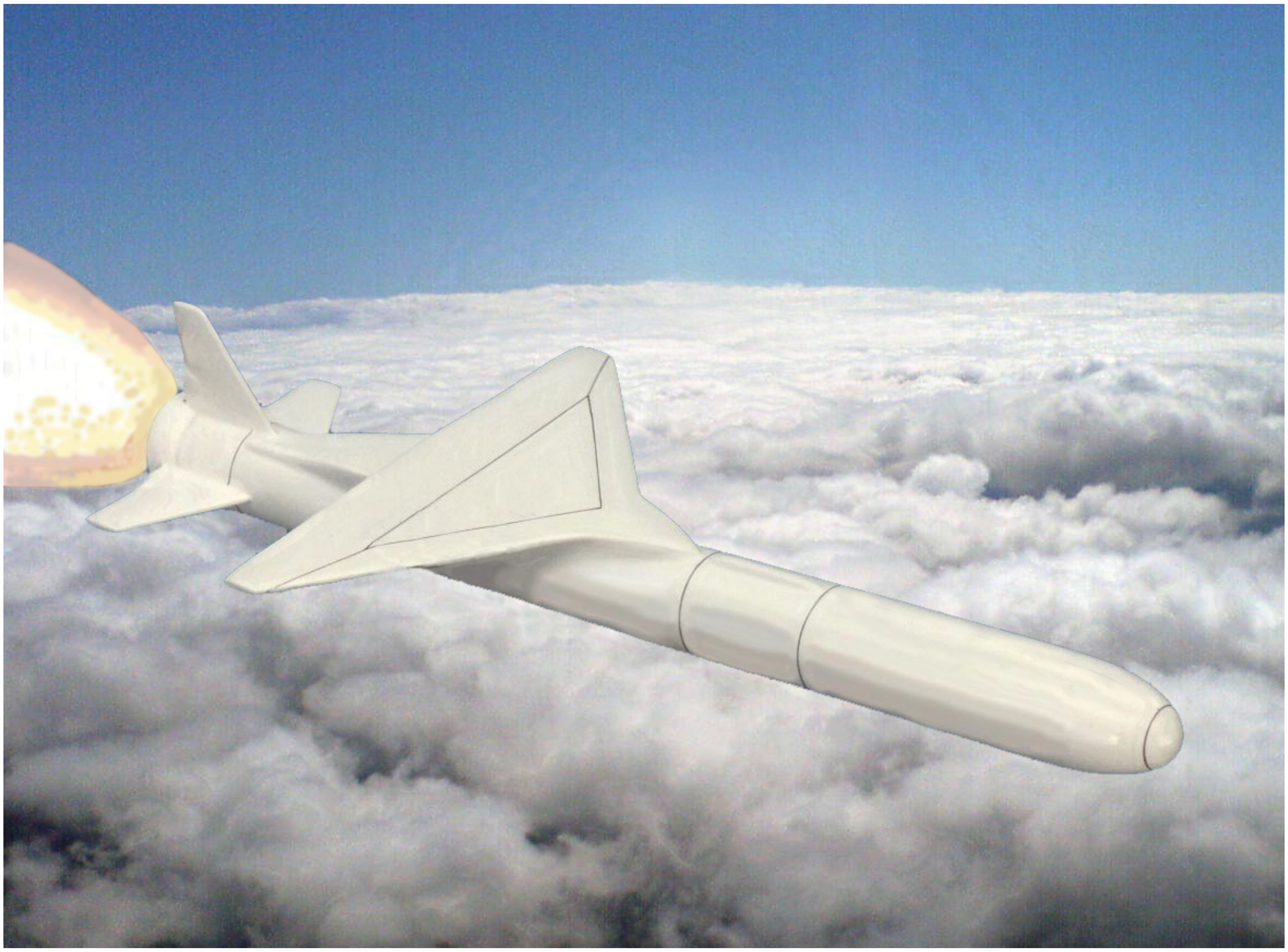
Missile with Wing and Aerosurfaces



In this example we analyze a cruising missile that has a small wing to provide lift and a fixed thrust engine that does not gimbal nor throttle. The missile is released horizontally from an aircraft and it climbs at high altitudes. It is controlled by three aero-surfaces located in the tail section consisting of: a vertical rudder mainly for yaw control and two horizontal rotating fins for pitch and roll control, see Figure 1.1. There are no control surfaces on the wing. Since the engine is not gimbaling, it is the wing in combination with the elevon aerosurfaces that provides the necessary lift for the vehicle to climb. The attitude, rate, and acceleration are measured by an Inertial Measurements Unit (IMU) located in the front section. The angles of attack and sideslip are not measured relative to the wind but the flight path angle γ and the heading direction ξ are inertially estimated from navigation. We will use Flixan to generate dynamic models at a critical flight condition, which is: Mach 2.5, 10 degrees of angle of attack, and high dynamic pressure of 1220 psf. We will design control laws for the pitch and lateral dynamics, and analyze stability and performance using Matlab.

1. Analysis Overview

In this Section we will use the Flixan to generate the missile dynamic systems. The systems will be separated into longitudinal and lateral subsystems that will be analyzed separately. We will examine classical control laws for stability and develop simulations to analyze the missile response to commands and to wind gusts.

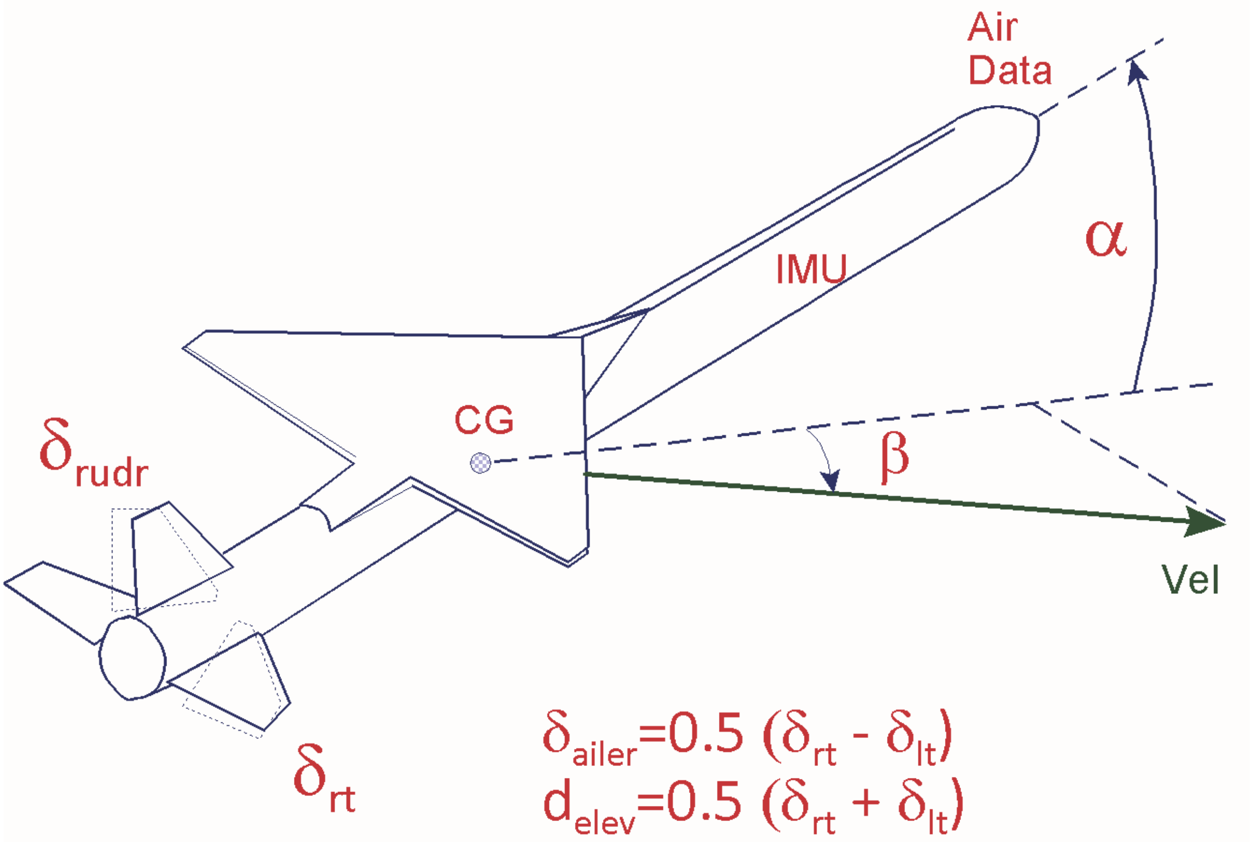

Air Data
α
IMU
CG
β
δ_{rudr}
Vel
δ_{rt}
$\delta_{ailer}=0.5\ (\delta_{rt} - \delta_{lt})$
$d_{elev}=0.5\ (\delta_{rt} + \delta_{lt})$

Figure 1.1 Missile Configuration showing the Aerosurfaces, Wing, CG, and Sensor Locations.

In section 2 we will use the Linear Quadratic Regulator (LQR) method to design and analyze pitch and lateral control laws. In section 3 we will demonstrate how to use Flixan to create parameter uncertainty models and Matlab to analyze the flight control system robustness and sensitivity to wind gusts by using μ-analysis methods.

1.1 Creating the Vehicle Model

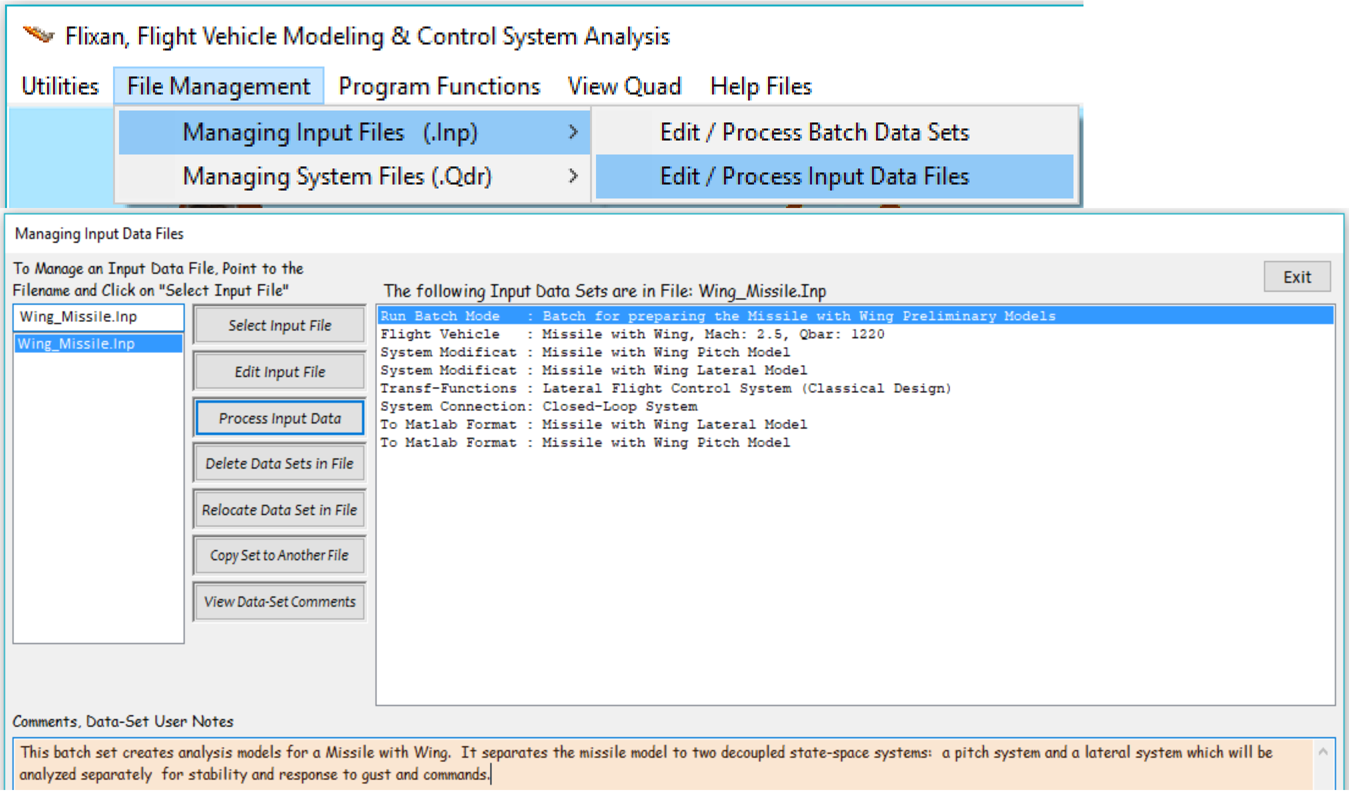
Figure 1.1 shows the missile with the wing and the three tail aerosurfaces consisting of a vertical rudder for yaw control, an elevon for pitch control produced by equally rotating the left and right fins in the same direction, and an aileron for roll control produced by rotating the left and right fins differentially. The missile is released horizontally from an aircraft, climbs to orbital altitude and tracks a pre-calculated flight path and heading directions mainly along the direction that it is released. The thrust is not used for control but it produces an acceleration which is captured in the vehicle data and model. The purpose of the flight control system is to stabilize the vehicle and to track a predesigned trajectory path in both: longitudinal and lateral directions.

The files used in this section are in folder "*C:\Flixan\Examples\Missile with Wing\(a) Classical Analysis*" and it includes two subfolders "*Pitch*" and "*Lateral*" for the corresponding Matlab analysis. The vehicle dataset is in the input file "*Wing_Missile.Inp*" and its title is: "*Missile with Wing, Mach: 2.5, Qbar: 1220*". The flags line (located below the title and comments) indicates that the output rates are body rates, and attitudes are defined to be Euler angles, because the commands are referenced in Euler angles. The flag label "**NoWind Alpha**" is also included because the incidence angles α and β, which are used to synthesize the flight-path and heading directions must not see the effects of a wind-gust directly because γ and ξ are estimated from navigation and they do not represent motion relative to the moving air mass. It means that the α and β calculations (states: 7, 8) will not include the wind velocity components w_{gust} and v_{gust} but only the vehicle velocities w and v. A wind-gust, however, will produce forces and moments on the vehicle and it will affect its motion, but the gust itself is not seen directly in the output as it would be if it was an air-data probe, only its effect on the vehicle are observable.

The gust disturbance is applied perpendicular to the body x-axis, and at 45° in azimuth. It means that the flow is along the -z and the -y directions exciting both: pitch and lateral axes. The aero surfaces tail-wag-dog option is turned off "**No TWD**" since at this point we are not including flexibility and detailed actuators. The aero-derivatives which are functions of the deflection rate are zero. There are two accelerometers defined in the input data: a normal and a lateral accelerometer located at the IMU section. Separate gyros are not defined because the vehicle model is rigid and the rigid attitude and rotational rates are always included in the outputs. There is also an air-data (vane) sensor included which is located in front of the vehicle and it measures α and β relative to the wind. The vehicle modeling program processes this dataset and creates a state-space model of the missile in file "*Wing_Missile.Qdr*". The system title is the same "*Missile with Wing, Mach: 2.5, Qbar: 1220*" and it includes both pitch and lateral dynamics. There are two system modification datasets below the vehicle data which separate the above coupled model in two uncoupled pitch and lateral subsystems. The first set generates the subsystem "*Missile with Wing Pitch Model*" which consists of only the longitudinal states, inputs, and outputs. The second set generates the subsystem "*Missile with Wing Lateral Model*" which consists of only the lateral states, inputs, and outputs. The pitch and lateral subsystems are also saved in systems file "*Wing_Missile.Qdr*", and from there they are converted to Matlab functions "vehi_pitch.m" and "vehi_later.m".

The next section describes how to process the input file interactively using Flixan. It is used in the initial phase of vehicle modeling and it takes longer to complete. When the modeling has been debugged and checked-out, there is no need to repeat this lengthy process for a second time when you simply want to modify a few parameters but you can rerun it in batch mode which is faster because it is not interactive. To re-process the data in batch mode you must create a batch dataset which is usually placed on the top of the input file. In this example we have already created a batch set "*Batch for preparing the Missile with Wing Models*". To process the batch set, from the Flixan main menu bar, go to "*File Management*", "*Manage Input Files*", and then "Edit/ *Process Input Data*

Files". The input file manager dialog comes up and from the left menu select the input filename "*Wing-Missile.Inp*" and click on *"Select Input File"*. From the menu on the right side select the top title which is the batch set and click on "*Process Input Data*" to process the batch. When the program completes the data processing, click on "Exit".

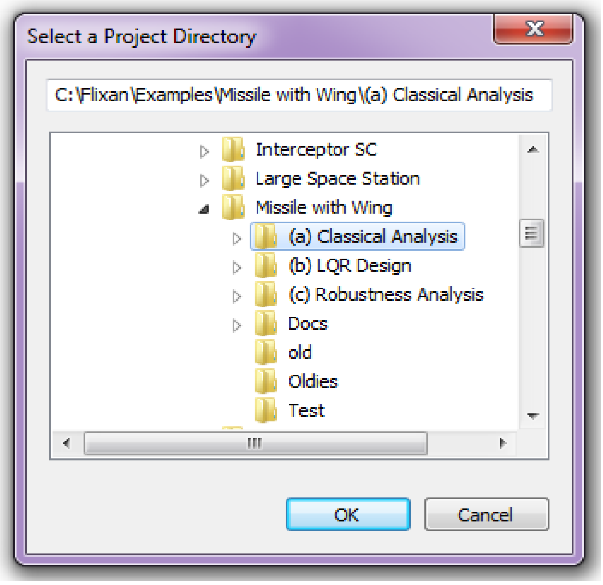

Flixan, Flight Vehicle Modeling & Control System Analysis
Utilities
File Management
Program Functions
View Quad
Help Files
Managing Input Files (.Inp)
Managing System Files (.Qdr)
Edit / Process Batch Data Sets
Edit / Process Input Data Files
Managing Input Data Files
To Manage an Input Data File, Point to the Filename and Click on "Select Input File"
Wing_Missile.Inp
Wing_Missile.Inp
Select Input File
Edit Input File
Process Input Data
Delete Data Sets in File
Relocate Data Set in File
Copy Set to Another File
View Data-Set Comments
Exit
The following Input Data Sets are in File: Wing_Missile.Inp
Comments, Data-Set User Notes

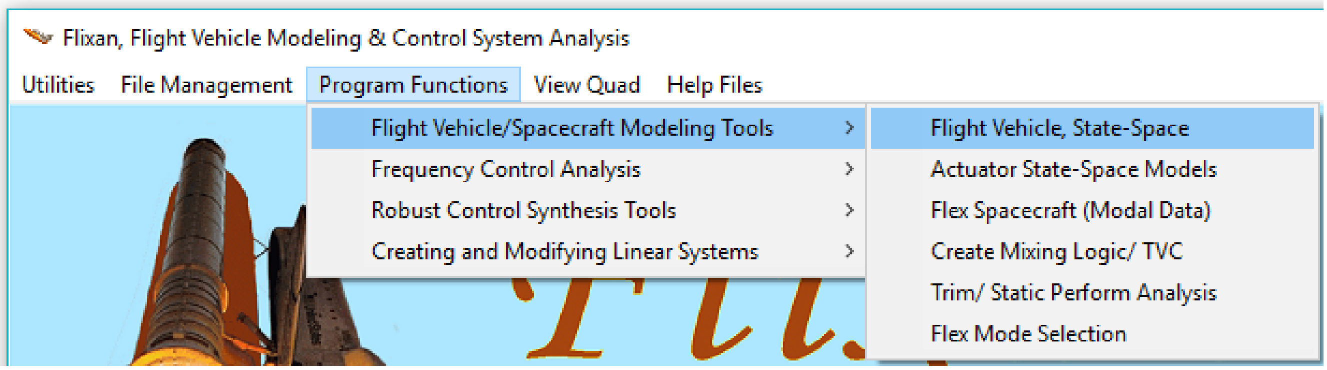
```
Run Batch Mode    : Batch for preparing the Missile with Wing Preliminary Models
Flight Vehicle    : Missile with Wing, Mach: 2.5, Qbar: 1220
System Modificat  : Missile with Wing Pitch Model
System Modificat  : Missile with Wing Lateral Model
Transf-Functions  : Lateral Flight Control System (Classical Design)
System Connection: Closed-Loop System
To Matlab Format  : Missile with Wing Lateral Model
To Matlab Format  : Missile with Wing Pitch Model
```

This batch set creates analysis models for a Missile with Wing. It separates the missile model to two decoupled state-space systems: a pitch system and a lateral system which will be analyzed separately for stability and response to gust and commands.

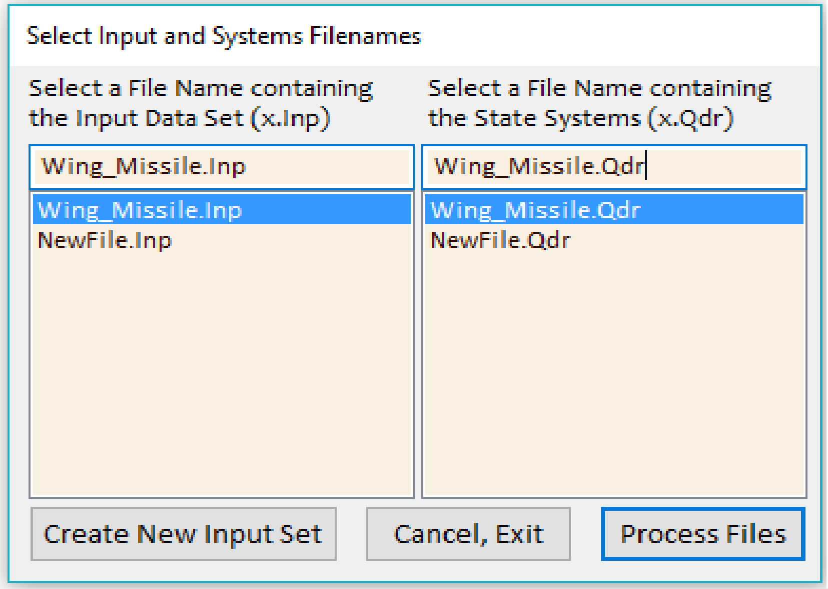
1.2 Running Flixan Interactively

To process the input file interactively, you must first select the directory "*C:\Flixan\Examples\Missile with Wing\(a) Classical Analysis*". Then go to "*Program Functions*", "*Flight Vehicle/ Spacecraft Modeling Tools*", and "*Flight Vehicle State-Space*", as shown below.



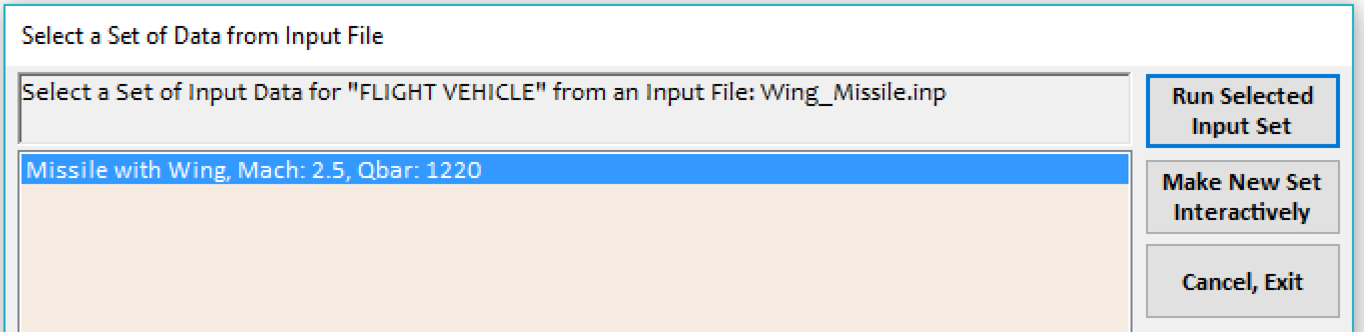

Flixan, Flight Vehicle Modeling & Control System Analysis
Utilities
File Management
Program Functions
View Quad
Help Files
Flight Vehicle/Spacecraft Modeling Tools
Frequency Control Analysis
Robust Control Synthesis Tools
Creating and Modifying Linear Systems
Flight Vehicle, State-Space
Actuator State-Space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

From the filenames selection menu select the input data file "*Wing_Missile.Inp*" and the systems file "*Wing_Missile.Qdr*", and click on "*Process Files*". The program searches the input data file and finds one already created vehicle dataset "*Missile with Wing, Mach: 2.5, Qbar: 1220*" that contains the missile data. Click on "*Run Selected Input Set*" to process the data.


Select Input and Systems Filenames

Select a File Name containing the Input Data Set (x.Inp): Wing_Missile.Inp

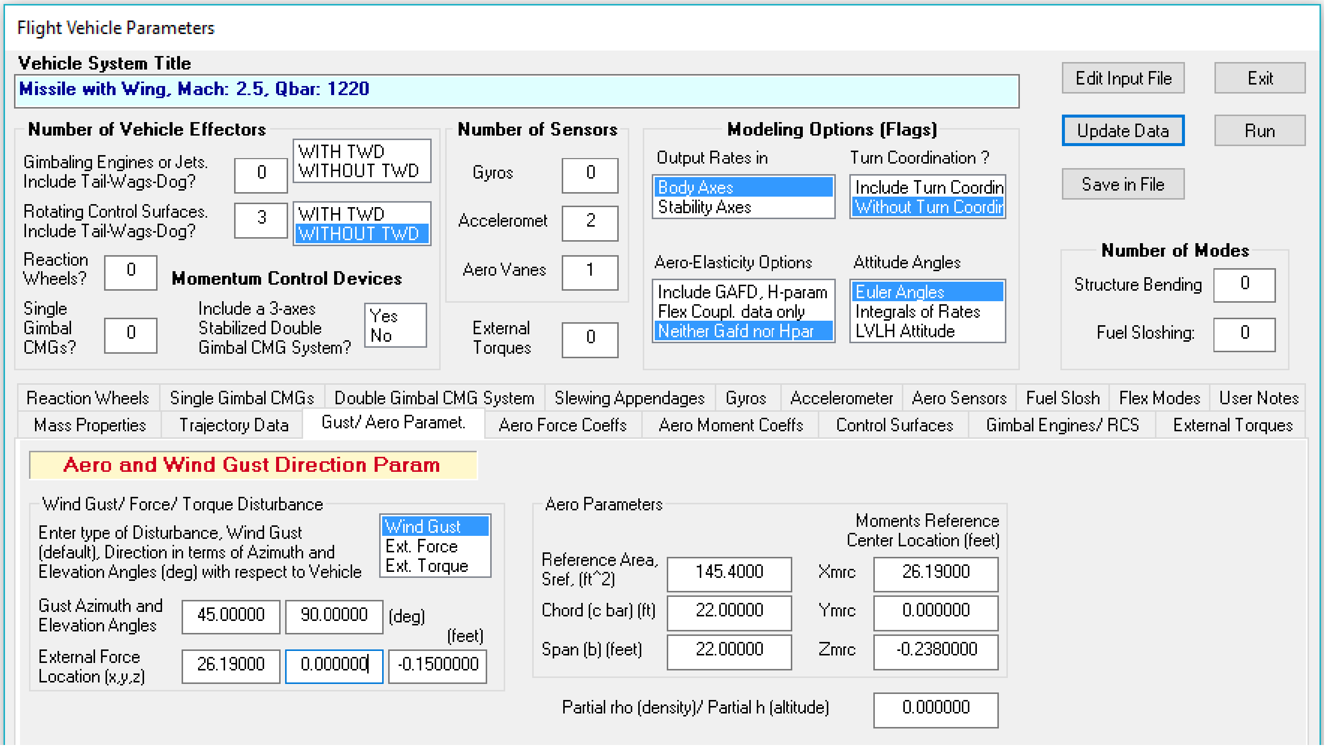
Select a File Name containing the State Systems (x.Qdr): Wing_Missile.Qdr


Select a Set of Data from Input File

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

Missile with Wing, Mach: 2.5, Qbar: 1220

The program opens a dialog that displays the missile data that will be used to create the missile state-space model. On the top of the display we see the vehicle title and some vehicle parameters. For example, there are 3 aerosurfaces without TWD, etc. Further down we see various tabs. One of the tabs is "Control Surfaces". Click on the "Control Surfaces" and the dialog will display the aero-data for each control surface. While still in the Control Surface tab, click on the "Next Surface" button and it will display the aero-coefficients for the next aerosurface which is the aileron, and then rudder. If you click on another tab, for example, the aero moment coefficients, the dialog will display the aero-moments. Click on the "Run" button located on the upper right of the display and the program will create the missile model in file "*Wing_Missile.Qdr*".


Flight Vehicle Parameters

Vehicle System Title: Missile with Wing, Mach: 2.5, Qbar: 1220

Number of Vehicle Effectors

Gimbaling Engines or Jets. Include Tail-Wags-Dog?: 0 (WITH TWD / WITHOUT TWD)

Rotating Control Surfaces. Include Tail-Wags-Dog?: 3 (WITH TWD / WITHOUT TWD, selected: WITHOUT TWD)

Reaction Wheels?: 0

Momentum Control Devices

Single Gimbal CMGs?: 0

Include a 3-axes Stabilized Double Gimbal CMG System?: Yes / No

Number of Sensors

Gyros: 0

Accelerometer: 2

Aero Vanes: 1

External Torques: 0

Modeling Options (Flags)

Output Rates in: Body Axes (selected) / Stability Axes

Turn Coordination ?: Include Turn Coordin / Without Turn Coordir (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

Aero and Wind Gust Direction Param

Wind Gust/ Force/ Torque Disturbance

Enter type of Disturbance, Wind Gust (default), Direction in terms of Azimuth and Elevation Angles (deg) with respect to Vehicle: Wind Gust (selected) / Ext. Force / Ext. Torque

Gust Azimuth and Elevation Angles: 45.00000, 90.00000 (deg)

External Force Location (x,y,z) (feet): 26.19000, 0.000000, -0.1500000

Aero Parameters

Reference Area, Sref, (ft^2): 145.4000

Chord (c bar) (ft): 22.00000

Span (b) (feet): 22.00000

Moments Reference Center Location (feet)

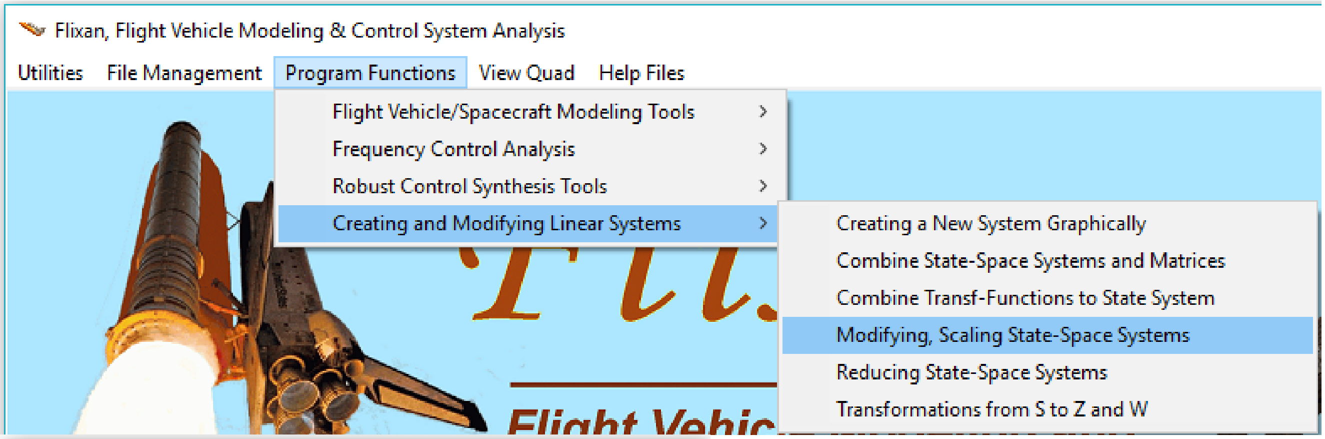
Xmrc: 26.19000

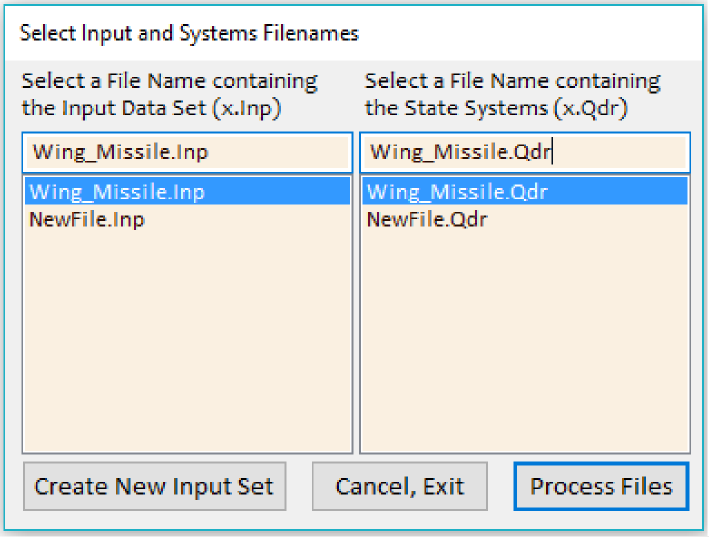
Ymrc: 0.000000

Zmrc: -0.2380000

Partial rho (density)/ Partial h (altitude): 0.000000

The next step is to separate the pitch and lateral subsystems from the coupled vehicle system “*Missile with Wing, Mach: 2.5, Qbar: 1220*”. Start Flixan and use the same directory. From the main menu, select: "*Program Functions*", then go to "*Create and Modify Linear Systems*", and then "*Modifyng, Scaling State-Space Systems*". From the filenames selection menu select the input data file "*Wing_Missile.Inp*" and the systems file "*Wing_Missile.Qdr*", and click on "*Process Files*". The program finds two system modification datasets in the input file. The first one for creating the pitch subsystem and the second one is for the lateral subsystem. Select the first one “*Missile with Wing Pitch Model*” and click on "*Run Selected Input Set*" to process the system modification data and to extract the pitch system. Repeat the process and select the second title to create the lateral system "*Missile with Wing Lateral Model*". The two reduced models are saved in the systems file "*Wing_Missile.Qdr*".


Flixan, Flight Vehicle Modeling & Control System Analysis
Utilities
File Management
Program Functions
View Quad
Help Files
Flight Vehicle/Spacecraft Modeling Tools
Frequency Control Analysis
Robust Control Synthesis Tools
Creating and Modifying Linear Systems
Creating a New System Graphically
Combine State-Space Systems and Matrices
Combine Transf-Functions to State System
Modifying, Scaling State-Space Systems
Reducing State-Space Systems
Transformations from S to Z and W

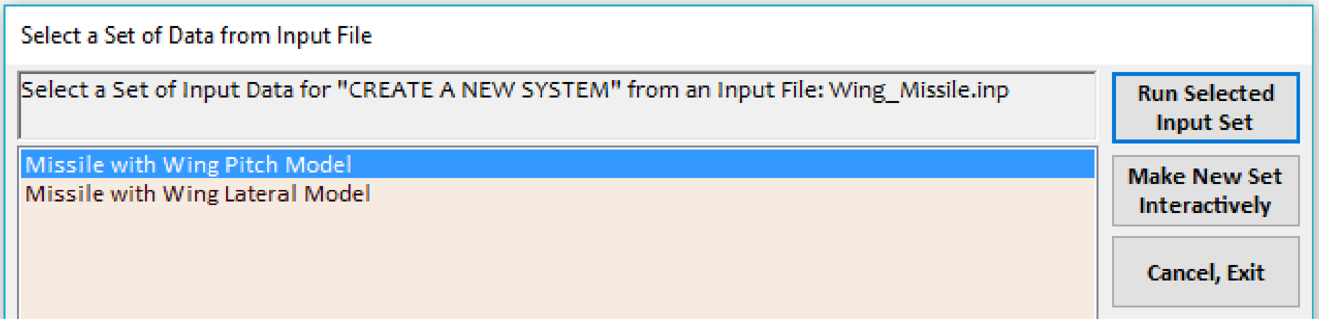

Select Input and Systems Filenames

Select a File Name containing the Input Data Set (x.Inp)

Wing_Missile.Inp

Select a File Name containing the State Systems (x.Qdr)

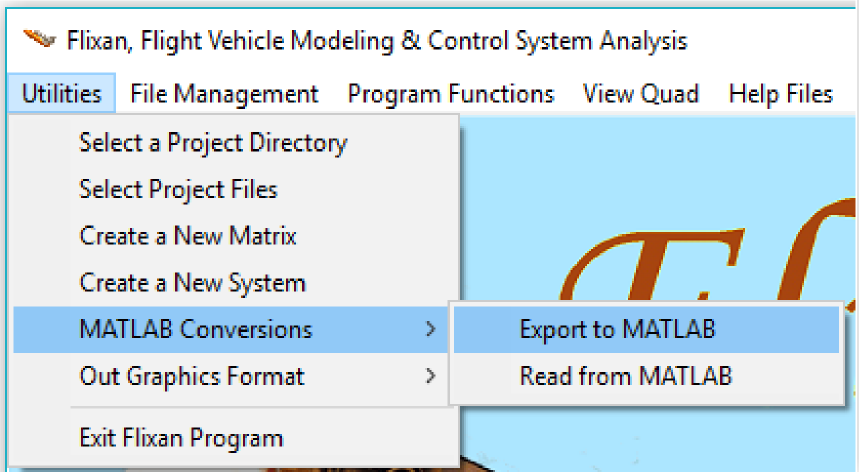
Wing_Missile.Qdr

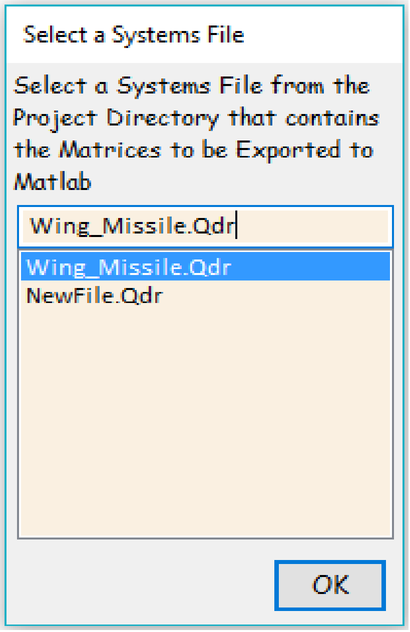

Select a Set of Data from Input File

Select a Set of Input Data for "CREATE A NEW SYSTEM" from an Input File: Wing_Missile.inp

Missile with Wing Pitch Model
Missile with Wing Lateral Model
Run Selected Input Set
Make New Set Interactively
Cancel, Exit

The next step is to convert the pitch and lateral missile systems from file "*Wing_Missile.Qdr*" to Matlab-function format so that they can be loaded into Matlab. The conversion utility will convert to Matlab function one system at a time starting with the pitch system. From the top Flixan menu bar select "*Utilities*", "*Matlab Conversions*", and "*Export to Matlab*". The program will display a menu from where you can select the filename containing the systems to be converted. Select the file "*Wing_Missile.Qdr*" and click "OK".

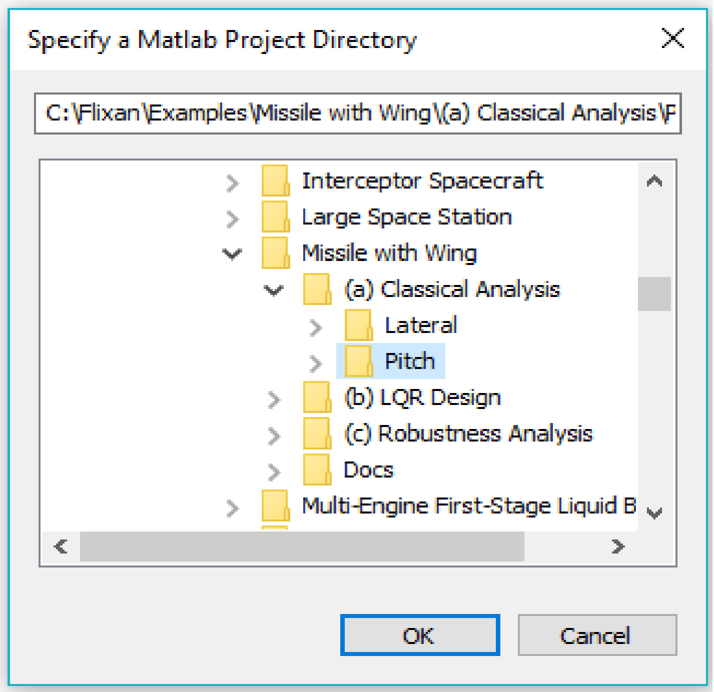



Select a Systems File

Select a Systems File from the Project Directory that contains the Matrices to be Exported to Matlab

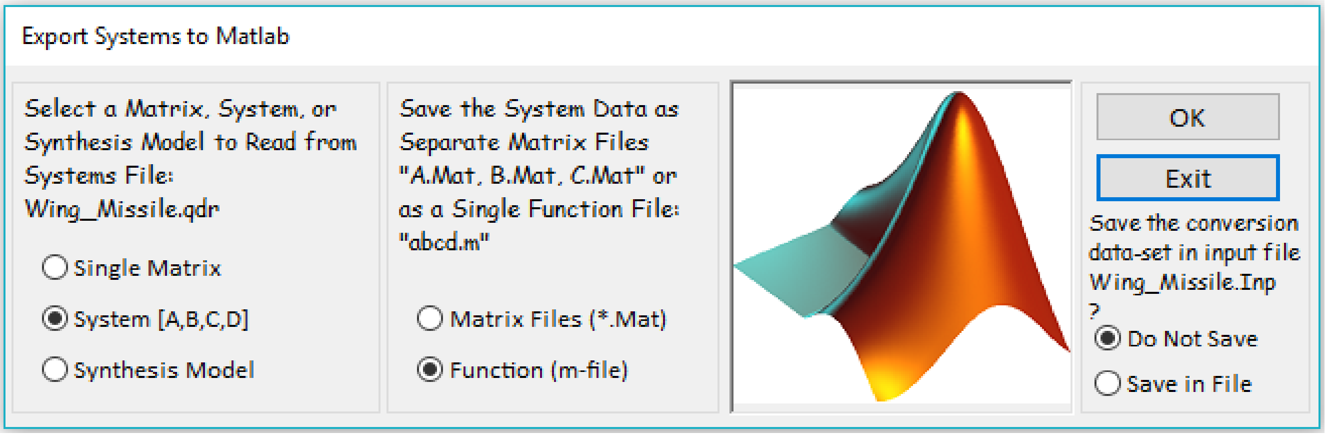
Wing_Missile.Qdr

The next dialog is for selecting the directory to save the pitch Matlab function system. That is, "*Flixan\Examples\Missile with Wing\ (a) Classical Analysis\Pitch*", and click "OK".


Specify a Matlab Project Directory

C:\Flixan\Examples\Missile with Wing\(a) Classical Analysis\P

The next Matlab exporting dialog has menus and it is used to specify what needs to be converted into Matlab format. It is a system in this case to be converted to a Matlab function m-file, and you don’t need to save the operation as a dataset because it is already there. Click "OK".



Export Systems to Matlab

Select a Matrix, System, or Synthesis Model to Read from Systems File: Wing_Missile.qdr

○ Single Matrix

● System [A,B,C,D]

○ Synthesis Model

Save the System Data as Separate Matrix Files "A.Mat, B.Mat, C.Mat" or as a Single Function File: "abcd.m"

○ Matrix Files (*.Mat)

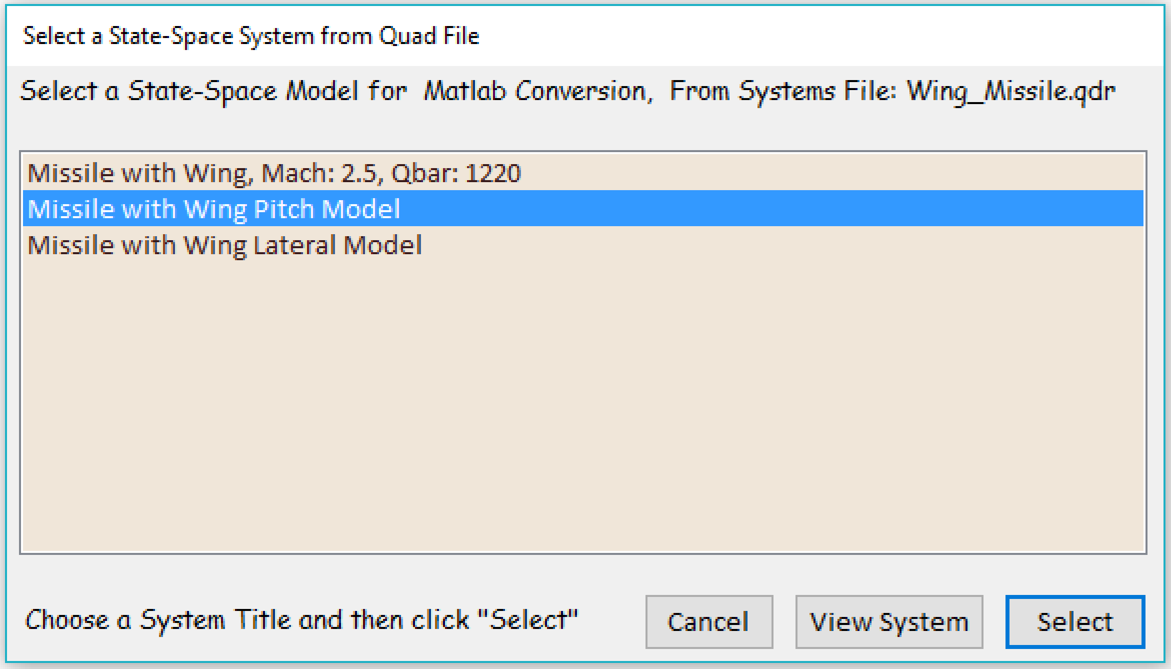
● Function (m-file)

Save the conversion data-set in input file Wing_Missile.Inp ?

● Do Not Save

○ Save in File

The next menu shows the titles of the systems which are already saved in file "*Wing_Missile.Qdr*". Since you are converting the pitch system select its title "*Missile with Wing Pitch Model*" and click "OK".



Select a State-Space System from Quad File

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

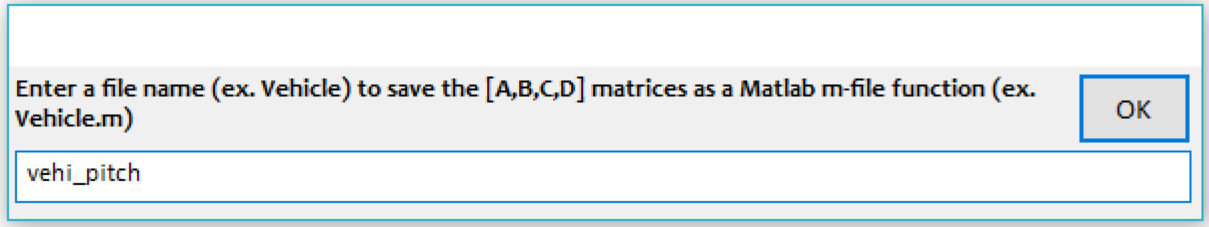
Missile with Wing, Mach: 2.5, Qbar: 1220

Missile with Wing Pitch Model

Missile with Wing Lateral Model

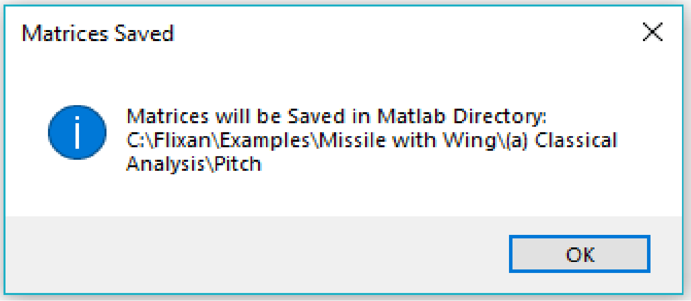
Choose a System Title and then click "Select"

The program also needs to know the name of the system m-file. It will be saved as an m-file that can be loaded into Matlab. Enter the filename (without the .m) and the pitch system will be saved as "vehi_pitch.m".



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

vehi_pitch


Matrices Saved

Matrices will be Saved in Matlab Directory: C:\Flixan\Examples\Missile with Wing\(a) Classical Analysis\Pitch

The same process is repeated for the lateral system "*Missile with Wing Lateral Model*" which is saved in file "vehi_later.m" in folder "*Examples**Missile with Wing*\ *(a) Classical Analysis**Lateral*".

1.3.1 Pitch Flight Control System Analysis

We begin with an already prepared pitch flight control system. In Section 2 we will demonstrate how to design gains using the Linear Quadratic Regulator (LQR) method. The control law presented in this Section consists of feedback from the pitch rate (q), the angle of attack (α), and the elevon deflection (δ_{elev}). Tracking of the flight-path angle (γ) is accomplished by commanding α. The angle of attack, however, is not directly measurable and alpha-feedback will be implemented using two approaches that both stabilize the system:

- Alpha-feedback from the air data probe (which has some errors due to vehicle rotation and its offset from the CG. This approach is implemented in Simulink model "*Pitch_Sim1.mdl*".
- Estimated alpha-feedback from an α-estimator that uses the normal accelerometer N_z and elevon deflection δ_{elev} signals. The second approach is implemented in Simulink model "*Pitch_Sim2.mdl*".

Figure 1.2 shows the state-feedback subsystem that consists of gain feedback from 4 pitch states: pitch rate (q), angle of attack (α), α-integral, and the elevon actuator position (δ_{elev}). The output is the elevon deflection command to the actuator. The state-feedback from the elevon actuator output provides some additional phase-lead for stabilization and improved performance. The angle of attack α is either measured from the air-data probe that is located at the nose of the missile, or it is estimated from the normal accelerometer which is located 20 feet in front of the CG. In either case the α-measurements are relative to the air-flow, including winds, and slightly corrupted with errors. The angle of attack estimator in Figure 1.3 uses N_z and the actuator deflection δ_{elev} outputs to estimate α based on the normal force Equation 1.1. The estimator requires knowledge of the vehicle mass, dynamic pressure, reference area and aero the derivative coefficients. The quality of the estimate obviously depends on the knowledge of these parameters.

$$M_V N_Z = \overline{Q} S_{ref} \left(C_{Z\alpha}\, \alpha + C_{Z\delta e}\, \delta e \right) \tag{1.1}$$

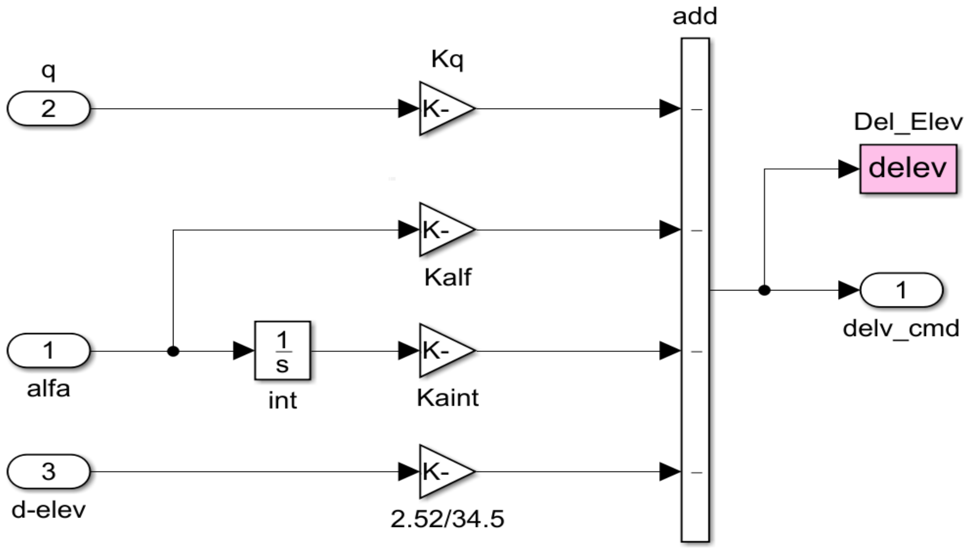


Figure 1.2 Pitch Axis State-Feedback Controller

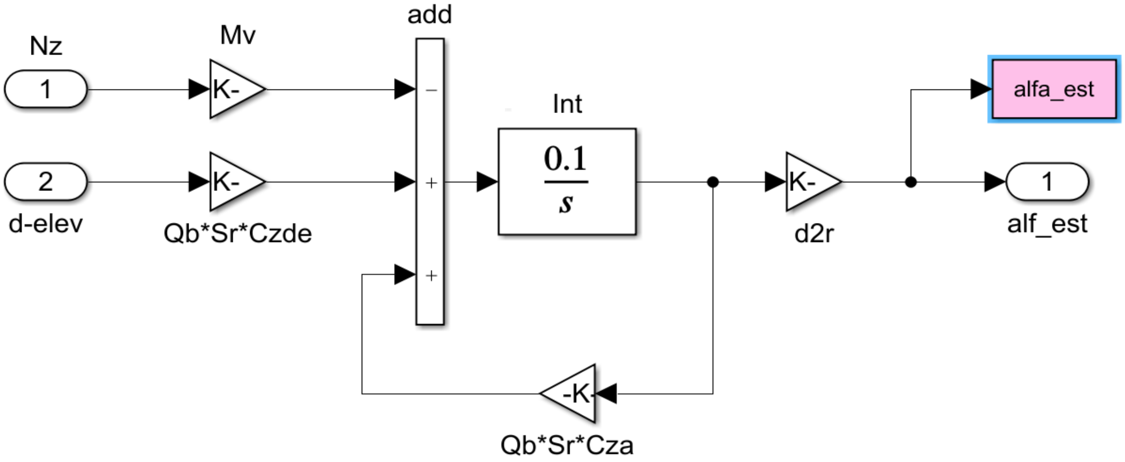


Figure 1.3 Angle of Attack Estimator uses Normal Acceleration and Elevon Actuator Measurements

1.3.2 Pitch Axis Simulations

The folder "\Flixan\Examples\Missile with Wing\\ *(a) Classical Analysis\Pitch*" includes two very similar simulation models which are stabilized by the same state-feedback. Their difference is in the α-measurement. In file "*Pitch_Sim1.mdl*", shown in Figure 1.4, the measurement comes from the air-data probe, and in file "*Pitch_Sim2.mdl*", shown in Figure 1.5, an estimator is used to estimate α from the normal accelerometer and the actuator outputs. There is an outer flight control loop that controls the flight path angle γ by adjusting α. The γ-error is filtered by a PI compensator and becomes α-command to the inner control loop. We are indirectly controlling, therefore, the flight path angle by commanding alpha. A second order actuator model of 5.5 Hz bandwidth is also included in the simulation which drives the elevon deflection input from the deflection command.

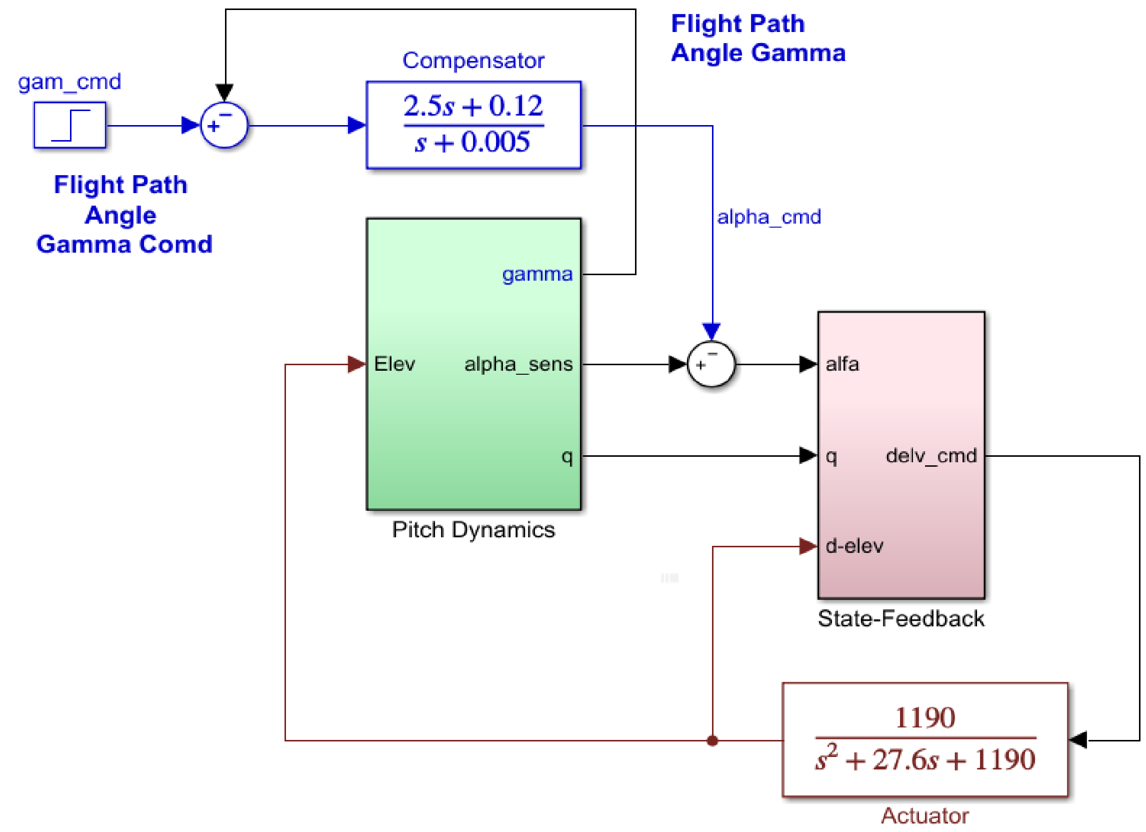


Figure 1.4 Pitch Simulation Model "Pitch_Sim1.mdl" using Alpha-Measurement from the Air-Data probe

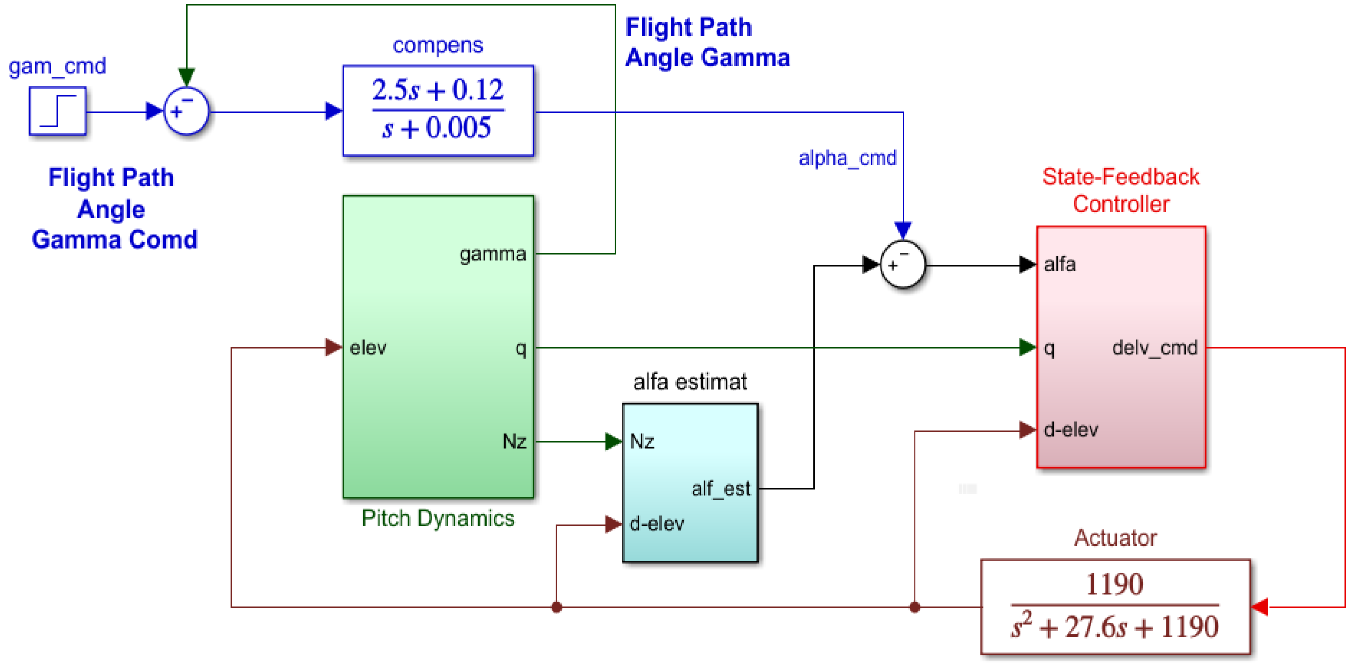


Figure 1.5 Pitch Simulation Model "Pitch_Sim2.mdl" using Estimated Alpha from the Estimator

Figure 1.6 shows the vehicle dynamic model inside the green block. It uses the Flixan generated system “*Missile with Wing Pitch Model*” from file "*vehi_pitch.mat*". It is loaded into Matlab during initialization from file "*init.m*". This system has two inputs: elevon actuator deflection in (radians) and wind gust velocity in (feet/sec). The wind-gust direction is defined in the vehicle input data. A low pass filter is included to smooth out the square gust pulse so that it resembles a real wind-gust. The vehicle subsystem outputs are: pitch attitude, pitch rate, angle of attack (α_{-}inertial) in (radians), normal acceleration in (feet/sec^2), altitude change, and α_w relative to wind which is measured by the air-data sensor located at the front. It is assumed that the flight path angle γ (rad) is not directly measured but it is estimated from navigation relative to the inertial frame and not relative to the moving air-mass. It is calculated from θ and α_i outputs: $\gamma=\theta-\alpha_I$, where α_i must not include the wind component. Gamma also has a significant delay since it takes some time to process the navigation data.

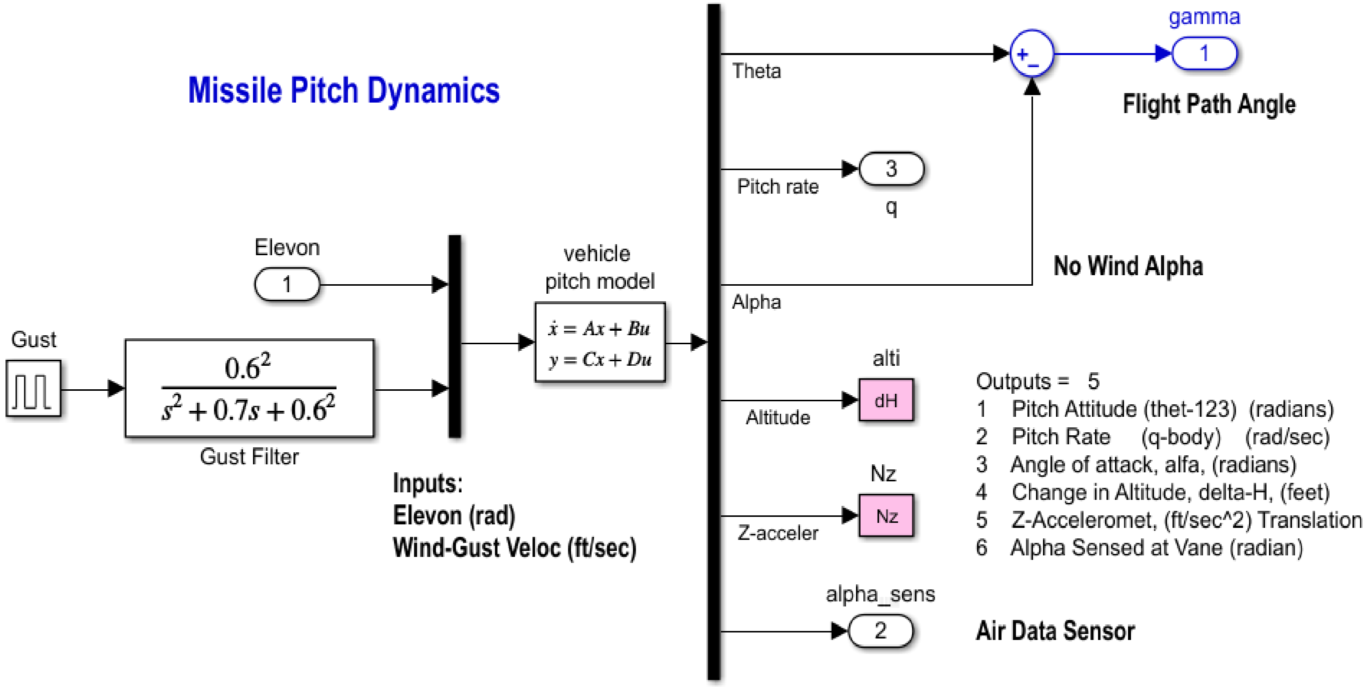


Figure 1.6 Longitudinal Axis Missile Dynamic Model

The simulation results from the two systems are similar so we will only show the transient responses from the second model "*Pitch_Sim2.mdl*" that uses the α-estimator. Figure 1.7 shows the system’s response to 1^o γ-command, and Fig. 1.8 shows its response to a 30 (feet/sec) wind-gust pulse of 2 sec duration. In the first case the elevon response is negative to generate positive alpha and negative acceleration as the vehicle is climbing at 1^o of γ. The alpha estimate is very good because the exact parameters were used in the alpha estimator. In the second case in Figure 1.8 the gust impulse is applied from below causing negative pitch attitude and rate. The vehicle responds with negative elevon deflection which temporarily makes it lose altitude and causes negative gamma. The wind gust makes the two alphas to be different in this case. The alpha relative to wind increases since the gust is coming from below.

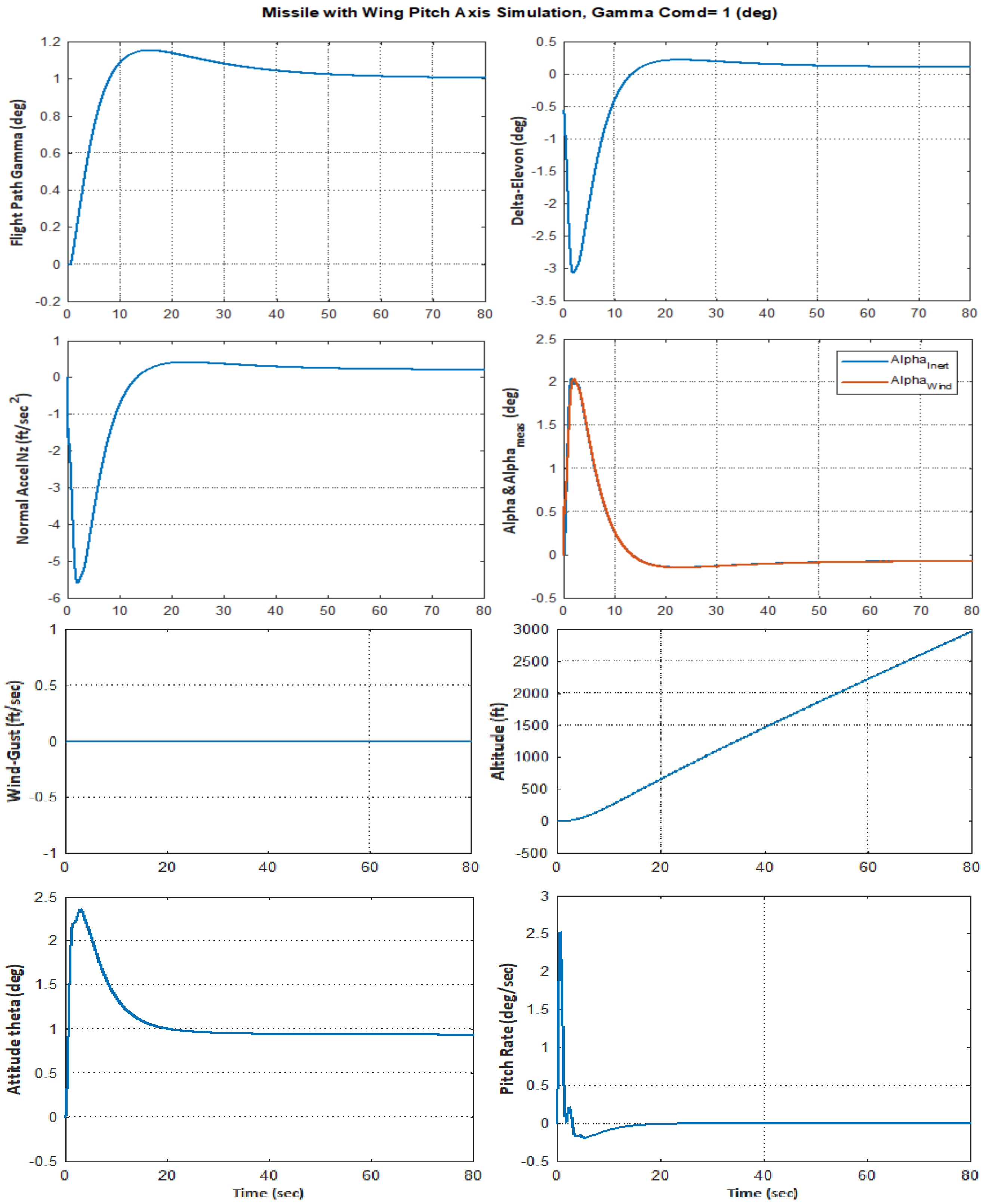


Figure 1.7 Pitch System Response to 1 (deg) of Gamma Command (No Gust)

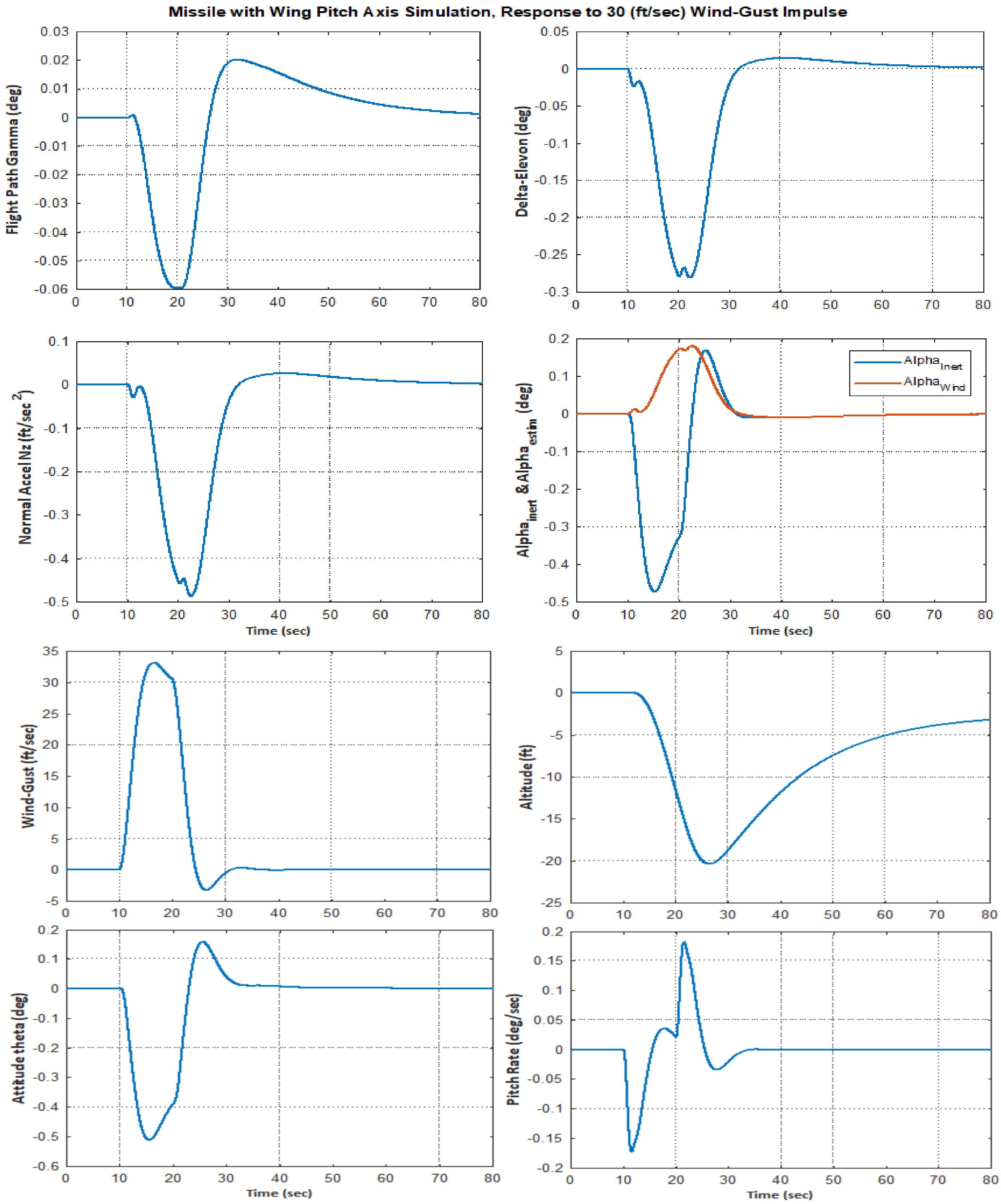


Figure 1.8 Response of the Pitch System to Wind-Gust Impulse (No Command)

1.3.3 Pitch Axis Stability Analysis

The Simulink model "*Open_Loop2.mdl*", shown in Figure 1.9, is used to compute the frequency response of the open-loop system and to calculate the pitch flight control system's phase and gain margins. The control loop is broken at the elevon actuator output, as shown. The direct α-feedback is replaced by including the alpha estimator in the control loop. A delay of 0.7 sec was included in the gamma output to account for computational delays in navigation. Figure 1.10 shows the Bode and Nichols plots which demonstrate system stability. There is a short-period resonance at 3.84 (rad/sec) because the missile is statically stable.

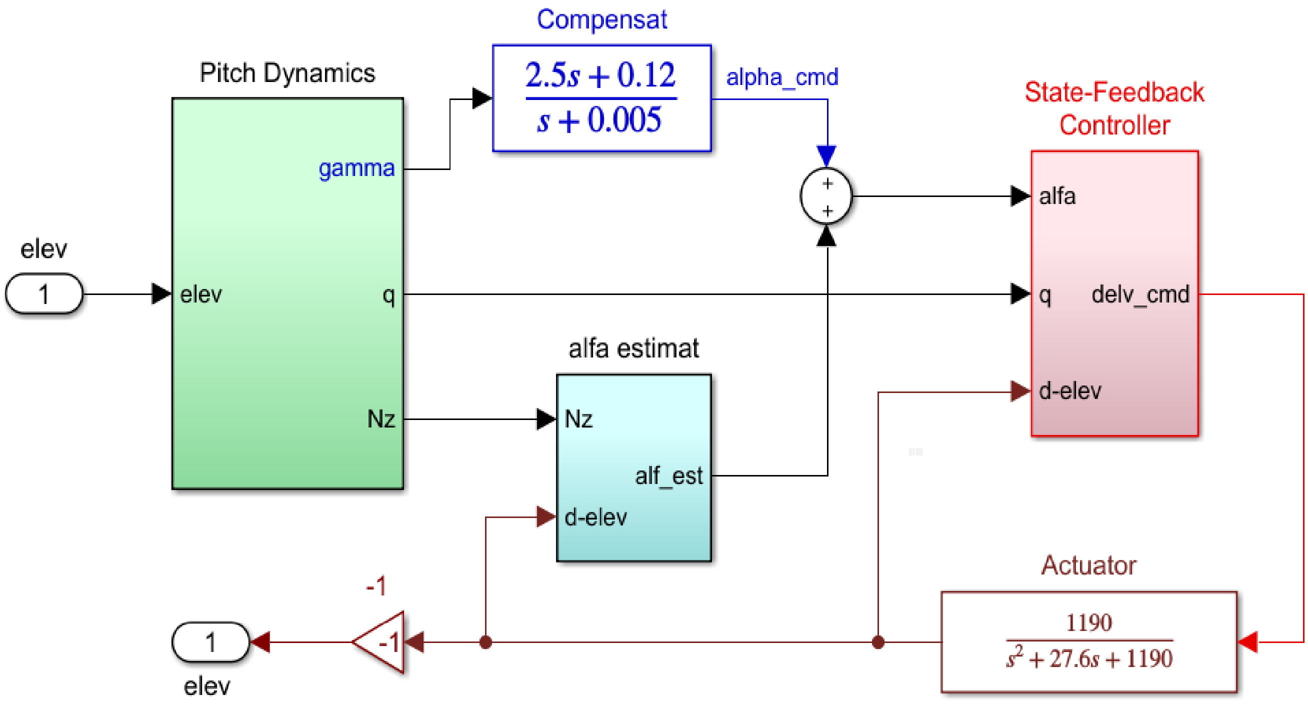


Figure 1.9 Simulink Model "*Open_Loop2.mdl*" Used for Stability Analysis

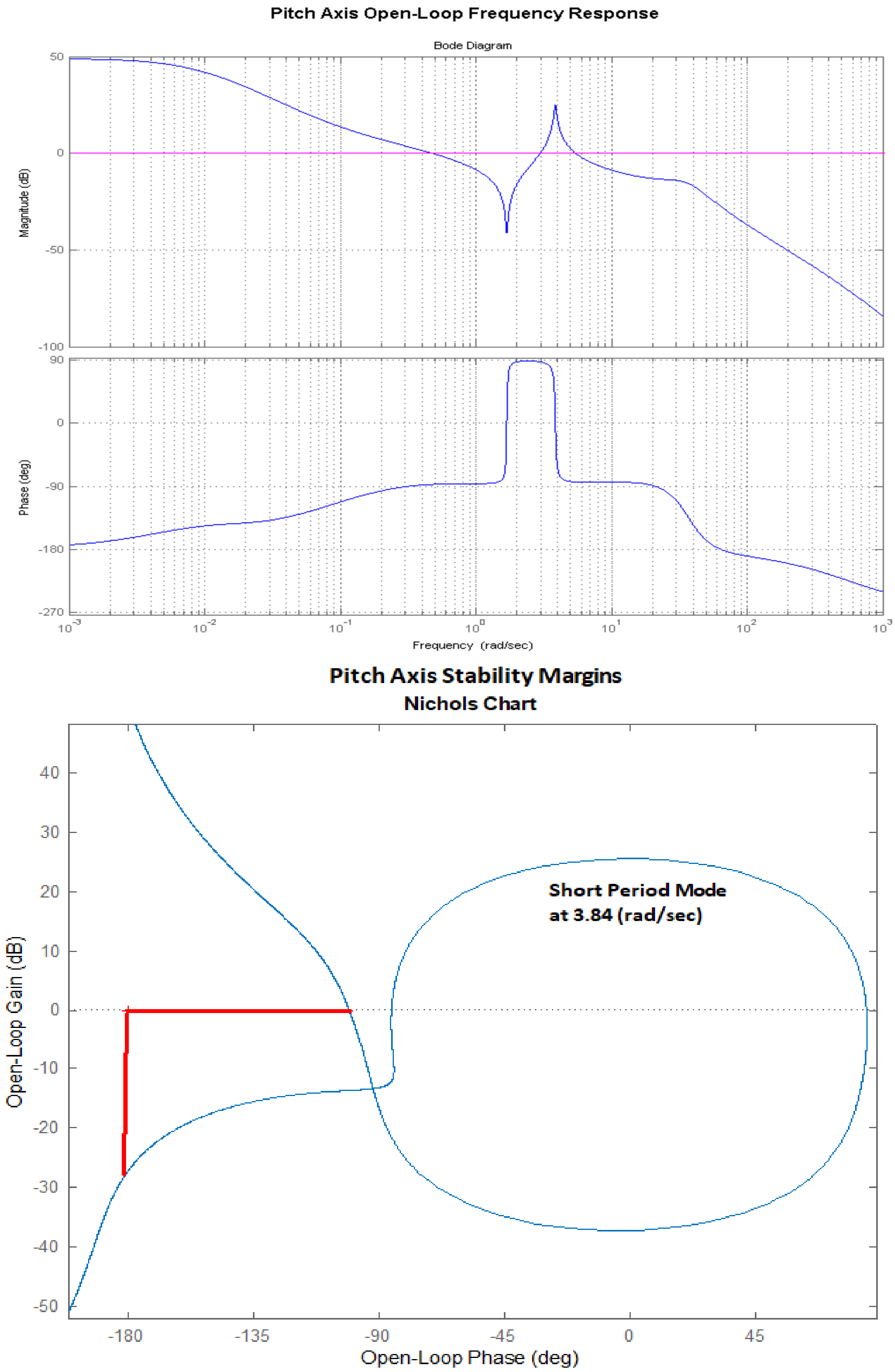


Figure 1.10 Pitch Axis Bode and Nichols Plots Showing the Phase and Gain Margins

1.4 Lateral Axes Analysis

In the roll and yaw axes we will first analyze a flight control design that was created using classical control methods. In section 2 we will design a lateral control system in Flixan using the LQR method. We will analyze and compare the two designs with simulations and frequency response stability analysis.

1.4.1 Lateral Flight Control System (Classical Approach)

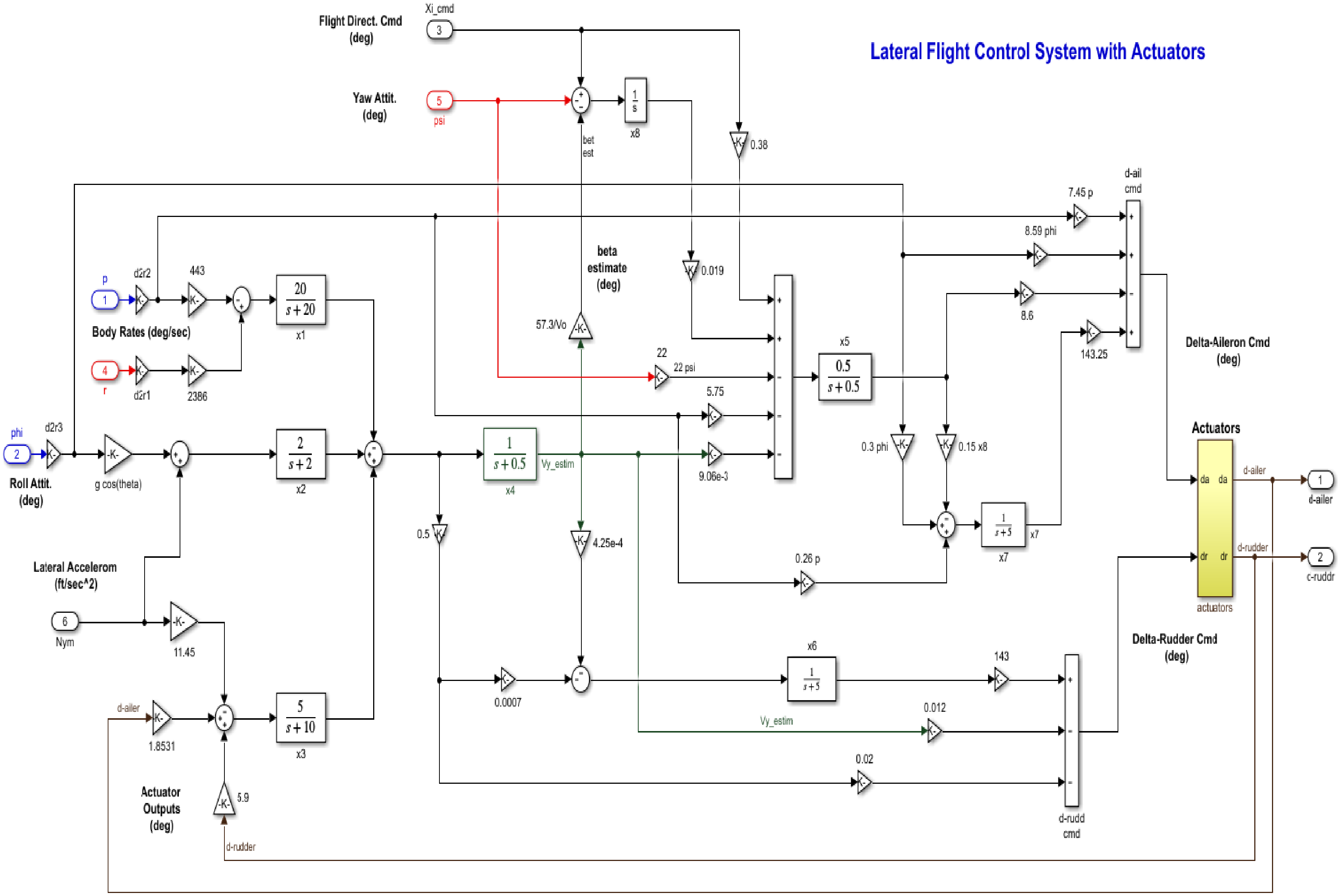
In the classical design the input to the flight control system is heading direction command (ξ_{cmd}) which is compared with the estimated heading direction (ξ) and the heading error commands the aileron actuator. The heading direction is a function of the yaw attitude ψ plus the estimated beta β_{est}, that is: $\xi=\psi+\beta_{est.}$ It is estimated from navigation and there is a computational delay involved. The β-estimate used to calculate ξ is calculated relative to the inertial frame and not relative to the moving air-mass. It is obtained from output # 8 of the vehicle model when the label "**NoWind Alpha**" is included in the flags line. Otherwise, the default is relative to the wind. The β-measurement from the vane, however, or the beta that is estimated from the lateral accelerometer, are calculated relative to the moving airflow.

The measured roll and yaw attitudes are Euler angles. The beta estimate is equal to the estimated lateral velocity divided by the nominal speed (v_y/V_0). The lateral velocity estimate mainly controls the rudder but it also cross-couples into the aileron for better roll coordination. The block diagram in Figure 1.11 shows the lateral flight control system and is including the aileron and rudder actuators. On the left side the control system attempts to estimate the lateral velocity v_y. It uses the body rates the gravity component [$g.\cos(\Theta)\phi$], and the lateral accelerometer measurement Ny. The aileron and rudder aerosurface deflections outputs are also used in the estimation of the lateral velocity.

1.4.2 Lateral Axes Simulation Model of the Classical Control Design

The lateral simulation model derived from classical controls is "*Closed_Sim_Classic.mdl*". It is shown in Figure 1.12, and it is located in folder "*C:\Flixan\Examples\Missile with Wing\ (a) Classical Analysis\Lateral*". The input to the simulation model is heading direction command $\xi_{command}$ in (deg) and wind-gust velocity in (feet/sec).

Lateral Flight Control System with Actuators



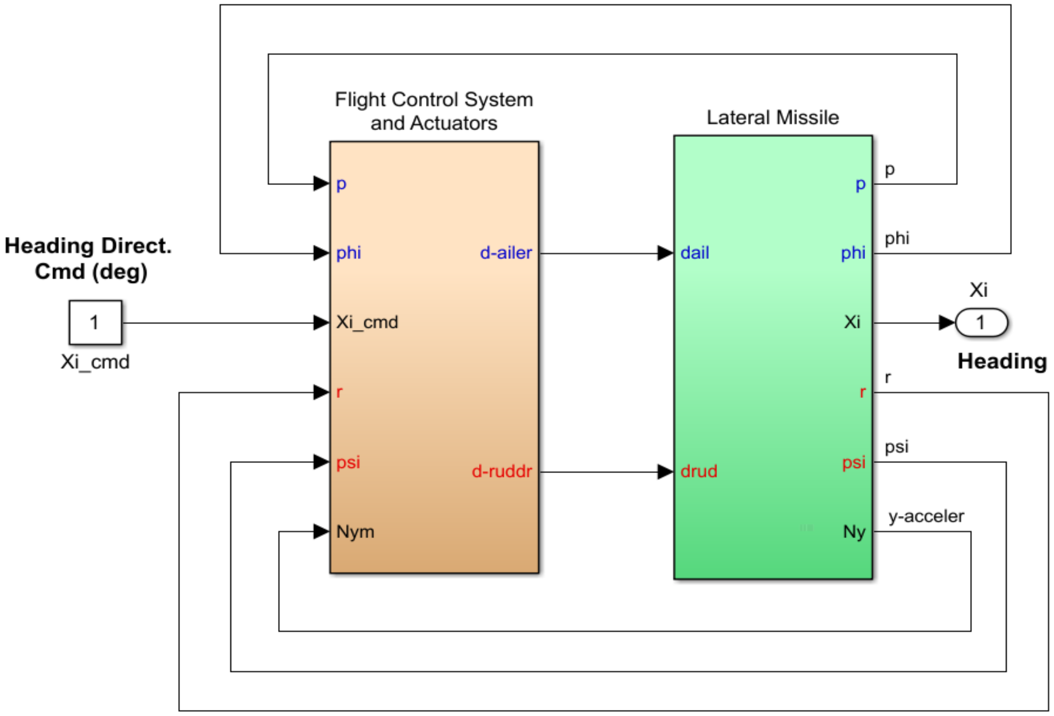


Figure 1.12 Roll and Yaw axes Simulation model "*Closed_Sim_Classic.mdl*" using the Classical FCS Design

The missile dynamic model is shown in detail in Figure 1.13. It consists of the Flixan generated state-space system: "*Missile with Wing Lateral Model*" that is loaded from file "*vehi_later.m*". The inputs to the vehicle model are aileron and rudder deflections from the aileron and rudder actuator outputs. There is also a wind-gust impulse that is smoothed by a low-lass filter. The vehicle outputs are: roll and yaw attitudes and rates, inertial beta used for calculating the heading direction, lateral accelerometer measurement (Ny), cross range velocity (Vcr), and beta relative to the wind measurement from the air-data probe. The heading direction output is: $\xi=\psi+\beta$. A delay of 0.7 sec is included in the heading direction output to capture the computational delay due to navigation.

Simulation Results

We will use the lateral simulation model to analyze the system's response to a small change in flight direction (ξ_{commd}) and also to a 30 (feet/sec) wind-gust impulse in one simulation. The initialization file "init.m" is used to load the vehicle state-space model "*vehi_later.m*", and the file "pl.m" plots the simulation results.

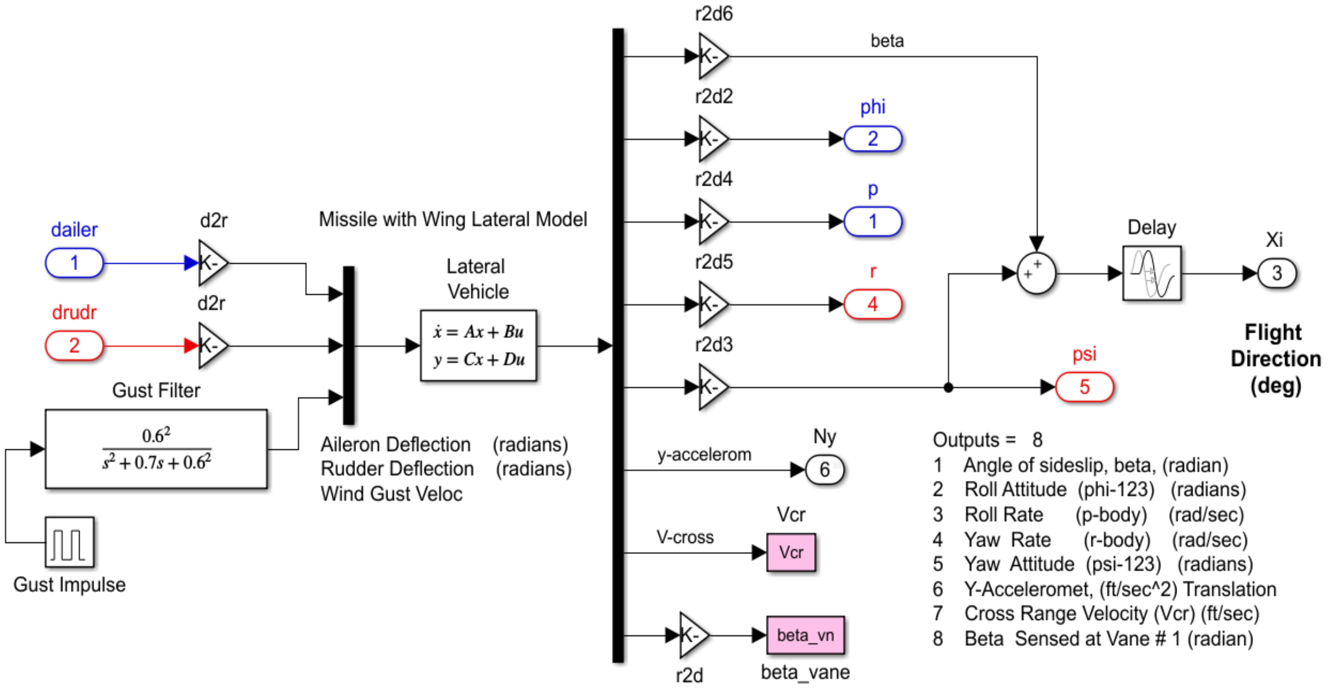

r2d6
beta
r2d2
phi
2
r2d4
p
1
dailer
1
d2r
K-
Missile with Wing Lateral Model
r2d5
r
4
Delay
Xi
3
drudr
2
d2r
K-
Lateral
Vehicle
$\dot{x} = Ax + Bu$
$y = Cx + Du$
r2d3
psi
5
Flight
Direction
(deg)
Gust Filter
$\frac{0.6^2}{s^2 + 0.7s + 0.6^2}$
Aileron Deflection (radians)
Rudder Deflection (radians)
Wind Gust Veloc
y-accelerom
Ny
6
Outputs = 8
1 Angle of sideslip, beta, (radian)
2 Roll Attitude (phi-123) (radians)
3 Roll Rate (p-body) (rad/sec)
4 Yaw Rate (r-body) (rad/sec)
5 Yaw Attitude (psi-123) (radians)
6 Y-Acceleromet, (ft/sec^2) Translation
7 Cross Range Velocity (Vcr) (ft/sec)
8 Beta Sensed at Vane # 1 (radian)
Gust Impulse
V-cross
Vcr
Vcr
beta_vn
r2d
beta_vane

Figure 1.13 Lateral Dynamic Model including a gust shaping filter

System Response to 1^o of ξ-command with Gust

The following plots in Figure 1.14 show the lateral flight control system response to 1^o adjustment in the heading direction. The control system is tracking the heading commanded with 5 (sec) lag. The command causes a small transient in beta and in roll attitude which decay to zero. The command also causes a 1 (deg) change in yaw attitude, and a 37 (feet/sec) cross-range velocity, which is perpendicular to the original flight direction.

Closed-Loop System Eigenvalues

-13.0398 +30.9142i
-13.0398 -30.9142i
-12.4918 +28.6814i
-12.4918 -28.6814i
-15.9433
-4.3486 +10.6942i
-4.3486 -10.6942i
-10.0000
-1.2105 + 3.1172i
-1.2105 - 3.1172i
-4.5274
-3.6930
-1.9850
-0.0548 + 0.0172i
-0.0548 - 0.0172i
-0.6693
-0.5244
-0.3500 + 0.4873i
-0.3500 - 0.4873i

The gust impulse is along the –Y direction and it is applied at 20 sec. It causes a negative roll and positive yaw which are compensated by the aileron and rudder deflections. The two betas (inertial and relative to wind) are plotted together for comparison. The measured beta relative to the wind initially increases since the gust is coming from the right side. The inertial beta and the lateral acceleration Ny become negative since the vehicle is pushed towards the left by the wind gust.

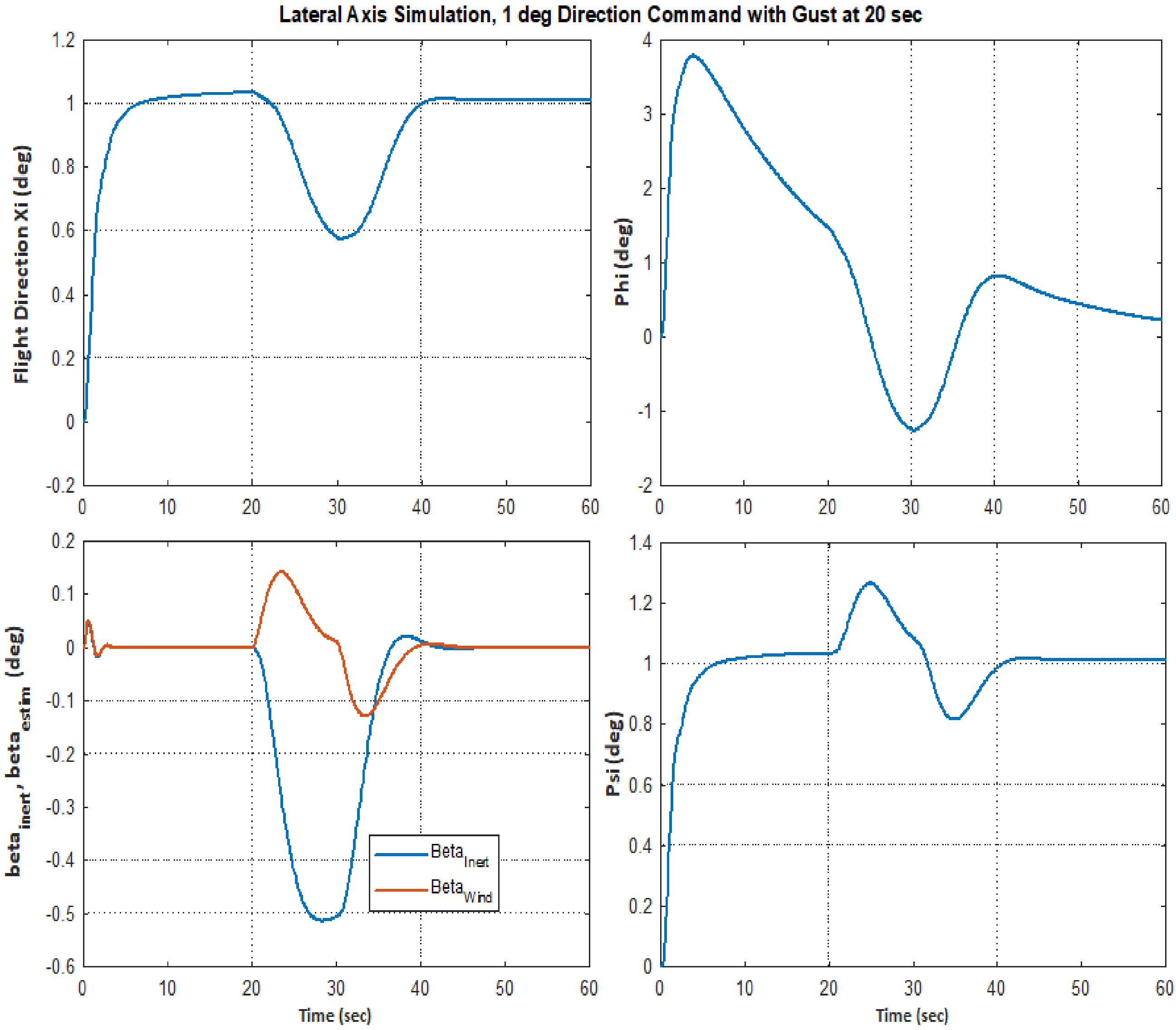


Figure 1.14a Lateral System Response to 1^o change in Heading Direction and Gust. The gust causes the vehicle to roll negative which changes its heading from the command but eventually it returns to the commanded heading. The measured beta relative to the wind increases when the gust is applied from the right side, but the other beta relative to the inertial frame is negative since the vehicle is pushed towards the left by the wind.

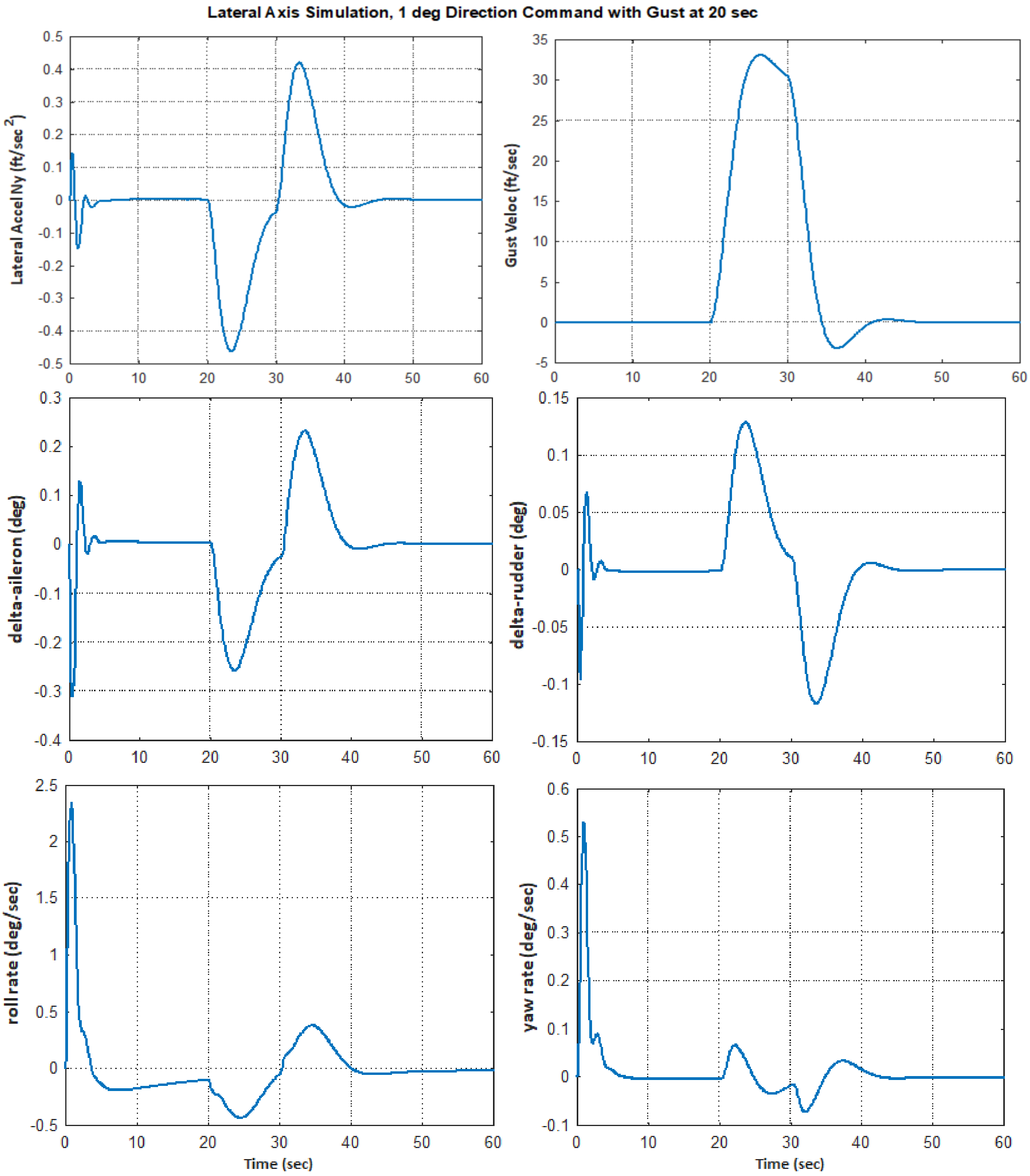


Figure 1.14b Lateral System Response to 1^o change in Heading Direction and Gust. The initial aileron and rudder deflections perform the coordinated roll/ yaw maneuver to change the heading direction. When the gust hits at 20 sec, the aileron rotates negative to counteract the negative roll rate and the rudder rotates positive to counteract the positive yaw rate produced by the gust hitting the vertical rudder

1.4.3 Lateral Stability Analysis (Classical Design)

The Simulink model "*Open_Loop_Classic.mdl*" in Figure 1.16 is used for frequency domain stability analysis. It includes the classical flight control system and the vehicle dynamic model used in the simulation model. It has two control loops, aileron and rudder, and the loops are opened at the aileron or rudder actuator outputs, one at a time, as shown in Figure 1.16 for aileron stability analysis. For rudder stability analysis we must modify the model, close the aileron loop and open the rudder. The Matlab file "freq.m" loads the lateral vehicle dynamics and calculates the frequency response from this model, as shown in Figures 1.17 (a & b), from where we can measure the gain and phase margins.

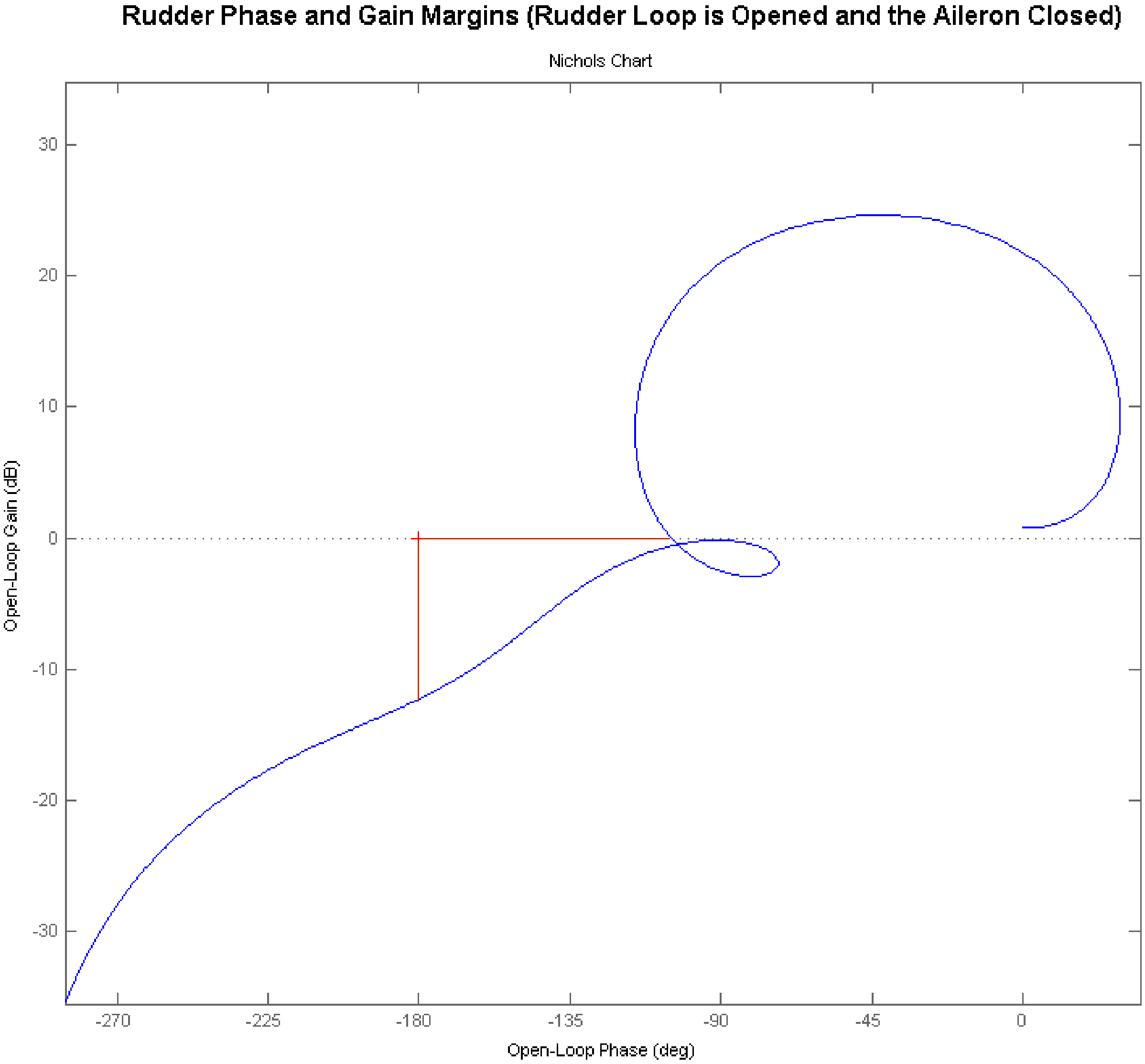


Figure 1.17a Rudder Loop Stability Analysis Showing Phase and Gain Margins

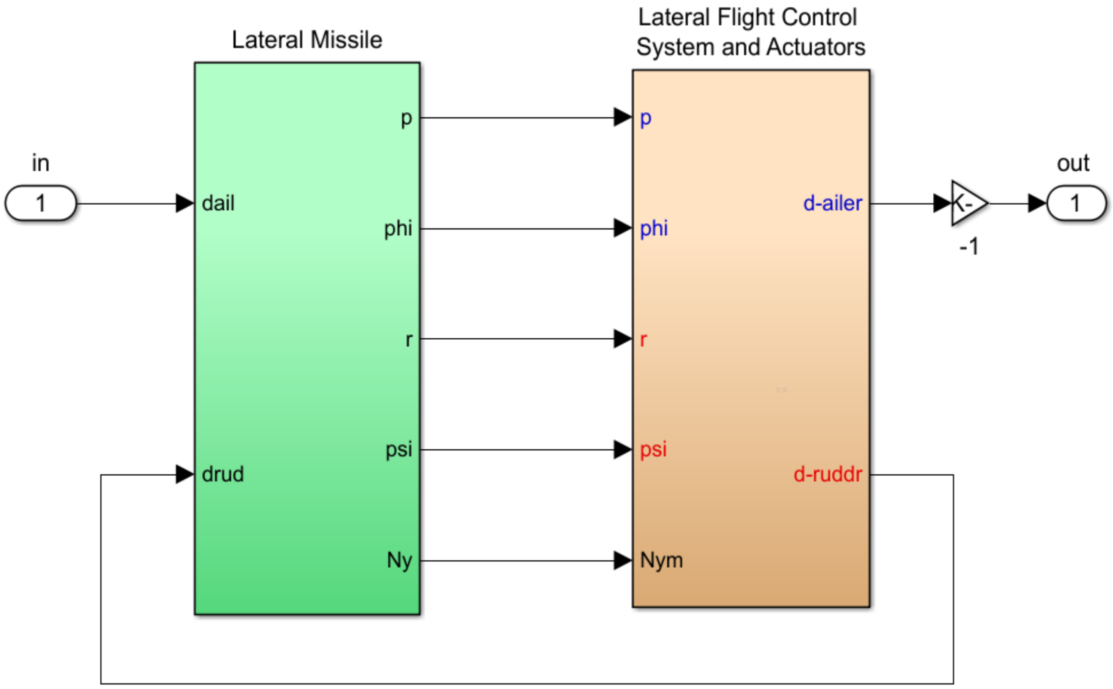


Figure 1.16 Simulink Model "Open_Loop_Classic.mdl" used for Lateral Stability Analysis. Configuration shown is for analyzing Aileron stability with Rudder Loop Closed

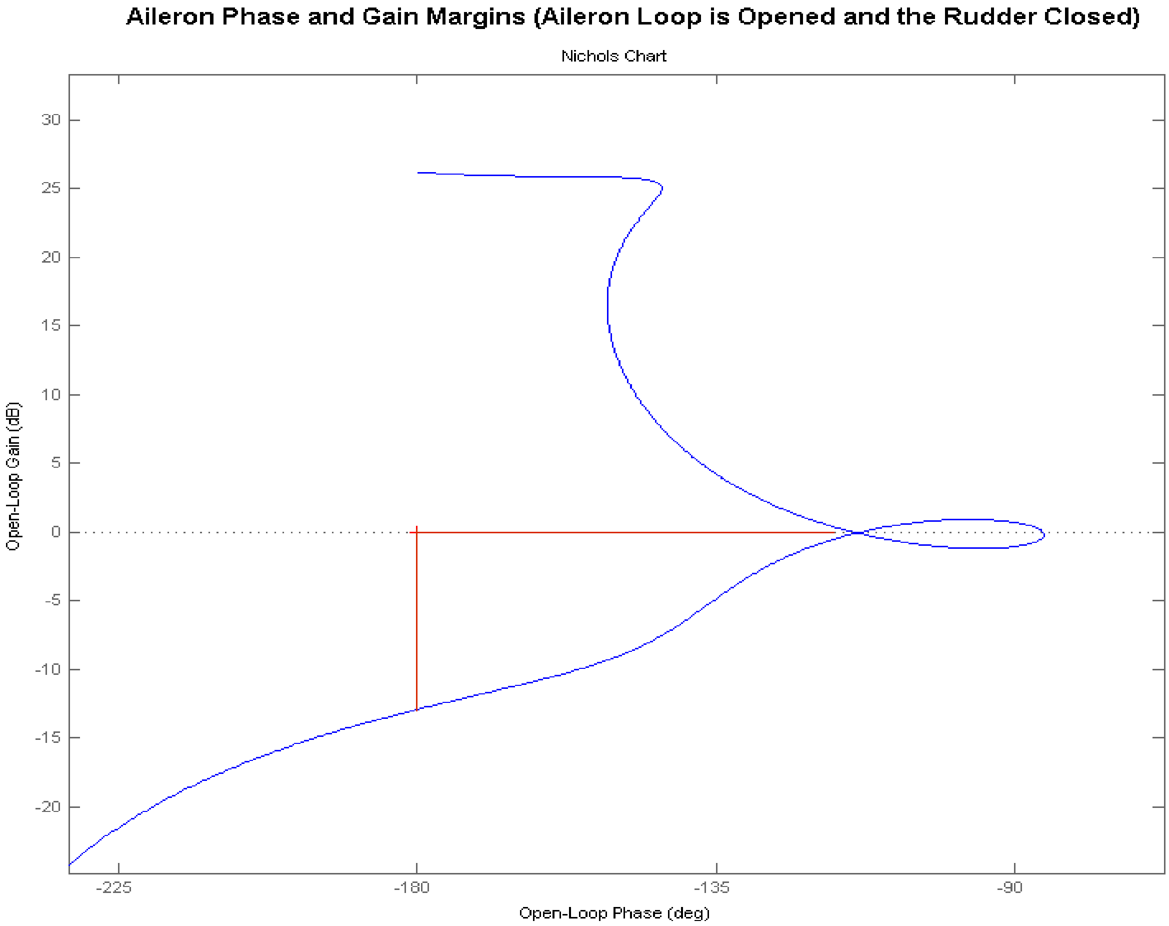


Figure 1.17b Aileron Loop Stability Analysis Showing Phase and Gain Margins

2. LQR Control Design

In this section we will demonstrate how to use the Linear Quadratic Regulator (LQR) method to design simple state-feedback control laws for this missile. The LQR method is an attractive choice for control design because it is simple to apply and the process can be automated and repeated for multiple flight conditions. The Flight Vehicle Modeling program is typically used to generate the vehicle design and analysis models at some critical flight conditions, such as: lift-off, max-Q, separation, engine cut-off, etc. A Flixan input file is created for each flight condition that includes the vehicle model and the LQR design dataset to create the state-feedback controller at each flight condition. The control gains are then interpolated in 6-dof simulations between the design points.

2.1 Control Design Overview

The missile in this example is perfectly symmetric and the control analysis will be separated into pitch and lateral analysis that will be performed in separate subdirectories. The pitch vehicle model is in the input file "*Pitch_LQR_Des.Inp*" which is located in directory "*Examples\Missile with Wing\ (b) LQR Design\Pitch LQR*". The lateral vehicle model is in the input file "*Later_LQR_Des.Inp*" which is located in directory "*Examples\Missile with Wing\ (b) LQR Design\Lateral LQR*". Pitch and Lateral control design models will be created for LQR state-feedback and we assume that all states $\underline{x}$ are available for feedback. In this example we will design state-feedback control laws for this missile at one flight condition: Mach 2.5, Qbar=1220 (psf), α=10 (deg). The vehicle dataset is already created in the input file and its title is: "*Missile with Wing, Mach: 2.5, Qbar: 1220*".

2.2 Longitudinal Control Design and Analysis

The input file for the longitudinal design is "*Pitch_LQR_Des.Inp*" located in subdirectory "*Examples\Missile with Wing\(b) LQR Design\Pitch LQR*". It contains several Flixan datasets that generate plant models and perform steady-state LQR state-feedback control design. They are processed by a batch set located at the top of the file. The batch first retains the control weight matrices Q_c and R_c in the systems file "*Pitch_LQR_Des.Qdr*" from getting erased. Then it generates the vehicle model "*Missile with Wing, Mach: 2.5, Qbar: 1220*" that includes both pitch and lateral dynamics. The initial pitch design model is then extracted from the above system and saved as "*Missile with Wing Pitch Design Model*". It consists of one input, Elevon deflection in (rad), and 3 outputs: pitch attitude, rate, and angle of attack in radians. A second longitudinal system is also created with title: "*Missile with Wing Pitch Analysis Model*". It includes a wind-gust velocity input in (feet/sec) and some additional outputs, and it will be used in simulations. The direction of the gust is perpendicular to the vehicle x-axis, and along the –z direction to excite the pitch dynamics, as defined in the vehicle input data by the wind azimuth and elevation angles (0^o and 90^o).

```
BATCH MODE INSTRUCTIONS ...............
Batch for Designing Missile with Wing Pitch Models and Gains
!
! This batch set creates the Design and Analysis models for a
! Missile with Wing at 2.5 Mach, and performs LQR design.
! The Missile has a fixed Thrust and it is controlled by 3 Aerosurfaces
!
!                  Control design Matrices
Retain Matrix    : State Weight Matrix Qc (5x5)
Retain Matrix    : Control Weight Matrix Rc
!
Flight Vehicle   : Missile with Wing, Mach: 2.5, Qbar: 1220
System Modificat : Missile with Wing Pitch Design Model
System Modificat : Missile with Wing Pitch Analysis Model
Transf-Functions : Actuator: 34/(s+34)
Transf-Functions : Integrator
System Connection: Augmented Pitch Design Model
System Modificat : Augmented Pitch Design Model-2
LQR Control Des  : LQR Control Design for Augmented Design Model
!                  Convert to Matlab
To Matlab Format : LQR State-Feedback Control for Augmented Design Model
To Matlab Format : Missile with Wing Pitch Analysis Model
----------------------------------------------------------------------------------------------------
```

```
FLIGHT VEHICLE INPUT DATA ......
Missile with Wing, Mach: 2.5, Qbar: 1220
! Rigid-Body Missile controlled by 3 aerosurfaces. The engine has fixed thrust
! and does not gimbal
Body Axes Output, Attitude=Euler Angles, NoWind Alpha

Vehicle Mass (lb-sec^2/ft), Gravity Accelerat. (g) (ft/sec^2), Earth Radius (Re) (ft)    :   1219.1, 32.07,
Moments and products of Inertias Ixx, Iyy, Izz, Ixy, Ixz, Iyz, in (lb-sec^2-ft)          :   0.4063E+04 0.1654E+06
CG location with respect to the Vehicle Reference Point, Xcg, Ycg, Zcg, in (feet)        :   26.19, 0.0,  -0.15
Vehicle Mach Number, Velocity Vo (ft/sec), Dynamic Pressure (psf), Altitude (feet)       :   2.5, 2427.4,1220.6,
Inertial Acceleration Vo_dot, Sensed Body Axes Accelerations Ax,Ay,Az (ft/sec^2)         :   60.0, 60.0, 0.0, 10.5
Angles of Attack and Sideslip (deg), alpha, beta rates (deg/sec)                         :   10.5, 0.0, 0.0, 0.0
Vehicle Attitude Euler Angles, Phi_o,Thet_o,Psi_o (deg), Body Rates Po,Qo,Ro (deg/sec)   :   0.0,39.6,0.0, 0.0, 0.132,
Wind Gust Vel wrt Vehi (Azim & Elev) angles (deg), or Force(lb), Torque(ft-lb), locat:xyz:   Gust  00.0  90.0
Surface Reference Area (feet^2), Mean Aerodynamic Chord (ft), Wing Span in (feet)        :   145.4, 22.0, 22.0
Aero Moment Reference Center (Xmrc,Ymrc,Zmrc) Location in (ft), {Partial_rho/ Partial_H} :   26.19, 0.0, -0.238, 0.0
Aero Force Coef/Deriv (1/deg), Along -X, {Cao,Ca_alf,PCa/PV,PCa/Ph,Ca_alfdot,Ca_q,Ca_bet}:   0.1, 0.002, 0.0, 0.0,
Aero Force Coeffic/Derivat (1/deg), Along Y, {Cyo,Cy_bet,Cy_r,Cy_alf,Cy_p,Cy_betdot,Cy_V}:   0.0, -0.023, 0.0, 0.0,
Aero Force Coeff/Deriv (1/deg), Along Z, {Czo,Cz_alf,Cz_q,Cz_bet,PCz/Ph,Cz_alfdot,PCz/PV}:  -0.1, -0.032, 0.0, 0.0,
Aero Moment Coeffic/Derivat (1/deg), Roll: {Clo, Cl_beta, Cl_betdot, Cl_p, Cl_r, Cl_alfa}:   0.0, -0.0017,0.0,-0.243,
Aero Moment Coeff/Deriv (1/deg), Pitch: {Cmo,Cm_alfa,Cm_alfdot,Cm_bet,Cm_q,PCm/PV,PCm/Ph}:  -0.037,-0.011, 0.0,0.0,
Aero Moment Coeffic/Derivat (1/deg), Yaw : {Cno, Cn_beta, Cn_betdot, Cn_p, Cn_r, Cn_alfa}:   0.0, 5.6e-4, 0.0,0.1388,

Number of Control Surfaces, With or No TWD (Tail-Wags-Dog and Hinge Moment Dynamics) ?   :  3   No TWD

Control Surface No:  1                                                                   Elevator
Trim Angle, Max/Min Deflection Angles from Trim, Hinge Line Angles: phi_h, lamda_h  (deg):  0.0  30.0 -30.0 0.000
Surface Mass, Inertia about Hinge, Moment Arm (Hinge to Surface CG), Surface Chord, Area :  0.0  0.0  0.0  0.0
Hinge Moment Derivatives (1/deg), { Chm_Alpha, Chm_Beta, Chm_Delta, Chm_Mach }           :  0.0  0.0  0.0  0.0
Location of the Hinge Line Center with respect to Vehicle Reference (feet), {Xcs,Ycs,Zcs}:  0.0  0.0  0.0
Forces (-x,y,z) due to Deflect. and Rates {Ca_del,Cy_del,Cz_del, Ca_deld,Cy_deld,Cz_deld}:  0.00003 -0.0 -0.0087, 0.00
Moments due to Deflections and Rates {Cl_del,Cm_del,Cn_del,Cl_deldot,Cm_deldot,Cn_deldot}:  0.0 -0.0072 0.0  0.0

Control Surface No:  2                                                                   Aileron
Trim Angle, Max/Min Deflection Angles from Trim, Hinge Line Angles: phi_h, lamda_h  (deg):  0.0 30.0  -30.0 0.0
Surface Mass, Inertia about Hinge, Moment Arm (Hinge to Surface CG), Surface Chord, Area :  0.0   0.0  0.0  0.0
Hinge Moment Derivatives (1/deg), { Chm_Alpha, Chm_Beta, Chm_Delta, Chm_Mach }           :  0.0  0.0   0.0    0.0
Location of the Hinge Line Center with respect to Vehicle Reference (feet), {Xcs,Ycs,Zcs}:  0.0    0.0     0.0
Forces (-x,y,z) due to Deflect. and Rates {Ca_del,Cy_del,Cz_del, Ca_deld,Cy_deld,Cz_deld}:  0.00003 0.0011 0.0  0.0
Moments due to Deflections and Rates {Cl_del,Cm_del,Cn_del,Cl_deldot,Cm_deldot,Cn_deldot}: -6.54e-4 0.0  -0.0014  0.0

Control Surface No:  3                                                                   Rudder
Trim Angle, Max/Min Deflection Angles from Trim, Hinge Line Angles: phi_h, lamda_h  (deg):  0.0 30.0 -30.0 0.000
Surface Mass, Inertia about Hinge, Moment Arm (Hinge to Surface CG), Surface Chord, Area :  0.0 0.0 0.0 0.0  0.0
Hinge Moment Derivatives (1/deg), { Chm_Alpha, Chm_Beta, Chm_Delta, Chm_Mach }           :  0.0 0.0 0.0 0.0
Location of the Hinge Line Center with respect to Vehicle Reference (feet), {Xcs,Ycs,Zcs}:  0.0 0.0 0.0
Forces (-x,y,z) due to Deflect. and Rates {Ca_del,Cy_del,Cz_del, Ca_deld,Cy_deld,Cz_deld}:  0.00001 0.0034 0.0 0.0
Moments due to Deflections and Rates {Cl_del,Cm_del,Cn_del,Cl_deldot,Cm_deldot,Cn_deldot}:  5.9456e-4 0.0 -0.0035

Number of Vane Sensors, (Measuring Alpha or Beta)                                        :  1
Vane No 1 Location in Vehicle Axes  (feet)                                               :  53.0, 0.0, 0.0

Number of Bending Modes                                                                  :  0
-----------------------------------------------------------------------------------------------------------
```

The missile vehicle dataset, shown above, includes 3 aerosurfaces: Elevator, Aileron and Rudder. It also includes an air-data sensor located in front of the vehicle for measuring the angles (α and β). The wind-gust direction is perpendicular to the x-axis and along the –z direction. The vehicle dataset also has the "**NoWind Alpha/ Beta**" flag included. It means that we want the incidence angles α and β to be inertial and not relative to the moving air-flow. In this example they are used to synthesize the flight-path and heading direction angles γ and ξ which are estimated from navigation and they do not "feel" the effects of a wind-gust directly. It means that the wind velocity components w_{gust} and v_{gust} are not included in the α and β calculations, only the vehicle velocities w and v. A wind-gust, however, will produce forces and moments on the vehicle and it will affect its motion, but the gust itself is not seen directly in the α and β output (states: 7, 8) as it is seen in the α and β measurements of the air-data probe, only its effect on the vehicle are observable.

```
CREATE A NEW SYSTEM FROM AN OLD SYSTEM... (Titles of the New and Old Systems)
Missile with Wing Pitch Design Model
Missile with Wing, Mach: 2.5, Qbar: 1220
! The initial pitch design system is extracted from the coupled RB system above
!
TRUNCATE OR REORDER THE SYSTEM INPUTS, STATES, AND OUTPUTS
Extract Inputs :   1
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)
Missile with Wing Pitch Analysis Model
Missile with Wing, Mach: 2.5, Qbar: 1220
! The Pitch Analysis/ Simulation system is extracted from the coupled RB system above
!
TRUNCATE OR REORDER THE SYSTEM INPUTS, STATES, AND OUTPUTS
Extract Inputs :   1   4
Extract States :   3   4   7   9  10
Extract Outputs:   3   4   7   9  10  13  14
-----------------------------------------------------------------------------------------------------------------------
```

The system modification datasets extract the longitudinal variables from the coupled system "*Missile with Wing, Mach: 2.5, Qbar: 1220*" and save them in file "*Pitch_LQR_Des.Qdr*" as separate design and analysis models.

The LQR gains are calculated from rigid-body design models without tail-wag-dog and flexibility included, and assuming that all states are available for feedback. They are, however, often adjusted in order to accommodate high order dynamics which are missing in the original design plant. The original design plant, however, is not capable of producing an efficient control design. In the longitudinal direction we would like to command and track a pre-calculated flight path angle γ and the initial design model is not equipped to regulate γ. We must create, therefore, and regulate a "γ–integral" state and include it in the design model. It is also good to include a simple actuator model in the plant dynamics because it introduces more plant information in the design and makes the control system more efficient with less phase-lag. The two additional variables γ and δ_{elevon} are both measurable.

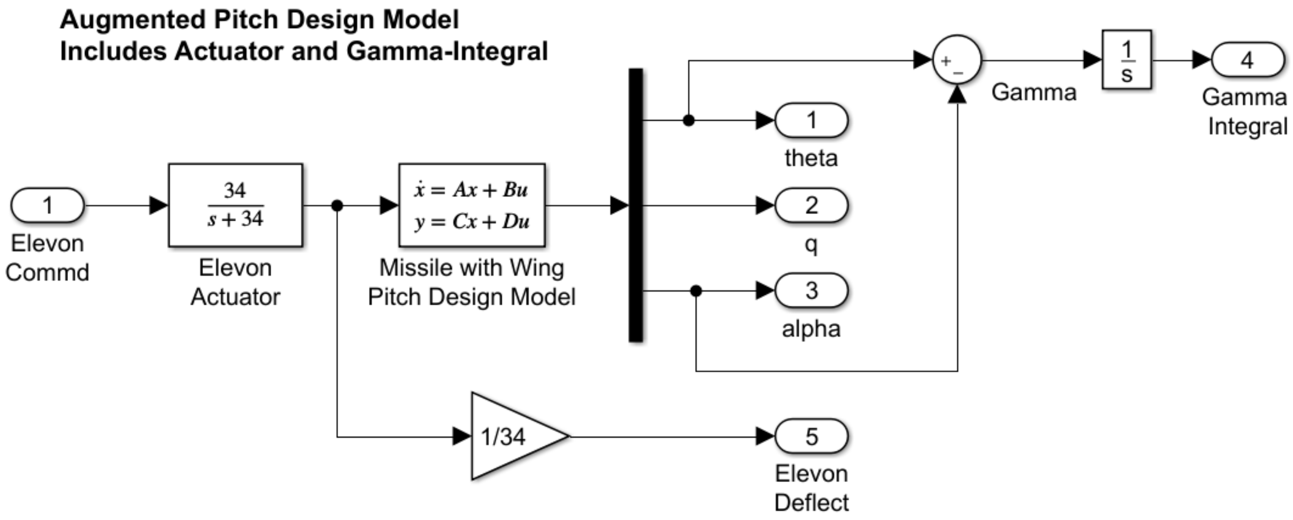


Figure 2.1 Augmented Longitudinal Design Plant for LQR Control Design

Figure 2.1 shows the augmented plant for the longitudinal LQR control design. The interconnection dataset below combines the 3 subsystems and generates the augmented system as "*Augmented Pitch Design Model*". The order of the states, however, is not the same as the outputs and it is modified for convenience to "*Augmented Pitch Design Model-2*" which makes the C matrix equal to the identity I$_5$.

```
INTERCONNECTION OF SYSTEMS .....
Augmented Pitch Design Model
! Create a 5-State Augmented Model that Includes Gamma-integral and
! Elevon deflection in the state vector for Pitch Control Design
!
Titles of Systems to be Combined
Title 1 Actuator: 34/(s+34)
Title 2 Missile with Wing Pitch Design Model
Title 3 Integrator
SYSTEM INPUTS TO SUBSYSTEM  1                                                          to Actuator
System Input  1 to Subsystem  1, Input  1, Gain= 1.0                                   Delta Command
..............................................................................
SYSTEM OUTPUTS FROM SUBSYSTEM  2                                                       Vehicle Plant
System Output  1 from Subsystem  2, Output  1, Gain= 1.0                               theta
System Output  2 from Subsystem  2, Output  2, Gain= 1.0                               q - pitch rate
System Output  3 from Subsystem  2, Output  3, Gain= 1.0                               alpha
..............................................................................
SYSTEM OUTPUTS FROM SUBSYSTEM  3                                                       Integrator
System Output  4 from Subsystem  3, Output  1, Gain= 1.0                               gamma-integral
..............................................................................
SYSTEM OUTPUTS FROM SUBSYSTEM  1                                                       Actuator
System Output  5 from Subsystem  1, Output  1, Gain= 0.0294118                         delta-elevon
..............................................................................
SUBSYSTEM NO  1 GOES TO SUBSYSTEM NO  2                                                Actuator to Vehicle
Subsystem  1, Output  1 to Subsystem  2, Input  1, Gain=   1.0000                      Elevon deflect
......................................................................
SUBSYSTEM NO  2 GOES TO SUBSYSTEM NO  3                                                Vehicle to Integrator
Subsystem  2, Output  1 to Subsystem  3, Input  1, Gain=   1.0000                      Gamma= Theta
Subsystem  2, Output  3 to Subsystem  3, Input  1, Gain=  -1.0000                            -Alpha
......................................................................
Definitions of Inputs  =    1
Elevon Deflection Command (delta) rad

Definitions of Outputs =    5
Pitch Attitude, theta  (rad)
Pitch Rate, q (rad/sec)
Angle of Attack, alpha (rad)
Gamma-Integral (rad-sec)
Elevon Deflection, delta-elev (rad)
--------------------------------------------------------------------------
```

```
SYSTEM OF TRANSFER FUNCTIONS ...
Actuator: 34/(s+34)
! First order Actuator 34 (rad/sec) Bandwidth
Continuous
TF. Block #  1 34/(s+34)                                   Order of Numer, Denom=  0  1
Numer 0.0         34.0
Denom 1.0         34.0
-------------------------------------------------------------------------------
Block #, from Input #, Gain
  1  1    1.00000
...........................
Outpt #, from Block #, Gain
  1  1    1.00000
...........................
Definitions of Inputs  =   1
Delta Command

Definitions of Outputs =   1
Delta Out
-------------------------------------------------------------
```

```
SYSTEM OF TRANSFER FUNCTIONS ...
Integrator
Continuous
TF. Block #  1  1/s                                       Order of Numer, Denom=  0  1
Numer 0.0         1.0
Denom 1.0         0.0
-------------------------------------------------------------------------------
Block #, from Input #, Gain
  1  1    1.00000
...........................
Outpt #, from Block #, Gain
  1  1    1.00000
...........................
-------------------------------------------------------------
```

```
CREATE A NEW SYSTEM FROM AN OLD SYSTEM... (Titles of the New and Old Systems)
Augmented Pitch Design Model-2
Augmented Pitch Design Model
! Rearange the Order of States to be the same as the Outputs
! Makes C=Identity
TRUNCATE OR REORDER THE SYSTEM INPUTS, STATES, AND OUTPUTS
Extract States :   2   3   4   5   1
Extract Outputs:   1   2   3   4   5
------------------------------------------------------------------------------
```

The elements of the diagonal matrices Q and R are usually adjusted sequentially by observing the system’s response to commands or disturbances in closed-loop simulations, and trading between speed of response and Elevon deflection.

```
LINEAR QUADRATIC REGULATOR STATE-FEEDBACK CONTROL DESIGN
LQR Control Design for Augmented Design Model
Plant Model Used to Design the Control System from:      Augmented Pitch Design Model-2
Criteria Optimization Output is Matrix C
State Penalty Weight (Qc) is Matrix:   Qc5               State Weight Matrix Qc (5x5)
Control Penalty Weight (Rc) is Matrix: Rc                Control Weight Matrix Rc
Continuous LQR Solution Using Laub Method
LQR State-Feedback Control Gain Matrix Kc                LQR State-Feedback Control for Augmented Design Model
------------------------------------------------------------------------------------------------------------

CONVERT TO MATLAB FORMAT ........      (Title, System/Matrix, m-filename)
LQR State-Feedback Control for Augmented Design Model
Matrix Kc
------------------------------------------------------------------------------------------------------------
CONVERT TO MATLAB FORMAT ........      (Title, System/Matrix, m-filename)
Missile with Wing Pitch Analysis Model
System
Vehi_pitch.m
------------------------------------------------------------------------------------------------------------
END-END-END-END-END-END-END-END-END-END-END-END-END-END-END-END-END-END-END-END-END-END-END-END-END-END-END-END-END-
```

The dataset "*LQR Control Design for Augmented Design Model*" performs the LQR control design on the plant "Augmented Pitch Design Model-2". It uses the C matrix for criteria which is the identity matrix and the (5x5) weight matrix Q_c to penalize the states individually. The scalar R_c penalizes the Elevon control. The weight matrices are already set in the systems file. The LQR program generates the (1x5) state-feedback matrix K_c that stabilizes the plant by closing the control loop between the state vector and the Elevon input. The matrix is also saved in the systems file under the title "*LQR State-Feedback Control for Augmented Design Model*". The matrix K_c and the pitch analysis model are also saved in Matlab format as "Kc.mat" and "vehi_pitch.m" respectively for further analysis.

2.3 Longitudinal Simulation

There are two simulation models is in folder "*Examples\Missile with Wing\ (b) LQR Design\Pitch LQR*". They both use state-feedback from the 5 plant states via matrix K_c which includes γ-integral and δ_{elevon} feedback in addition to the feedback from the original vehicle states (θ, q, α). However, a perfect alpha cannot be measured directly and we will examine two options: an alpha feedback from the vane sensor located at the nose of the missile, shown in Figure 2.2 and simulation file: "*Pitch_Sim1.mdl*", and an estimated alpha from the accelerometer and elevon signals, shown in Figure 2.3 and in simulation file "*Pitch_Sim2.mdl*". The vehicle model in Figure 2.4 includes the Flixan generated system "*Missile with Wing Pitch Analysis Model*" from file "*vehi_pitch.m*", a second order Elevon actuator, a wind-gust filter, and 0.7 sec delay on the gamma output. The alpha-estimator used in Fig. 2.2 is shown in detail in Fig. 2.4 and it includes a low-pass filter to improve stability. Both simulations produce similar results.

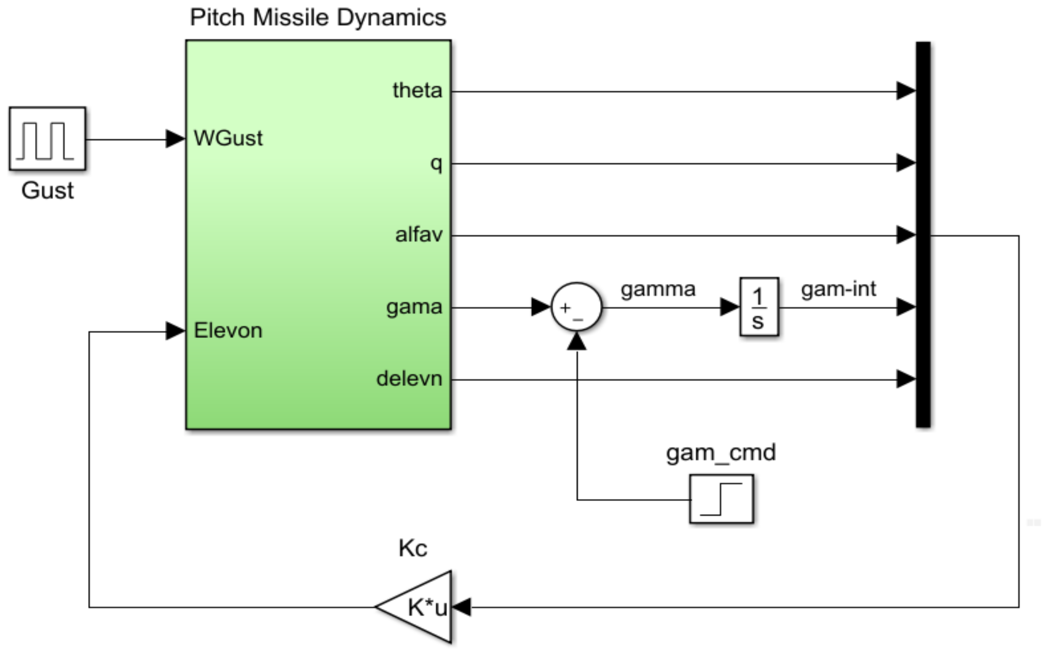


Figure 2.2 Pitch Axes Closed-Loop Simulation Model "Pitch_Sim1.mdl" Using Alpha-Feedback from the Air-Data Sensor

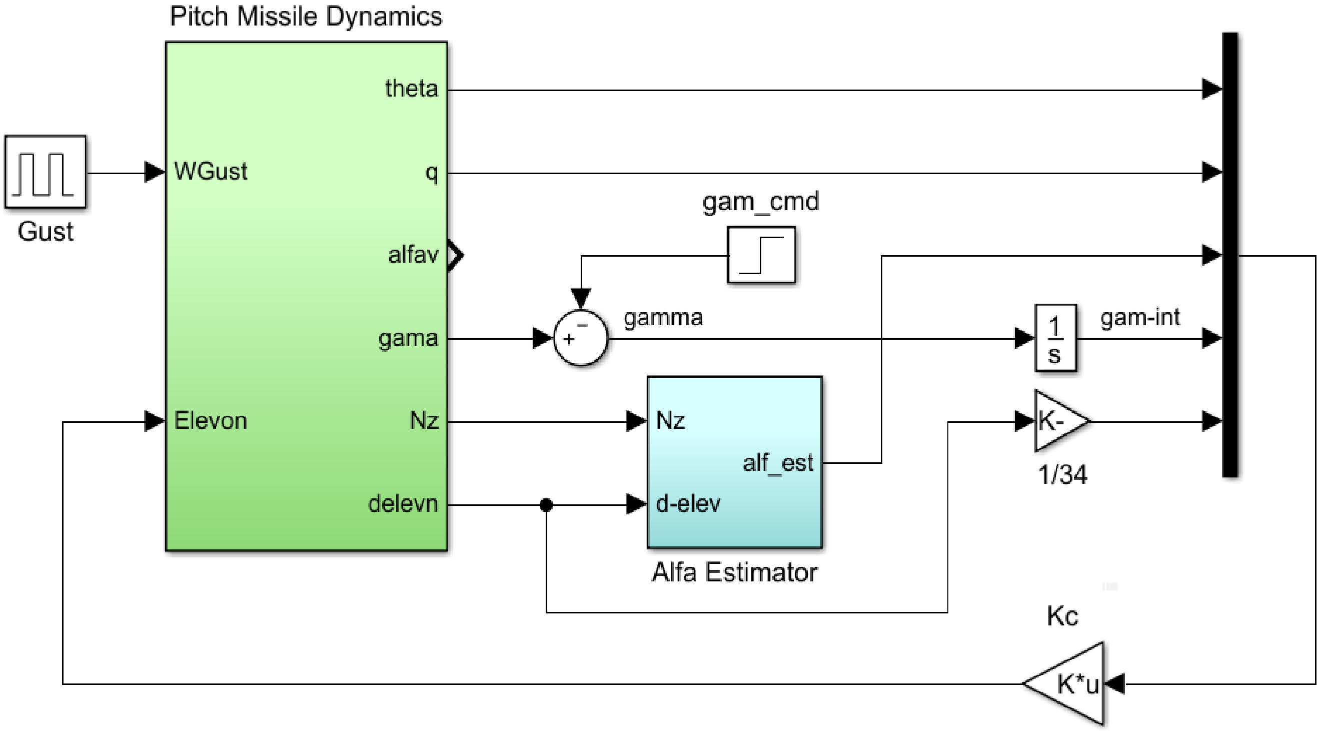


Figure 2.2 Pitch Axes Closed-Loop Simulation Model "Pitch_Sim2.mdl" Using Alpha-Feedback from the Alpha Estimator

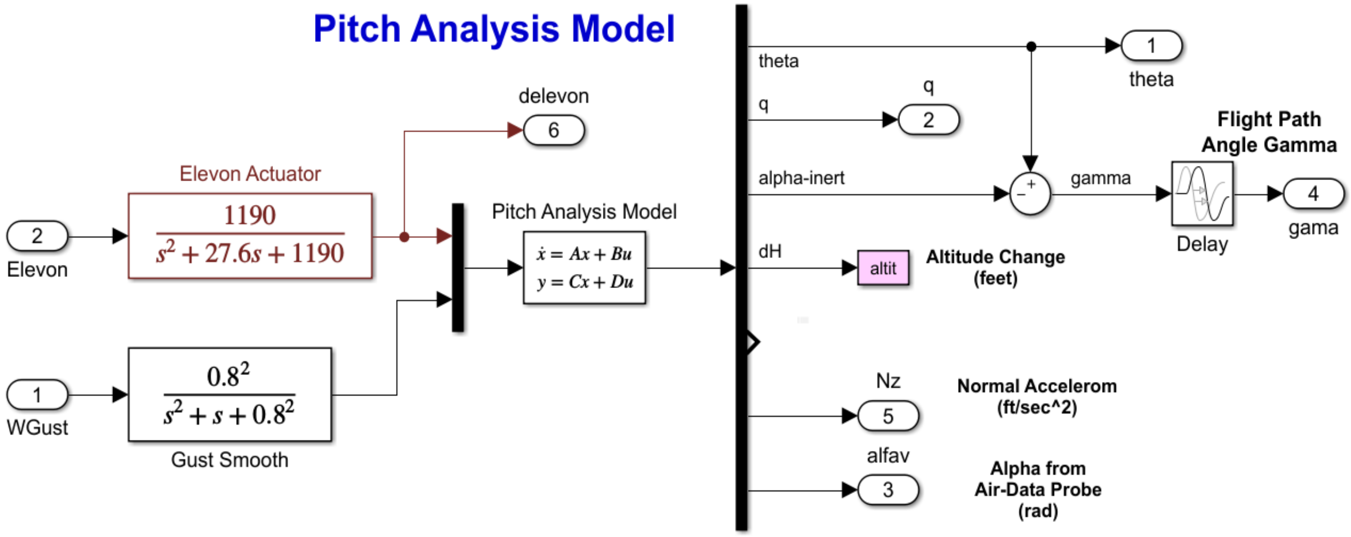


Figure 2.3 Longitudinal Flight Vehicle Subsystem with Actuator, Wind-Gust, and Gamma Delay Included

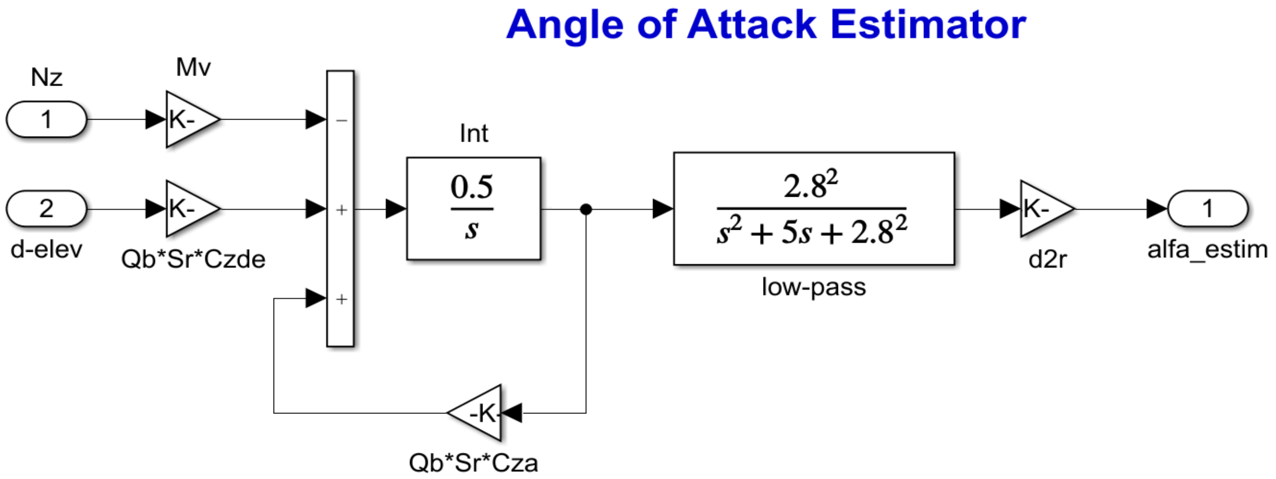


Figure 2.4 Angle of Attack Estimator processing signals from the Nz Accelerometer and the Actuator Deflection. Includes a Low-Pass Filter

In Figure 2.5 the second simulation model is commanded to perform 1^{o} increase in γ. In Figure 2.6 the missile is excited by an upward impulse of wind-gust velocity that causes the vehicle to respond by deflecting the elevon. Notice the three different alphas present: α-measured, α-inertial and α-estimated.

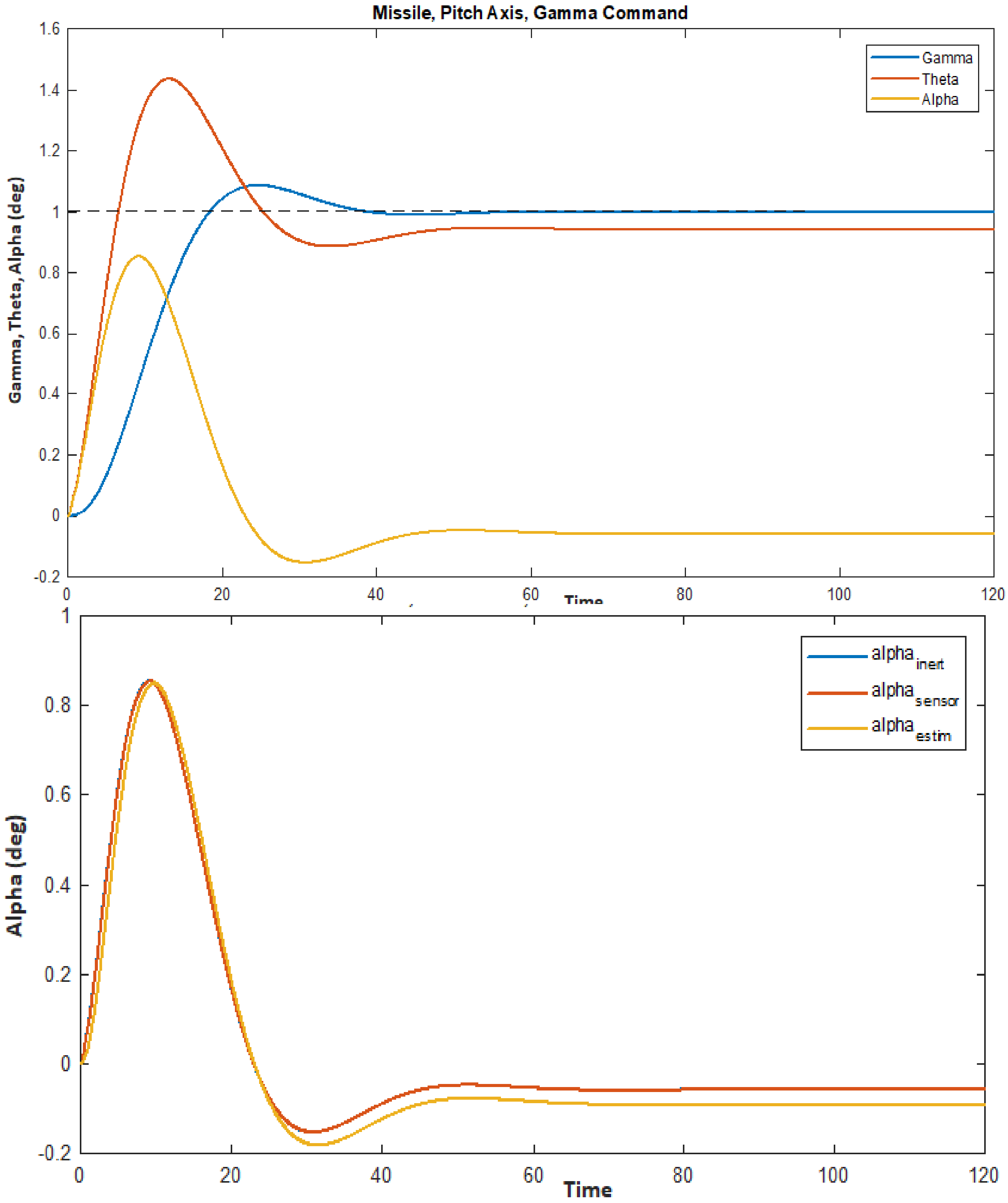


Figure 2.5 Flight Path Angle Responds to 1 degree Gamma Command. Alpha Estimate is similar to the Measured Alpha. In the absence of Wind: Alpha-Inertial and Alpha-Measured Are Equal

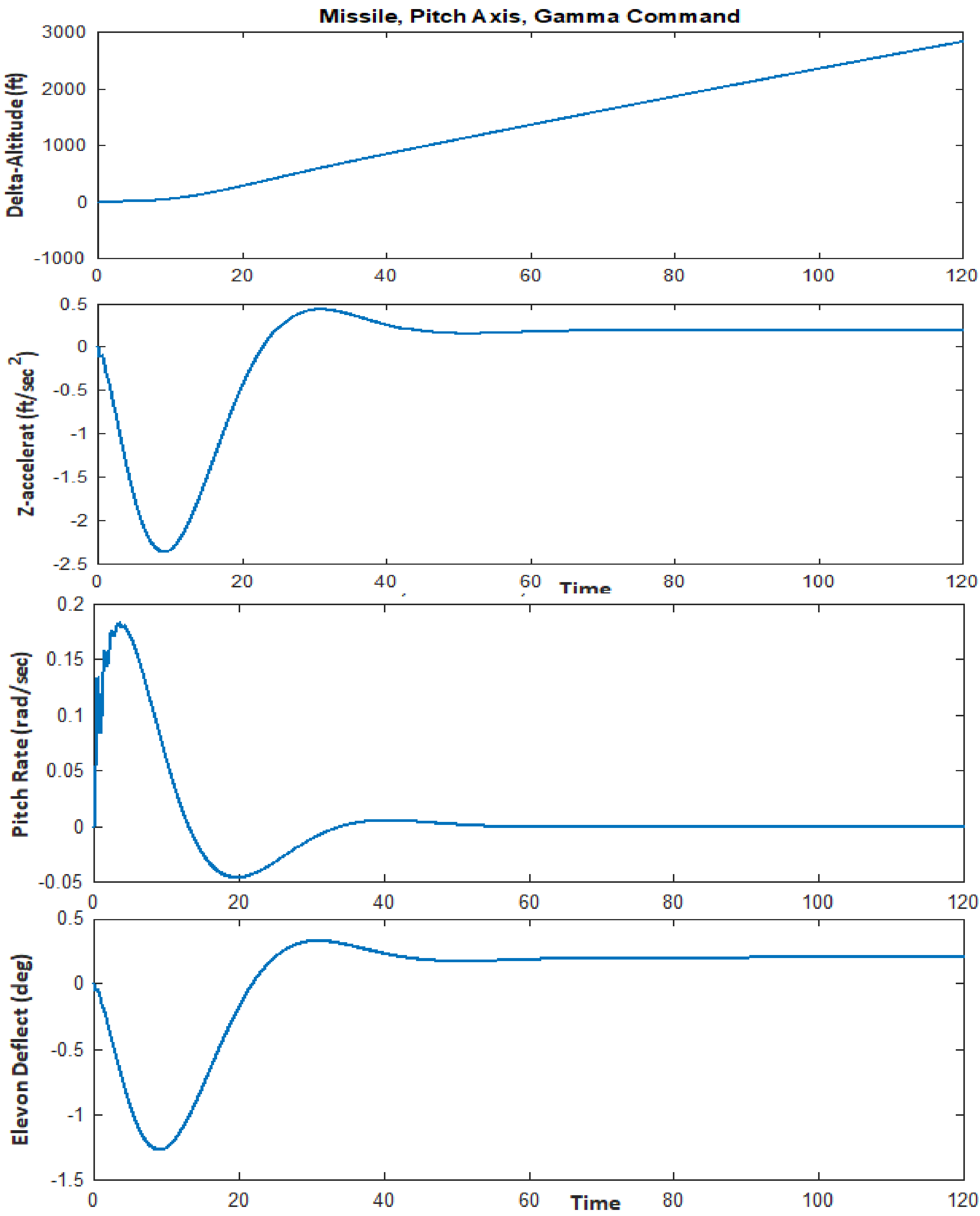


Figure 2.5b Missile Responds to the Gamma Command. Negative Elevon produces a Positive Pitch Rate, Negative (upwards) Acceleration and the Missile is Steadily Climbing at Higher Altitudes

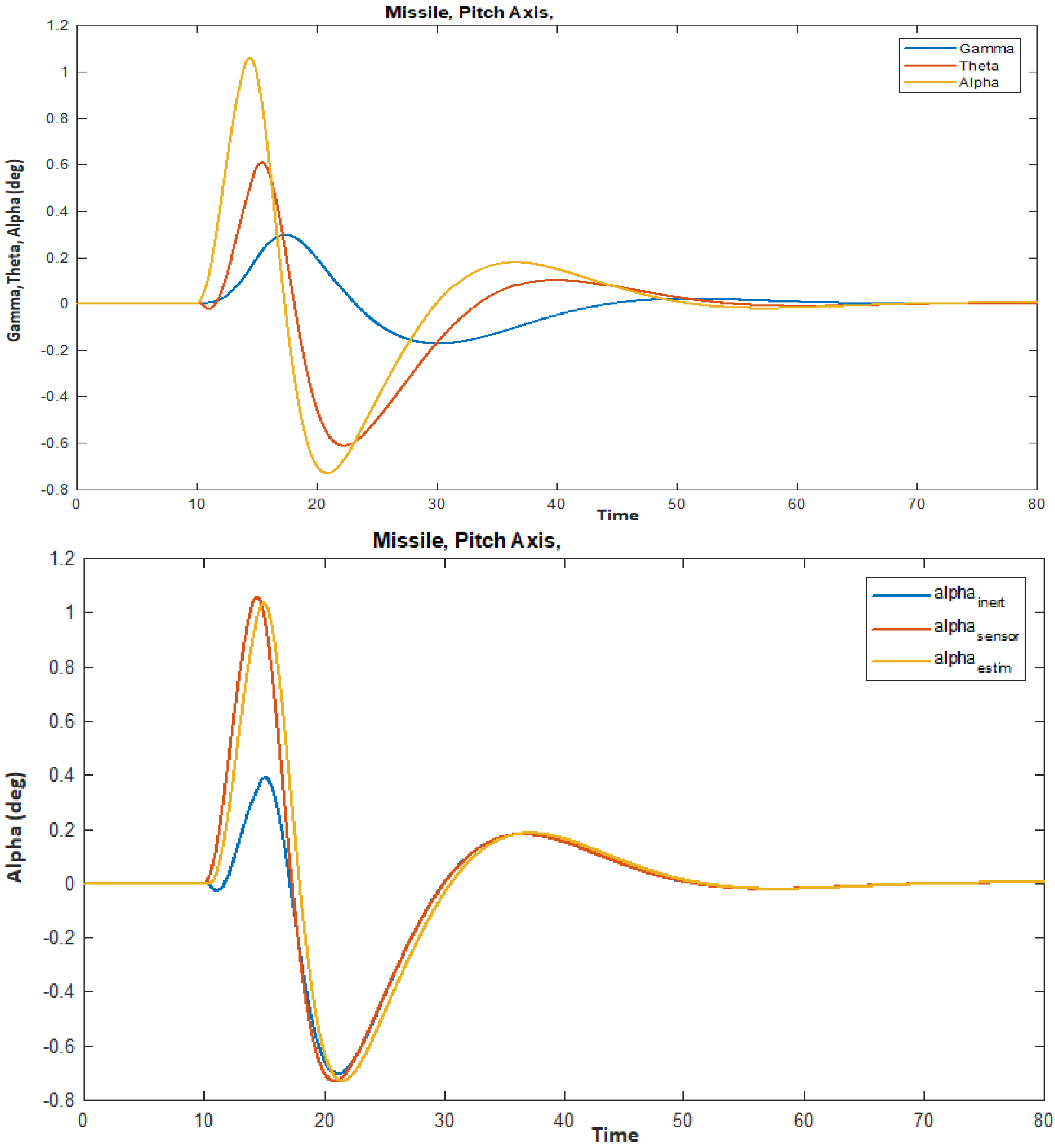


Figure 2.6a The Missile is excited by a Wind-Gust from below. The Elevon Responds with Negative Deflection (up) which causes Positive Pitch and Increases Alpha. The Alpha-Measured and Alpha-Estimated Relative to the Wind are almost the same. The Alpha-Inertial is smaller but it is different only while the Wind is present. It becomes the same as the other two when the Wind Decays.

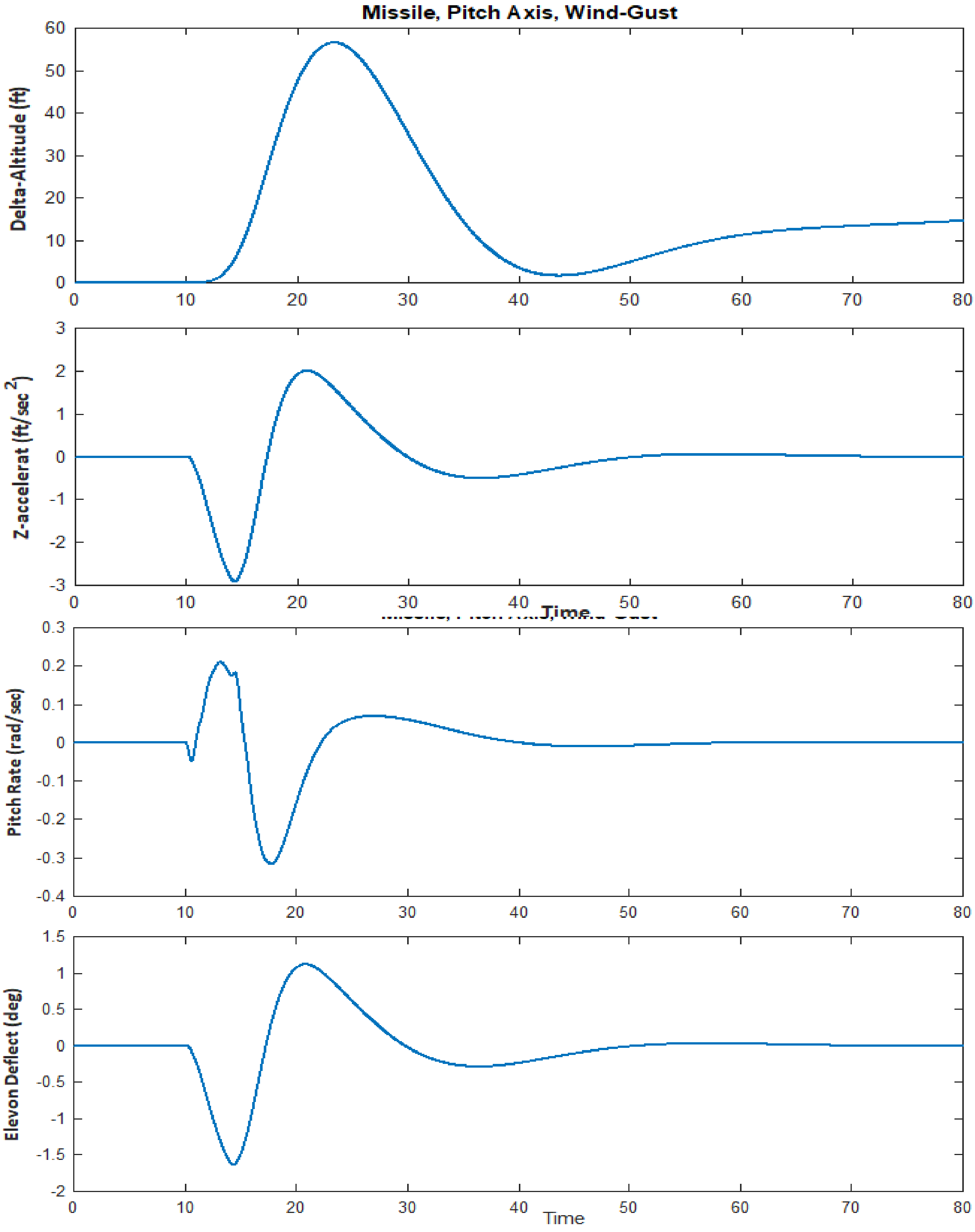


Figure 2.6b When the Gust Hits the Vehicle from Below it begins to develop Negative Pitch Rate and Negative (upwards) Acceleration. The Elevon Reacts with Negative Deflection (up) and changes the Pitch Rate to Positive and increases Altitude

2.4 Stability Analysis

The system stability is analyzed in the frequency domain by calculating the Nichols plots for both designs using the open-loop Simulink models shown in Figure 2.7. The model "*Open_Pitch1.mdl*" is used for the measured alpha case, and the model "*Open_Pitch2.mdl*" is used for the estimated alpha.

The Matlab script file "frequ.m" calculates the frequency response across the opened loop. The model includes the first order actuator. The loop is broken between the state-feedback gain output and the actuator input. Figure 2.8 shows the control system's stability margins for both designs.

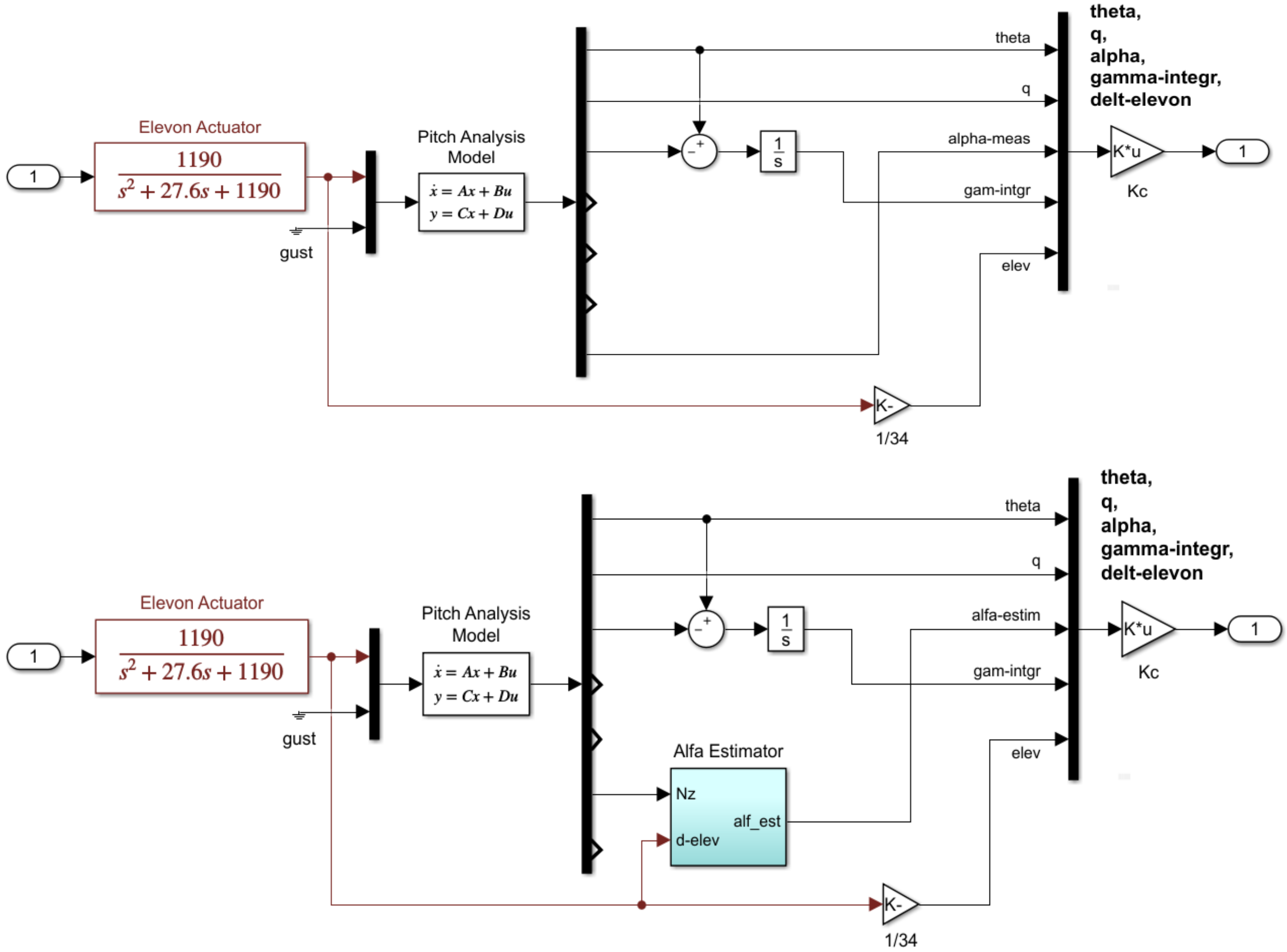


Figure 2.7 Open-Loop Models for Pitch Stability Analysis: (a) using measured Alpha, and (b) using Estimated Alpha

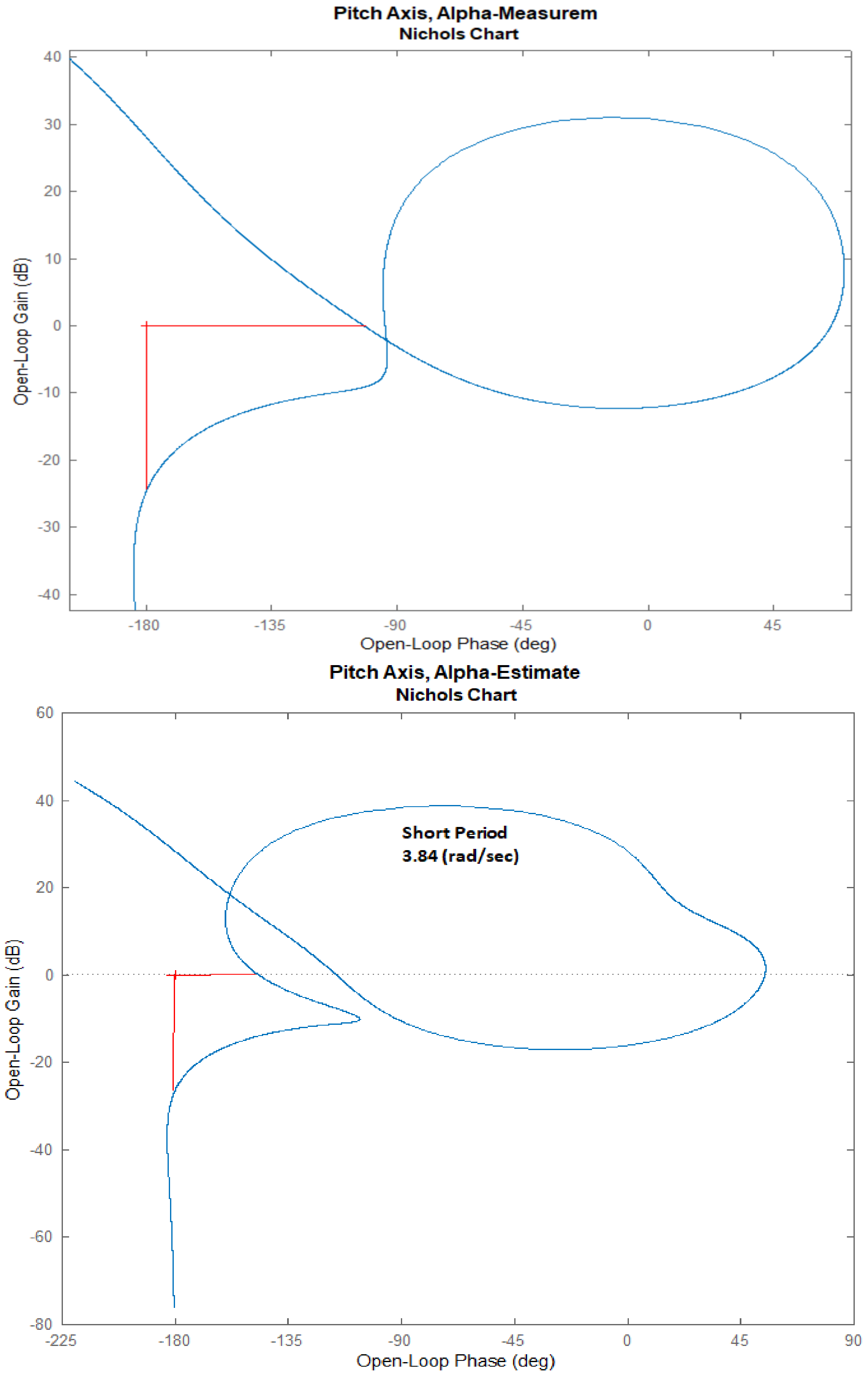


Figure 2.8 Nichols Plots for the Pitch LQR Control System with Alpha Feedback: (a) directly from the Vane Sensor at the front, and (b) using the Alpha-Estimator. They both have sufficient Margin, but (a) is better. The System also has a Short-Period Mode at 3.84 (rad/sec)

2.5 Lateral Control Design and Analysis

The input file for the coupled Roll and Yaw axes design is "*Later_LQR_Des.Inp*" located in subdirectory "*Control Analysis\LQG\Examples\Missile Control Design\Lateral LQR*". It contains several Flixan datasets that generate plant models and perform steady-state LQR state-feedback control design. They are processed by a batch set located at the top of the file. The batch first retains the control weight matrices Q_c and R_c from being erased from the systems file "*Later_LQR_Des.Qdr*". Then it generates the vehicle model "*Missile with Wing, Mach: 2.5, Qbar: 1220*" that includes both pitch and lateral dynamics. The initial lateral design model is then extracted from the above system and saved as "*Missile with Wing Lateral Design Model*". It consists of two inputs, Aileron and Rudder deflections in (rad), and 5 outputs: roll attitude and rate, yaw attitude and rate and the angle of sideslip in radians. A second longitudinal system is also created with title: "*Missile with Wing Lateral Analysis Model*". It includes wind velocity input in (feet/sec) and some sensor outputs, and it will be used in simulations. The direction of the gust is different than in the pitch model. It is now perpendicular to the vehicle x-axis, and along the –y direction to excite the roll and yaw dynamics, as defined in the vehicle input data by the wind azimuth and elevation angles: 90^{o} and 90^{o}.

```
BATCH MODE INSTRUCTIONS ...............
Batch for Designing Lateral Models and Gains for a Missile with Wing
!
! This batch set creates the Design and Analysis models for a
! Missile with Wing at 2.5 Mach, and performs LQR design.
! The Missile has a fixed Thrust and it is controlled by 3 Aerosurfaces
!
!                  Control Design Matrices
Retain Matrix    : State Weight Matrix Qc (9x9)
Retain Matrix    : Control Weight Matrix Rc (2x2)
!
Flight Vehicle   : Missile with Wing, Mach: 2.5, Qbar: 1220
System Modificat : Missile with Wing Lateral Design Model
System Modificat : Missile with Wing Lateral Analysis Model
Transf-Functions : Actuator: 34/(s+34)
Transf-Functions : Integrator
System Connection: Augmented Lateral Design Model
System Modificat : Augmented Lateral Design Model-2
LQR Control Des  : LQR Control Design for Augmented Lateral Design Model
!
!                  Convert to Matlab
To Matlab Format : Missile with Wing Lateral Analysis Model
To Matlab Format : LQR State-Feedback Control for Augmented Lateral Design Model
----------------------------------------------------------------------------------------------
```

```
FLIGHT VEHICLE INPUT DATA ......
Missile with Wing, Mach: 2.5, Qbar: 1220
! Rigid-Body Missile controlled by 3 aerosurfaces. The engine has fixed thrust
! and does not gimbal
Body Axes Output, Attitude=Euler Angles, NoWind Alpha

Vehicle Mass (lb-sec^2/ft), Gravity Accelerat. (g) (ft/sec^2), Earth Radius (Re) (ft)    :   1219.1, 32.07,
Moments and products of Inertias Ixx, Iyy, Izz, Ixy, Ixz, Iyz, in (lb-sec^2-ft)          :   0.4063E+04 0.1654E+06
CG location with respect to the Vehicle Reference Point, Xcg, Ycg, Zcg, in (feet)        :   26.19, 0.0,  -0.15
Vehicle Mach Number, Velocity Vo (ft/sec), Dynamic Pressure (psf), Altitude (feet)       :   2.5, 2427.4,1220.6,
Inertial Acceleration Vo_dot, Sensed Body Axes Accelerations Ax,Ay,Az (ft/sec^2)         :   60.0, 60.0, 0.0, 10.5
Angles of Attack and Sideslip (deg), alpha, beta rates (deg/sec)                         :   10.5, 0.0, 0.0, 0.0
Vehicle Attitude Euler Angles, Phi_o,Thet_o,Psi_o (deg), Body Rates Po,Qo,Ro (deg/sec)   :   0.0,39.6,0.0, 0.0, 0.132,
Wind Gust Vel wrt Vehi (Azim & Elev) angles (deg), or Force(lb), Torque(ft-lb), locat:xyz:   Gust  90.0  90.0
Surface Reference Area (feet^2), Mean Aerodynamic Chord (ft), Wing Span in (feet)        :   145.4, 22.0, 22.0
Aero Moment Reference Center (Xmrc,Ymrc,Zmrc) Location in (ft), {Partial_rho/ Partial_H} :   26.19, 0.0, -0.238, 0.0
Aero Force Coef/Deriv (1/deg), Along -X, {Cao,Ca_alf,PCa/PV,PCa/Ph,Ca_alfdot,Ca_q,Ca_bet}:   0.1, 0.002, 0.0, 0.0,
Aero Force Coeffic/Derivat (1/deg), Along Y, {Cyo,Cy_bet,Cy_r,Cy_alf,Cy_p,Cy_betdot,Cy_V}:   0.0, -0.023, 0.0, 0.0,
Aero Force Coeff/Deriv (1/deg), Along Z, {Czo,Cz_alf,Cz_q,Cz_bet,PCz/Ph,Cz_alfdot,PCz/PV}:  -0.1, -0.032, 0.0, 0.0,
Aero Moment Coeffic/Derivat (1/deg), Roll: {Clo, Cl_beta, Cl_betdot, Cl_p, Cl_r, Cl_alfa}:   0.0, -0.0017,0.0,-0.243,
Aero Moment Coeff/Deriv (1/deg), Pitch: {Cmo,Cm_alfa,Cm_alfdot,Cm_bet,Cm_q,PCm/PV,PCm/Ph}:  -0.037,-0.011, 0.0,0.0,
Aero Moment Coeffic/Derivat (1/deg), Yaw : {Cno, Cn_beta, Cn_betdot, Cn_p, Cn_r, Cn_alfa}:   0.0, 5.6e-4, 0.0,0.1388,

Number of Control Surfaces, With or No TWD (Tail-Wags-Dog and Hinge Moment Dynamics) ?   :  3   No TWD

Control Surface No:  1                                                                     Elevator
Trim Angle, Max/Min Deflection Angles from Trim, Hinge Line Angles: phi_h, lamda_h  (deg):  0.0  30.0 -30.0 0.000
Surface Mass, Inertia about Hinge, Moment Arm (Hinge to Surface CG), Surface Chord, Area :  0.0  0.0  0.0  0.0
Hinge Moment Derivatives (1/deg), { Chm_Alpha, Chm_Beta, Chm_Delta, Chm_Mach }           :  0.0  0.0  0.0  0.0
Location of the Hinge Line Center with respect to Vehicle Reference (feet), {Xcs,Ycs,Zcs}:  0.0  0.0  0.0
Forces (-x,y,z) due to Deflect. and Rates {Ca_del,Cy_del,Cz_del, Ca_deld,Cy_deld,Cz_deld}:  0.00003 -0.0 -0.0087, 0.00
Moments due to Deflections and Rates {Cl_del,Cm_del,Cn_del,Cl_deldot,Cm_deldot,Cn_deldot}:  0.0 -0.0072 0.0  0.0

Control Surface No:  2                                                                     Aileron
Trim Angle, Max/Min Deflection Angles from Trim, Hinge Line Angles: phi_h, lamda_h  (deg):  0.0 30.0  -30.0 0.0
Surface Mass, Inertia about Hinge, Moment Arm (Hinge to Surface CG), Surface Chord, Area :  0.0   0.0  0.0  0.0
Hinge Moment Derivatives (1/deg), { Chm_Alpha, Chm_Beta, Chm_Delta, Chm_Mach }           :  0.0  0.0   0.0    0.0
Location of the Hinge Line Center with respect to Vehicle Reference (feet), {Xcs,Ycs,Zcs}:  0.0    0.0     0.0
Forces (-x,y,z) due to Deflect. and Rates {Ca_del,Cy_del,Cz_del, Ca_deld,Cy_deld,Cz_deld}:  0.00003 0.0011 0.0  0.0
Moments due to Deflections and Rates {Cl_del,Cm_del,Cn_del,Cl_deldot,Cm_deldot,Cn_deldot}: -6.54e-4 0.0  -0.0014  0.0

Control Surface No:  3                                                                     Rudder
Trim Angle, Max/Min Deflection Angles from Trim, Hinge Line Angles: phi_h, lamda_h  (deg):  0.0 30.0 -30.0 0.000
Surface Mass, Inertia about Hinge, Moment Arm (Hinge to Surface CG), Surface Chord, Area :  0.0 0.0 0.0 0.0  0.0
Hinge Moment Derivatives (1/deg), { Chm_Alpha, Chm_Beta, Chm_Delta, Chm_Mach }           :  0.0 0.0 0.0 0.0
Location of the Hinge Line Center with respect to Vehicle Reference (feet), {Xcs,Ycs,Zcs}:  0.0 0.0 0.0
Forces (-x,y,z) due to Deflect. and Rates {Ca_del,Cy_del,Cz_del, Ca_deld,Cy_deld,Cz_deld}:  0.00001 0.0034 0.0 0.0
Moments due to Deflections and Rates {Cl_del,Cm_del,Cn_del,Cl_deldot,Cm_deldot,Cn_deldot}:  5.9456e-4 0.0 -0.0035

Number of Bending Modes                                                                  :  0
----------------------------------------------------------------------------------------------------------------------
```

```
CREATE A NEW SYSTEM FROM AN OLD SYSTEM... (Titles of the New and Old Systems)
Missile with Wing Lateral Design Model
Missile with Wing, Mach: 2.5, Qbar: 1220
! The initial Lateral Design system is extracted from the coupled RB system above
!
TRUNCATE OR REORDER THE SYSTEM INPUTS, STATES, AND OUTPUTS
Extract Inputs :   2   3
Extract States :   1   2   5   6   8
Extract Outputs:   1   2   5   6   8
----------------------------------------------------------------------------------------------------
```

```
CREATE A NEW SYSTEM FROM AN OLD SYSTEM... (Titles of the New and Old Systems)
Missile with Wing Lateral Analysis Model
Missile with Wing, Mach: 2.5, Qbar: 1220
! The lateral Analysis/ Simulation system is extracted from the coupled RB system above
!
TRUNCATE OR REORDER THE SYSTEM INPUTS, STATES, AND OUTPUTS
Extract Inputs :   2   3   4
Extract States :   1   2   5   6   8
Extract Outputs:   1   2   5   6   8  11  12  15
----------------------------------------------------------------------------------------------------
```

The system modification datasets extract the lateral variables from the coupled system “*Missile with Wing, Mach: 2.5, Qbar: 1220*” and save them in file “*Later_LQR_Des.Qdr*” as separate systems.

In the lateral direction we would like to command and track the heading angle (ξ) which is the vehicle direction in the inertial frame. The heading angle can be controlled by a coordinated roll and yaw command that is achieved with good roll and yaw attitude tracking performance. We introduce, therefore, two additional states in the design model: ϕ-integral and ψ-integral because we want to be able to command roll and yaw independently. A combined roll and yaw command minimizes the β-transients. It is also good idea to include simple first order aileron and rudder actuator models in the synthesis model because it introduces more plant information in the design and makes the control system more efficient with less phase-lag. We introduce therefore two additional states in the state-vector: $\delta_{aileron}$ and δ_{rudder}, a total of 9 states. This will create a (2x9) state-feedback LQR gain matrix.

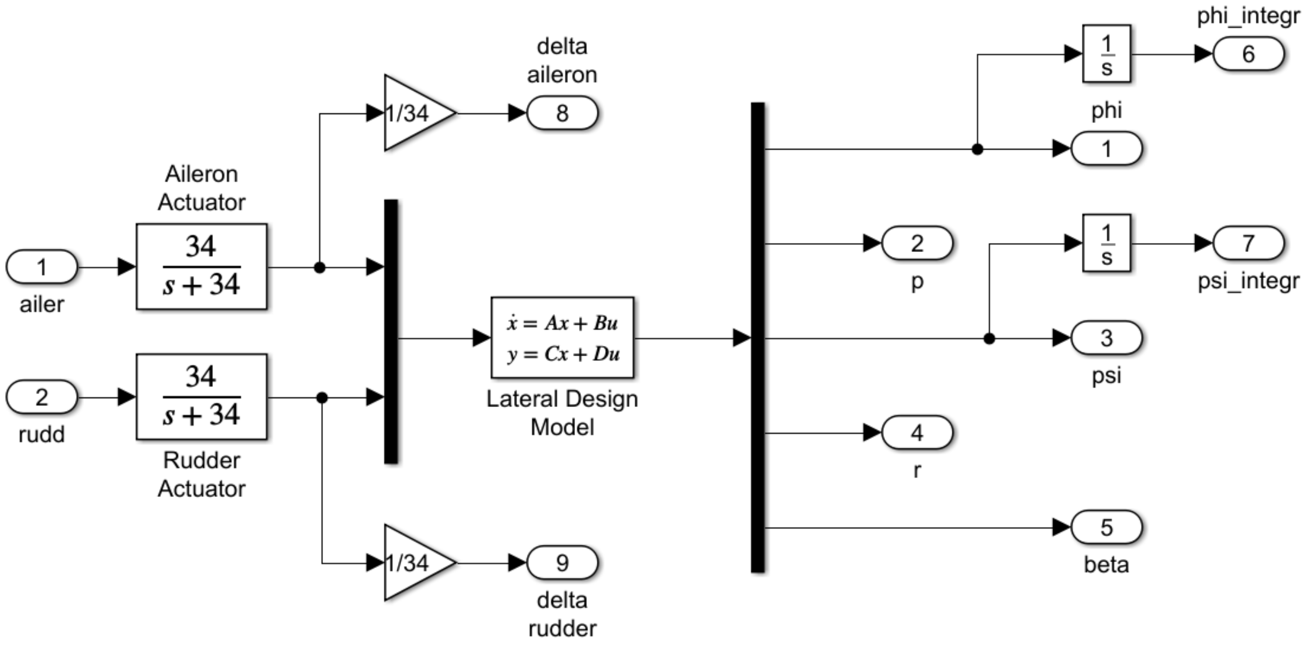


Figure 2.9 Augmented Lateral Design Plant for LQR Control Design

Figure 2.9 shows the augmented plant for the Roll/ Yaw LQR control design. It is implemented in the interconnection dataset below that combines the 5 subsystems together and generates the augmented design plant, which is: "*Augmented Lateral Design Model*". The augmented state vector consists of 9 states: [ϕ, p, ψ, r, β, ϕ-integral, ψ-integral, δ-aileron and δ-rudder]. This system's states, however, are not in the same sequence as the outputs and it is modified to system "*Augmented Lateral Design Model-2*" by reordering the states which conveniently makes the C matrix equal to the identity I_9.

```
INTERCONNECTION OF SYSTEMS .....
Augmented Lateral Design Model
! Create a 9-State Augmented Model that Includes Phi-integr, Psi-integr,
! Aileron and Rudder deflections in the state vector for Lateral Control Design
!
Titles of Systems to be Combined
Title 1 Actuator: 34/(s+34)
Title 2 Actuator: 34/(s+34)
Title 3 Missile with Wing Lateral Design Model
Title 4 Integrator
Title 5 Integrator
SYSTEM INPUTS TO SUBSYSTEM  1                                                          to Ailern Actuator
System Input  1 to Subsystem  1, Input  1, Gain= 1.0                                   Delta-ailer Command
..............................................................................
SYSTEM INPUTS TO SUBSYSTEM  2                                                          to Rudder Actuator
System Input  2 to Subsystem  2, Input  1, Gain= 1.0                                   Delta-ruddr Command
..............................................................................
SYSTEM OUTPUTS FROM SUBSYSTEM  3                                                       Vehicle Plant
System Output  1 from Subsystem  3, Output  1, Gain= 1.0                               Phi
System Output  2 from Subsystem  3, Output  2, Gain= 1.0                               p - roll rate
System Output  3 from Subsystem  3, Output  3, Gain= 1.0                               Psi
System Output  4 from Subsystem  3, Output  4, Gain= 1.0                               r - yaw rate
System Output  5 from Subsystem  3, Output  5, Gain= 1.0                               Beta
..............................................................................
SYSTEM OUTPUTS FROM SUBSYSTEM  4                                                       Integrator
System Output  6 from Subsystem  4, Output  1, Gain= 1.0                               Phi-integral
..............................................................................
SYSTEM OUTPUTS FROM SUBSYSTEM  5                                                       Integrator
System Output  7 from Subsystem  5, Output  1, Gain= 1.0                               Psi-integral
..............................................................................
SYSTEM OUTPUTS FROM SUBSYSTEM  1                                                       Actuator
System Output  8 from Subsystem  1, Output  1, Gain= 0.0294118                         delta-aileron
..............................................................................
SYSTEM OUTPUTS FROM SUBSYSTEM  2                                                       Actuator
System Output  9 from Subsystem  2, Output  1, Gain= 0.0294118                         delta-rudder
..............................................................................
SUBSYSTEM NO  1 GOES TO SUBSYSTEM NO  3                                                Ailer-Actuat to Vehicle
Subsystem  1, Output  1 to Subsystem  3, Input  1, Gain=   1.0000                      Aileron-deflect
......................................................................
SUBSYSTEM NO  2 GOES TO SUBSYSTEM NO  3                                                Ruddr-Actuat to Vehicle
Subsystem  2, Output  1 to Subsystem  3, Input  2, Gain=   1.0000                      Rudder-deflect
......................................................................
SUBSYSTEM NO  3 GOES TO SUBSYSTEM NO  4                                                Vehicle to Integrator-4
Subsystem  3, Output  1 to Subsystem  4, Input  1, Gain=   1.0000                      Phi
......................................................................
SUBSYSTEM NO  3 GOES TO SUBSYSTEM NO  5                                                Vehicle to Integrator-4
Subsystem  3, Output  3 to Subsystem  5, Input  1, Gain=   1.0000                      Psi
......................................................................
Definitions of Inputs  =   2
Aileron Deflection Command (delta) rad
Rudder  Deflection Command (delta) rad

Definitions of Outputs =   9
Roll Attitude, Phi (rad)
Roll Rate, p (rad/sec)
Yaw  Attitude, Psi (rad)
Yaw  Rate, r (rad/sec)
Angle of Sideslip beta (rad)
Phi-Integral (rad-sec)
Psi-Integral (rad-sec)
Aileron Deflection, delta-ailer (rad)
Rudder Deflection, delta-rudder (rad)
-------------------------------------------------------------------------------
```

This interconnection Dataset creates the augmented synthesis model of Figure 2.9

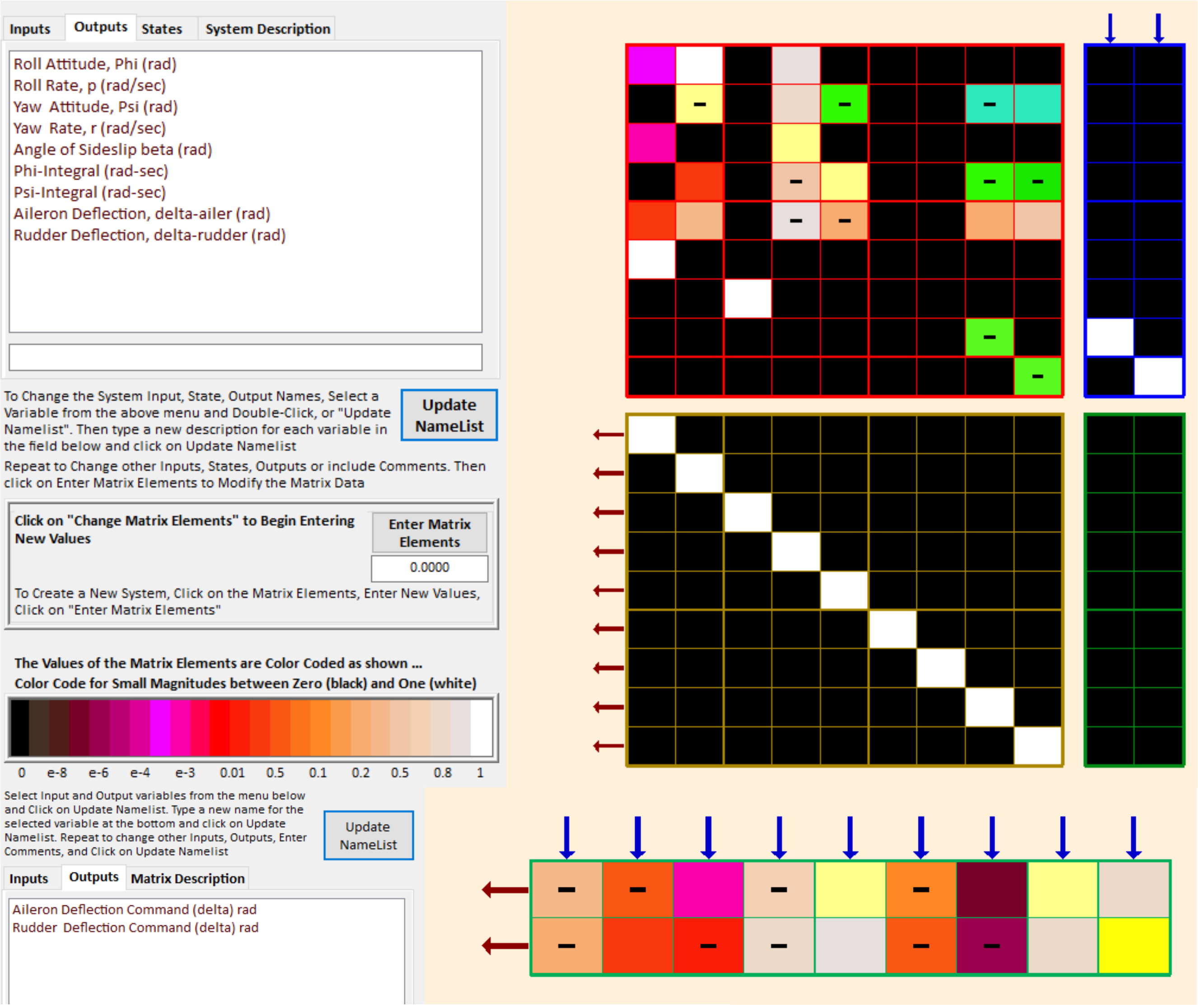
```
SYSTEM OF TRANSFER FUNCTIONS ...
Actuator: 34/(s+34)
! First order Actuator 34 (rad/sec) Bandwidth
Continuous
TF. Block #  1 34/(s+34)                                         Order of Numer, Denom=  0  1
Numer 0.0         34.0
Denom 1.0         34.0
-------------------------------------------------------------------------------
Block #, from Input #, Gain
  1  1    1.00000
...........................
Outpt #, from Block #, Gain
  1  1    1.00000
...........................
Definitions of Inputs  =   1
Delta Command

Definitions of Outputs =   1
Delta Out
-------------------------------------------------------------------------------

SYSTEM OF TRANSFER FUNCTIONS ...
Integrator
Continuous
TF. Block #  1  (1/s)                                           Order of Numer, Denom=  0  1
Numer 0.0         1.0
Denom 1.0         0.0
-------------------------------------------------------------------------------
Block #, from Input #, Gain
  1  1    1.00000
...........................
Outpt #, from Block #, Gain
  1  1    1.00000
-------------------------------------------------------------------------------

CREATE A NEW SYSTEM FROM AN OLD SYSTEM... (Titles of the New and Old Systems)
Augmented Lateral Design Model-2
Augmented Lateral Design Model
! Rearange the Order of States to be the same as the Outputs
! Makes C=Identity
TRUNCATE OR REORDER THE SYSTEM INPUTS, STATES, AND OUTPUTS
Extract States :   3   4   5   6   7   8   9   1   2
Extract Outputs:   1   2   3   4   5   6   7   8   9
-------------------------------------------------------------------------------------------
```

The dataset “*LQR Control Design for Augmented Lateral Design Model*” below performs the LQR control synthesis using the design plant “*Augmented Lateral Design Model-2*”. It uses the C matrix for output criteria which is Identity and the (9x9) weight matrix Q_c penalizes the individual states. The (2x2) matrix R_c penalizes the two controls, which are: aileron and rudder activity. The weight matrices are already set in the systems file. The LQR program generates the (2x9) state-feedback matrix K_{pr} that stabilizes the plant by closing the control loop between the state-vector and the two aerosurface inputs. The gain matrix is also saved in the systems file under the title “*LQR State-Feedback Control for Augmented Lateral Design Model*”. The matrix K_{pr} and the lateral analysis model are also saved in Matlab format as “Kpr.mat” and “vehi_lateral.m” respectively for further analysis.

```
LINEAR QUADRATIC REGULATOR STATE-FEEDBACK CONTROL DESIGN
LQR Control Design for Augmented Lateral Design Model
Plant Model Used to Design the Control System:  Augmented Lateral Design Model-2
Criteria Optimization Output is Matrix C
State Penalty Weight (Qc) is Matrix:   Qc9      State Weight Matrix Qc (9x9)
Control Penalty Weight (Rc) is Matrix: Rc2      Control Weight Matrix Rc (2x2)
Continuous LQR Solution Using Laub Method
LQR State-Feedback Control Gain Matrix Kpr      LQR State-Feedback Control for Augmented Lateral Design Model
------------------------------------------------------------------------------------------
CONVERT TO MATLAB FORMAT ........      (Title, System/Matrix, m-filename)
LQR State-Feedback Control for Augmented Lateral Design Model
Matrix Kpr
------------------------------------------------------------------------------------------------
CONVERT TO MATLAB FORMAT ........      (Title, System/Matrix, m-filename)
Missile with Wing Lateral Analysis Model
System
Vehi_lateral.m
------------------------------------------------------------------------------------------
END-END-END-END-END-END-END-END-END-END-END-END-END-END-END-END-END-END-END-END-END-END-END
```

2.6 Lateral Simulations

There are two simulation models is in folder “*Examples\Missile with Wing\ (b) LQR Design\ Lateral LQR*”. They both use state-feedback from the 9 plant states via matrix K_{pr} which includes: ϕ-integral, ψ-integral, $\delta_{aileron}$ and δ_{rudder} feedback in addition to the feedback from the original vehicle states: (ϕ, p, ψ, r, β). They will be used to analyze the system’s response to gusts and to direction change commands. The heading direction is $\xi=\psi+\beta_{inert}$. Since beta cannot be measured directly, we will examine again two options: a beta feedback from the vane sensor located at the nose of the missile, in simulation file: “*Lateral _Sim1.mdl*” shown in Figure 2.10a, and feedback from estimated beta created from the accelerometer, aileron and rudder signals, which is implemented in simulation file “*Lateral _Sim2.mdl*” shown in Figure 2.10b. The heading error is converted through a PI filter to combined roll and yaw attitude command which makes the vehicle to perform a coordinated roll/ yaw turn. Yaw attitude command is needed in addition to the roll command to reduce beta and subsequently lateral loading. This is because a 10^o alpha is considerably high and a pure roll would induce a significant sideslip. A coordinated roll and yaw command reduces beta and hence the lateral loading. Both simulations produce similar results.



Roll Attitude, Phi (rad)
Roll Rate, p (rad/sec)
Yaw Attitude, Psi (rad)
Yaw Rate, r (rad/sec)
Angle of Sideslip beta (rad)
Phi-Integral (rad-sec)
Psi-Integral (rad-sec)
Aileron Deflection, delta-ailer (rad)
Rudder Deflection, delta-rudder (rad)

To Change the System Input, State, Output Names, Select a Variable from the above menu and Double-Click, or "Update Namelist". Then type a new description for each variable in the field below and click on Update Namelist

Repeat to Change other Inputs, States, Outputs or include Comments. Then click on Enter Matrix Elements to Modify the Matrix Data

Click on "Change Matrix Elements" to Begin Entering New Values

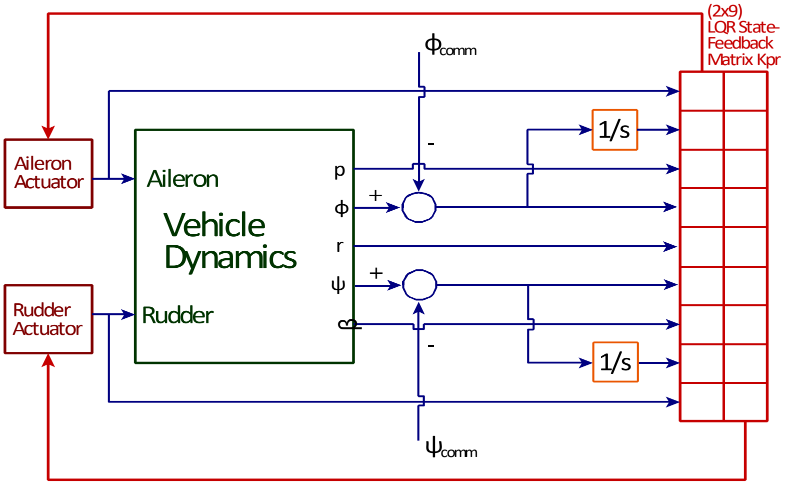
To Create a New System, Click on the Matrix Elements, Enter New Values, Click on "Enter Matrix Elements"

The Values of the Matrix Elements are Color Coded as shown ...

Color Code for Small Magnitudes between Zero (black) and One (white)

Select Input and Output variables from the menu below and Click on Update Namelist. Type a new name for the selected variable at the bottom and click on Update Namelist. Repeat to change other Inputs, Outputs, Enter Comments, and Click on Update Namelist

Aileron Deflection Command (delta) rad
Rudder Deflection Command (delta) rad



Lateral State-Feedback Control System

The Augmented Lateral Design Model-2 in State-Space form, and the State-Feedback Gain Matrix Kpr. Notice that matrix C is the Identity and D=0.

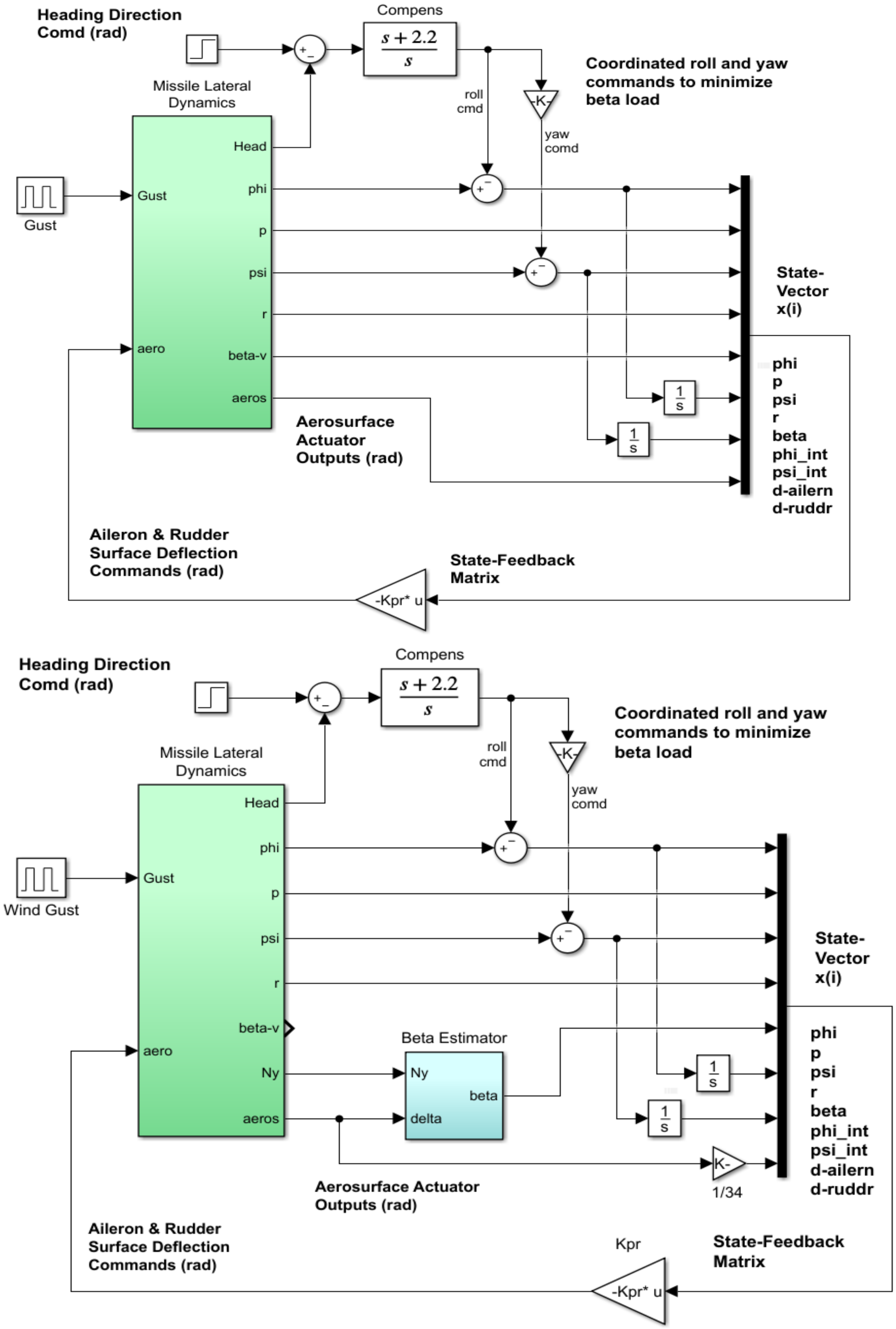


Figure 2.10 Roll and Yaw Axes Closed-Loop Simulation Models “Lateral_Sim1.mdl” and “Lateral_Sim2.mdl”

The vehicle model in Figure 2.11 includes the Flixan generated system "*Missile with Wing Pitch Analysis Model*" from file "*vehi_lateral.m*", two second order actuators for the Aileron and Rudder, a wind-gust filter, and a computational delay in the heading angle output.

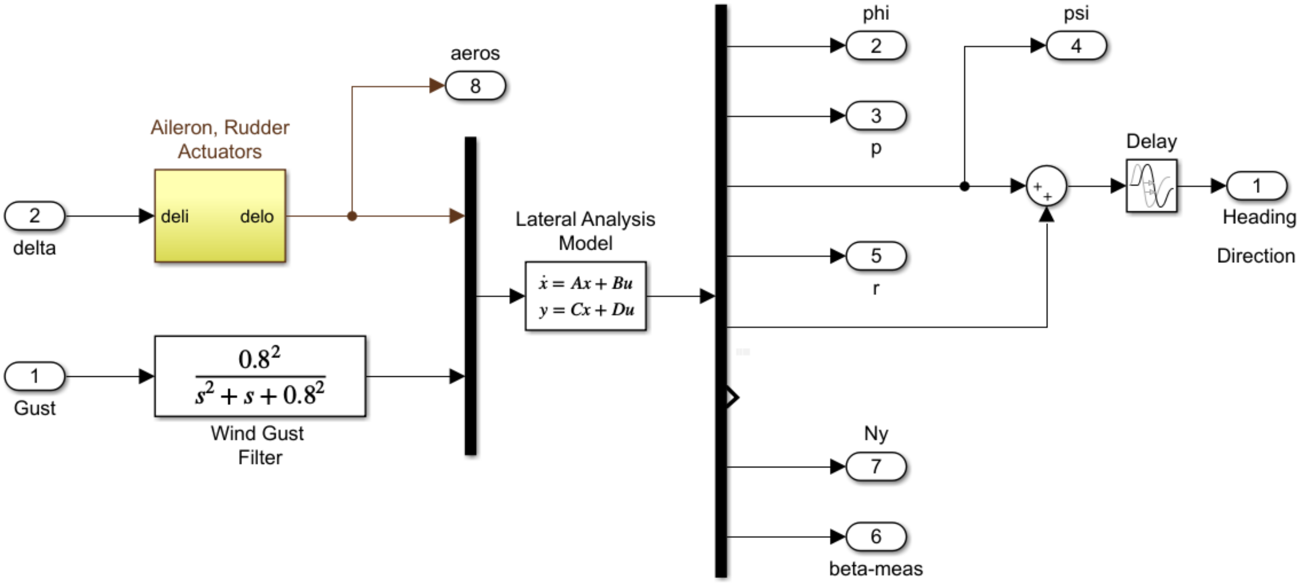


Figure 2.11 Roll and Yaw Dynamics Subsystem

The beta-estimator is used in "*Lateral _Sim2.mdl*" to estimate the sideslip β relative to the wind. It is shown in detail in Fig. 2.12 and it basically solves Equation 2.2. It requires knowledge of the vehicle mass, dynamic pressure, reference area, and aero coefficients. A low-pass filter is included to improve stability. The quality of the estimate obviously depends on the knowledge of those parameters.

$$M_V N_Y = \overline{Q} S_{ref} \left(C_{Y\beta}\, \beta + C_{Y\delta rudd}\, \delta_{rudd} + C_{Y\delta ail}\, \delta_{ailer} \right) \qquad (2.2)$$

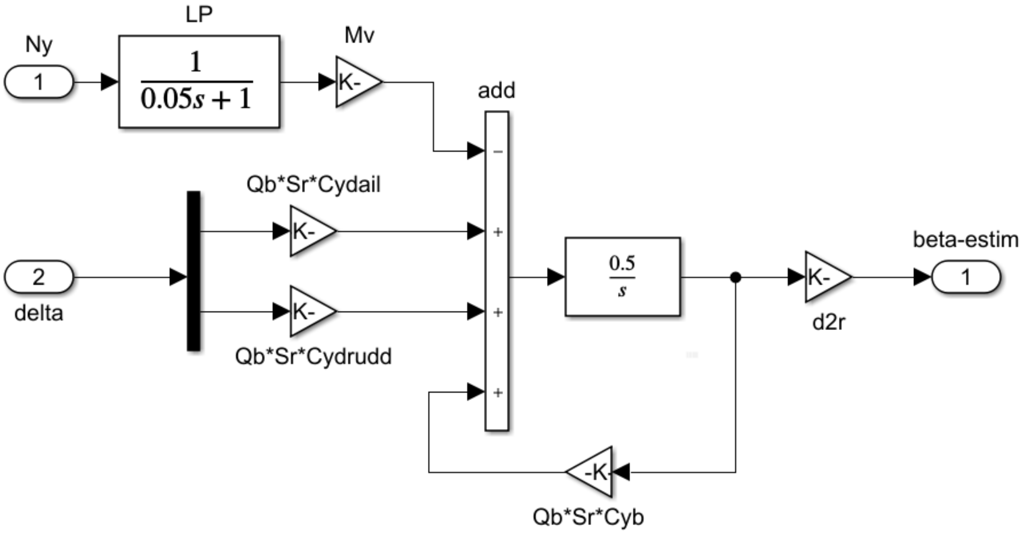


Figure 2.12 Angle of Sideslip Estimator is processing signals from the Ny Accelerometer and the Actuator Deflections. Includes a Low-Pass Filter

In Figure 2.13 the vehicle is commanded to perform a 10^{o} change in heading direction ξ which is achieved by a coordinated roll and yaw command to minimize the β-transients which are undesirable at high Q_{bar}. Fig. 2.13a shows the three betas: inertial, measured, and estimated. Although there is no wind, the measured beta is slightly different from the inertial because the air-data sensor is located ahead of the CG and it is picking up some of the rotational effects.

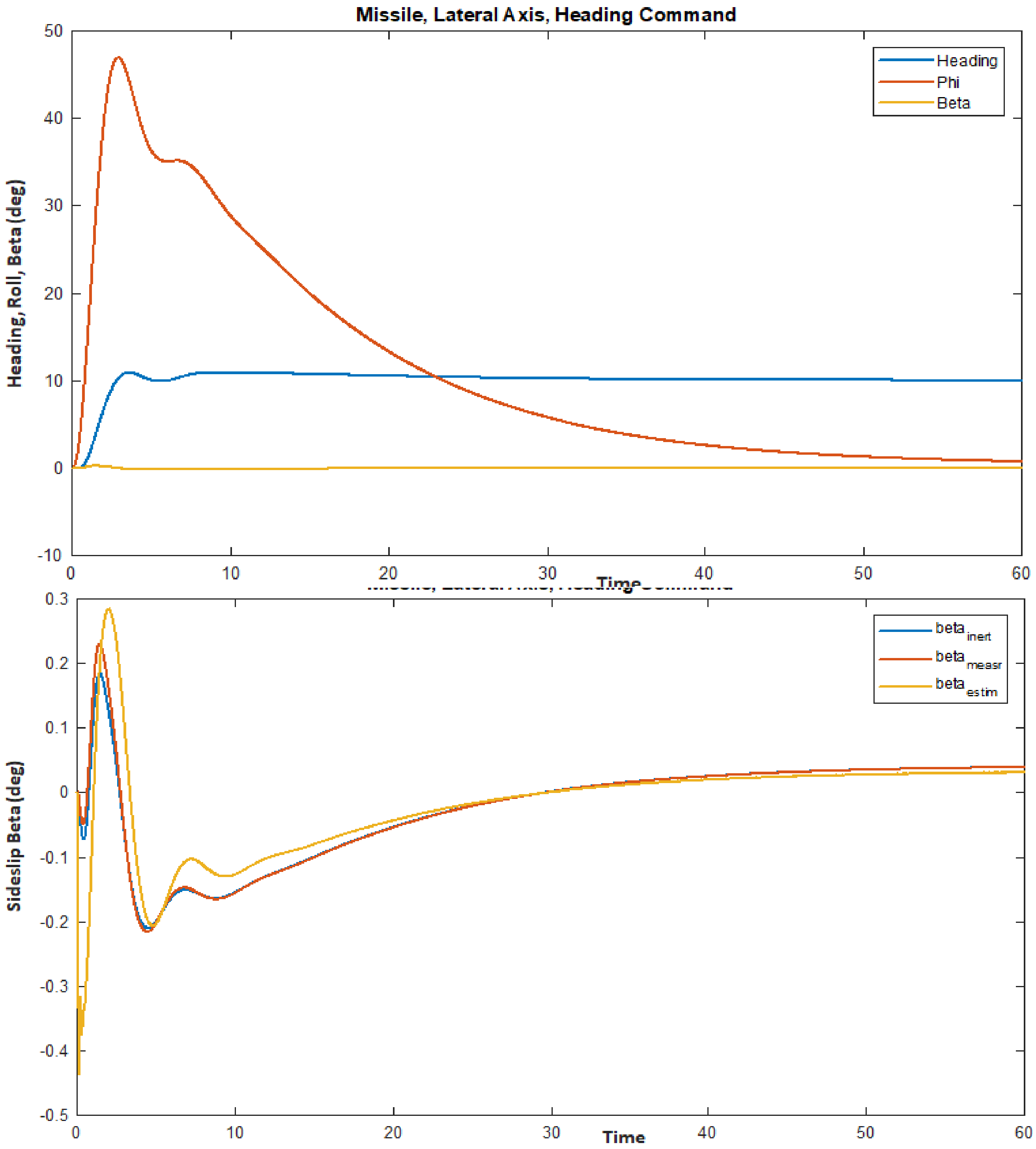


Figure 2.13a The 10 (deg) Change in Heading Direction is achieved by a significant amount of Roll. The three Betas are slightly different from each other. The estimated beta has an initial error.

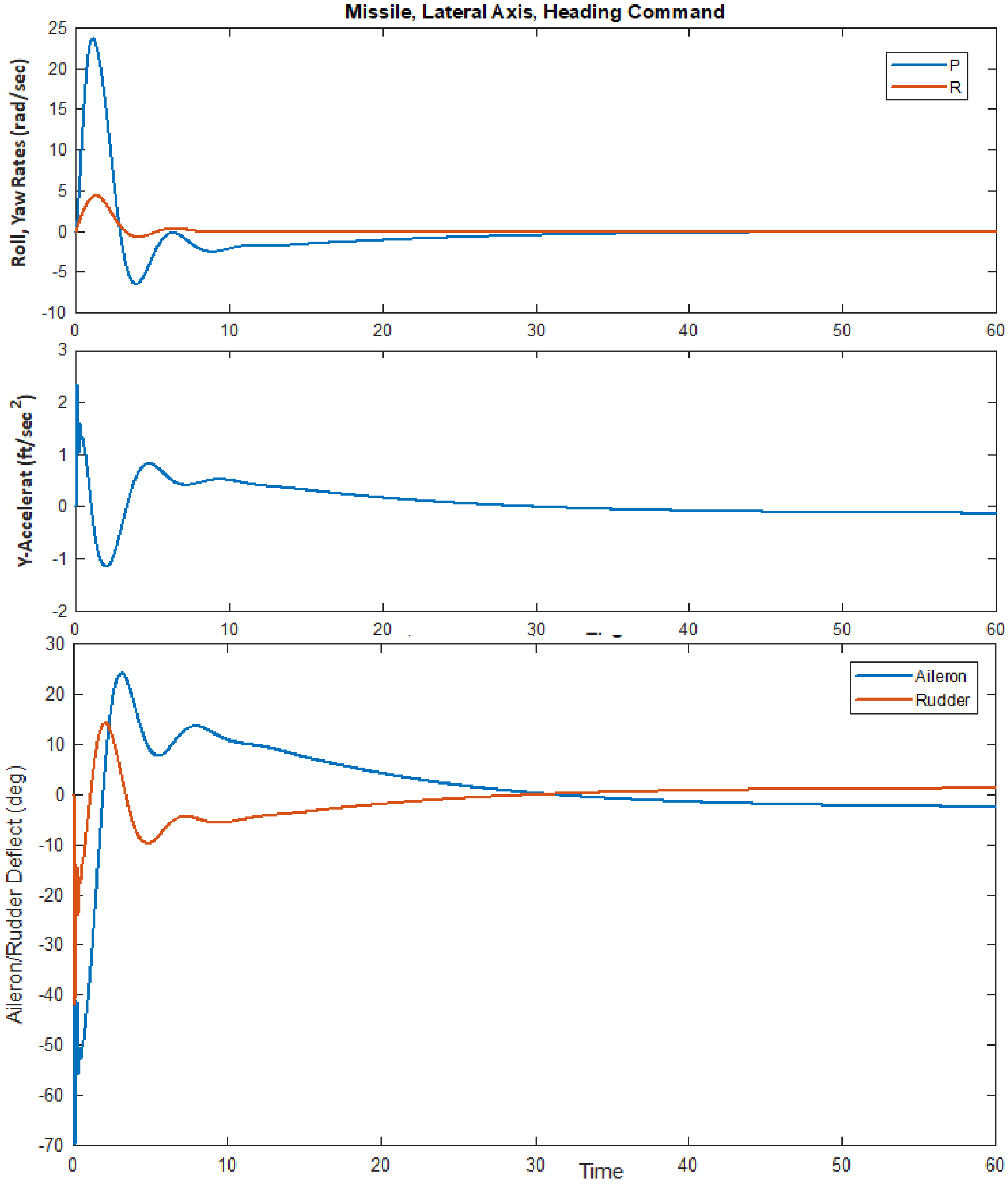


Figure 2.13b Both Aileron and Rudder Respond to the Heading Command and the Roll and Yaw Rates are coordinated to Minimize Beta Transients

In Figure 2.14 the missile is excited by a lateral wind-gust impulse of 30 (feet/sec) velocity for 2 seconds, along the –y direction, which causes a transient in attitude and beta. The vehicle responds by deflecting the aerosurfaces.

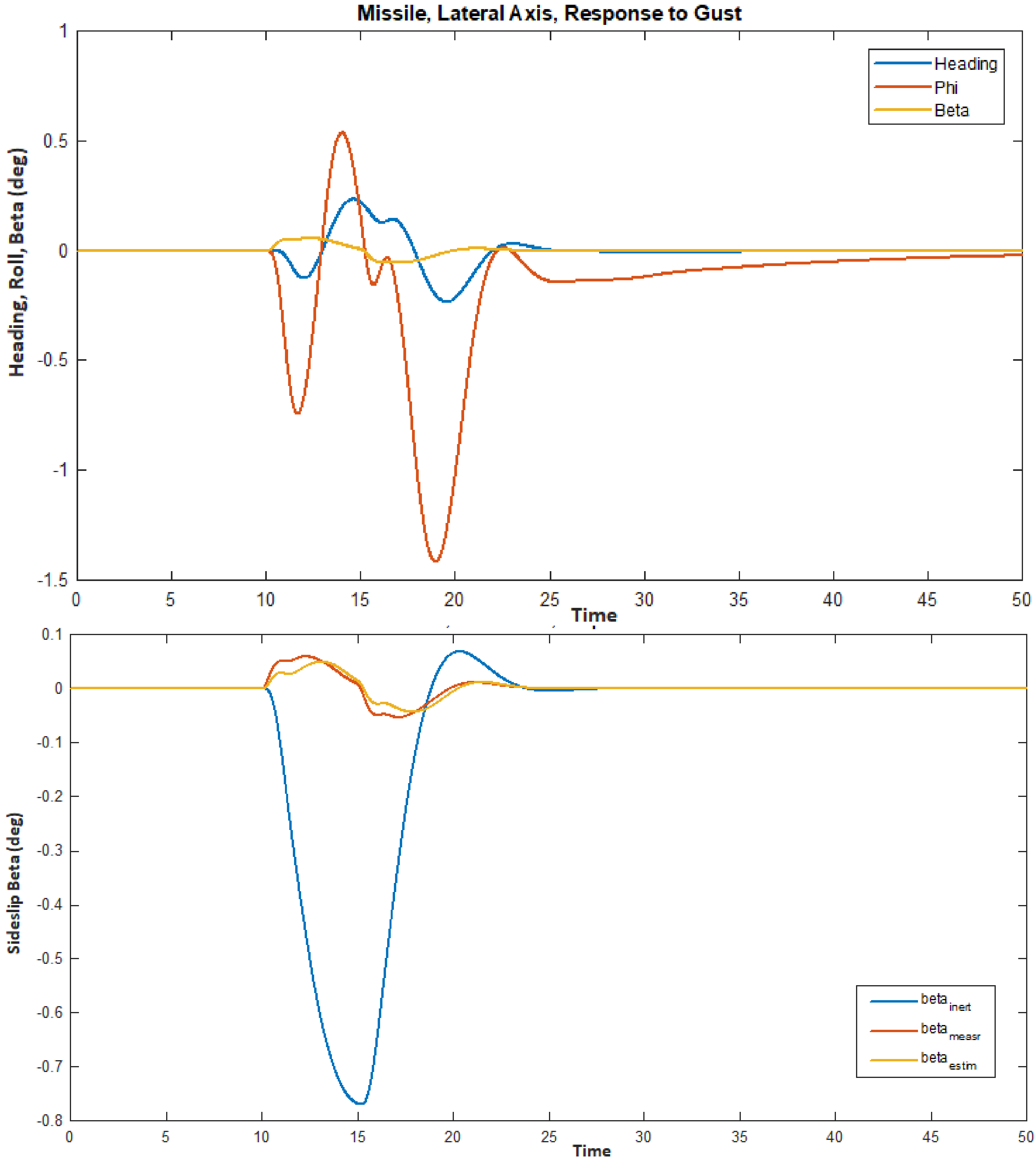


Figure 2.14a The Measure and Estimated Betas are similar and Relative to the Wind. The Inertial Beta is Negative because the Vehicle is pushed by the Wind to the left (-Y). Beta is defined as Inertial in the Vehicle data by the (NoWind Alpha/ Beta) flag

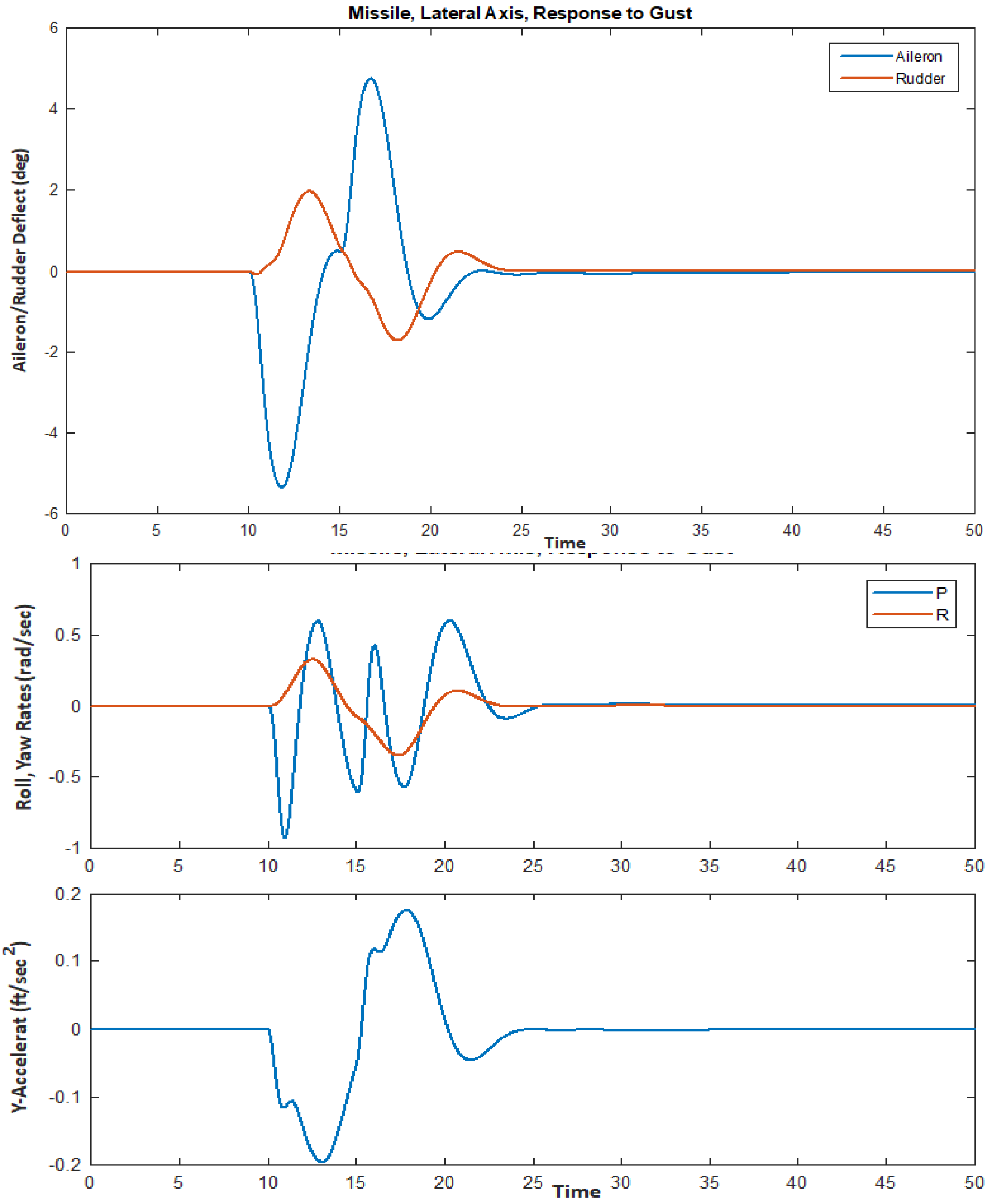
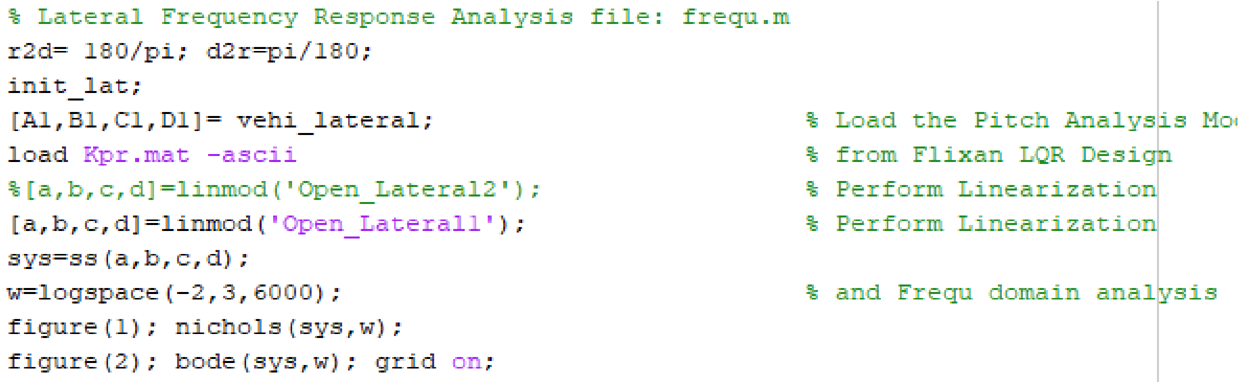


Figure 2.14b Missile is excited by Lateral Wind-Gust from the right side, along –Y, causing –Y Acceleration which also causes Negative Roll and Positive Yaw due to the Vertical Stabilizer. The aileron and Rudder accordingly respond to counteract the Roll and Yaw Rates

2.7 Roll/ Yaw Stability Analysis

The system stability of both designs is analyzed in the frequency domain by calculating the Nichols plots from the open-loop Simulink models “*Open_Lateral1.mdl*” and “*Open_Lateral2.mdl*”, shown in Figure 2.15 and measuring the phase and gain margins. Since the control system in this case consists of two control loops, one loop must be opened and the other loop closed. The script file “frequ.m” calculates the frequency response across the opened loop. The models include the two actuators and the loop is broken between one of the state-feedback outputs and the corresponding actuator input. In Fig. 2.15 the top model is “*Open_Lateral1.mdl*” and the aileron loop is shown opened to analyze roll stability. In the bottom model which is “*Open_Lateral2.mdl*” and the rudder loop is shown opened to analyze yaw stability. The models are modified to check the other axis stability.



```
% Lateral Frequency Response Analysis file: frequ.m
r2d= 180/pi; d2r=pi/180;
init_lat;
[A1,B1,C1,D1]= vehi_lateral;                          % Load the Pitch Analysis Mo
load Kpr.mat -ascii                                   % from Flixan LQR Design
%[a,b,c,d]=linmod('Open_Lateral2');                   % Perform Linearization
[a,b,c,d]=linmod('Open_Lateral1');                    % Perform Linearization
sys=ss(a,b,c,d);
w=logspace(-2,3,6000);                                % and Frequ domain analysis
figure(1); nichols(sys,w);
figure(2); bode(sys,w); grid on;
```

Figure 2.12 shows the LQR control system’s stability in the Roll and Yaw axes calculated from the second model with the beta estimator. The phase and gain margins demonstrate the capability of the LQR method to provide good closed-loop system stability, even though we lost some fidelity in the beta estimation.

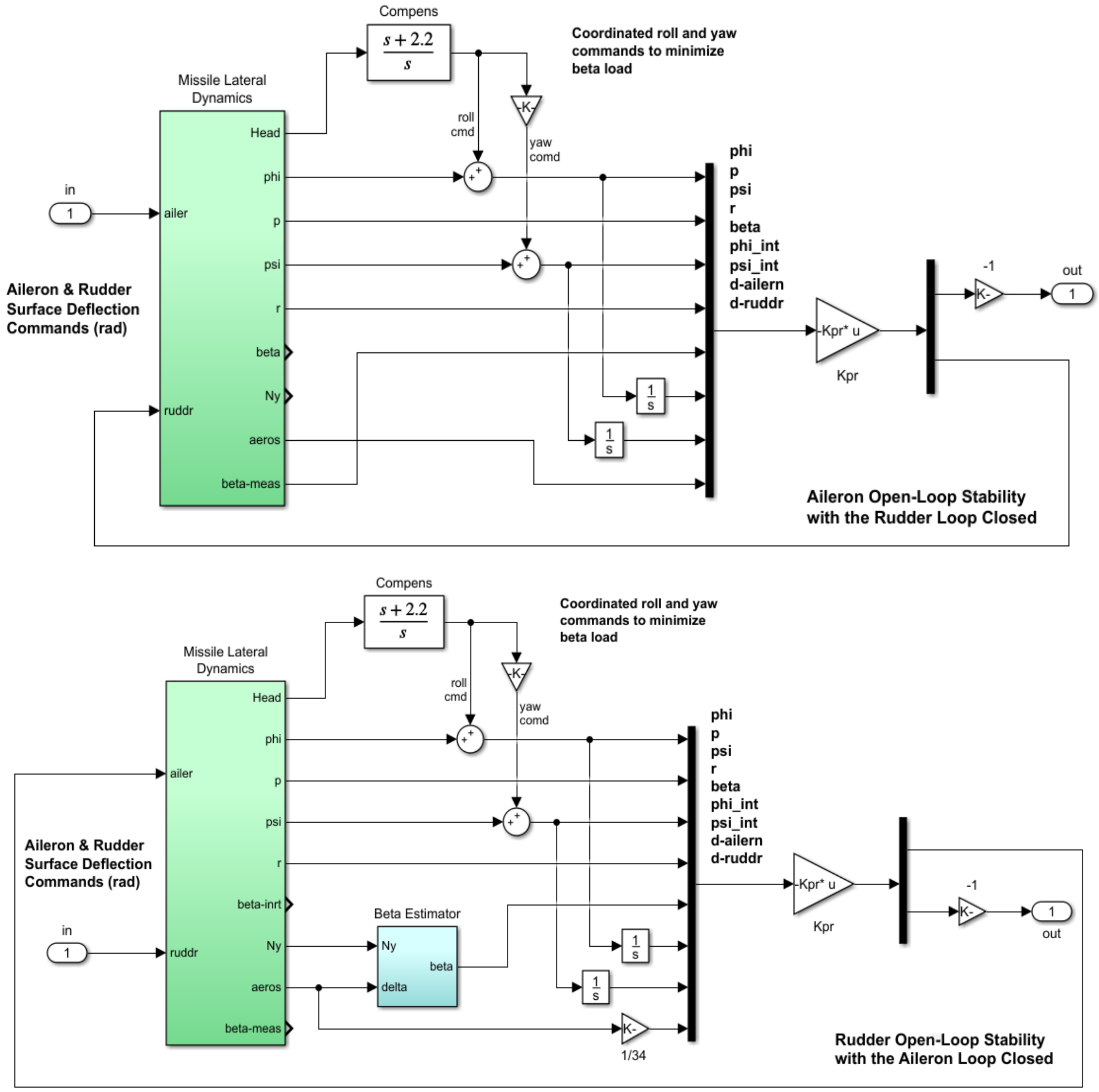


Figure 2.15 Open-Loop Models "Open_Lateral1.mdl" and "Open_Lateral2.mdl" used for Roll and Yaw Stability Analysis

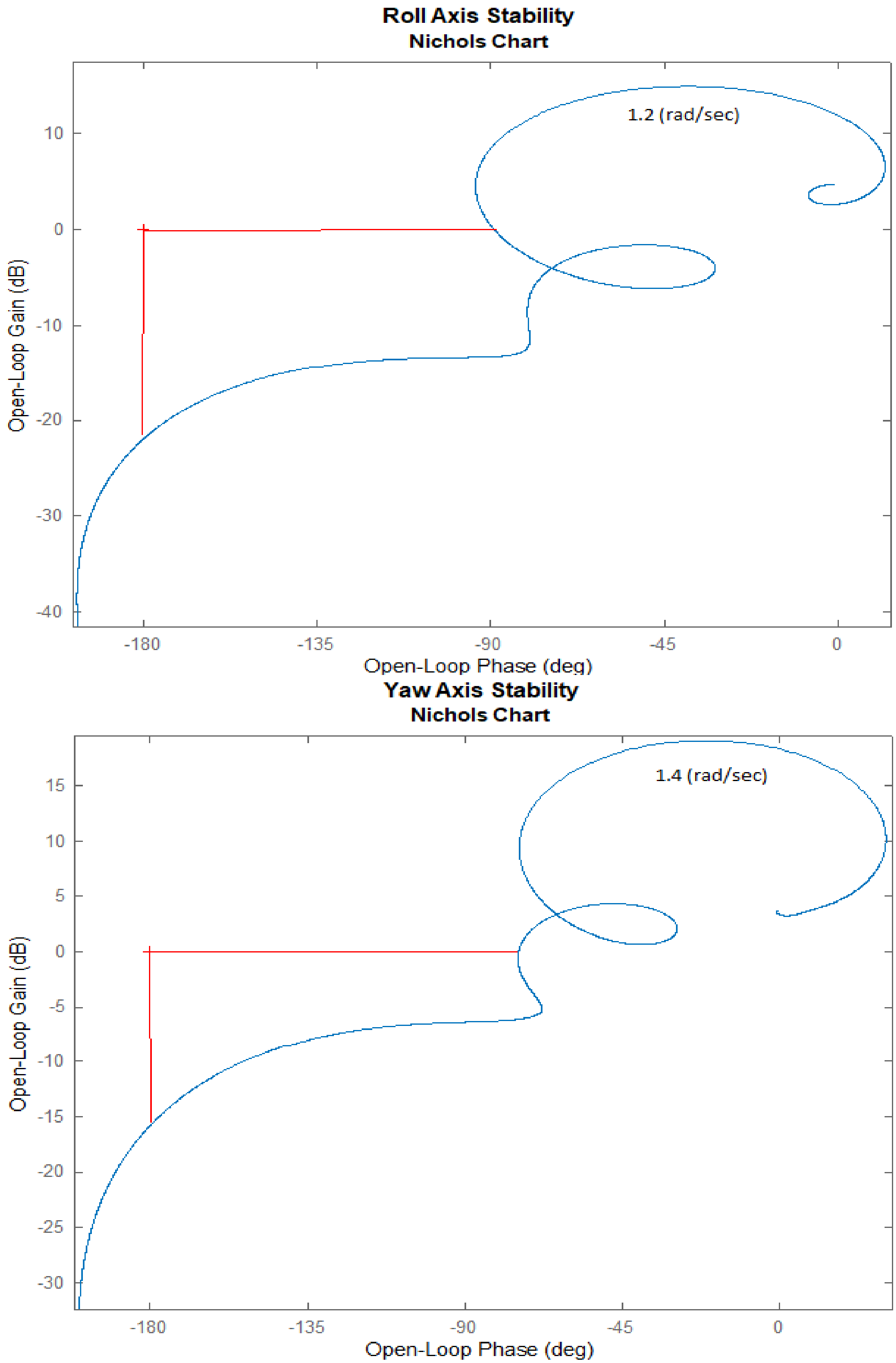


Figure 2.16 Nichols Plots Showing Stability of the LQR Control System in the Roll and Yaw Axes. The System also has Short-Period Modes at 1.2 and 1.4 (rad/sec)

3. Robustness Analysis

A control system is robust when it can tolerate a certain amount of uncertainty in vehicle parameters before it becomes unstable. Parameter uncertainties can be seen as imprecise knowledge of the plant model parameters, such as: the mass properties, moments of inertia, aerodynamic coefficients, center of gravity, etc. The uncertainties of the flight vehicle are specified in terms of variations in the actual plant parameters, above or below their nominal values. The Flixan program generates the uncertain vehicle model by creating an additional input/ output pair for every parameter uncertainty in the model. Each parameter variation is "pulled out" of the plant model and it is placed inside a diagonal block Δ that contains only the uncertainties, while the remaining plant is assumed to be known (best guess). The uncertainty block Δ is attached to the known plant M(s) by means of (n) input/ output "wires", where (n) is the number of the uncertainties, as shown in Figure (3.1). In essence we create (n) additional inputs and outputs in the plant that connect to the uncertainties block Δ, which is a block diagonal matrix Δ= diag(δ_1,δ_2,δ_3,...δ_n). The individual perturbation elements δi may be scalars or matrices and each represents a real uncertainty in the plant. They may be aerodynamic coefficient variations from nominal values, moment of inertia variations, thrust variations, etc. The magnitude of each element represents the maximum possible variation of the parameter above or below its nominal value. M(s) represents the known dynamics consisting of the plant model with the control system included in closed-loop form.

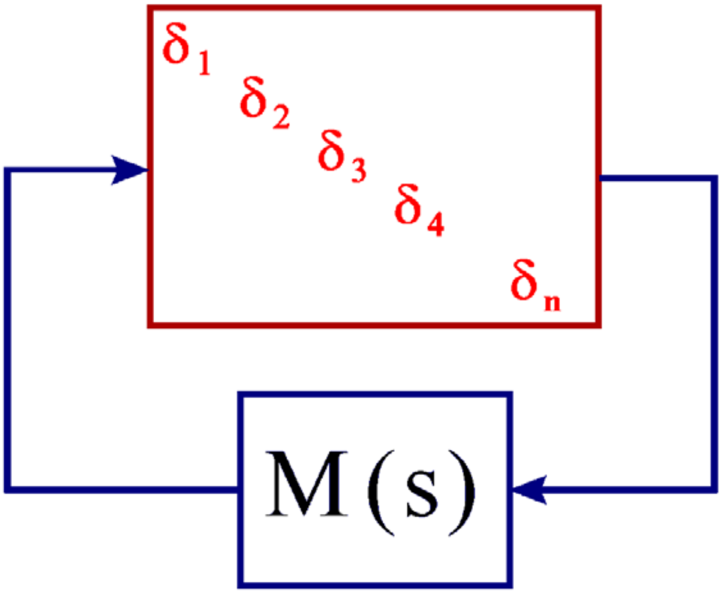


Figure 3.1 Robustness analysis model, where M(s) is the Nominal Closed-Loop system connected with n uncertainties

The closed-loop system M(s) is used to perform robustness analysis using μ-methods. The system in this configuration is considered to be robust, assuming that M(s) is stable, if it remains stable despite all possible variations in the Δ block as long as the magnitude of the individual variations are bounded below $\delta_{(i)}$. The structured singular value (μ) is used for analyzing the system robustness in the frequency domain. To make things easier the diagonal block Δ is normalized so that each individual element is allowed to vary between +1 and -1. We do this by normalizing the plant M(s), by scaling its I/O elements as needed to connect with the normalized Δ whose elements are now bounded to within (±1). The value of $1/\mu(M)$ represents the magnitude of the smallest perturbation that will destabilize the normalized system M(s). According to the small gain theorem, the closed-loop system is robust as long as $\mu(M)$ across the normalized block Δ is less than one at all frequencies.

Robust Performance

Now, in addition to being robust against parameter variations, a system is said to have a "Robust Performance" when it can also satisfies nominal performance requirements in addition to robustness requirements in the presence of parameter variations.

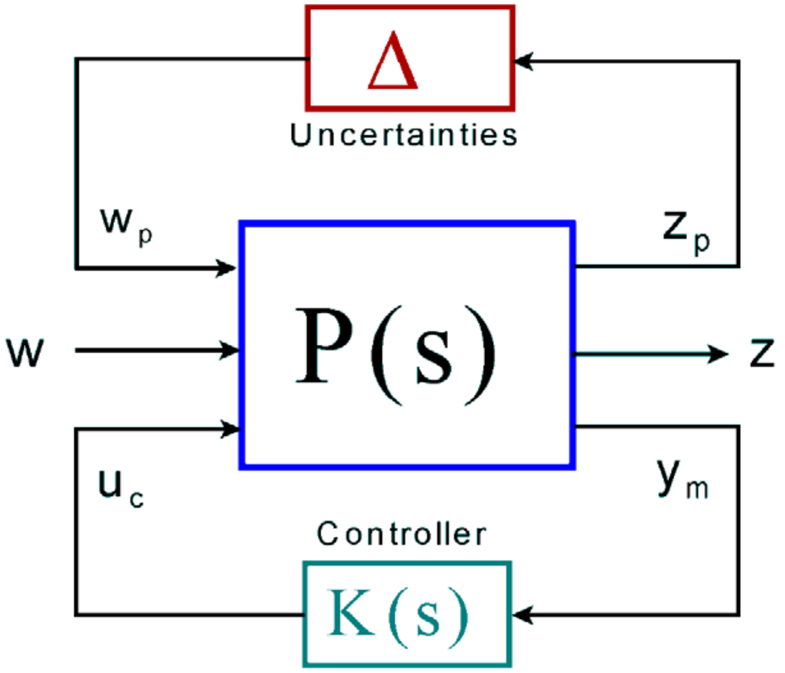


Figure 3.2 Robust Performance Analysis Model

Consider the closed-loop system configuration in Figure (3.2), where the normalized parameter uncertainties block Δ is pulled out of the plant as it was described earlier. P(s) is the open-loop plant dynamics and K(s) is the control system wrapped around the plant, and we assume that the closed-loop system without the block Δ is stable. The plant is connected to the uncertainty block Δ by means of the inputs and outputs w_p and z_p. It is also connected to the controller K(s) by means of the inputs and outputs u_c and y_m. The input (w) is a disturbance input, such as, a wind-gust velocity. The output (z) is a performance criterion that should be kept within certain limits. For example, an angle of attack or sideslip, which should not be allowed to exceed a certain value in order to minimize structural loads. According to the previous definition the system is robust when the μ of the transfer function between (w_p and z_p) is less than 1 at all frequencies. Nominal performance, that is, performance without uncertainties means that the wind disturbance (w) does not violate the max (α) criterion in output (z). This happens when the μ of the transfer function between the disturbance inputs (w) and the performance criteria (z) is less than 1 at all frequencies. The transfer path of P(s) between (w) and (z) is assumed to be normalized to unity. For example, if we know that the maximum disturbance (w) is let's say 30 (ft/sec) we include a gain of 30 at the plant input (w), and if we know that the maximum acceptable angle of attack is, let's say 4 (deg), we include a gain of (1/4) at the plant output (z). After including the scaling gains within the plant we specify the performance requirement, which is: the magnitude of the scaled P(s) between (w) and (z) should be less than 1 at all frequencies. When the plant P(s) satisfies this requirement then we can say that we have achieved "Nominal Performance". We can go one step further and define "Robust Performance". This happens when we satisfy both: robustness and performance simultaneously, meaning that not only the uncertainties will not drive the system to instability but at the same time they will not violate the performance requirement between w and z. This happens, when the μ of the transfer function between the combined input vector [wp, w] and the output vector [zp, z] of the scaled plant P(s) is less than 1 at all frequencies.

Creating Robustness Analysis Models

Now that we have covered the background information for robustness analysis our next challenge is to generate the missile robustness analysis plants P(s) for both pitch and lateral and to analyze them separately. We are going to define 24 parameter uncertainties for the combined pitch and lateral coupled model, 8 of them are longitudinal and 16 are lateral. The coupled vehicle model with uncertainties will be separated in two systems: a pitch system with 8 uncertainty I/Os, and a lateral system with 16 uncertainty I/Os.

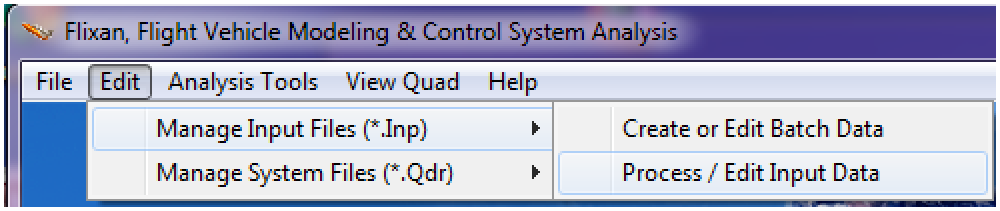
3.1 Pitch System Robustness Analysis

The longitudinal system robustness analysis is performed in directory "*C:\Flixan\ Examples\ Missile with Wing\(c) Robustness Analysis\Pitch Robust*". The missile input data is in file "*WM_Pitch_Robust.Inp*". The title of the vehicle model is "*Missile with Wing with 24 Uncertainties*". The nominal vehicle parameters are the same as the parameters used for classical analysis in section 1. At the bottom of this vehicle data-set, however, there is a call to a set of parameter uncertainties data-set: "*Missile Uncertainties at Mach=2.5, Alpha=10*" which is included right below the nominal vehicle data. The missile uncertainties data specifies variations from nominal in the aero coefficients, the moments of inertia, the vehicle X-cg location, and the control surface aero coefficients. The Flixan Vehicle Modeling program processes the vehicle data and the parameter variations data and saves the coupled vehicle state-space system in file "*WM_Pitch_Robust.Qdr*". This plant model is basically the same as the nominal plant created in Section 1, but it has 24 additional inputs and outputs. An I/O pair is created for every uncertainty. Each I/O pair is scaled as we described earlier assuming that the corresponding uncertainty δi varies between ±1. Most uncertainties have rank-1 dependency, meaning that they create a single I/O pair. The only uncertainty which is a rank-2 uncertainty is the Xcg variation and it creates 2 inputs and 2 outputs. This is because the CG variation affects both: pitch and lateral dynamics. It comes out that the first I/O pair of the X-cg variation couples only with the lateral model, and the second I/O pair of the X-cg variation couples only with the pitch model. So in the model separation process, one input-output pair is included in the pitch model and the other I/O pair goes in the lateral model.

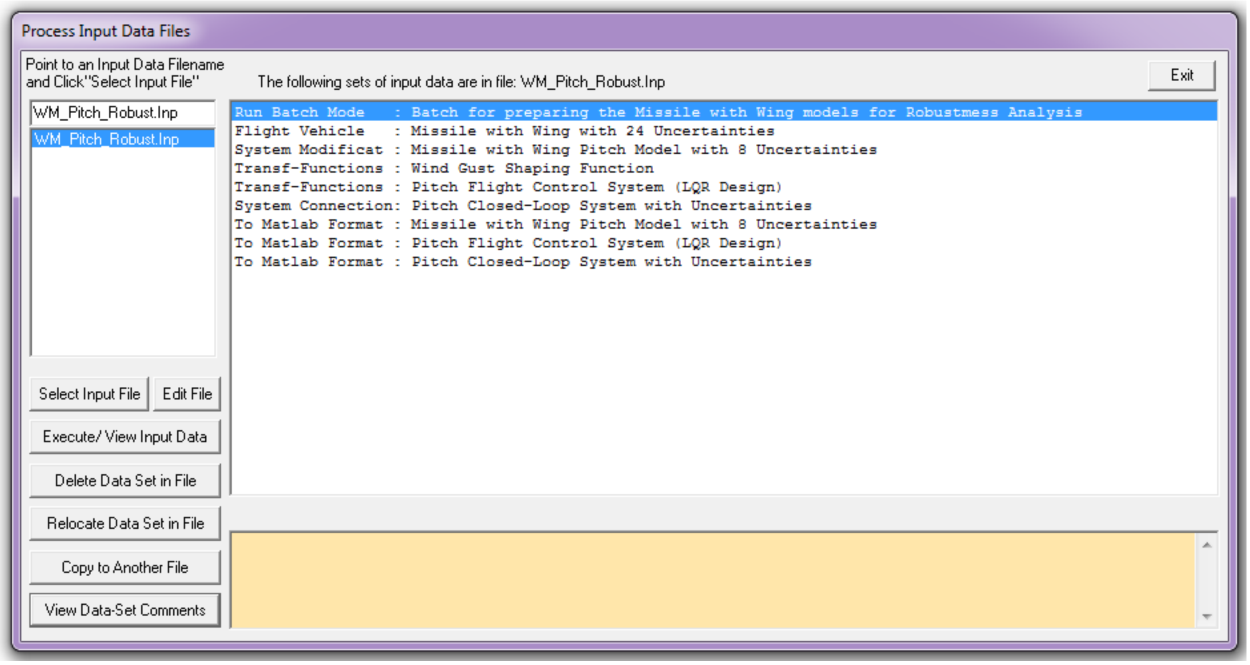
If you look further down in file "*WM_Pitch_Robust.Inp*" there is a system modification data set which extracts the pitch system with the 8 uncertainties from the fully coupled missile model. Its title is "*Missile with Wing Pitch Model with 8 Uncertainties*". It not only extracts the longitudinal states, inputs, and outputs, but also extracts the parameter uncertainty I/O pairs for the following uncertainties: [Cmα, Czα, Cm0, Cz0, Iyy, Xcg (second pair), Cz_δelev, Cm_δelev]. This longitudinal system with uncertainties is also converted to Matlab function format in file "*vehi_pitch_unc.m*". Further down the pitch flight control system is implemented in transfer-function form. Its title is "*Pitch Flight Control System (LQR Design)*". There is also a systems interconnection set of data which generates the closed-loop system by combining the pitch model with uncertainties, the pitch FCS, and the wind-gust shaping transfer-function. Its title is "*Pitch Closed-Loop System with Uncertainties*". This system is also saved as a Matlab function in file "*pitch_closed.m*".

Generating the Pitch system files in batch mode.

On the top of the vehicle data file "*WM_Pitch_Robust.Inp*" there is a batch set which can be used to run all the data sets in that file in batch mode, instead of running them one at a time. Start the Flixan program, select the subdirectory "*\Pitch Robust*", and from the top menu select: "Edit", "Manage Input Files (*.inp)", and "Process/ Edit Input Data".

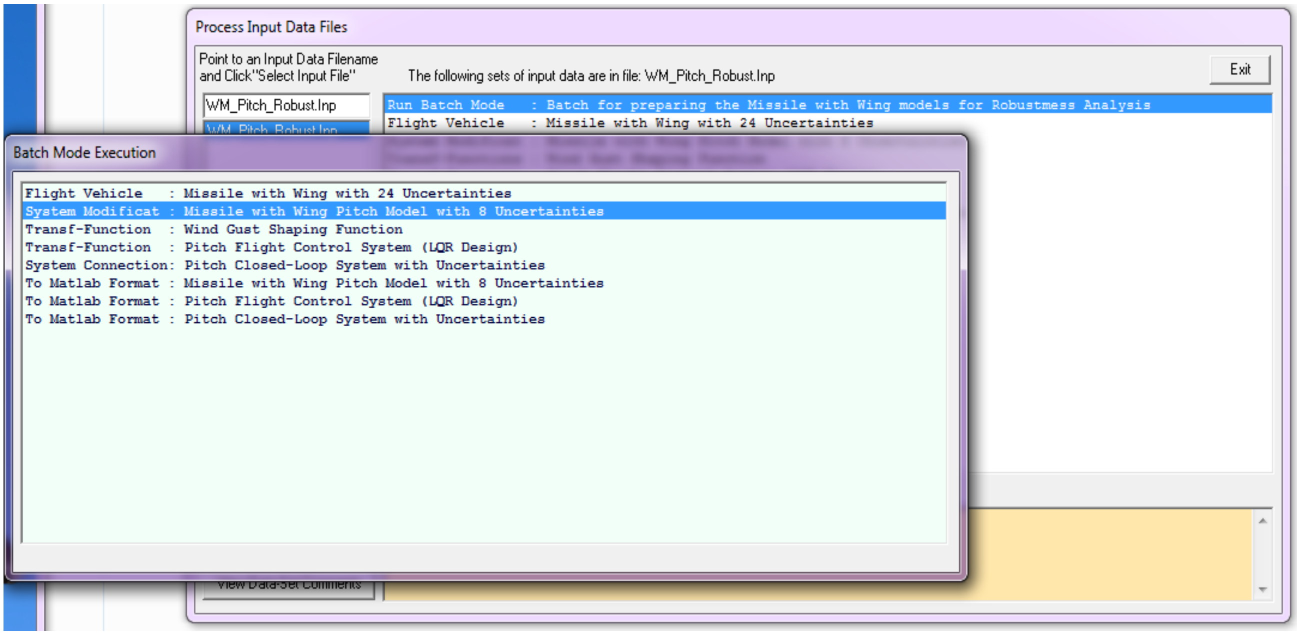

Flixan, Flight Vehicle Modeling & Control System Analysis
File
Edit
Analysis Tools
View Quad
Help
Manage Input Files (*.Inp)
Manage System Files (*.Qdr)
Create or Edit Batch Data
Process / Edit Input Data

The following dialog appears, and initially only the menu on the left hand side is filled with the input data filenames containing the vehicle data. Choose the only filename "*WM_Pitch_Robust.Inp* " and click on "Select Input Data".



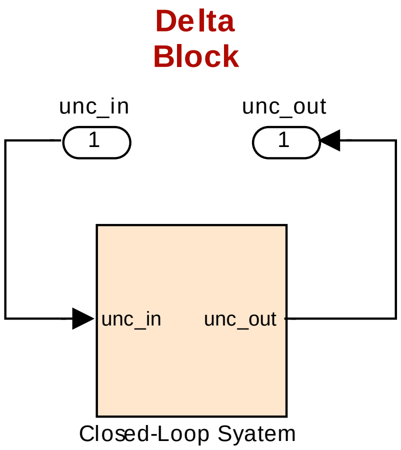
```
Run Batch Mode    : Batch for preparing the Missile with Wing models for Robustmess Analysis
Flight Vehicle    : Missile with Wing with 24 Uncertainties
System Modificat  : Missile with Wing Pitch Model with 8 Uncertainties
Transf-Functions  : Wind Gust Shaping Function
Transf-Functions  : Pitch Flight Control System (LQR Design)
System Connection: Pitch Closed-Loop System with Uncertainties
To Matlab Format  : Missile with Wing Pitch Model with 8 Uncertainties
To Matlab Format  : Pitch Flight Control System (LQR Design)
To Matlab Format  : Pitch Closed-Loop System with Uncertainties
```

The menu on the right hand side then shows the titles of all the data sets included in file "*WM_Pitch_Robust.Inp*" to be processed by the Flixan program. Each title is processed by a Flixan utility program shown on the left side of the title. Select the top title which is the batch set that will process all the other data sets, and click "Execute/ View Input Data". The program will ask you if it is okay to replace the systems which have already been created in systems file "*WM_Pitch_Robust.Qdr*". Say "Yes", and the program will process all the data in that file to create the pitch plants. A display list appears highlighting each data set while it is executing and it disappears when the batch execution is complete. Click "Exit" to close Flixan and go to Matlab.

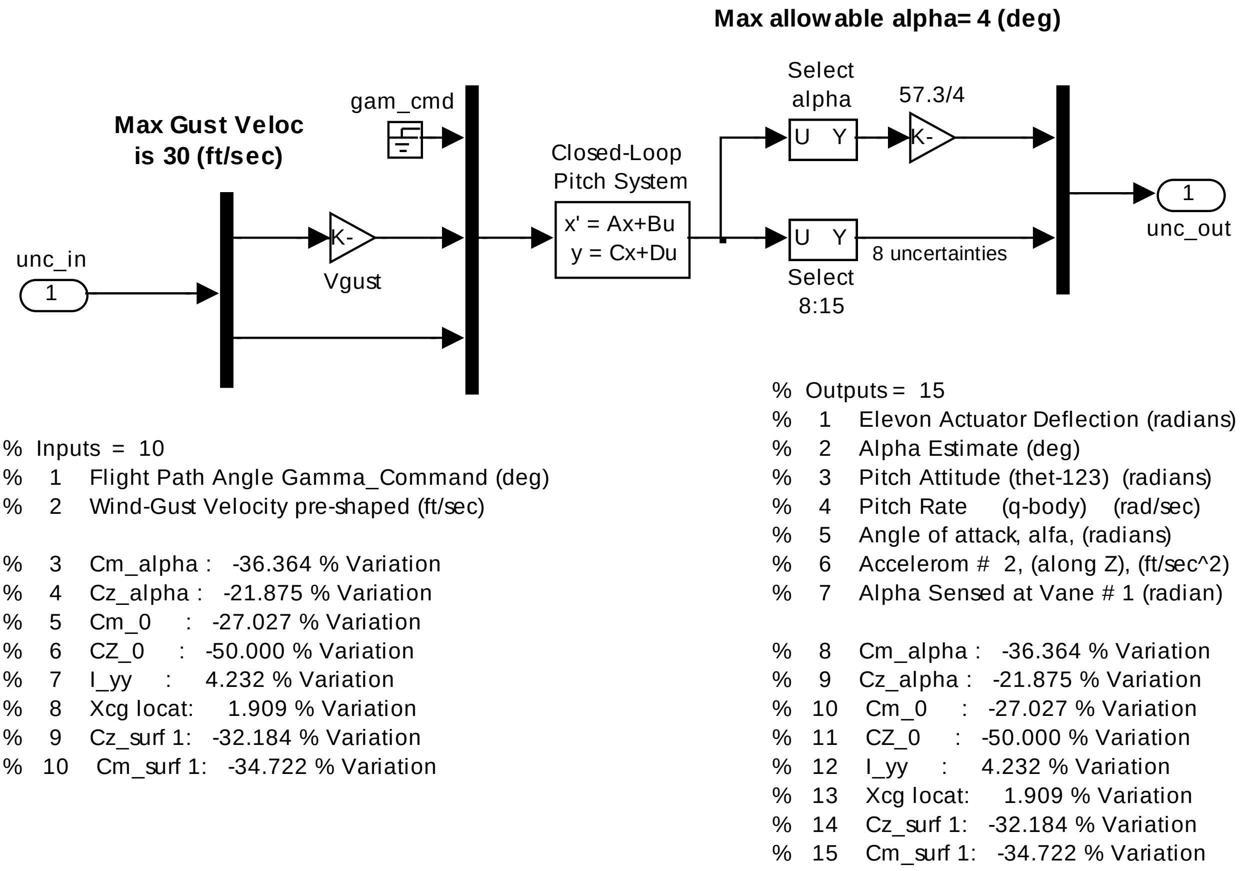

Process Input Data Files
Point to an Input Data Filename and Click"Select Input File"
The following sets of input data are in file: WM_Pitch_Robust.Inp
Exit
WM_Pitch_Robust.Inp
Run Batch Mode   : Batch for preparing the Missile with Wing models for Robustmess Analysis
Flight Vehicle   : Missile with Wing with 24 Uncertainties
View Data-Set Comments
Batch Mode Execution

```
Flight Vehicle    : Missile with Wing with 24 Uncertainties
System Modificat  : Missile with Wing Pitch Model with 8 Uncertainties
Transf-Function   : Wind Gust Shaping Function
Transf-Function   : Pitch Flight Control System (LQR Design)
System Connection: Pitch Closed-Loop System with Uncertainties
To Matlab Format  : Missile with Wing Pitch Model with 8 Uncertainties
To Matlab Format  : Pitch Flight Control System (LQR Design)
To Matlab Format  : Pitch Closed-Loop System with Uncertainties
```

Robustness and Performance Analysis of the Pitch FCS



The pitch closed-loop system model for analyzing robustness/ performance is implemented in two different versions for demo purposes. In the first version which is in model "*Closed_Robust_M1.mdl*", the entire closed-loop system, which includes: plant model, the FCS, and the gust shaping function, is implemented using Flixan. Its title is "*Pitch Closed-Loop System with Uncertainties*". It is saved as Matlab state-space function in file "*pitch_closed.m*", and it is loaded into Matlab using the initialization file "init.m". The closed-loop plant model is shown in detail in Fig. (25), and it is set up to analyze Robust-Performance. Its inputs include the wind-gust disturbance and the 8 pitch uncertainties. The gust is scaled up by a gain factor of 30 to reflect the max expected wind velocity. The plant outputs include the angle of attack (α) and the 8 pitch uncertainties. The α-output is converted to (deg) and divided by 4 because 4 (deg) is the max acceptable α. In other words, all 9 plant I/Os (1 performance I/O and 8 parameter variation I/Os) are normalized to correspond to a Δ block whose elements vary between ±1.



```
% Inputs = 10
%   1    Flight Path Angle Gamma_Command (deg)
%   2    Wind-Gust Velocity pre-shaped (ft/sec)

%   3    Cm_alpha :   -36.364 % Variation
%   4    Cz_alpha :   -21.875 % Variation
%   5    Cm_0     :   -27.027 % Variation
%   6    CZ_0     :   -50.000 % Variation
%   7    I_yy     :     4.232 % Variation
%   8    Xcg locat:     1.909 % Variation
%   9    Cz_surf 1:   -32.184 % Variation
%  10    Cm_surf 1:   -34.722 % Variation
```

```
% Outputs = 15
%   1    Elevon Actuator Deflection (radians)
%   2    Alpha Estimate (deg)
%   3    Pitch Attitude (thet-123)  (radians)
%   4    Pitch Rate      (q-body)    (rad/sec)
%   5    Angle of attack, alfa, (radians)
%   6    Accelerom #  2, (along Z), (ft/sec^2)
%   7    Alpha Sensed at Vane # 1 (radian)

%   8    Cm_alpha :   -36.364 % Variation
%   9    Cz_alpha :   -21.875 % Variation
%  10    Cm_0     :   -27.027 % Variation
%  11    CZ_0     :   -50.000 % Variation
%  12    I_yy     :     4.232 % Variation
%  13    Xcg locat:     1.909 % Variation
%  14    Cz_surf 1:   -32.184 % Variation
%  15    Cm_surf 1:   -34.722 % Variation
```

Figure 3.3 Simulink Model "*Closed_Robust_M1.mdl* " used to analyze Robust Performance in Pitch

In the analysis, of course, we don't connect it with any Δ block. We only prepare the closed-loop system M(s) that corresponds to a normalized Δ block having unity elements, and then we compute the structured singular-value (μ) across M(s). The following script computes the SSV plot across the 9 I/Os in order to determine if the system achieves a robust performance.

```
init
w=logspace(-2,3,300);                  % and Frequ domain analysis
npv=9;                                 % Number of Param Variations

[Acl,Bcl,Ccl,Dcl]=linmod('Closed_Robust_M1');
sys=ss(Acl,Bcl,Ccl,Dcl);
sysf= FRD(sys,w);
blk=[-ones(npv,1), zeros(npv,1)];
[bnd,muinfo]= mussv(sysf,blk);

ff= get(muinfo.bnds, 'frequency');
muu=get(muinfo.bnds, 'responsedata');
muu=squeeze(muu);
muu=muu(1,:);
figure(14); loglog(ff,muu,'LineWidth',1.5)
axis([0.01, 1000, 0.01, 1])
xlabel('Frequency (rad/sec)')
ylabel('Structured Sing. Value (mu)')
```

The longitudinal system robust performance analysis is shown in Figure (3.4b). It barely meets the Robust Performance requirements because $\mu(M)$ is barely below 1. If we relax the robust performance requirement and check only for robustness across the 8 parameter uncertainty I/Os alone, ignoring the gust sensitivity I/O, the $\mu(M)$ drops further down, see Fig. (3.4a), allowing more margin towards robustness but at the expense of performance.

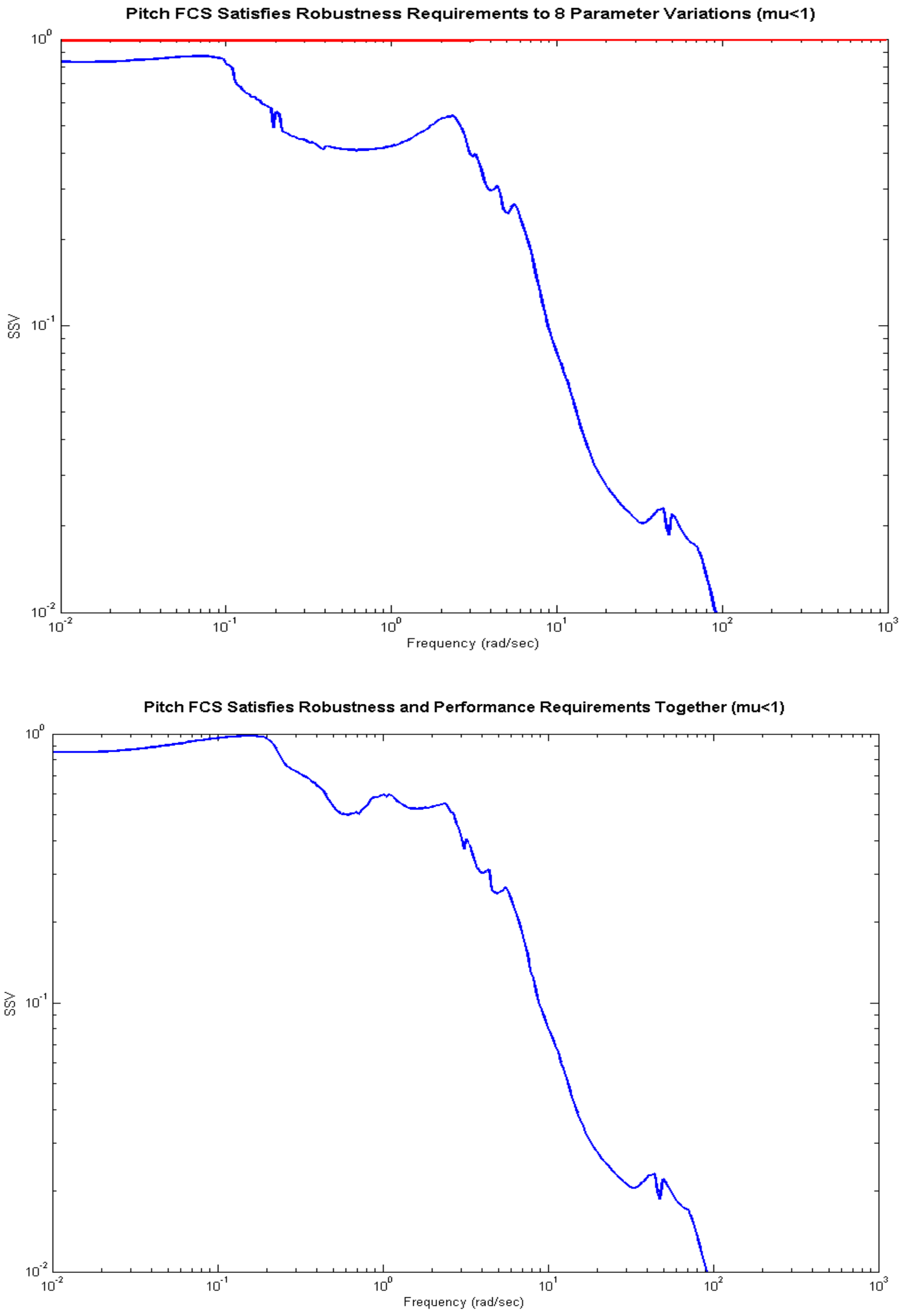


Figure 3.4 Pitch System Robustness and Robust Performance Results

For demo purposes and also for checking things out, we have created the same closed-loop system by combining smaller subsystems using Simulink. This new version is implemented in Simulink model "*Closed_Robust_M2.mdl*", shown in Fig. (3.5). The green block in Fig. (3.5) is the missile pitch dynamics which includes also the 8 uncertainties. The pink block is the pitch FCS which closes the loop around the vehicle. The FCS block is shown in detail in Figure (3.6). It consists of the LQR gains, the alpha estimator, and includes also the elevon actuator.

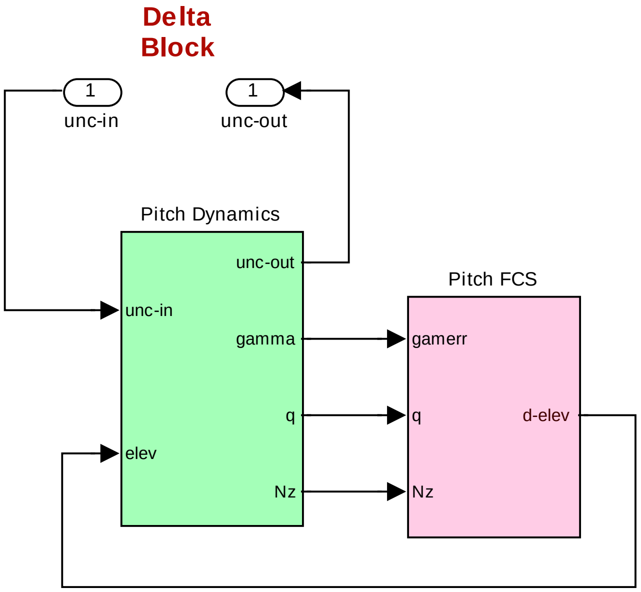


Figure 3.5 Simulink Model "*Closed_Robust_M2.mdl*" for analyzing Pitch Robust Performance

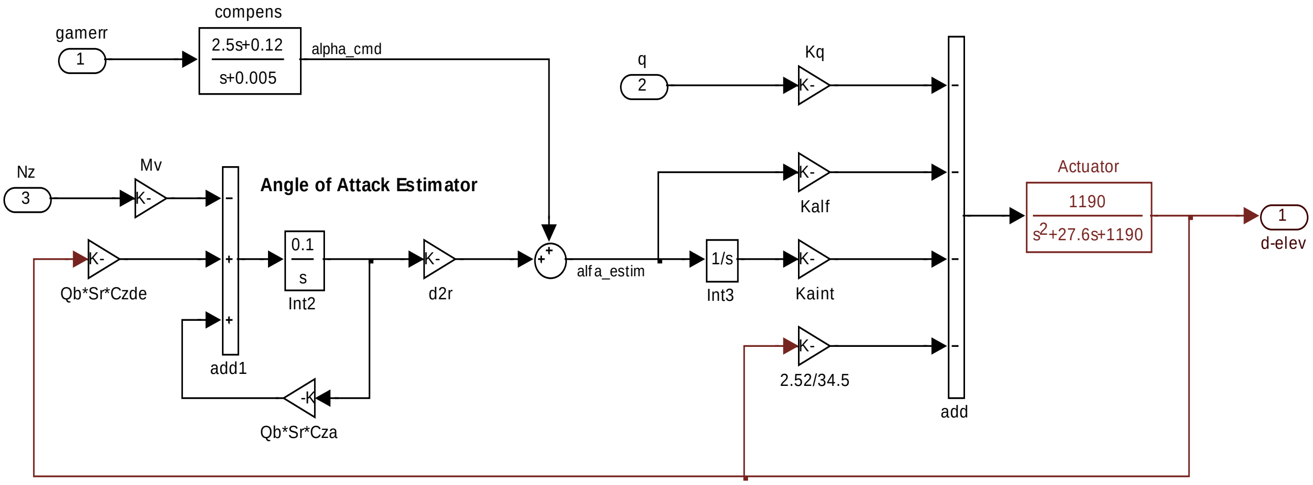
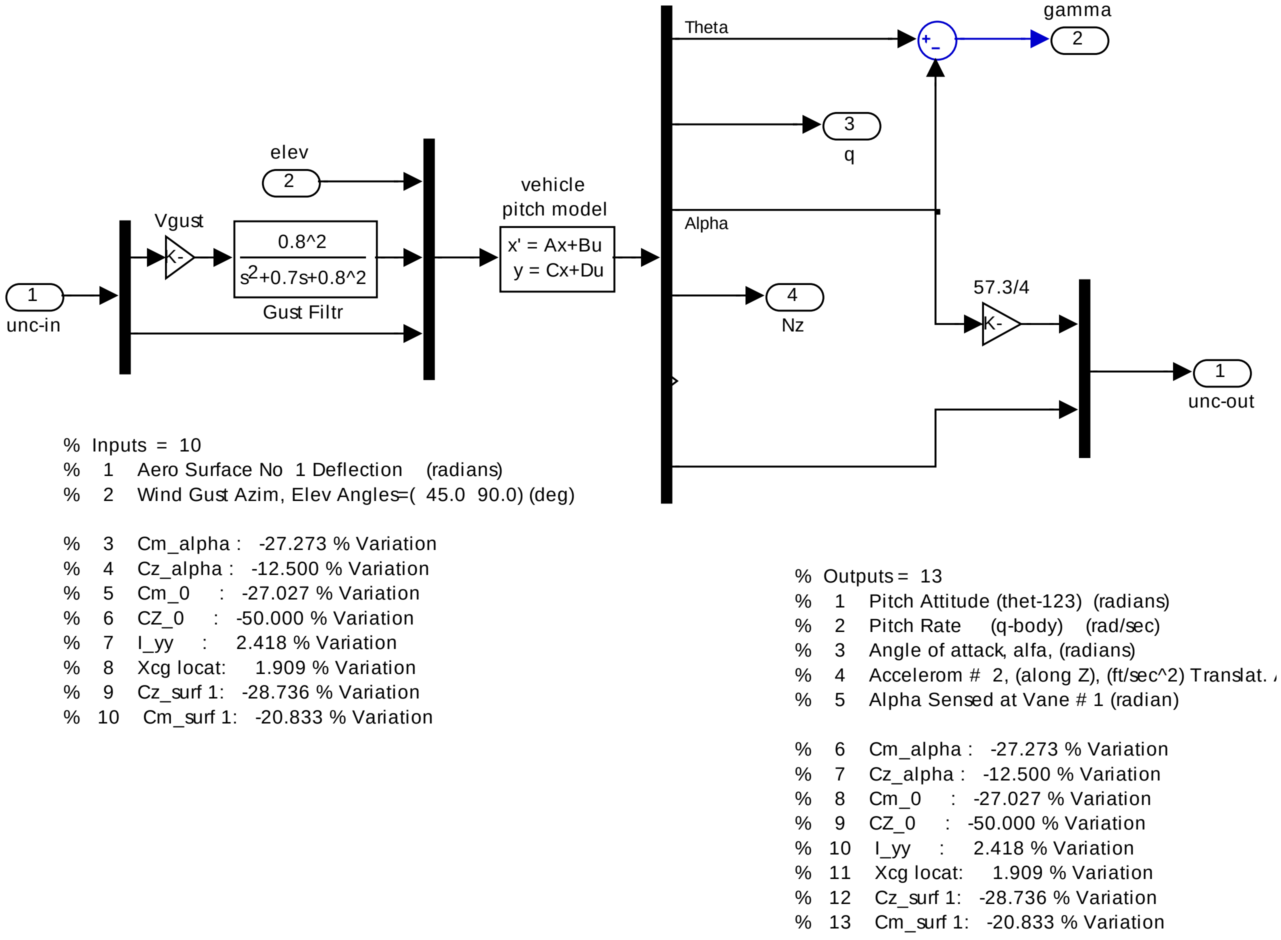


Figure 3.6 State-Feedback Pitch LQR Controller

Figure (3.7) shows the vehicle pitch dynamics with the 8 uncertainty I/Os included. It uses the missile state-space model "*Missile with Wing Pitch Model with 8 Uncertainties*" which is loaded from Matlab function file "*vehi_pitch_unc.m*". The gust shaping filter is also included at the gust input. In this configuration the dynamic model is set up to analyze Robust-Performance. Its 9 inputs include the wind-gust disturbance and the 8 pitch uncertainties. The gust is scaled up by a gain factor of 30 to reflect the max expected wind velocity. The 9 vehicle outputs include the angle of attack (α) and the 8 pitch uncertainties. The α-output is converted to (deg) and divided by 4 because 4 (deg) is the max acceptable α. In other words, all 9 plant I/Os (1 performance I/O and 8 parameter variation I/Os) are normalized to correspond to a structured Δ block whose elements vary between ±1. The μ-analysis results obtained from this model are identical to the Figure (3.4) results obtained from the previous model.



```
% Inputs = 10
%   1    Aero Surface No  1 Deflection    (radians)
%   2    Wind Gust Azim, Elev Angles=(  45.0  90.0) (deg)

%   3    Cm_alpha :   -27.273 % Variation
%   4    Cz_alpha :   -12.500 % Variation
%   5    Cm_0     :   -27.027 % Variation
%   6    CZ_0     :   -50.000 % Variation
%   7    I_yy     :     2.418 % Variation
%   8    Xcg locat:     1.909 % Variation
%   9    Cz_surf 1:   -28.736 % Variation
%   10   Cm_surf 1:   -20.833 % Variation
```

```
% Outputs = 13
%   1    Pitch Attitude (thet-123)  (radians)
%   2    Pitch Rate     (q-body)    (rad/sec)
%   3    Angle of attack, alfa, (radians)
%   4    Accelerom #  2, (along Z), (ft/sec^2) Translat.
%   5    Alpha Sensed at Vane # 1 (radian)

%   6    Cm_alpha :   -27.273 % Variation
%   7    Cz_alpha :   -12.500 % Variation
%   8    Cm_0     :   -27.027 % Variation
%   9    CZ_0     :   -50.000 % Variation
%   10   I_yy     :     2.418 % Variation
%   11   Xcg locat:     1.909 % Variation
%   12   Cz_surf 1:   -28.736 % Variation
%   13   Cm_surf 1:   -20.833 % Variation
```

Figure 3.7 Vehicle Pitch dynamic model with uncertainties for Robust Performance analysis

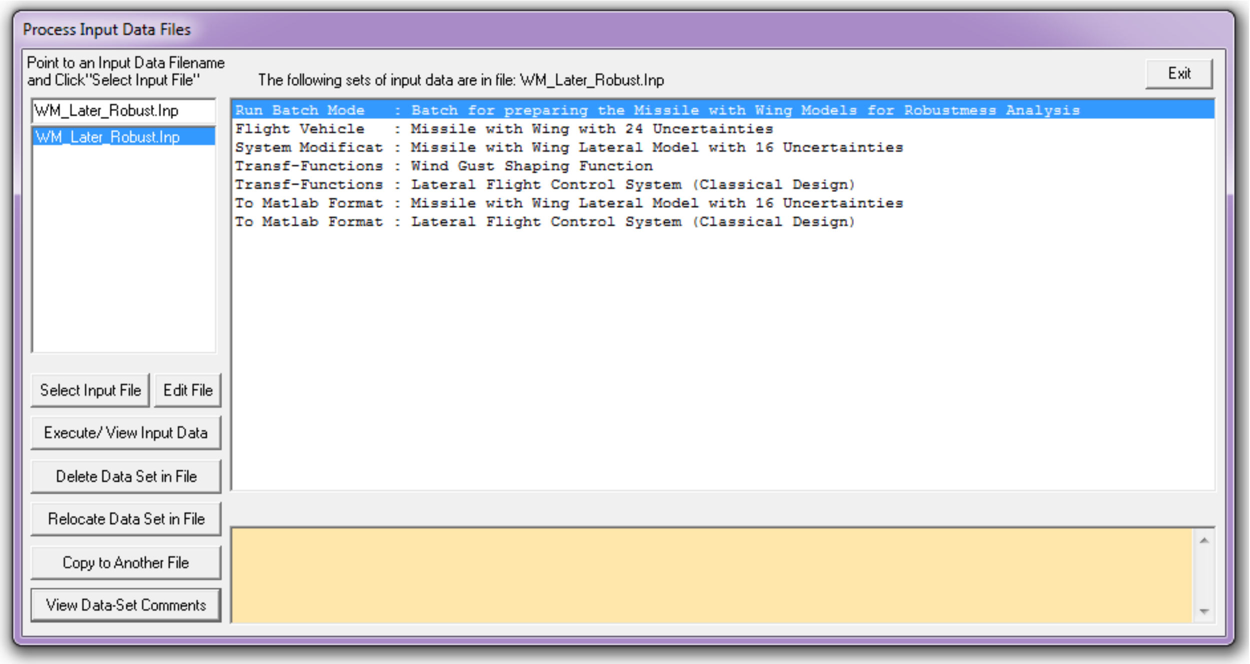
3.2 Lateral System Robustness Analysis

The lateral system robustness analysis is performed in directory "*C:\Flixan\ Examples\ Missile with Wing\(c) Robustness Analysis\Lateral Robust*". We are going to evaluate and compare Robustness/ Sensitivity of the two lateral FCS designs to parameter uncertainties and lateral load sensitivity to wind-gusts. The missile input data is in file "*WM_Later_Robust.Inp*". It is the same as in the pitch model and its title is "*Missile with Wing with 24 Uncertainties*". At the bottom of the vehicle data-set there is a call to the parameter uncertainties data-set: "*Missile Uncertainties at Mach=2.5, Alpha=10*" which is located right below the nominal vehicle data. The missile uncertainties data is the same as in the pitch model. The Flixan Vehicle Modeling program processes the vehicle and uncertainties data and saves the vehicle state-space system in file "*WM_Later_Robust.Qdr*". This plant model is basically the same as the nominal plant created in Section 1, but it has 24 additional inputs and outputs. An I/O pair is created for every uncertainty. Each I/O pair is scaled as we described earlier assuming that the corresponding uncertainty δi varies between ±1. Most uncertainties have rank-1 dependency, meaning that they create a single I/O pair. The only uncertainty which is a rank-2 uncertainty is the X-cg variation and it creates 2 inputs and 2 outputs. This is because the CG variation affects both: pitch and lateral dynamics. It comes out that the first I/O pair of the X-cg variation couples only with the lateral model, while the second I/O pair couples only with the pitch model as we discussed in the previous section. So in the model separation process, only the first input-output pair of the Xcg uncertainty is included in the lateral model.

If you look further down in file "*WM_Later_Robust.Inp*" there is a system modification data set which extracts the lateral system with the 16 uncertainties from the fully coupled missile model. Its title is "*Missile with Wing Lateral Model with 16 Uncertainties*". It not only extracts the lateral states, inputs, and outputs, but it also extracts the parameter uncertainty I/O pairs for the following 16 uncertainties: [Clβ, Cnβ, Cyβ, Cnp, Cnr, Clp, Clr, Ixx, Izz, Xcg (first pair), Cy_δail, Cl_δail, Cn_δail, Cy_δrud, Cl_δrud, Cn_δrud]. This lateral system with uncertainties is also converted to Matlab function format in file "*vehi_later_unc.m*". Further down the lateral flight control system is implemented in transfer-function form. Its title is "*Lateral Flight Control System (Classical Design)*".

Generating the Lateral system files in batch mode.

On the top of the vehicle data file "*WM_Later_Robust.Inp*" there is a batch set which can be used to run all the data sets in that file in batch mode, instead of running them one at a time. Start the Flixan program, select the subdirectory "*\Lateral Robust*", and from the top menu select: "Edit", "Manage Input Files (*.inp)", and "Process/ Edit Input Data". The following dialog appears, and initially only the menu on the left hand side is filled with the input data filenames containing the vehicle data. Choose the only filename "*WM_Later_Robust.Inp* " and click on "Select Input Data".



The following sets of input data are in file: WM_Later_Robust.Inp

```
Run Batch Mode    : Batch for preparing the Missile with Wing Models for Robustmess Analysis
Flight Vehicle    : Missile with Wing with 24 Uncertainties
System Modificat  : Missile with Wing Lateral Model with 16 Uncertainties
Transf-Functions  : Wind Gust Shaping Function
Transf-Functions  : Lateral Flight Control System (Classical Design)
To Matlab Format  : Missile with Wing Lateral Model with 16 Uncertainties
To Matlab Format  : Lateral Flight Control System (Classical Design)
```

The menu on the right hand side then shows the titles of all the data sets included in file "*WM_Later_Robust.Inp*" to be processed by the Flixan program. Each title is processed by a Flixan utility program shown on the left side of the title. Select the top title which is the batch set that will process all the other data sets, and click "Execute/ View Input Data". The program will ask you if it is okay to replace the systems which have already been created in systems file "*WM_Later_Robust.Qdr*". Say "Yes", and the program will process all the data in that file to create the lateral systems. Click "Exit" to close Flixan and go to Matlab.

Robustness and Performance Analysis of the Lateral FCS

There are two Simulink models in subdirectory "\Later Robust" for analyzing lateral system robustness. The model "Robust_Classic.mdl", shown in Fig. (3.8), that analyzes the classical FCS design, and the model "Robust_LQR.mdl", shown in Fig. (3.10), that analyzes the lateral LQR flight controller which was designed in section 2. They both use the missile lateral state-space model with 16 uncertainty I/Os, which is system title: "Missile with Wing Lateral Model with 16 Uncertainties", in Matlab state-space system file "vehi_later_unc.m". The initialization file "init.m" loads this system into Matlab, plus some additional parameters needed by the FCS.

Robustness/ Sensitivity to Disturbance Analysis (Classical Design)

The robustness analysis model that uses the classical FCS design is shown in Fig. (3.8-a). The missile lateral dynamics is in the green block which is shown in detail in Fig. (3.8-b). The FCS block closes the loop between the plant outputs and the aileron and rudder deflection inputs, and we already know that the nominal closed-loop system is stable (must be). This system is set up to analyze robust performance in the lateral axes. There are 17 input-output pairs which are normalized to correspond to a 17 element diagonal Δ block. Meaning, that it is normalized and its diagonal elements can vary up to ±1. There are 16 parameter uncertainty I/Os and one I/O for sensitivity analysis to lateral wind-gust disturbances. The gust is scaled up by a gain factor of 30 to reflect the max expected wind velocity. A gust shaping filter is used to capture the gust bandwidth characteristics. The plant outputs, in addition to the 16 uncertainties, include also the angle of sideslip (β). The β-output is converted to (deg) and divided by 4 because 4 (deg) is the max acceptable β.

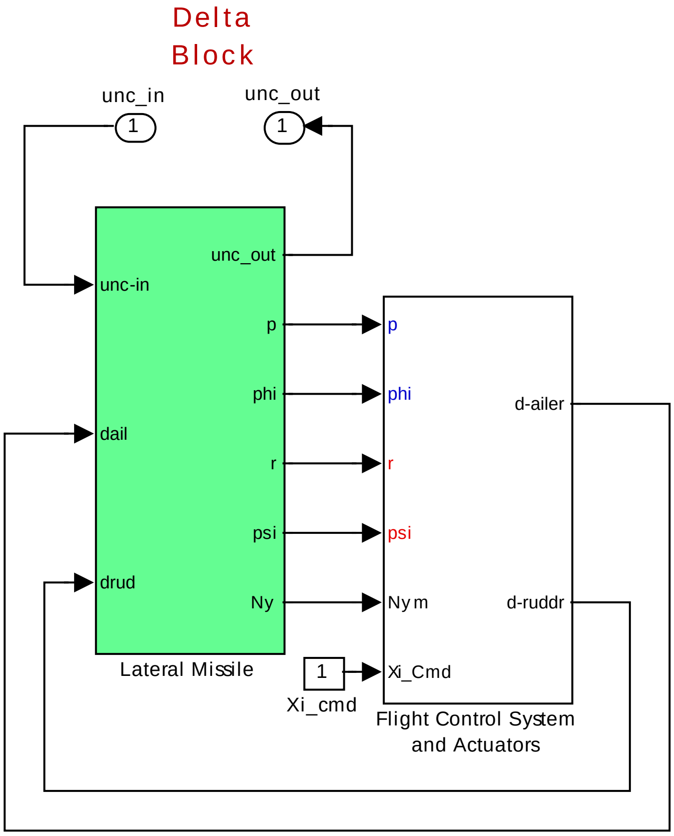
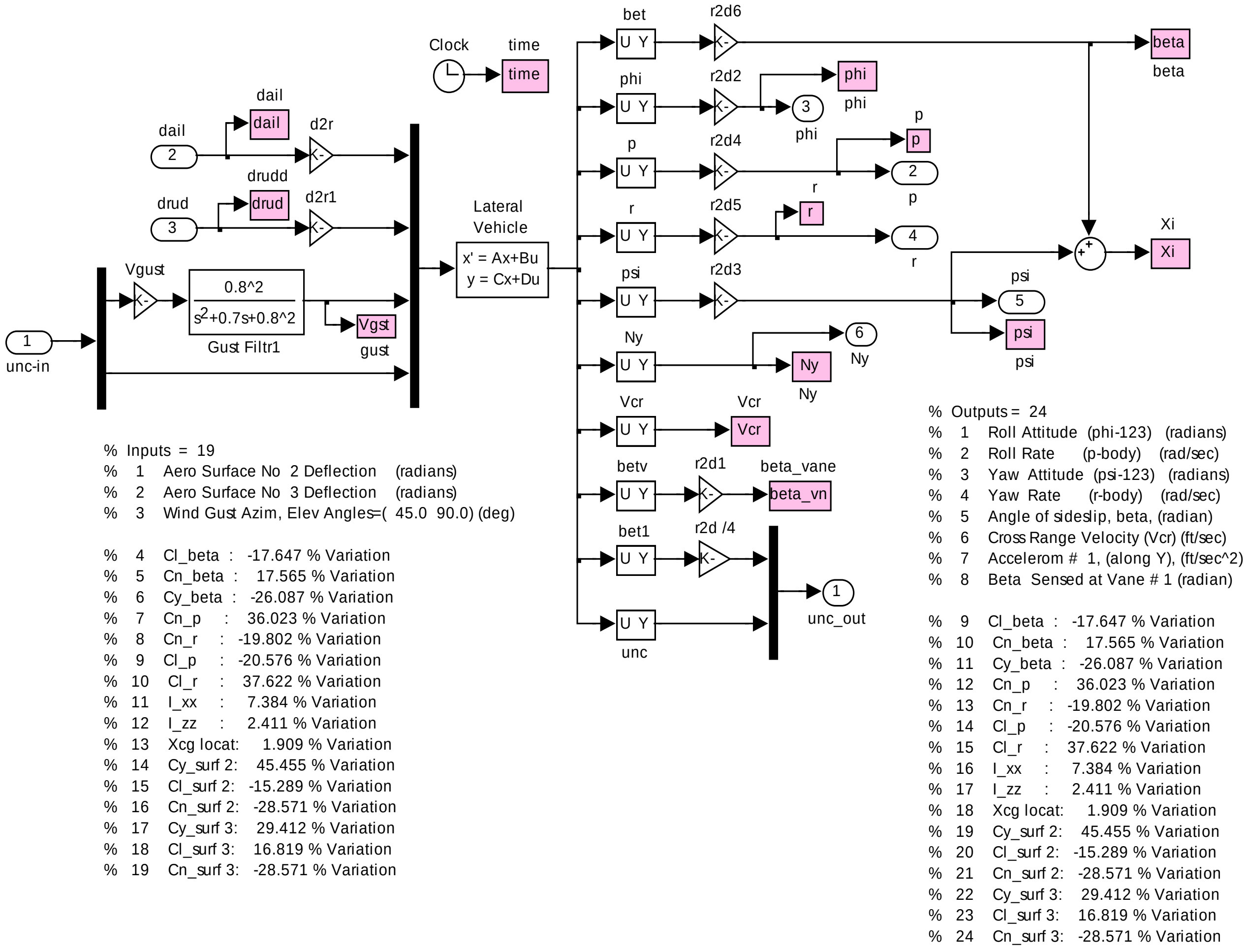


Figure 3.8 Lateral Classical Flight Control Robustness Analysis model "Robust_Classic.mdl"



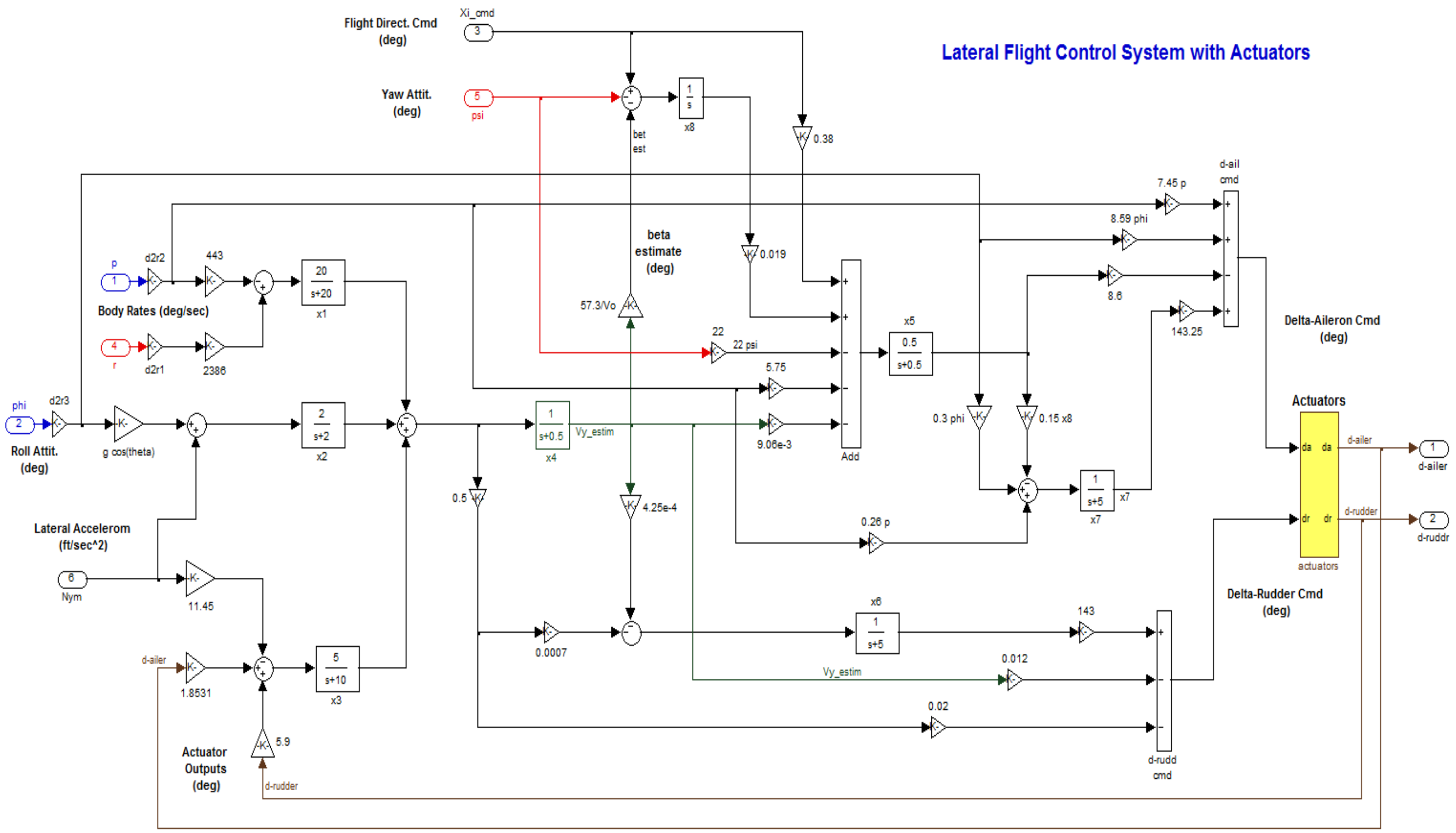
```
%  Inputs  =  19
%    1    Aero Surface No  2 Deflection    (radians)
%    2    Aero Surface No  3 Deflection    (radians)
%    3    Wind Gust Azim, Elev Angles=(  45.0  90.0) (deg)

%    4    Cl_beta  :   -17.647 % Variation
%    5    Cn_beta  :    17.565 % Variation
%    6    Cy_beta  :   -26.087 % Variation
%    7    Cn_p     :    36.023 % Variation
%    8    Cn_r     :   -19.802 % Variation
%    9    Cl_p     :   -20.576 % Variation
%   10    Cl_r     :    37.622 % Variation
%   11    I_xx     :     7.384 % Variation
%   12    I_zz     :     2.411 % Variation
%   13    Xcg locat:     1.909 % Variation
%   14    Cy_surf 2:    45.455 % Variation
%   15    Cl_surf 2:   -15.289 % Variation
%   16    Cn_surf 2:   -28.571 % Variation
%   17    Cy_surf 3:    29.412 % Variation
%   18    Cl_surf 3:    16.819 % Variation
%   19    Cn_surf 3:   -28.571 % Variation
```

```
%  Outputs =  24
%    1    Roll Attitude  (phi-123)   (radians)
%    2    Roll Rate      (p-body)    (rad/sec)
%    3    Yaw  Attitude  (psi-123)   (radians)
%    4    Yaw  Rate      (r-body)    (rad/sec)
%    5    Angle of sideslip, beta, (radian)
%    6    Cross Range Velocity (Vcr) (ft/sec)
%    7    Accelerom #  1, (along Y), (ft/sec^2)
%    8    Beta  Sensed at Vane # 1 (radian)

%    9    Cl_beta  :   -17.647 % Variation
%   10    Cn_beta  :    17.565 % Variation
%   11    Cy_beta  :   -26.087 % Variation
%   12    Cn_p     :    36.023 % Variation
%   13    Cn_r     :   -19.802 % Variation
%   14    Cl_p     :   -20.576 % Variation
%   15    Cl_r     :    37.622 % Variation
%   16    I_xx     :     7.384 % Variation
%   17    I_zz     :     2.411 % Variation
%   18    Xcg locat:     1.909 % Variation
%   19    Cy_surf 2:    45.455 % Variation
%   20    Cl_surf 2:   -15.289 % Variation
%   21    Cn_surf 2:   -28.571 % Variation
%   22    Cy_surf 3:    29.412 % Variation
%   23    Cl_surf 3:    16.819 % Variation
%   24    Cn_surf 3:   -28.571 % Variation
```

Figure 3.8-b Missile Lateral dynamic model showing the 17 normalized inputs/ outputs that connect to the uncertainty block Δ

Robust Performance of this lateral closed-loop system is evaluated by calculating the (μ) frequency response across the inputs and outputs that correspond to the diagonal Δ block. The μ-analysis results are shown: (a) for robustness alone in Fig. (3.9-a) (across only the 16 uncertainties), and (b) for combined robust performance in Fig. (3.9-b) (across all 17 inputs and outputs including the sensitivity i/o). The plots clearly show that robust performance is satisfied with some margin. The Matlab scrip file "SSV_frq.m" computes and plots the SSV frequency responses.

Lateral Flight Control System with Actuators



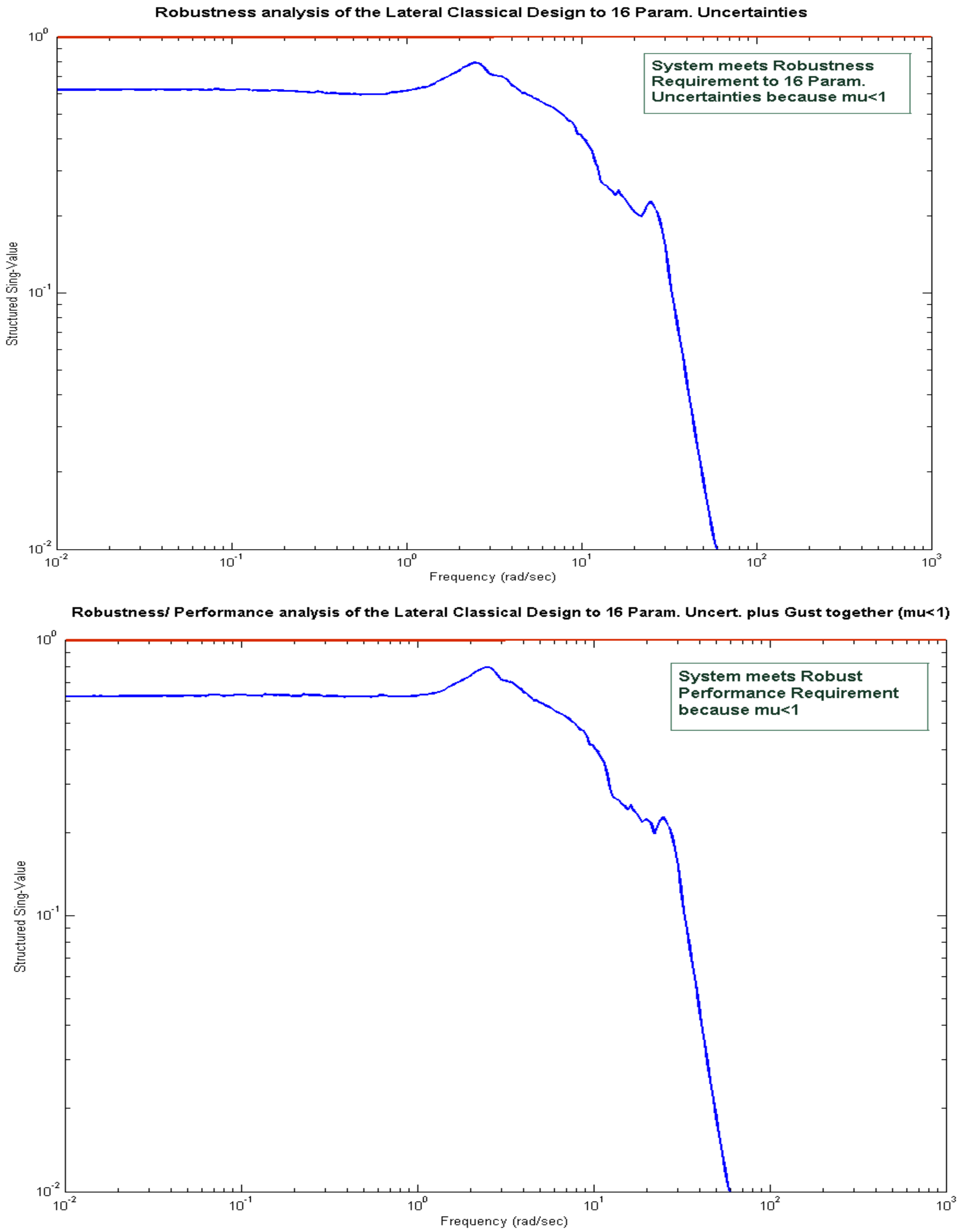


Figure 3.9 The Classical Lateral FCS Design Satisfies Robust Performance Requirements

Robustness/ Sensitivity to Disturbance Analysis (LQR Design)

The Simulink model "Robust_LQR.mdl" in Fig. (3.10) is used to analyze the LQR controller which is basically a state-feedback design. The (2x9) state-feedback gain matrix (Kpr) is loaded into Matlab by the initialization file. The angle of sideslip (b) is not directly measureable but it is estimated from (Ny) as it was discussed earlier. The structure and the scaling of the 17 inputs and outputs is the same as in the previous classical design analysis.

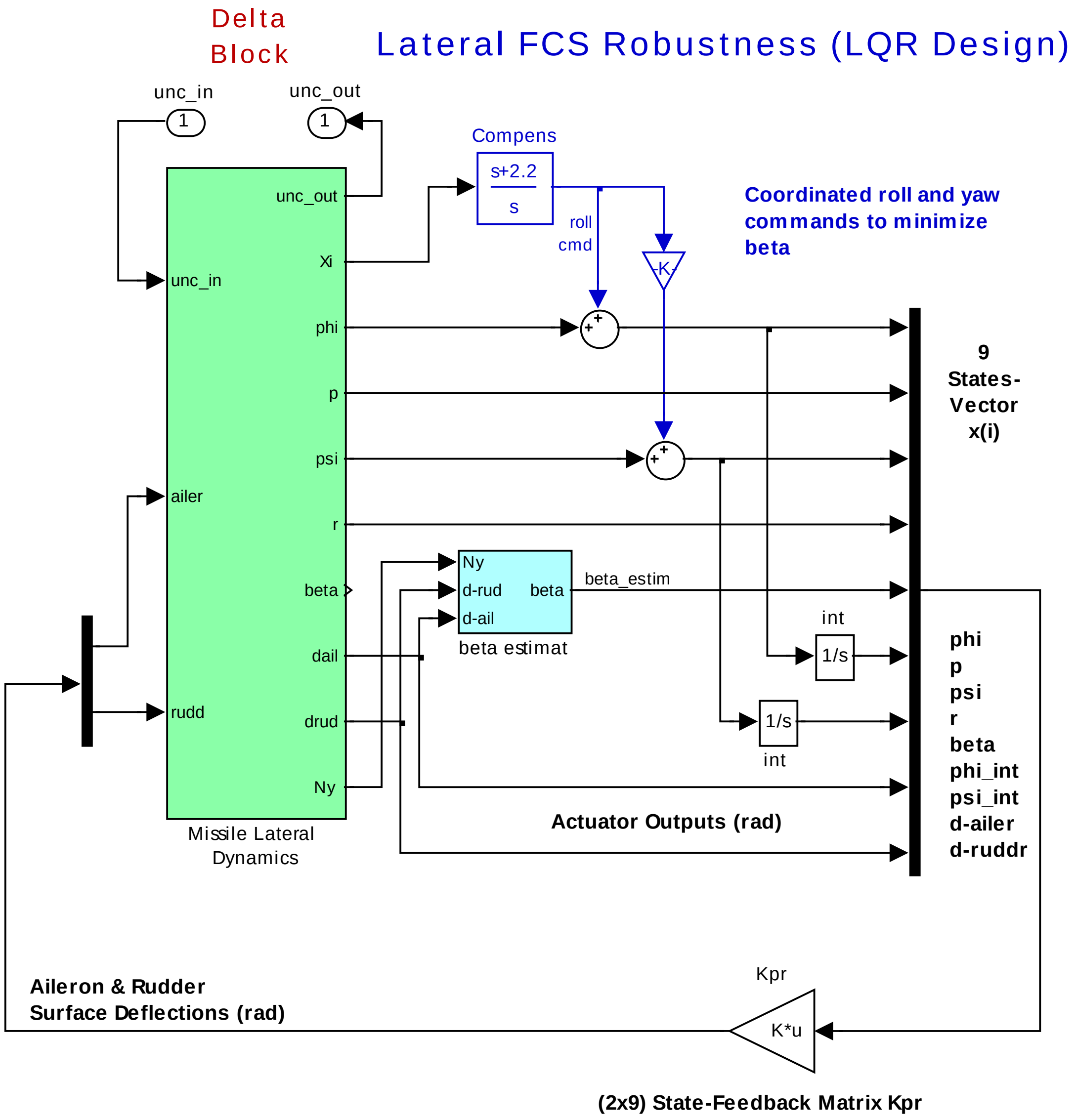
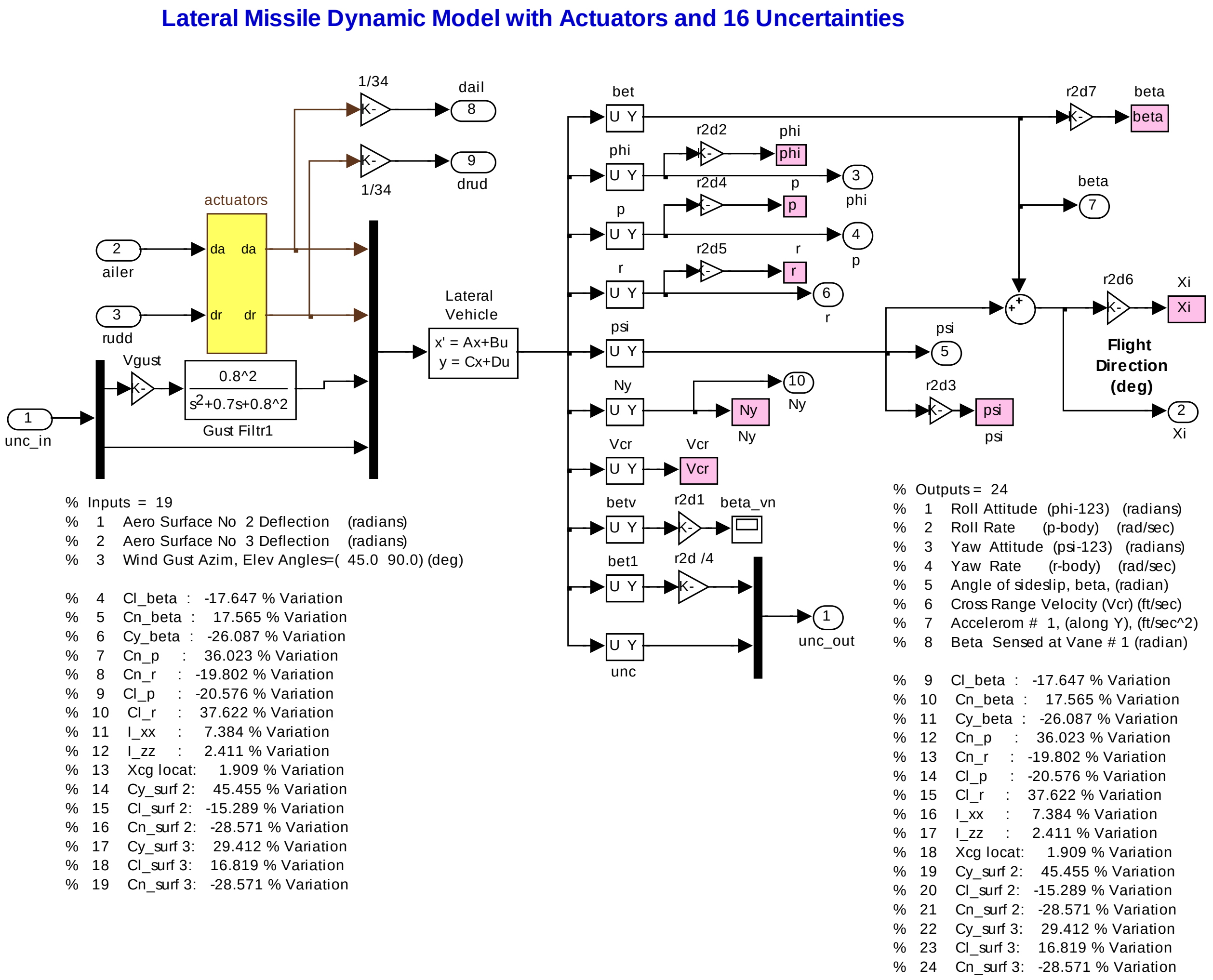


Figure 3.10 Lateral LQR Flight Control System Robustness Analysis Mode "Robust_LQR.mdl"



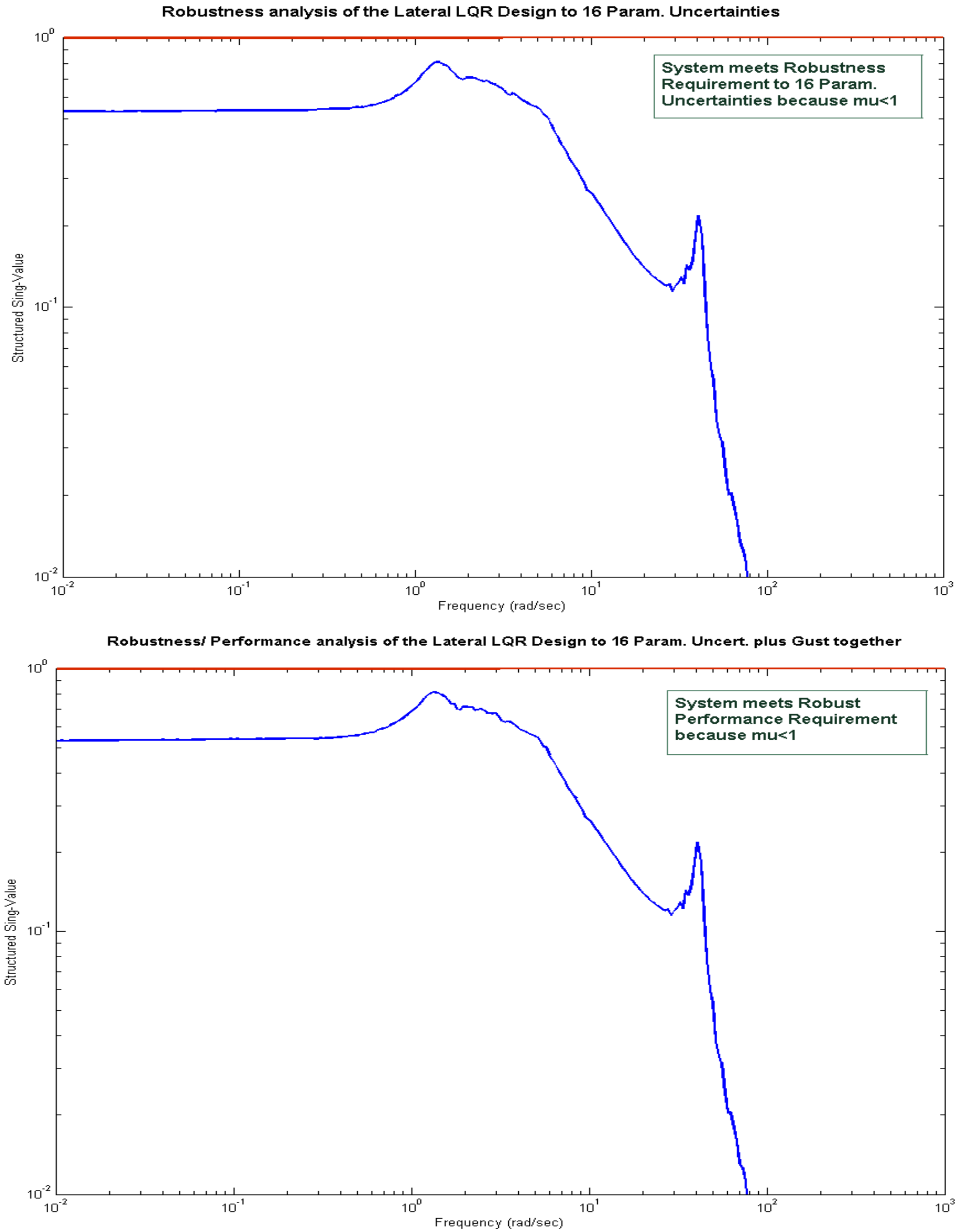
Lateral Missile Dynamic Model with Actuators and 16 Uncertainties

```
%  Inputs  =  19
%   1    Aero Surface No  2 Deflection    (radians)
%   2    Aero Surface No  3 Deflection    (radians)
%   3    Wind Gust Azim, Elev Angles=(  45.0  90.0) (deg)

%   4    Cl_beta  :   -17.647 % Variation
%   5    Cn_beta  :    17.565 % Variation
%   6    Cy_beta  :   -26.087 % Variation
%   7    Cn_p     :    36.023 % Variation
%   8    Cn_r     :   -19.802 % Variation
%   9    Cl_p     :   -20.576 % Variation
%  10    Cl_r     :    37.622 % Variation
%  11    I_xx     :     7.384 % Variation
%  12    I_zz     :     2.411 % Variation
%  13    Xcg locat:     1.909 % Variation
%  14    Cy_surf 2:    45.455 % Variation
%  15    Cl_surf 2:   -15.289 % Variation
%  16    Cn_surf 2:   -28.571 % Variation
%  17    Cy_surf 3:    29.412 % Variation
%  18    Cl_surf 3:    16.819 % Variation
%  19    Cn_surf 3:   -28.571 % Variation
```

```
%  Outputs =  24
%   1    Roll Attitude  (phi-123)   (radians)
%   2    Roll Rate      (p-body)    (rad/sec)
%   3    Yaw  Attitude  (psi-123)  (radians)
%   4    Yaw  Rate      (r-body)    (rad/sec)
%   5    Angle of sideslip, beta, (radian)
%   6    Cross Range Velocity (Vcr) (ft/sec)
%   7    Accelerom #  1, (along Y), (ft/sec^2)
%   8    Beta  Sensed at Vane # 1 (radian)

%   9    Cl_beta  :   -17.647 % Variation
%  10    Cn_beta  :    17.565 % Variation
%  11    Cy_beta  :   -26.087 % Variation
%  12    Cn_p     :    36.023 % Variation
%  13    Cn_r     :   -19.802 % Variation
%  14    Cl_p     :   -20.576 % Variation
%  15    Cl_r     :    37.622 % Variation
%  16    I_xx     :     7.384 % Variation
%  17    I_zz     :     2.411 % Variation
%  18    Xcg locat:     1.909 % Variation
%  19    Cy_surf 2:    45.455 % Variation
%  20    Cl_surf 2:   -15.289 % Variation
%  21    Cn_surf 2:   -28.571 % Variation
%  22    Cy_surf 3:    29.412 % Variation
%  23    Cl_surf 3:    16.819 % Variation
%  24    Cn_surf 3:   -28.571 % Variation
```

Figure 3.10-b Lateral Missile Dynamic Model with 16 Uncertainties used for Robust Performance Analysis. It includes also the Aileron and Rudder Actuator Dynamics.

The missile lateral dynamics is in the green block which is shown in detail in Fig. (3.10-b). The aileron and rudder actuators (yellow block) are also included in this subsystem. Robust Performance is evaluated by calculating the (μ) frequency response across the inputs and outputs that correspond to the diagonal Δ block. The μ-analysis results are shown: (a) for robustness alone in Fig. (3.11-a) (across only the 16 uncertainties), and (b) for combined robust performance in Fig. (3.11-b) (across all 17 inputs and outputs including the sensitivity i/o). The plots clearly show that the LQR design also satisfies robust performance requirements with approximately the same margins as the classical design. Overall, however, if you also consider the transient responses, the LQR design has the advantage that it uses less aileron and rudder deflections. The Matlab scrip file "SSV_frq.m" computes and plots the SSV frequency responses.



Robustness analysis of the Lateral LQR Design to 16 Param. Uncertainties

System meets Robustness Requirement to 16 Param. Uncertainties because mu<1

Robustness/ Performance analysis of the Lateral LQR Design to 16 Param. Uncert. plus Gust together

System meets Robust Performance Requirement because mu<1

Figure 3.11 The LQR Lateral control system design also satisfies Robust Performance requirements