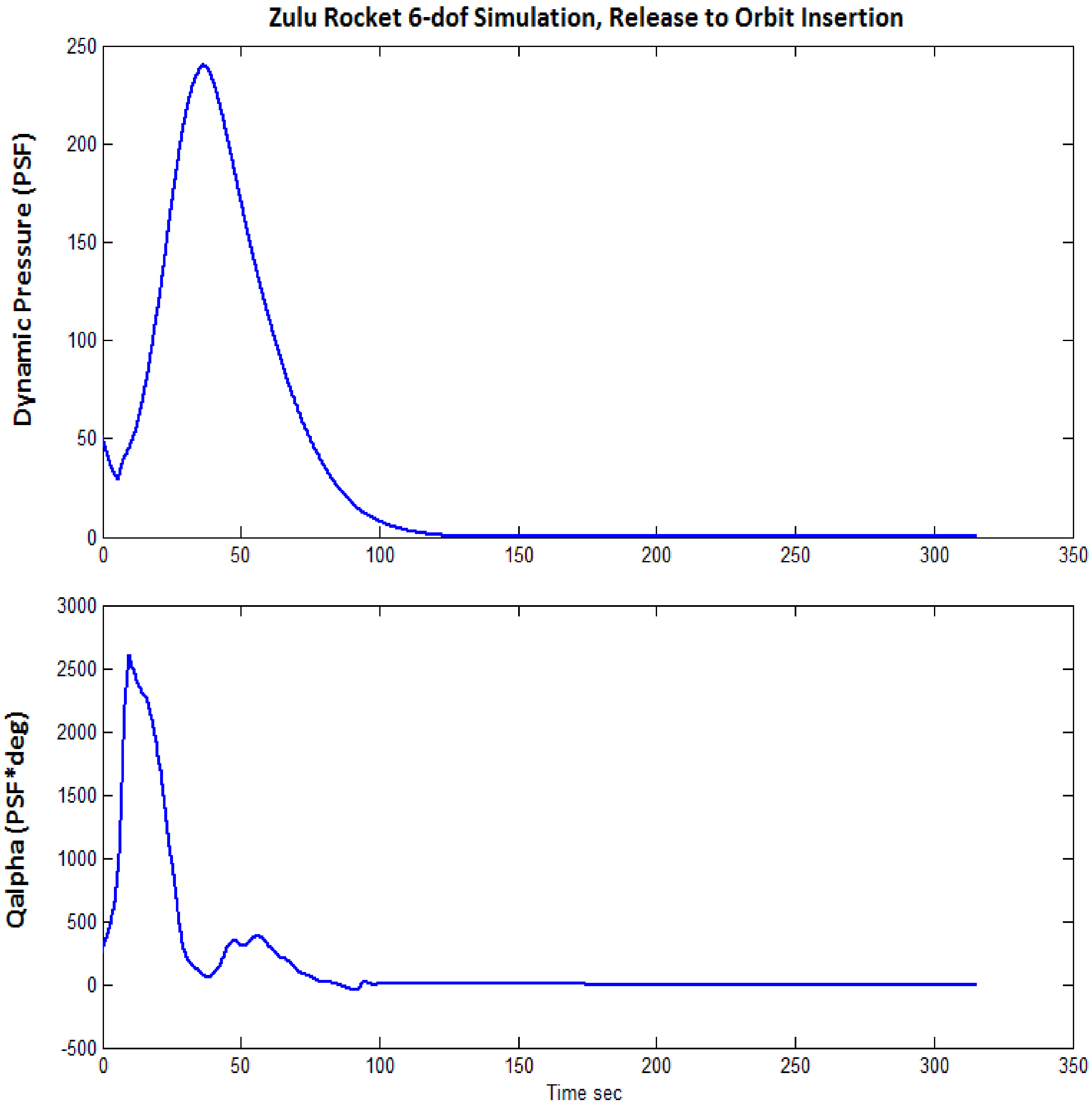


Figure 3.28 Dynamic Pressure and Qalpha. MaxQ Occurs at 40 sec and Qalpha is Less than 2600 (psf-deg)

Figure 3.28 shows the dynamic pressure that reaches a maximum of 240 (pounds/ft^2) at 40 seconds from release. This value is low in comparison with a vehicle that takes off from the ground, such as the Space Shuttle that reaches 580 (pounds/ft^2). This is one of the advantages of launching from a high altitude. The Qalpha is a measure of aerodynamic loading on the structure and it should typically be less than 3,500 (PSF-deg). In this case it is 2600 and it occurs during the pitch maneuver. This is another reason for the speedy pitch up maneuver while the dynamic pressure is still low. Notice that Qalpha is almost zero at MaxQ because alpha is reduced to zero.

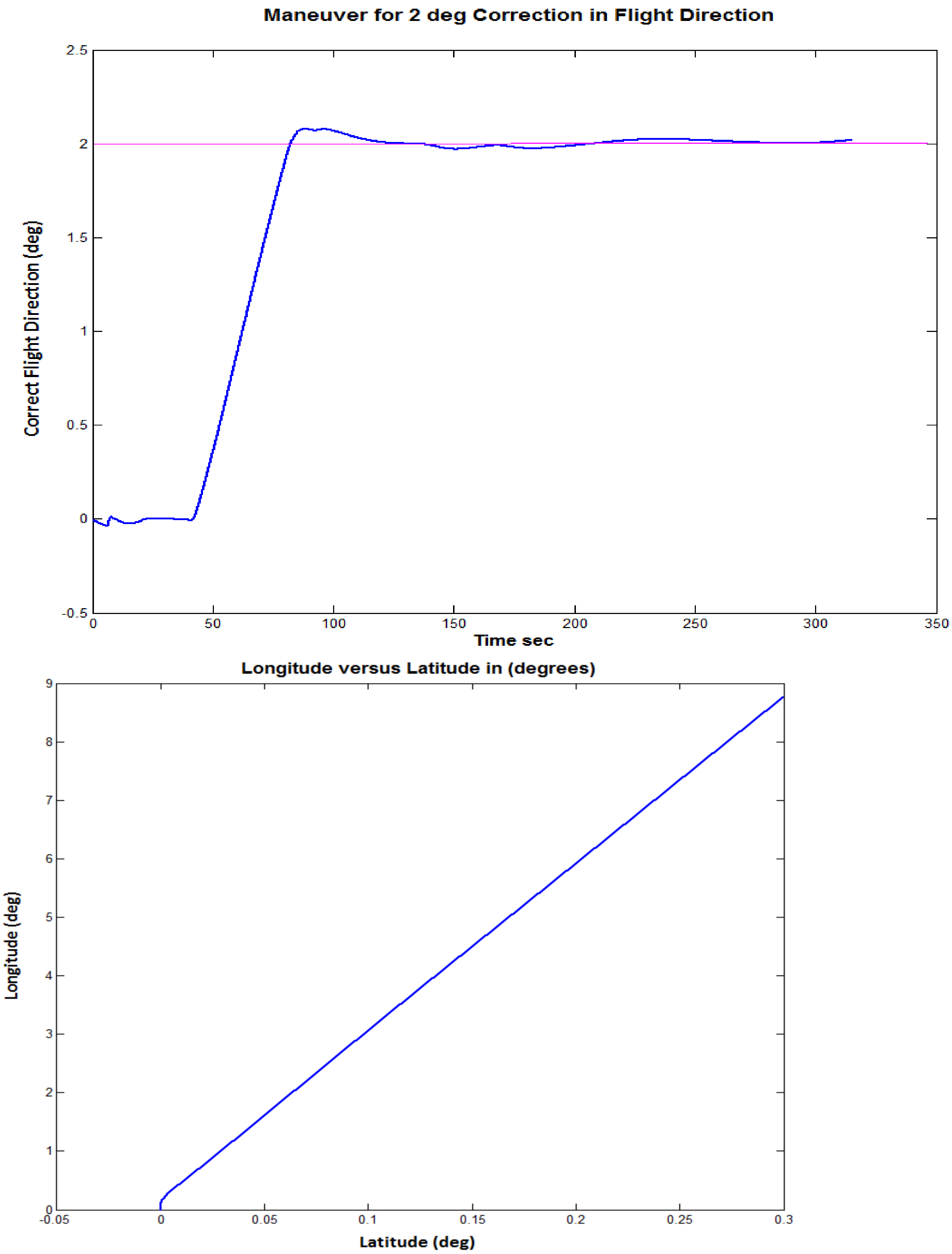


Figure 3.29 Two Degrees Maneuver to Change the Flight Direction. It shows the Modified Latitude

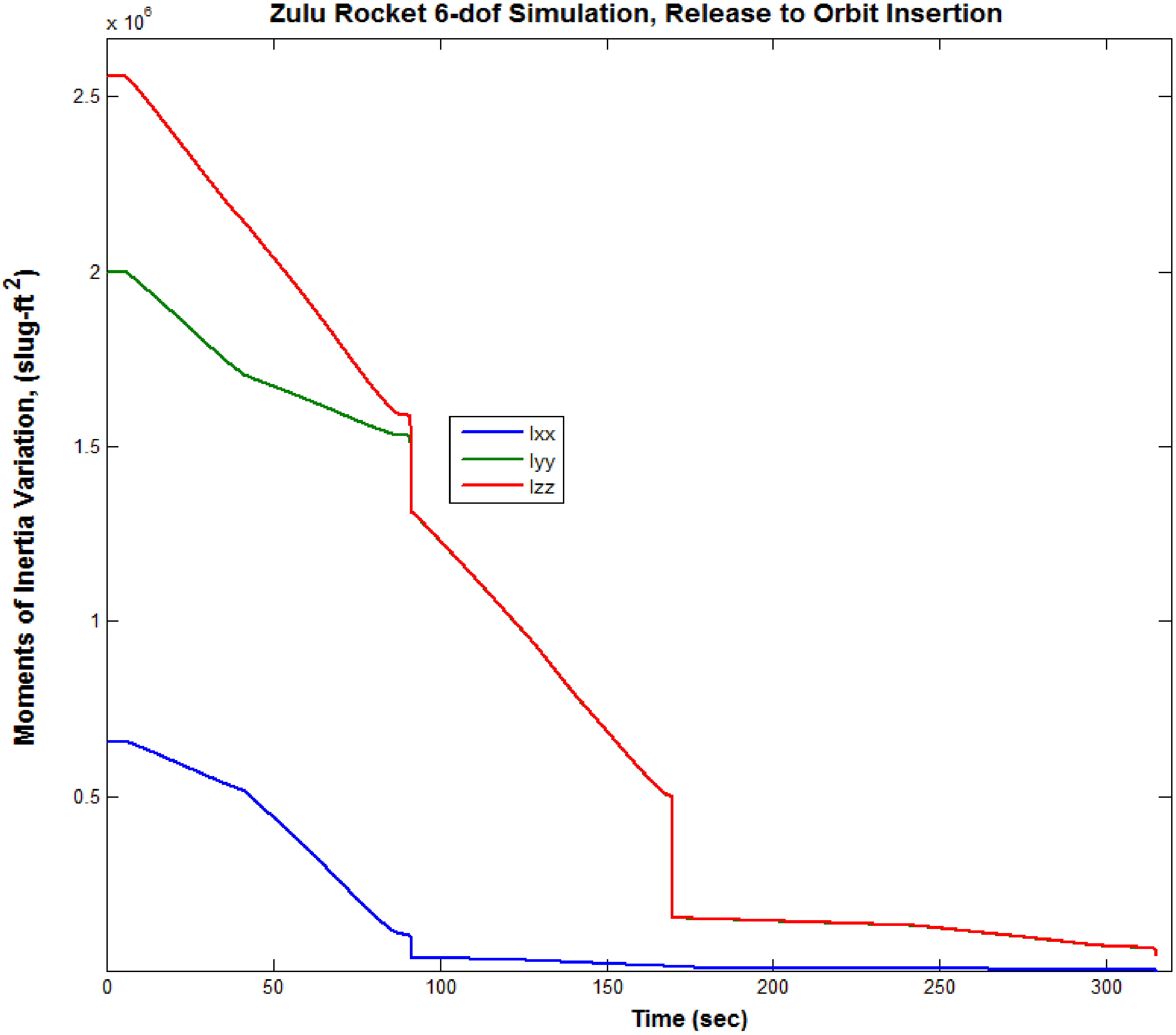


Figure 3.30 Variations of the Moments of Inertia versus Time

Figure 3.29 shows the results of executing the lateral maneuver to change the flight direction. The vehicle was initially released in a Northward direction at zero latitude, and it would have remained at zero latitude to place the spacecraft in a polar orbit if we had not commanded it to slightly change its direction. At 40 seconds it is commanded to perform a 2 degrees increment in the flight direction towards the right, and the lateral guidance system executes the command. The vehicle continues in a straight line to place the payload at the proper inclination.

Figure 3.30 shows the continuous reduction in the vehicle moments of inertia as the propellants are depleted. There is a step reduction in the MOI values during staging, after booster separation. The pitch and yaw moments of inertia become the same after stage-1 separation since the vehicle becomes a cylinder.

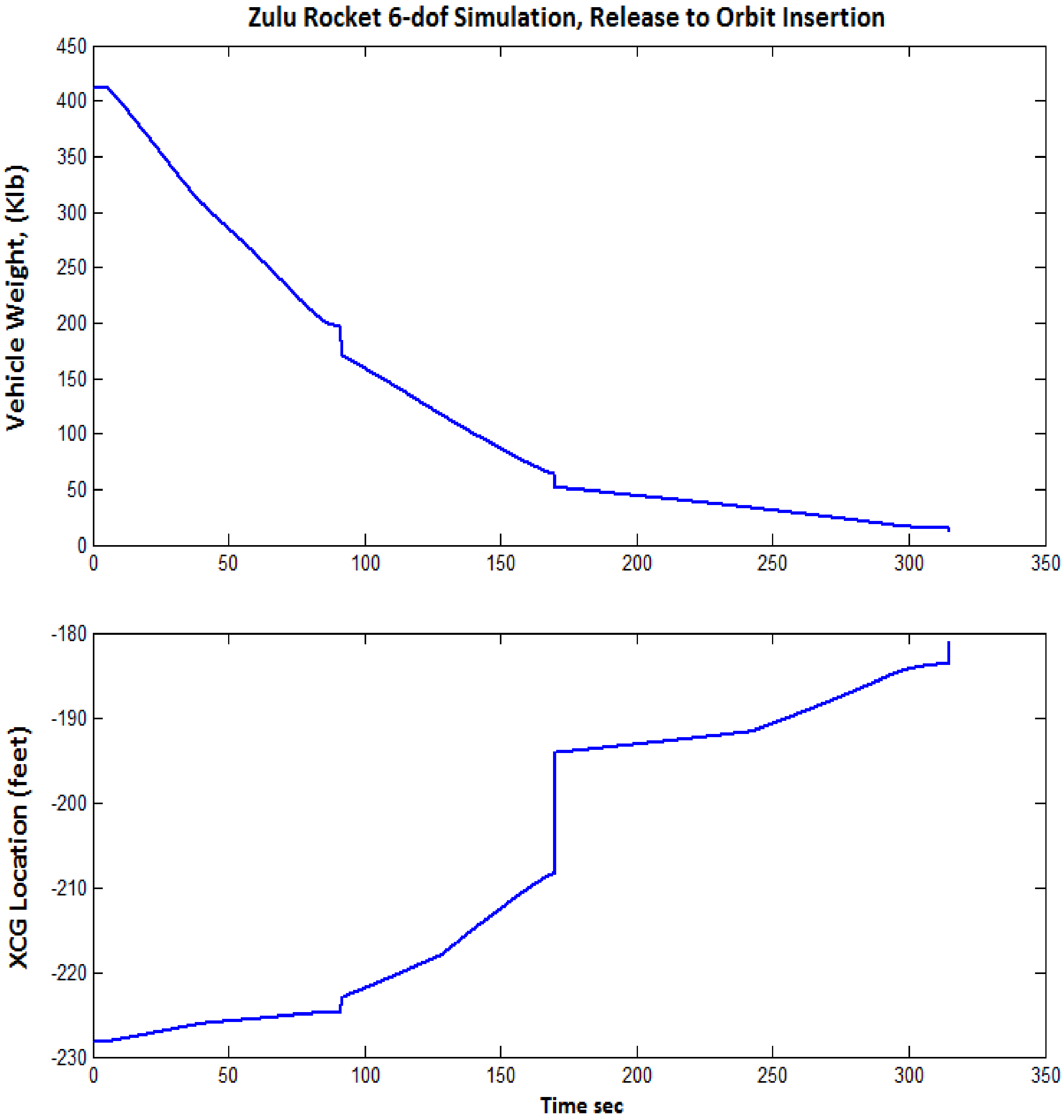


Figure 3.31 Vehicle Weight and CG Variation versus Time

Figure 3.31 shows the vehicle weight and X_{CG} variation against time. The X_{CG} value is negative measured from a point located in front of the vehicle. It naturally moves forward and the vehicle weight is reduced as the propellant is depleted. There is a step change in value during first, second, and third staging.

4. Further Analysis with Structural Flexibility and Propellant Sloshing

Our analysis so far was focused on rigid-body design and a 6-dof simulation without taking into consideration high order dynamic effects, such as, structural flexibility, propellant sloshing, tail-wag-dog, and structure/ actuator interactions. This vehicle experiences cyclic disturbances that may destabilize it. They are caused by the sloshing of the two liquid oxygen and two liquid hydrogen propellant tanks in the side boosters. It also has a flexible structure that may interact with the flight control system, the actuators, and excite some of the structural modes. The control system typically requires further modifications in order to account for those effects, and a detailed control analysis must be performed in order to verify that the system is ready for flight. We usually select some important flight conditions, generate advanced vehicle models including more efficient actuators, analyze stability and performance using the preliminary flight control system that was designed based on rigid-body models, and modify it accordingly by introducing filters, compensators, and gain adjustments. In this section we will use the Flixan program to create linear models of the Zulu vehicle at maximum dynamic pressure. Structural flexibility and propellant sloshing will be included in the dynamic models. We will design control laws and analyze vehicle stability in the frequency domain, for both pitch and lateral axes. We will finally simulate and analyze its capability to track guidance commands and its response to wind gust disturbances.

4.1.1 Data Files and Dynamic Model Preparation

In this example we will prepare two vehicle systems at Max-Q, a pitch model and a lateral (roll & yaw) model. The data files for this analysis are located in "*\Flixan\Examples\Twin Booster Rocket\4-Flex-Slosh Analysis*". The first step is to generate the flight vehicle input data files at Max-Q flight condition. In this example we have created two separate input data files, one for pitch analysis, "*Zulu_MaxQ-Ptch.Inp*", and a separate file for the lateral axes "*Zulu_MaxQ_Latr.Inp*". The input datasets used in this example were created interactively by the flight vehicle modeling program and they are similar. Their only difference is in the modal data sets that will be combined with the rigid model to generate the flex vehicle models. The vehicle dataset in the pitch input file is using a set of 8 pitch dominant modes "*Zulu, Stage-1, Max-Q, 8 Flex Modes, Pitch Modes*" that were selected using pitch mode selection criteria. The mode set in the lateral file includes 7 roll/yaw dominant modes with a title: "*Zulu, Stage-1, Max-Q, 7 Flex Modes, Roll Modes*". They were selected by the mode selection program using roll criteria. Another important difference between this advanced model and the previous rigid model used in section 2 is in the TVC engines. Notice that the tail-wags-dog option has been turned on by replacing the "No TWD" flag with "With TWD" that will create a vehicle model with engine acceleration inputs. The TWD dynamics tends to excite structural flexibility.

4.1.2 Mode Selection

The selected modes are already saved in the input data files. The selection was performed by using the Mode Selection program which is located in the Flight Vehicle Modeling group. The mode selection process has already been described in detail in other examples and it will be presented briefly here. It requires three input data files: the modal data file, the nodes file, and the vehicle input data file. The modal data file "*Zulu_MaxQ.Mod*" was generated by a finite elements model (FEM) and it contains the mode frequencies, shapes and slopes (3 translations and 3 rotations) at 6 vehicle locations (nodes) and for 13 modes (structure resonances). The nodes file (or map) "*Zulu_Stg1.Nod*" is used to identify the nodes (locations) in the modal data file and includes their coordinates in vehicle axes. It is used in the interactive process to identify the locations of the engine pivots, the slosh masses, and the flight control sensors. The flight vehicle input data is also needed in the mode selection process in order to identify and to associate FEM nodes with important vehicle locations such as engine pivots, gyros, accelerometers, etc.

The mode selection program analyzes the modal strength of each mode in the FEM file by applying (in this example) forces, measuring the structural rotations at the sensors, and selecting a number of strong modes according to a modal strength criterion. In this example we selected the two gimbals to apply forces and the attitude sensors to measure the rotations. We did not select any torque excitations and translation measurements points in the mode evaluation process. During the pitch mode selection, the two forces were applied at the gimbals and both their directions were applied along the positive Z axis to excite the structure symmetrically. The rotation at the attitude sensor was measured in the pitch direction. During the roll mode selection, the two gimbal forces were applied in opposite directions. The force at the left gimbal was applied in the -Z, and the force at the right gimbal was applied along the +Z direction to provide a torsional excitation of the structure about the X-axis. The rotation at the attitude sensor was measured in the roll direction. The mode selection program also requires the user to associate various vehicle locations such as: sensors, gimbals, and slosh masses, that appear in the vehicle dataset, with the corresponding structural nodes using menus from the map file "*Zulu_Stg1.Nod*".

The algorithm extracts a set of modes from the modal data file, changes its units and mode shape directions, because the FEM units and axes are typically different from the GN&C standards defined in the flight vehicle data. The Nastran modal data are in units of (inches), and (snails) which are different from the GN&C units of (feet) and (slugs) or any other units. In addition, the axes in the Nastran model, typically X and Z, are in opposite directions from the GN&C directions, where X is forward and Z is down. In this example we selected the default option that rescales the mode shapes by reversing the X and Z directions. The modal mass is scaled up by a factor of 12 to be converted into slugs. The mode slopes are also scaled up by a factor of 12, while the modal shapes remain the same. The modes are finally selected graphically from a bar chart according to strength and saved in the input data files. Two separate sets of modal data were extracted from the FEM file. A set of 8 pitch modes that was saved in file "*Zulu_MaxQ_Ptch.Inp*", and a set of 7 roll/yaw dominant modes that was saved in file "*Zulu-MaxQ-Latr.Inp*". The titles of the selected modal data sets are also included at the bottom of the corresponding vehicle datasets in order to associate the vehicle data with the corresponding modes.

4.1.3 Generating the State-Space Systems

After completing the mode selection in pitch and lateral axes you must make sure that the titles of the modal data sets are included in the vehicle datasets. The next step is to create the flight vehicle state-space systems. We run the flight vehicle modeling program twice to create the two systems and save them in two separate system files. The system with the pitch modes is "*Zulu, Stage-1, Max-Q, 8 Flex Modes with Slosh*" and it is saved in "*Zulu-MaxQ-Ptch.Qdr*". The system with the lateral modes is "*Zulu, Stage-1, Max-Q, 7 Flex Modes with Slosh*" and it is saved in "*Zulu_MaxQ_Latr.Qdr*". The only difference between the two systems is in their modal data. The pitch system includes 8 pitch modes and the lateral model includes 7 lateral modes. Otherwise, the vehicle parameters are the same in both systems.

The next step is to reduce the models. Although it is acceptable to analyze the dynamically coupled pitch and lateral system, it is always simpler to decouple the systems to the absolutely necessary variables before analyzing them. We, therefore, extract the pitch subsystem from the coupled system in the pitch systems file and extract also the lateral subsystem from the coupled system in the lateral systems file. The model reduction is performed using the system modification/ extraction program by retaining only the pitch inputs, states, and outputs to be used in the pitch model, and in the lateral model retaining only the lateral inputs, states, and outputs. The datasets with instructions that perform the system modifications are located in the corresponding input files (.Inp). The extracted and decoupled reduced variables systems are also saved in the corresponding system files (.Qdr). The reduced pitch system title is "*Zulu, Stage-1, Max-Q, 8 Flex Modes with Slosh, Pitch Axis*" and it is saved in "*Zulu-MaxQ-Ptch.Qdr*" and the reduced lateral system title is "*Zulu, Stage-1, Max-Q, 7 Flex Modes with Slosh, Roll & Yaw Axes*" and it is saved in "*Zulu-MaxQ-Latr.Qdr*".

4.1.4 Propellant Sloshing

In addition to the rigid body and flexibility in this example we are also modeling the dynamic behavior of the two liquid oxygen propellant tanks in the two boosters using the spring-mass analogy. The sloshing of the liquid hydrogen is not included in the model because the slosh mass and disturbance are much smaller than LOX. For more details on propellant sloshing go to section 2.6 in the flight vehicle equations section. Figure 4.1 shows the top and side view of the vehicle configuration during first stage with the two partially filled sloshing tanks in the side boosters. The middle booster is neither active nor sloshing during first stage but it ignites after separation. In the reduced pitch model the slosh mass oscillates only in the z direction and perpendicular to the acceleration vector so the slosh displacements and slosh velocity states along the y direction are truncated. In the lateral model, however, the slosh masses are free to oscillate in both y and z directions, thus exciting both roll and yaw axes, and therefore, the model includes both z and y slosh displacements and rates in the state vector because the differential z-slosh displacements in the two booster tanks excite the roll axis and the y displacements excite yaw. The slosh parameters for the left and right tanks are specified in the input data. The slosh frequencies at 1g are defined to be 2.43 (rad/sec) in both directions. However, in the model the slosh frequencies increase to 3.45 because the acceleration is almost 2g. The damping coefficients are low ζ=0.001 and the slosh mass is 350 slugs located near the bottom of the vehicle.

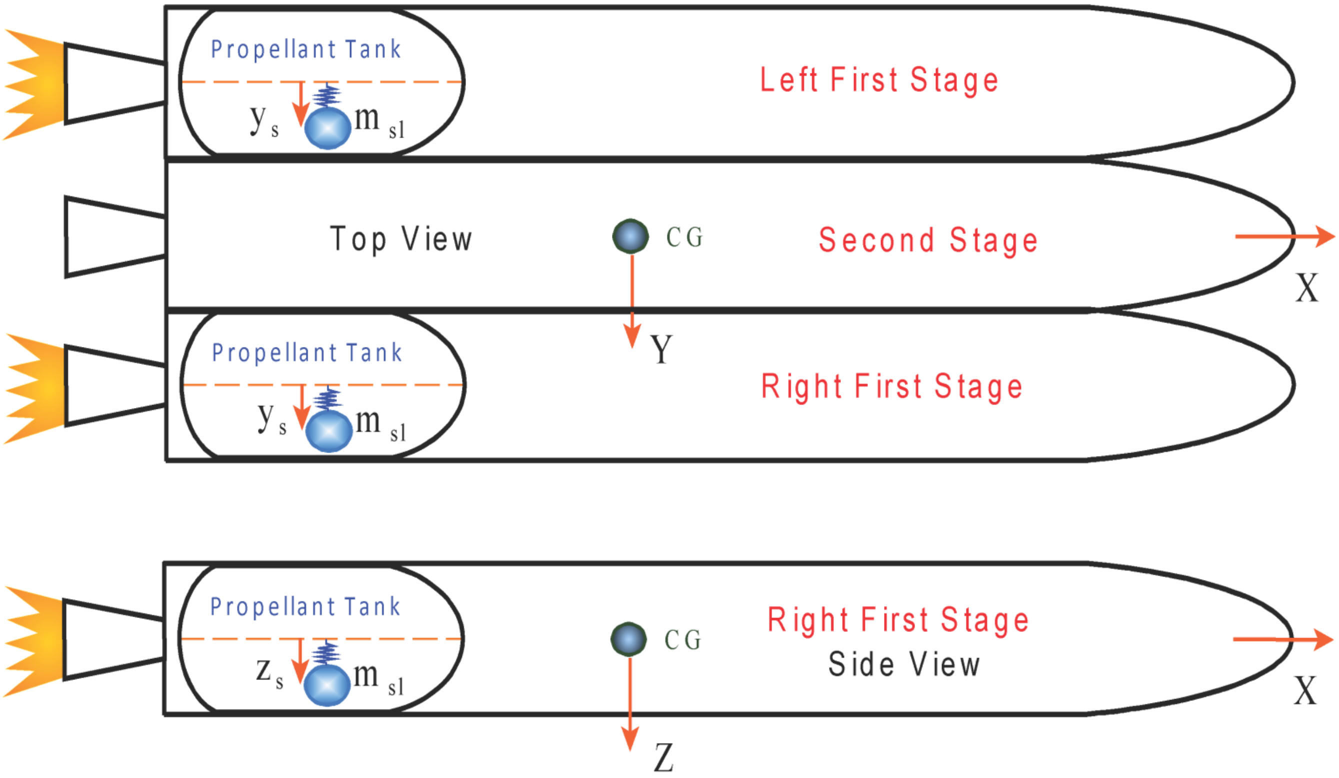


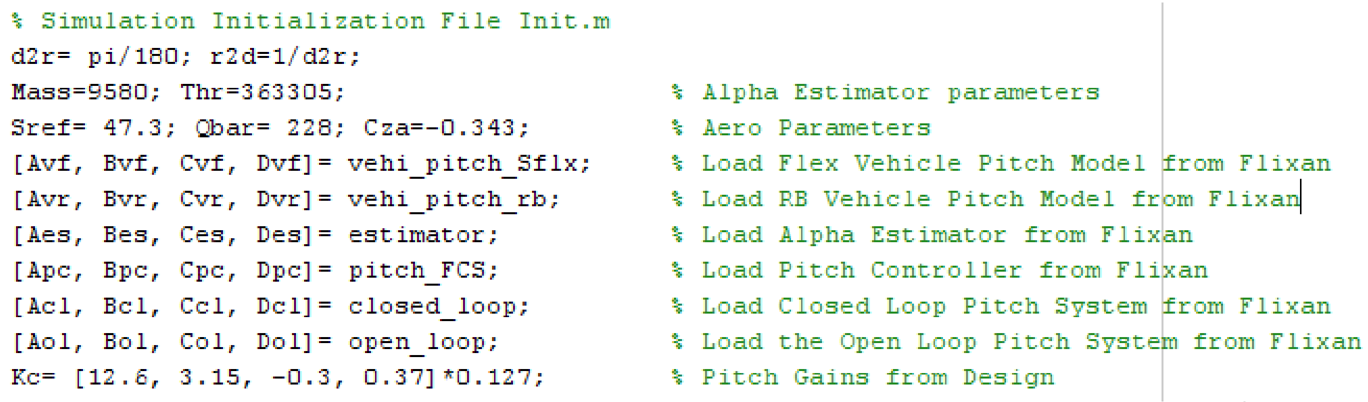
Figure 4.1 Top and Side view of the Rocket Booster during First Stage showing the two Sloshing Tanks, the Spring-Mass Model, and the Displacement of the Sloshing Masses in the y and z directions.

4.1.5 Analysis

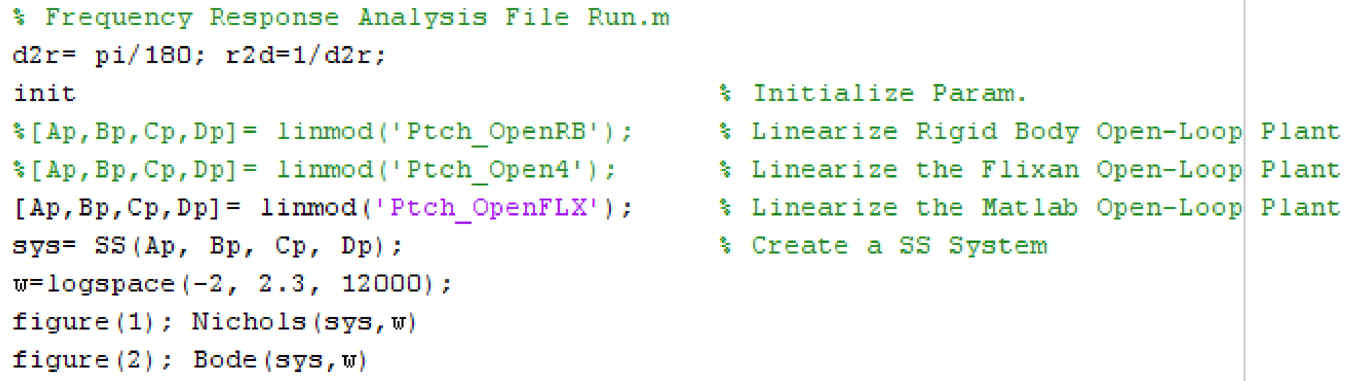
This vehicle has a significant amount of coupling in the roll and yaw axes. This dynamic coupling is caused by the CG offset in the z direction and also due to aerodynamic coupling, mainly Clβ, which is a rolling moment due to sideslip angle. We cannot, therefore, analyze the yaw and roll axes separately, as in pitch, but we must analyze them together as a coupled lateral system similar to an aircraft. We shall, therefore, perform two separate stability analysis and simulations: for pitch, and for the coupled roll/ yaw systems.

4.2.1 Pitch Axis Analysis

The pitch axis vehicle model "*Zulu, Stage-1, Max-Q, 8 Flex Modes with Slosh, Pitch Axis*" that includes slosh and flexibility was exported to Matlab and saved as a state-space function m-file: "*Vehi_Pitch_Sflx.m*" in the subdirectory "*Twin Booster Rocket\4-Flex-Slosh Analysis\Pitch Mat*". The Simulink diagram "*Ptch_OpenFLX.Mdl*" in Figure 4.2 is used for open-loop frequency response analysis, and the vehicle subsystem contains this pitch state-space system. The Matlab script files (run.m and init.m) load the Flixan generated data files into Matlab and calculate the Bode and Nichols plots. Figure 4.5 shows a closed-loop simulation diagram that includes the commands and the flight control system. The control system is designed to receive two commands from guidance, an attitude command (θ-com), and a flight path angle command (γ-com). The two variables cannot be independently controlled and the two commands must, therefore, be coordinated. They were calculated from a point mass trajectory optimization program. The controller is a state-feedback gain matrix Kc from the four pitch states: attitude θ, pitch rate q, angle of attack α, and integral of the flight path angle γ-integral. The γ integrator provides better tracking of the flight path angle. It was originally designed using rigid plant models and the Linear Quadratic Regulator (LQR) method, as it was described in in Section 2.



```
% Simulation Initialization File Init.m
d2r= pi/180; r2d=1/d2r;
Mass=9580; Thr=363305;                   % Alpha Estimator parameters
Sref= 47.3; Qbar= 228; Cza=-0.343;       % Aero Parameters
[Avf, Bvf, Cvf, Dvf]= vehi_pitch_Sflx;   % Load Flex Vehicle Pitch Model from Flixan
[Avr, Bvr, Cvr, Dvr]= vehi_pitch_rb;     % Load RB Vehicle Pitch Model from Flixan
[Aes, Bes, Ces, Des]= estimator;         % Load Alpha Estimator from Flixan
[Apc, Bpc, Cpc, Dpc]= pitch_FCS;         % Load Pitch Controller from Flixan
[Acl, Bcl, Ccl, Dcl]= closed_loop;       % Load Closed Loop Pitch System from Flixan
[Aol, Bol, Col, Dol]= open_loop;         % Load the Open Loop Pitch System from Flixan
Kc= [12.6, 3.15, -0.3, 0.37]*0.127;      % Pitch Gains from Design
```



```
% Frequency Response Analysis File Run.m
d2r= pi/180; r2d=1/d2r;
init                                         % Initialize Param.
%[Ap,Bp,Cp,Dp]= linmod('Ptch_OpenRB');       % Linearize Rigid Body Open-Loop Plant
%[Ap,Bp,Cp,Dp]= linmod('Ptch_Open4');        % Linearize the Flixan Open-Loop Plant
[Ap,Bp,Cp,Dp]= linmod('Ptch_OpenFLX');       % Linearize the Matlab Open-Loop Plant
sys= SS(Ap, Bp, Cp, Dp);                     % Create a SS System
w=logspace(-2, 2.3, 12000);
figure(1); Nichols(sys,w)
figure(2); Bode(sys,w)
```

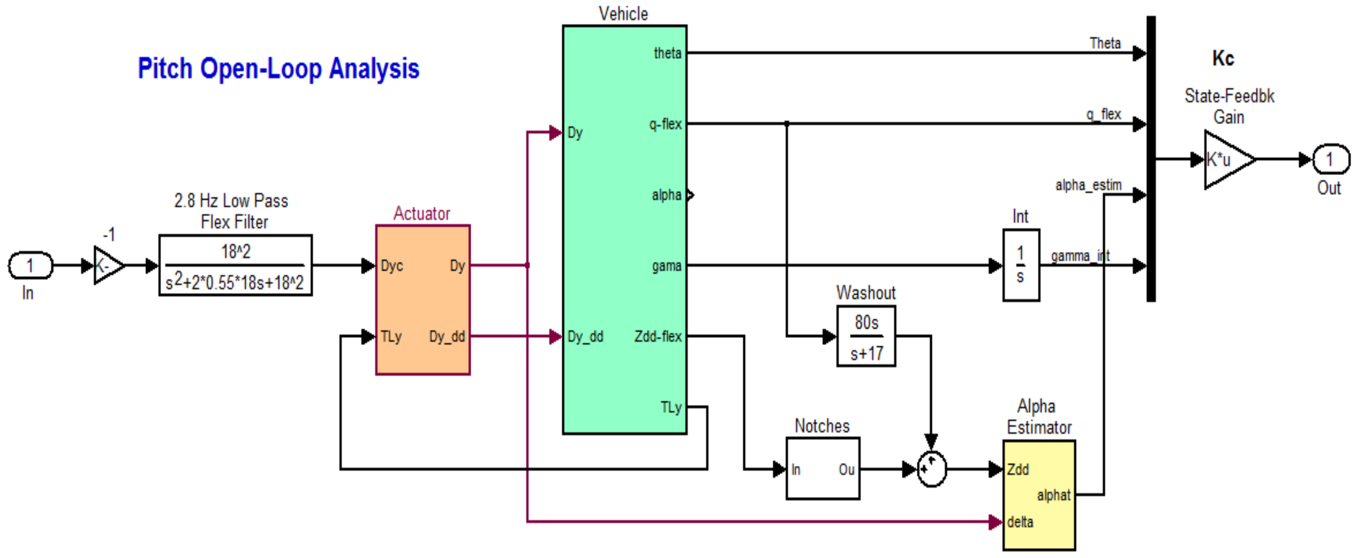


Figure 4.2 Open-Loop Diagram of the Simulink Model "Ptch_OpenFLX.mdl" used for Frequency Response Analysis

The gains are adjusted by the weights selection in the LQR design to emphasize controlling either attitude or tracking gamma in different phases of the trajectory. In the early phase attitude tracking is more important and it is emphasized more than other states. At high dynamic pressures, where structural loading becomes more important than responding to guidance, the feedback gain from alpha is increased in order to reduce loading caused by excesses in the angle of attack, or sideslip in the lateral design. Gamma tracking is emphasized later in flight as we move toward the low Q-bar region in order to achieve our trajectory goals. However, some adjustments in the gains were made and filters were included in order to accommodate structural flexibility and propellant sloshing. Three of the feedback states are almost directly measurable. The only state that is not measured directly is α. The angle of attack is estimated using an estimator and measurements from an accelerometer and the gimbal angles. In this linear analysis we assume that ($\gamma = \theta - \alpha$). In reality γ is estimated from navigation using GPS and inertial sensors.

4.2.2 Angle of Attack Estimator

The angle of attack is the only state among the pitch state-feedback variables that it is not directly measurable. Although α can be estimated by the navigation system at a slower rate, we will use an α-estimator to calculate it, that basically solves the normal force equation.

$$m\ddot{z} = \overline{Q}\, S_{ref}\, C_{Z\alpha}\, \alpha - T\sin(\delta)$$

Figure 4.3 is basically a filter that estimates α from the accelerometer measurement, the aerodynamic force, and the normal component of the TVC force. The α-estimator also requires approximate values of the vehicle mass, dynamic pressure, the normal force aero coefficient, thrust, measurements from the normal accelerometer (Nz), and the pitch gimbal angles δ_Y. Note, that the normal accelerometer does not only measures the translational vehicle acceleration along the z axis. It senses also an undesirable rotational acceleration component due to vehicle rotation. The accelerometer signal at the input of the alpha estimator is, therefore, compensated by using an additional feedback from the pitch rate gyro. A washout filter is introduced which differentiates the pitch rate gyro signal at frequencies below 17 (rad/sec). The signal from the washout filter attempts to counteract (at low frequencies only) the rotational acceleration component measured by the accelerometer and to provide a better alpha estimate.

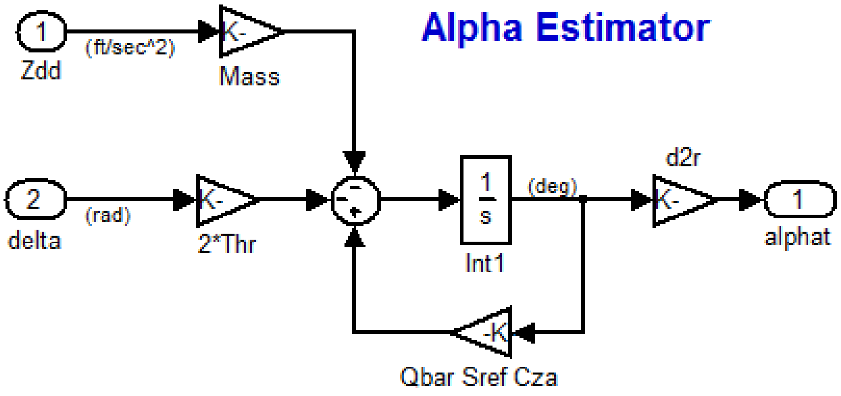
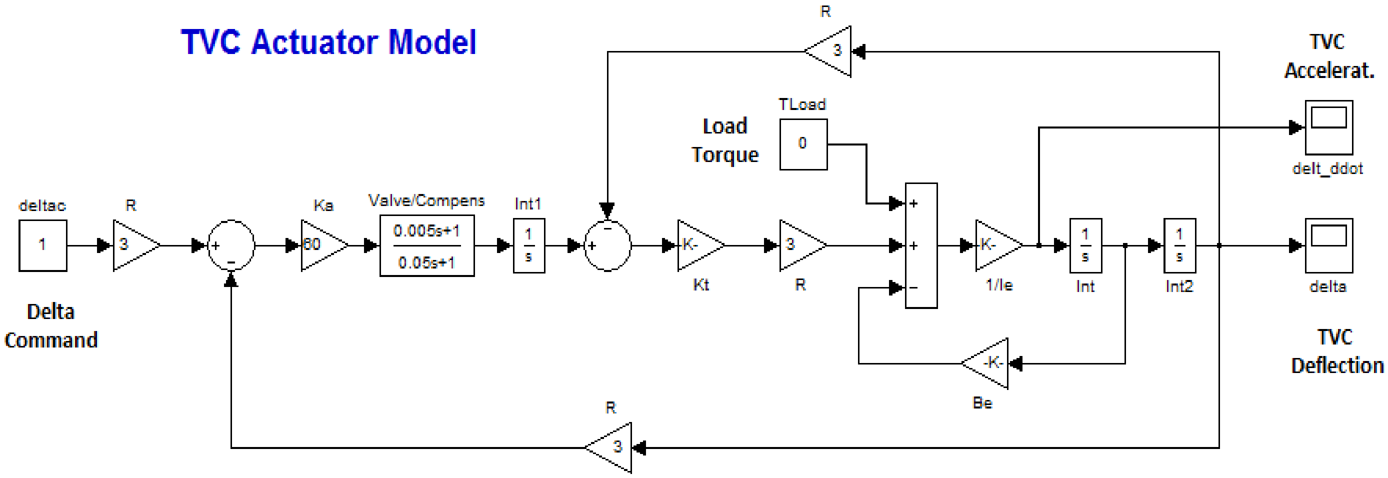


Figure 4.3 Angle of Attack Estimator

4.2.3 Actuator Model

A mode advanced actuator model is used in this control analysis, shown below. This actuator introduces dynamic coupling between engines motion, structural flexibility, and actuator dynamics at the gimbal. It was implemented using the Flixan actuator modeling option which converts the actuator dataset to a state-space model. The actuator title is "*Engine TVC Actuator*", and it is converted to a state-space function m-file "actuator.m" that is loaded into Matlab together with the other systems during initialization. This actuator model includes a gimbal acceleration output that connects to the vehicle model input in order to excite the tail-wag-dog (TWD) dynamics. Note that by setting the "With TWD" flag in the vehicle dataset, the program creates additional TVC acceleration inputs in the vehicle model that can connect with the actuator outputs. The actuator model has a second input, the load-torque input that connects to the load-torque output of the vehicle model, thus closing a mechanical feedback loop. A simple transfer function cannot capture the tail-wags-dog and load-torque feedback effects.



4.2.4 Pitch Stability Analysis

The Simulink model "*Ptch_OpenFLX.mdl*" in Figure 4.2 is used for calculating the pitch axis open-loop frequency response by running the script file "run.m". However, in this case the frequency responses were generated by Flixan using the system "*Pitch Open-Loop System*" which is the same as Figure 4.2 but it was implemented using the Flixan program. The Bode plot in Figure 4.4 shows the LOX slosh resonance at 0.6 Hz which is near the cross-over frequency but it is phase-stable with plenty of margin. There are also two strong structural modes at 3.52 Hz and at 7.95 Hz. The Nichols plot shows that the system has sufficient phase and gain margins. The locus encircles the critical region without crossing it. The 3.52 Hz first fuselage bending mode is also strong and phase stable with sufficient margin. It peaks at +16 (dB) but it does not encircle the critical region. Its mode shape is not expected to change much during flight but to remain phase stable. The second body-bending mode at 7.95 Hz is barely gain stable and well behaved in phase, that is, far enough from encircling the critical regions that threaten stability. We conclude, therefore, that the system is stabilizable with filters. The alpha gain was reduced from the LQR design value in order accommodate the alpha estimator.

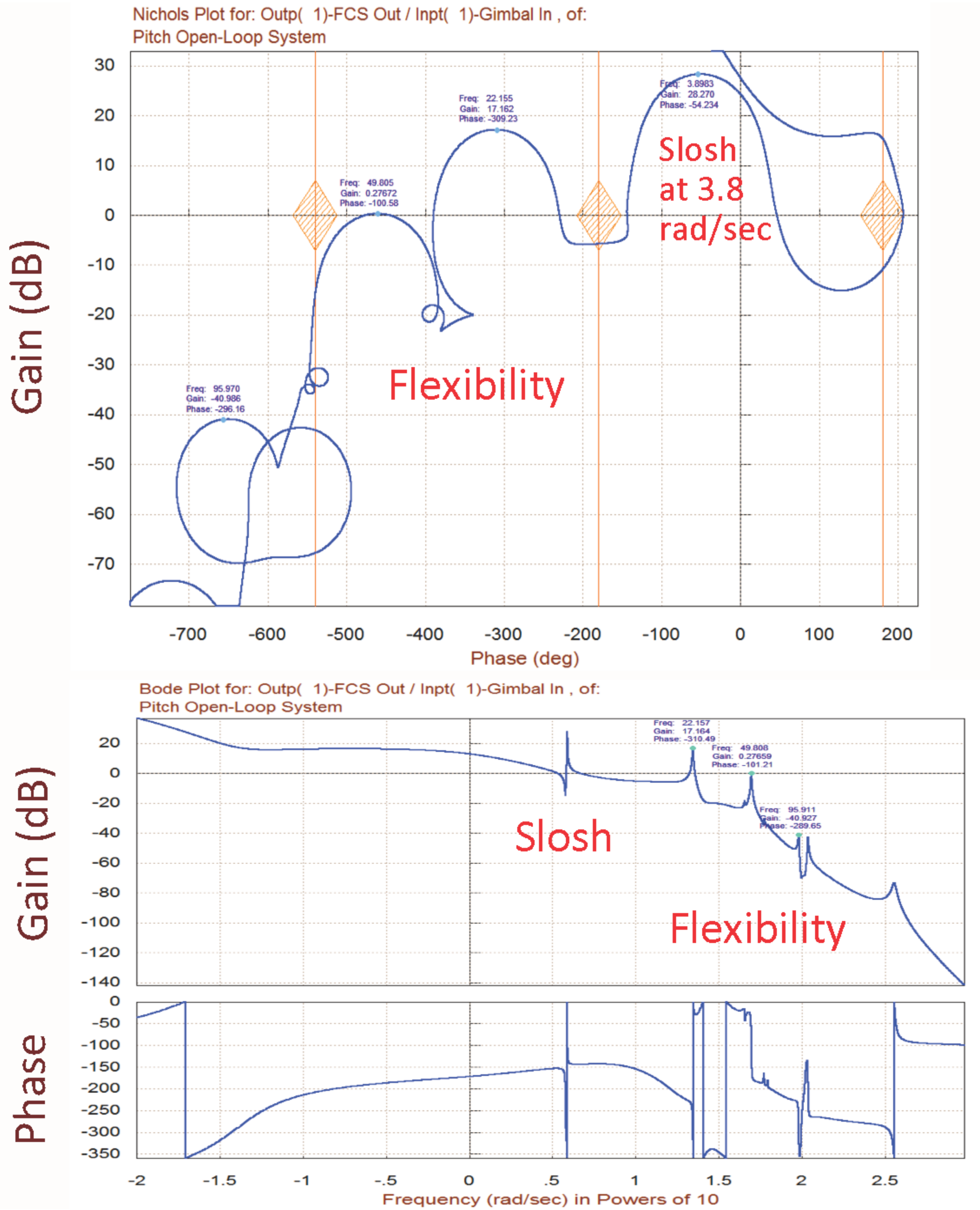


Figure 4.4 Pitch Axis Stability Analysis Using the Open-Loop Model

4.2.5 Pitch Axis Simulation

The simulation is used to analyze the system performance to step guidance commands and also to the winds. It uses the Simulink model "*Ptch_Sim.Mdl*", shown in Figure 4.5, located in subdirectory *"\Twin Booster Rocket\4-Flex-Slosh Analysis\Pitch Mat*". The model receives two commands: an attitude (θ-command), and a flight path (γ-command). The state-space systems and other parameters are loaded into Matlab by the script file "init.m".

In the simulation results, shown in Figure 4.6, the vehicle is commanded to perform 2 degrees increment in gamma. The pitch attitude is not commanded. In addition to the guidance commands the simulation includes a wind gust disturbance input. The input is wind velocity in (feet/sec). The wind disturbance is generated from a noise source and the random signal is limited to 0.2 Hz in bandwidth and 10 (feet/sec) in wind velocity. The wind direction with respect to the vehicle is specified in the vehicle input data. It is perpendicular to the x axis. The file "pl.m" plots the data.

Figure 4.6a shows the vehicle angles (α, γ, and θ) responding to the 2 degrees γ-command in presence of the wind noise. The pitch attitude and alpha respond as necessary in order to achieve the commanded gamma. The alpha-estimate is almost identical to the real alpha, as shown in Figure 4.6b. It is used instead of alpha in the state-feedback. Figure 4.6b also shows the responses of the pitch rate gyro and the Nz-accelerometer.

The gamma response is slow because it is not easy to change the direction of the vehicle as fast as the attitude. Figure 4.7 shows the response of the control system to 1^o of pitch attitude command. The gamma command is zero. The system responds fast to the attitude change command but it slowly drifts from the desired attitude because of the gamma-integral feedback, since it is physically impossible to independently command those two variables. The plots also show the effects of sloshing and structural flexibility on the vehicle responses

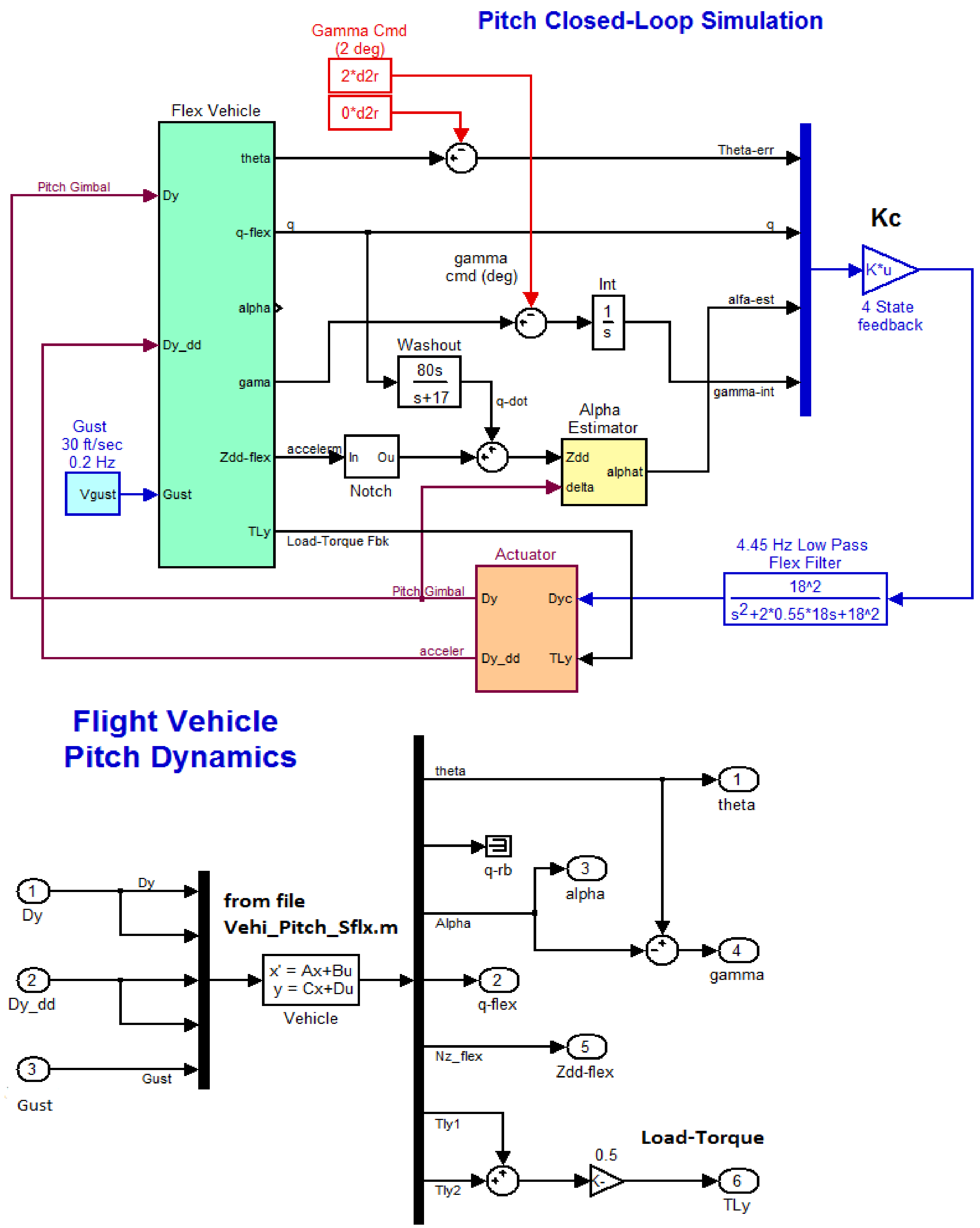
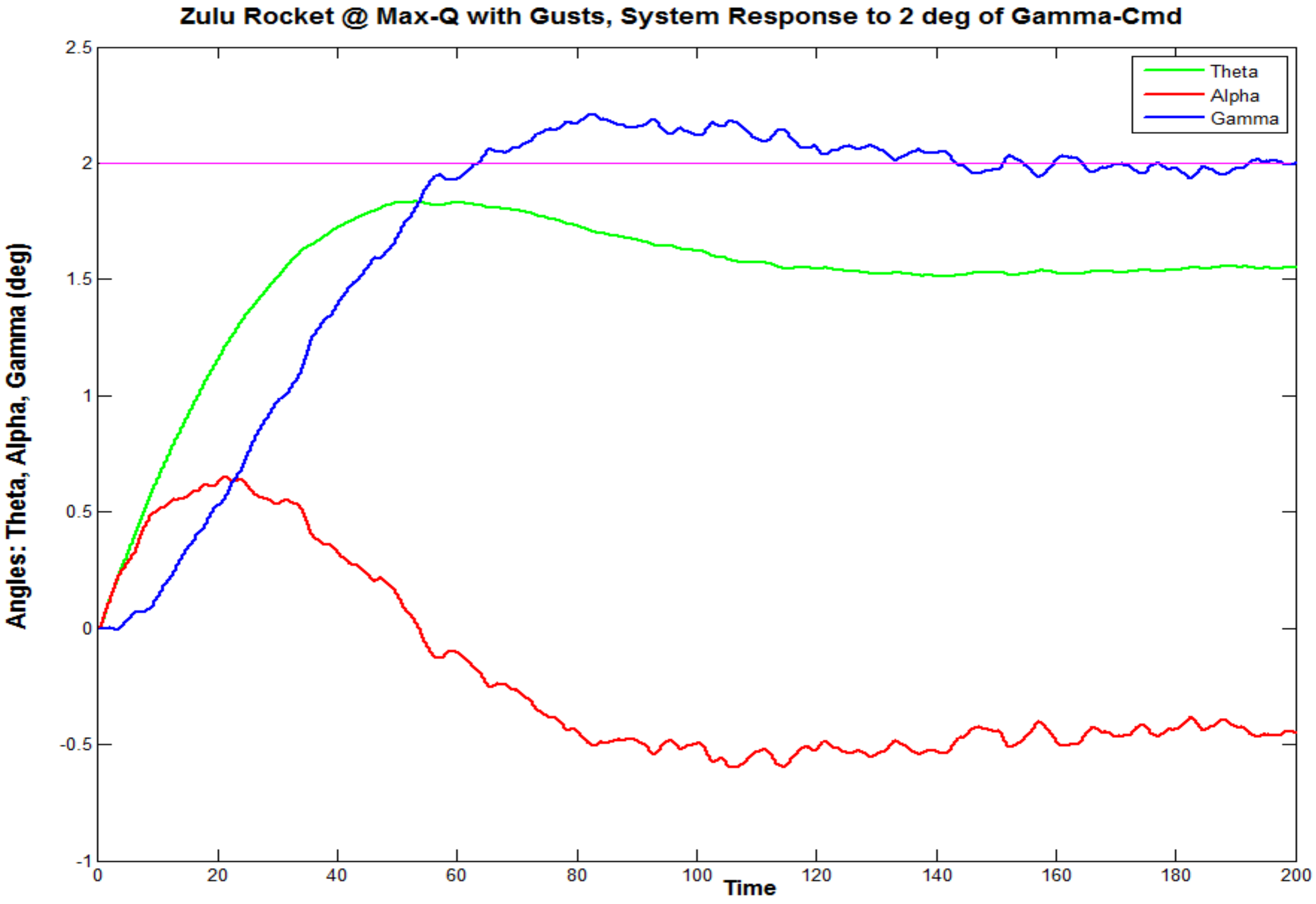
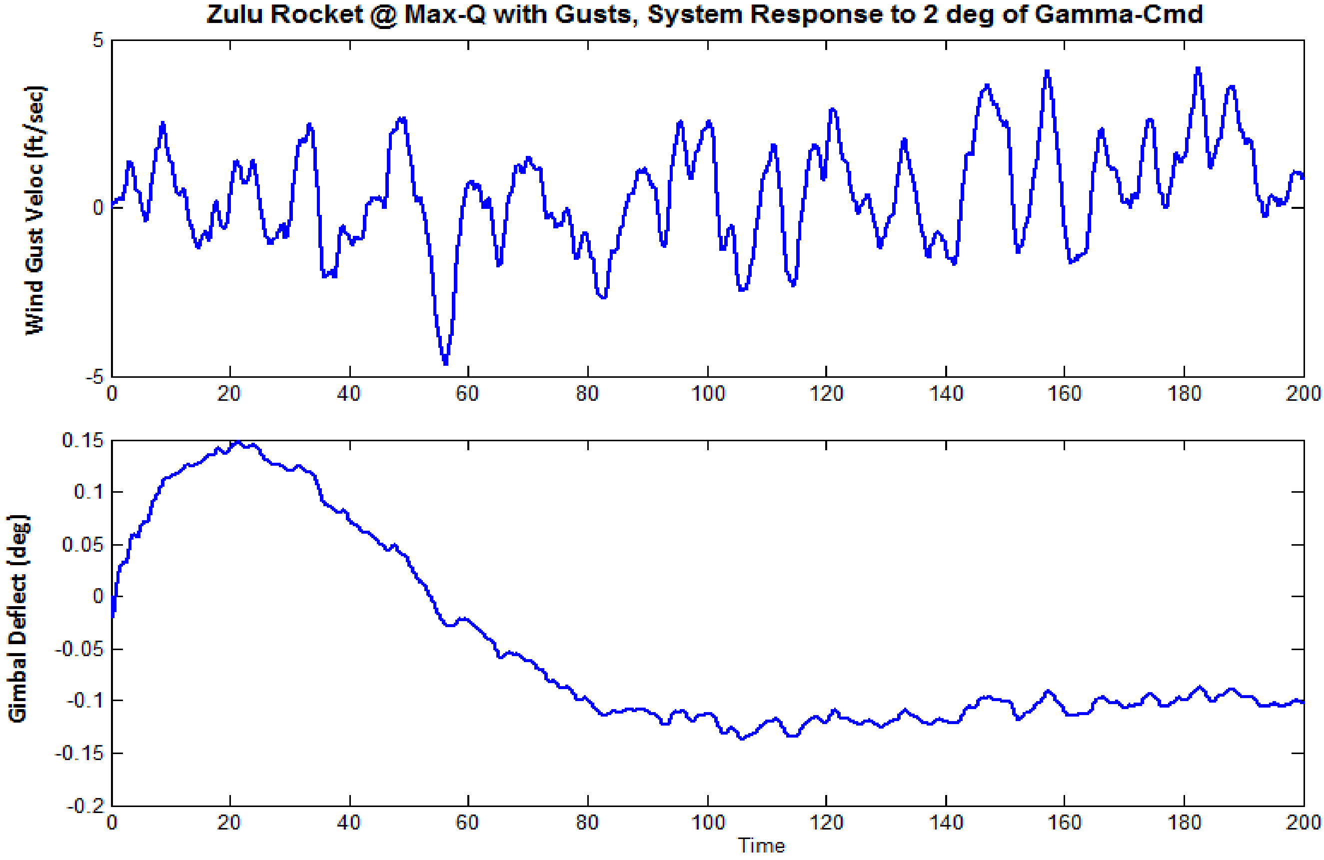


Figure 4.5 Closed-Loop Simulation Model in File "Ptch_Sim.Mdl"



Zulu Rocket @ Max-Q with Gusts, System Response to 2 deg of Gamma-Cmd



Zulu Rocket @ Max-Q with Gusts, System Response to 2 deg of Gamma-Cmd

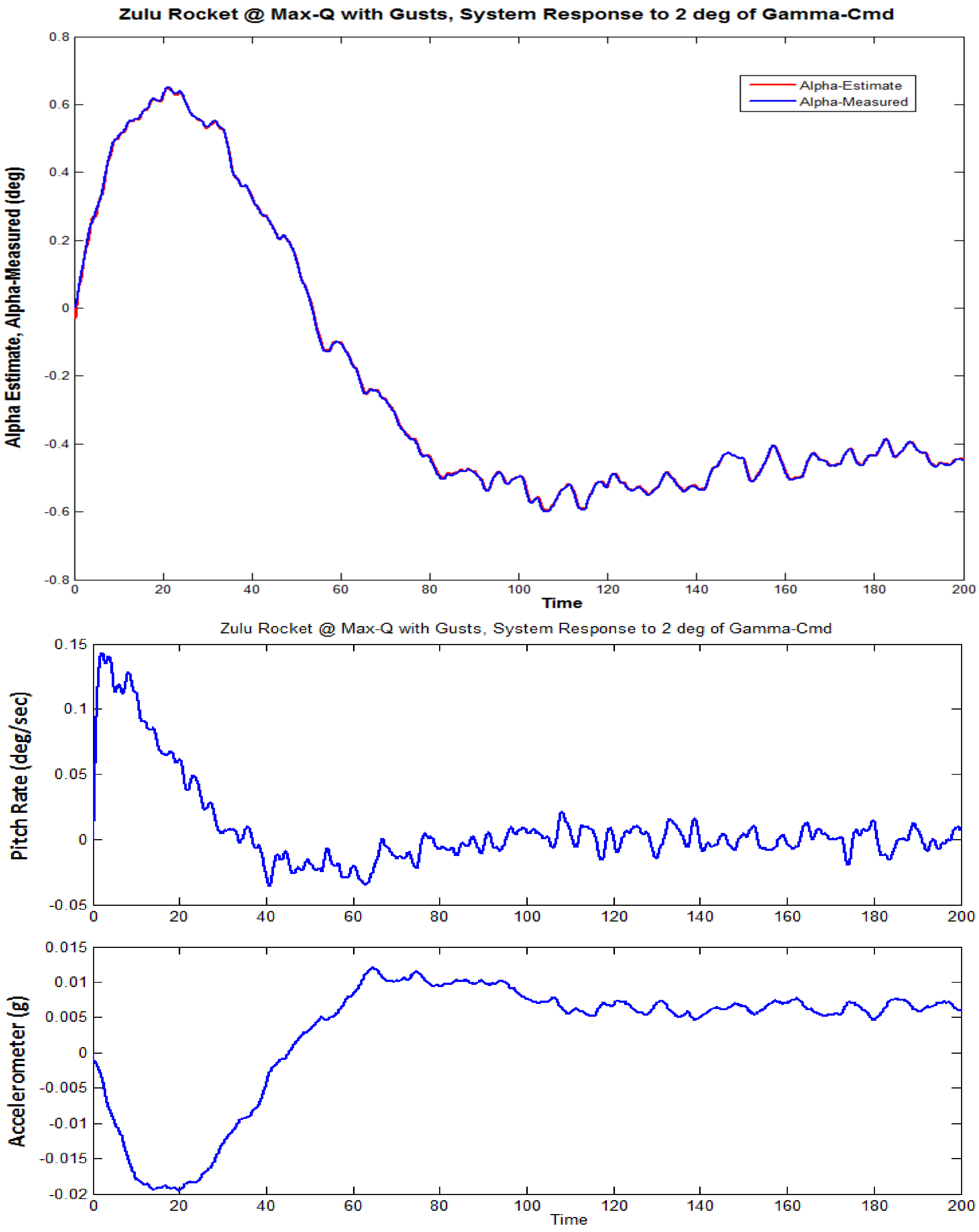


Figure 4.6 Pitch Simulation Response to Gamma Command with Noise using "Ptch_Sim.Mdl"

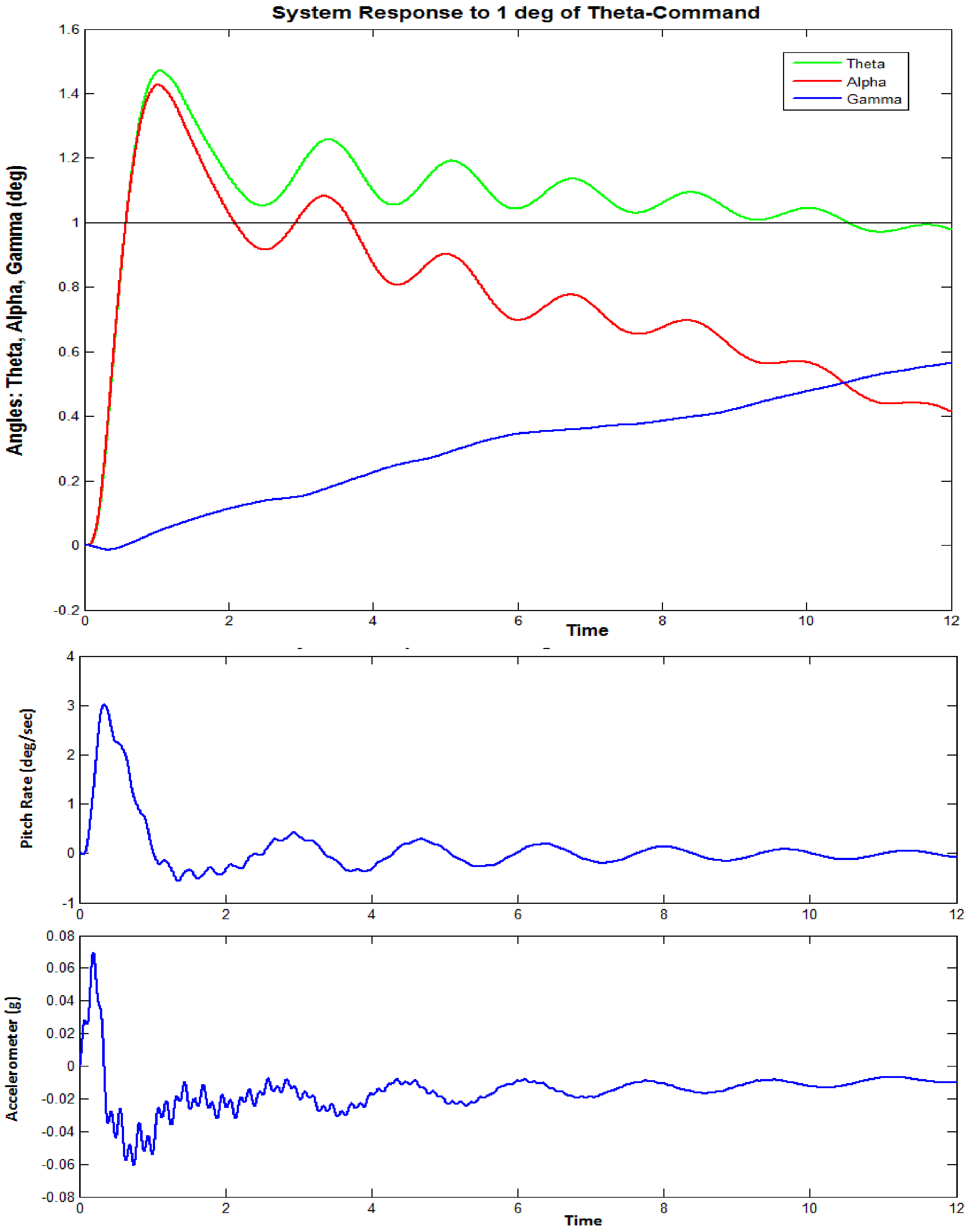


Figure 4.7 System Response to 1^{o} of Pitch Attitude Command

4.2.6 Creating the Models by Using Flixan Utilities

The open and closed loop systems used in the previous analysis were implemented using Simulink. The Simulink models were then linearized and converted into state-space form using the "Linmod" utility in Matlab. The Linmod utility, however, is not always reliable and in some situations it may create problems, especially, when there are direct feed-through terms in the model (although in this case and most of the times it works fine). In addition, Linmod introduces numerical errors. It is, therefore, not very efficient to have to rely on Simulink and Linmod for generating the state-space systems (either open or closed-loop) which are needed to analyze the control system performance. A better and more exact approach is to use the Flixan transfer function and system combination utilities to combine the state-space systems.

In this section we describe an alternative approach to combine the subsystems and to generate the open-loop and closed-loop systems using Flixan utilities. At the bottom of the pitch input data file "*Zulu_MaxQ_Ptch.Inp*" we have included some additional datasets. There are two transfer function datasets for the "*Alpha Estimator*" and the "*Pitch Flight Control System*", a dataset for an actuator model, and two systems combination datasets, a "*Pitch Open-Loop System*", and a "*Pitch Closed-Loop System*". There is also a batch set at the top of file "*Zulu-MaxQ-Ptch.Inp*" that will process the data, generate the systems, and save them in the pitch systems file "*Zulu-MaxQ-Ptch.Qdr*". The "*Pitch Closed-Loop System*" set combines the flex vehicle model "*Zulu, Stage-1, Max-Q, 8 Flex Modes with Slosh, Pitch Axis*" with the alpha-estimator, the actuator, and the pitch flight control system in a single state-space system. The "*Pitch Open-Loop System*" set combines the same subsystems for open-loop frequency response analysis. The two systems are finally exported as function m-files: "closed-loop.m" and "open-loop.m" and they are loaded into Matlab.

The Simulink model "*Ptch_Sim_4.Mdl*" in Figure 4.7 uses the entire Flixan generated closed-loop system in a single block. It performs a time-slice simulation and produces results which are identical to the previous Simulink analysis in Fig. 4.6, where the subsystems were combined together in the Simulink model "*Ptch_Sim.Mdl*".

Similarly, the Simulink model "*Ptch_Open4.Mdl*" includes the open-loop system "*Pitch Open-Loop System*" and it is used to calculate the open-loop frequency response in Matlab. However, the frequency response results shown in Figure 4.4 were calculated using the Flixan frequency response program, directly from the system "*Pitch Open-Loop System*" in file "*Zulu_MaxQ_Ptch.Qdr*", without using Matlab.

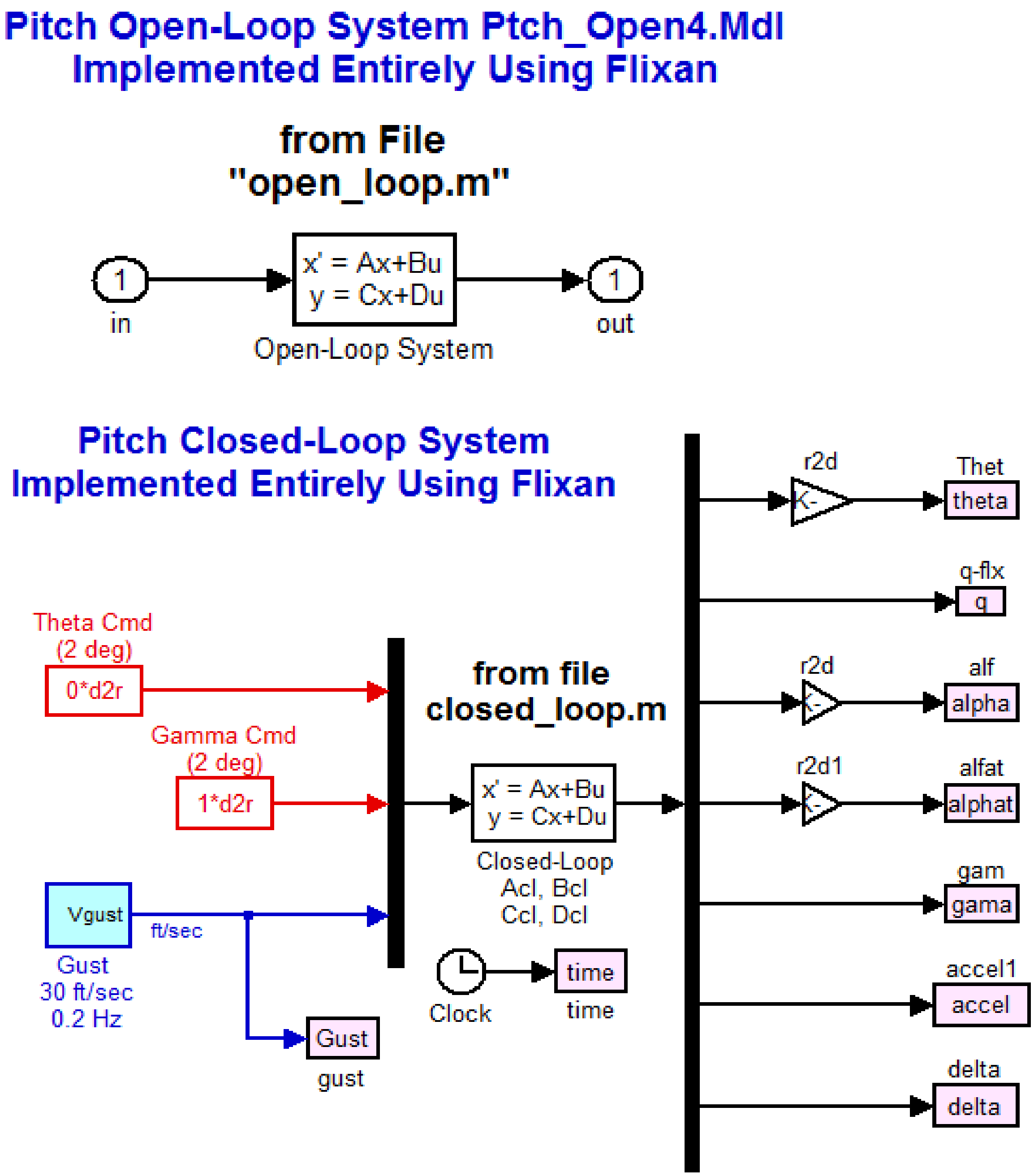


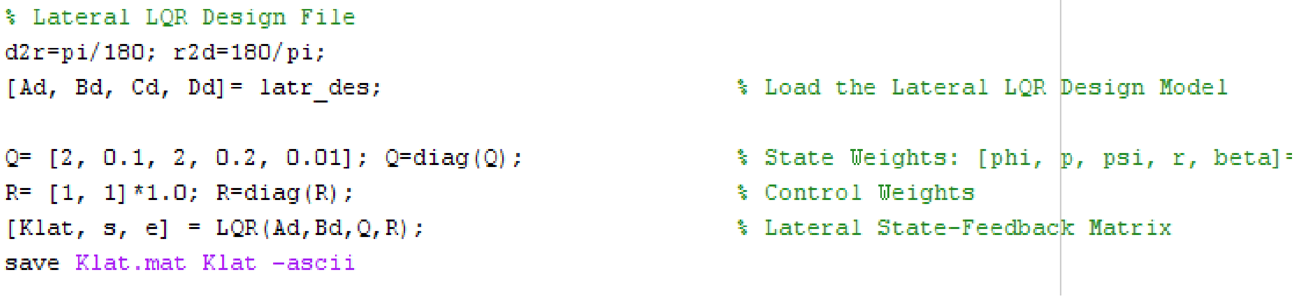
Figure 4.8 Open and Closed-Loop State Systems Implemented Using Flixan. They are included in Simulink Files Ptch_Open4.Mdl and Ptch_Sim_4.Mdl respectively

4.3.1 Lateral Axes Analysis

The lateral system consists of the roll and yaw dynamics coupled together. The dynamic models are created by executing the batch set “*Batch for calculating the Lateral Models for the Zulu Rocket*” that processes the input file “*Zulu_MaxQ_Latr.Inp*”, and saves the systems in file “*Zulu-MaxQ-Latr.Qdr*”. The dynamic model “*Zulu, Stage-1, Max-Q, 7 Flex Modes with Slosh*” includes the seven lateral modes that were selected from file: “*Zulu_MaxQ.Mod*”. The title of the selected set of modes is: “*Zulu, Stage-1, Max-Q, 7 Flex Modes, Roll Modes*”. The lateral state-space system was extracted from the systems file and saved under the title “*Zulu, Stage-1, Max-Q, 7 Flex Modes with Slosh, Roll & Yaw Axes*”. It was converted to a Matlab function using the “*Exporting to Matlab*” utility and saved as a separate system m-file “*Vehi_Latr_Sflx.m*” in subdirectory “*Twin Booster Rocket\4-Flex-Slosh Analysis\Lateral Mat*”. This model will be used in the control analysis using Matlab. A design model was also created for calculating the state-feedback gain matrix Klat. The design model title is “*Lateral Design Model*” and it was saved in an m-file “*Latr_des.m*”.

4.3.2 Flight Control System

The lateral control system is also a state-feedback having two control loops that independently control yaw and roll attitude by means of the two boosters TVC. It is a (2 x 5) state-feedback gain matrix (Klat) that was designed using the LQR method by running the Matlab script file “*LQR_des.m*”, see below. It feeds-back from the five states (ϕ, p, ψ, r, and β) and calculates the roll and yaw acceleration demands. This is almost identical to the design process in Section 2. The demands drive the (4 x 2) mixing logic matrix (Kmix) that regulates the TVC deflections in pitch and yaw. Notice, that when we use the LQR method to calculate the state-feedback matrix Klat, the “Lateral Design Model” consists of the rigid-body vehicle with the matrix Kmix in series. The design plant, therefore, has only two inputs and it contains the five rigid states.



```
% Lateral LQR Design File
d2r=pi/180; r2d=180/pi;
[Ad, Bd, Cd, Dd]= latr_des;                         % Load the Lateral LQR Design Model

Q= [2, 0.1, 2, 0.2, 0.01]; Q=diag(Q);               % State Weights: [phi, p, psi, r, beta]=
R= [1, 1]*1.0; R=diag(R);                           % Control Weights
[Klat, s, e] = LQR(Ad,Bd,Q,R);                      % Lateral State-Feedback Matrix
save Klat.mat Klat -ascii
```

4.3.3 Mixing Logic Matrix

The mixing logic matrix Kmix is calculated by the Flixan mixing logic algorithm in batch mode. The algorithm is called to process the mixing-logic dataset “*Lateral Mixing Logic Matrix for the Twin Engine Zulu Rocket*”. It reads the rigid vehicle model “*Zulu, Stage-1, Max-Q, Rigid Body*” from the input file, generates the Kmix matrix, and saves it in the lateral systems file. Kmix attempts to decouple the roll and yaw dynamics based on vehicle geometry and thrusts. It transforms the roll and yaw control system demands to pitch (δy) and yaw (δz) gimbal deflections. It decouples the lateral system and shapes the roll and yaw accelerations to be equal to the accelerations demanded by the flight control system. The two roll and yaw flight control directions to be decoupled are specified in the mixing-logic dataset. Pitch is not included.

4.3.4 Beta Estimator/ Filters

The lateral flight control system includes a beta estimator, Figure 4.9, which is similar to the alpha estimator described in section 4.2.2. It estimates the angle of sideslip by solving the lateral force equation and using the lateral accelerometer measurement Ny, and the yaw gimbal deflections from the two engines (δz_1 & δz_2). It includes a compensator from the yaw rate gyro to cancel the rotational acceleration signal measured by the accelerometer, similar to the α-estimator. The beta estimate replaces the actual beta in the state feedback. The remaining states used for state-feedback are directly measurable. Some additional vehicle parameters are also loaded into Matlab for the beta-estimator. Additional low pass and notch filters were included in the flight control loops, and notch filters in the lateral acceleration input to the β-estimator because there is a strong flexibility present in that signal. There is also a lateral slosh compensator.

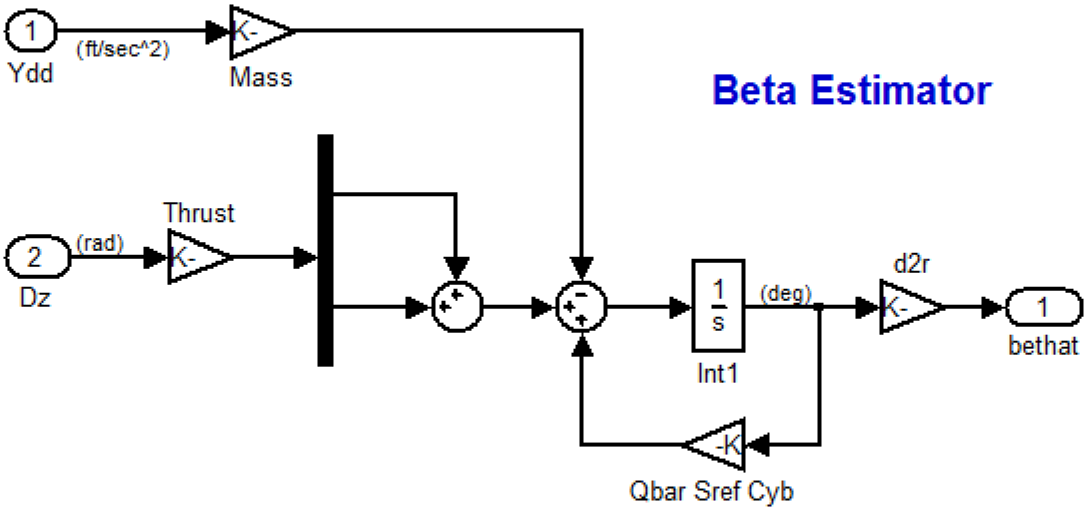
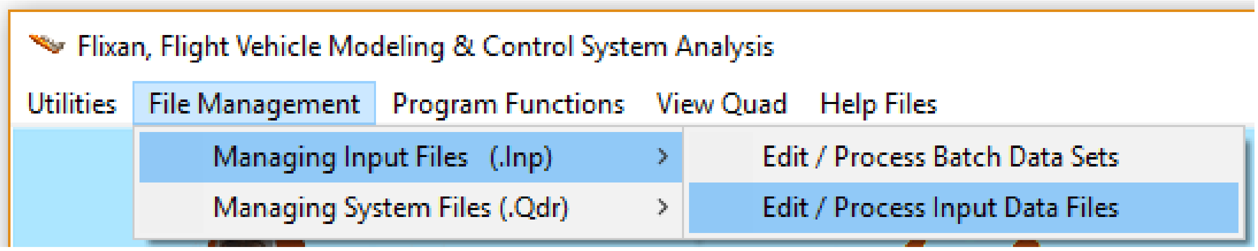


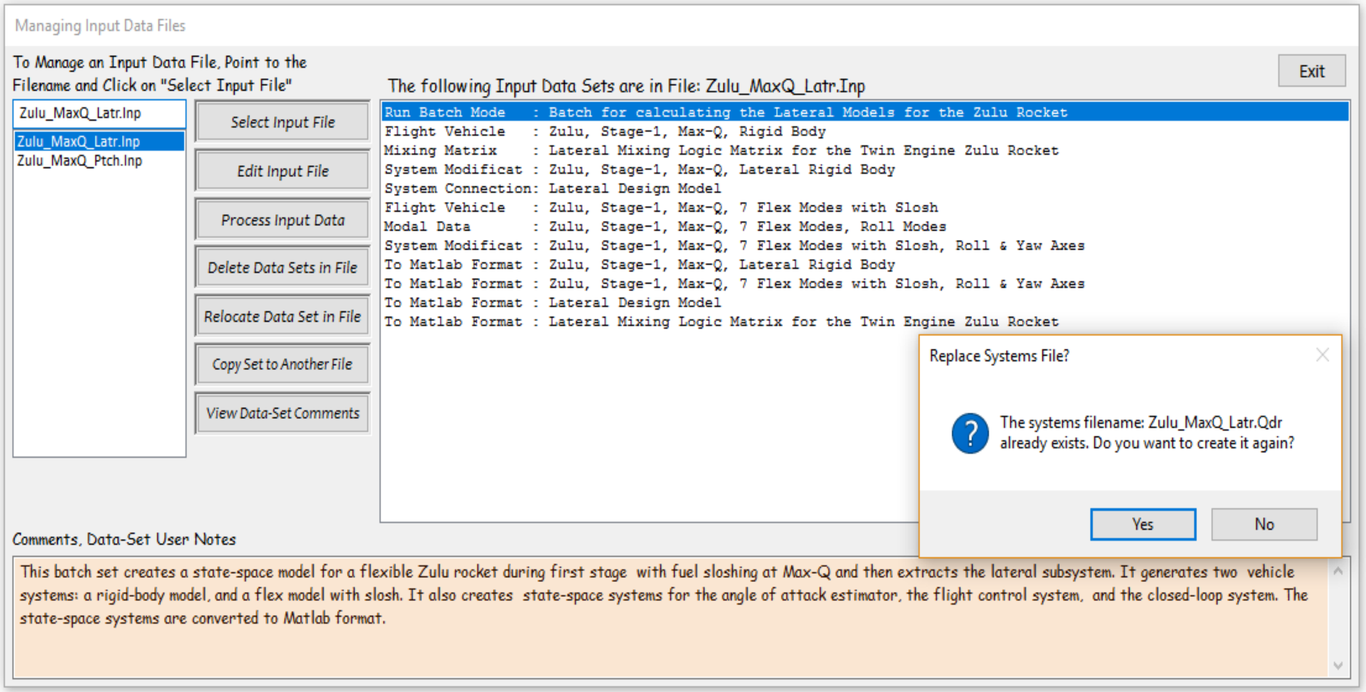
Figure 4.9 Angle of Sideslip Estimator

4.3.5 Processing the Data File in Batch and Exporting Systems to Matlab

The lateral input file "*Zulu_MaxQ_Latr.Inp*" includes several datasets to be processed by Flixan. They can be processed in batch mode by executing the batch set which is located at the top of the file. The title of the batch set is "*Batch for calculating the Lateral Models for the Zulu Rocket*".

It will process the vehicle data, generate the rigid and flex models, modify the systems to create the lateral LQR design model, create the mixing-logic matrix, a more comprehensive actuator model, and it will save the systems and matrix in file "*Zulu_MaxQ_Latr*.Qdr". It will also convert and export those systems to the Matlab analysis folder "*Twin Booster Rocket**4-Flex-Slosh Analysis**Lateral Mat*". The batch can be processed from the File Manager utility by selecting "*Edit/ Process Input Data Files*", as shown below.





To Manage an Input Data File, Point to the Filename and Click on "Select Input File"

Zulu_MaxQ_Latr.Inp

Zulu_MaxQ_Latr.Inp
Zulu_MaxQ_Ptch.Inp

The following Input Data Sets are in File: Zulu_MaxQ_Latr.Inp

```
Run Batch Mode    : Batch for calculating the Lateral Models for the Zulu Rocket
Flight Vehicle    : Zulu, Stage-1, Max-Q, Rigid Body
Mixing Matrix     : Lateral Mixing Logic Matrix for the Twin Engine Zulu Rocket
System Modificat  : Zulu, Stage-1, Max-Q, Lateral Rigid Body
System Connection: Lateral Design Model
Flight Vehicle    : Zulu, Stage-1, Max-Q, 7 Flex Modes with Slosh
Modal Data        : Zulu, Stage-1, Max-Q, 7 Flex Modes, Roll Modes
System Modificat  : Zulu, Stage-1, Max-Q, 7 Flex Modes with Slosh, Roll & Yaw Axes
To Matlab Format  : Zulu, Stage-1, Max-Q, Lateral Rigid Body
To Matlab Format  : Zulu, Stage-1, Max-Q, 7 Flex Modes with Slosh, Roll & Yaw Axes
To Matlab Format  : Lateral Design Model
To Matlab Format  : Lateral Mixing Logic Matrix for the Twin Engine Zulu Rocket
```

Replace Systems File?

The systems filename: Zulu_MaxQ_Latr.Qdr already exists. Do you want to create it again?

Comments, Data-Set User Notes

This batch set creates a state-space model for a flexible Zulu rocket during first stage with fuel sloshing at Max-Q and then extracts the lateral subsystem. It generates two vehicle systems: a rigid-body model, and a flex model with slosh. It also creates state-space systems for the angle of attack estimator, the flight control system, and the closed-loop system. The state-space systems are converted to Matlab format.

The input file manager dialog comes up, and from the menu on the left side select the input file "*Zulu_MaxQ_Latr.Inp*" and click on "*Select Input File*". The menu on the right shows the titles of the datasets that are saved in this input file. Select the first one which is the batch set and click on "*Process Input Data*". Click on "Yes" to overwrite the previous data. The batch will then process the datasets which are in this file and save the systems in file "*Zulu_MaxQ_Latr.Qdr*". It will also save the mixing matrix in file Kmix, the actuator in file "actuator.m", and the simulation and design vehicle models in files "*vehi_latr_sflx.m*" and "*latr_des.m*".

These systems must go in the Matlab analysis subdirectory "*Lateral Mat*". The vehicle systems and the matrices Kmix and Klat are loaded into Matlab for analysis by the initialization file "run.m". The state-feedback matrix Klat transforms the 5 state-feedback errors to roll and yaw acceleration demands, and Kmix transforms the two acceleration demands to 4 TVC deflections (2-pitch and 2-yaw).

4.3.6 Lateral Axes Simulation

Figure 4.10 shows the lateral axes simulation model. The Simulink file name is “*Latr_Sim.Mdl*”, and it is located in subdirectory “*Lateral Mat*”. This model analyzes the launch vehicle’s performance in the presence of wind gusts and its response to roll and yaw attitude commands. The wind gust is a noisy signal limited in bandwidth to 0.2 Hz, and in amplitude to 25 (feet/sec). The wind direction relative to the vehicle is defined in the vehicle input data set.

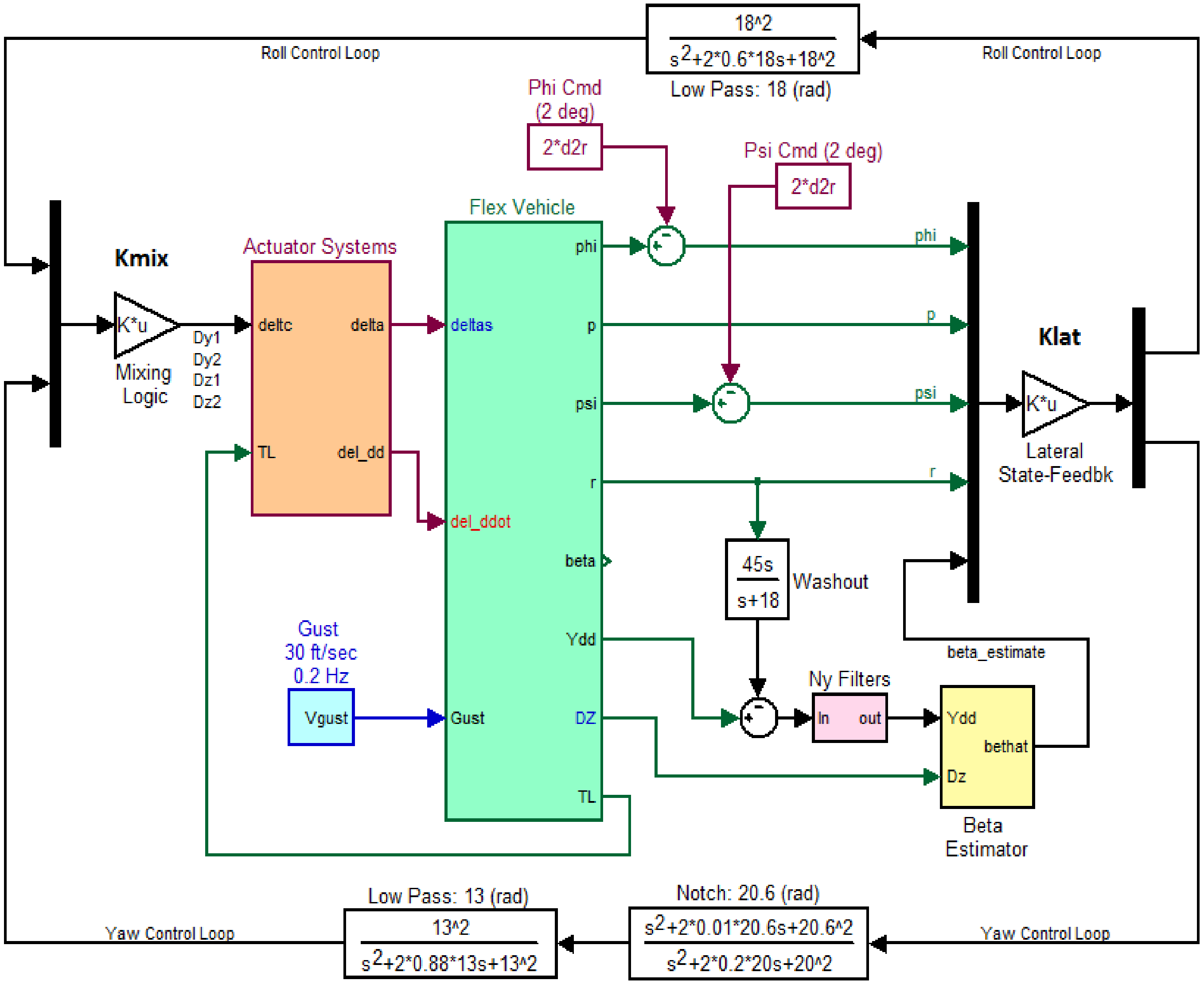


Figure 4.10 Roll and Yaw Simulation Model “*Latr_Sim.Mdl*”

Figure 4.11 shows the actuator system which consists of 4 actuators, 2 for pitch and 2 for yaw deflections. The actuator model was generated by Flixan and it was loaded into Matlab as a state-space system from file “actuator.m”. Each unit has two outputs: δ and $\ddot{\delta}$. The acceleration output excites the TWD dynamics in the vehicle. In addition to the deflection command each actuator includes a load-torque input that couples it with the corresponding load-torque output from the vehicle model, as shown in Figure 4.9.

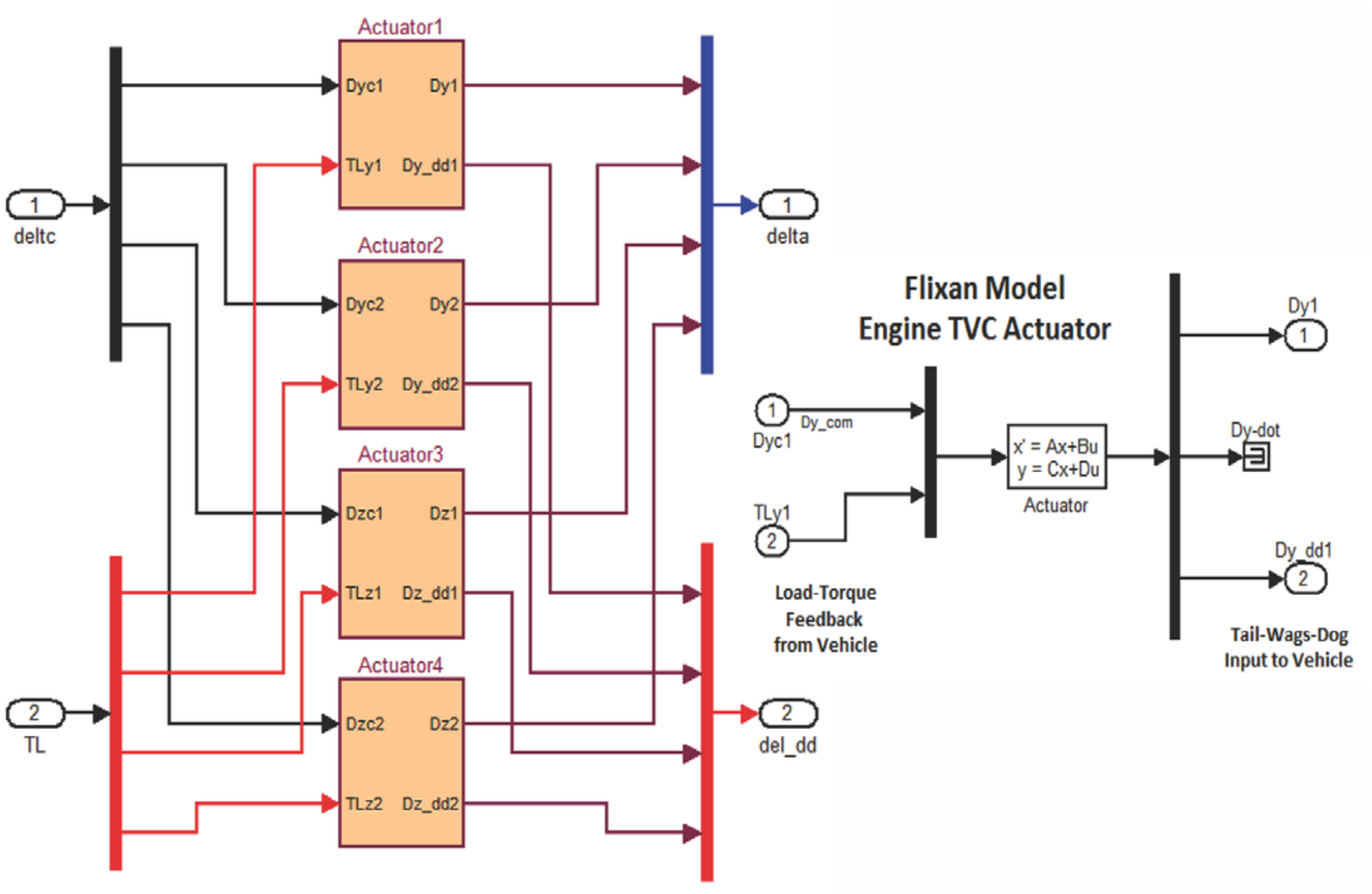


Figure 4.11 Actuator System Consisting of four Actuators, two per TVC Engine.

Note that the actuator model is stiff. That is, the backup and load stiffnesses are not introduced in the model but they are set infinitely high because they are included in the vehicle model, since the vehicle model is flexible and the flexibility at the gimbals is captured by the mode shapes. Only the actuator shaft stiffness is included in the actuator model.

When the simulation is complete the script file “pl.m” is used to plot the data. The simulation plots are shown in Figures 4.12 and 4.13. In this case the roll and yaw attitudes are simultaneously commanded to a 2^o step up. After an initial transient, the beta estimate converges to the sideslip angle. The small oscillations in the rates and gimbals are caused by the propellant sloshing. The Ny accelerometer is also sensing that disturbance and it is processed by the control system. The gimbal angles show a differential pitch deflection of the two engines that is needed for the roll maneuver.

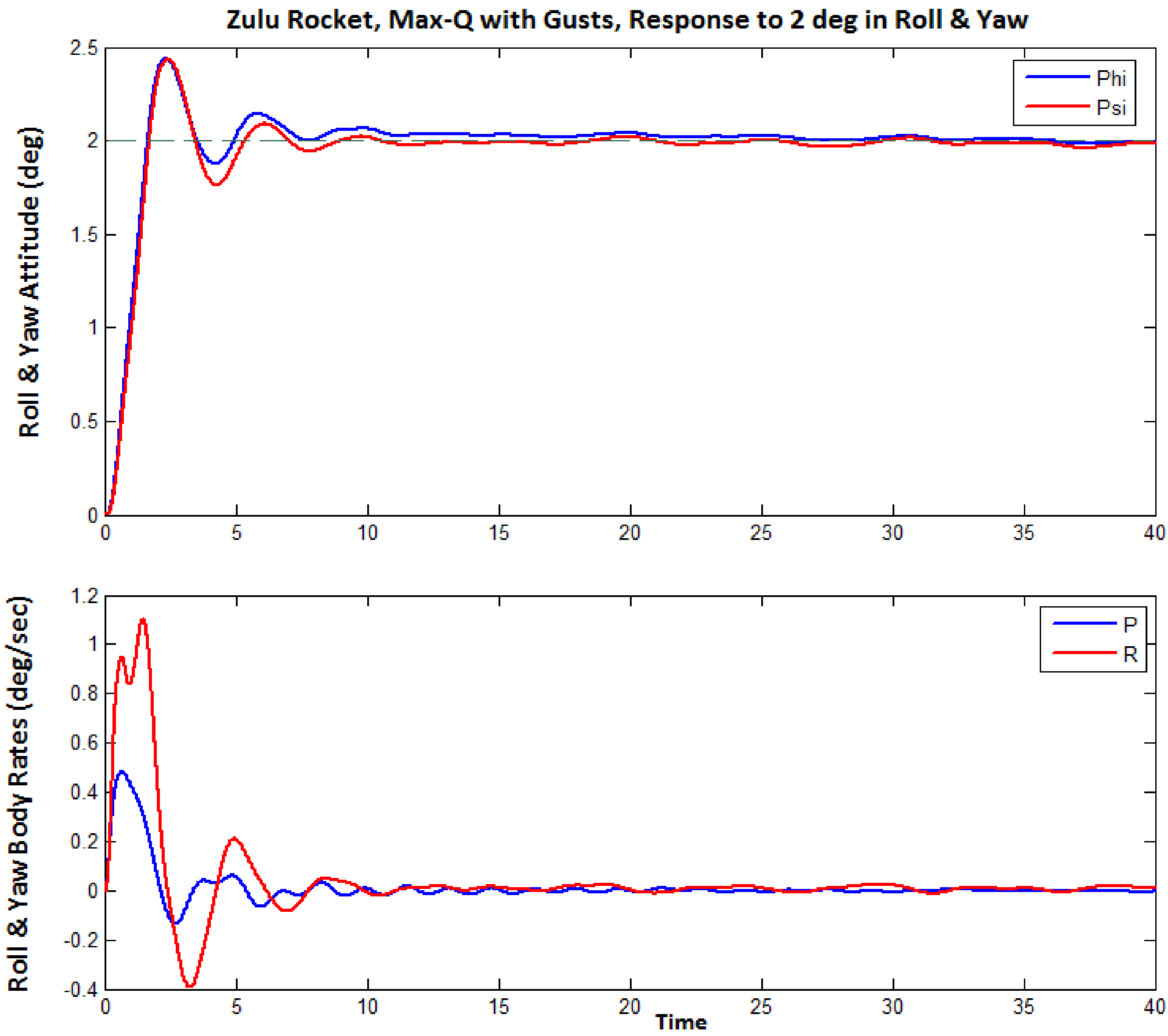


Figure 4.12 Roll and Yaw Attitude Response to Two Simultaneously Applied 2^o Roll and Yaw Commands

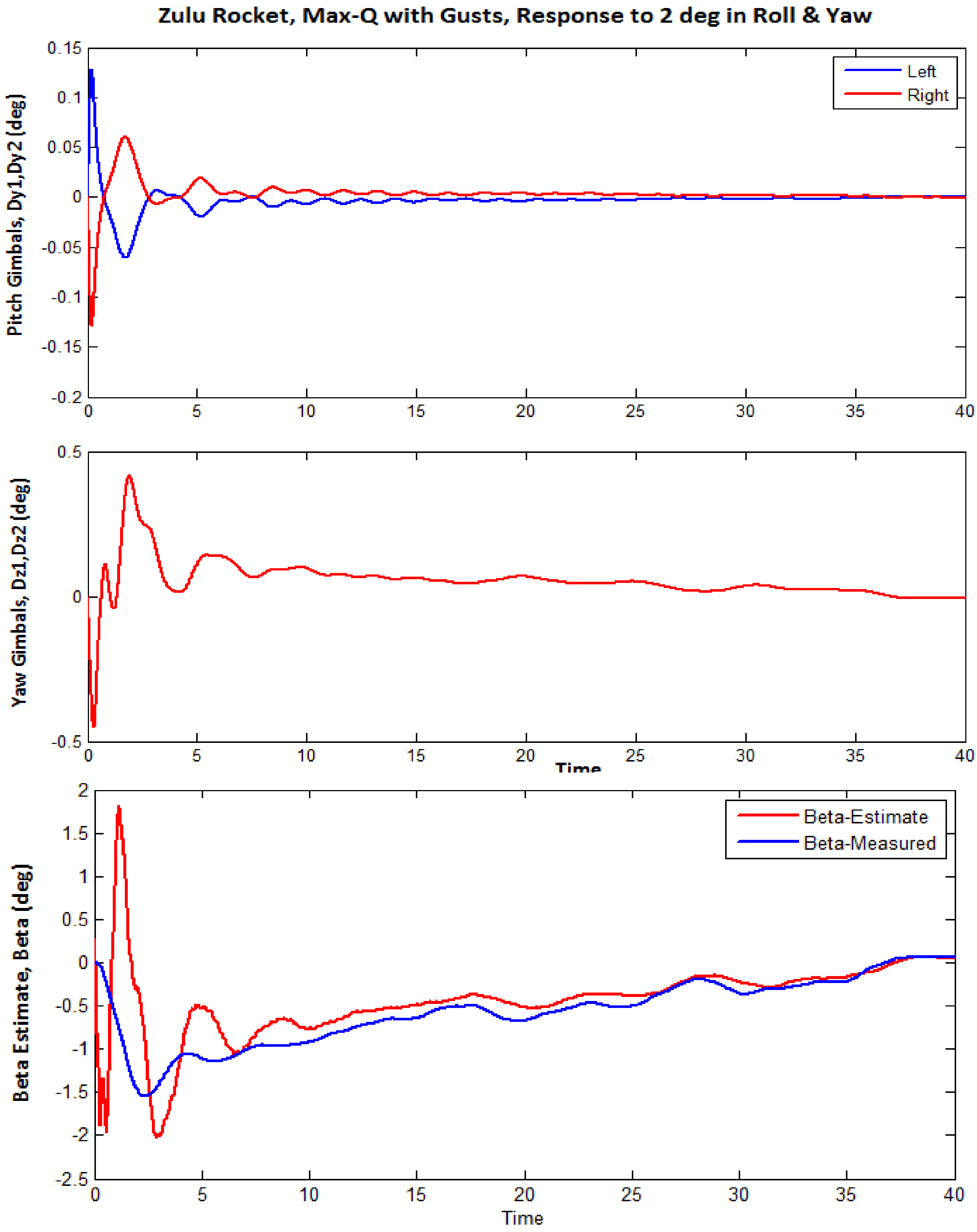
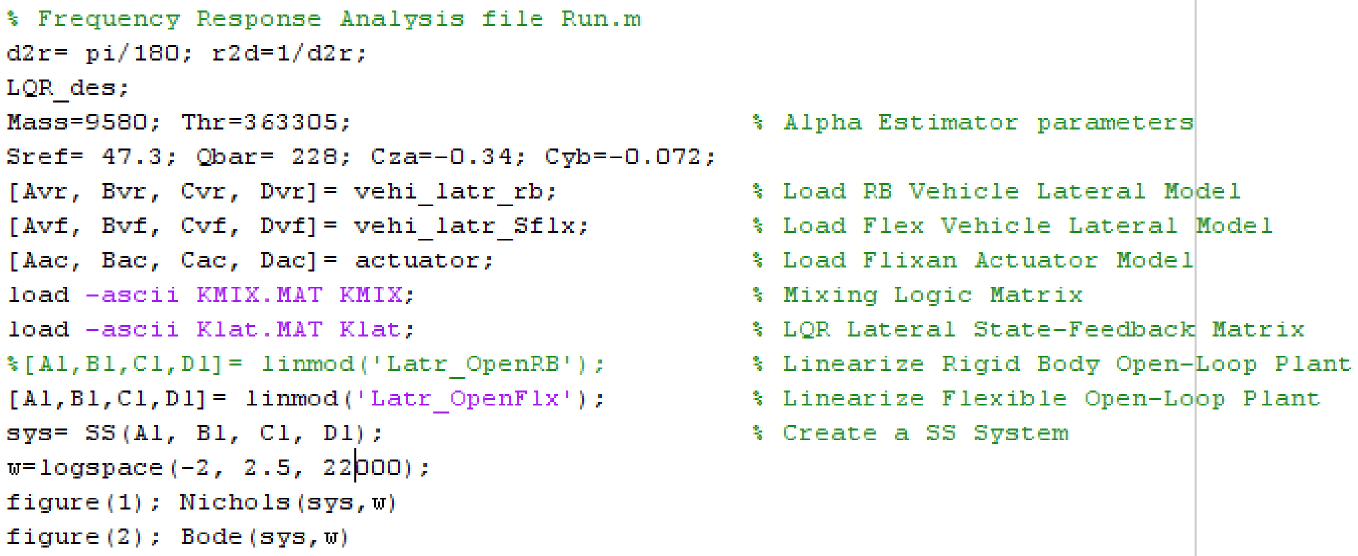


Figure 4.13 Pitch and Yaw Engine Gimbal Deflections and Beta Estimate versus Beta Measurement

4.3.7 Stability Analysis

Figure 4.14 shows the Simulink model "*Latr_OpenFLX.mdl*" that is used for open-loop stability analysis. The file "run.m" loads the systems and matrices and calculates the frequency response. It calculates the Nichol's and Bode plots for the roll and yaw control loops for measuring the phase and gain margins. When one loop is analyzed (roll in this case) the other loop (yaw) must be closed, as shown. The flex vehicle subsystem (green block) is using the lateral vehicle model "*Zulu, Stage-1, Max-Q, 7 Flex Modes with Slosh, Roll & Yaw Axes*" that was saved in file "*vehi_latr_sflx.m*".



```
% Frequency Response Analysis file Run.m
d2r= pi/180; r2d=1/d2r;
LQR_des;
Mass=9580; Thr=363305;                              % Alpha Estimator parameters
Sref= 47.3; Qbar= 228; Cza=-0.34; Cyb=-0.072;
[Avr, Bvr, Cvr, Dvr]= vehi_latr_rb;                 % Load RB Vehicle Lateral Model
[Avf, Bvf, Cvf, Dvf]= vehi_latr_Sflx;               % Load Flex Vehicle Lateral Model
[Aac, Bac, Cac, Dac]= actuator;                     % Load Flixan Actuator Model
load -ascii KMIX.MAT KMIX;                          % Mixing Logic Matrix
load -ascii Klat.MAT Klat;                          % LQR Lateral State-Feedback Matrix
%[A1,B1,C1,D1]= linmod('Latr_OpenRB');              % Linearize Rigid Body Open-Loop Plant
[A1,B1,C1,D1]= linmod('Latr_OpenFlx');              % Linearize Flexible Open-Loop Plant
sys= SS(A1, B1, C1, D1);                            % Create a SS System
w=logspace(-2, 2.5, 22000);
figure(1); Nichols(sys,w)
figure(2); Bode(sys,w)
```

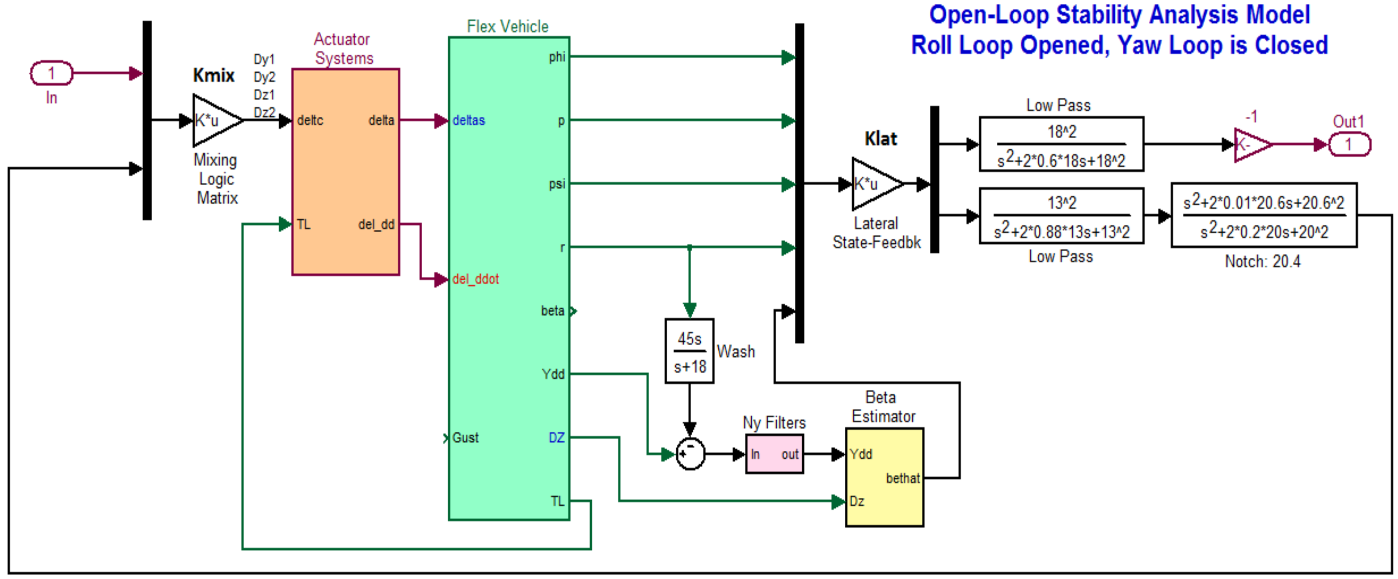


Figure 4.14 Simulink Model "*Latr_OpenFLX.mdl*" for Calculating Bode and Nichols Plots to Analyze the Lateral System Stability

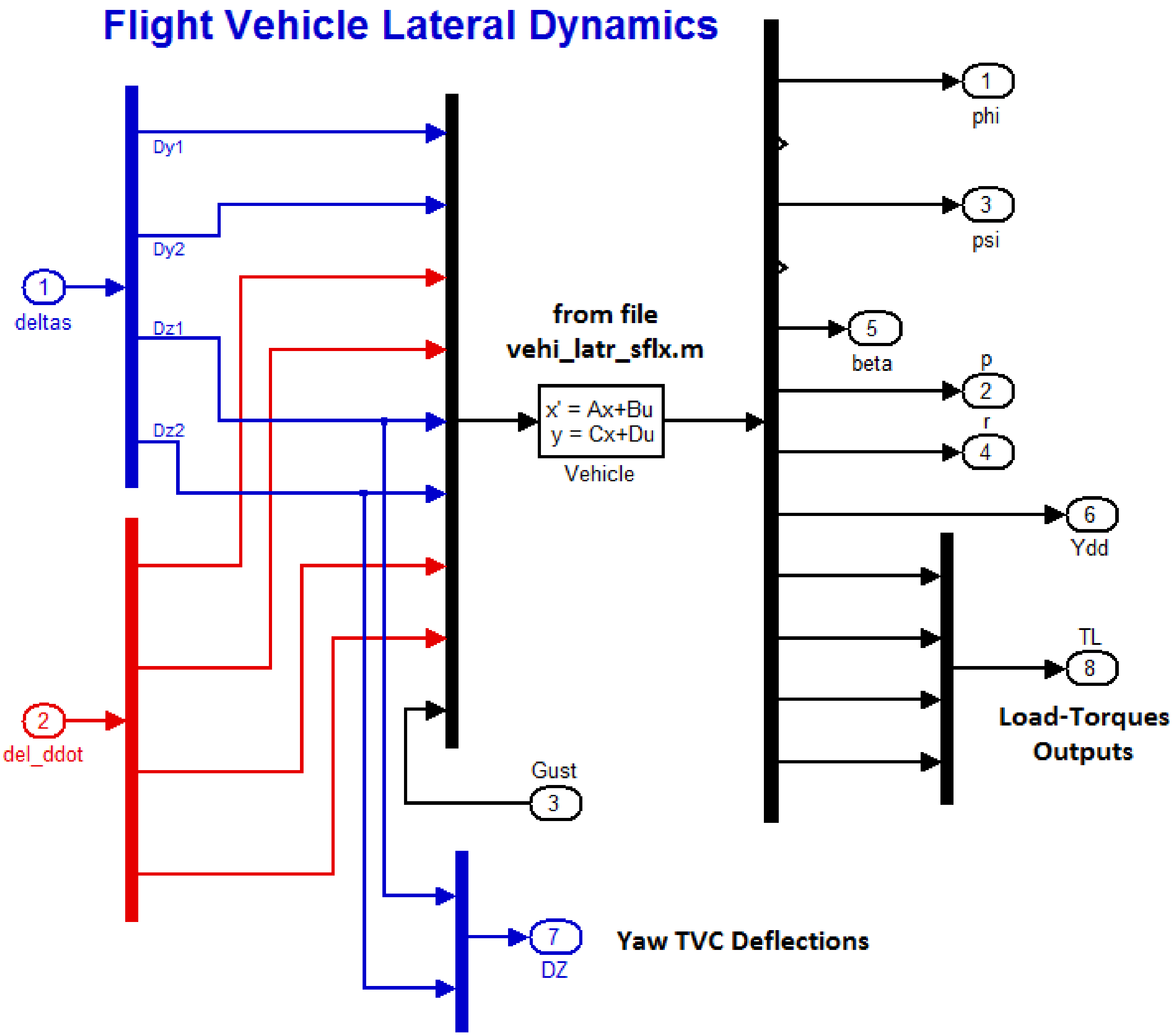


Figure 4.15 The Flixan Generated Flexible Vehicle Model is loaded in this Simulink Block as a State-Space System

Figure 4.15 shows the vehicle subsystem and its interconnections with the actuator subsystems. The Nichols plots in Figures 4.16 and 4.17 show the stability margins of the vehicle in roll and yaw. The roll Nichols was calculated from the model shown in Figure 4.14. The slosh mode at 3.9 (rad/sec) is very strong but it is phase stable. The structural modes are well attenuated. The yaw Nichols was obtained similarly by opening the yaw loop and closing roll. Both loops have sufficient phase and gain margin. Notch filtering was necessary in order to achieve a satisfactory modal attenuation. The slosh resonance in yaw is also strong and phase stabilized.

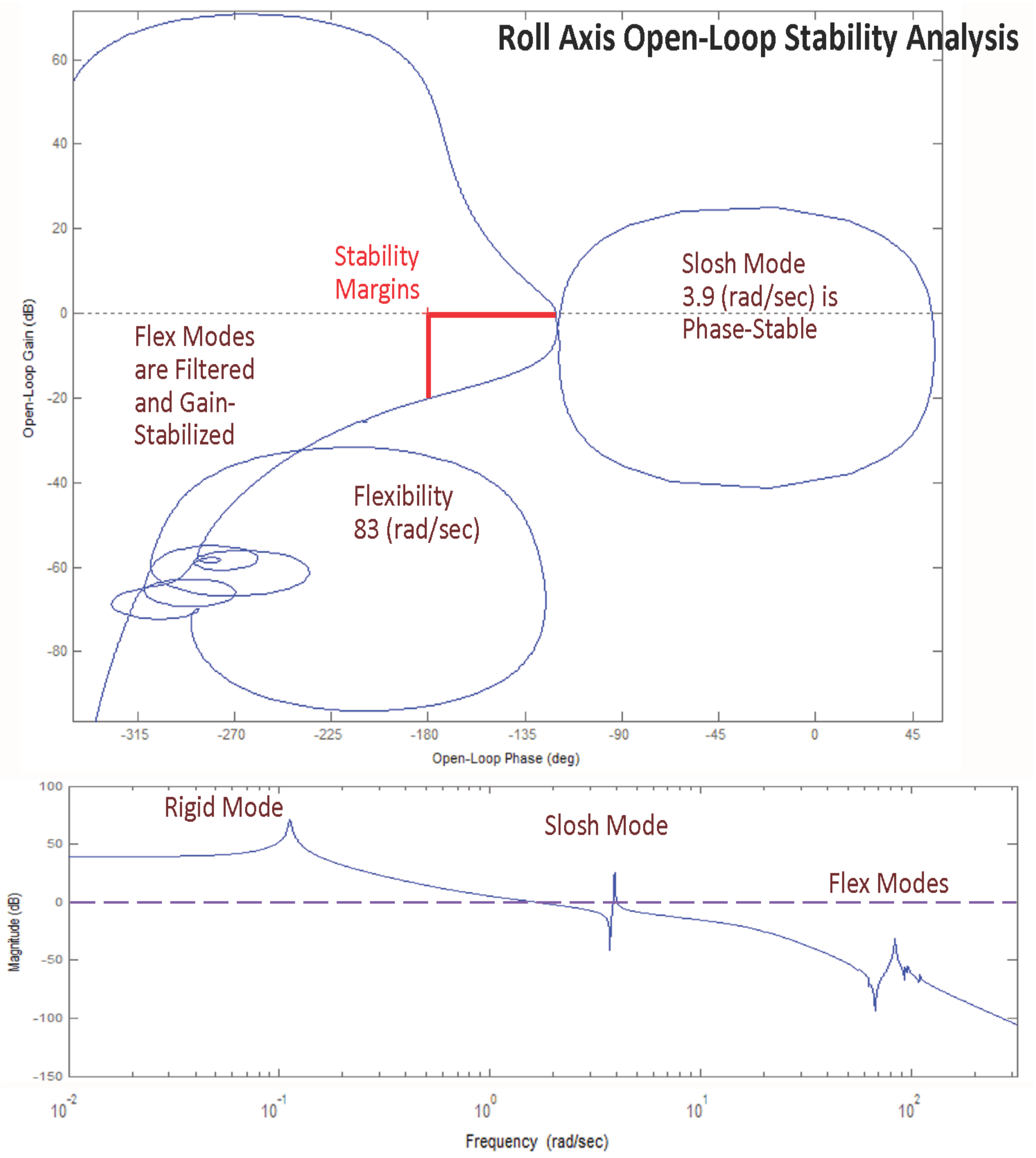


Figure 4.16 Roll Axis Stability Analysis Using Bode and Nichols Plots

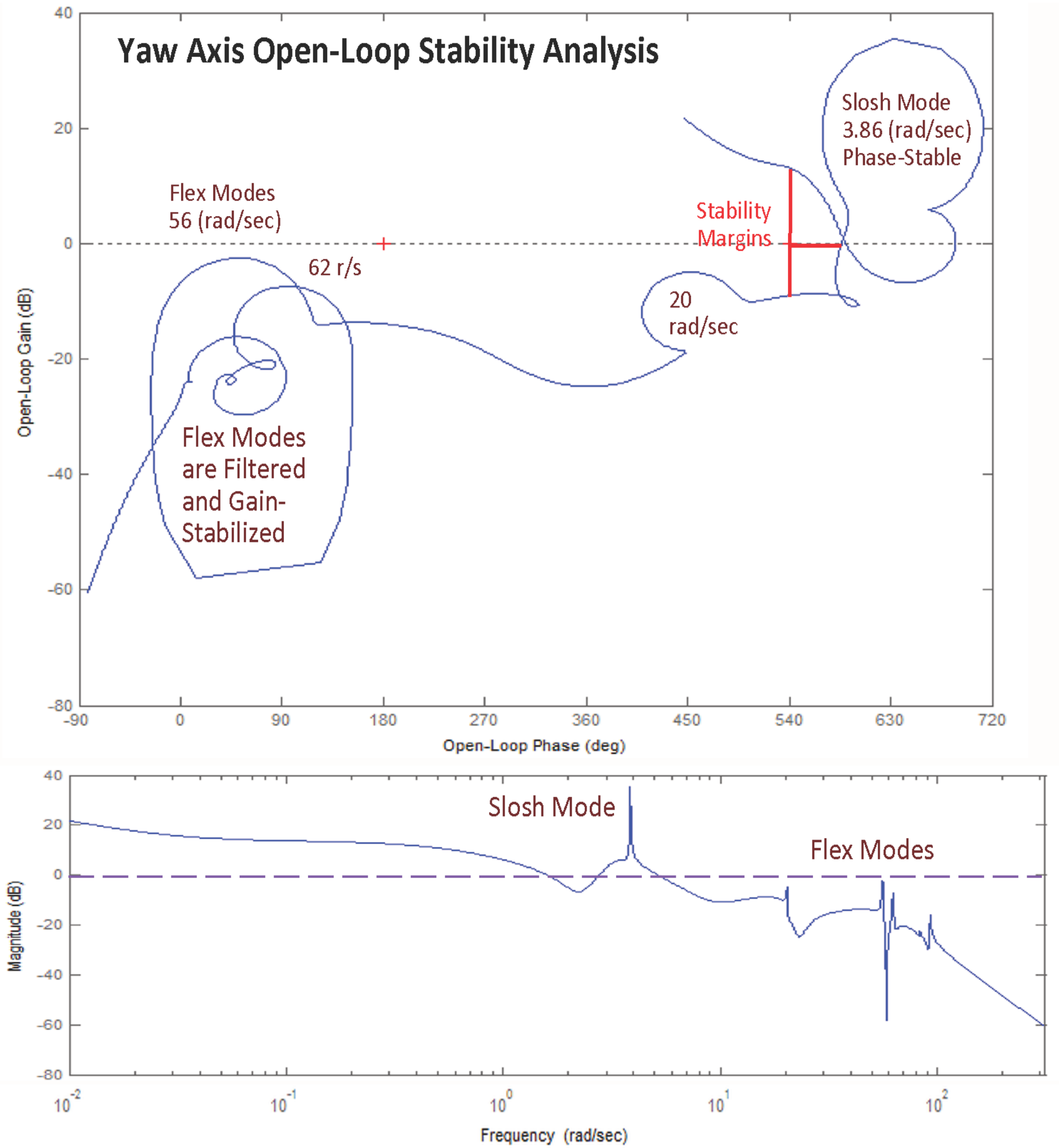


Figure 4.17 Yaw Axis Stability Analysis Using Bode and Nichols Plots