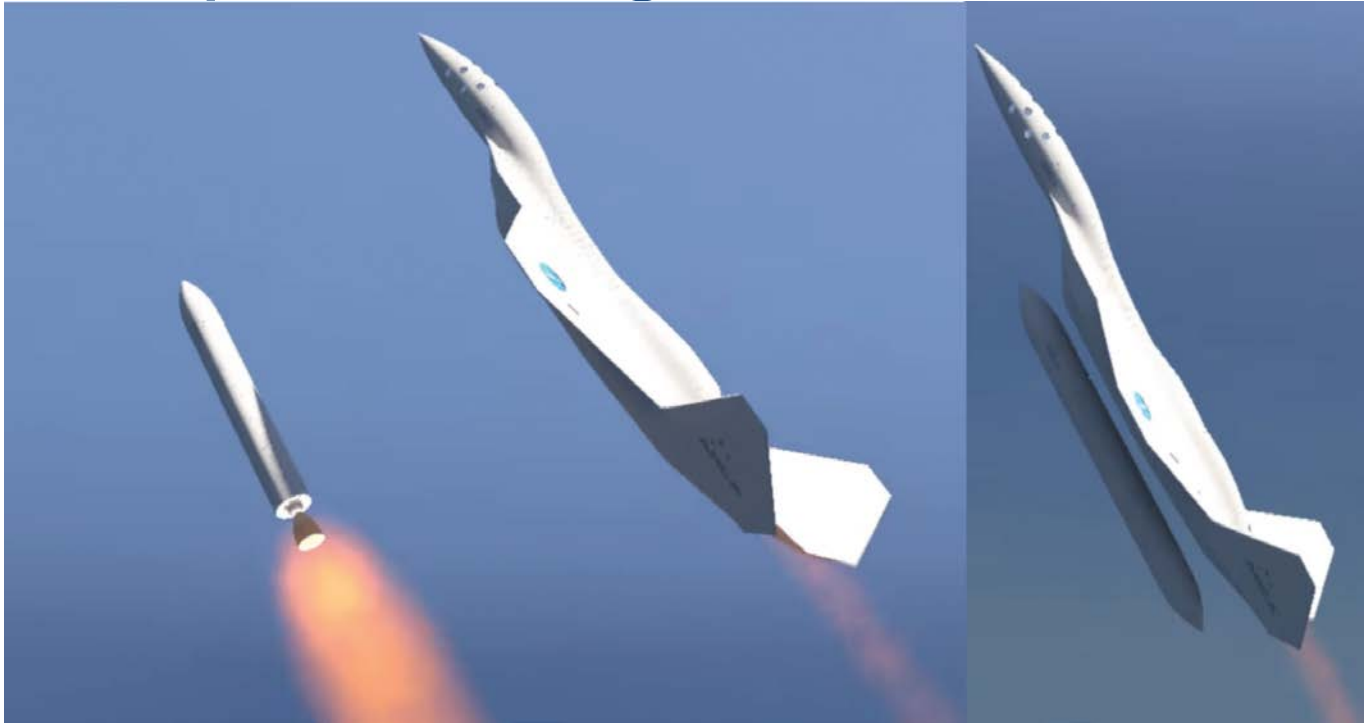


Air Launched Booster to Orbit Complete Design and Simulation



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.

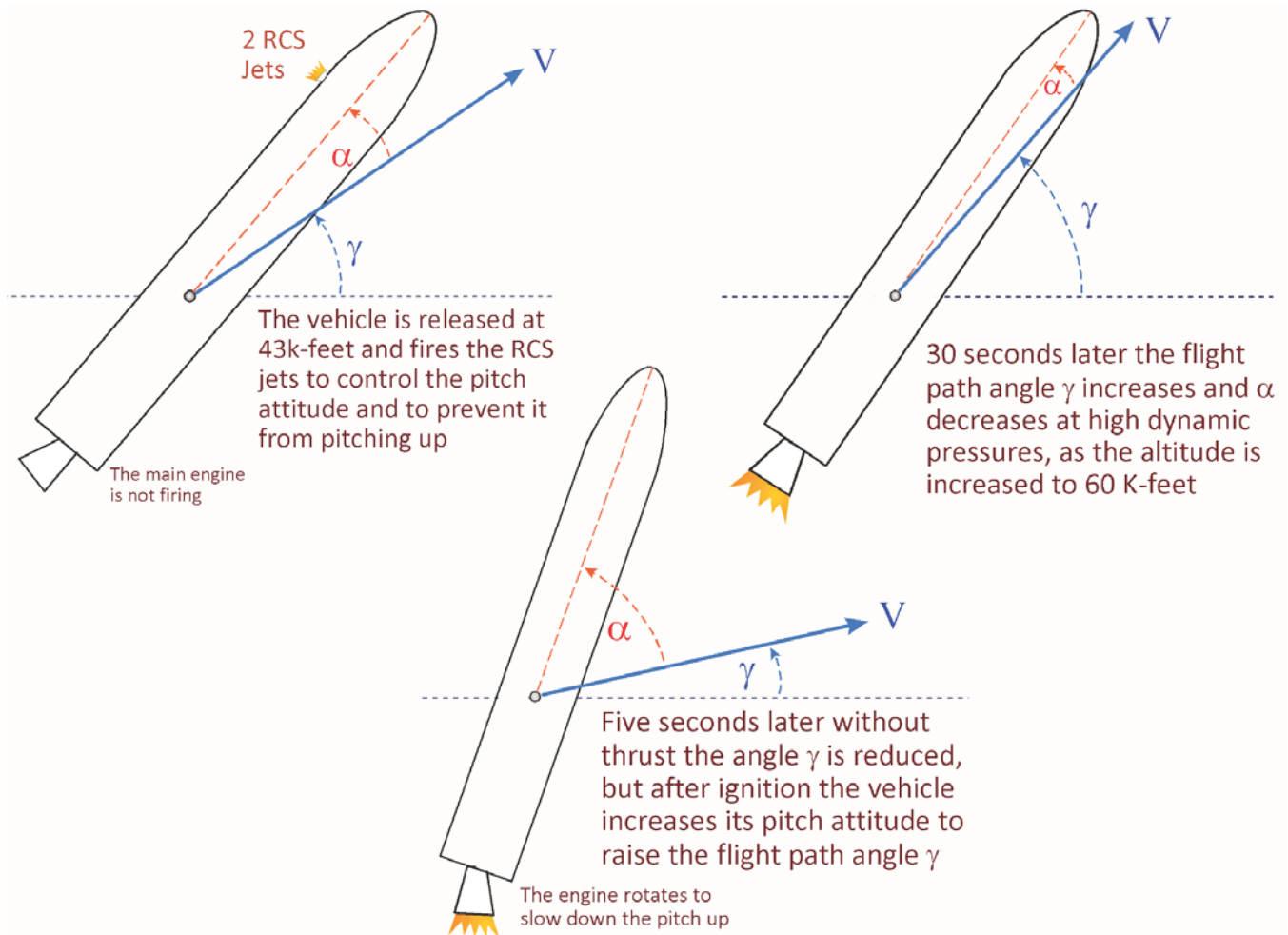
The carrier aircraft is also special because it is capable of climbing up to 43,000 feet and releasing the booster in an upward direction, that is, with a positive flight-path angle γ , as shown in the picture above. This maneuver requires an aircraft with a rocket that ignites at about 30,000 feet and 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.

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.

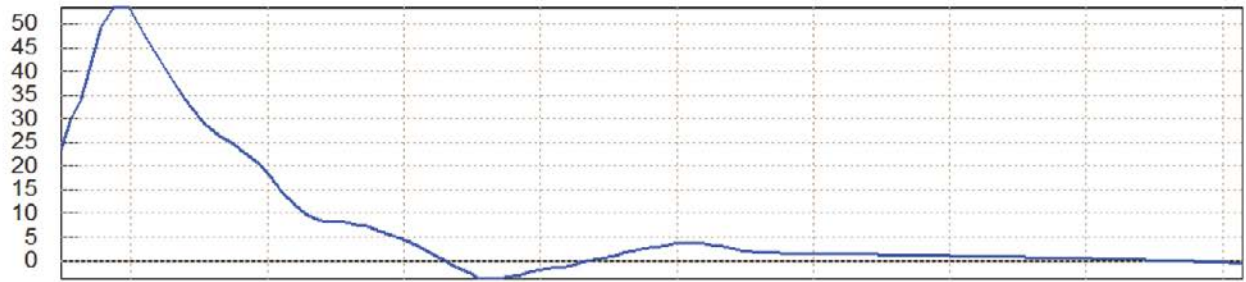
1. Static Analysis

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.

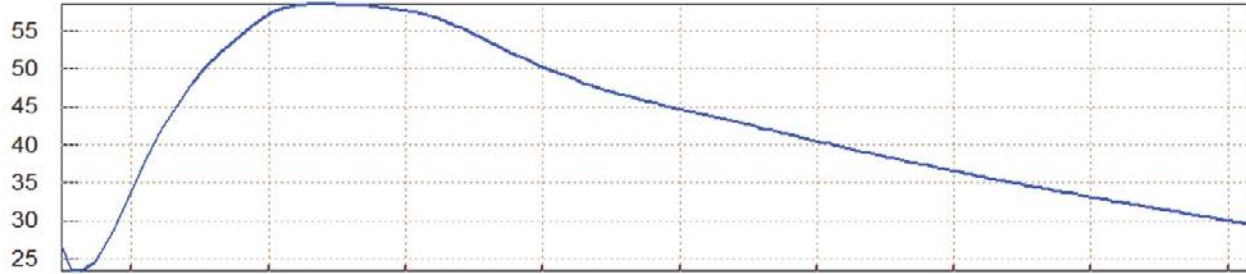


1.1 First Stage Static Analysis

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 $\gamma=45^\circ$. 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° . As soon as the engine ignites the vehicle pitches up to an attitude $\theta=86^\circ$ and increases

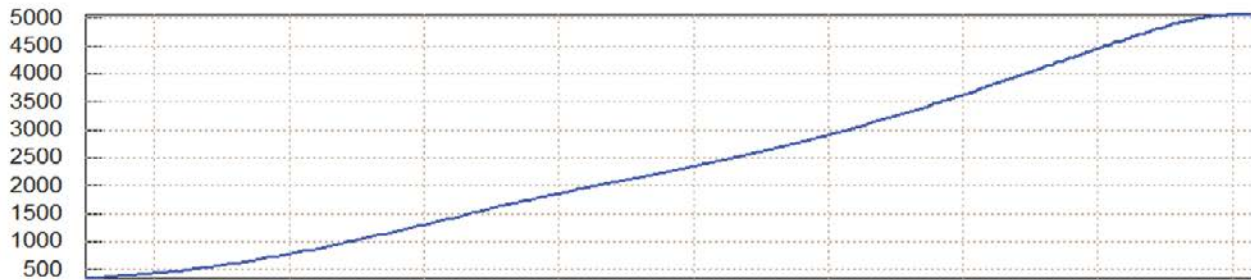


Alpha

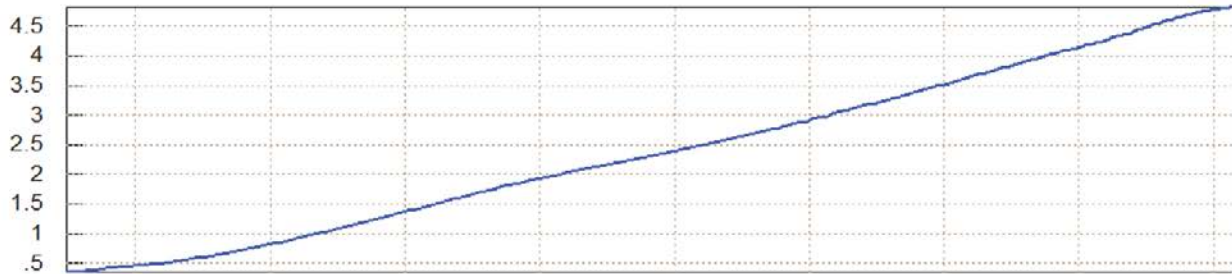


Gamma

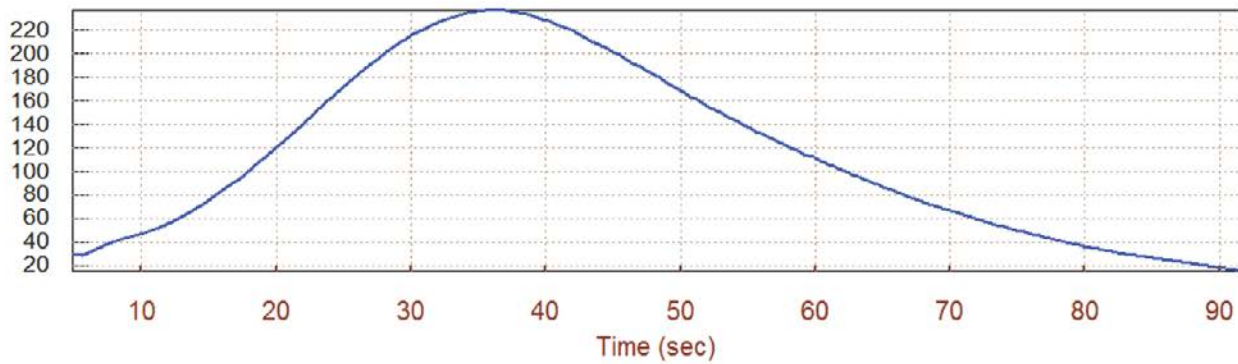
Velocity, Dynamic Pressure, Zulu Launch Vehicle, First Stage Trajectory



Veloc (ft/s)

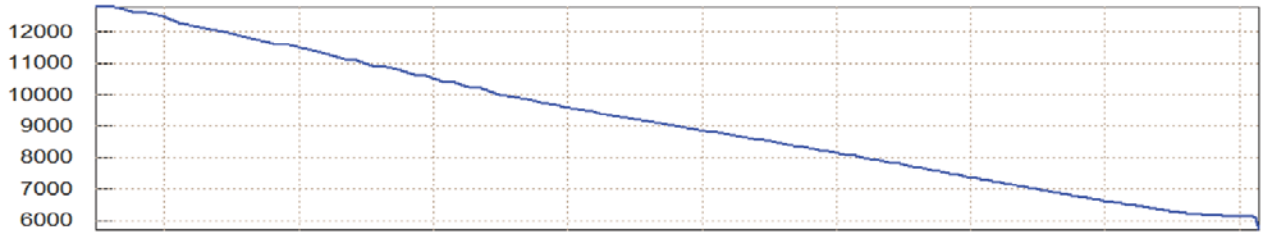


Mach Number



Q-bar (PSF)

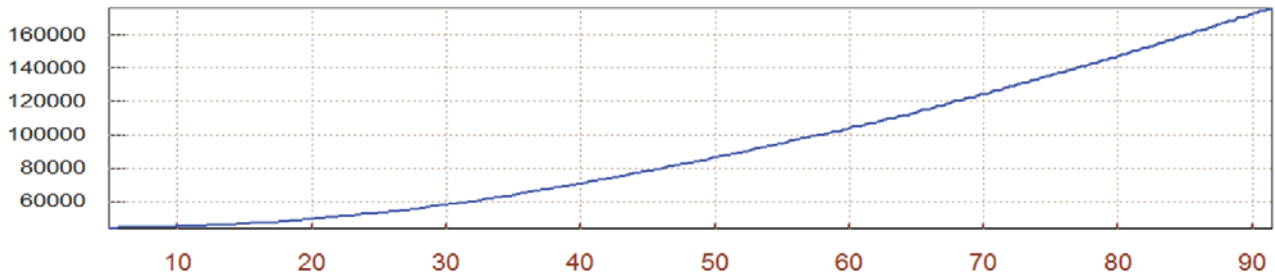
Vehicle Altitude, Mass, Bank Angle, Zulu Launch Vehicle, First Stage T



Mass (slugs)

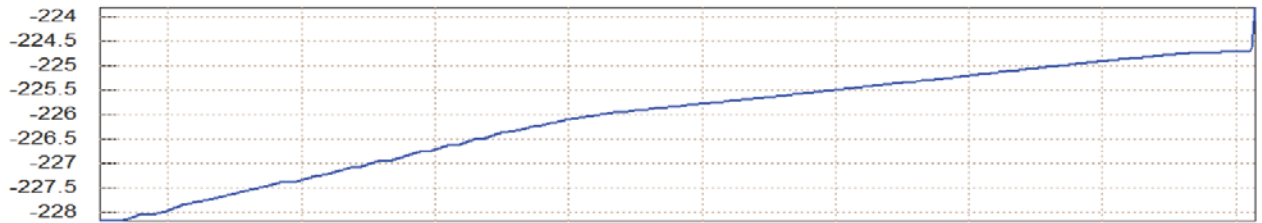


Bank (degr)

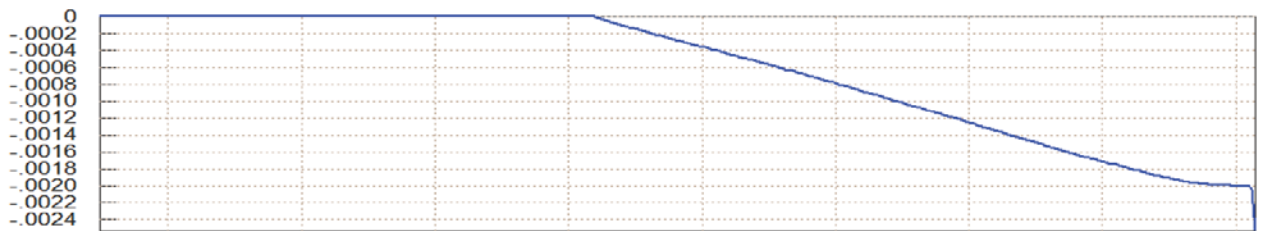


Altitud (ft)

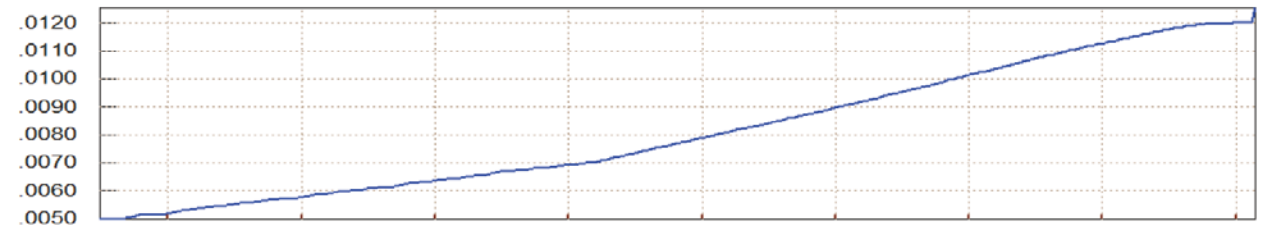
Vehicle CG in (feet), Zulu Launch Vehicle, First Stage Trajectory



Xcg



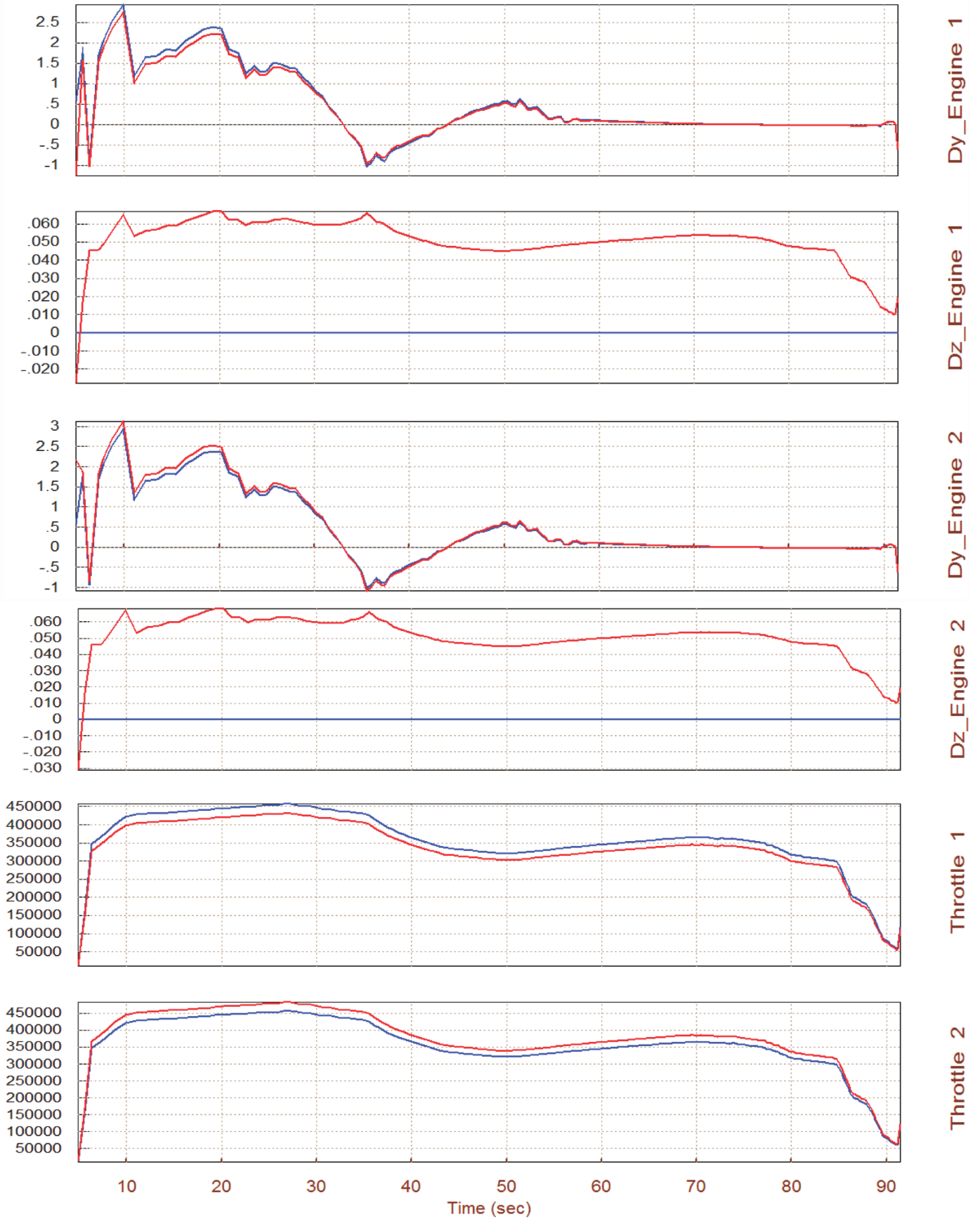
Ycg



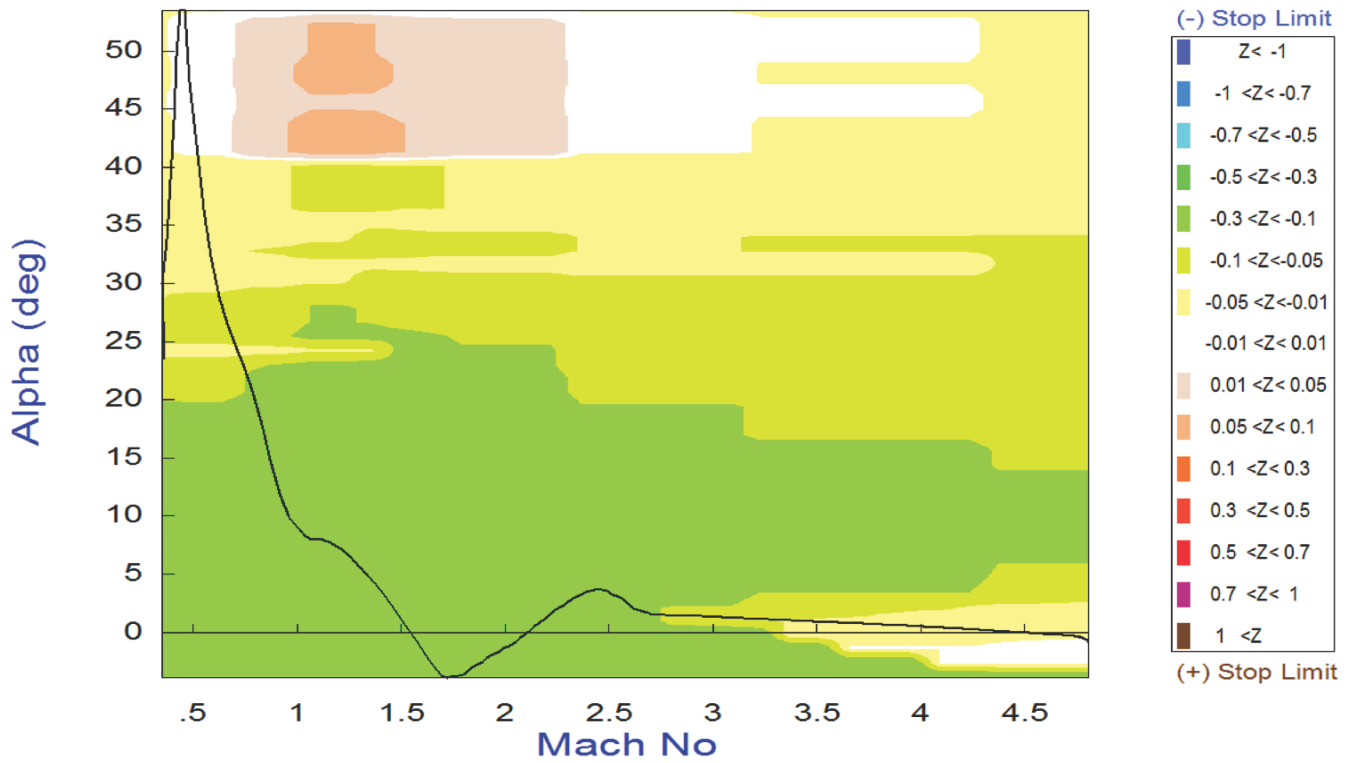
Zcg

Time (sec)

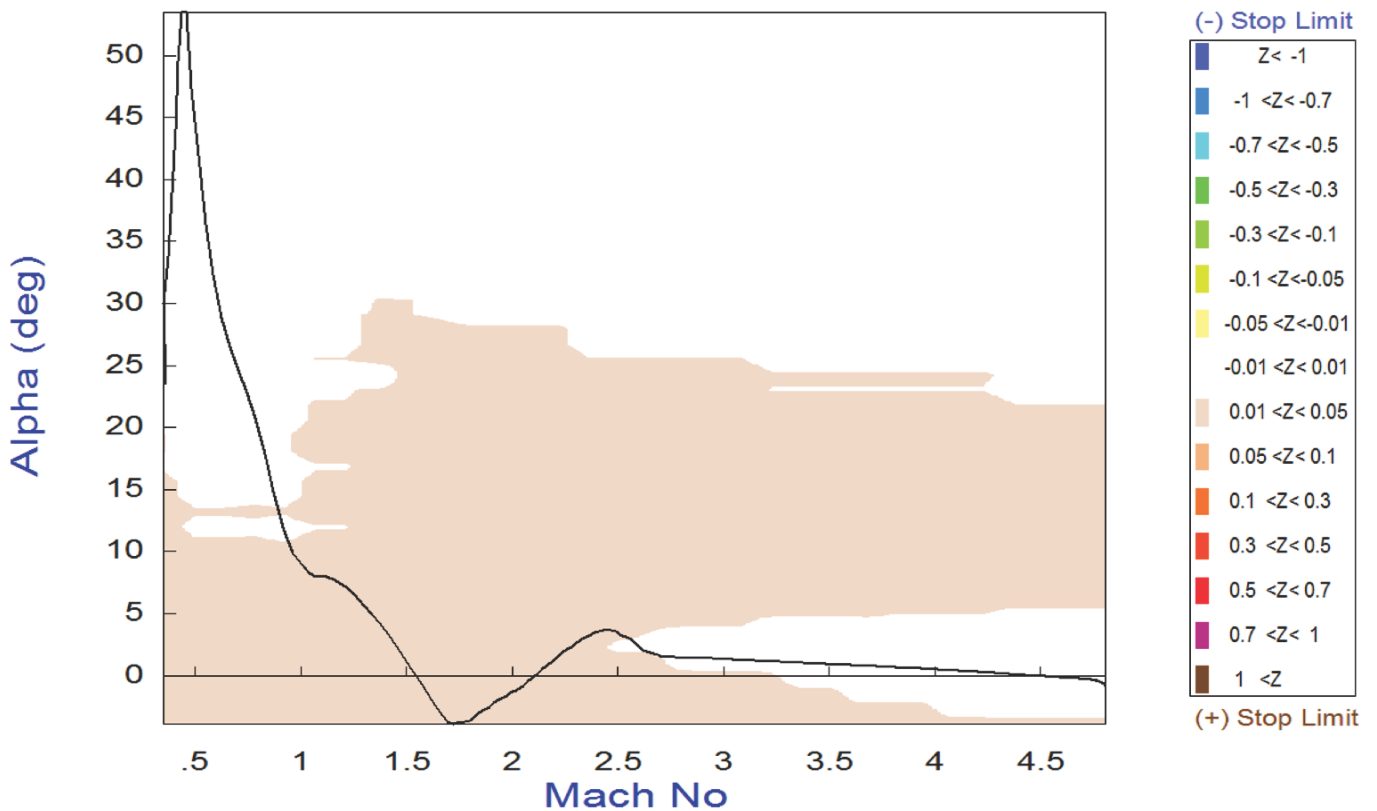
Surface & Engine Deflections/ Thrusts, Zulu Launch Vehicle, First Stage



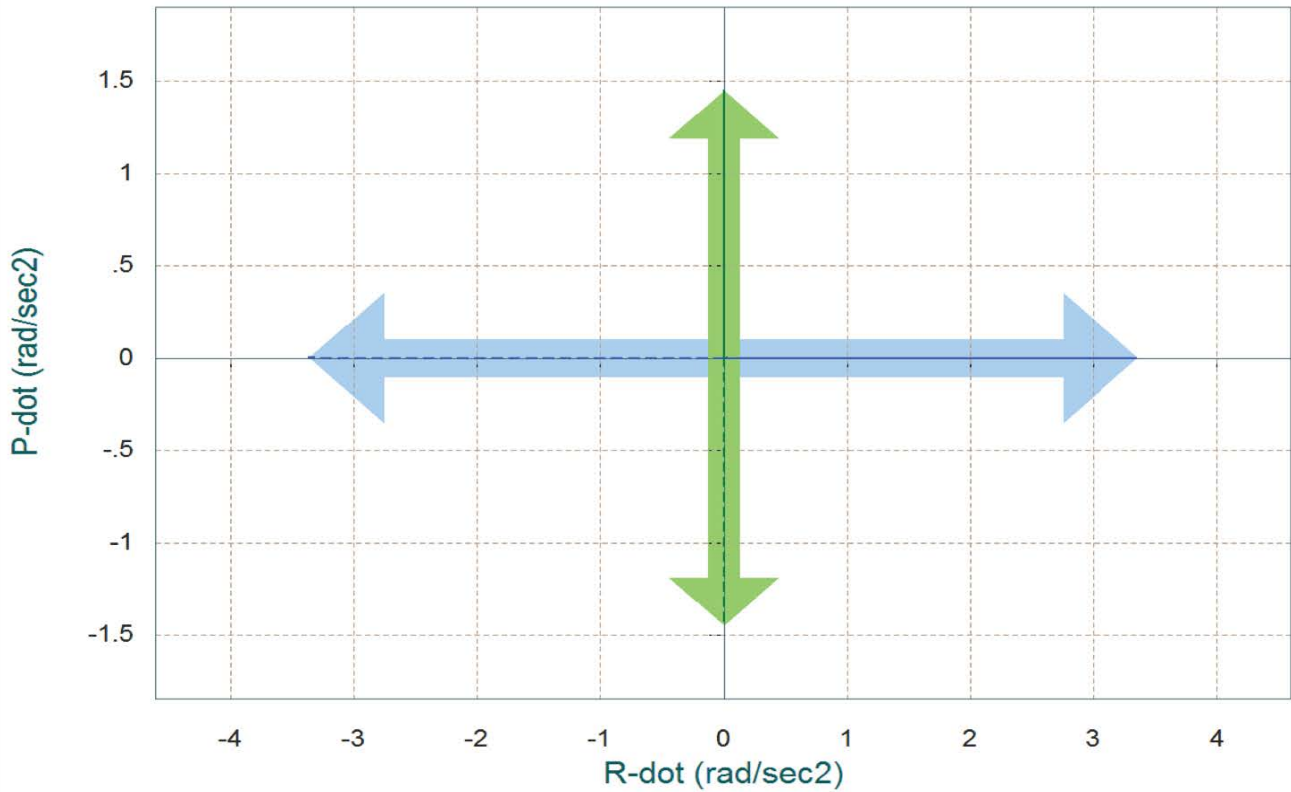
Pitch Control Effort Contour Plot (Mach vs Alpha)



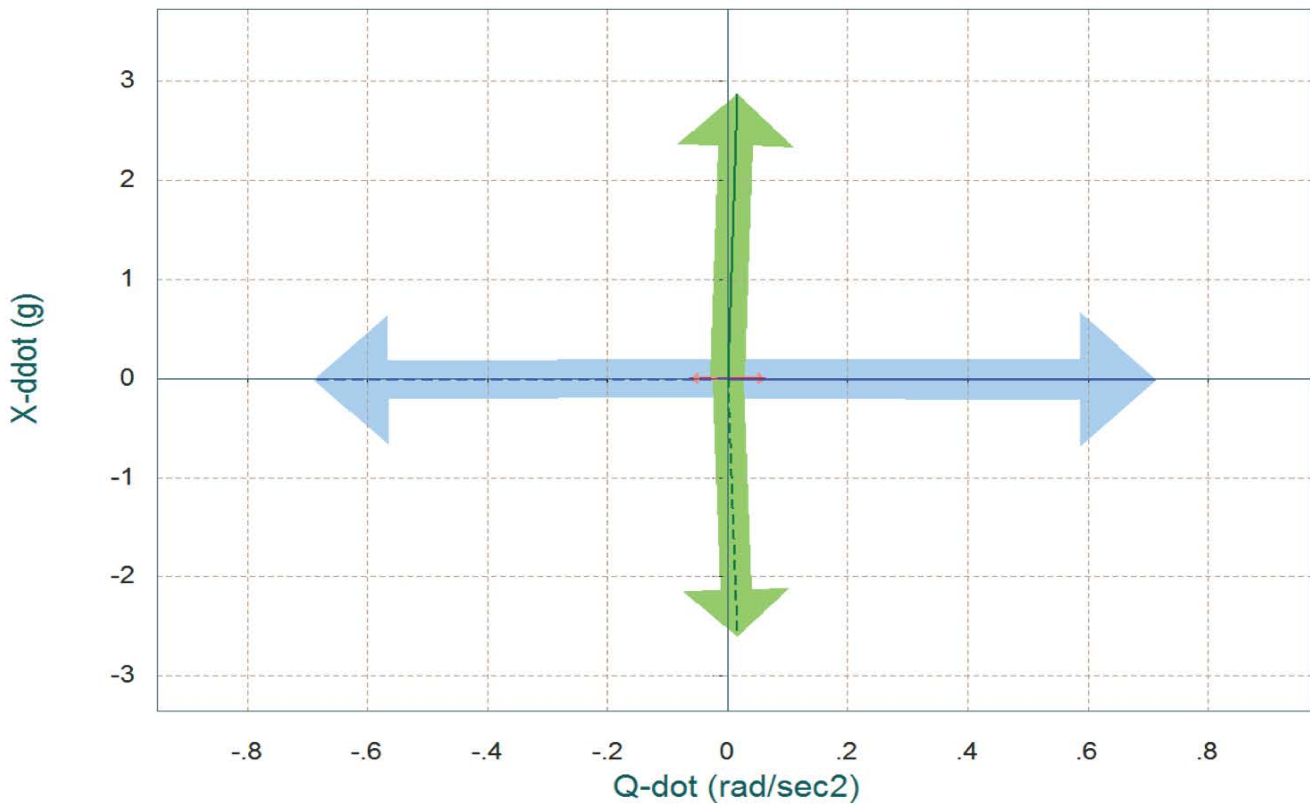
Yaw Control Effort Contour Plot (Mach vs Alpha)



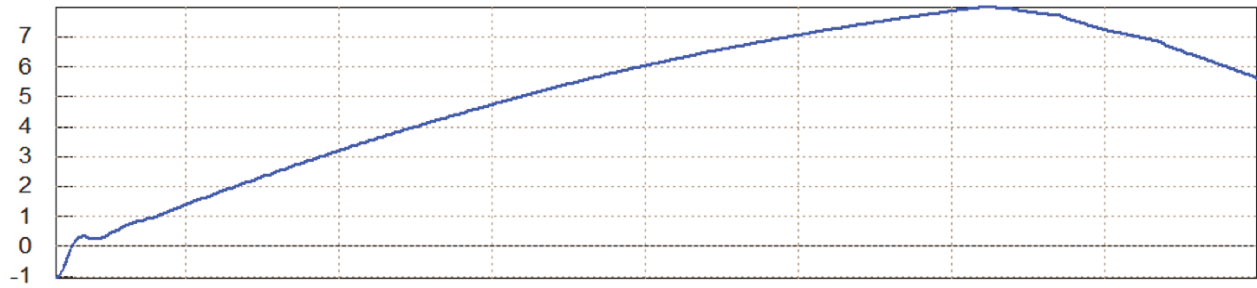
Comparison Between Maximum Control Accelerations and Max Accels due to Beta (red)
Roll & Yaw Accelerations due to Maximum Roll/ Yaw Control and due to Max Beta



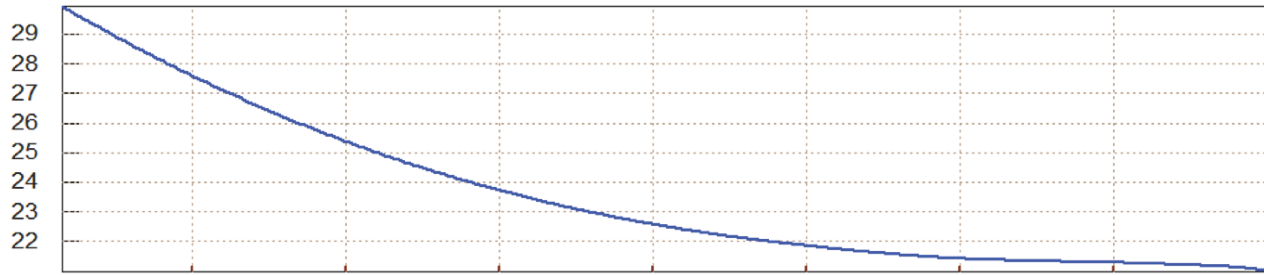
Comparison Between Maximum Control Accelerations and Max Accels due to Alpha (red)
Pitch Accelerat and Axial X-Accelerations due to Max Control and due to Max Alpha



Angles of Attack/Sideslip/Flight Path (deg), Zulu Launch Vehicle, Seco

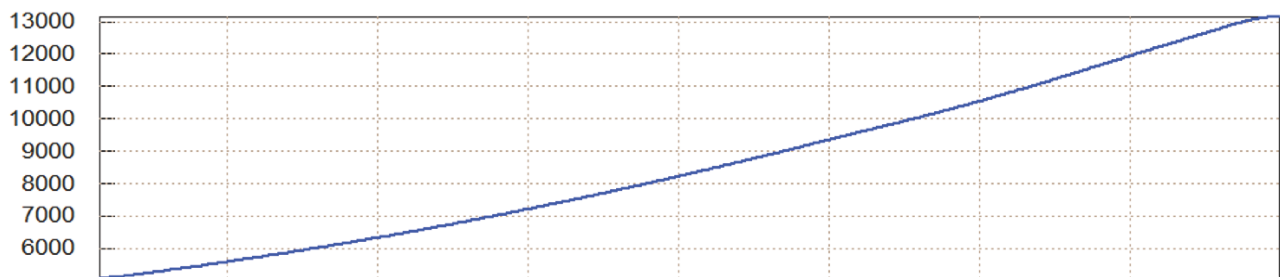


Alpha

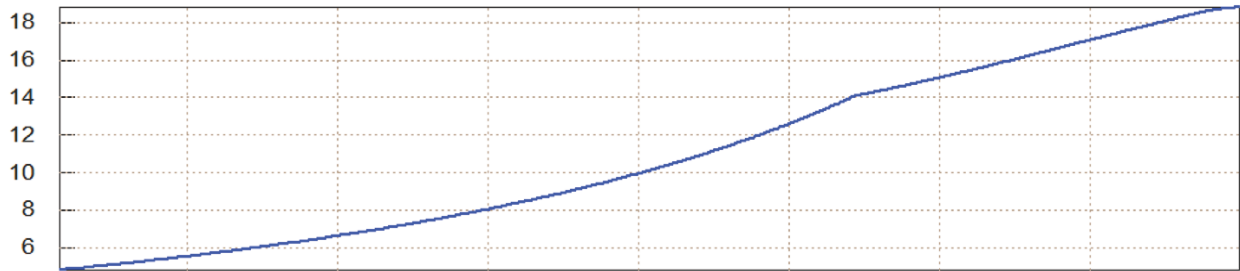


Gamma

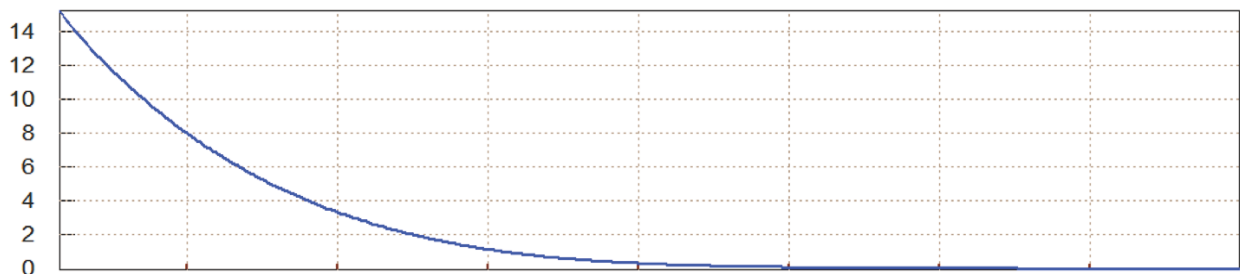
Velocity, Dynamic Pressure, Zulu Launch Vehicle, Second Stage Trajectory



Veloc (ft/s)



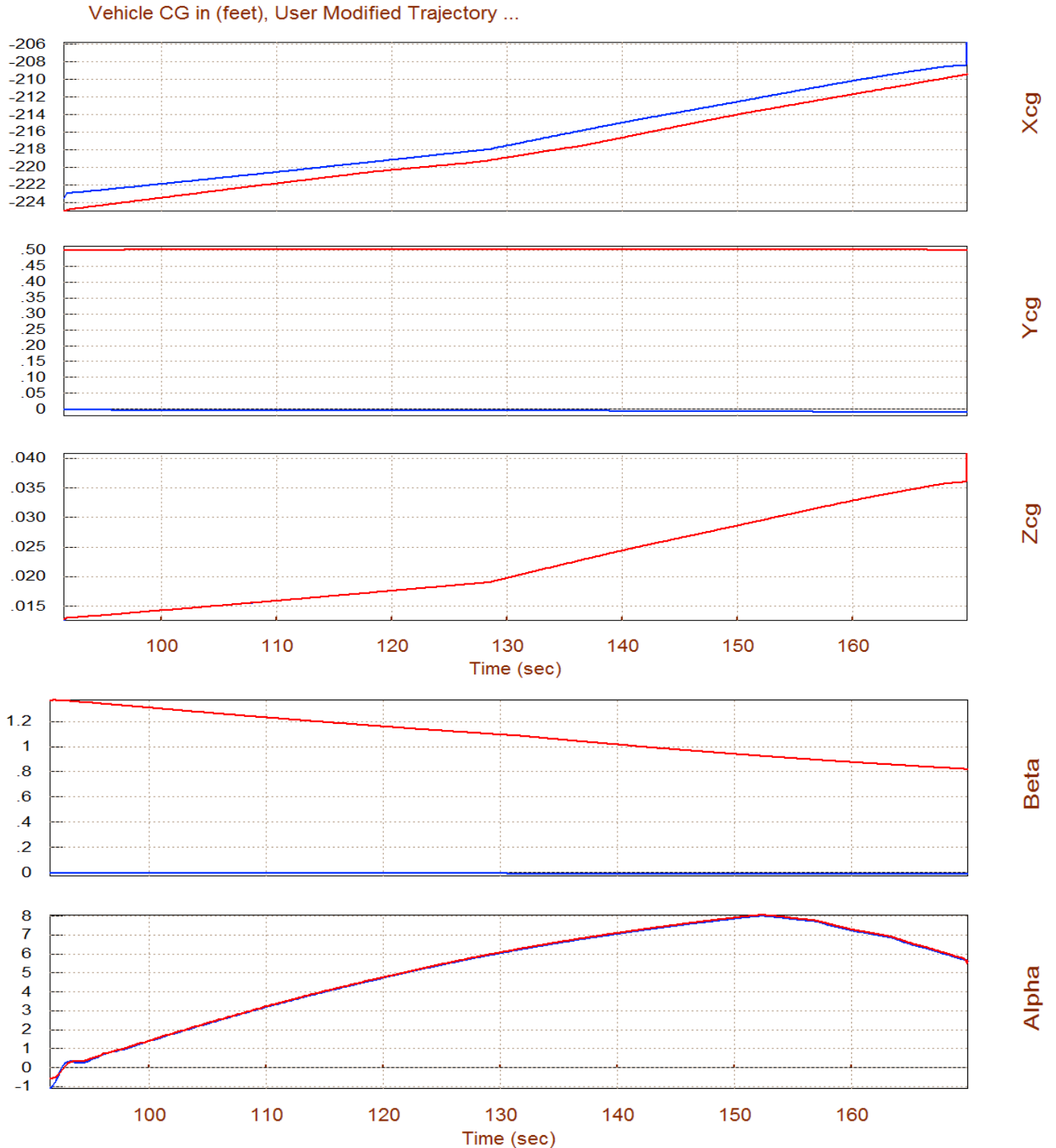
Mach Number



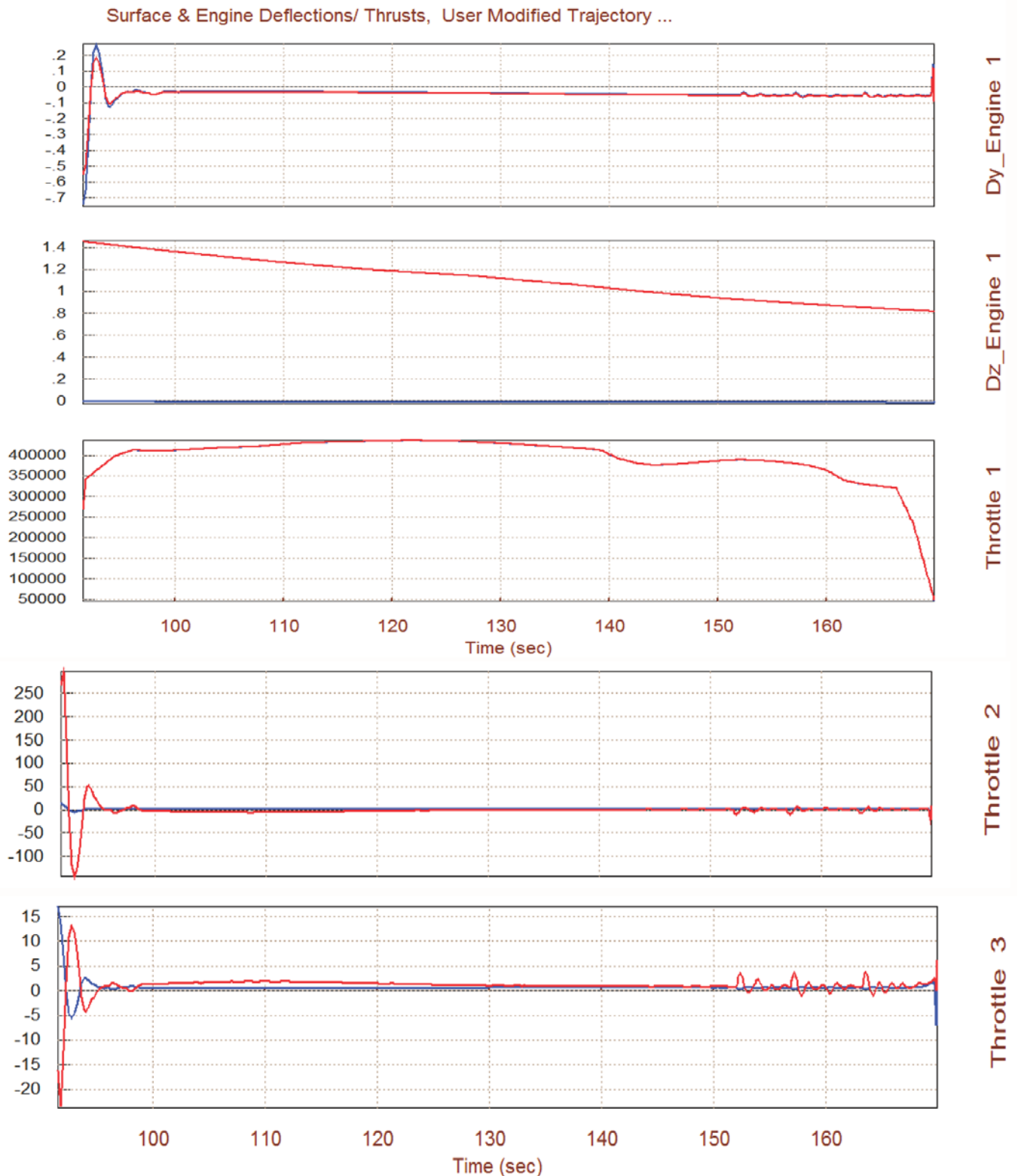
Q-bar (PSF)

Time (sec)

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 plots are obtained from Option-6 and they show the static performance parameters along the second stage. The calculation of the performance parameters require a mixing logic matrix that will combine the five effectors (2 TVC, 1 engine throttle, and 2 RCS thrusters) to achieve the four demanded accelerations (roll, pitch, yaw, and axial). A precalculated mixing logic matrix K_{mix3} is already saved in file "Mixing_Matrix.Qdr" and it must be selected prior to the analysis.

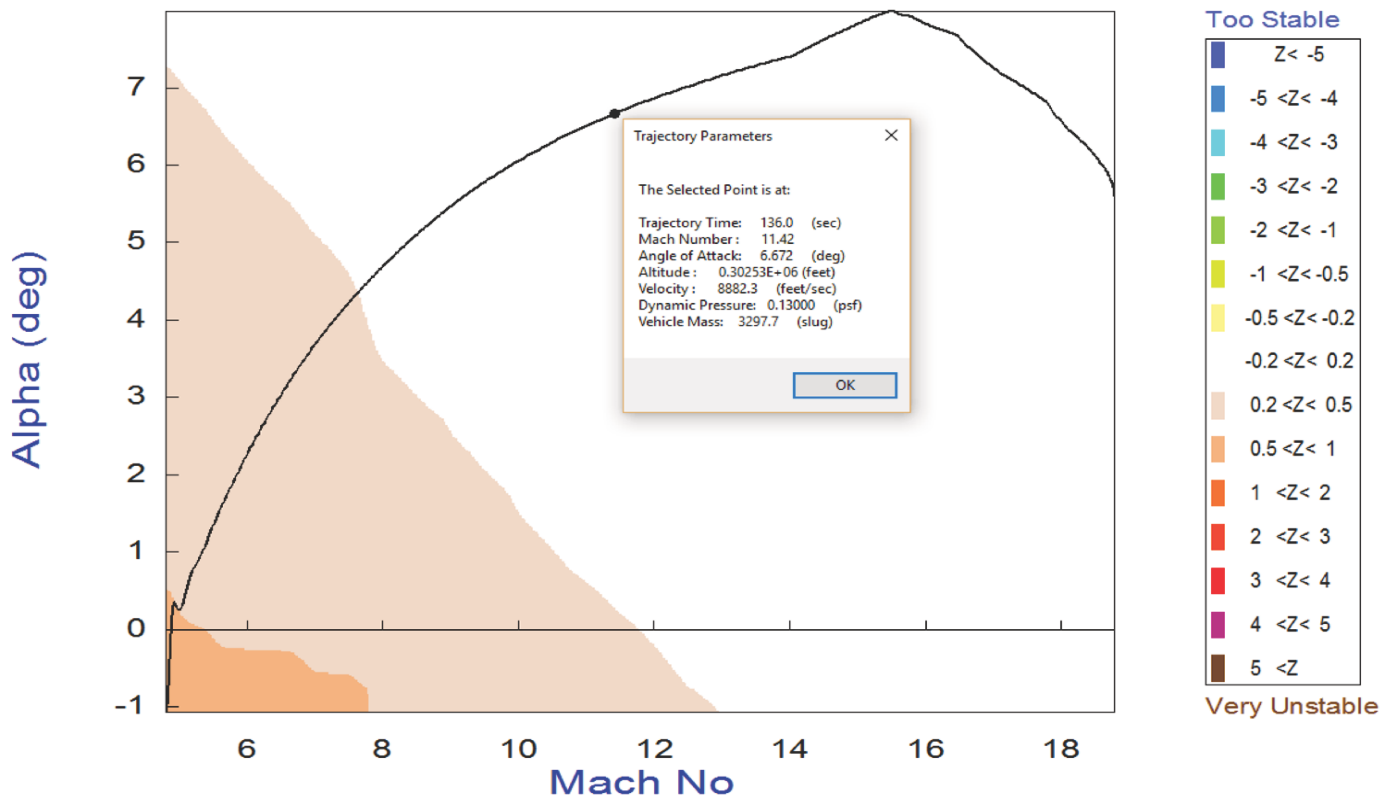
The T2-inverse parameters show that the vehicle is statically unstable in both pitch and yaw, but the time-to-double amplitude is sufficiently long, about 2 seconds. The $C_n\beta$ -dynamic parameter is negative, which is also a sign of lateral instability. The LCDP defines dynamic roll controllability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

Vector Diagrams

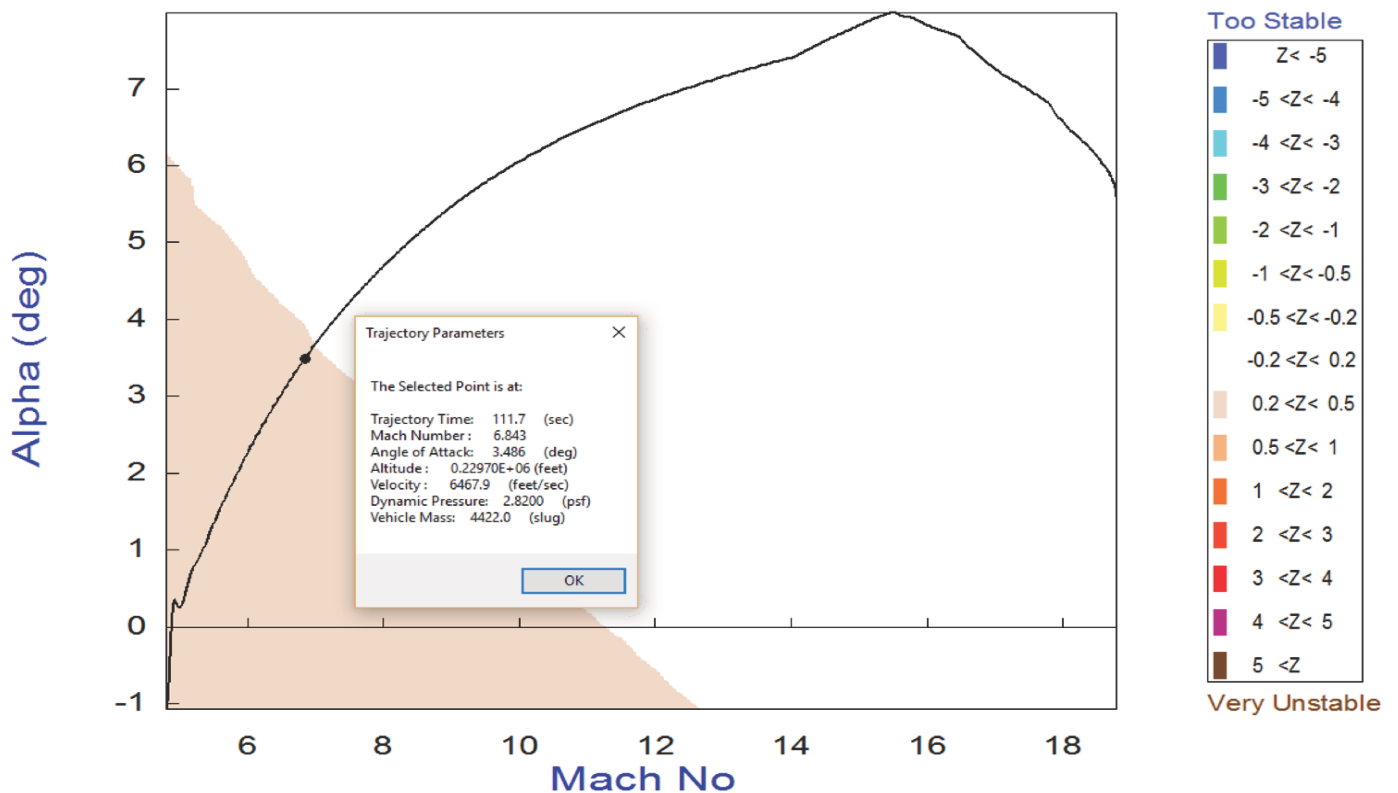
The two vector diagrams show the maximum roll and yaw torques generated by the effectors in the positive and in the negative directions. The roll and yaw demands are combined by the precalculated mixing logic matrix that calculates the TVC deflection and throttling commands, and it must be selected from 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 \pm maximum normalized roll and yaw moments at \pm 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 $\delta z=+1.4^\circ$ 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 $+5^\circ$ 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° 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.

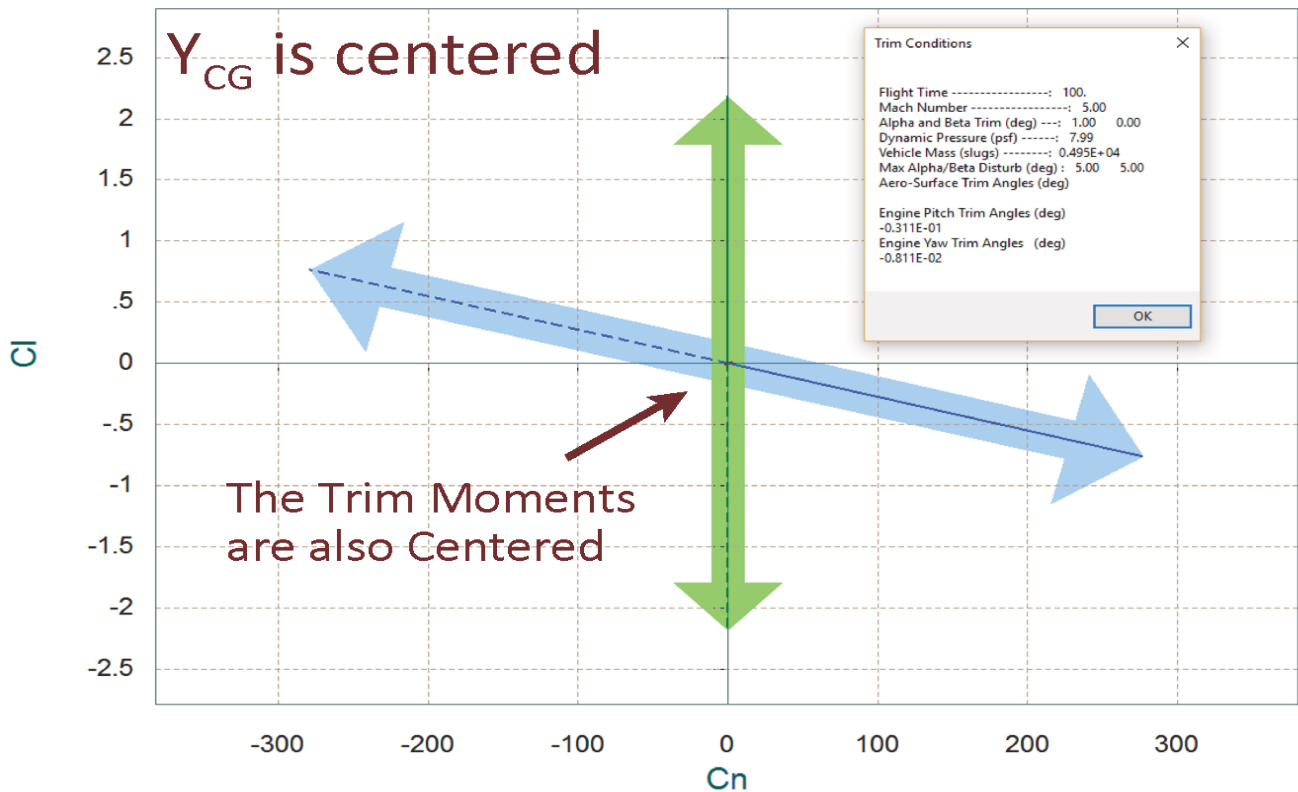
Pitch Stability Contour Plot (Mach vs Alpha)



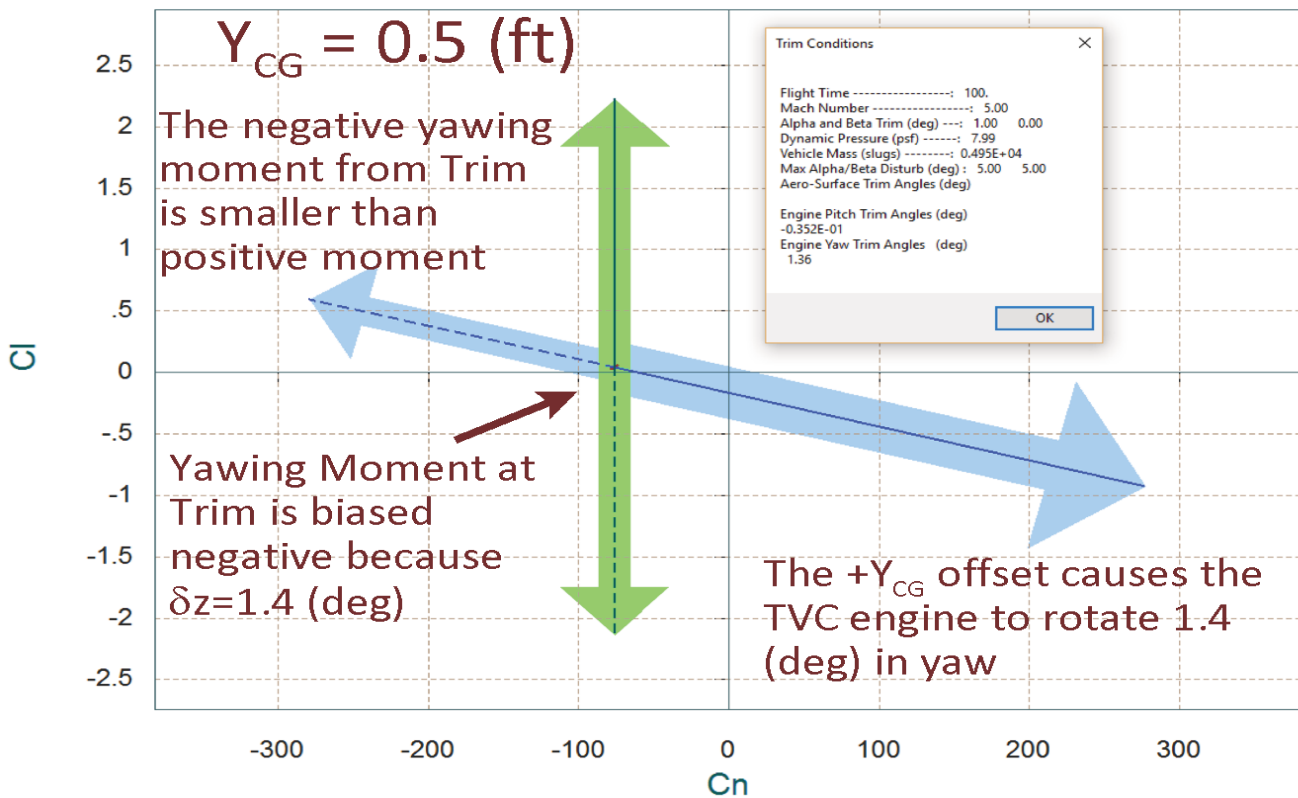
Lateral Stability Contour Plot (Mach vs Alpha)



Comparison Between Maximum Control Moments Against Maximum Disturb Moments (red)
Roll & Yaw Control Moments (non-dimension) vers Disturb Moment due to Max Beta/Alpha



Comparison Between Maximum Control Moments Against Maximum Disturb Moments (red)
Roll & Yaw Control Moments (non-dimension) vers Disturb Moment due to Max Beta/Alpha



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.

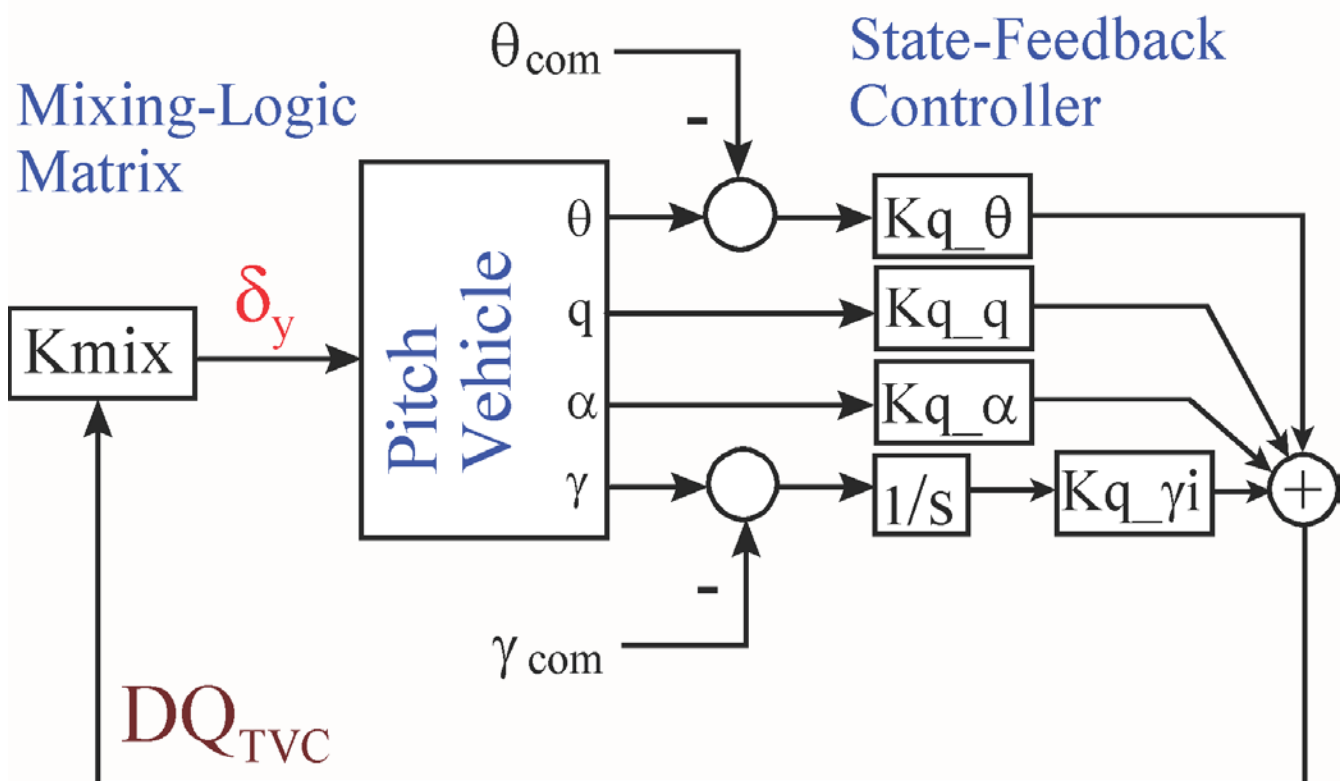


Figure 2.1 Pitch Control System uses State-Feedback to Generate the Pitch Control Demand DQ_{TVC}

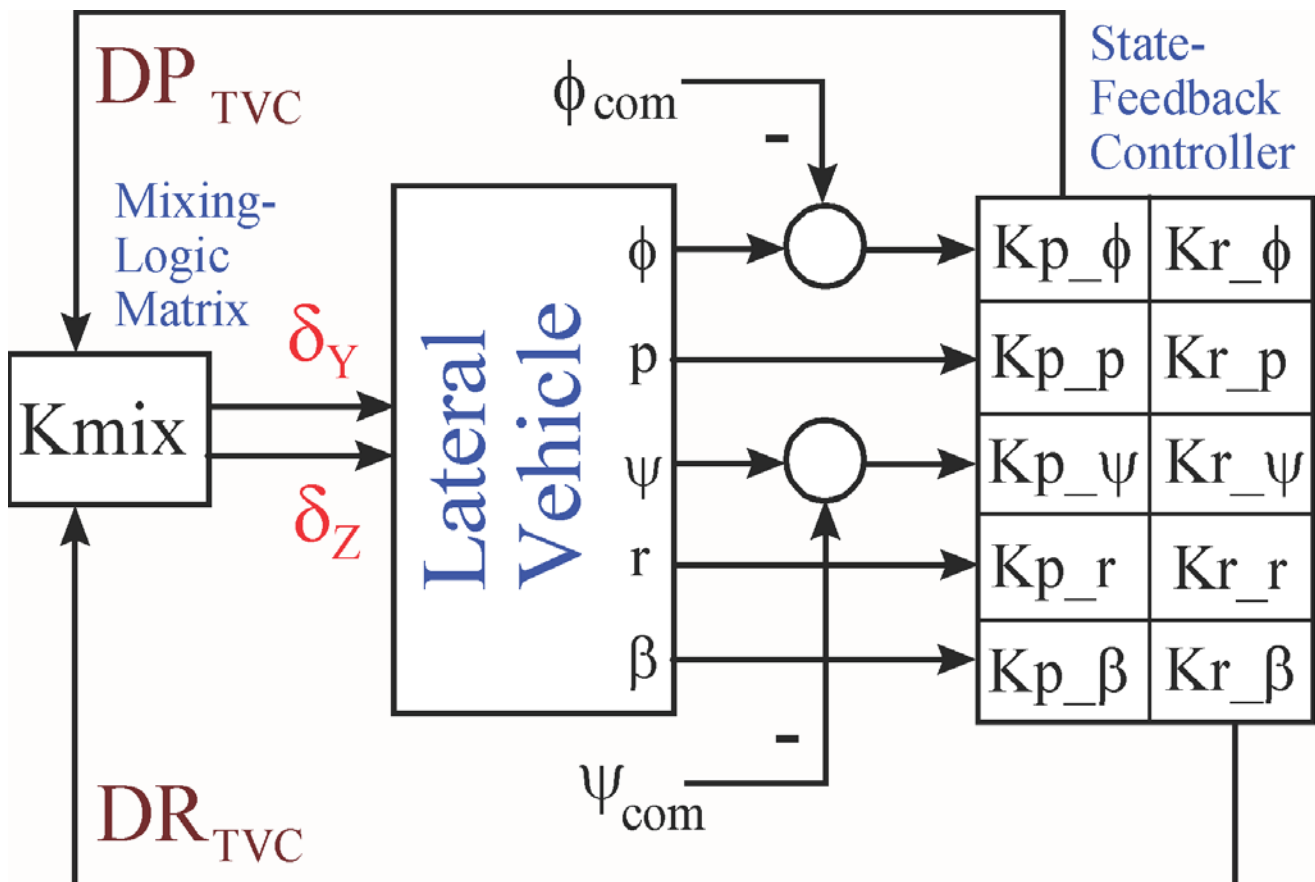


Figure 2.2 Lateral Control System uses State-Feedback to Generate the Roll and Yaw Control Demands

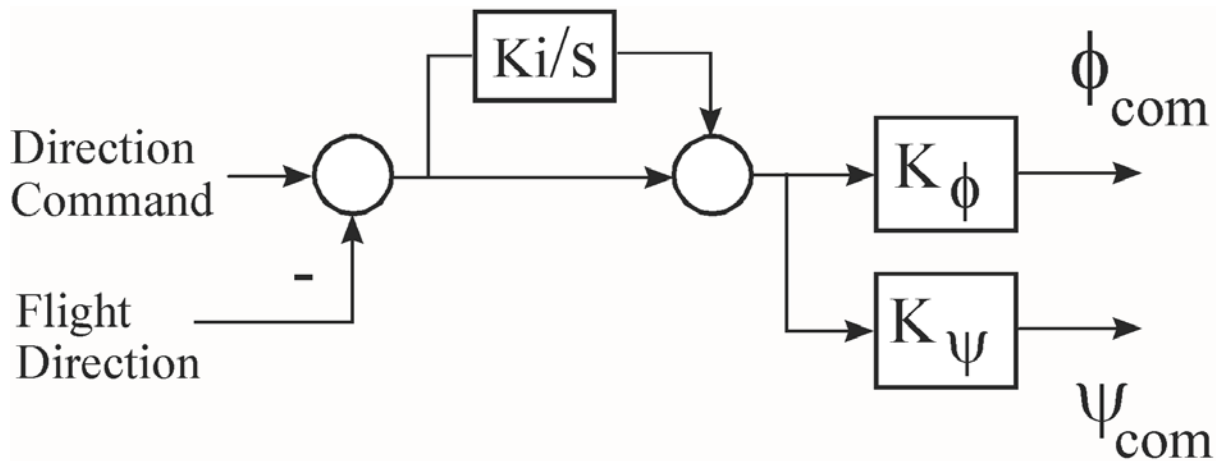
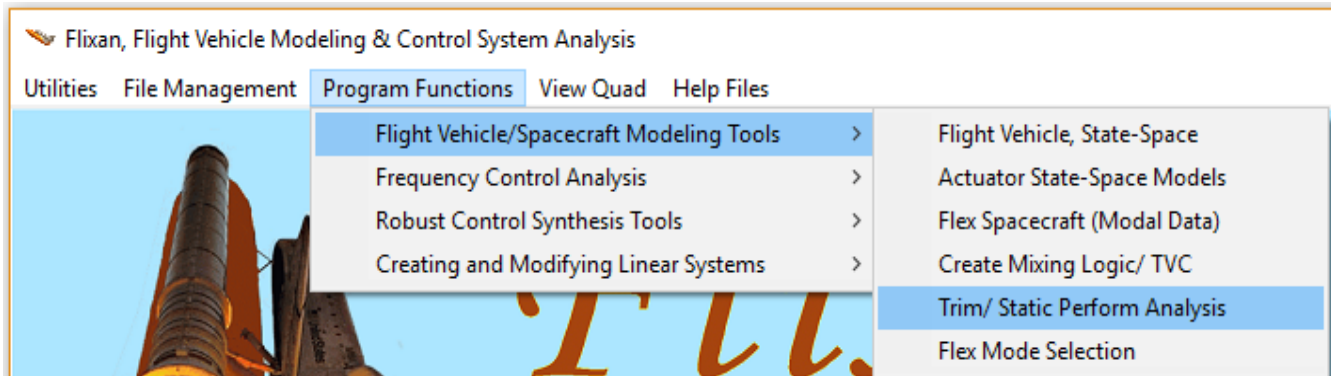
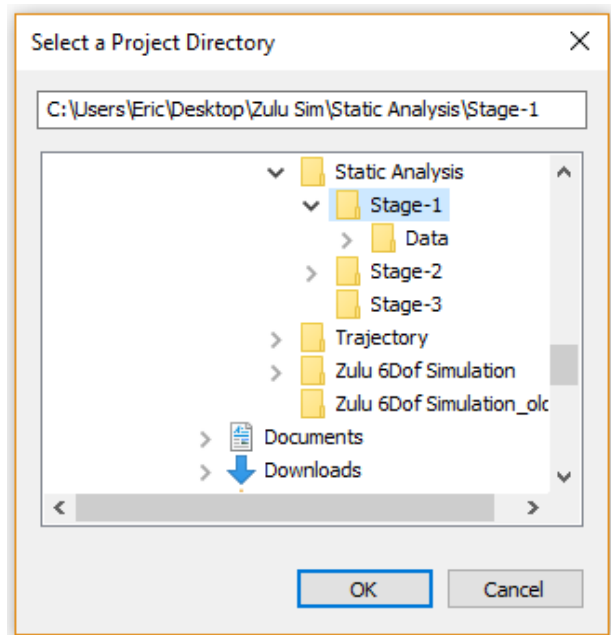
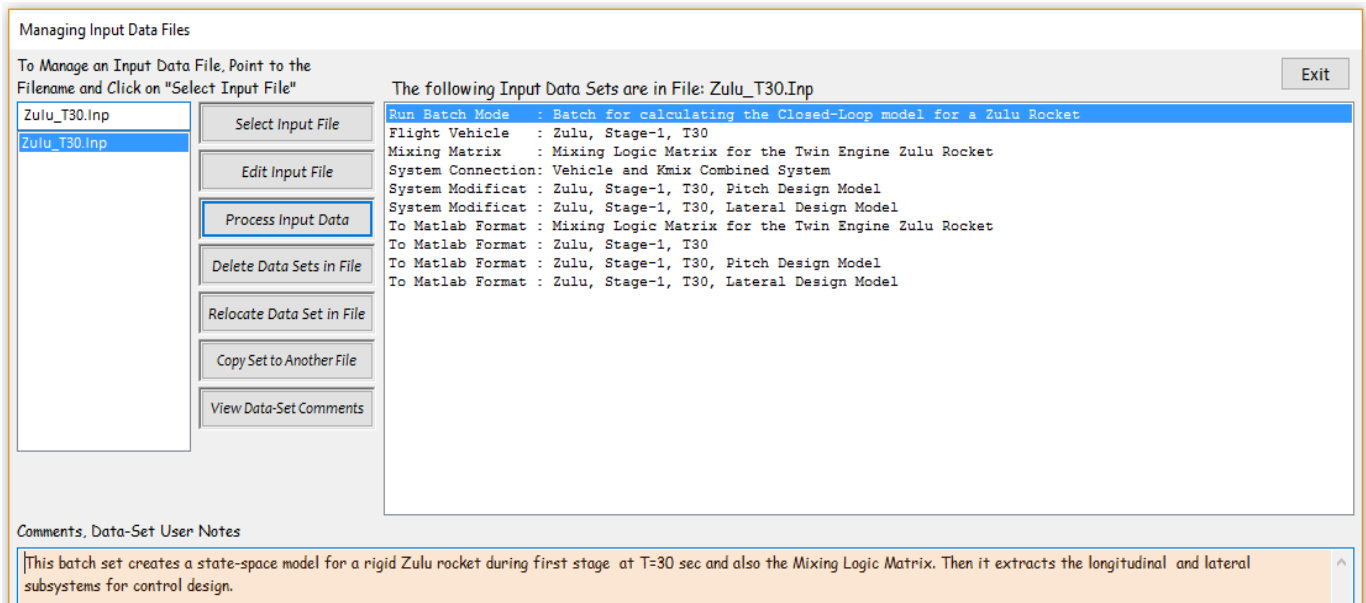


Figure 2.3 Lateral Guidance Generates the Roll and Yaw Attitude Commands

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”.





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 (δy_i and δz_i). 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.

Flight Control Demands



First Stage Mixing-Logic Matrix

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

```

Pitch Design Model

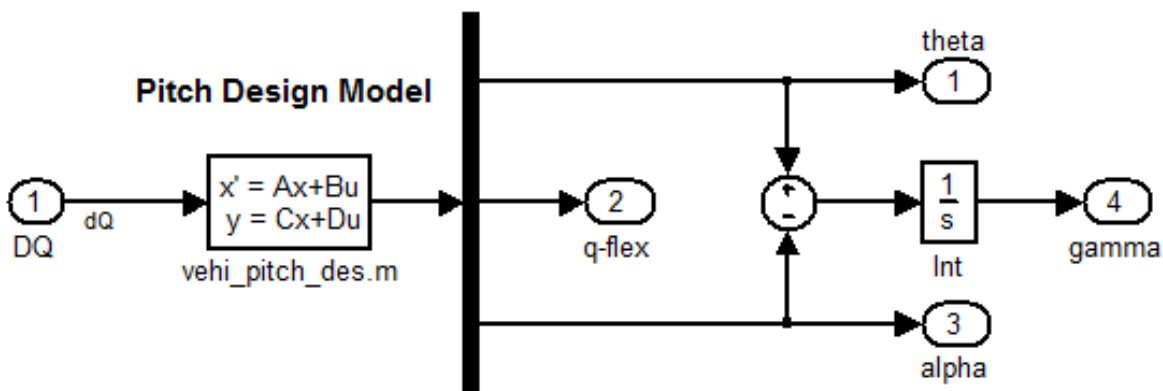


Figure 2.4 The Pitch State-Space System is Augmented to Include Gamma Integral

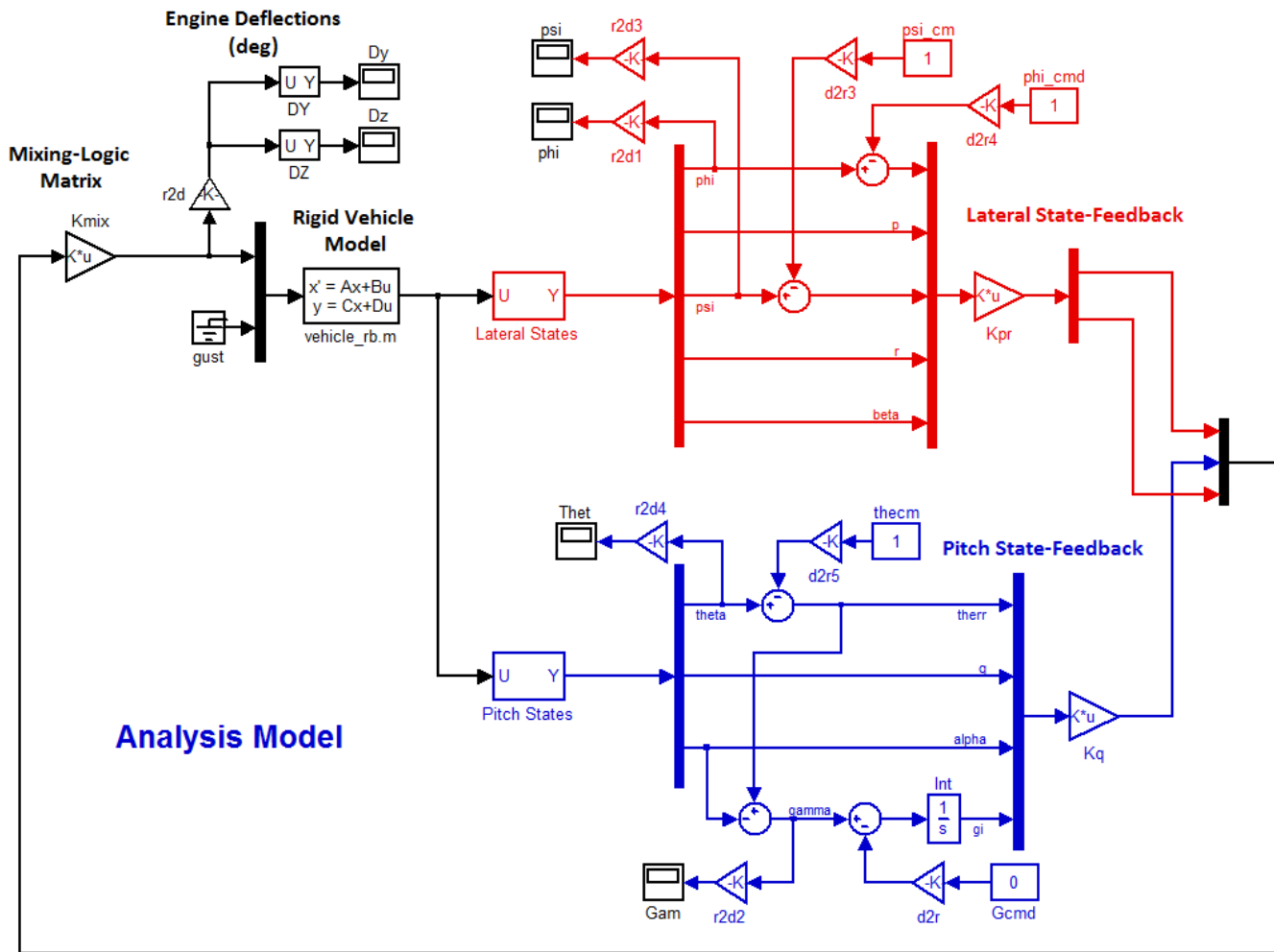


Figure 2.5 Simulink Model “Sim.mdl” used for analyzing the System’s Response to Commands during First Stage

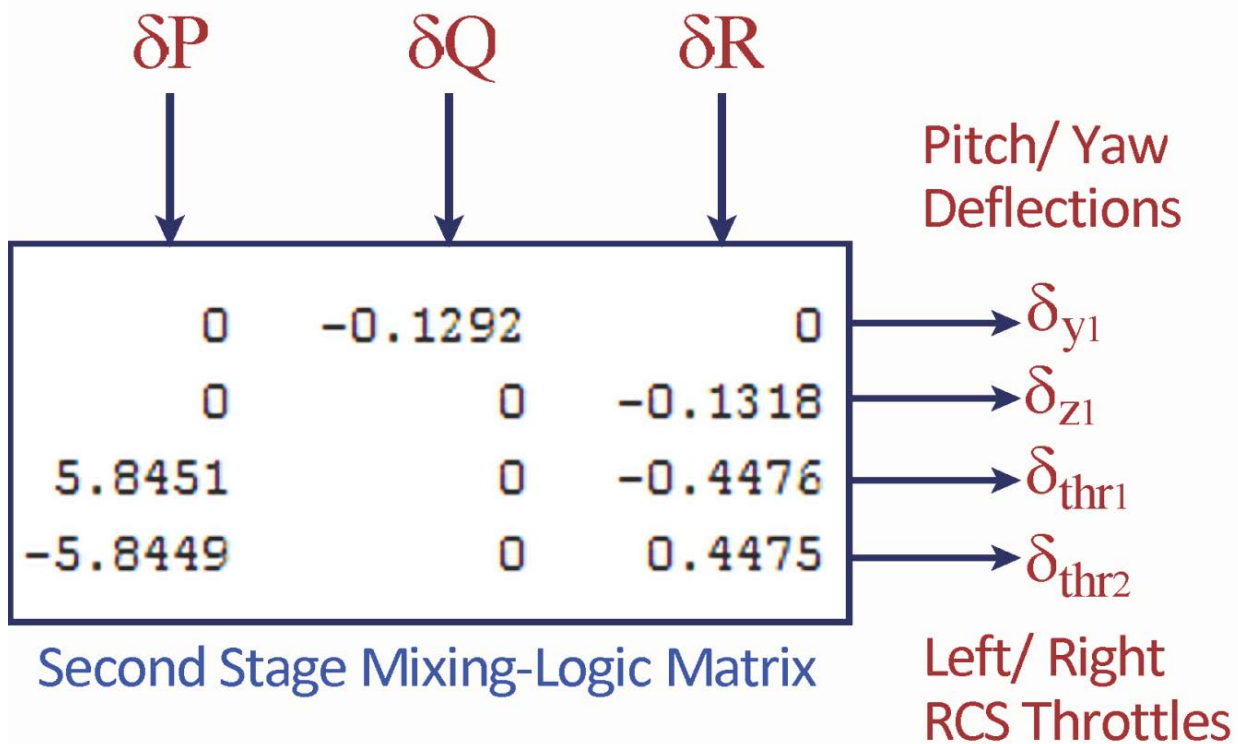
Notice that in the first five seconds after ignition (before $t=10$ sec), and while the dynamic pressure is still low, our design requirement is not to track γ_{com} but to control only the pitch attitude command θ_{com} . The pitch design model in this flight condition, therefore, is simplified to include 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 $\pm Z$ 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 (δy_1 and δz_1) 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.

Flight Control Demands



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.

2.4.2 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 K_{mix} , and the pitch and lateral state-feedback gains K_q and K_{pr} 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.

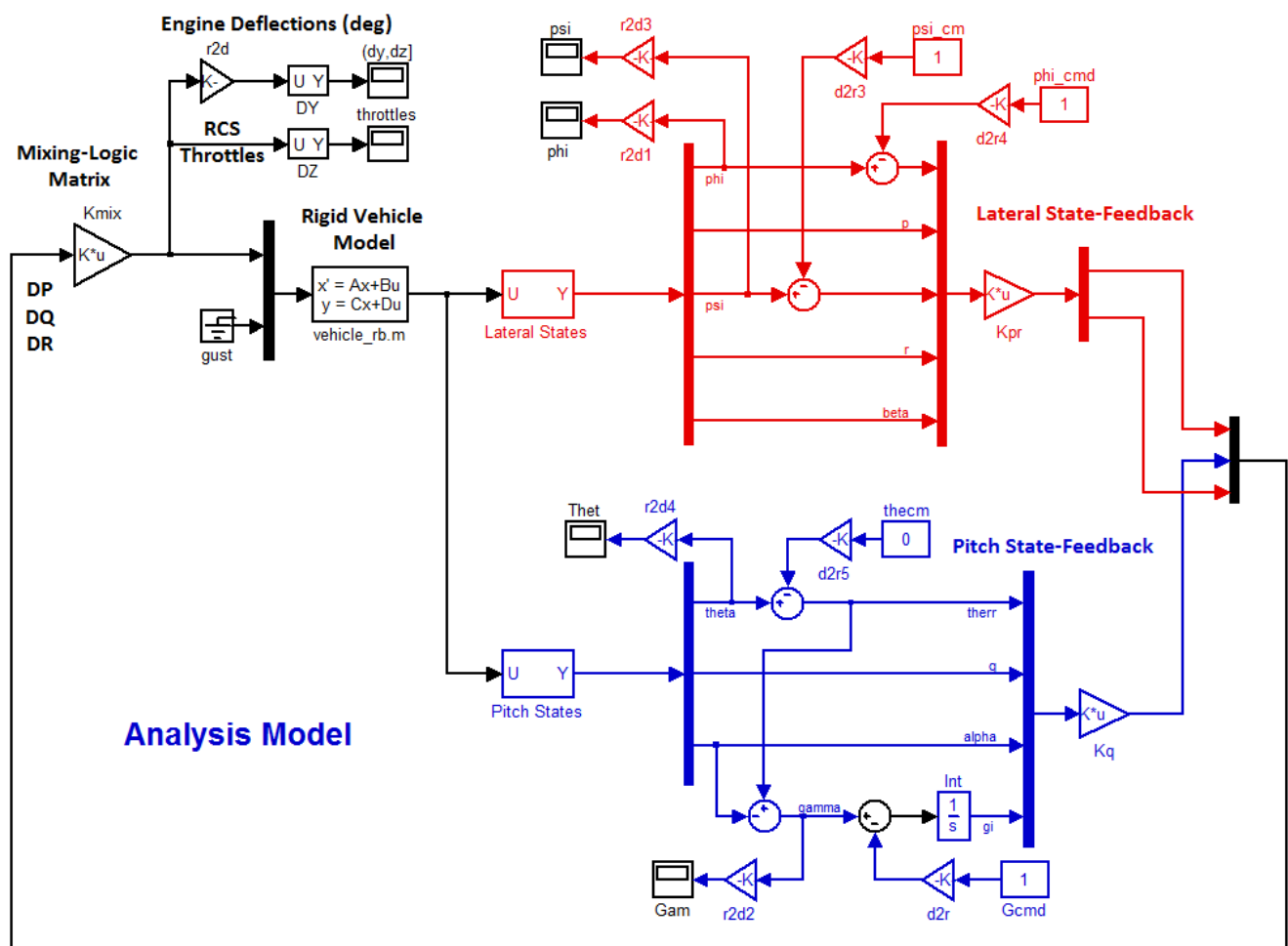


Figure 2.6 Simulink Model “*Sim.Mdl*” for analyzing the System’s Response to Commands during Second Stage

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;
Tst2=170.0; % First Stage Separation Time
% 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;
    0,0,1,0,0,0,0,0,0,0;
    0,1,0,0,0,0,0,0,0,0;
    1,0,0,0,0,0,0,0,0,0];
```

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.

Zulu Rocket Non-Linear 6DOF Simulation

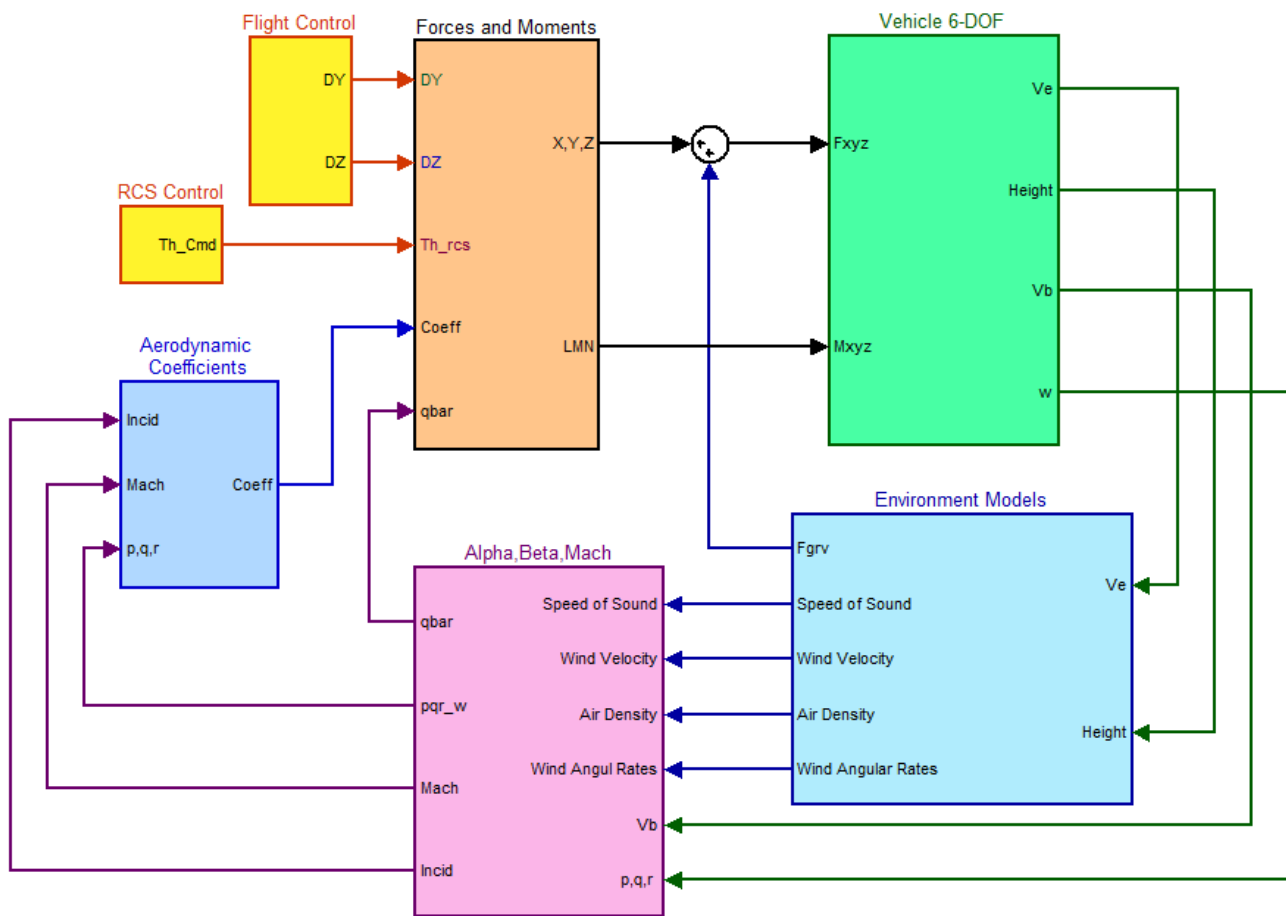


Figure 3.1 Top Level Block Diagram of the Simulink Model "Zulu_6dof_Sim.Mdl" of the Zulu Air-Launched Vehicle

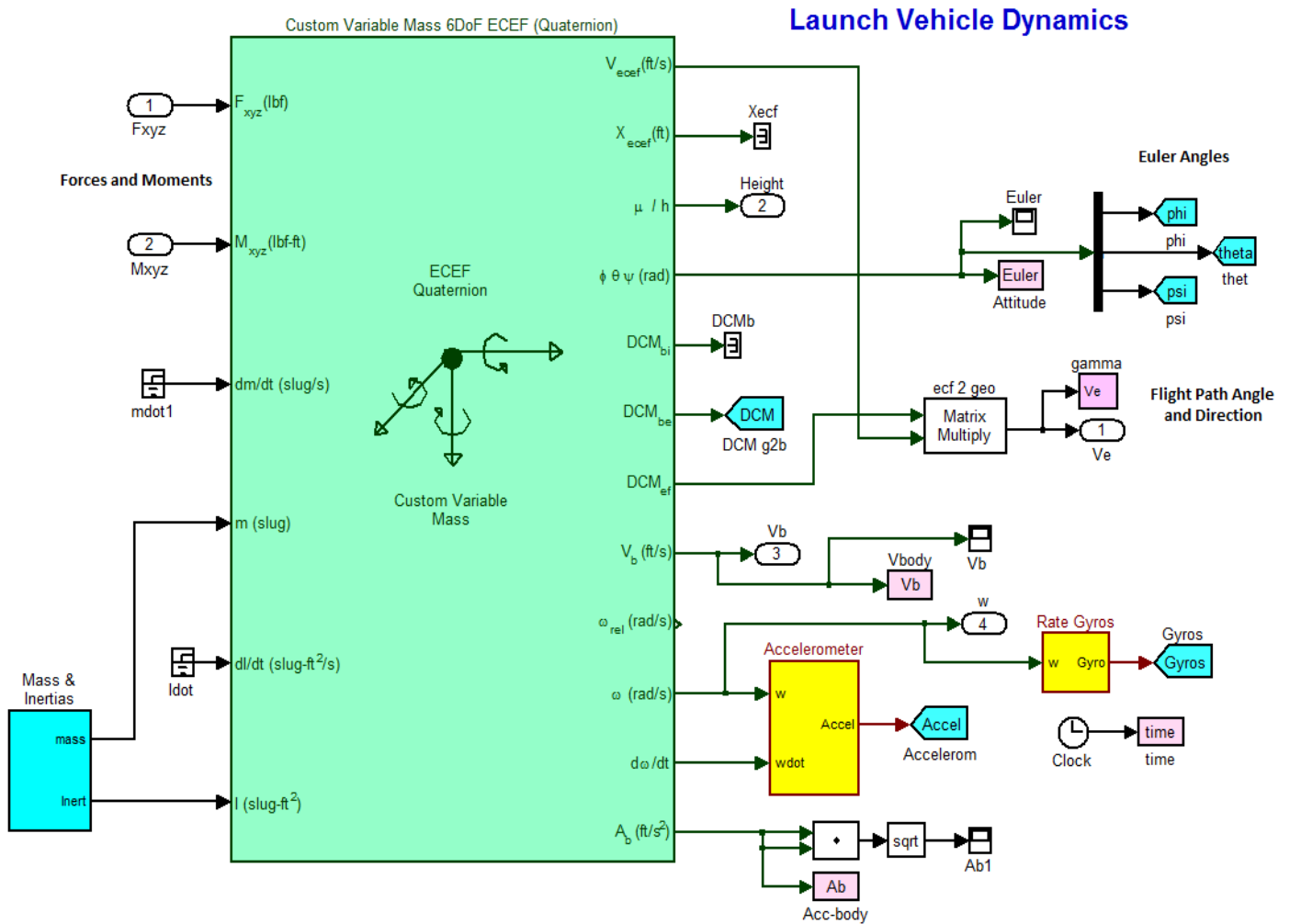


Figure 3.2 Vehicle Dynamics Block, Mass Properties Calculations, and Sensor Blocks

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 [F_x F_y F_z] 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.

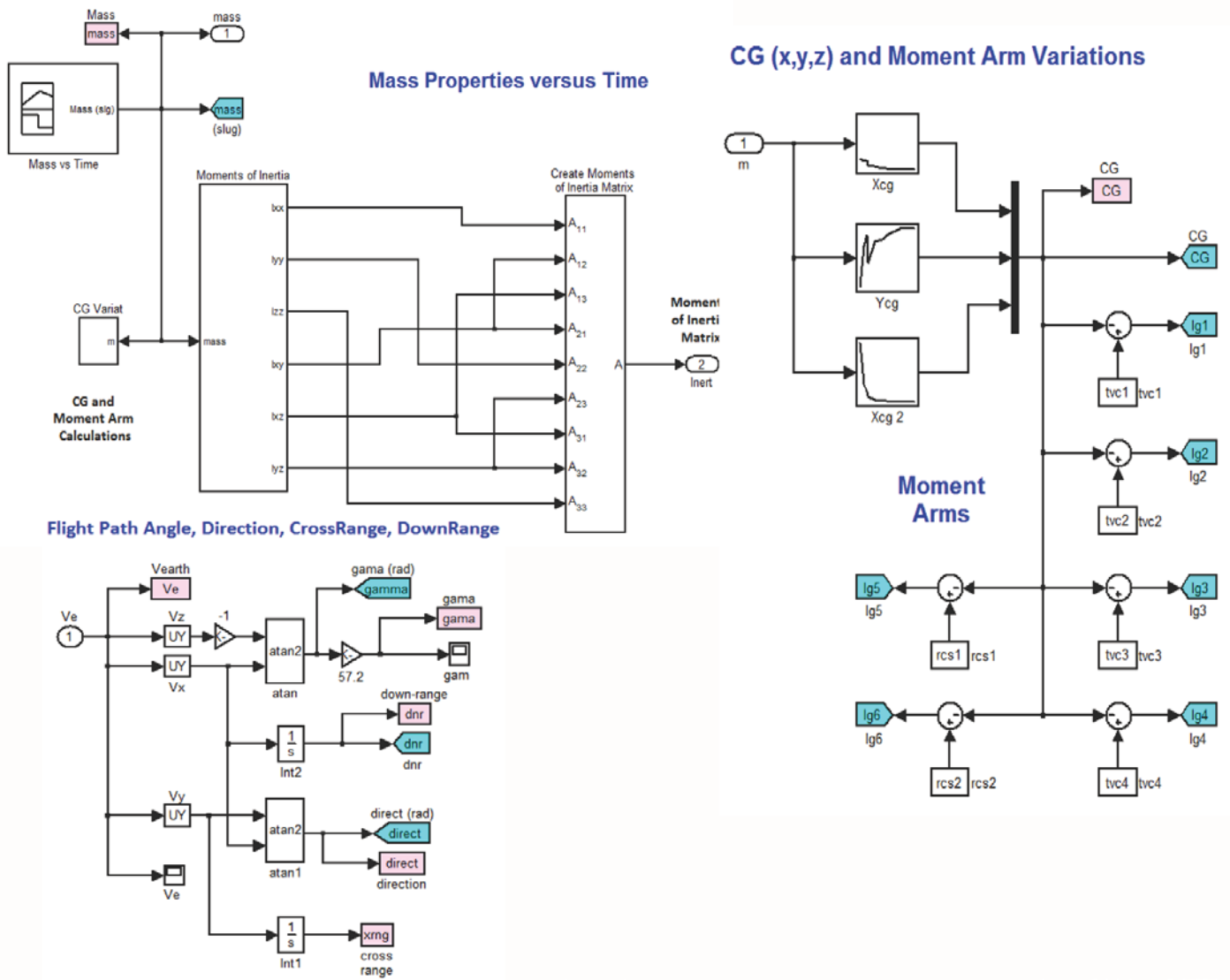
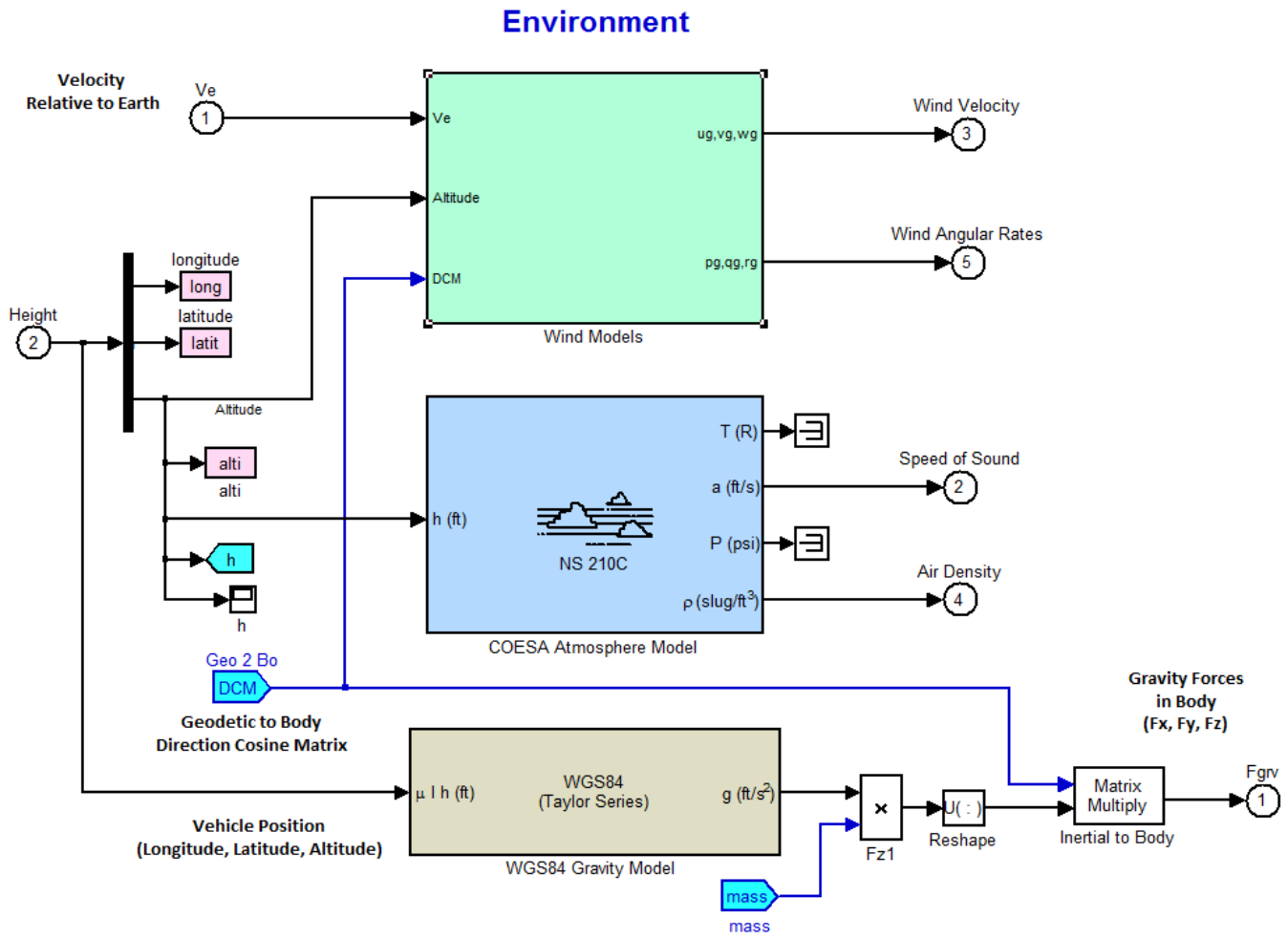


Figure 3.3 Mass-Properties Block that Calculates the Vehicle Mass and Moments of Inertia Tensor. It Includes a Subsystem that calculates the Flight-Path angle and Flight Direction.

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.



Wind Disturbance Model (Shear, Gust, Turbulence)

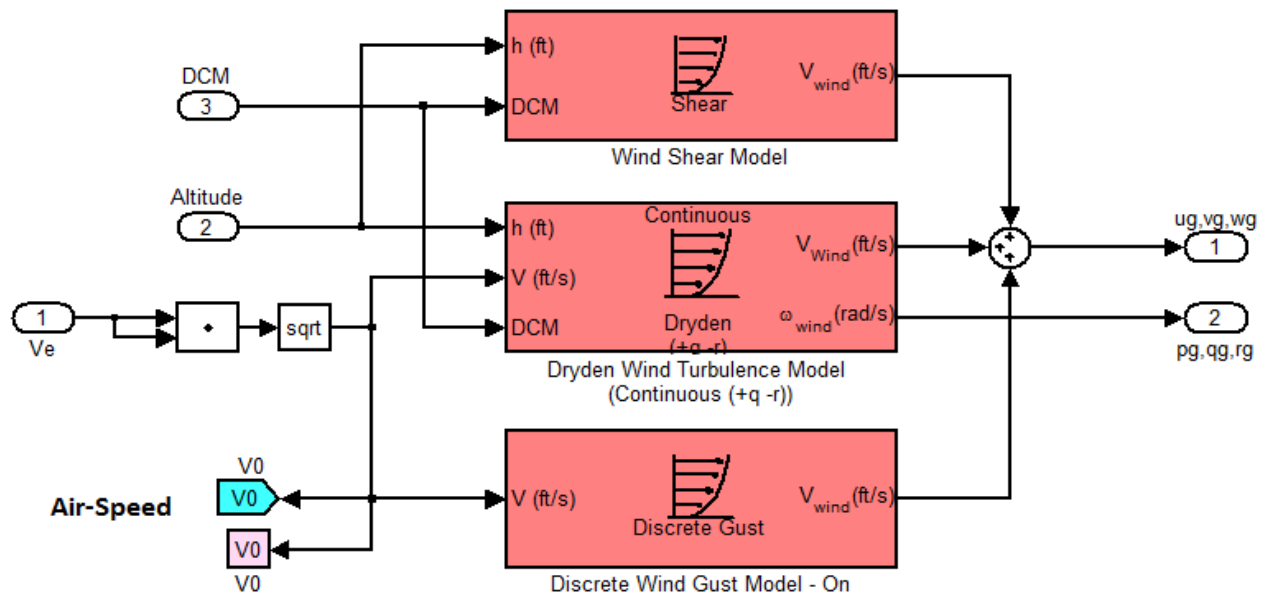


Figure 3.5 Environment and Wind Disturbances Subsystems

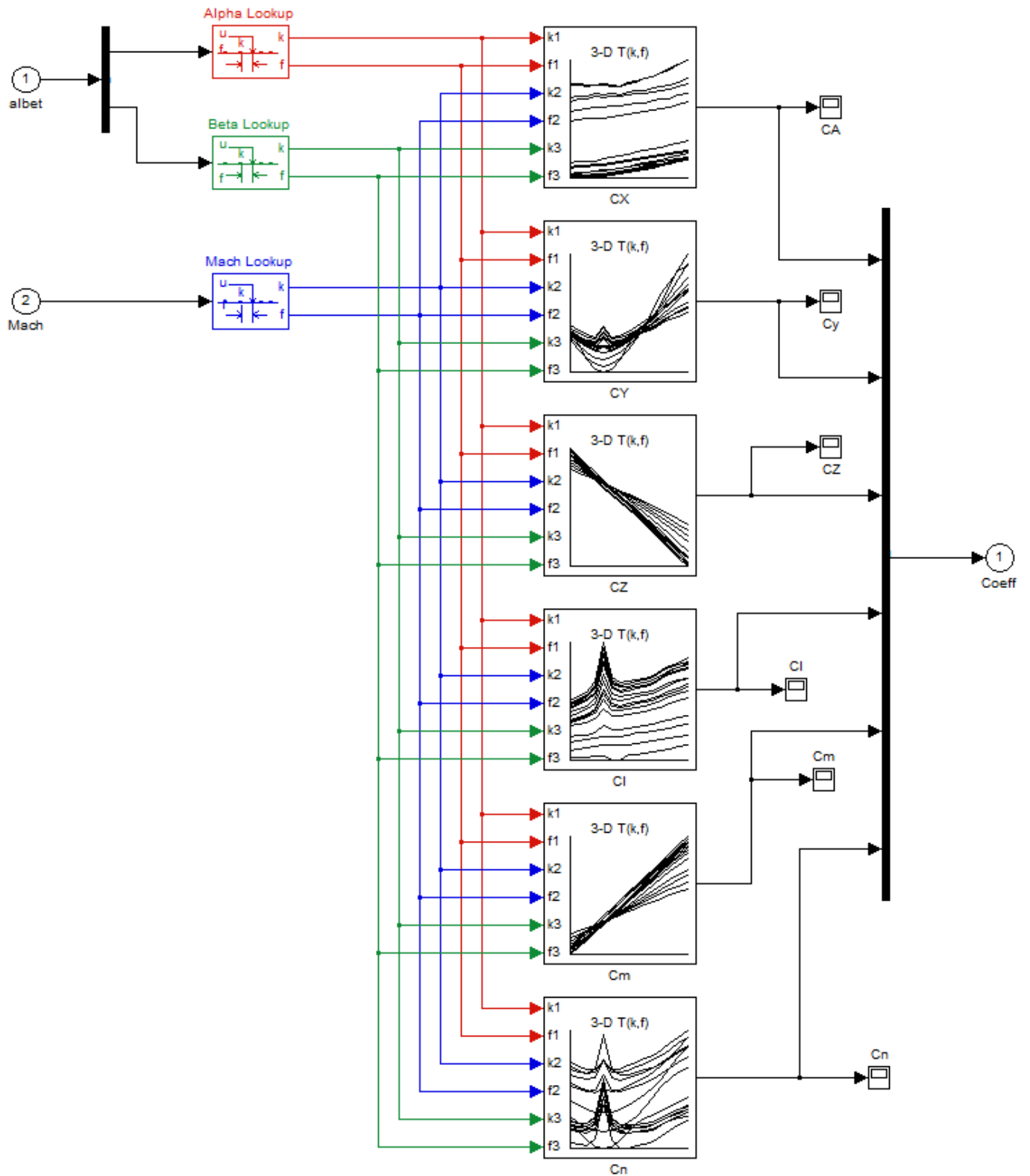


Figure 3.8 First Stage Aerodynamic Coefficients are Functions of Mach Number, Alpha, and Beta. They are loaded from file "Stage1_Aero_Coeff.m". Second Stage Aerodynamic Coefficients are loaded from file "Stage2_Aero_Coeff.m".

Moments and Forces due to Aero and Engine Thrusts

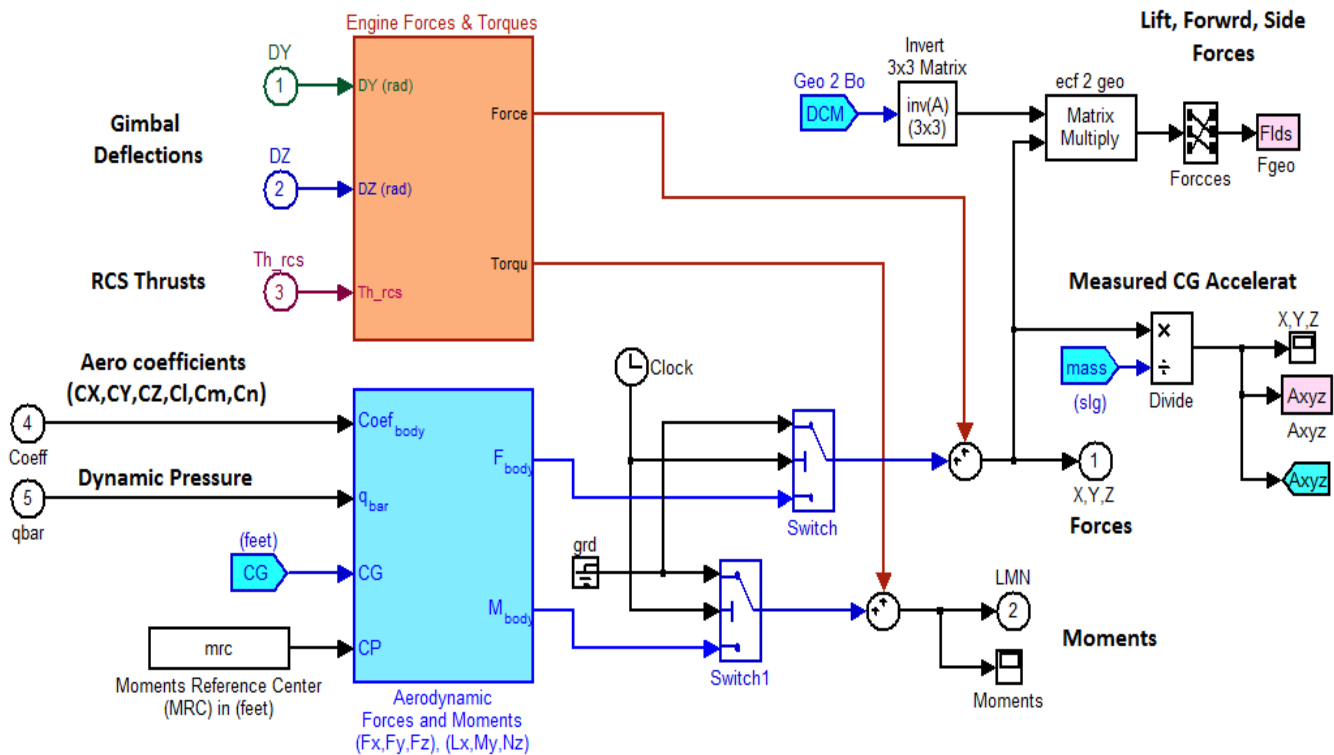


Figure 3.9 Aerodynamic and TVC Force and Moment Subsystems

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 (l_{gi}) 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.

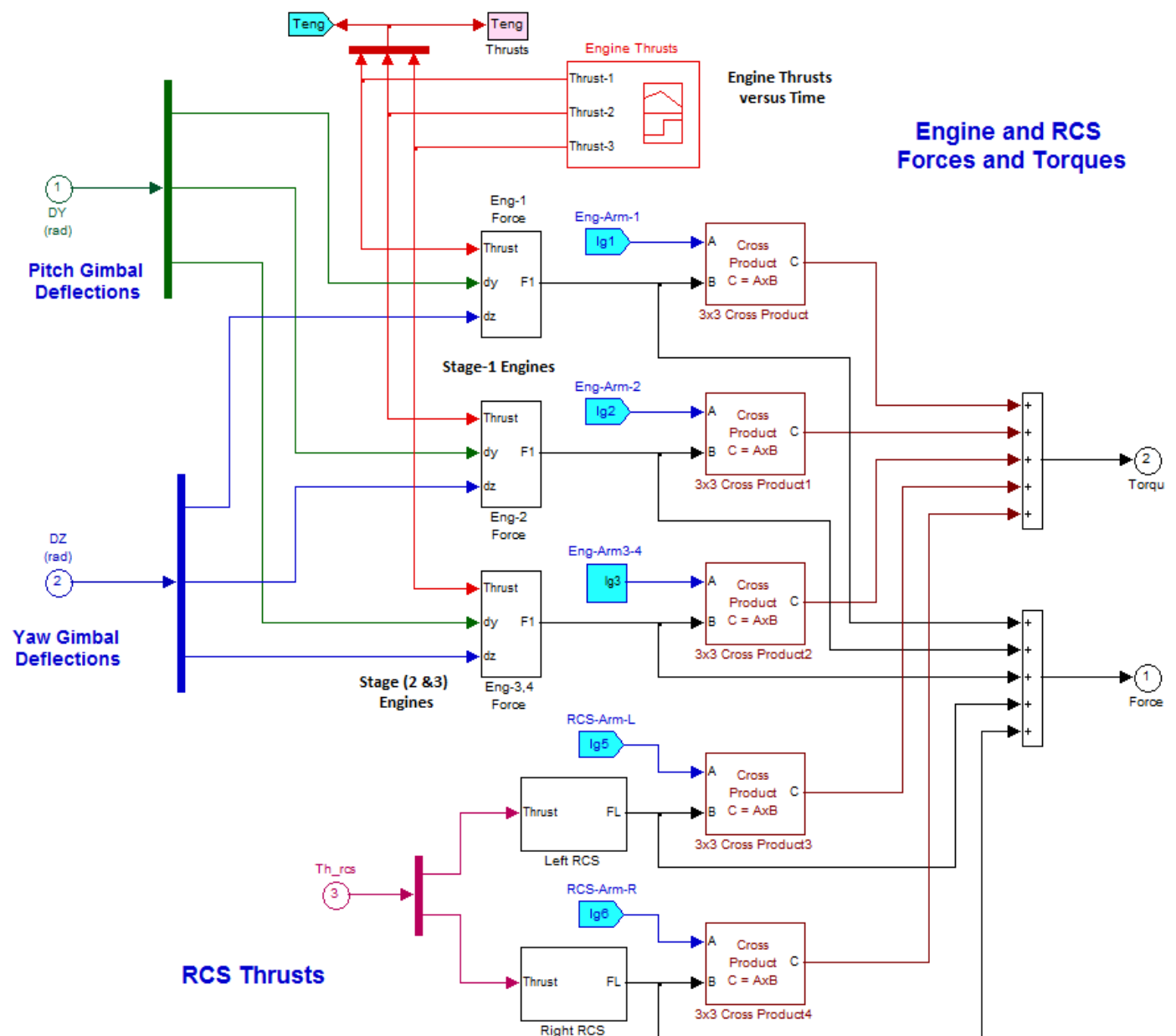


Figure 3.10 Engine Forces and Torques Subsystem

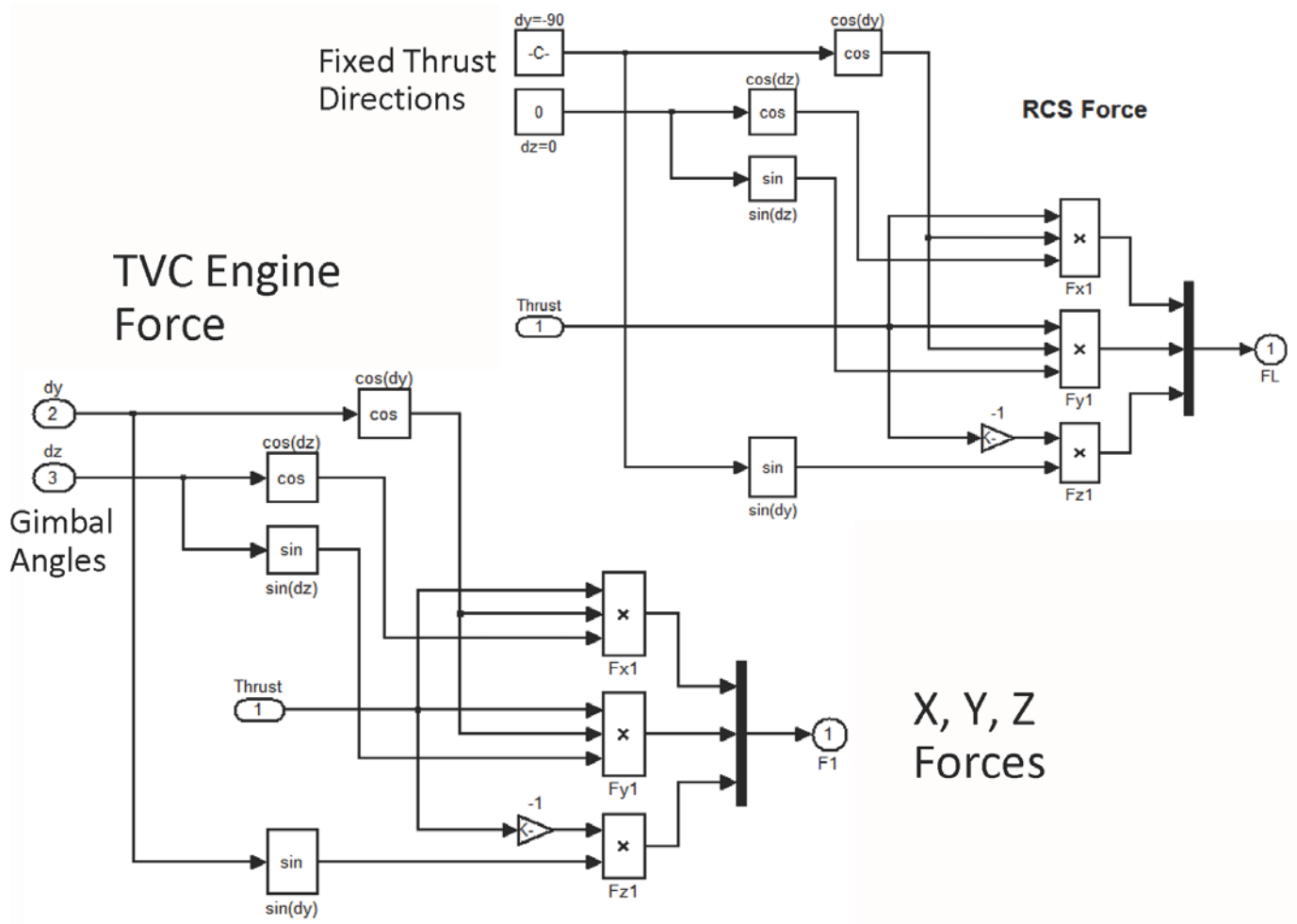


Figure 3.11 TVC and RCS X,Y, and Z Forces as a Function of Gimbal Angles and Thruster Orientation

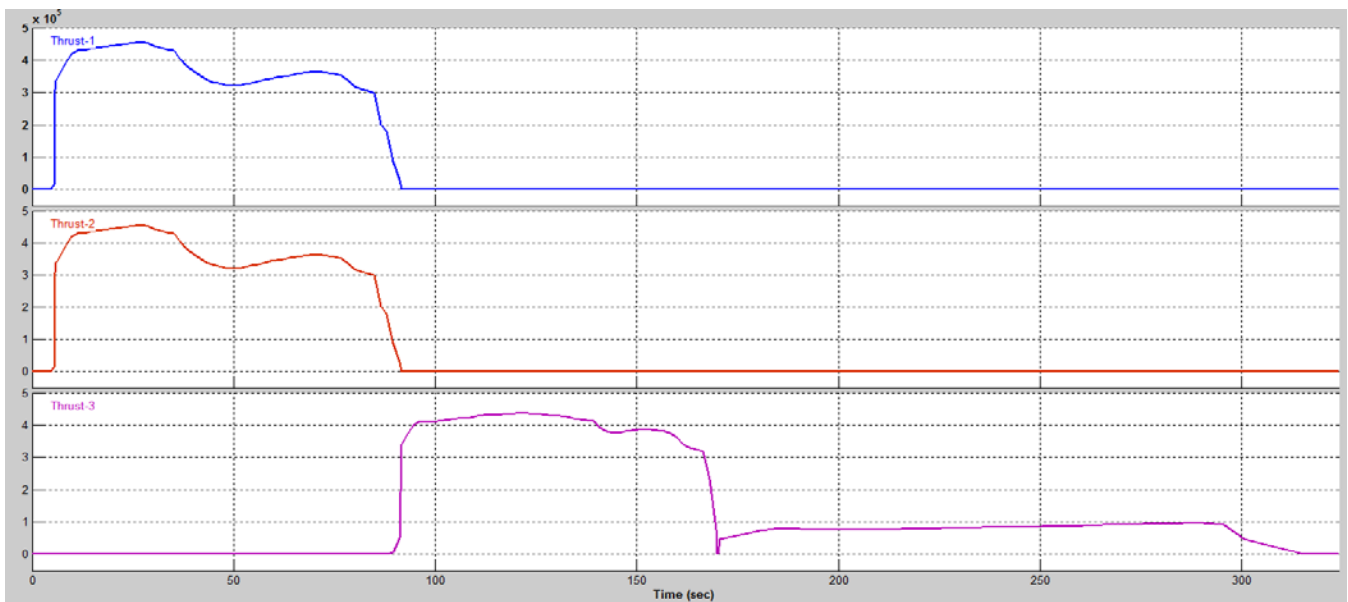


Figure 3.12 Engine Thrusts for the First, Second, and Third Stages, as a Function of Time

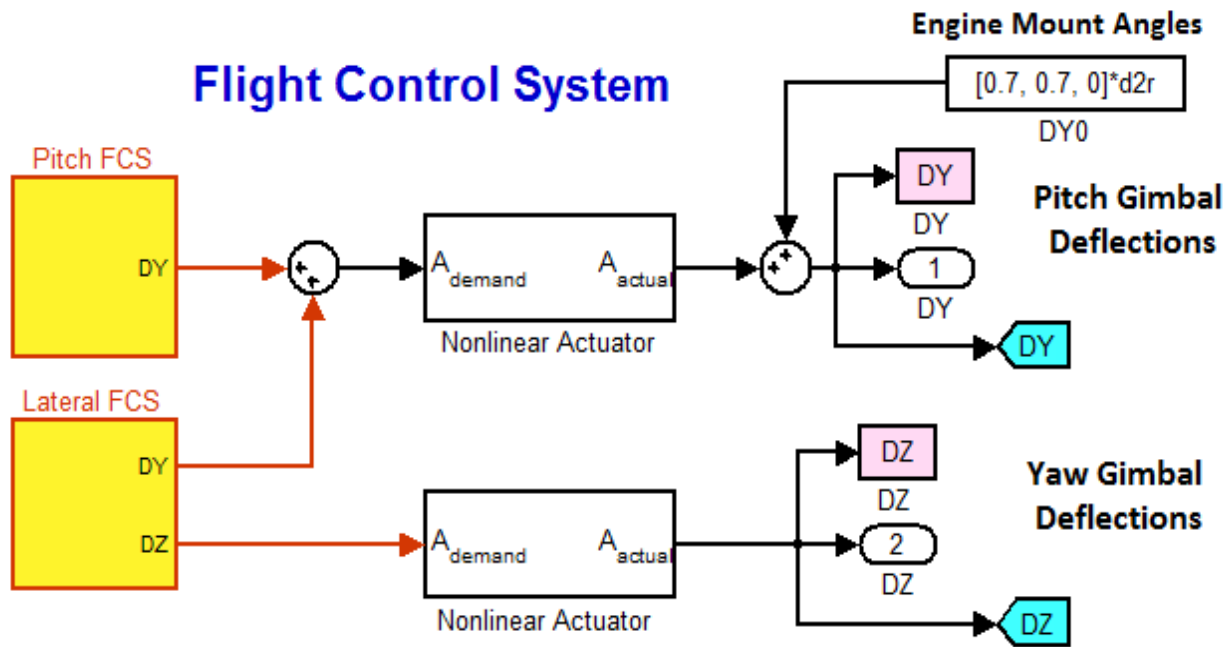


Figure 3.13 Pitch and Lateral Flight Control Systems for TVC, Including the TVC Actuators

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 $\pm 5^\circ$ 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° 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.

Pitch Flight Control System

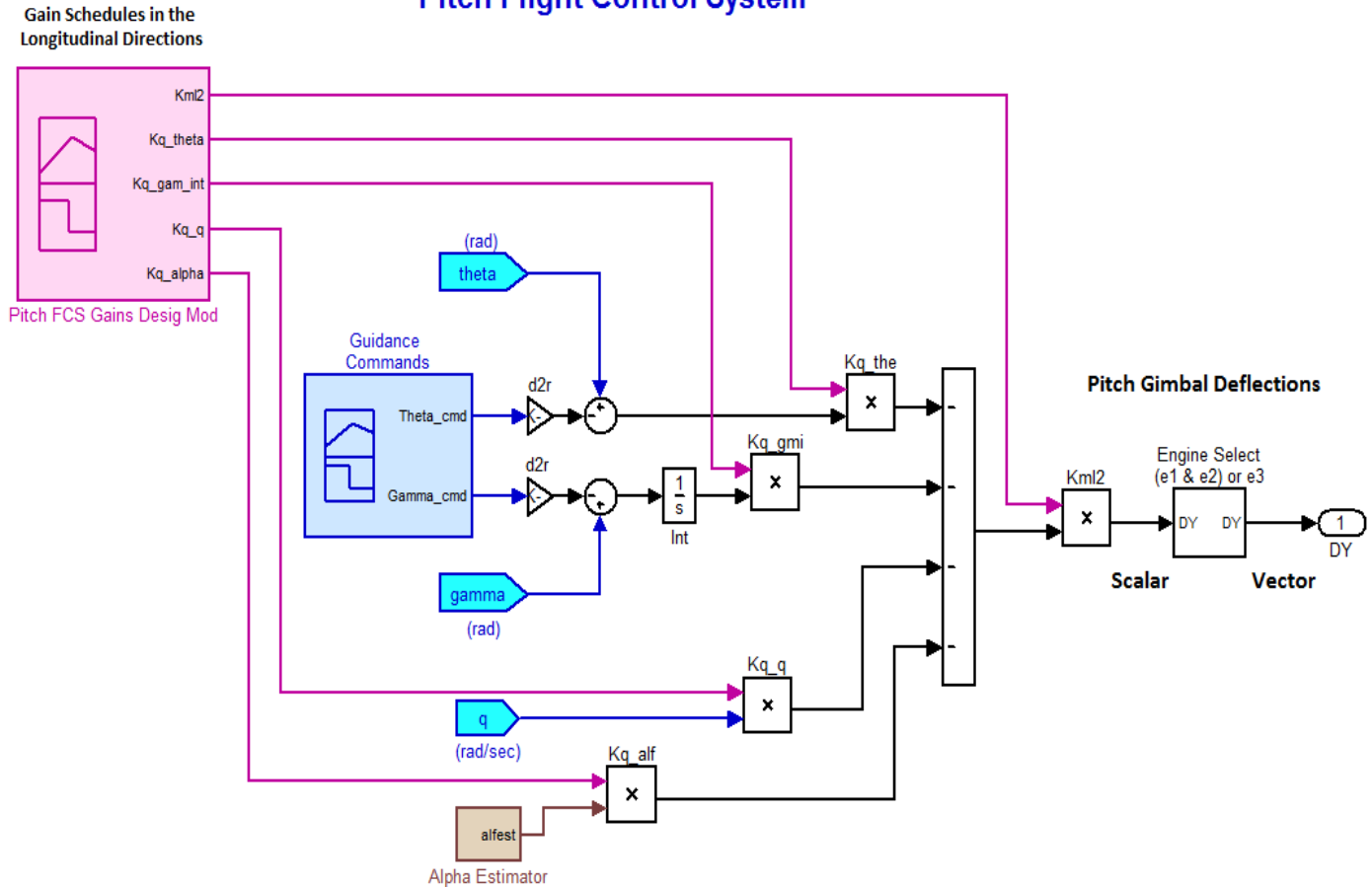


Figure 3.14 Pitch Flight Control System Includes State-Feedback, Guidance Commands, Gain Scheduling and an Angle of Attack Estimator

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 $C_{z\alpha}$ and C_{z0} versus time which are loaded from file "Estim_Gains.Mat". They are needed for calculating the normal aero force as a function of Q_{bar} . 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 $C_{y\beta}$ coefficient look-up table.

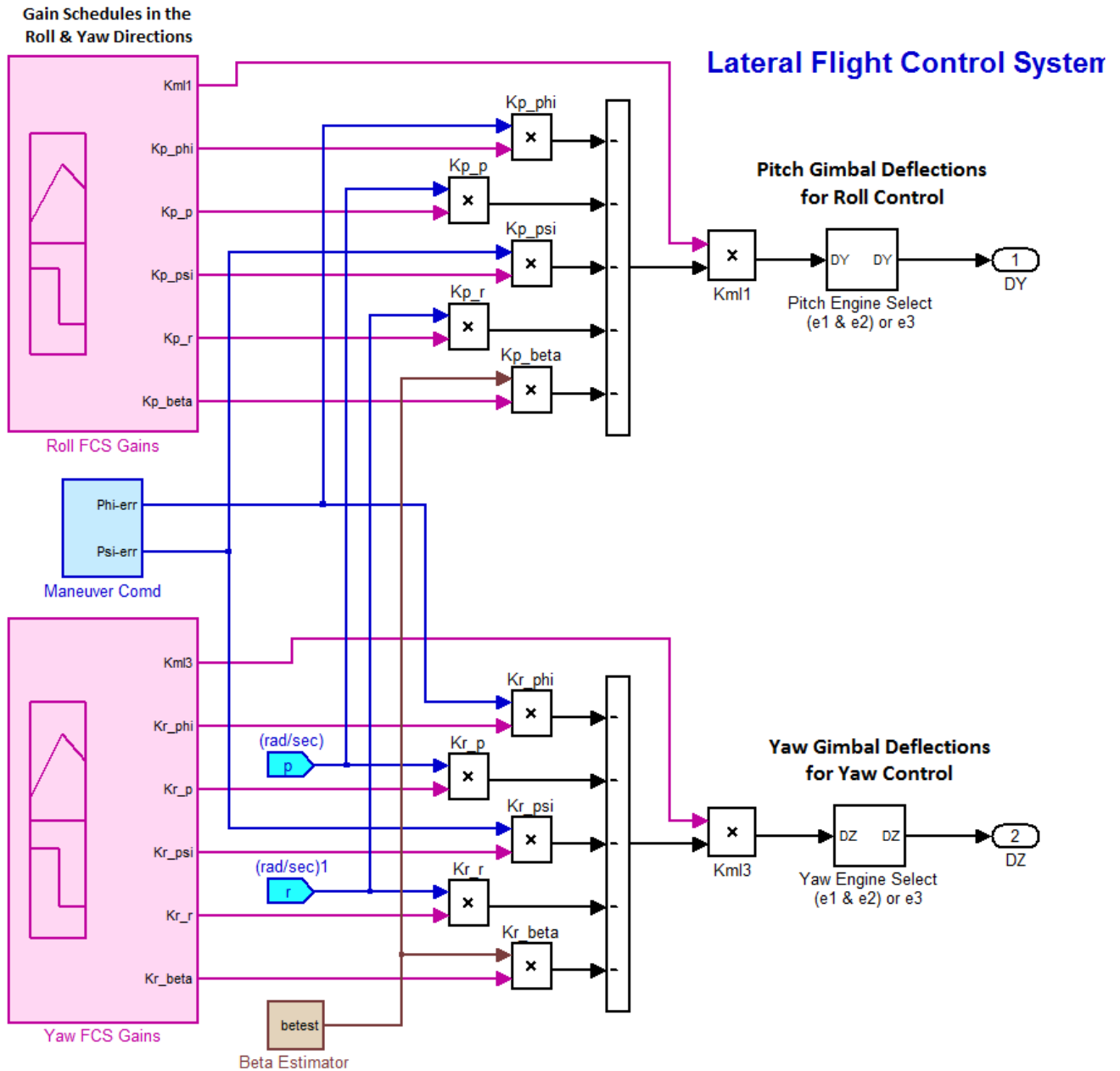


Figure 3.16 Lateral Flight Control System Includes a (2x5) State-Feedback Matrix, Directional Guidance, Gain Scheduling and an Angle of Sideslip Estimator

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.

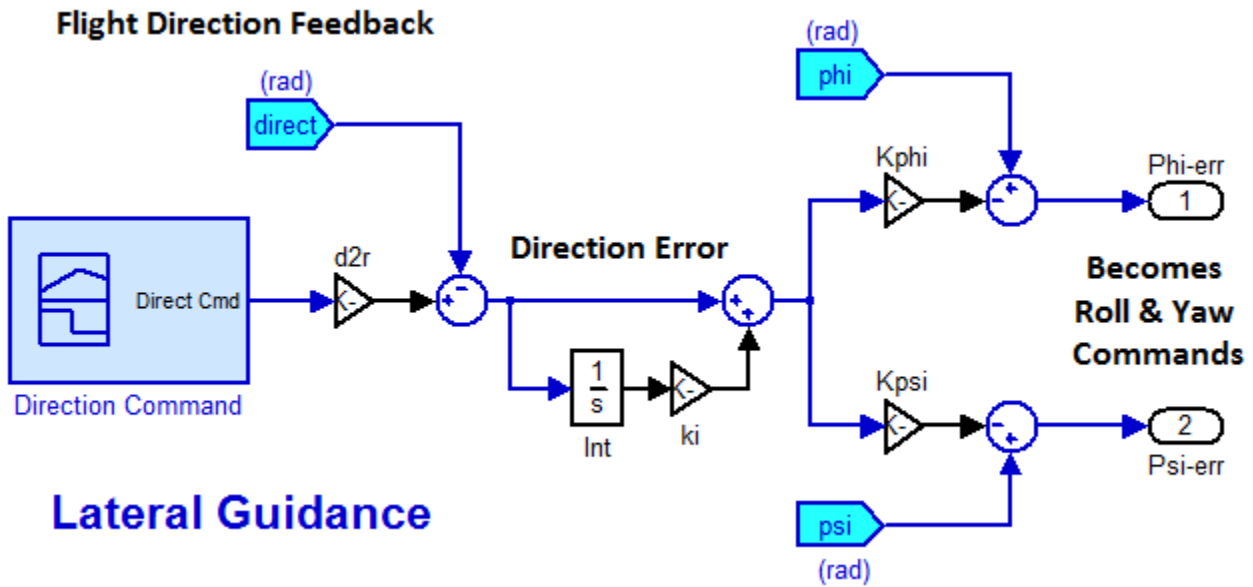


Figure 3.19 Lateral Guidance System Generates Roll and Yaw Commands to Correct Flight Direction Errors

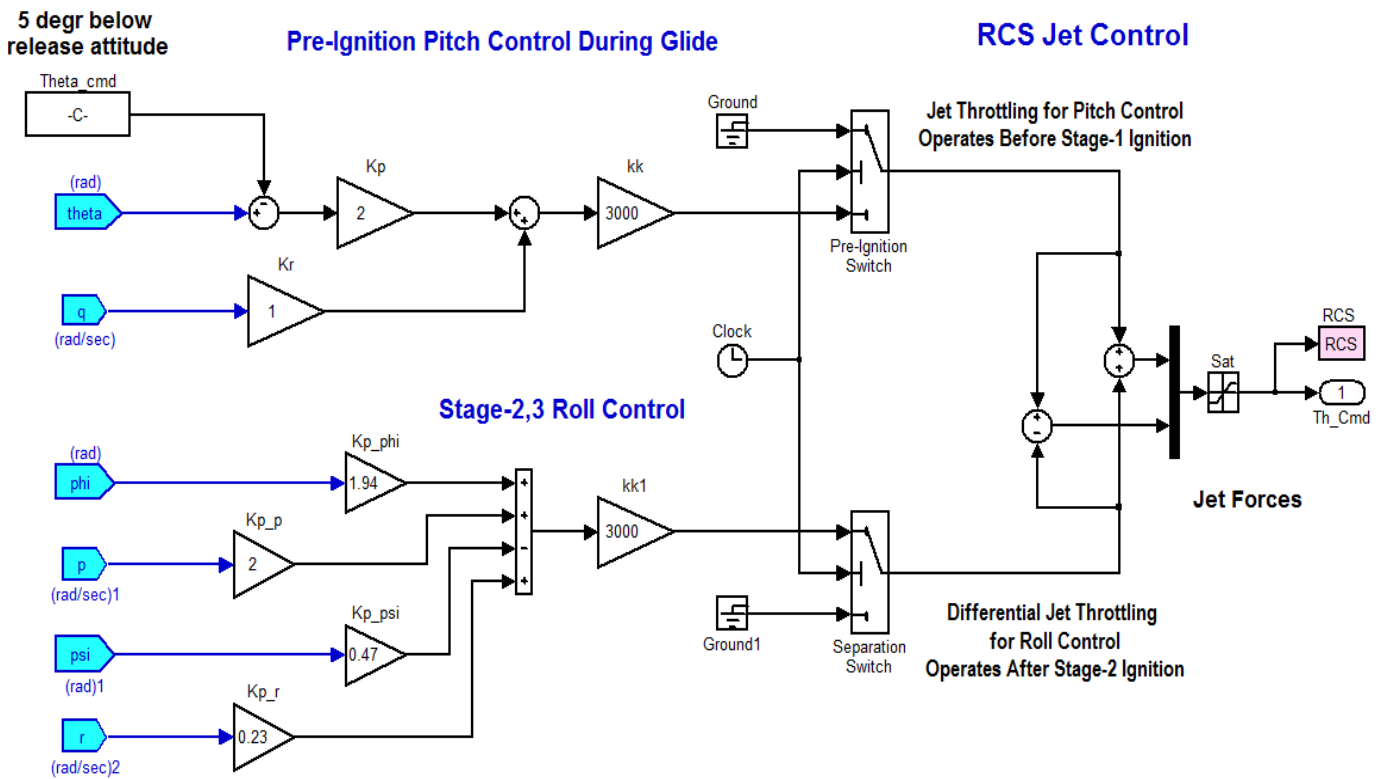


Figure 3.20 Reaction Control System for the Two Fixed Thrusters

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.

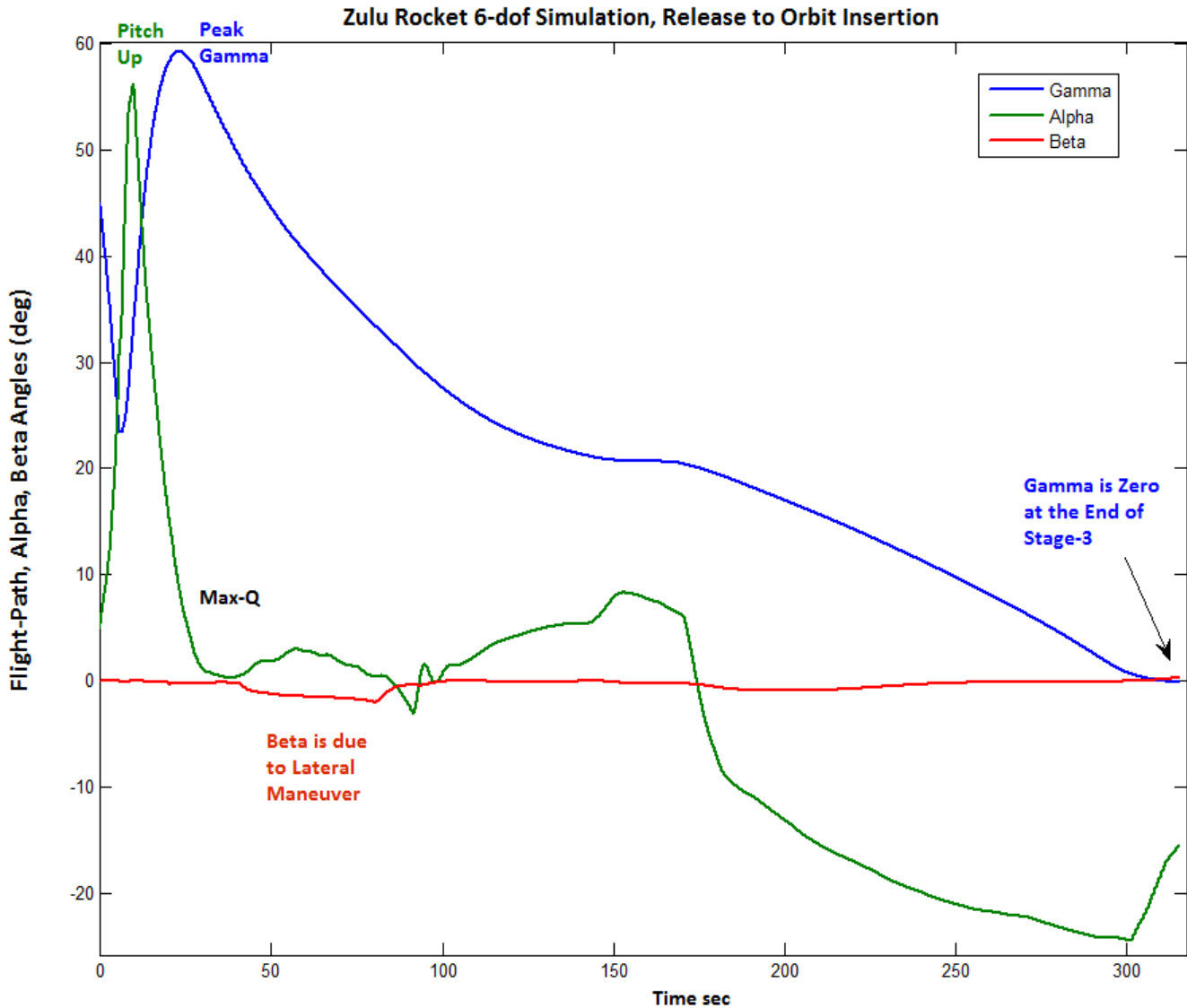


Figure 3.21 Flight Path Angle, Angle of Attack, and Angle of Sideslip

Figure 3.21 shows the flight-path angle (blue line). It begins at 45° at release and it drops to 24° 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° 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.

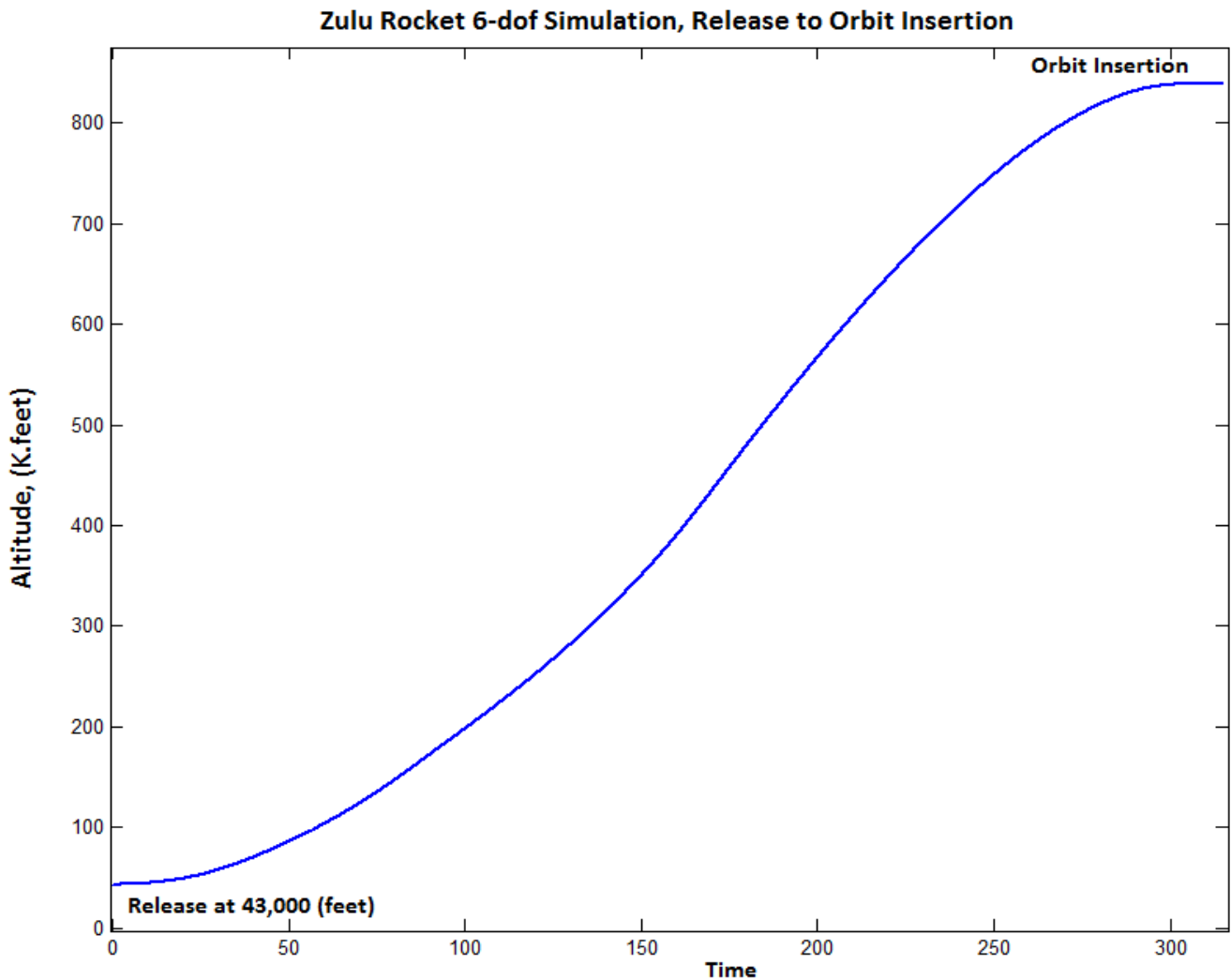


Figure 3.23 Altitude versus Time from Release to Orbit Insertion

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.

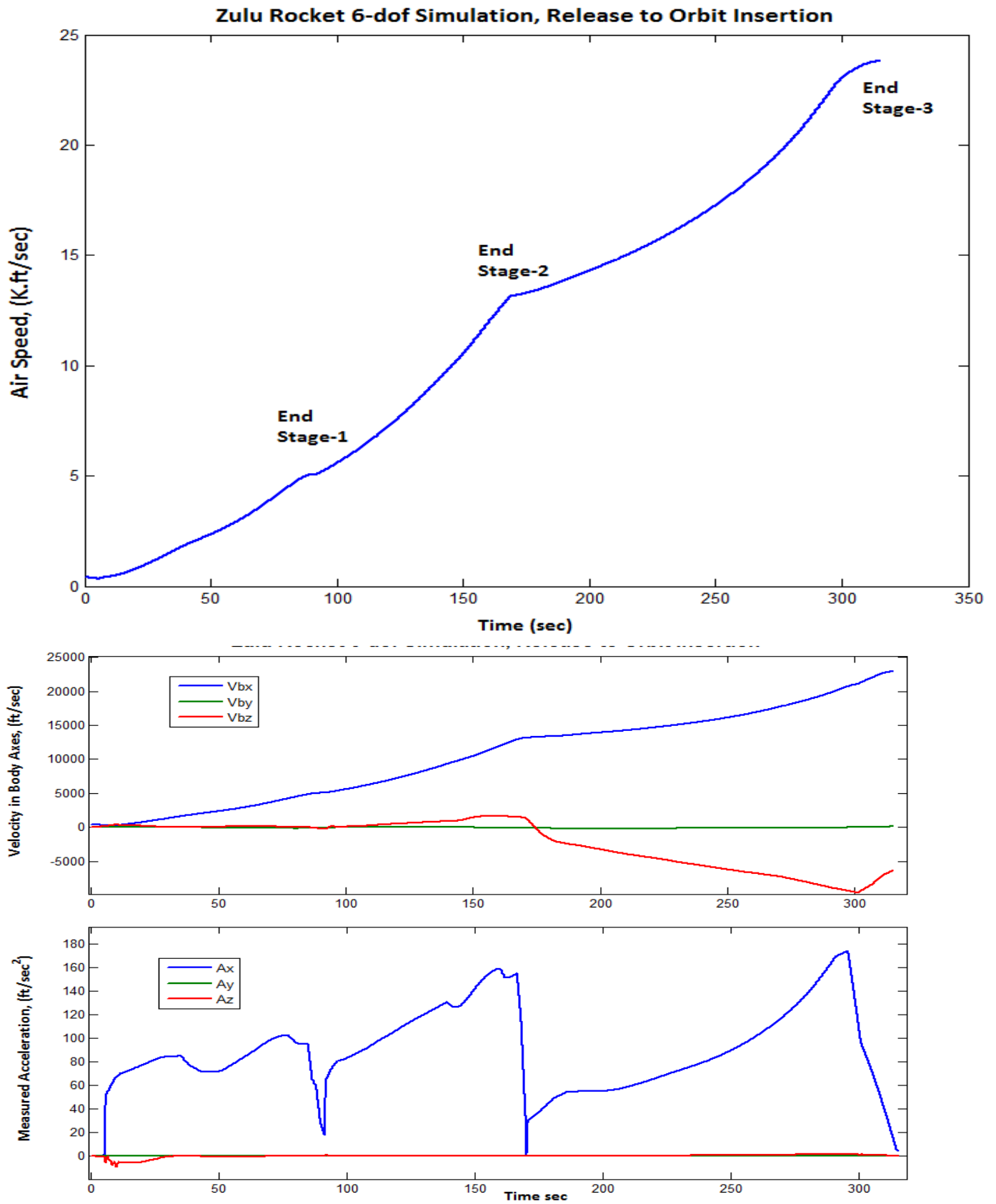


Figure 3.25 Air Speed and Accelerations during the 3 Stages

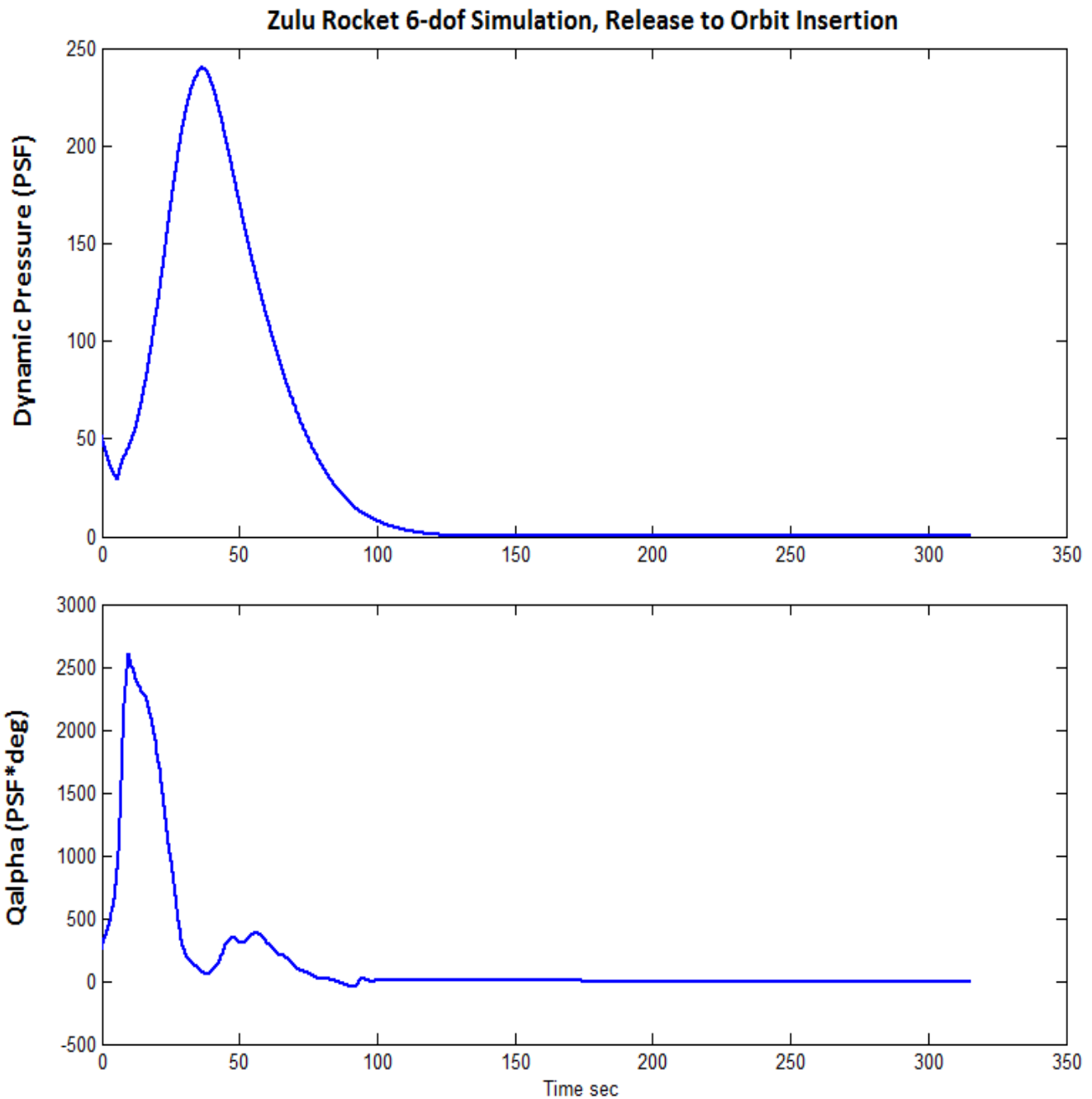


Figure 3.28 Dynamic Pressure and Qalpha. MaxQ Occurs at 40 sec and Qalpha is Less than 2600 (psf-deg)

Figure 3.28 shows the dynamic pressure that reaches a maximum of 240 (pounds/ft²) 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/ft²). 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.

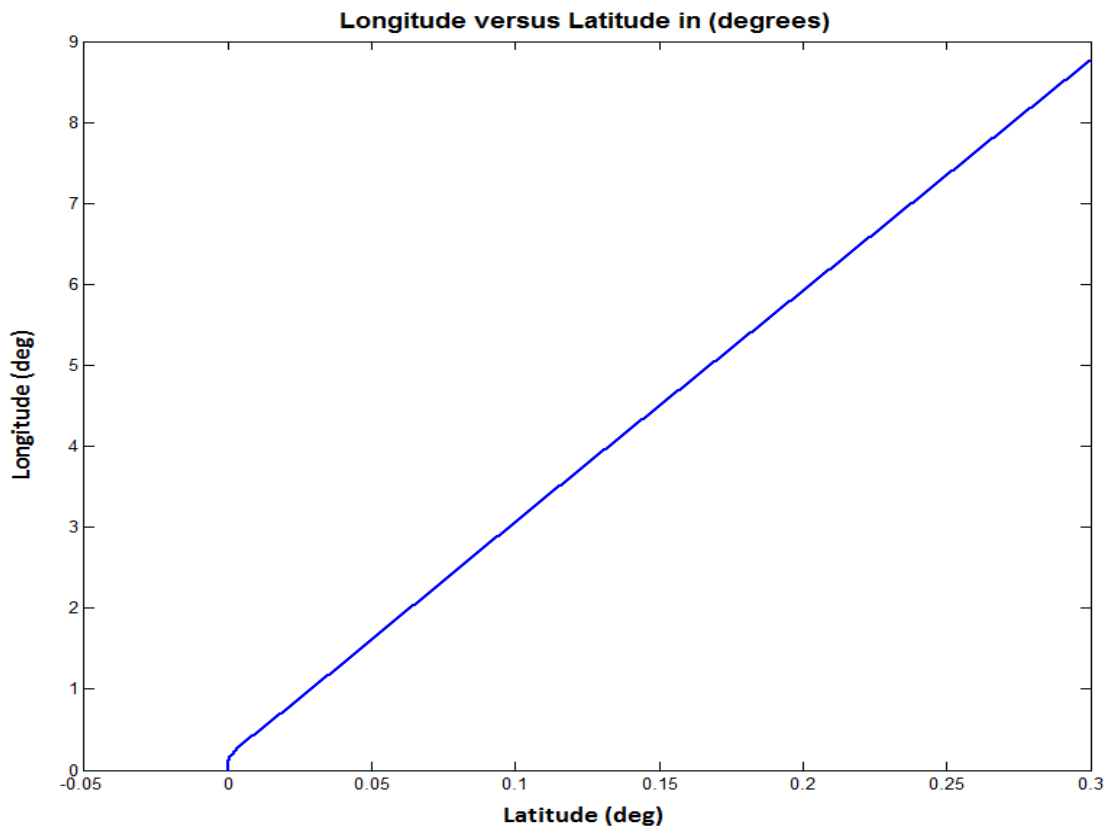
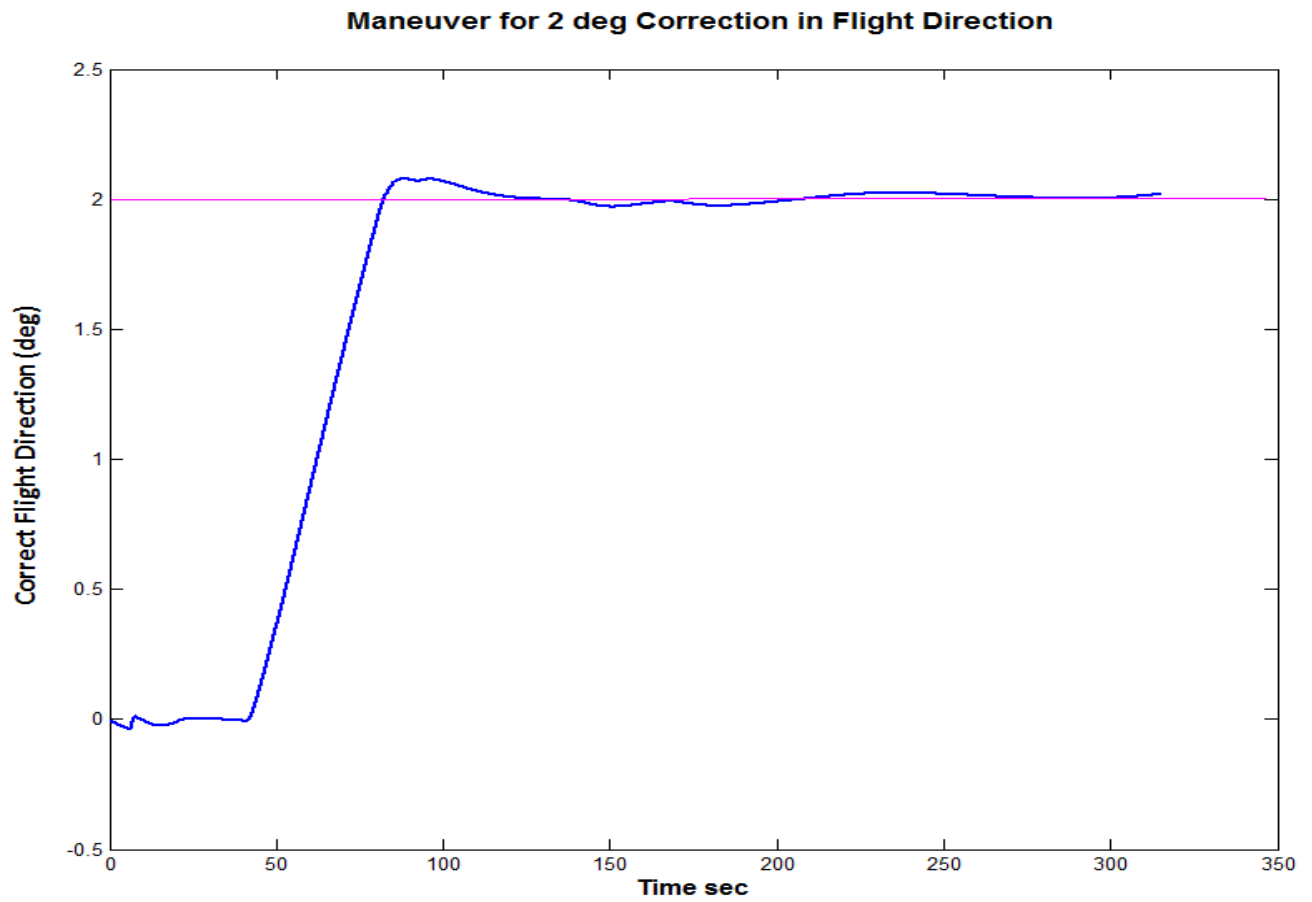


Figure 3.29 Two Degrees Maneuver to Change the Flight Direction. It shows the Modified Latitude

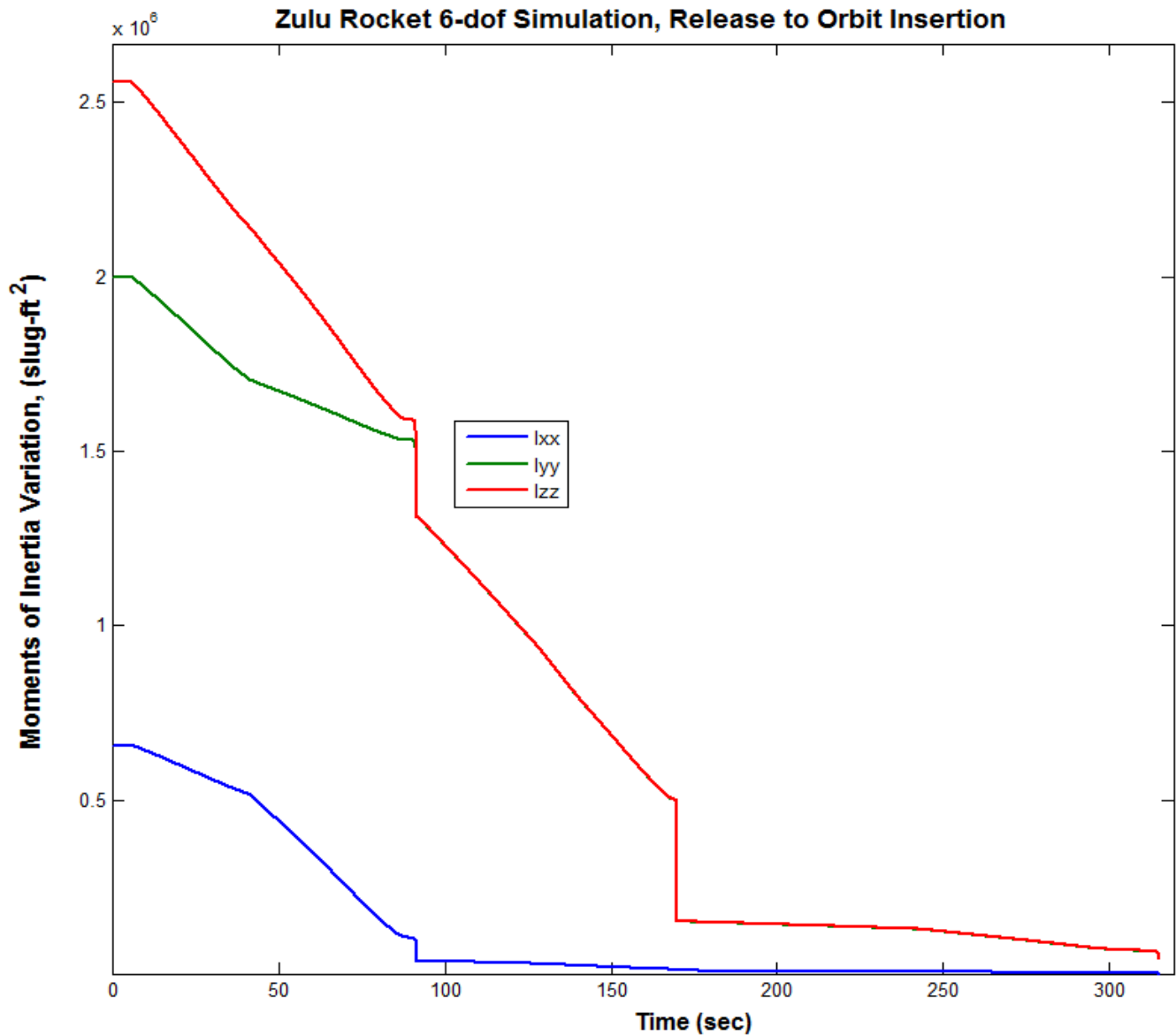


Figure 3.30 Variations of the Moments of Inertia versus Time

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.

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.

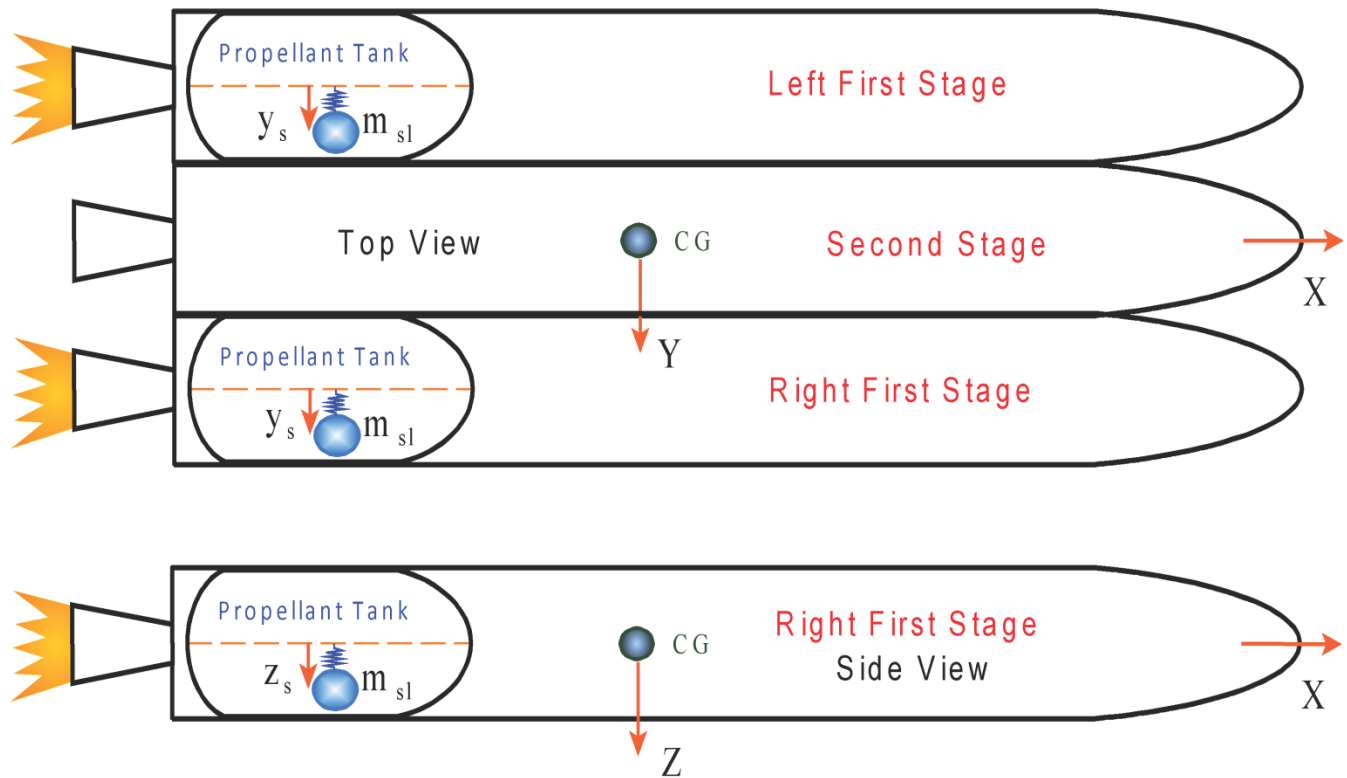


Figure 4.1 Top and Side view of the Rocket Booster during First Stage showing the two Sloshing Tanks, the Spring-Mass Model, and the Displacement of the Sloshing Masses in the y and z directions.

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 $C_l\beta$, 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 K_c 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;
Sref= 47.3; Qbar= 228; Cza=-0.343;
[Avf, Bvf, Cvf, Dvf]= vehi_pitch_Sflx;
[Avr, Bvr, Cvr, Dvr]= vehi_pitch_rb;
[Aes, Bes, Ces, Des]= estimator;
[Apc, Bpc, Cpc, Dpc]= pitch_FCS;
[Acl, Bcl, Ccl, Dcl]= closed_loop;
[Aol, Bol, Col, Dol]= open_loop;
Kc= [12.6, 3.15, -0.3, 0.37]*0.127;

% Alpha Estimator parameters
% Aero Parameters
% Load Flex Vehicle Pitch Model from Flixan
% Load RB Vehicle Pitch Model from Flixan
% Load Alpha Estimator from Flixan
% Load Pitch Controller from Flixan
% Load Closed Loop Pitch System from Flixan
% Load the Open Loop Pitch System from Flixan
% Pitch Gains from Design

% Frequency Response Analysis File Run.m
d2r= pi/180; r2d=1/d2r;
init
% [Ap,Bp,Cp,Dp]= linmod('Ptch_OpenRB');
% [Ap,Bp,Cp,Dp]= linmod('Ptch_Open4');
[Ap,Bp,Cp,Dp]= linmod('Ptch_OpenFLX');
sys= SS(Ap, Bp, Cp, Dp);
w=logspace(-2, 2.3, 12000);
figure(1); Nichols(sys,w)
figure(2); Bode(sys,w)

```

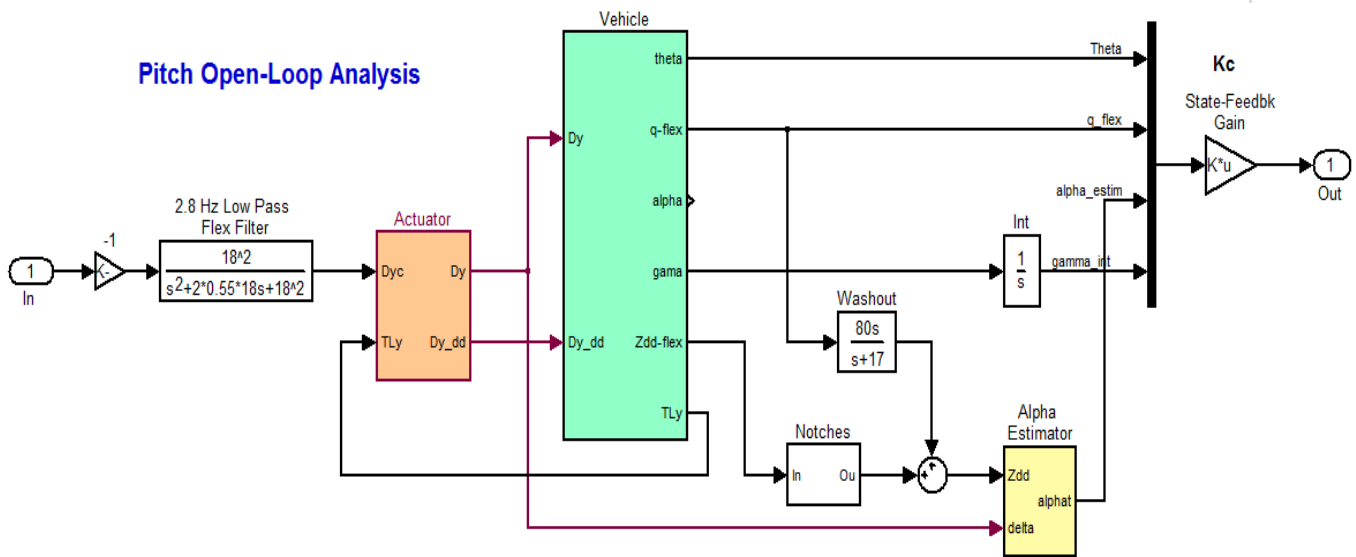


Figure 4.2 Open-Loop Diagram of the Simulink Model “Ptch_OpenFLX.mdl” used for Frequency Response Analysis

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.

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

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_v . 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.

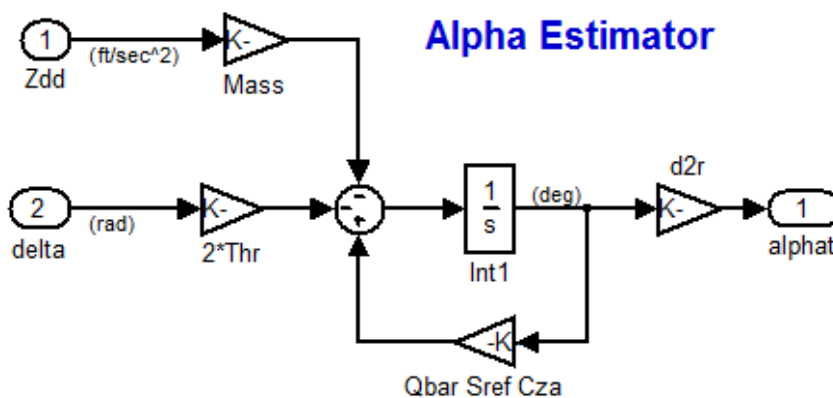
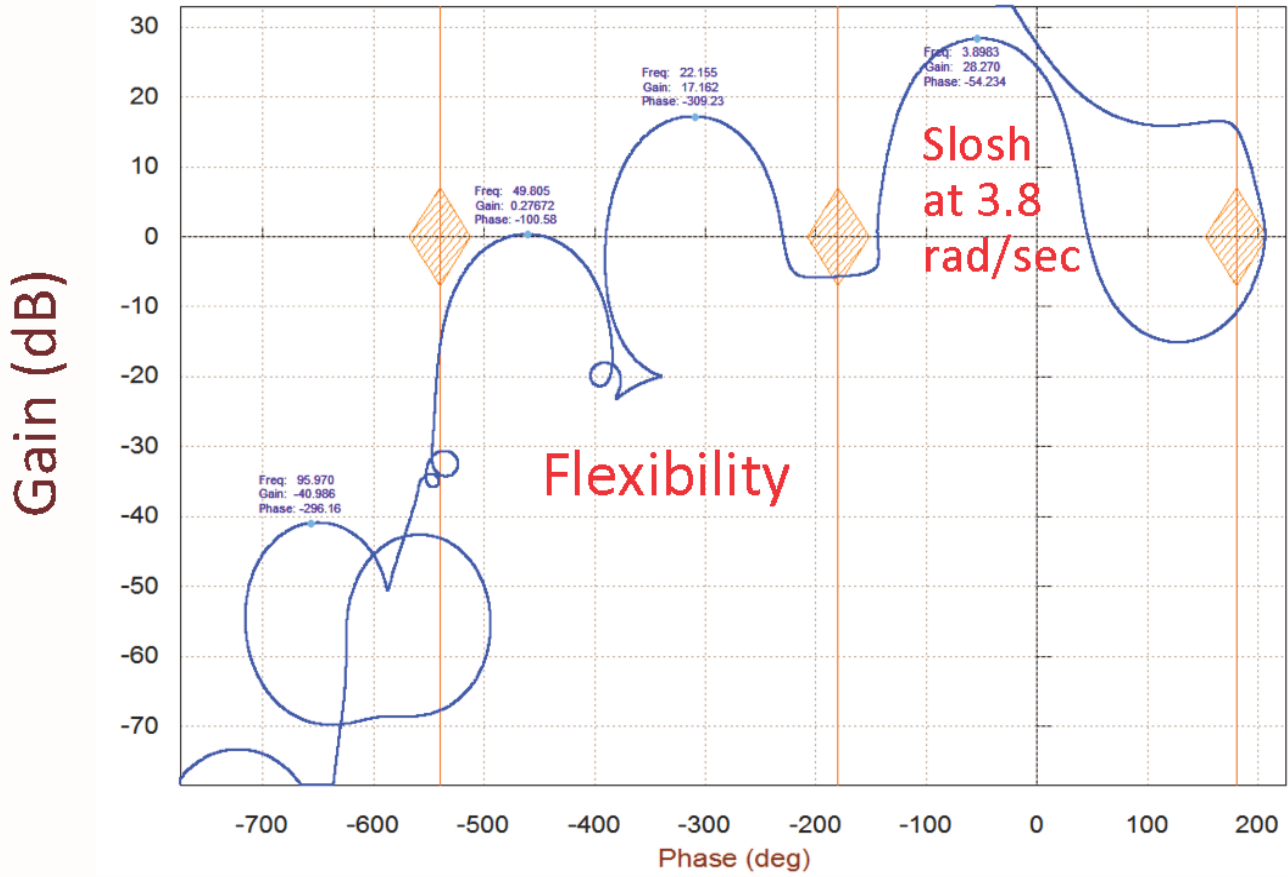


Figure 4.3 Angle of Attack Estimator

Nichols Plot for: Out(1)-FCS Out / Inpt(1)-Gimbal In , of:
Pitch Open-Loop System



Bode Plot for: Out(1)-FCS Out / Inpt(1)-Gimbal In , of:
Pitch Open-Loop System

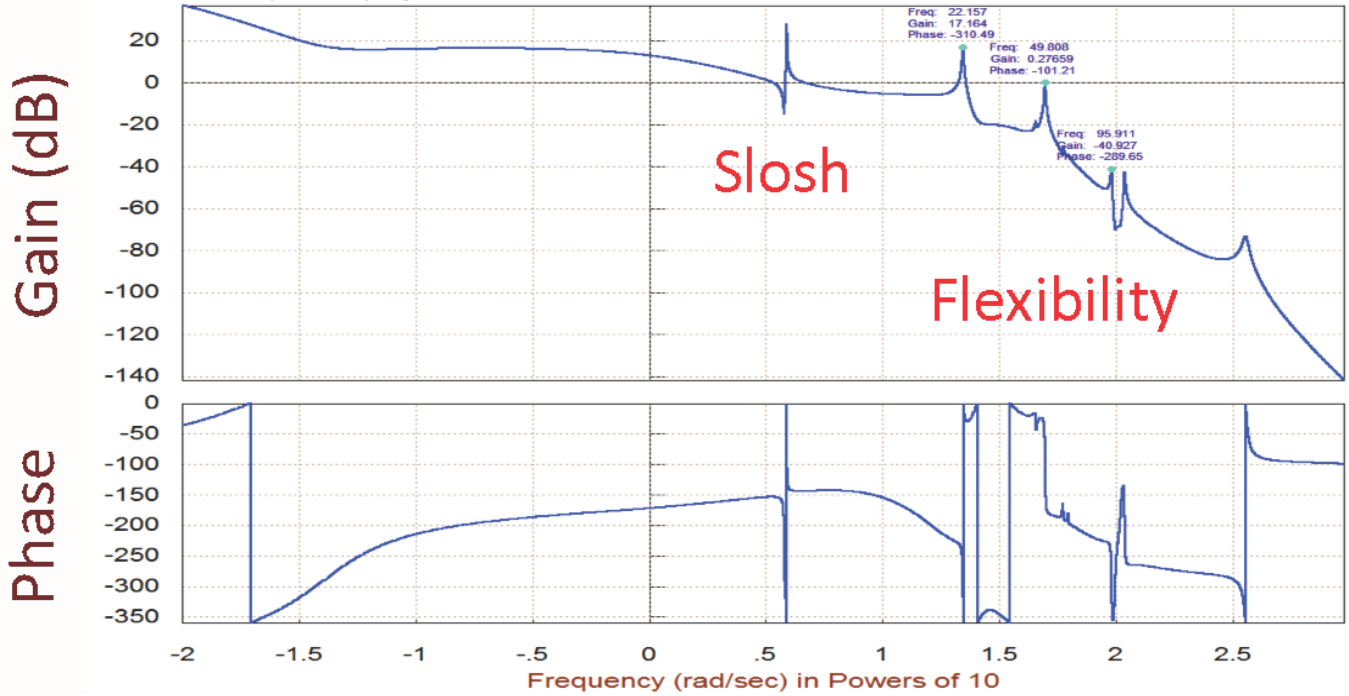
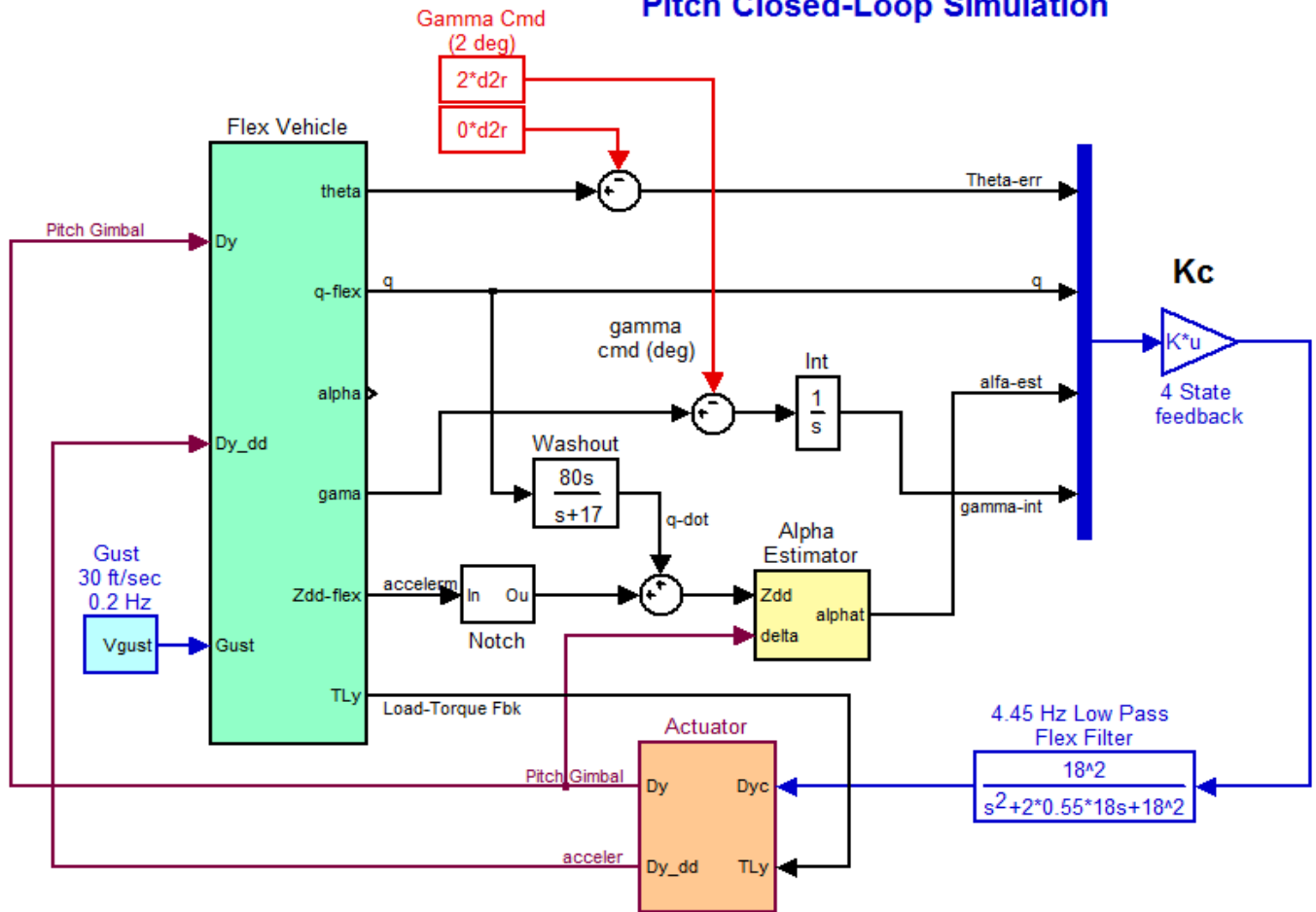


Figure 4.4 Pitch Axis Stability Analysis Using the Open-Loop Model

Pitch Closed-Loop Simulation



Flight Vehicle Pitch Dynamics

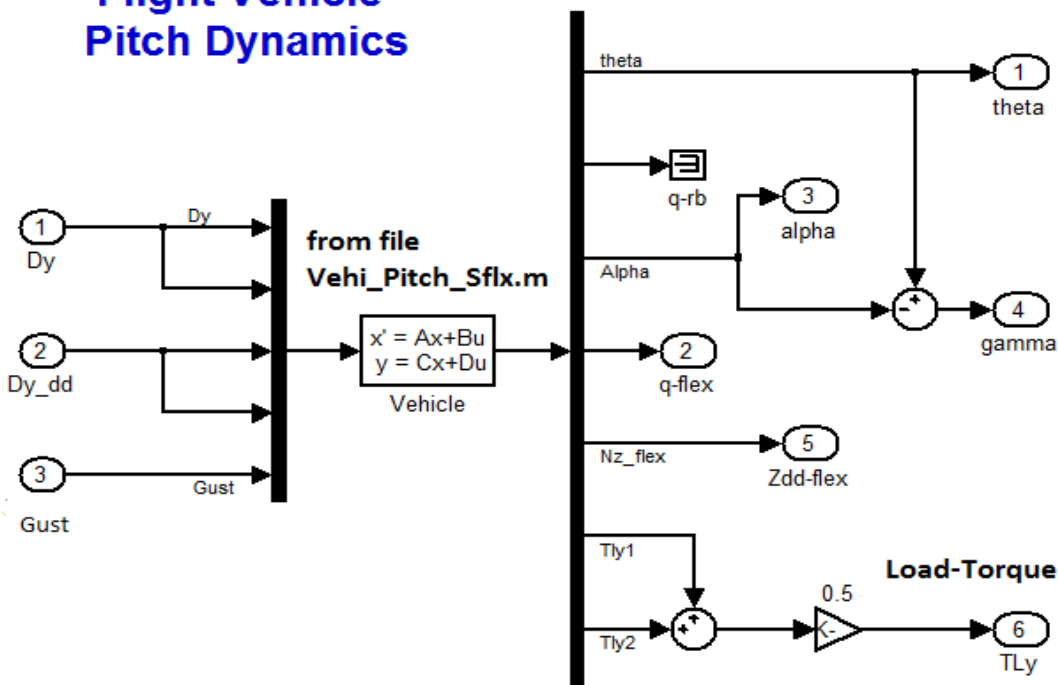
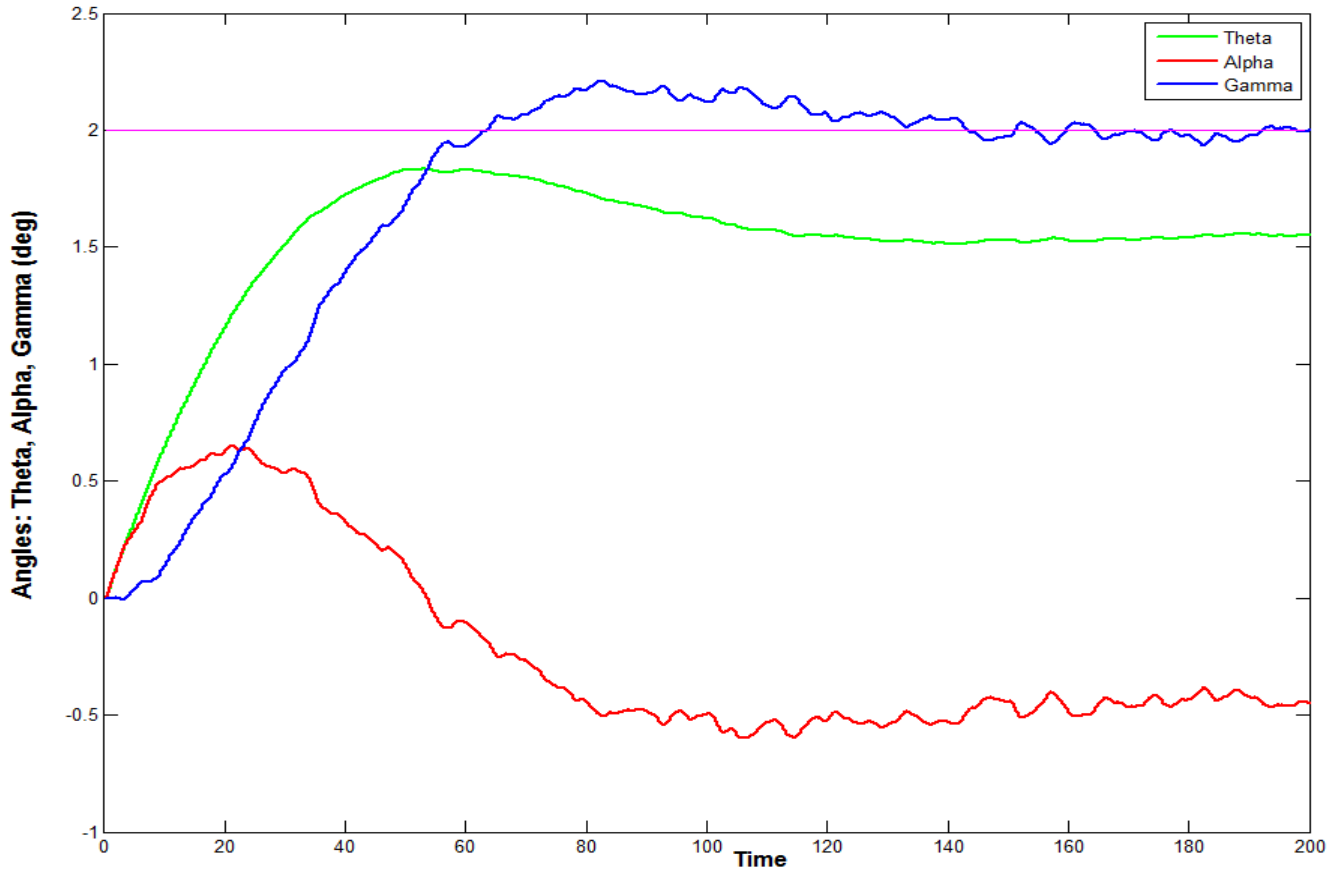
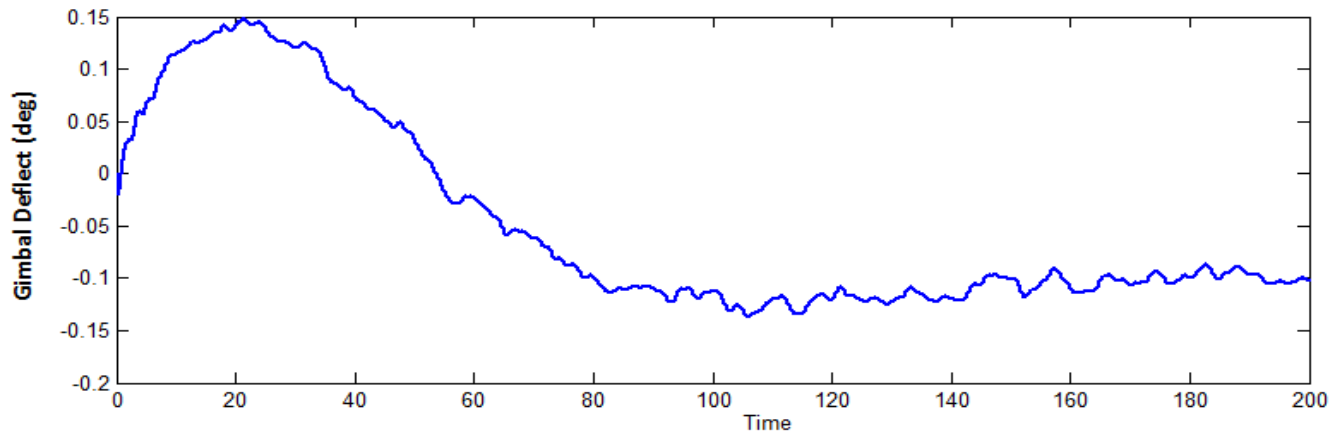
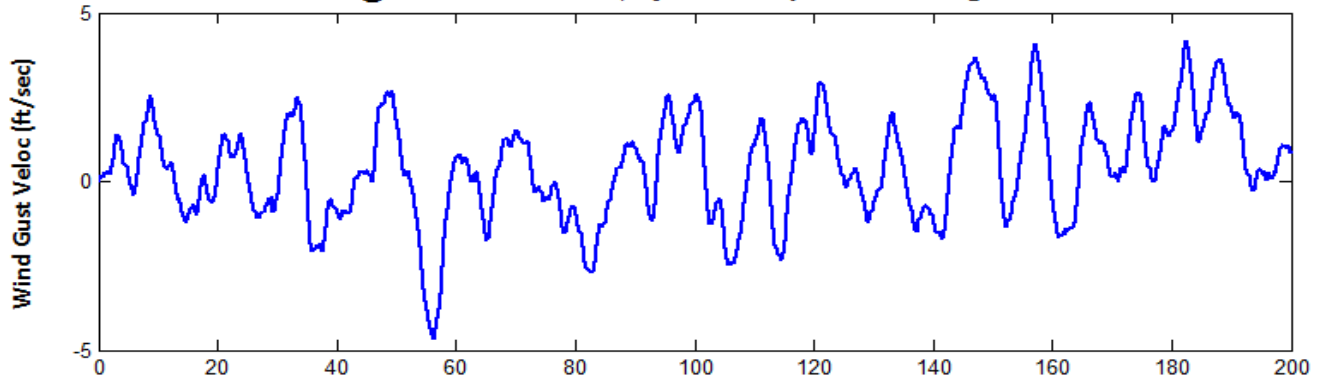


Figure 4.5 Closed-Loop Simulation Model in File "Ptch_Sim.Mdl"

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



4.2.6 Creating the Models by Using Flixan Utilities

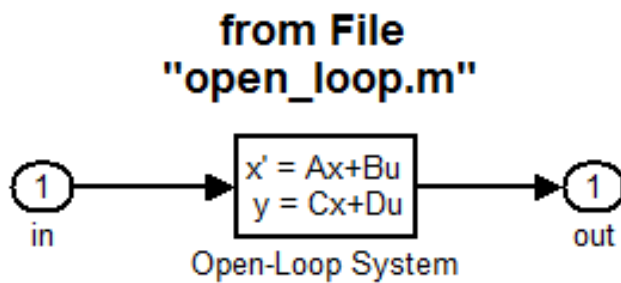
The open and closed loop systems used in the previous analysis were implemented using Simulink. The Simulink models were then linearized and converted into state-space form using the “Linmod” utility in Matlab. The Linmod utility, however, is not always reliable and in some situations it may create problems, 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.

Pitch Open-Loop System Ptch_Open4.Mdl Implemented Entirely Using Flixan



Pitch Closed-Loop System Implemented Entirely Using Flixan

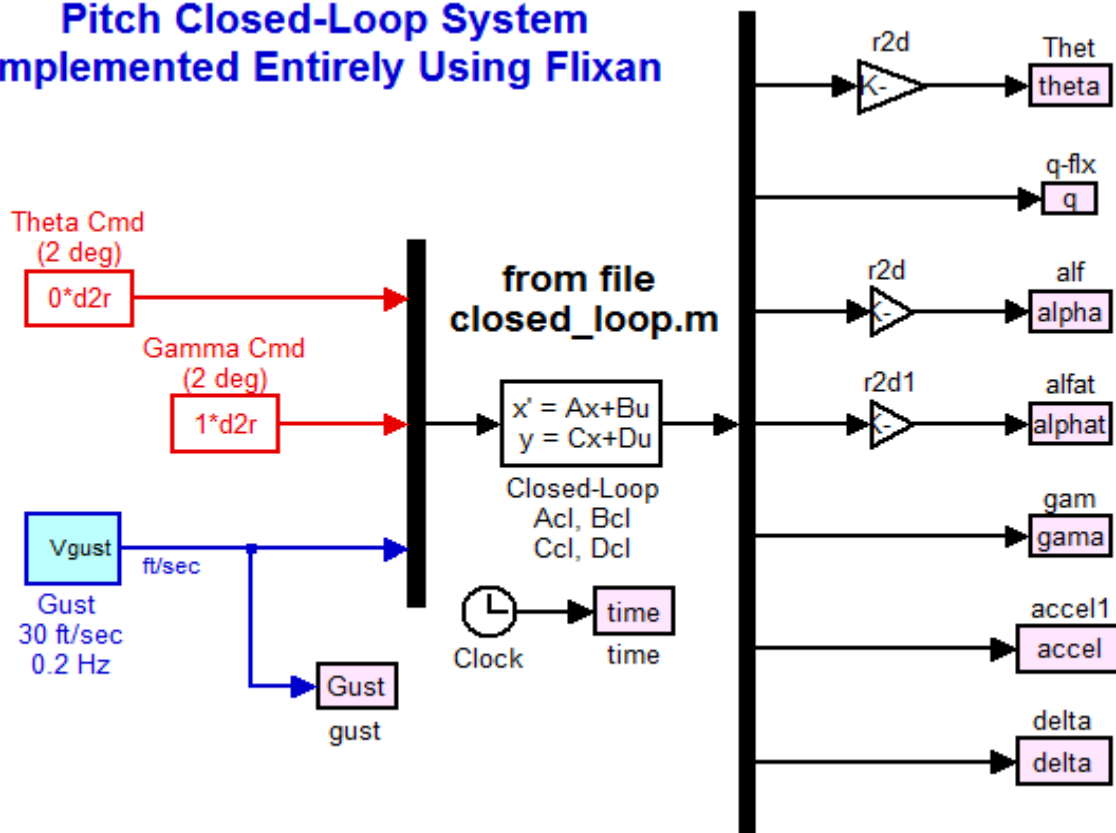


Figure 4.8 Open and Closed-Loop State Systems Implemented Using Flixan. They are included in Simulink Files Ptch_Open4.Mdl and Ptch_Sim_4.Mdl respectively

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;

Q= [2, 0.1, 2, 0.2, 0.01]; Q=diag(Q);
R= [1, 1]*1.0; R=diag(R);
[Klat, s, e] = LQR(Ad,Bd,Q,R);
save Klat.mat Klat -ascii

% Load the Lateral LQR Design Model
% State Weights: [phi, p, psi, r, beta]=
% Control Weights
% Lateral State-Feedback Matrix
```

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 N_y , 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.

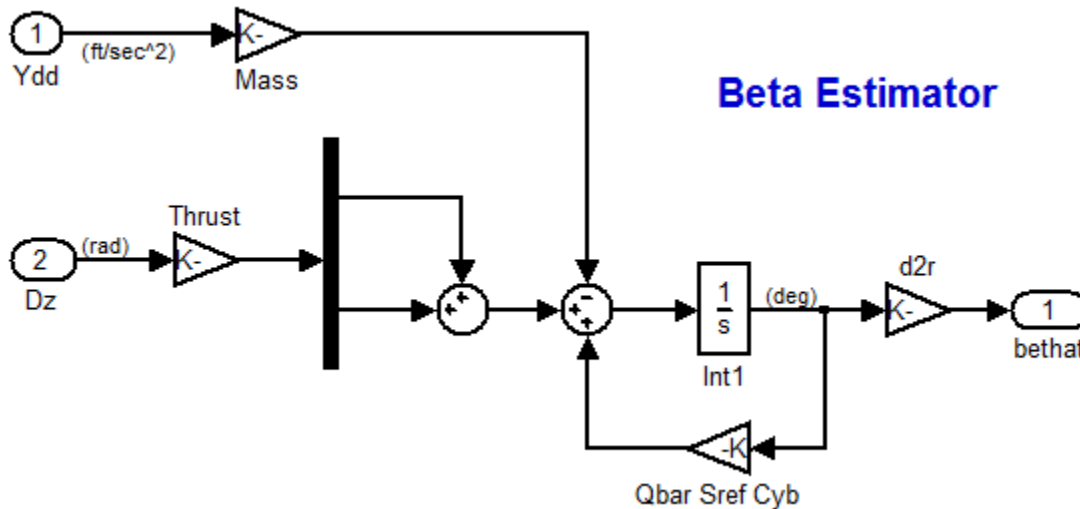


Figure 4.9 Angle of Sideslip Estimator

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.

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.

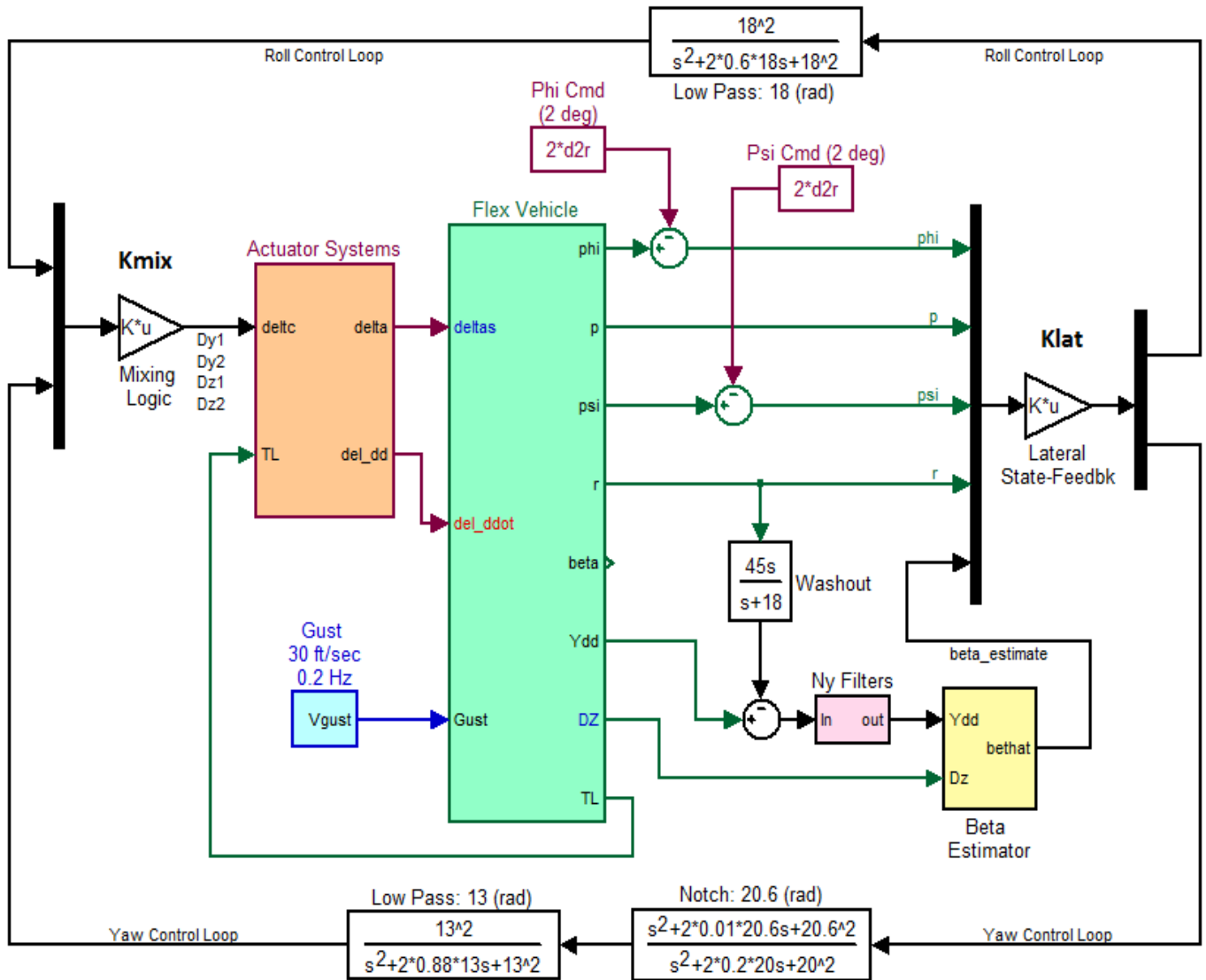


Figure 4.10 Roll and Yaw Simulation Model “*Latr_Sim.Mdl*”

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° 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.

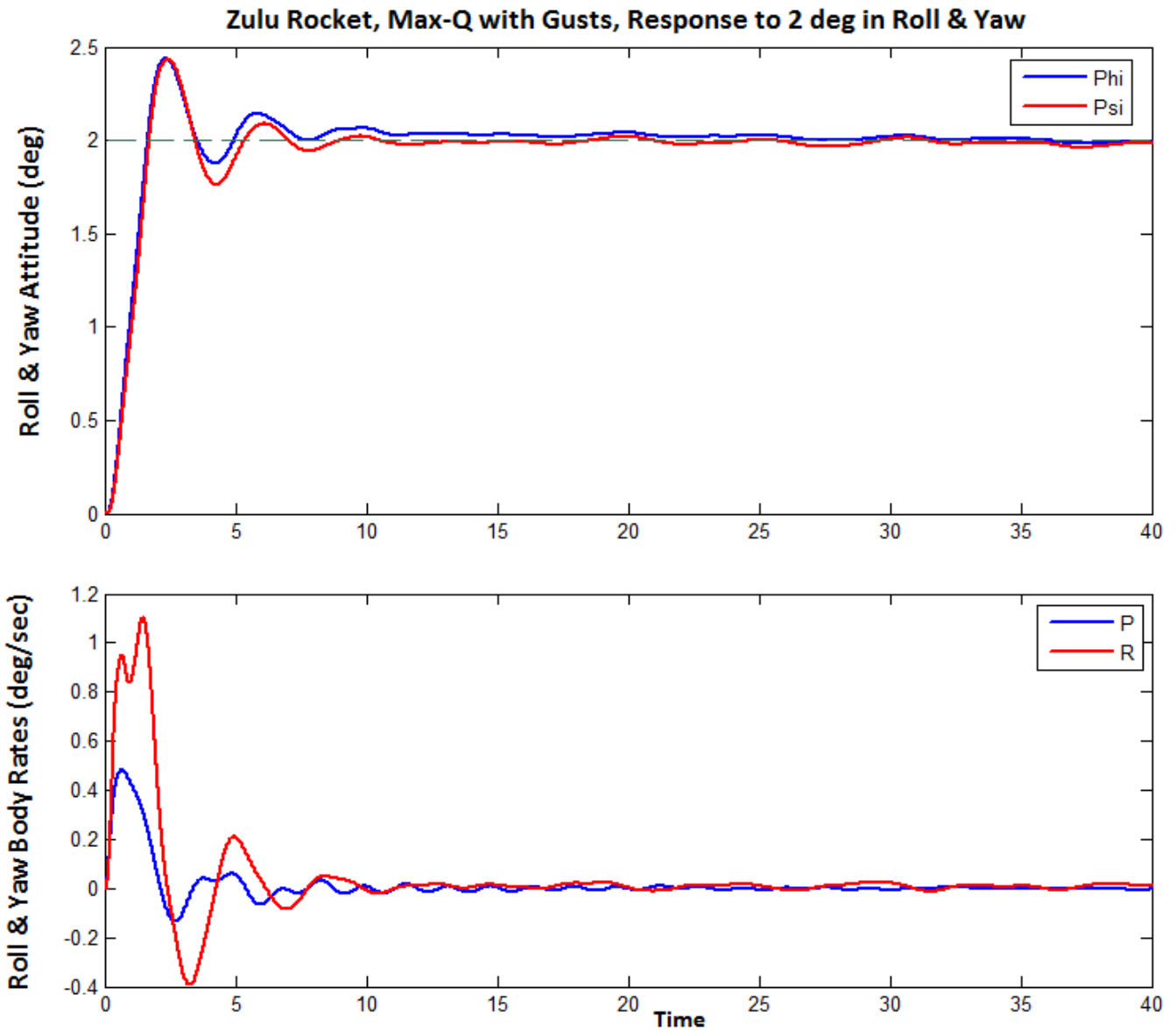


Figure 4.12 Roll and Yaw Attitude Response to Two Simultaneously Applied 2° Roll and Yaw Commands

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 “*vehl_latr_sflx.m*”.

```

% Frequency Response Analysis file Run.m
d2r= pi/180; r2d=1/d2r;
LQR_des;
Mass=9580; Thr=363305;
Sref= 47.3; Qbar= 228; Cza=-0.34; Cyb=-0.072;
[Avr, Bvr, Cvr, Dvr]= vehl_latr_rb;
[Avf, Bvf, Cvf, Dvf]= vehl_latr_sflx;
[Aac, Bac, Cac, Dac]= actuator;
load -ascii KMIX.MAT KMIX;
load -ascii Klat.MAT Klat;
%[A1,B1,C1,D1]= linmod('Latr_OpenRB');
[A1,B1,C1,D1]= linmod('Latr_OpenFlx');
sys= SS(A1, B1, C1, D1);
w=logspace(-2, 2.5, 22000);
figure(1); Nichols(sys, w)
figure(2); Bode(sys, w)
    
```

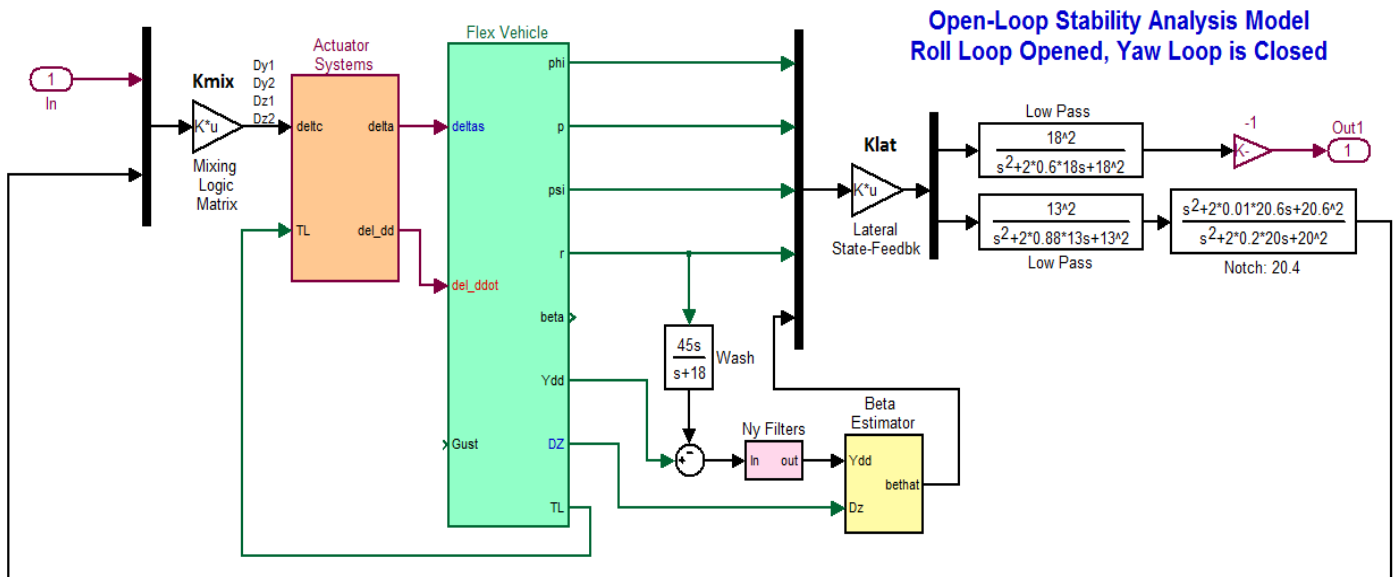


Figure 4.14 Simulink Model “*Latr_OpenFLX.mdl*” for Calculating Bode and Nichols Plots to Analyze the Lateral System Stability

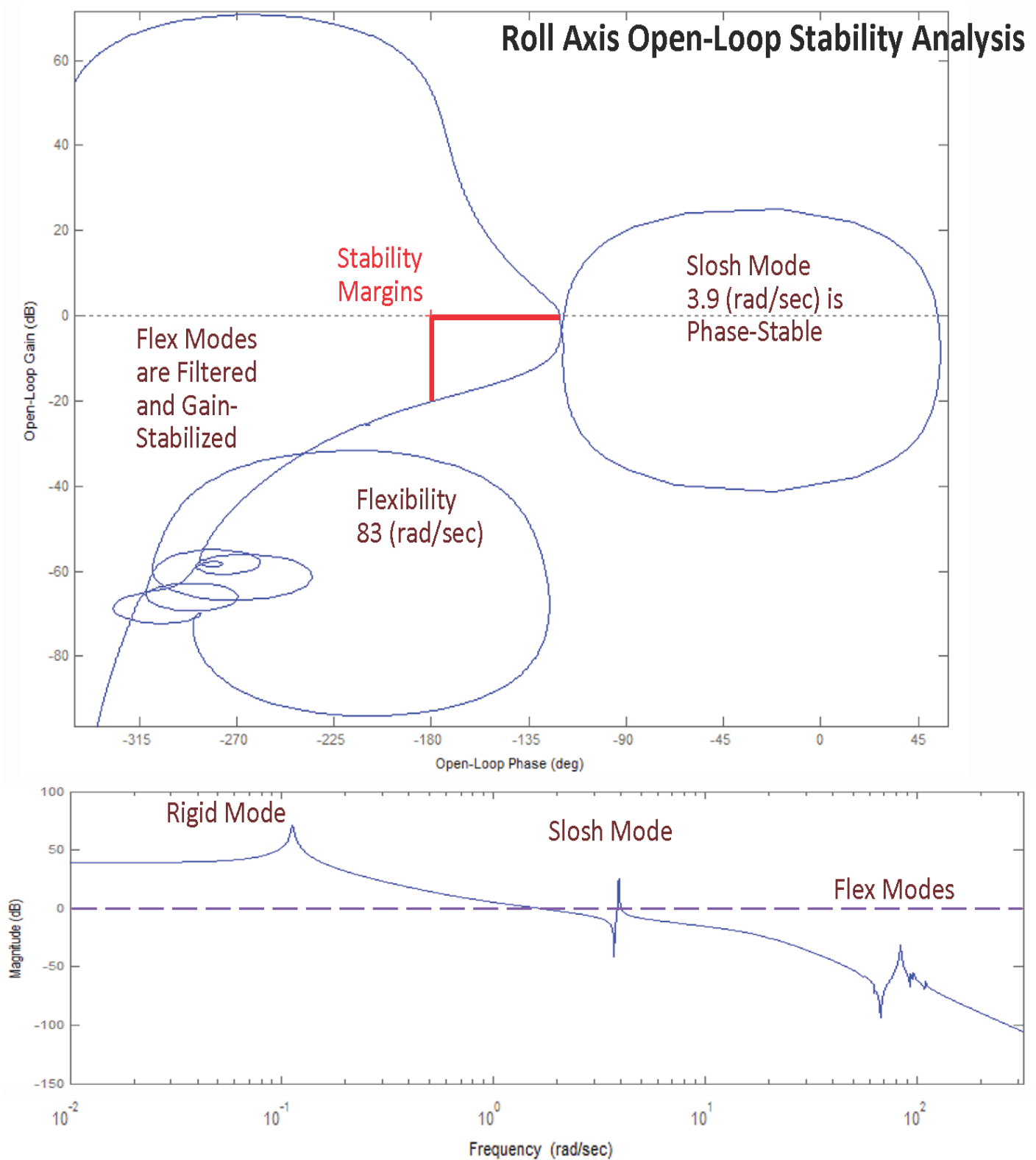


Figure 4.16 Roll Axis Stability Analysis Using Bode and Nichols Plots