Linear Quadratic Control Examples

6.1 Overhead Crane Example

In this example the plant system consists of two masses m₁ and m₂ connected with a rope. The mass m₁ is suspended from m₂ by the rope, as shown in Figure 6.1.1, representing a simple model of an overhead crane. The mass m₂ can only move along they y direction as a result of the control force F which is applied on m₂ along they y direction. Equations 6.1.1 describe the motion of the two masses

$$m_1 \ddot{y}_1 = m_1\, g \sin\theta$$

$$m_2 \ddot{y}_2 = F - m_1\, g \sin\theta \qquad \sin\theta = \frac{x_2 - x_1}{l} \qquad (6.1.1)$$

Where:

g	is the acceleration due to gravity
θ	is the angle of the string from vertical
L	is the length of the pendulum

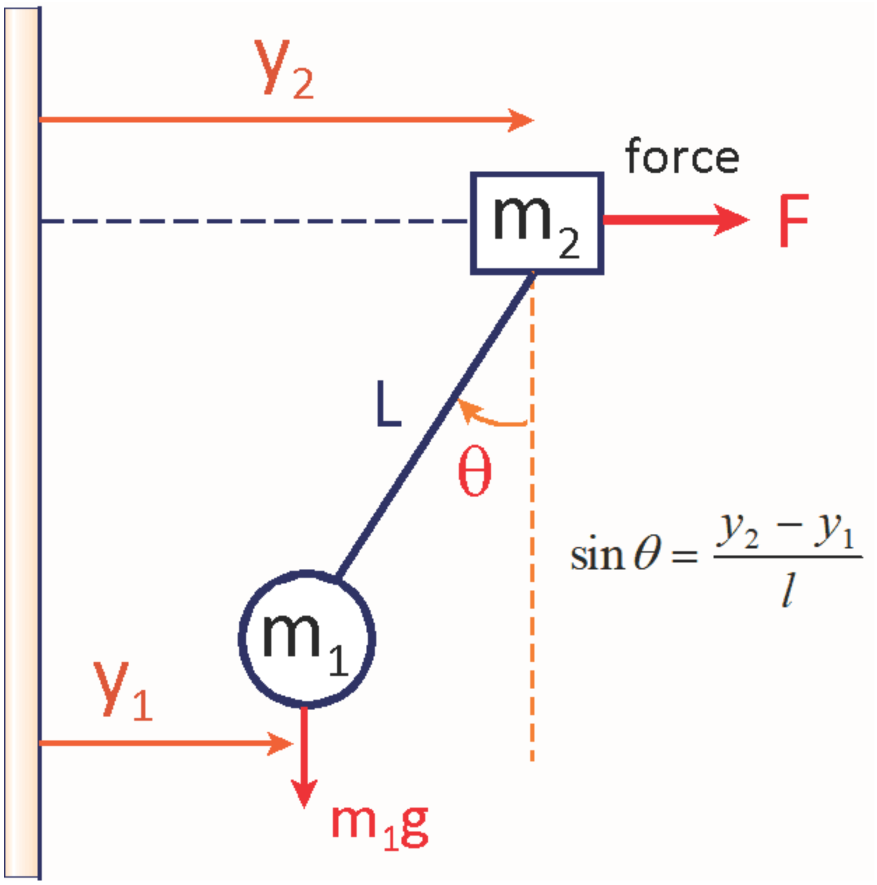


Figure 6.1.1 Simple Overhead Crane Plant Model

The design requirement for this plant is to control the position y_1 of the bottom mass m_1 by applying a control force F on the top mass m_2 and controlling its motion y_2. From equations 6.1.1 we can write the plant equations in state space form, assuming that g/l =1 and m_1=m_2

$$\underline{\dot{x}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \underline{x} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} u + G\, w \qquad (6.1.2)$$

Where:

<u>x(t)</u>	is the state vector, <u>x</u>= [y_1, y_2, $\dot{y}_1, \dot{y}_2$]
u(t)	is the control force F
w(t)	is the process noise vector

The output vector in equation 6.1.3 consists of two deterministic measurements: the position y_1 of the mass m_1 and the pendulum angle θ of the rope from vertical.

$$\underline{z} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \end{bmatrix} \underline{x} + \underline{v} \qquad (6.1.3)$$

Where: <u>v</u> is a zero mean white measurement noise vector

The Analysis

In this example we shall apply the Linear Quadratic Regulator method to design a positioning control system for the bottom mass using the two output measurements (y1, θ) to calculate the control force on m_2. The LQR will guarantee a stable solution, but since we want to control the position y_1 of the bottom mass m_1 we will introduce one additional state in the design model (y_1-integral) and we will design a state-feedback controller for the augmented 5-state plant. However, we cannot directly apply state-feedback because most of the states are not measurable, only y1 and θ are. We will therefore design an estimator for the four state vector, <u>x</u>= [y_1, y_2, $\dot{y}_1, \dot{y}_2$] and apply the state-feedback from the estimated states plus the y_1-integral which is known and it does not have to be estimated. We will use Flixan to generate the dynamic models for design and analysis, and design the LQR state-feedback and the Kalman-Filter observer. Then we will analyze the control system performance in Matlab using simulations and perform frequency response analysis.

The Flixan Files

The files for the Overhead Crane example are in subdirectory: "*Flixan\LQG\Examples\Crane*". The input file is "*Crane.Inp*" and contains the Flixan datasets for generating the plant models, calculating the steady-state LQR gains, and the Kalman-Filter. The crane design model is "*Overhead Crane Design Model*". It is augmented by including the y_1-integral state which is intended to improve control of the y_1 position. The augmented system title is "*Crane Design Model with Y1 Integral*" and it is used to design the LQR 5-state-feedback gain Kc1. The 3 outputs of matrix C (y_1-integral, y_1, and θ) are penalized in the LQR optimization via matrix Qc2. The control force is penalized via the scalar Rc to achieve a satisfactory trade-off between speed of convergence and force usage. The matrices and systems are in file "*Crane.Qdr*". The batch set is used to process the entire file.

```
BATCH MODE INSTRUCTIONS ...............
Batch to prepare models for the Overhead Crane Analysis and Design
! This batch Generates Dynamic Models, LQR State-Feedback Control,
! and Kalman-Filter Gain and Estimator for the Overhead Crane
!
! Retain the Old System and Matrices
Retain System    : Overhead Crane Design Model
Retain System    : Overhead Crane Analysis Model
Retain Matrix    : Output Weight Matrix Qc2
Retain Matrix    : Control Weight Matrix Rc
Retain Matrix    : State Weight Matrix Qc4
Retain Matrix    : Process Noise Covariance Matrix Qpn4
Retain Matrix    : Measurement Noise Covariance Rmn2
!
!                  Control and Estimator Design
Transf-Function  : Integrator
System Connection: Crane Design Model with Y1 Integral
LQR Control Des  : LQR Control Design 1 for Crane Design Model with Y1 Integral
State Estimator  : Kalman-Filter Design 1 for Overhead Crane Design Model
!
!                  Convert the Design and Analysis Models and Gains for Matlab Analysis
To Matlab Format : Crane Design Model with Y1 Integral
To Matlab Format : Overhead Crane Analysis Model
To Matlab Format : LQR State-Feedback Control 1 for Crane Design Model with Y1 Integral
To Matlab Format : Kalman-Filter Estimator 1 for Overhead Crane Design Model
-------------------------------------------------------------------------------------------------

SYSTEM OF TRANSFER FUNCTIONS ...
Integrator
! Integrates the Mass-1 Displacem Y1
!
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
...........................
Definitions of Inputs  =   1
Mass-1 Displacem (y1)

Definitions of Outputs =   1
Integral od Mass-1 Displacem (y1-integr)
------------------------------------------------------------------------------
```

```
INTERCONNECTION OF SYSTEMS .....
Crane Design Model with Y1 Integral
! Creates an Augmented plant for control Design by including the integral
! of mass-1 displacement in the states and output.
!
Titles of Systems to be Combined
Title 1 Overhead Crane Design Model
Title 2 Integrator
SYSTEM INPUTS TO SUBSYSTEM  1                                                Plant(s)
System Input  1 to Subsystem  1, Input  1, Gain= 1.0                         Control Force
..............................................................................
SYSTEM OUTPUTS FROM SUBSYSTEM  2                                             Integrator
System Output  1 from Subsystem  2, Output  1, Gain= 1.0                     y1 integral
..............................................................................
SYSTEM OUTPUTS FROM SUBSYSTEM  1                                             Plant Outputs
System Output  2 from Subsystem  1, Output  1, Gain= 1.0                     y1 displ
System Output  3 from Subsystem  1, Output  2, Gain= 1.0                     theta
..............................................................................

SUBSYSTEM NO  1 GOES TO SUBSYSTEM NO  2                                      Plant Outp to
Control Input
Subsystem  1, Output  1 to Subsystem  2, Input  1, Gain= 1.0                 y1 displacem
..............................................................................
Definitions of Inputs  =   1
Disturbance Force   (Fdist)

Definitions of Outputs =   3
Mass-1 Displacem-Integral (y1-int)
Mass-1 Displacement       (y1)
Pendulum Angle            (theta)
-------------------------------------------------------------------------------------------------

LINEAR QUADRATIC REGULATOR STATE-FEEDBACK CONTROL DESIGN
LQR Control Design 1 for Crane Design Model with Y1 Integral
! State-Feedback Control Design for the Augmented 5-state Crane Model
! using the output matrix C in the optimization criteria
!
Plant Model Used to Design the Control System from:      Crane Design Model with Y1 Integral
Criteria Optimization Output is Matrix C
State Penalty Weight (Qc) is Matrix:   Qc2               Output Weight Matrix Qc2
Control Penalty Weight (Rc) is Matrix: Rc                Control Weight Matrix Rc
Continuous LQR Solution Using Laub Method
LQR State-Feedback Control Gain Matrix Kc1               LQR State-Feedback Control 1 for Crane
-------------------------------------------------------------------------------------------------

KALMAN-BUCY FILTER STATE ESTIMATOR DESIGN
Kalman-Filter Design 1 for Overhead Crane Design Model
! State Observer for the Original 4-state Crane Model, Estimating
! Positions and Velocities of the two masses from the plant output
!
Plant Model Used to Design the Kalman-Filter from:       Overhead Crane Design Model
Input Process Noise Matrix (G) is the  Identity
Process Noise Covariance Qpn is Matrix Qpn4              Process Noise Covariance Matrix Qpn4
Measurement Noise Covariance is Matrix Rmn2              Measurement Noise Covariance Rmn2
Kalman-Filter Estimator is Gain Matrix Kf1               Kalman-Filter Estimator 1 for Overhead
-------------------------------------------------------------------------------------------------
```

The estimator uses the original 4-state plant model: "*Overhead Crane Design Model*" which does not include the y_1-integral. The Flixan program calculates the Kalman-Filter gain Kf1 which is exported to Matlab and used in the observer simulation to estimate the 4 states from the outputs y_1 and θ. The noise covariance matrices Qpn4 and Rmn2 are located in the systems file "*Crane.Qdr*". Matlab conversion datasets are included at the bottom of the input file to create m-files for the gains and systems that can be loaded into Matlab.

```
CONVERT TO MATLAB FORMAT ........      (Title, System/Matrix, m-filename)
Overhead Crane Analysis Model
System
Analysis_Plant
-------------------------------------------------------------------------------------------------
CONVERT TO MATLAB FORMAT ........      (Title, System/Matrix, m-filename)
Crane Design Model with Y1 Integral
System
Design_Plant_Int
-------------------------------------------------------------------------------------------------
CONVERT TO MATLAB FORMAT ........      (Title, System/Matrix, m-filename)
LQR State-Feedback Control 1 for Crane Design Model with Y1 Integral
Matrix Kc1
-------------------------------------------------------------------------------------------------
CONVERT TO MATLAB FORMAT ........      (Title, System/Matrix, m-filename)
Kalman-Filter Estimator 1 for Overhead Crane Design Model
Matrix Kf1
-------------------------------------------------------------------------------------------------
```

Simulation Models

Figure 6.1.2 is a simulation model "*Crane_Sim-1.mdl*" used to test the state-feedback gain Kc1 directly from the four states: $\underline{x}$= [y_1, y_2, $\dot{y}_1, \dot{y}_2$] which they are not measurable, but it is intended to check out the control design. Figure 6.1.3 shows the system's response to y_1 displacement command, which is logically what a person would do naturally using common sense. First, move the top mass as fast as possible half way towards the intended position and stop for a short period waiting for the bottom mass to swing. The bottom mass does not immediately feel the motion until the pendulum angle θ is big enough. When the bottom mass swings over to the opposite extreme of the pendulum angle $-\theta$, which is very close to the intended position, the top mass is immediately moved above the target position to prevent it from oscillating further. This is essentially what the LQR control system does in Figure 6.1.3 but it also takes into consideration the limited control system bandwidth. This requires knowledge of the pendulum frequency which is captured in the design plant model and subsequently in the control design to dampen out the pendulum oscillations. Notice the "hick-up" in the y_2 response as it waits for the bottom mass to swing in the opposite side of the pendulum.

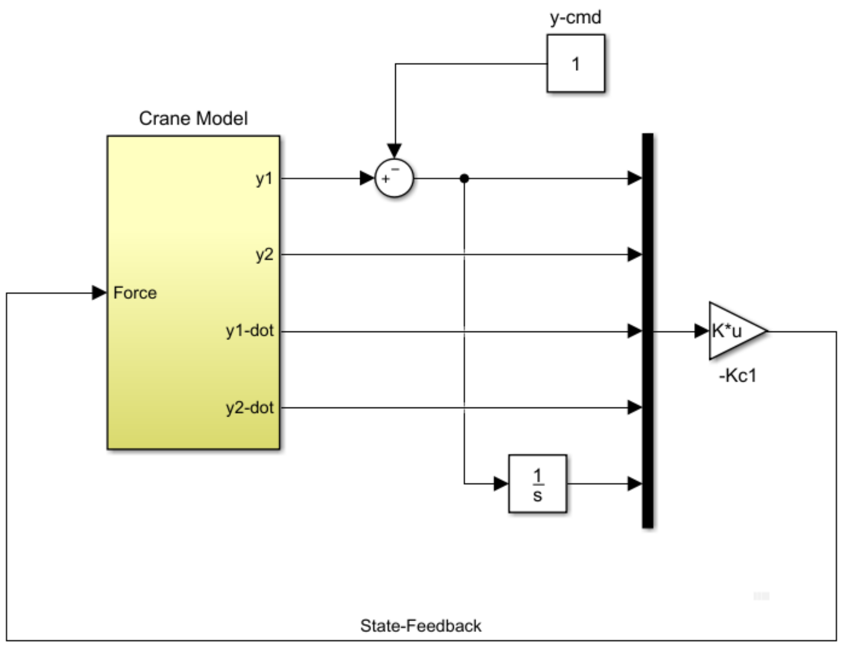


Figure 6.1.2 State-Feedback Simulation model "*Crane_Sim-1.mdl*"

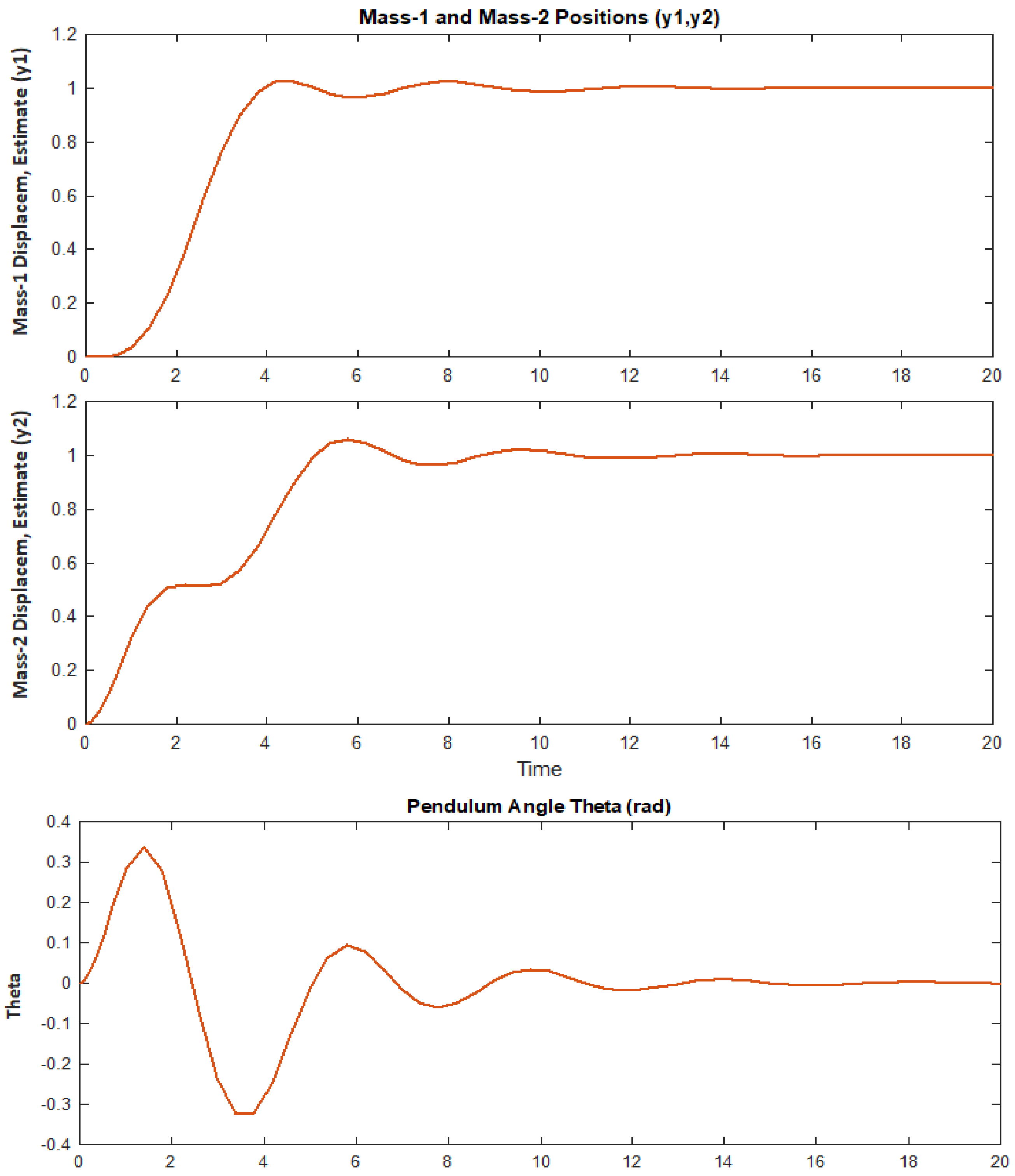


Figure 6.1.3 System's Response to a Displacement Command y1-command

But in reality the state vector is not available, that is why we designed the Kalman-Filter observer to estimate the states for feedback. The simulation in Figure 6.1.4 shows the control system which includes the state estimator in detail below. The inputs to the estimator are the two measurements: y_1, and θ, and also the control force. The outputs are the four states. The y1-itegral is not included because it is measurable. The file init.m loads the Flixan generated systems and matrices into Matlab for the analysis.

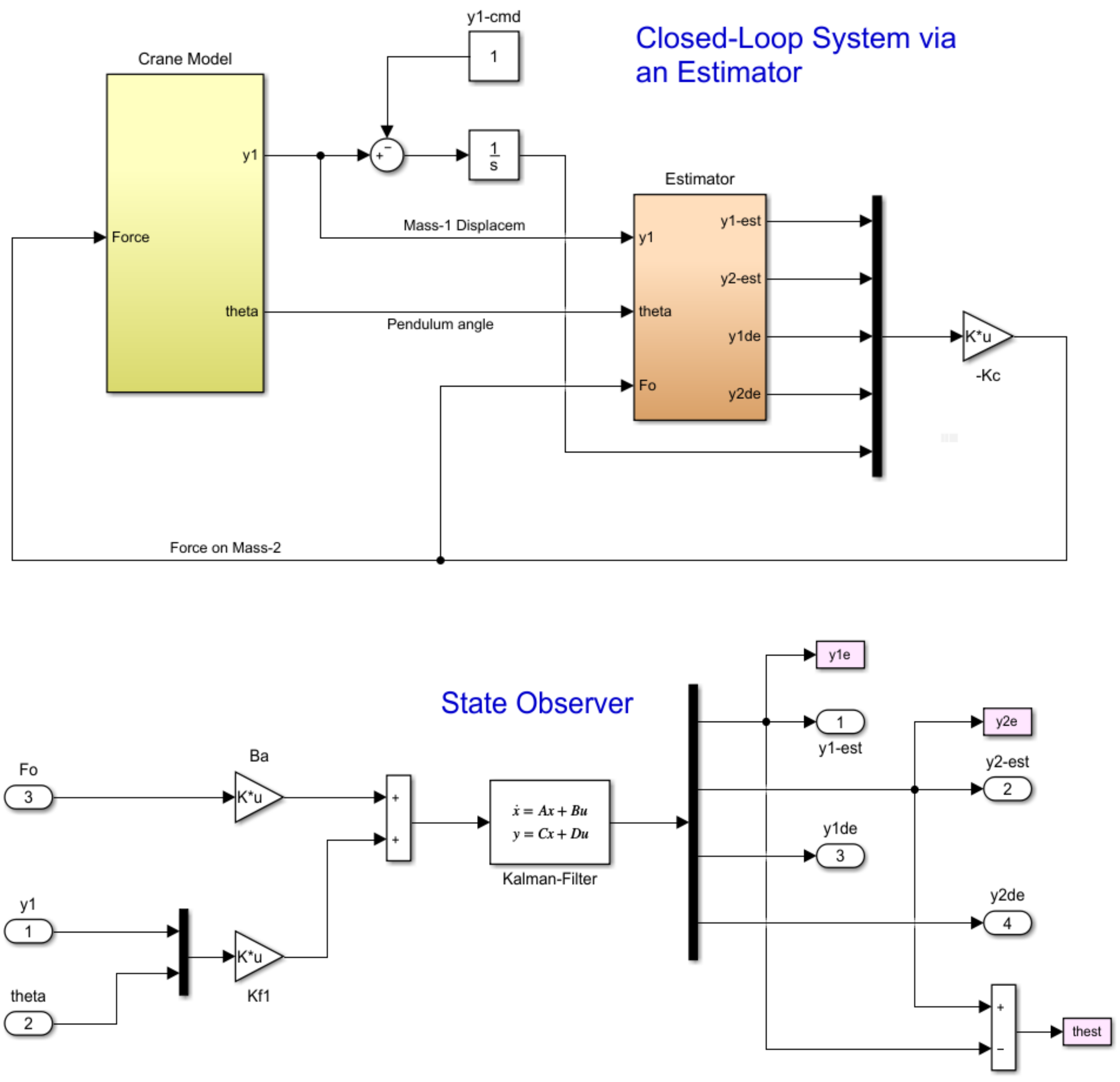


Figure 6.1.4 Output Feedback Simulation model "Crane_Sim-2.mdl" that includes the Estimator

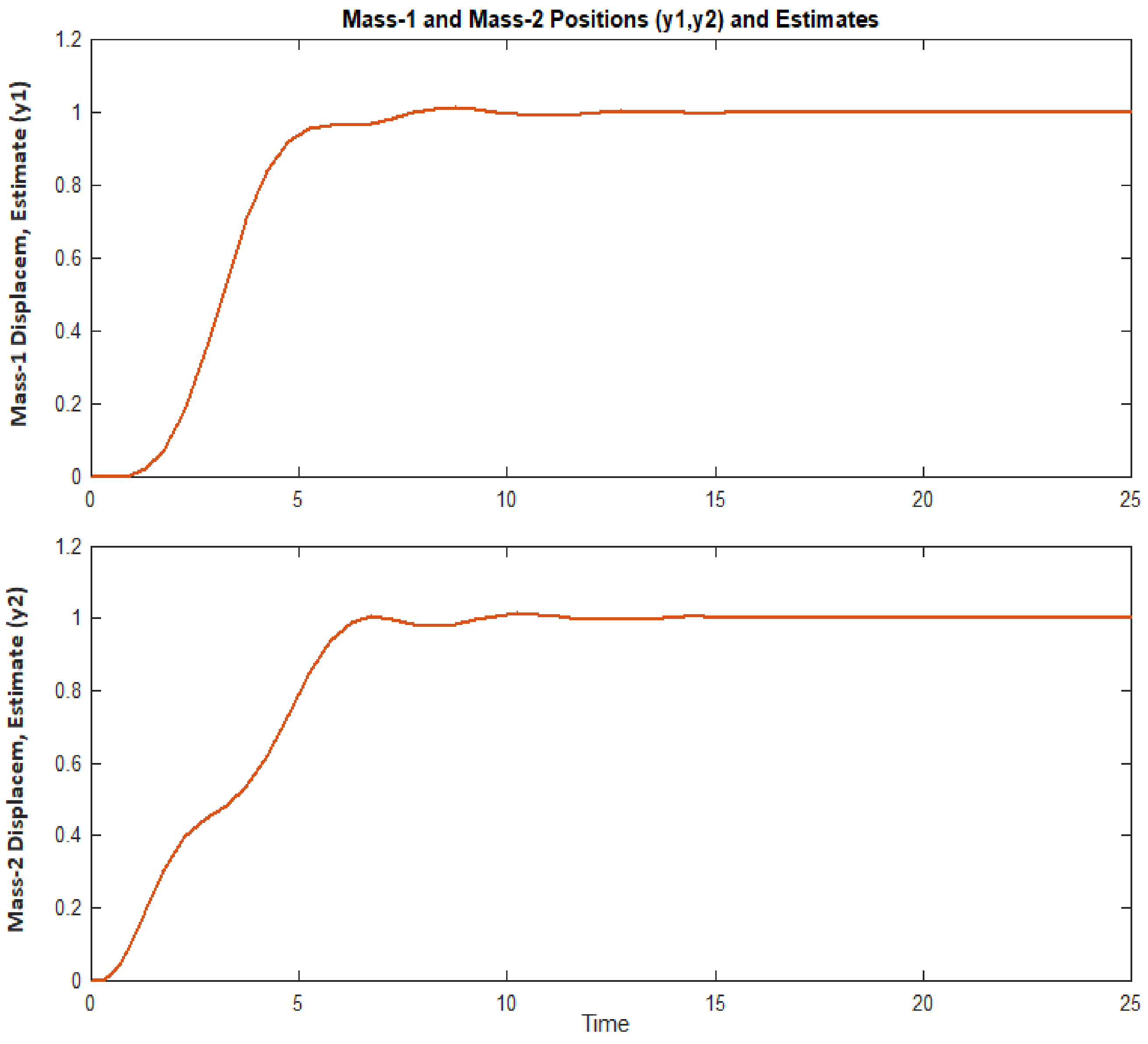


Figure 6.1.5a Response of System "Crane_Sim-2.mdl" to y_1 Displacement Command

Figure 6.1.5 shows the response of the output-feedback system which is not very different from the state-feedback system. The estimator changes slightly the response. The hick-up on the y2 displacement is not as intense as in the state-feedback case and the oscillation damping is slightly faster. Also, the reverse force amplitude is not as high as the forward force and it is applied for a longer period.

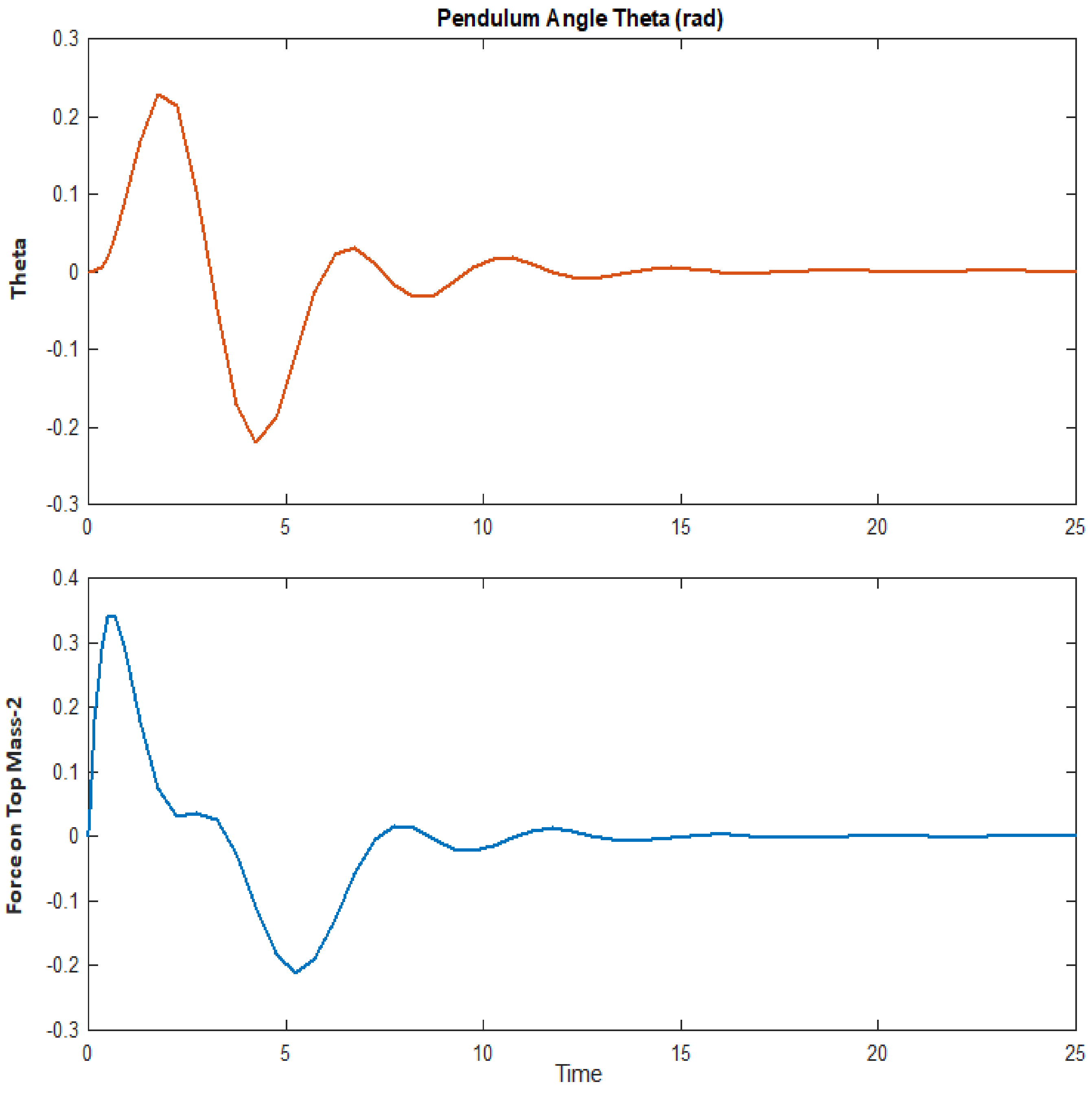


Figure 6.1.5b Response of System "Crane_Sim-2.mdl" to y1 Displacement Command

Frequency Response Analysis

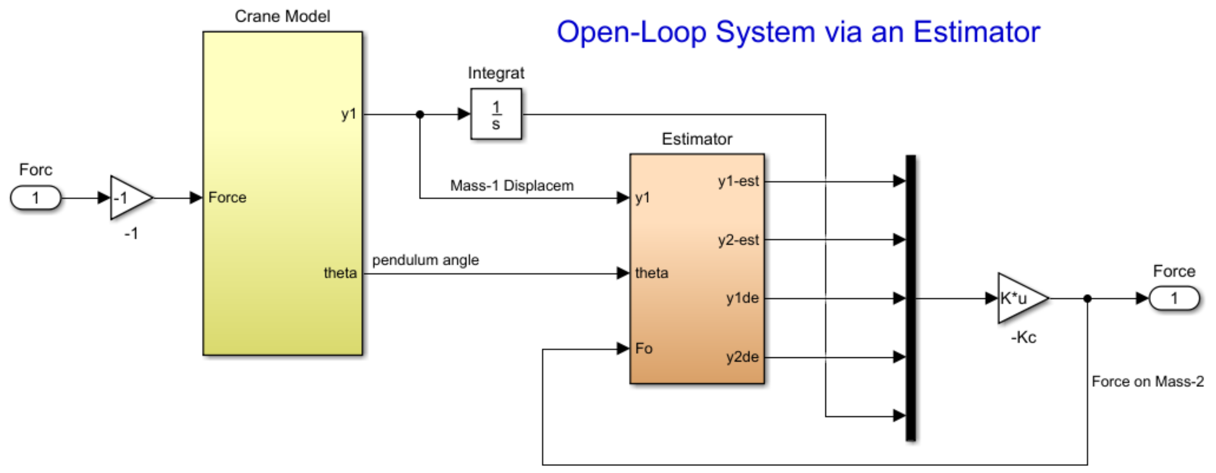
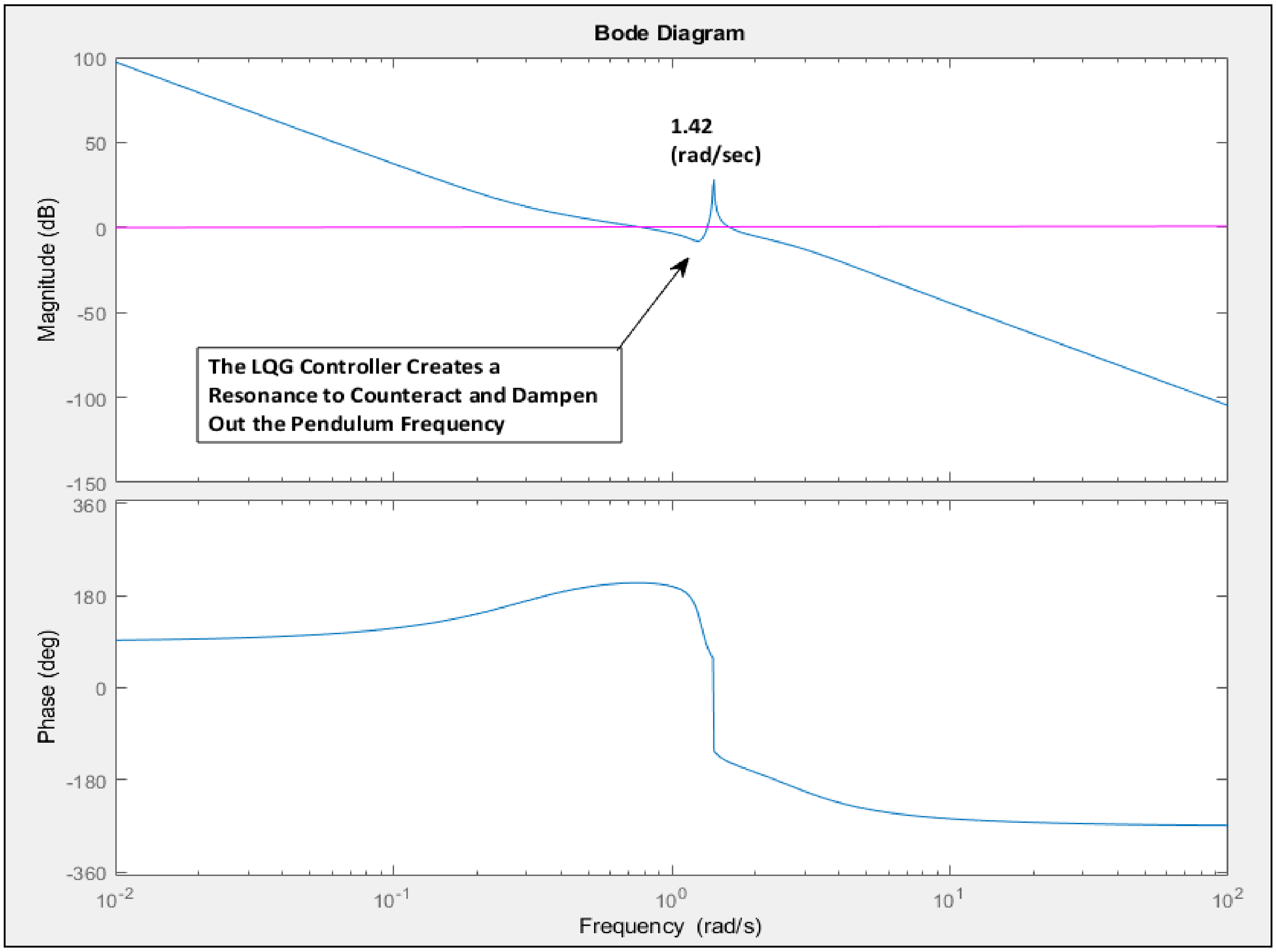


Figure 6.1.6 Frequency Response Analysis Model "Open_Loop.Mdl"



Frequency response analysis is used to check out the control system's stability in terms of gain and phase margins. The Simulink model "*Open_Loop.Mdl*" in Figure 6, that has the loop opened at the plant force input, is used to calculate the frequency response. Figure 6.1.7 shows the Bode and Nichols plots including the stability margins. Notice that the system has resonance of considerable amplitude at 1.42 (rad/sec) which is the pendulum frequency. This is how the control system counteracts the natural pendulum frequency by introducing an anti-resonance at the same frequency since it is designed around the plant model. Like we said earlier, the system needs to know how long to wait for the hick-up in order to counteract the natural frequency. Figure 6.1.7 below shows the phase and gain margins before and after the resonance and they are reasonable for stability.

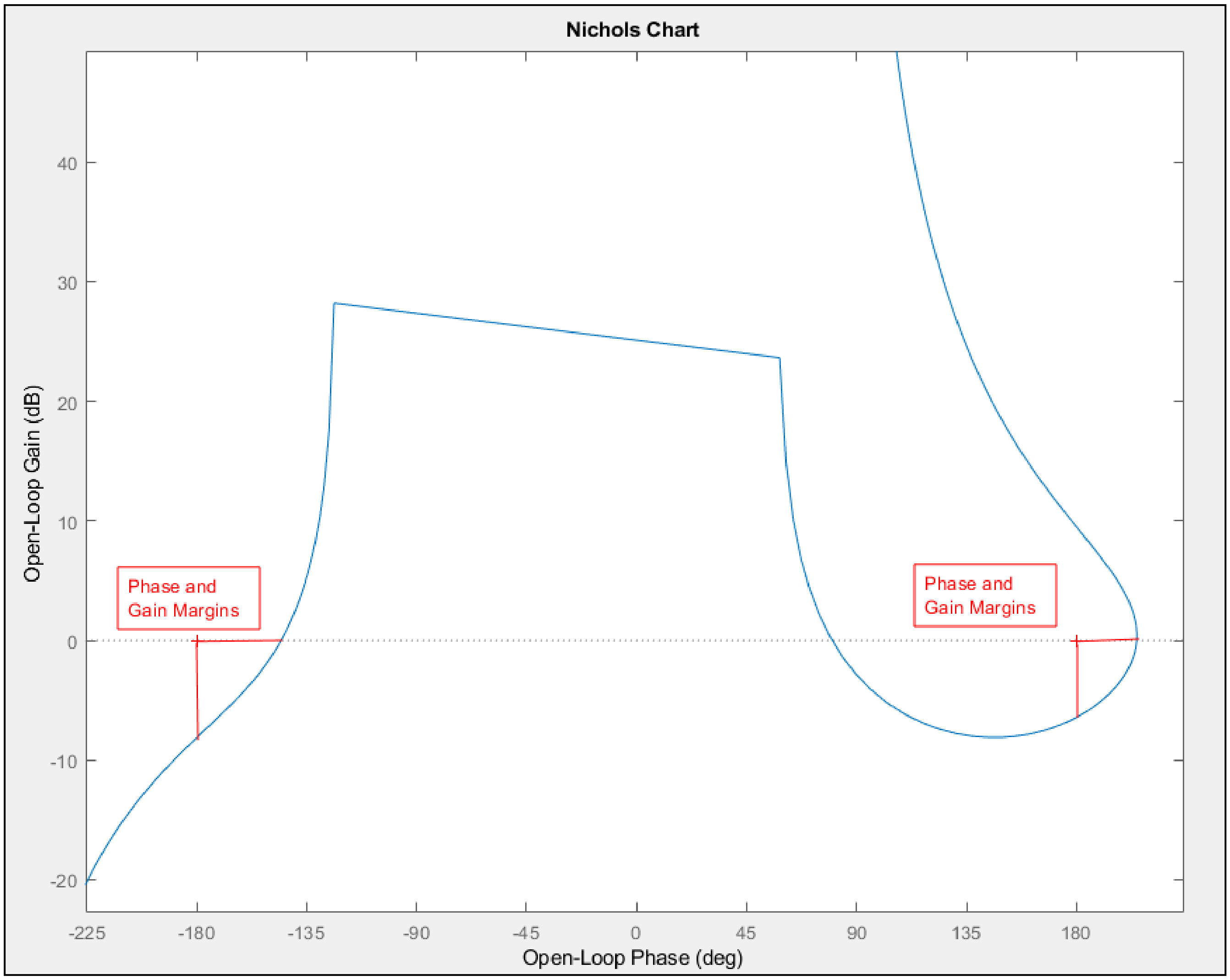
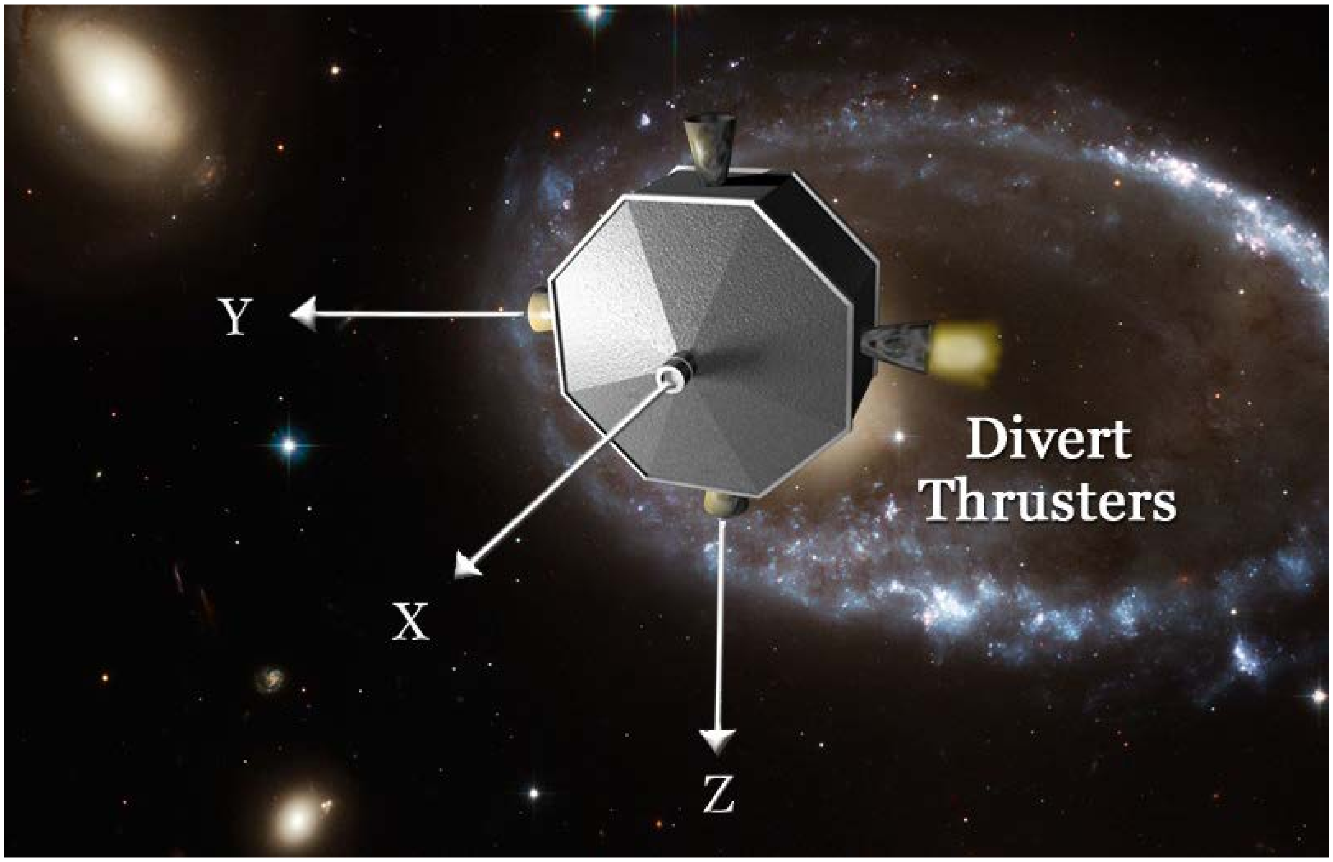


Figure 6.1.7 Open-Loop Frequency Response Analysis, Bode and Nichols Plots

6.2 Design of a Space Interceptor



In this example we will analyze a guided intercept between two space vehicles: an interceptor which is a kinetic vehicle and a target that may be a meteorite heading towards the earth, space debris, or an enemy missile. We assume that the interceptor has already been placed in a collision course with the target by a mid-course booster rocket and the vehicle is no longer accelerating but drifting towards the target. Its translational motion is controlled only in two directions (y and z) by firing divert thrusters perpendicular to the x-axis. The end-game is a dynamic engagement using closed-loop guidance to improve impact precision and probability, especially when the target is randomly accelerating in order to avoid getting hit. The interceptor uses an optical sensor to track the target and its line-of-sight (LOS) is always pointing towards the target. It has an Attitude Control System to track the target at the center of the field of view by maneuvering its attitude and aligning the x-axis with the target.

If the target is not maneuvering and if the mid-course boost was executed perfectly, the target would remain in the center of the seeker's field of view all the way to impact. If the seeker detects an error or the target is moving perpendicular to the LOS, the guidance will fire the corresponding divert thrusters to produce the necessary acceleration that will zero the error. It is assumed that the approximate relative position, velocity, and acceleration of the target are calculated from the seeker azimuth and elevation measurements, and from the target distance which is estimated from navigation and used in the End-Game algorithm.

6.2.1. End-Game Dynamic Model

The dynamic model in this Section describes the relative motion between the interceptor spacecraft and the target. The relative motion of the two spacecraft can be described by two sets of equations: one describing the relative motion along the x-direction which is along the main velocity direction and the LOS, and another set of equations that describe the motion perpendicular to the LOS. The motion along the LOS which is along the line joining the two spacecraft is uncontrollable because the interceptor has no thrust in the x-direction and the relative motion equation is only used to calculate the time to impact (t_{go}). The relative motion perpendicular to the LOS along the y and z directions is controlled by the interceptor's divert thrusters and it is identical in both perpendicular directions.

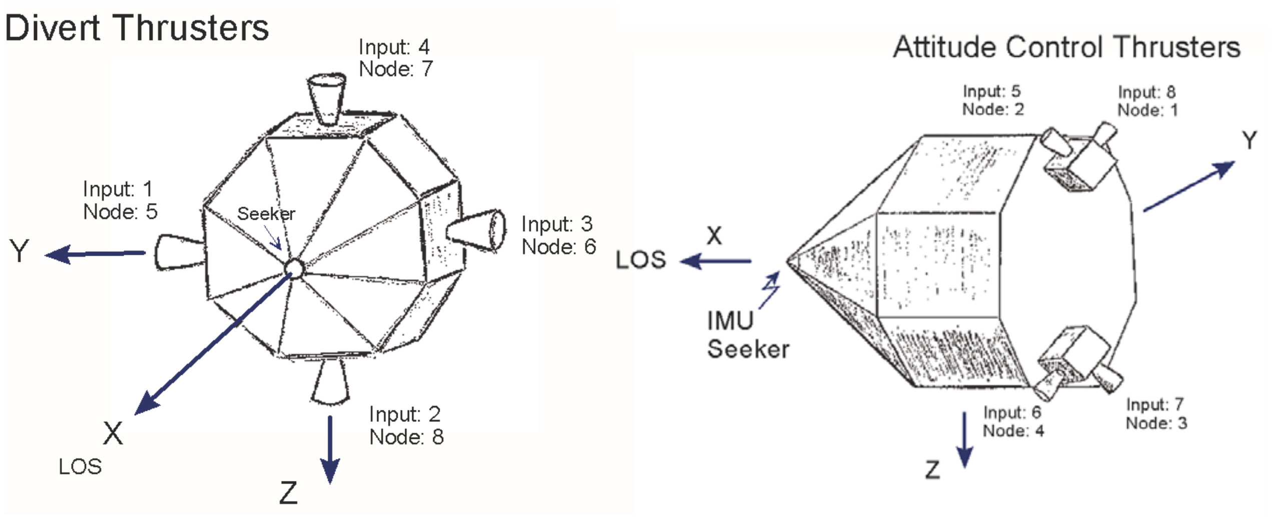


Figure 6.2.1 Interceptor Spacecraft

The time-to-go calculation t_{go} for an accelerating target in equation 6.2.1 is calculated from the estimated acceleration A_x, relative velocity V_x, and the distance to target R. If the target is not accelerating the time-to-go simplifies to: $t_{go} = R/V_x$

$$t_{go} = \frac{-V_x + \sqrt{V_x^2 - 2RA_x}}{A_x} \qquad (6.2.1)$$

Equation 6.2.2 describes the relative spacecraft motion perpendicular to the LOS (either y or z directions). It is controlled by the divert thrusters that provide the interceptor acceleration A_I. It describes the motion in the local inertial frame which is defined by the position of the interceptor at the initialization time, when it initially detects the target.

$$\begin{pmatrix} \dot{S}_r \\ \dot{V}_r \\ \dot{A}_T \\ \dot{A}_K \end{pmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -W_T & 0 \\ 0 & 0 & 0 & -W_I \end{bmatrix} \begin{pmatrix} S_r \\ V_r \\ A_T \\ A_I \end{pmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & W_T \\ W_I & 0 \end{bmatrix} \begin{pmatrix} A_{I\,com} \\ A_{T\,com} \end{pmatrix} \tag{6.2.2}$$

The state vector consists of four states:

S_r	is the relative vehicle position (target – interceptor)
V_r	is the relative vehicle velocity (target – interceptor)
A_T	is the target acceleration normal to the LOS
A_I	is the interceptor acceleration perpendicular to the LOS

The two inputs are:

A_{Icom}	Interceptor Acceleration Command perpendicular to the LOS
A_{Tcom}	Target Acceleration Command perpendicular to the LOS

The dynamic model in equation 6.2.2 captures the maneuverability of the two vehicles perpendicular to the LOS and introduces it in the control design. In addition to relative position and velocity it includes the target and interceptor bandwidths described by first order lags of frequencies W_T and W_I respectively, where: W_T=5 (rad/sec) and W_I=100 (rad/sec). Naturally the interceptor must have a broader bandwidth than the target because it is smaller in size. S_r and V_r are the target's position and velocity relative to the kill vehicle perpendicular to the LOS.

The control guidance of the interceptor must sense the relative motion of the target perpendicular to its x-axis using the optical sensor and apply the proper acceleration command to the thrusters in order to null-out the relative position at the estimated impact time. We can apply the Linear Quadratic Regulation to calculate the state-feedback control law that will take out the relative position at the expected impact. However, we must take into consideration two additional issues. The first issue is that there is a significant amount of noise in the measurement, especially when the distance-to-go is large. We don't want the kill vehicle to be chasing noise because it will consume its propellant fast. Therefore, we need a control system with variable bandwidth. Starting at low bandwidth for fuel efficiency and increasing it inversely proportional with time-to-go in order to improve performance near impact, where it is needed more and the signal to noise ratio is good. The second issue to be considered in the design is the uncertainty in the t_{go} calculation which is based on navigation measurements and subject to delays. We want a successful hit even if it occurs a little sooner or a little later than the expected time. One way to improve success is to reduce the relative side velocity V_r to zero a short time prior to the estimated impact time, and maintain a high gain system all the way to impact.

6.2.2. Optimal Control Design

The previously described design requirements can be captured in the performance index of the Linear Quadratic Regulator algorithm. After simplifying the state-space representation of the engagement model and assuming that all states are available for feedback, the performance index J is defined as follows

$$\dot{\underline{x}} = A\underline{x} + Bu$$

$$J = \int_0^{t_f} \left(\underline{x'} Q \underline{x + R u^2} \right) dt + \underline{x}(t_f)' P_1 \underline{x}(t_f) \qquad (6.2.3)$$

Where: the matrices Q and P1 and the scalar R in the performance index equation 6.2.3 are weights that trade control acceleration versus system performance to disturbances.

Q	is a positive semidefinite matrix that penalizes the state error along the trajectory
R	is a scalar that penalizes the control along the trajectory which is related to fuel
P_1	is a positive semidefinite matrix that penalizes the state vector error only at the final time t_f

They are selected to achieve a satisfactory trade-off between fuel consumption and robustness to miss distance errors, in the presence of seeker noise and range measurement errors. During the early part of the trajectory where the solution is steady-state, we avoid penalizing much the position and velocity errors perpendicular to the x-axis in the Q matrix. The terminal position and velocity states are heavily penalized by matrix P_1, because reducing the perpendicular components of the relative velocity to almost zero at impact, makes the optimal control law less sensitive to range errors.

The optimal state-feedback control law is obtained by synthesizing the time-varying LQR problem around the plant model of equation 6.2.2. Since we cannot control the target motion but only the interceptor's, for control design we ignore the second input to the dynamic model and keep only the A_{Icom} input. The target acceleration input will be used in the analysis. The last term in the performance index equation that includes the matrix P_1 produces the time-varying state-feedback gain. The matrix P_1 penalizes the terminal position and velocity, and by adjusting the velocity coefficient we can reduce the terminal velocity V_r to almost zero at impact, that is, in addition to the relative position S_r. The LQR solution in equation 6.2.4 calculates a time varying state-feedback gain matrix $K_c(t)$ that optimizes the performance index of equation 6.2.3 and satisfies the design requirements.

$$u^0(t) = -R^{-1} B^T P(t) \underline{x}(t) = -K_c(t) \underline{x}(t)$$

$$-\dot{P}(t) = Q - P(t) B R^{-1} B^T P(t) + P(t) A + A^T P(t) \qquad (6.2.4)$$

$$where\colon 0 < t < t_f$$

The time-varying matrix P(t) is always positive definite and symmetric and is obtained by solving the transient Riccati equation 6.2.4b. Its terminal value at impact is equal to the value of the terminal state weight matrix P_1, i.e. $P(t_f)=P_1$. This property is used for solving the Riccati equation numerically after initializing it at the terminal time t_f and integrating backwards in time to t=0. The resulting control law is a time varying state-feedback that provides normal acceleration to the interceptor as a function of the four states, which at this point we assume that they are all available for feedback. In our next step we will design a Kalman-Filter to estimate the states from the system output. The same control law is used for both: the y and z axes, since the spacecraft is symmetric and there is no coupling between the y and z directions. It is important to mention that the final time t_f is not necessarily the impact time but it may be a time before impact that we wish to switch control laws, to Proportional Navigation, for example. The terminal goal in this case may be to achieve favorable conditions for PN initialization. The optimal control design boils down to choosing a satisfactory trade-off between two scalars in equation 6.2.5: the fuel weight R and the terminal state weight p. A large R penalizes the fuel usage. The larger p gets, the more the final normal relative position and velocity are reduced close zero at impact. This, however, is achieved at the expense of propellant consumption or that the control demand may exceed the maximum acceleration capability of the interceptor's thrusters. T_{gi} is a short period before impact when you expect the position and velocity to converge to zero.

$$Q = diag\begin{pmatrix}0.1 & 0.1 & 0 & 0.1\end{pmatrix};\ R = 1$$

$$P_1 = p\begin{bmatrix} 1 & T_{gi}/2 & 0 & 0 \\ T_{gi}/2 & T_{gi}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad (6.2.5)$$

3. State Estimator Design

The LQR control law requires feedback from the state variables. However, most of the system states are not measurable and our next step is to design a state observer from the two outputs of the dynamic model, the relative position S_r and the interceptor acceleration A_I perpendicular to the LOS. In this section we will present the steady-state Kalman-Bucy filter, an observer that will be used to approximately reconstruct the state vector from the measurements so that we can apply our optimal state-feedback control law. The state observer also requires knowledge of the dynamic model in equation 6.2.2. We shall assume that the system is corrupted by two types of noise: state excitation noise, and measurement noise, as shown in equation 6.2.6. They are white noise, zero mean, and uncorrelated.

$$\dot{\underline{x}}(t) = A\,\underline{x}(t) + B u(t) + G\,w(t)$$

$$\underline{y}(t) = C\,\underline{x} + \underline{v}(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} S_r \\ V_r \\ A_T \\ A_I \end{pmatrix} + \underline{v}(t) \qquad (6.2.6)$$

Where:

$\underline{x}(t)$	is the state vector of dimension n
$\underline{u}(t)$	is the control input vector of dimension m
$\underline{y}(t)$	is the measurement vector of dimension r ($r \le n$)
$\underline{w}(t)$	is the process noise of dimension l, where ($l \le n$) and has a covariance matrix Q_{pn}, where: $Q_{pn} = Q_{pn}' \ge 0$
$\underline{v}(t)$	is the measurement noise with intensity $R_{mn} = R_{mn}' \ge 0$

The solution to the Kalman Filter is obtained by minimizing the quantity in equation 6.2.7, where the matrix W is (nxn) positive semi-definite. The state vector estimate $\underline{\hat{x}}$ from the KF output will converge to the actual system state $\underline{x}$. Figure 6.2.2 is a functional block diagram representation of the Kalman-Filter showing the output and its interconnection with the plant input $\underline{u}$ and output $\underline{y}$.

$$J = \lim E\left[\left(\underline{x}(t) - \underline{\hat{x}}(t)\right)' W \left(\underline{x}(t) - \underline{\hat{x}}(t)\right)\right] \qquad as\ t \to \infty \qquad (6.2.7)$$

The estimate is obtained by solving the following differential equation 6.2.8, where K_f is the Kalman-Filter gain. The solution exists when the pair (A',C') is stabilizable and the pair (A, $GQ_{pn}G^T$) is detectable. The state estimate is initialized with the expected initial state vector at t=0, $\hat{x}(0) = E[\underline{x}(0)]$.

$$\begin{aligned} \dot{\hat{\underline{x}}}(t) &= A\hat{\underline{x}}(t) + Bu(t) + K_f\left[y(t) - C\hat{\underline{x}}(t)\right] \\ K_f &= PC^T R_{mn}^{-1} \end{aligned} \qquad (6.2.8)$$

The matrix P is symmetric positive semi-definite and it is obtained from the steady-state solution of the asymptotic Riccati equation

$$AP + PA^T + GQ_{pn}G^T - PC^T R_{mn}^{-1} CP = 0 \qquad (6.2.9)$$

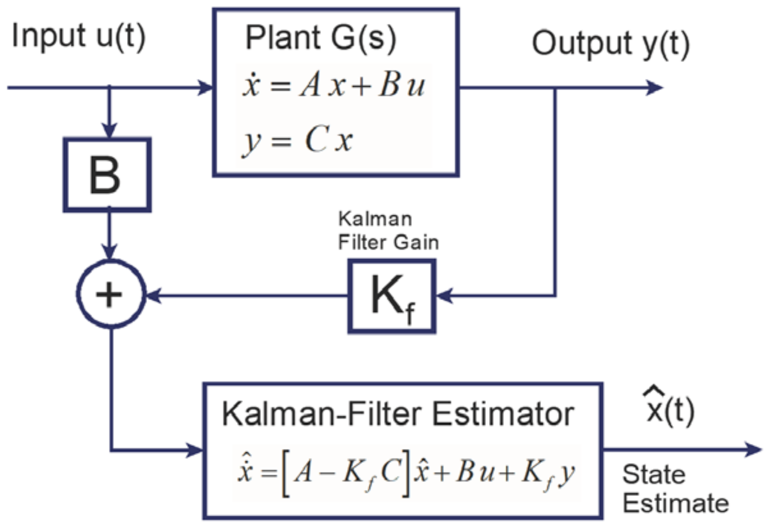


Figure 6.2.2 Kalman-Filter State-Vector Estimator

6.2.4. Continuous System Analysis

The analysis files for the continuous system in this example are located in folder: "*LQG\Examples\End-Game\Continuous*". There is also a discrete subfolder "*Discrete*" for the discrete-time analysis using dynamic models which are discretized at 40 (msec) sampling time. The directory includes the input file "*End_Game_s.Inp*", shown below, that contains input data for the continuous LQR design using Flixan. That is, for the steady-state LQR control, the time-varying state-feedback LQR, the Kalman-Filter, and for the steady-state output-feedback LQG design that combines the steady-state LQR gain K_c and the Kalman-Filter gain K_f to a dynamic output-feedback controller. The input file also includes a batch set for fast data processing and datasets for Matlab conversions. The systems filename "*End_Game_s.Qdr*" contains the End-Game dynamic model described in Equation 6.2.2, the control design matrices: Q_c, R_c, and P_1, the Kalman-Filter design matrices Q_{mn}, R_{mn}, the control gain K_c and the Kalman-Filter gain K_f which are generated by the Flixan LQR design program.

```
BATCH MODE INSTRUCTIONS ...............
Batch for preparing End-Game Control design Models
! This batch Generates LQR State-Feedback, Kalman-Filter and Output
! Feedback Dynamic Controller
Retain System    : Simple End-Game Model
Retain Matrix    : State Weight Matrix Qc (4x4)
Retain Matrix    : Output Weight Matrix Qc (2x2)
Retain Matrix    : Control Weight Matrix Rc
Retain Matrix    : Terminal State Weight Matrix P1 (4x4)
Retain Matrix    : Performance Criteria C1
Retain Matrix    : Output Performance Weight Matrix Qc3
Retain Matrix    : Measurement Noise Matrix Rmn (2x2)
Retain Matrix    : Process Noise Matrix Qpn (4x4)
!
LQR Control Des  : LQR Control Design for Simple End-Game Model
State Estimator  : Kalman-Filter Design for Simple End-Game Model
LQG Control Des  : LQG Control Design for Simple End-Game Model
Transient LQR    : Transient LQR Design for Simple End-Game Model
!
To Matlab Format : Simple End-Game Model
To Matlab Format : LQG Control Design for Simple End-Game Model
To Matlab Format : LQR State-Feedback Control for Simple End-Game Model
To Matlab Format : Kalman-Filter Estimator for Simple End-Game Model
----------------------------------------------------------------------------------------------------------
LINEAR QUADRATIC REGULATOR STATE-FEEDBACK CONTROL DESIGN
LQR Control Design for Simple End-Game Model
! Design the State-Feedback Matrix Kc using the Output
! Criteria Matrix C= Identity
!
Plant Model Used to Design the Control System from:        Simple End-Game Model
Criteria Optimization Output is Matrix Identity
State Penalty Weight (Qc) is Matrix:   Qc4                 State Weight Matrix Qc (4x4)
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 Simple End-
----------------------------------------------------------------------------------------------------------
KALMAN-BUCY FILTER STATE ESTIMATOR DESIGN
Kalman-Filter Design for Simple End-Game Model
! Design the Kalman-Filter Gain Matrix Kf using the
! Process Noise Matrix G = Identity
!
Plant Model Used to Design the Kalman-Filter from:         Simple End-Game Model
Input Process Noise Matrix (G) is the  Identity
Process Noise Covariance Qpn is Matrix Qpn                 Process Noise Matrix Qpn (4x4)
Measurement Noise Covariance is Matrix Rmn                 Measurement Noise Matrix Rmn (2x2)
Kalman-Filter Estimator is Gain Matrix Kf                  Kalman-Filter Estimator for Simple End-Game
----------------------------------------------------------------------------------------------------------
DYNAMIC OUTPUT FEEDBACK LQG CONTROL DESIGN
LQG Control Design for Simple End-Game Model
! Combine State-Feedback with KF Gain to Design a Linear Quadratic
! Gaussian Control System for the Plant: Simple End-Game Model
!
Plant Model Used to Design the Control System from:        Simple End-Game Model
State-Feedback (Kc) is Gain Matrix   : Kc                  LQR State-Feedback Control for Simple End-
Kalman-Filter Estim Kf is Gain Matrix: Kf                  Kalman-Filter Estimator for Simple End-Game
----------------------------------------------------------------------------------------------------------
TRANSIENT LQR CONTROL DESIGN WITH TIME-VARYING GAINS
Transient LQR Design for Simple End-Game Model
! To Generate Time-Varying State-Feedback Gain Kc(t)
! as a Function of Time-to-Go, in File: Gains.dat
!
Plant Model Used to Design the Control System from:        Simple End-Game Model
Criteria Optimization Output is Matrix Identity
State Penalty Weight (Qc) is Matrix:   Qc4                 State Weight Matrix Qc (4x4)
Control Penalty Weight (Rc) is Matrix: Rc                  Control Weight Matrix Rc
Terminal State Penalty Weigh P1 Matrix P1                  Terminal State Weight Matrix P1 (4x4)
Continuous LQR Solution, Final Time, Number of Points:     20.00     800.0
----------------------------------------------------------------------------------------------------------
```

```
CONVERT TO MATLAB FORMAT ........      (Title, System/Matrix, m-filename)
Simple End-Game Model
System
end_game
----------------------------------------------------------------------------------------------------
CONVERT TO MATLAB FORMAT ........      (Title, System/Matrix, m-filename)
LQG Control Design for Simple End-Game Model
System
Control
----------------------------------------------------------------------------------------------------
CONVERT TO MATLAB FORMAT ........      (Title, System/Matrix, m-filename)
LQR State-Feedback Control for Simple End-Game Model
Matrix Kc
----------------------------------------------------------------------------------------------------
CONVERT TO MATLAB FORMAT ........      (Title, System/Matrix, m-filename)
Kalman-Filter Estimator for Simple End-Game Model
Matrix Kf
----------------------------------------------------------------------------------------------------
```

The Flixan LQR transient design program also calculates the time varying state-feedback gain $K_c(t)$ as a function of time-to-go. It requires the weight matrices: Q_c, R_c, and P_1, the initial time-to-go (20 sec), and the number of gain calculation points (800). The gains versus time are saved in file "Gains.dat" with the time-to-go in the first column. This file is used as a look-up table in the simulation. Only the first 4 gains that correspond to the interceptor acceleration command are used in this case. The end-game dynamic model is saved in file "end_game.m", and the steady-state output-feedback control system is saved in "control.m" for Matlab analysis. The gain matrices Kc and Kf are also saved and loaded into Matlab.

6.2.4.1 Simulation Models

We will first analyze the steady state-performance of the spacecraft and then the transient motion with time varying state-state gains. Two simulation models were created to analyze the system's response from some initial relative condition of position, velocity and acceleration.

6.2.4.2 Continuous Steady-State Simulation

The steady-state analysis is applicable when the target is sufficiently far away from the interceptor and the control gains are constant. The Simulink model is "*EndGame_Sim1s.mdl*", shown in Figure 6.2.3, and it is located in the same "*End-Game/ Continuous*" subfolder.

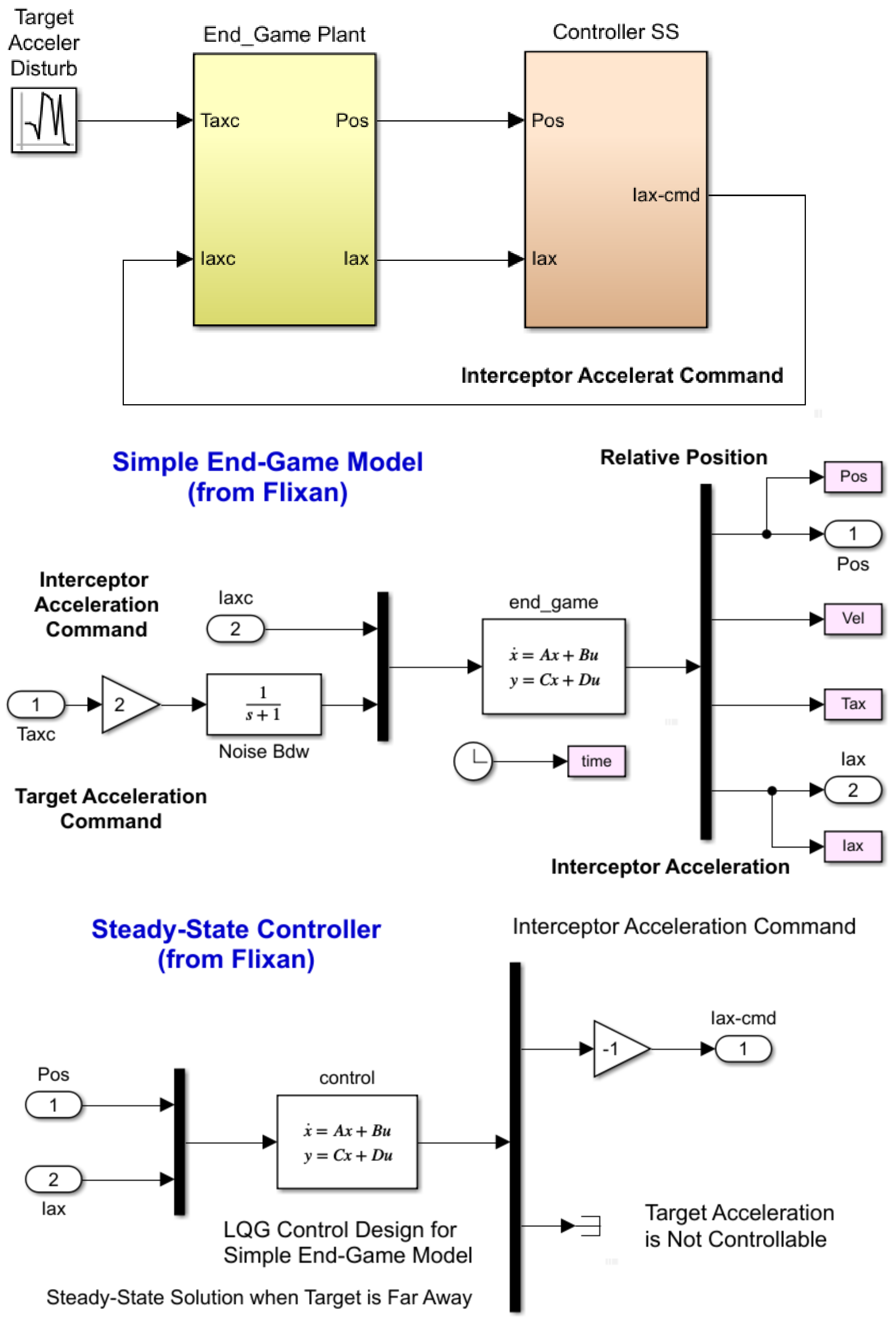

Target Acceler Disturb
End_Game Plant
Taxc
Pos
Iaxc
Iax
Controller SS
Pos
Iax-cmd
Iax
Interceptor Accelerat Command
Simple End-Game Model (from Flixan)
Relative Position
Pos
1
Pos
Interceptor Acceleration Command
Iaxc
2
1
Taxc
2
$\frac{1}{s+1}$
Noise Bdw
Target Acceleration Command
end_game
$\dot{x} = Ax + Bu$
$y = Cx + Du$
Vel
Tax
time
Iax
2
Iax
Interceptor Acceleration
Steady-State Controller (from Flixan)
Interceptor Acceleration Command
Iax-cmd
-1
1
Pos
1
2
Iax
control
$\dot{x} = Ax + Bu$
$y = Cx + Du$
LQG Control Design for Simple End-Game Model
Target Acceleration is Not Controllable
Steady-State Solution when Target is Far Away

Figure 6.2.3 Steady-State Simulation Model "EndGame_Sim1s.mdl"

The plant model is the Flixan generated system "*Simple End-Game Model*" in file "*end_game.m*". The LQG control system calculated by Flixan is: "*LQG Control Design for Simple End-Game Model*" in file "*Control.m*". It is the dynamic system shown in Figure 6.2.4, consisting of plant model parameters (A, B, C), the steady-state feedback gain K_c and the Kalman-Filter gain K_f. The two systems and matrices are loaded into Matlab by executing the m-file "init.m", which also initializes the state-vector $\underline{x}_0$. Notice, that this configuration cannot be used in the time-varying case because Kc(t) is varying.

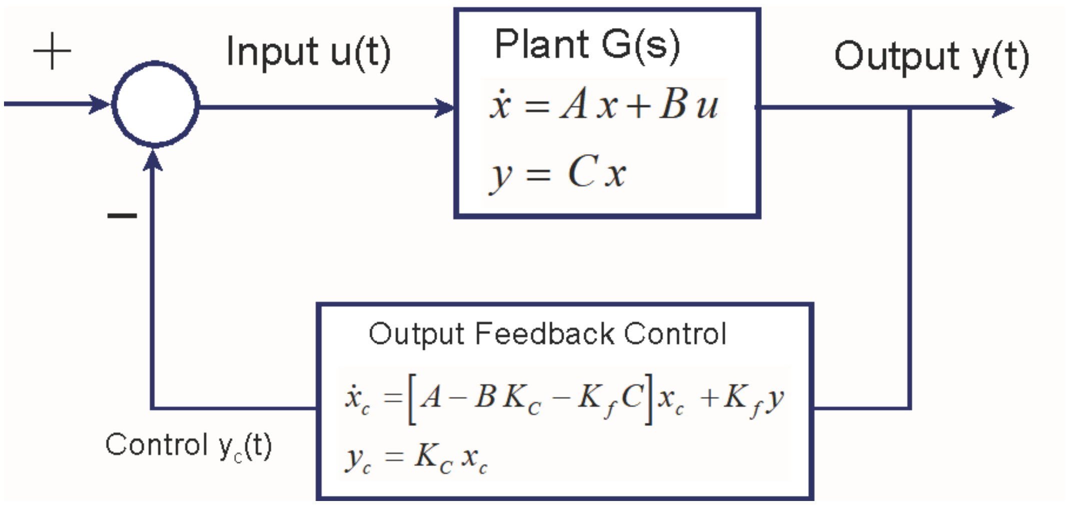


Figure 6.2.4 Steady-State LQG Output Feedback Controller/ Plant Interconnection

We can use this simulation model to calculate the system's response to an accelerating target disturbance, as shown in Figure 6.2.5.

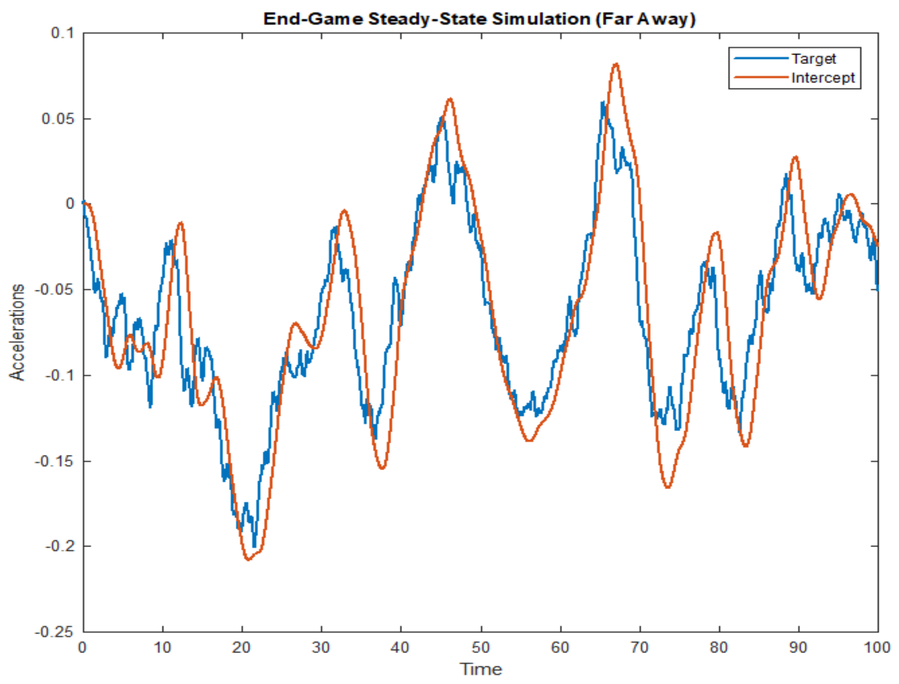


Figure 6.2.5 Interceptor Acceleration is responding to Noisy Target Accelerations

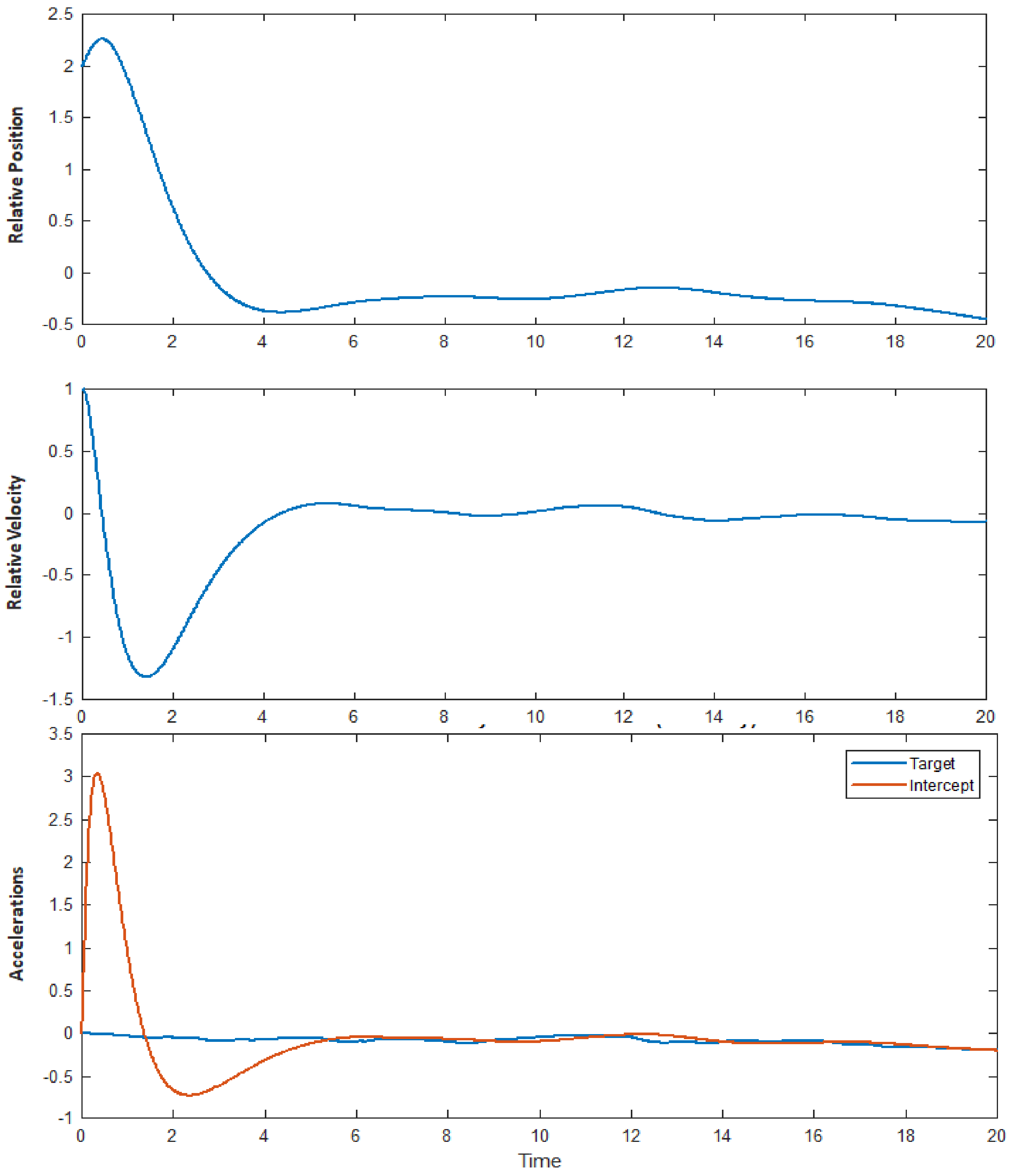
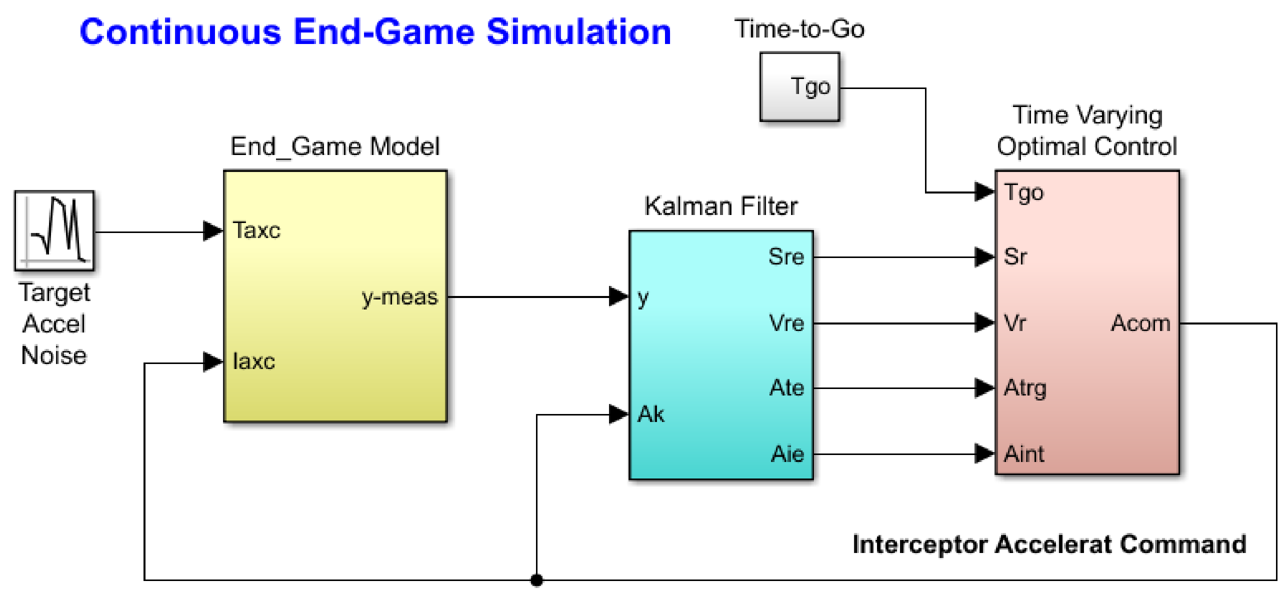


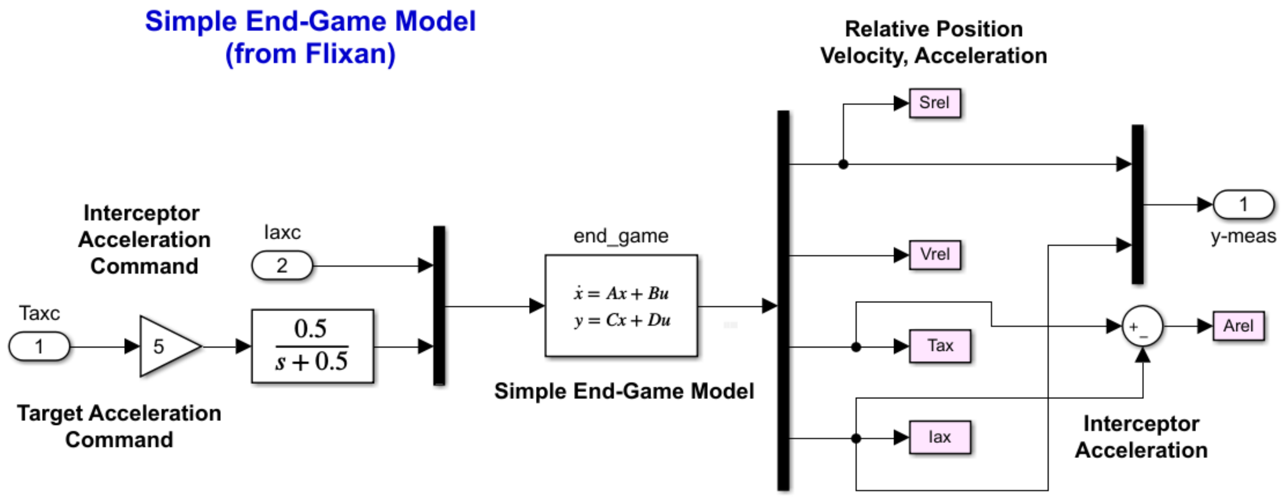
Figure 6.2.6 System's Response from Non-Zero Initial Conditions of Relative Position and Velocity

This simulation is also used to calculate the system's response to initial position and velocity errors with an accelerating target, see Figure 6.2.6. Notice that the system's response at steady-state is intentionally slow in order to save propellant since the target is still far away. This causes noticeable position error due to target acceleration.

6.2.4.3 Continuous Simulation with Time Varying Control Gains

The continuous simulation model in Figure 6.2.7 is in file "*EndGame_Sim2s.mdl*". It uses time-varying gains as a function of time-to-go which are loaded into Matlab look-up tables from file "Gains.dat". They were calculated by the Flixan Transient LQR program as already mentioned. The t_{go} is calculated from the relative axial position, velocity and acceleration and used to look-up the gains which increase as t_{go} gets shorter resulting into an exponentially increasing control bandwidth.





The Kalman-Filter and the control system are separate subsystems in this simulation. The Kalman-Filter is now used to estimate the four state variables from the relative position and the accelerometer measurements. The estimated states are multiplied with the four time varying gains to produce the acceleration command. The gains are functions of t_{go} which is calculated from the relative x-axis acceleration, velocity and range to go.

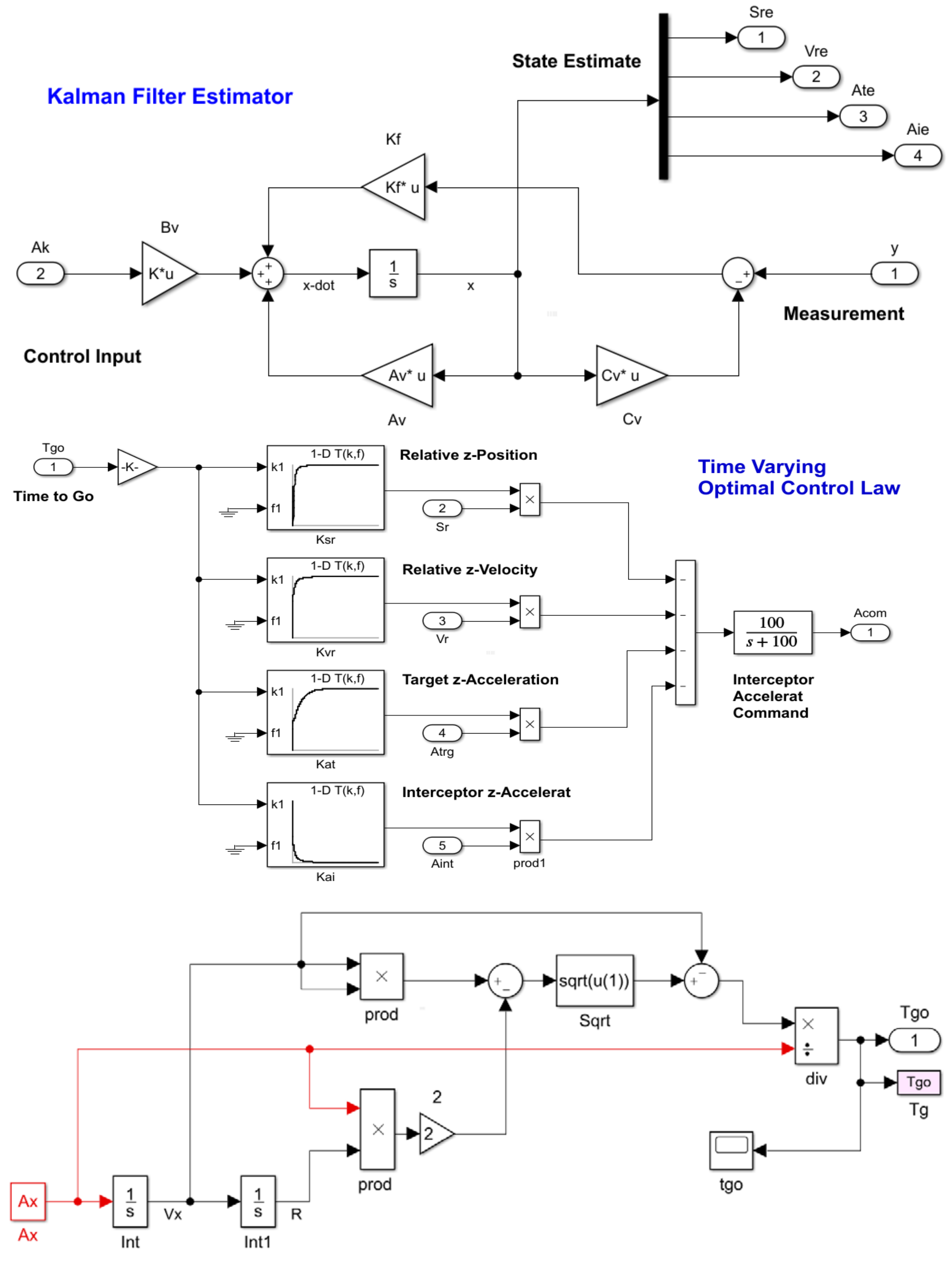


Figure 6.2.7 Simulation Model "EndGame_Sim2s.mdl" that uses Time-Varying Gains

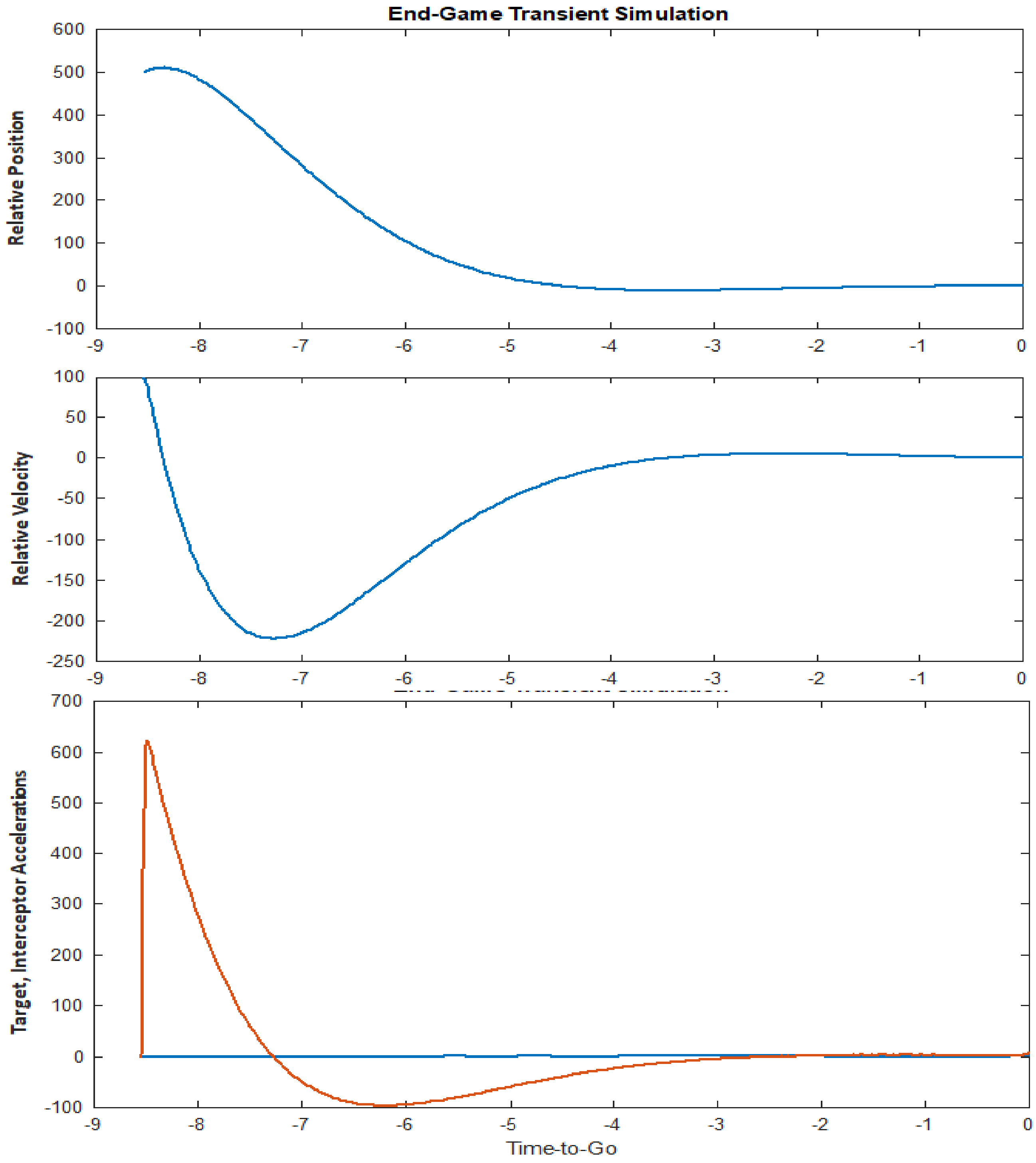


Figure 6.2.8 System's Response from Non-Zero Initial Conditions of Relative Position and Velocity

Figure 6.2.8 shows the response of the time-varying control system to non-zero initial conditions versus time-to-go, beginning 8.5 sec before impact when the gains are still at steady state. Beginning with initial position and velocity errors 500 (feet) and 100 (ft/sec) respectively, the interceptor (orange) accelerates in order to bring the final position and velocity errors very close to zero at impact.

6.2.5. Discrete-Time Analysis

The discrete-time analysis is similar. The files are located in subfolder "*LQG\Examples\End-Game\Discrete*". It includes the input file "*End_Game_z.Inp*", shown below, that contains input data for the design programs. That is, for the discrete steady-state LQR control, the discrete time-varying state-feedback LQR, the discrete Kalman-Filter, and for the discrete steady-state dynamic output-feedback LQG design. The input file also includes a batch set for fast data processing and datasets for Matlab conversions. The systems filename "*End_Game_z.Qdr*" contains the End-Game dynamic model "*Simple End-Game Model, Z-Transform*", which is a z-transformation of the continuous model "*Simple End-Game Model*" described in Equation 6.2.2, discretized at 40 (msec) sampling period. The systems file also contains the control design matrices: Q_c, R_c, and P_1, the Kalman-Filter design matrices Q_{mn}, R_{mn}, the control gain K_c and the Kalman-Filter gain K_f which are generated by the Flixan LQR design program.

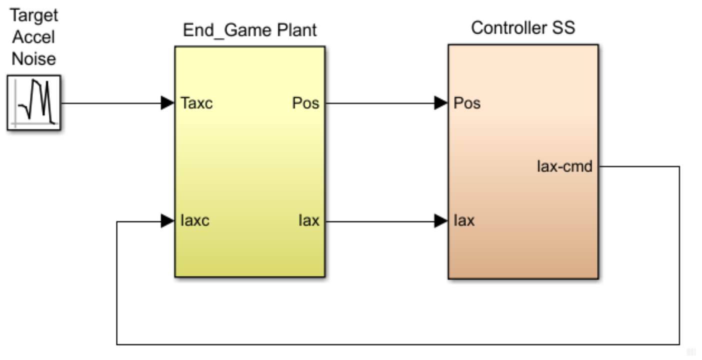
```
BATCH MODE INSTRUCTIONS ...............
Batch for preparing End-Game Control design Models
! This batch Generates LQR State-Feedback, Kalman-Filter and Output
! Feedback Dynamic Controller
Retain System    : Simple End-Game Model
Retain Matrix    : State Weight Matrix Qc (4x4)
Retain Matrix    : Control Weight Matrix Rc
Retain Matrix    : Terminal State Weight Matrix P1 (4x4)
Retain Matrix    : Measurement Noise Matrix Rmn (2x2)
Retain Matrix    : Process Noise Matrix Qpn (4x4)
Retain Matrix    : Input Noise Matrix G
Retain Matrix    : Process Noise Matrix Qpn1
!
S-Z-Transform    : Simple End-Game Model, Z-Transform
LQR Control Des  : LQR Control Design for Discrete End-Game Model
State Estimator  : Kalman-Filter Design 2 for Discrete End-Game Model
LQG Control Des  : LQG Control Design for Discrete End-Game Model
Transient LQR    : Transient LQR Design for Discrete End-Game Model
!
To Matlab Format : Simple End-Game Model, Z-Transform
To Matlab Format : LQG Control Design for Discrete End-Game Model
To Matlab Format : LQR State-Feedback Control for Discrete End-Game Model
To Matlab Format : Kalman-Filter Estimator 2 for Discrete End-Game Model
-----------------------------------------------------------------------------------------------------
TRANSFORM A SYSTEM (S-Z-W) ..... (Z system title, Comments, S-System title, Transform)
Simple End-Game Model, Z-Transform
! Discretize the Continuous End-Game Model at dT=0.04 sec Using the
! S to Z Transformation
!
Simple End-Game Model
From S-plane to Z-plane using the Z-Transform, dT= 0.04
--------------------------------------------------------------------------------------------------
LINEAR QUADRATIC REGULATOR STATE-FEEDBACK CONTROL DESIGN
LQR Control Design for Discrete End-Game Model
! Design the Discrete Steady-State-Feedback Matrix Kc using the
! Output Criteria Matrix C= Identity
!
Plant Model Used to Design the Control System from:        Simple End-Game Model, Z-Transform
Criteria Optimization Output is Matrix Identity
State Penalty Weight (Qc) is Matrix:   Qc4                 State Weight Matrix Qc (4x4)
Control Penalty Weight (Rc) is Matrix: Rc                  Control Weight Matrix Rc
Discrete LQR Solution Using Assyptotic Method
LQR State-Feedback Control Gain Matrix Kc                  LQR State-Feedback Control for Discrete End-
-----------------------------------------------------------------------------------------------------
```

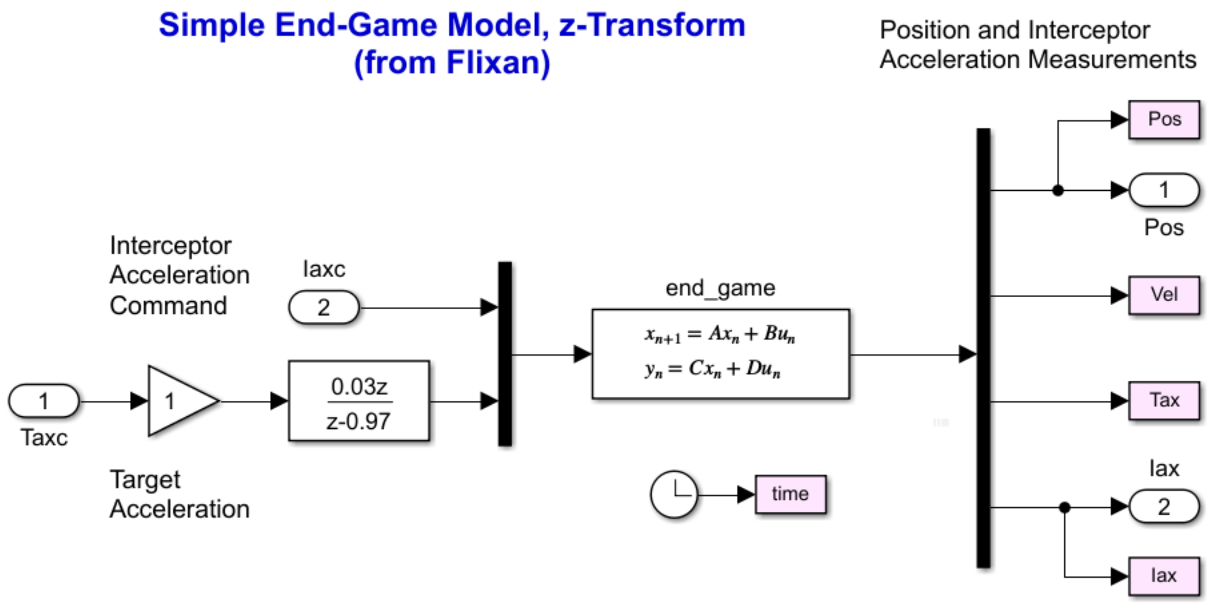
```
KALMAN-BUCY FILTER STATE ESTIMATOR DESIGN
Kalman-Filter Design for Discrete End-Game Model
! Design the Discrete Kalman-Filter Gain Matrix Kf using the
! Process Noise Matrix G
!
Plant Model Used to Design the Kalman-Filter from:         Simple End-Game Model, Z-Transform
Input Process Noise Matrix is Matrix   G                   Input Noise Matrix G
Process Noise Covariance Qpn is Matrix Qpn                 Process Noise Matrix Qpn1
Measurement Noise Covariance is Matrix Rmn                 Measurement Noise Matrix Rmn (2x2)
Kalman-Filter Estimator is Gain Matrix Kf                  Kalman-Filter Estimator for Discrete End-
-------------------------------------------------------------------------------------------------------
KALMAN-BUCY FILTER STATE ESTIMATOR DESIGN
Kalman-Filter Design 2 for Discrete End-Game Model
Plant Model Used to Design the Kalman-Filter from:         Simple End-Game Model, Z-Transform
Input Process Noise Matrix (G) is the  Identity
Process Noise Covariance Qpn is Matrix Qpn                 Process Noise Matrix Qpn (4x4)
Measurement Noise Covariance is Matrix Rmn                 Measurement Noise Matrix Rmn (2x2)
Kalman-Filter Estimator is Gain Matrix Kf                  Kalman-Filter Estimator 2 for Discrete End-
-------------------------------------------------------------------------------------------------------
DYNAMIC OUTPUT FEEDBACK LQG CONTROL DESIGN
LQG Control Design for Discrete End-Game Model
! Combine State-Feedback with KF Gain to Design a Linear Quadratic
! Gaussian Control System for the Plant: Simple End-Game Model, Z-Transform
!
Plant Model Used to Design the Control System from:        Simple End-Game Model, Z-Transform
State-Feedback (Kc) is Gain Matrix   : Kc                  LQR State-Feedback Control for Discrete End-
Game Model
Kalman-Filter Estim Kf is Gain Matrix: Kf                  Kalman-Filter Estimator 2 for Discrete End-
Game Model
-------------------------------------------------------------------------------------------------------
TRANSIENT LQR CONTROL DESIGN WITH TIME-VARYING GAINS
Transient LQR Design for Discrete End-Game Model
! To Generate Time-Varying State-Feedback Gain Kc(t)
! as a Function of Time-to-Go, in File: Gains.dat
!
Plant Model Used to Design the Control System from:        Simple End-Game Model, Z-Transform
Criteria Optimization Output is Matrix Identity
State Penalty Weight (Qc) is Matrix:   Qc4                 State Weight Matrix Qc (4x4)
Control Penalty Weight (Rc) is Matrix: Rc                  Control Weight Matrix Rc
Terminal State Penalty Weigh P1 Matrix P1                  Terminal State Weight Matrix P1 (4x4)
Discrete LQR Solution, Final Time (Tf) in (sec)             20.0
-------------------------------------------------------------------------------------------------------
CONVERT TO MATLAB FORMAT ........      (Title, System/Matrix, m-filename)
Simple End-Game Model, Z-Transform
System
end_game
-------------------------------------------------------------------------------------------------------
CONVERT TO MATLAB FORMAT ........      (Title, System/Matrix, m-filename)
LQG Control Design for Discrete End-Game Model
System
Control
-------------------------------------------------------------------------------------------------------
CONVERT TO MATLAB FORMAT ........      (Title, System/Matrix, m-filename)
Kalman-Filter Estimator 2 for Discrete End-Game Model
Matrix Kf
-------------------------------------------------------------------------------------------------------
CONVERT TO MATLAB FORMAT ........      (Title, System/Matrix, m-filename)
LQR State-Feedback Control for Discrete End-Game Model
Matrix Kc
-------------------------------------------------------------------------------------------------------
```

The discrete LQR transient design program also calculates the time varying state-feedback gain $K_c(t)$ as a function of time-to-go. It requires the weight matrices: Q_c, R_c, and P_1, and the initial time-to-go (20 sec). The gains versus time are saved in file “Gains.dat” which is used as a look-up table in the simulation. The discrete end-game dynamic model is saved in file “end_game.m”, and the discrete steady-state output-feedback controller is saved in “control.m” for Matlab analysis. The gain matrices K_c and K_f are also saved and loaded into Matlab. Two discrete Simulink models are included for analysis: a steady-state model, and a time-varying model, both running at 40 msec sampling.

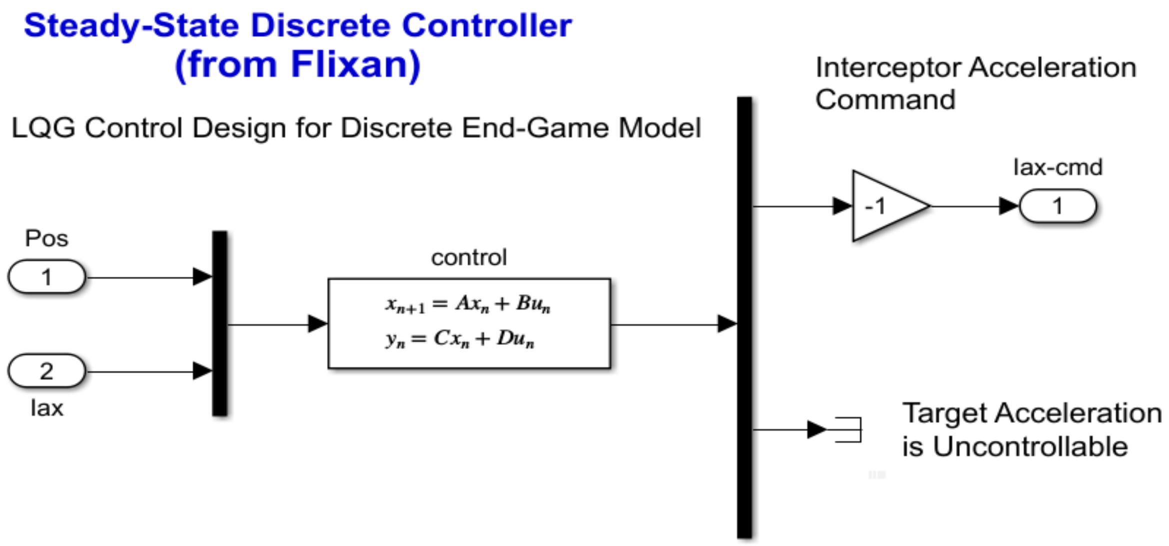
6.2.5.1 Discrete Steady-State Simulation

The steady-state model is used to analyze the system when the target is sufficiently far away from the interceptor and the control gains are constant. The Simulink model is "*EndGame-Sim1z.mdl*", shown in Figure 6.2.9, and it is located in "*LQG\ Examples\End-Game\ Discrete*" subfolder.





Simple End-Game Model, z-Transform (from Flixan)
Position and Interceptor Acceleration Measurements
Interceptor Acceleration Command
Iaxc
2
1
Taxc
1
0.03z / z-0.97
Target Acceleration
end_game
$x_{n+1} = Ax_n + Bu_n$
$y_n = Cx_n + Du_n$
time
Pos
1
Pos
Vel
Tax
Iax
2
Iax



Steady-State Discrete Controller (from Flixan)
LQG Control Design for Discrete End-Game Model
Pos
1
2
Iax
control
$x_{n+1} = Ax_n + Bu_n$
$y_n = Cx_n + Du_n$
Interceptor Acceleration Command
-1
Iax-cmd
1
Target Acceleration is Uncontrollable

Figure 6.2.9 Discrete Steady-State Simulation Model "EndGame_Sim1z.mdl"

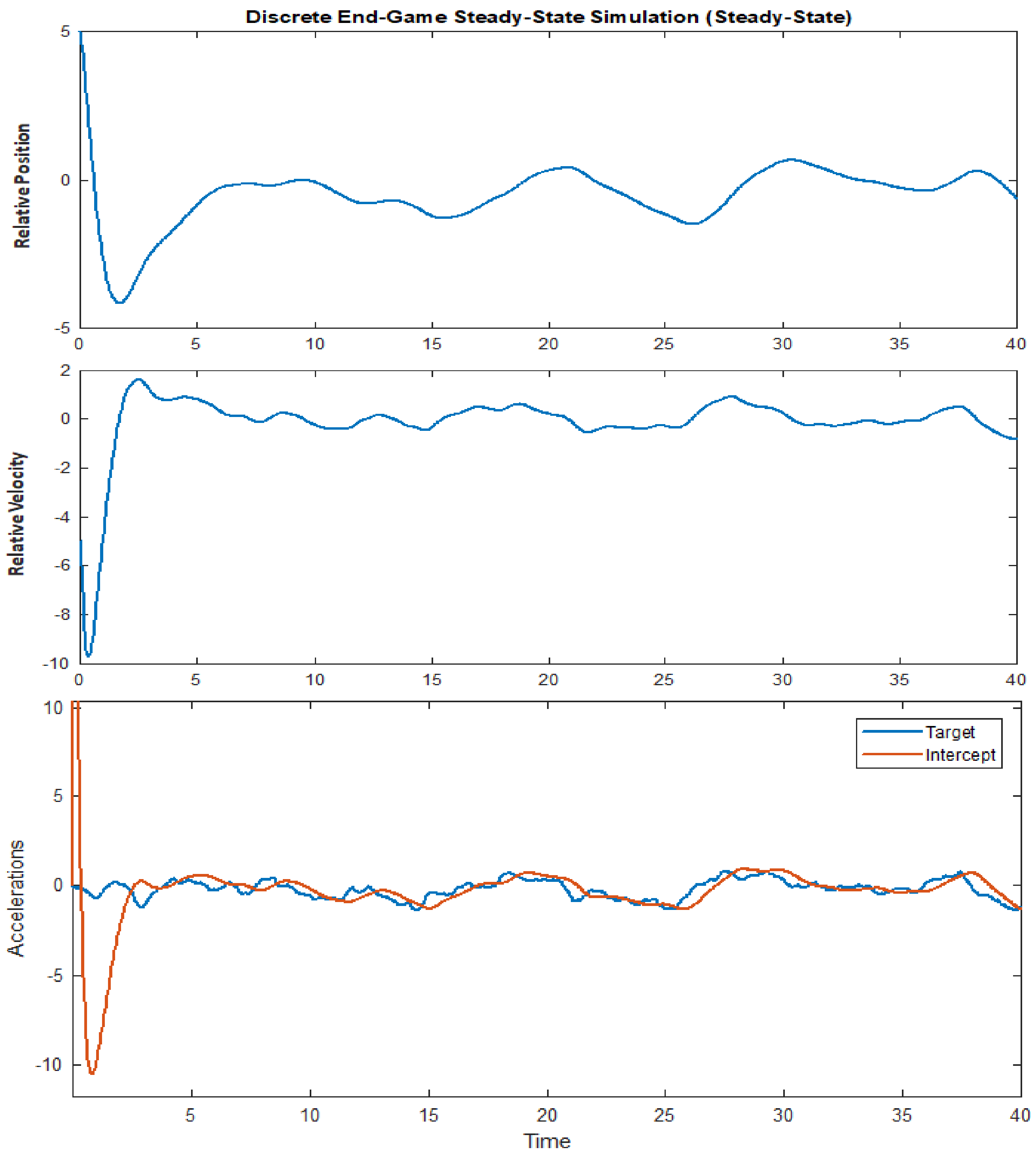
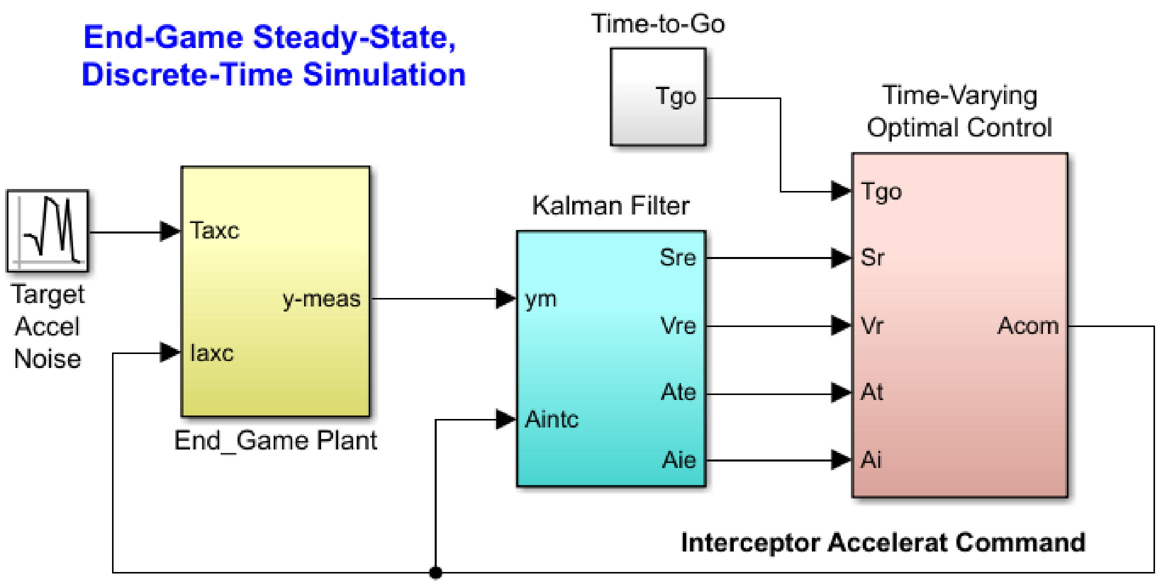


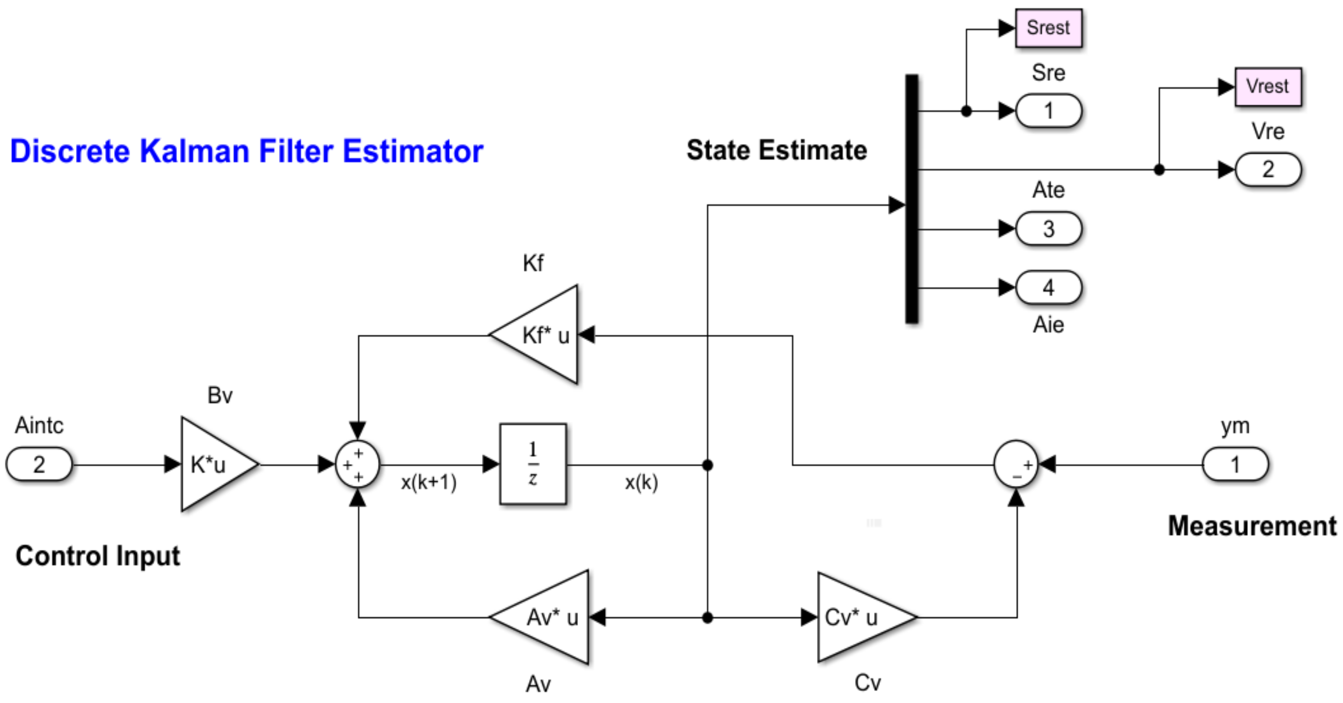
Figure 6.2.10 Discrete System's Response from Non-Zero Initial Conditions of Relative Position and Velocity

Figure 6.2.10 shows discrete system's response to a randomly accelerating target. The dynamic model is initialized at some arbitrary non-zero relative position and velocity errors. The interceptor acceleration responds to the target's average acceleration and the relative position and velocity are considerably reduced. Notice that this is steady-state condition where the interceptor's response is slow. The engagement becomes a lot more dynamic when t_{go} approaches zero.

6.2.5.2 Discrete Simulation with Time Varying Gains

The discrete simulation model in Figure 6.2.11 is in file "*EndGame_Sim2z.mdl*". It uses time-varying gains as a function of time-to-go loaded from file "Gains.dat", and it is very similar to the continuous model "*EndGame_Sim2s.mdl*". The gains are calculated from the discrete Transient LQR program as already described. The gains are functions of t_{go} which is calculated from the relative axial position, velocity and acceleration and they increase as the vehicle approaches the target and t_{go} becomes shorter. The discrete Kalman-Filter estimates the four state variables from the relative position and the accelerometer measurements. The response of the discrete system is similar to the continuous model's response. Both, relative position and relative velocity are reduced to almost zero at impact, as shown in Figure 6.2.12.





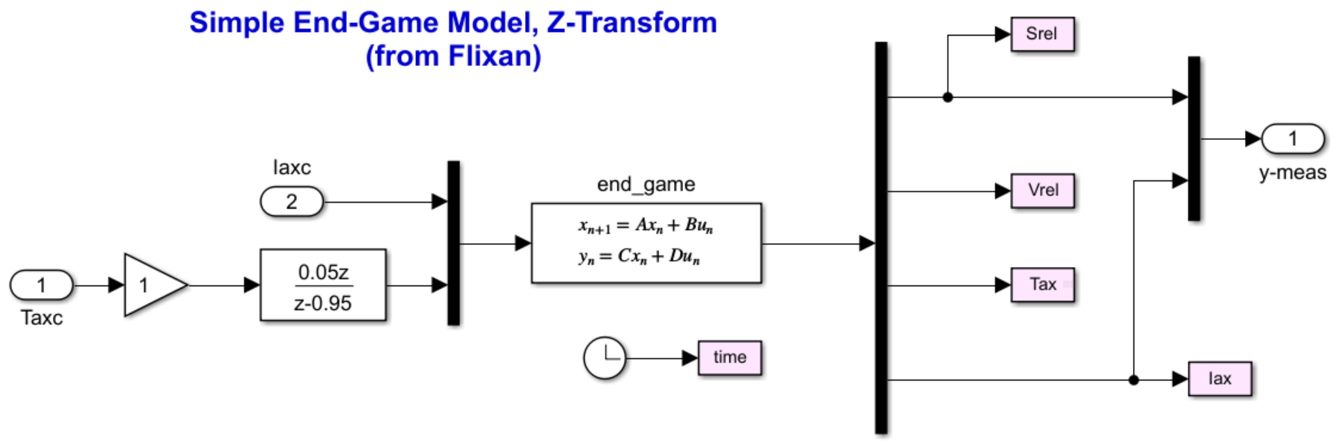


Figure 6.2.11 Discrete Simulation Model "EndGame_Sim2z.mdl" that uses Time-Varying Gains

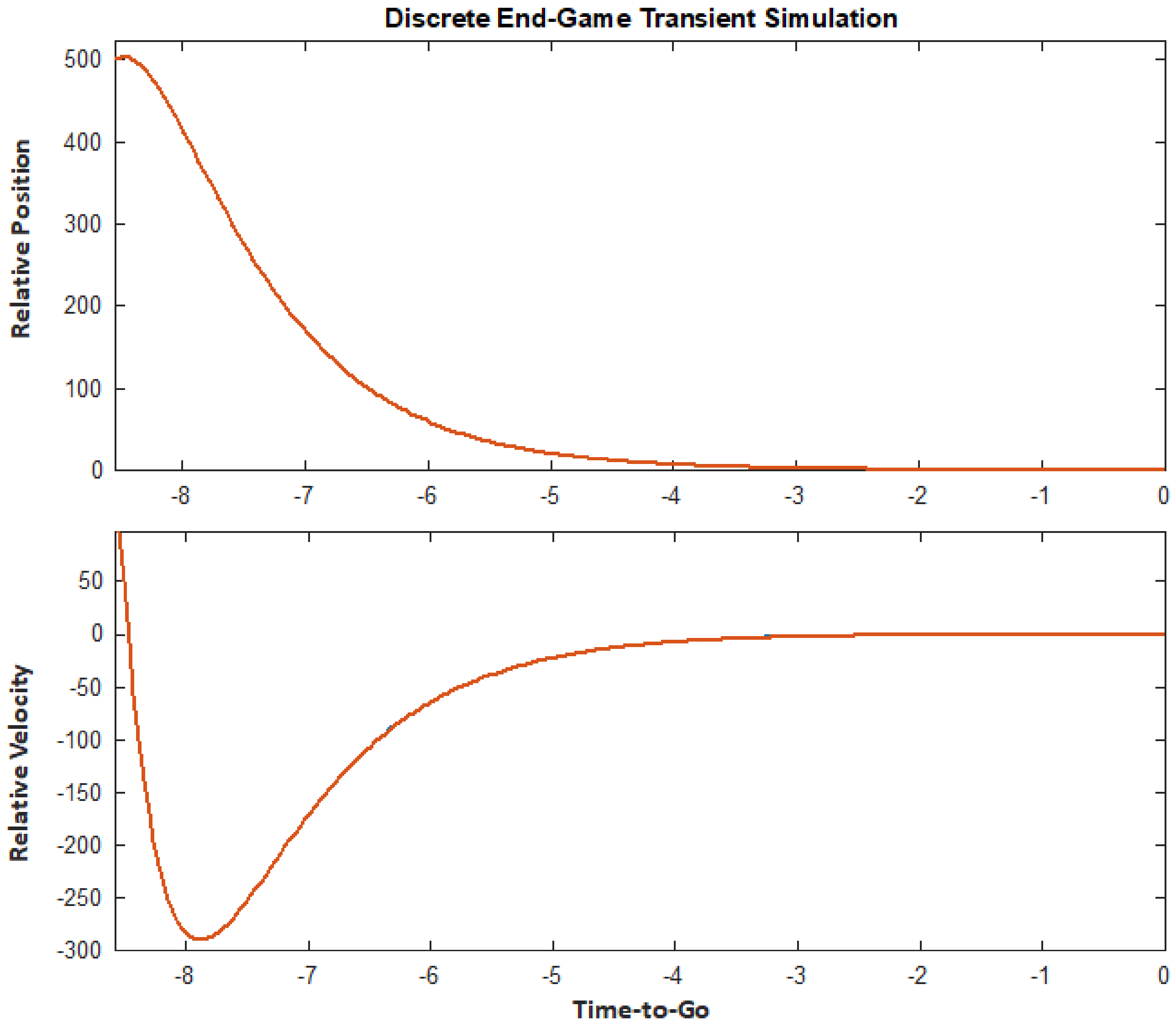
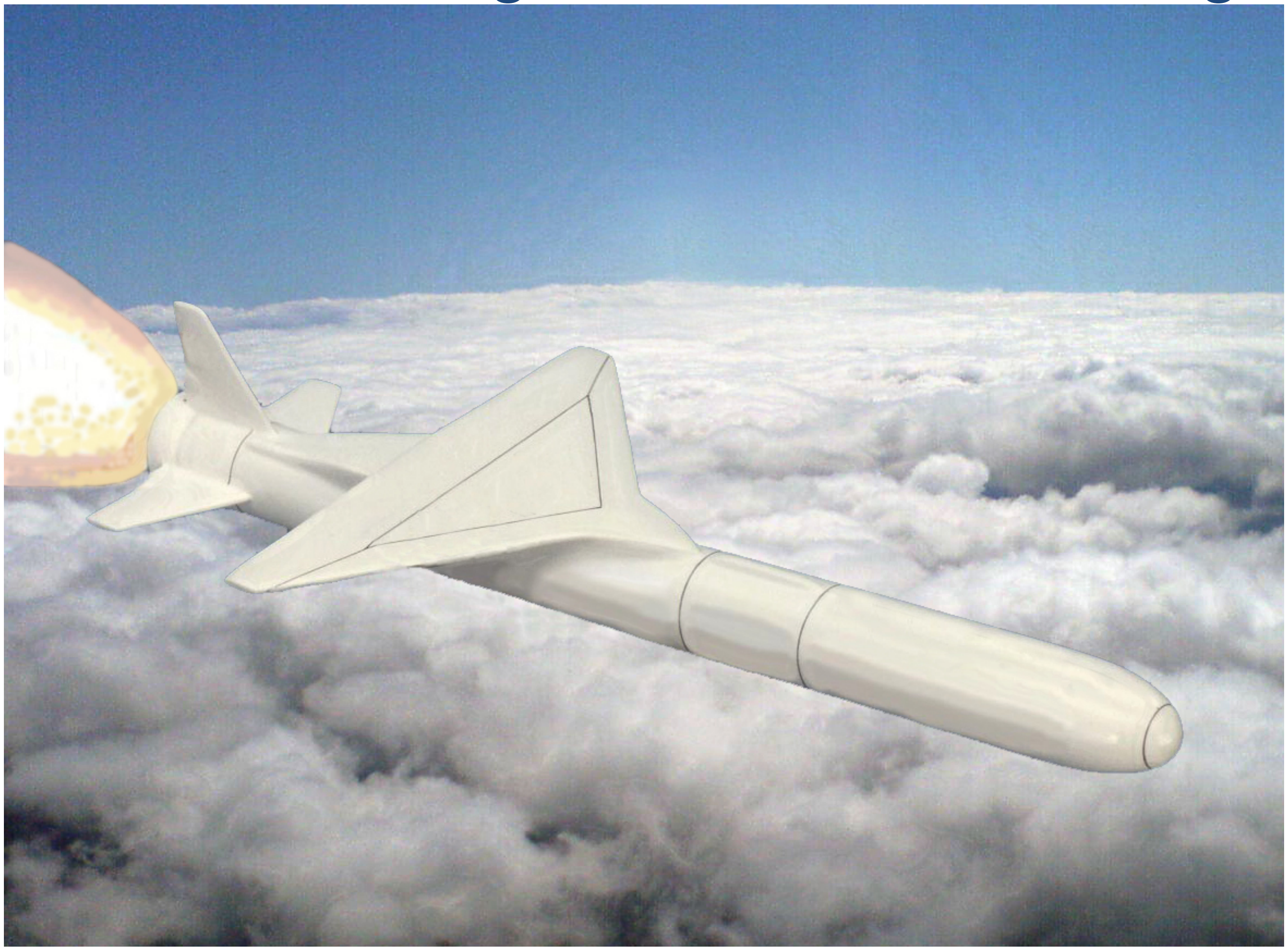


Figure 6.2.12 Discrete System's Response from Non-Zero Initial Conditions of Relative Position and Velocity

6.3 Control Design of a Missile with Wing



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 6.3.1. 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 LQR control laws for the pitch and lateral dynamics separately, and analyze stability and performance using Matlab.

6.3.1 Flight Control System Description

Figure 6.3.1 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 it is released.

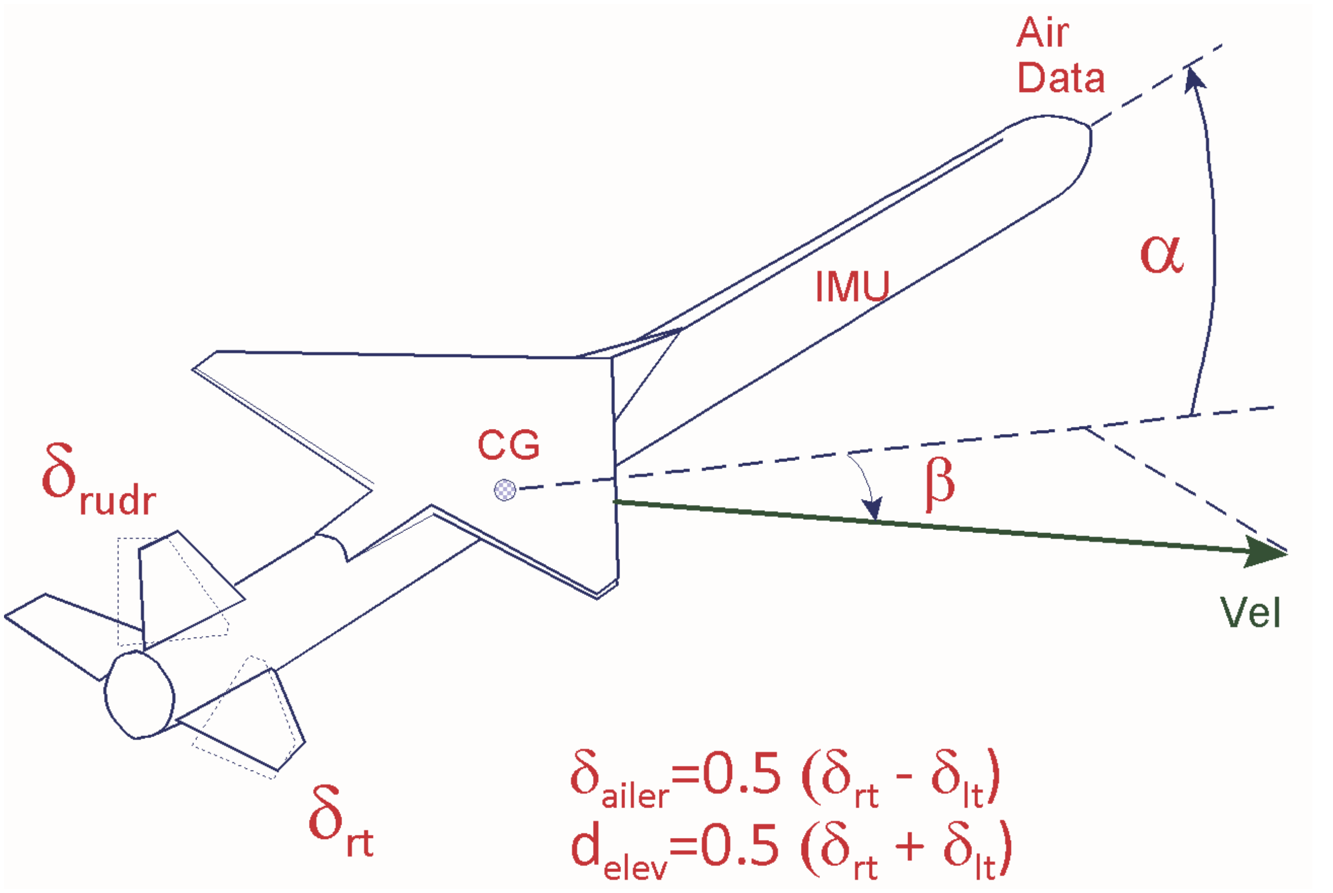

Air Data
α
IMU
CG
δ_{rudr}
β
Vel
δ_{rt}
$\delta_{ailer}=0.5\ (\delta_{rt} - \delta_{lt})$
$d_{elev}=0.5\ (\delta_{rt} + \delta_{lt})$

Figure 6.3.1 Missile Configuration showing the Aerosurfaces, Wing, CG, and Sensor Locations.

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 gamma-tracking, and in the lateral heading direction tracking. Since the vehicle is perfectly symmetric the analysis will be separated in pitch and lateral control design and 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 “Flixan\Control Analysis\LQG\Examples\Missile Control Design\Pitch LQR”. The lateral vehicle model is in the input file “Later_LQR_Des.Inp” which is located in directory “Flixan\Control Analysis\LQG\Examples\ Missile Control 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.

The design models will be augmented by including the aerosurface actuators and integrals of some of the states. The augmented design models improve speed of response and the tracking performance of the control system. We will also create pitch and lateral models for control analysis and simulations. The Flixan program will be used to perform dynamic modeling and control design and Matlab for the simulations. The dynamic models and control gains are converted from the system files and loaded into Matlab for analysis.

Note that in this example the incidence angles α and β, which are used to synthesize the flight-path and heading directions, do 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. This is introduced in the flight vehicle input data by a flag label "**NoWind Alpha**" in the flags line, to indicate the type of (α, β) measurement. 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 output as it would be if it was an air-data probe, only its effect on the vehicle will be observable.

6.3.2.1 Longitudinal Control Design and Analysis

The input file for the longitudinal axis design is “*Pitch_LQR_Des.Inp*” located in subdirectory “*Control Analysis\LQG\Examples\Missile Control 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 from getting erased in systems file “*Pitch_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 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 other 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 Bending Modes                                                                  :  0
----------------------------------------------------------------------------------------------------------------------

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  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 systems.

In the longitudinal direction we would like the estimated flight path angle (γ) to follow a pre-calculated flight path (γ_{comd}). The original design plant, however, is not equipped to track γ and to produce an efficient control design. We must create, therefore, and regulate a "γ–integral" state and include it in the design model. It is also good idea to include a simple actuator model in the plant because it introduces more plant information in the design which makes the control system more efficient with less phase-lag. The two additional variables γ and δ_{elevon} can easily be included in the LQR optimization because they are both measurable for feedback.

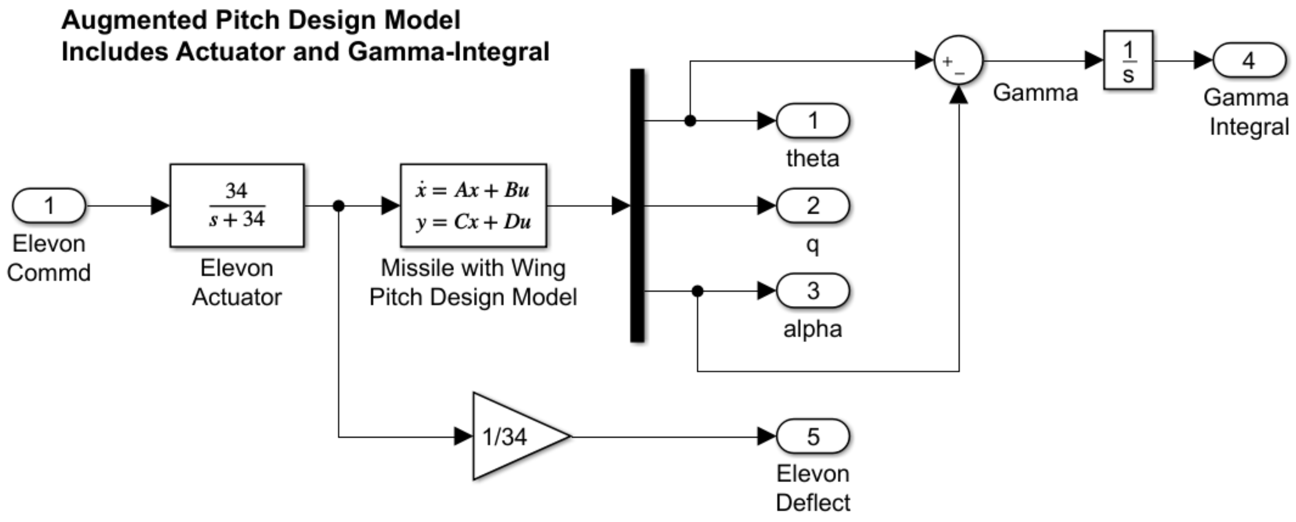


Figure 6.3.2 Augmented Longitudinal Design Plant for LQR Control Design

Figure 6.3.2 shows the augmented plant for the longitudinal LQR control design. The following interconnection dataset combines the 3 subsystems and generates the augmented system, which is: "*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
-----------------------------------------------------------------------------------------------

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-
```

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.

6.3.2.2 Longitudinal Simulation

The simulation model "*Pitch_Sim.mdl*" in Figure 6.3.3, is in folder "*Flixan\Control Analysis\LQG\Examples\Missile Control Design\Pitch LQR*". It has the 5-state-feedback loop closed via matrix K_c which includes feedback from γ-integral and δ_{elevon} in addition to the feedback from the original vehicle states (θ, q, α). In the simulation Figure 6.3.4 the vehicle is commanded to perform 1^o increase in γ, and in Figure 6.3.5 it is excited by an upward wind-gust velocity impulse.

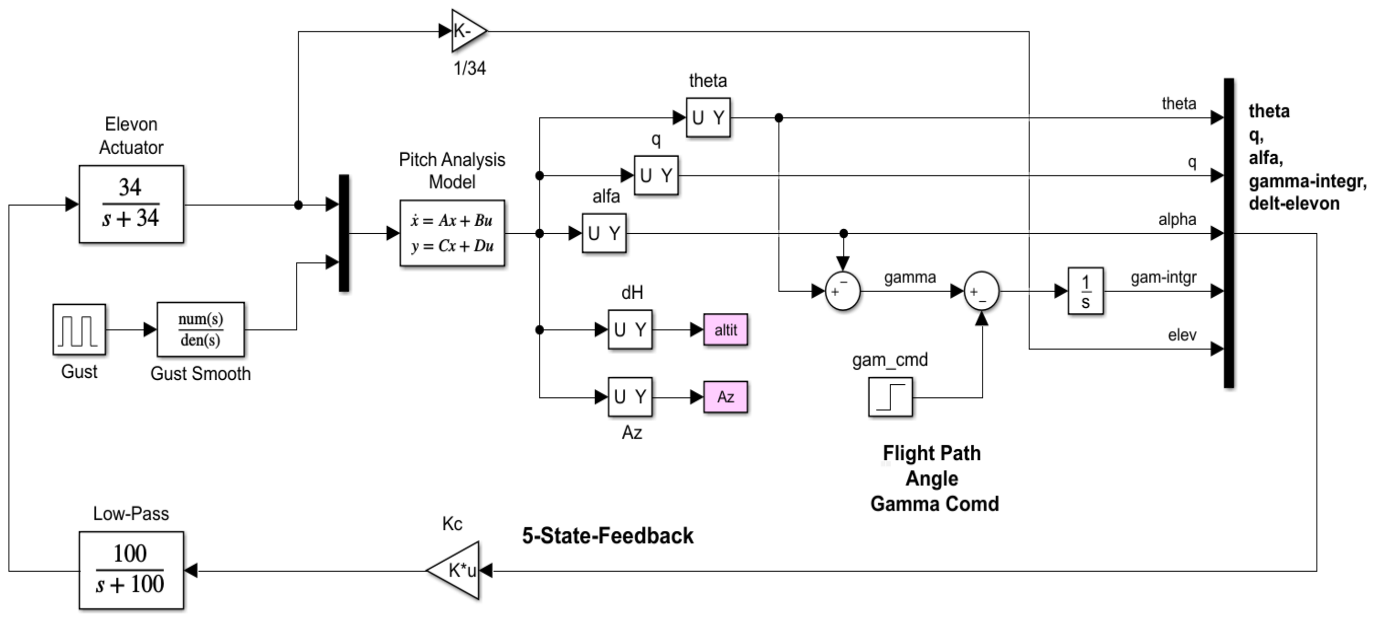


Figure 6.3.3 Longitudinal Axes Closed-Loop Simulation Model “Pitch_Sim.mdl”

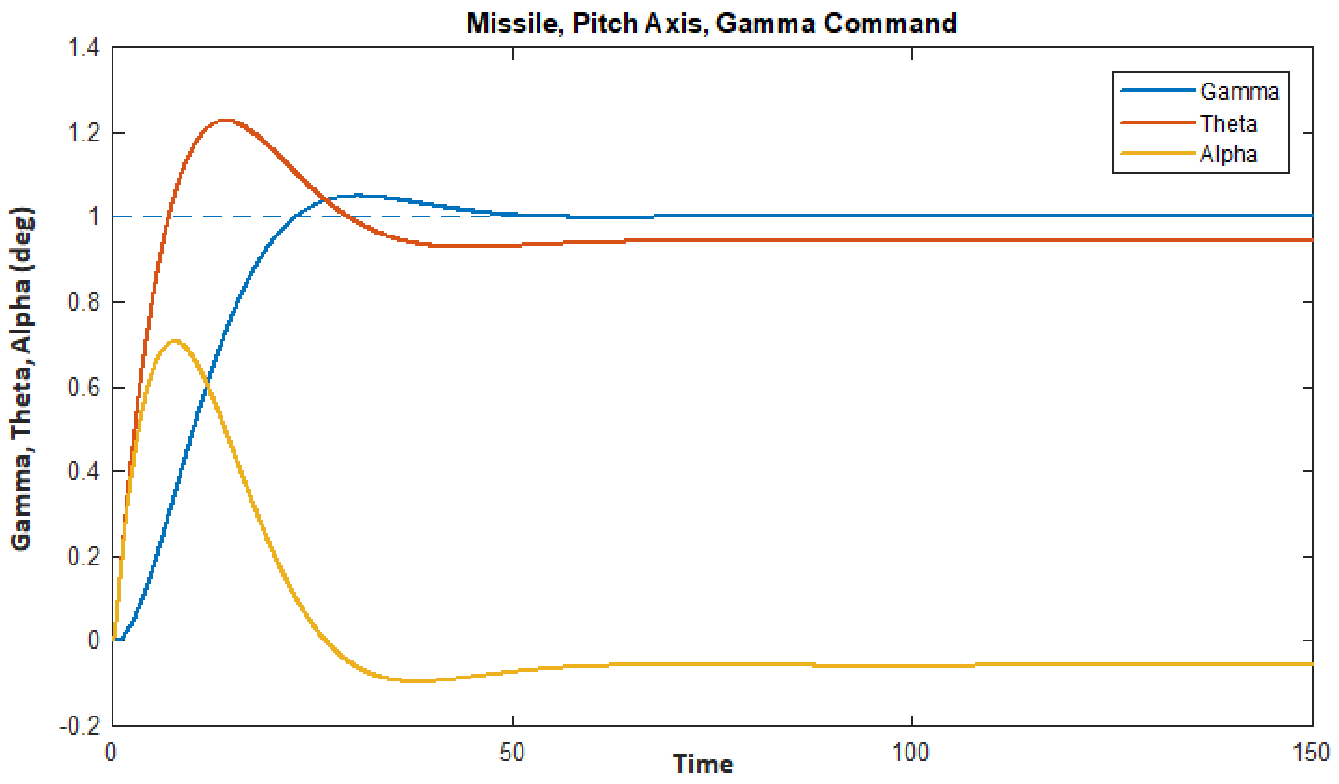


Figure 6.3.4 Flight Path Angle Responds to 1 degree Gamma Command

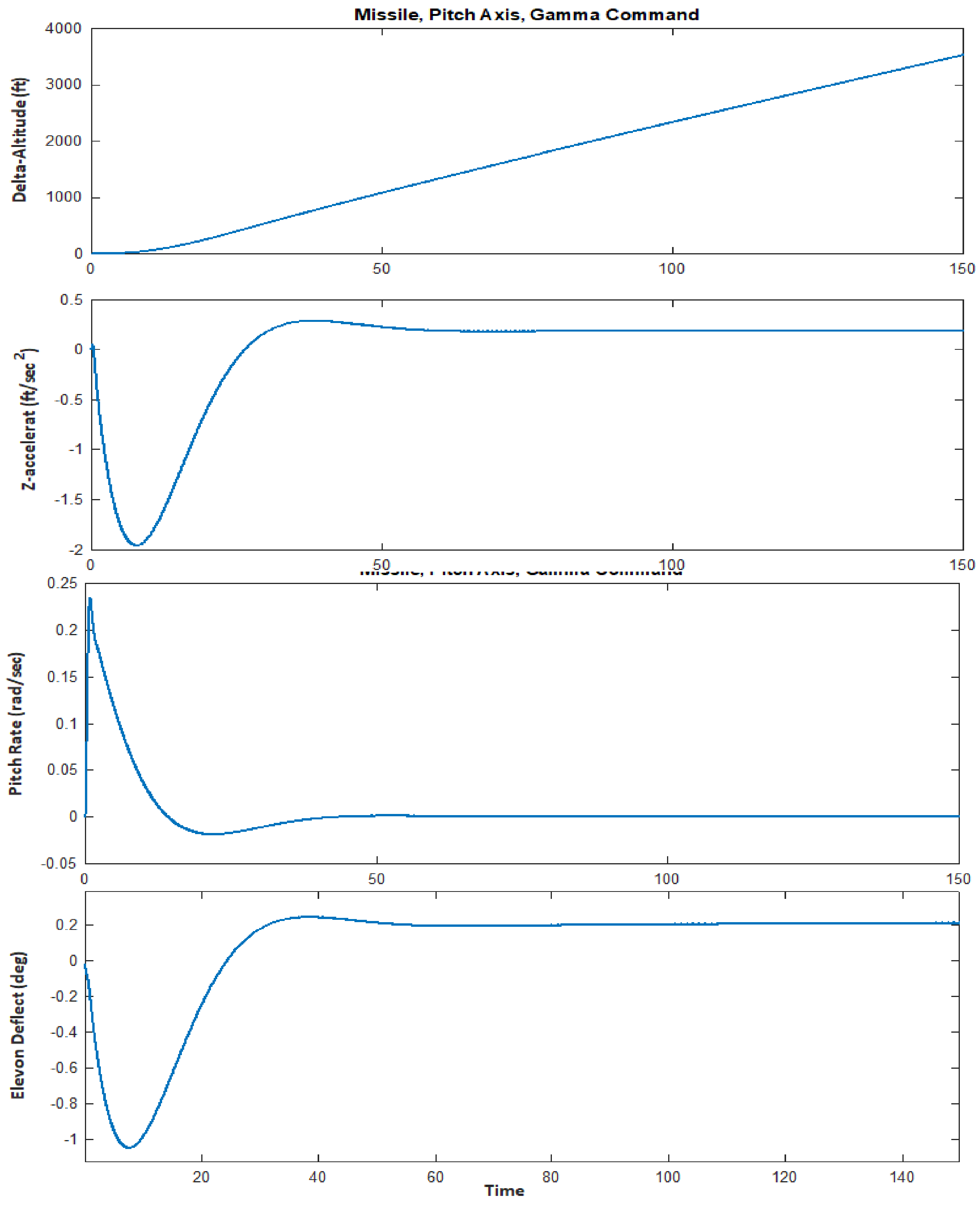


Figure 6.3.4b Missile Responds to the Gamma Command. Negative Elevon produces a Positive Pitch Rate, Negative (upwards) Acceleration and the Missile is Steadily Climbing at Higher Altitudes

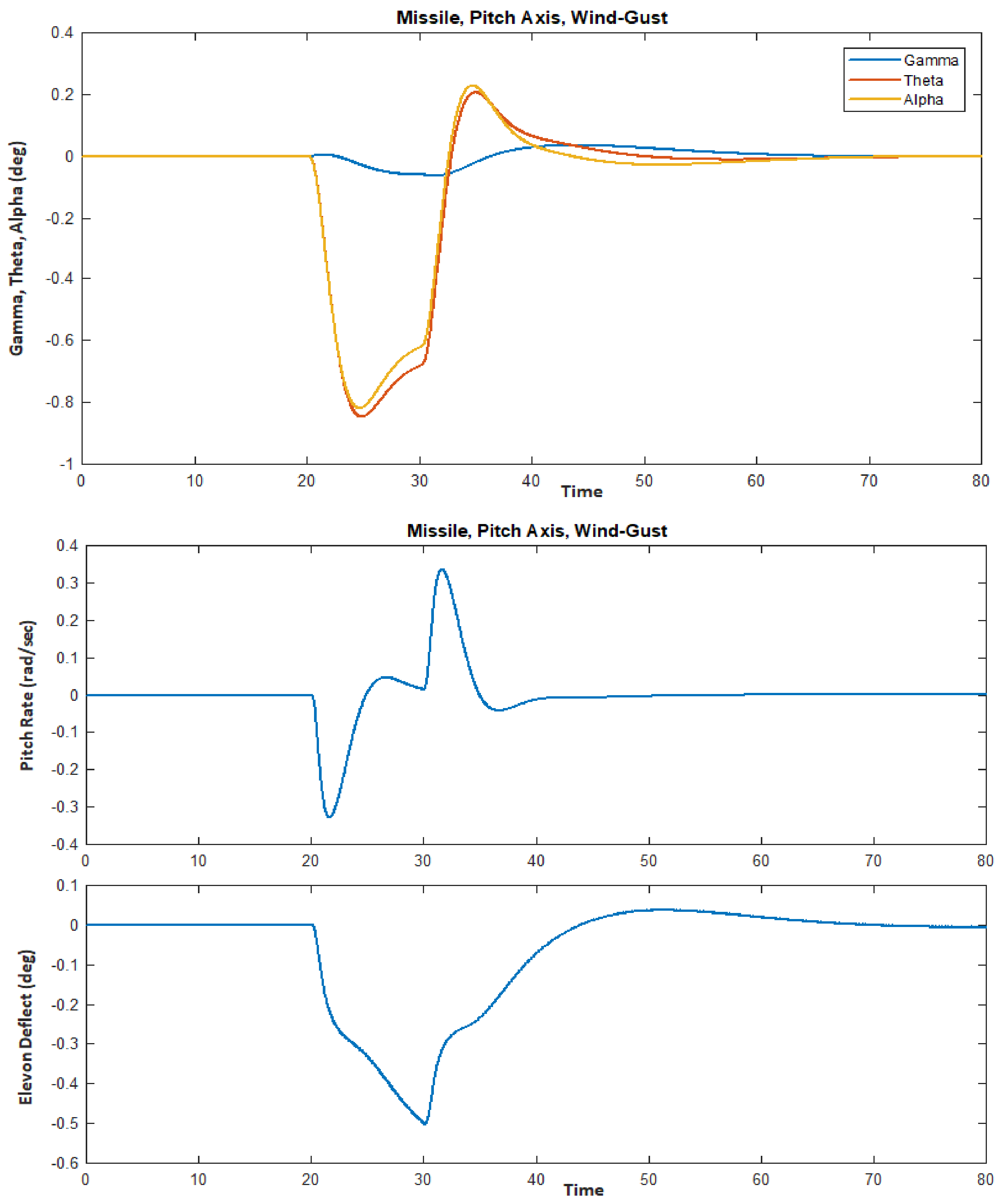


Figure 6.3.5 The Missile is excited by a Wind-Gust from below that causes Negative pitch and rate and Z-acceleration. The Elevon Responds with Negative Deflection to Counteract the Negative Pitch Rate

6.3.2.3 Stability Analysis

The system stability is analyzed in the frequency domain by calculating the Nichols plot using the open-loop Simulink model "*Open_Pitch.mdl*" shown in Figure 6.3.6. The script file "frequ.m" calculates the frequency response across the opened loop. The model includes the first order actuator and a low-pass filter. The loop is broken between the low-pass filter output and the actuator input. The Nichols plot in Figure 6.3.7 shows the control system's stability margin.

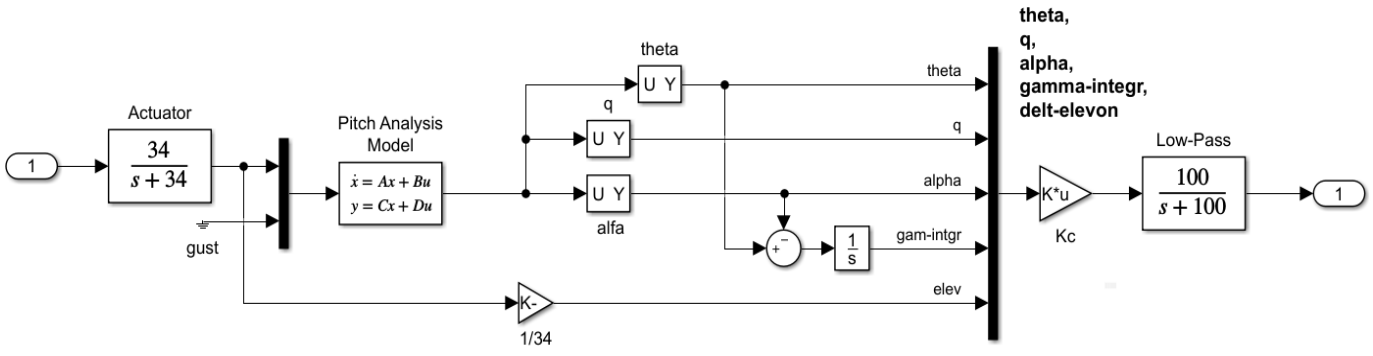


Figure 6.3.6 Open-Loop Model "Open_Pitch.mdl" for Pitch Stability Analysis

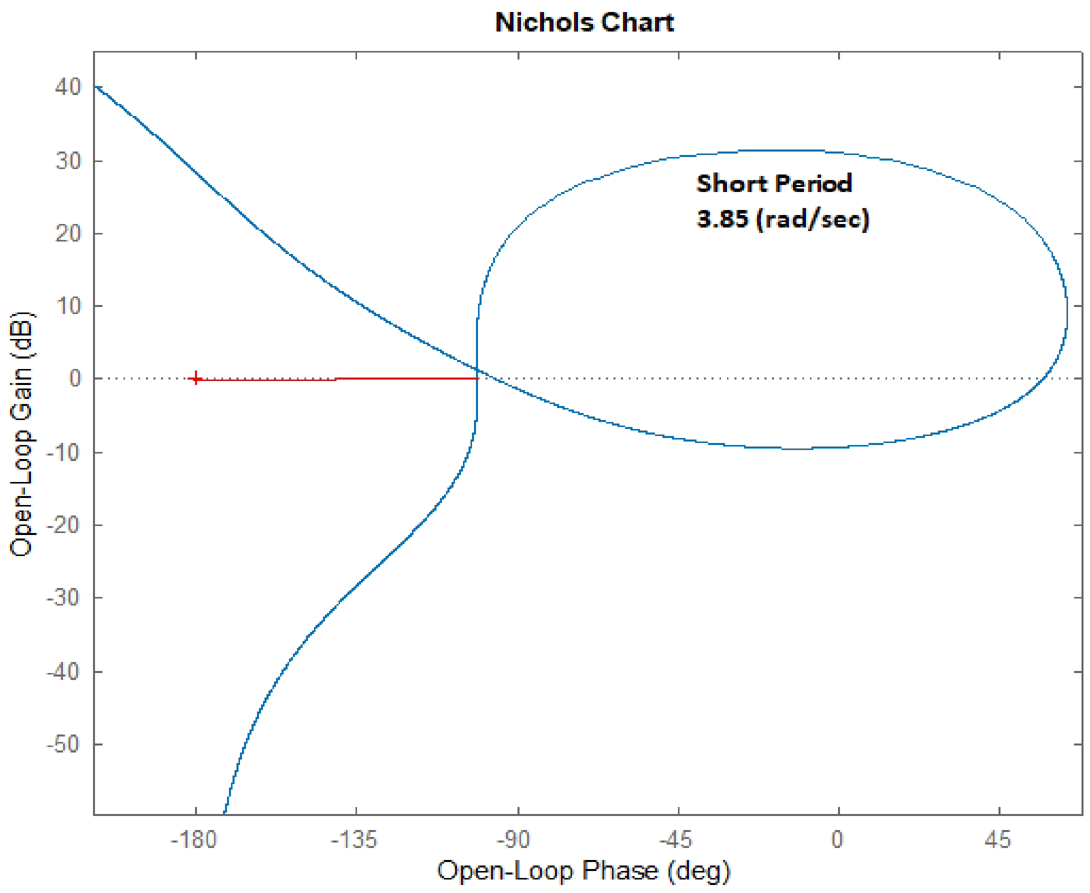


Figure 6.3.7 The Nichols Plot Shows that the LQR Control System has Plenty of Stability Margin in Pitch. The System also has a Short-Period Mode at 3.85 (rad/sec)

6.3.3.1 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 getting erased in 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 a wind-gust velocity input in (feet/sec) and other outputs, and it will be used in simulations. The direction of the gust is different than 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  13
----------------------------------------------------------------------------------------------------
```

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 direction angle (ξ). The heading angle can be controlled by a coordinated roll and yaw command that can be 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 them independently in order to minimize the β-transients. It is also good idea to include simple 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. The additional state variables ϕ-integral, ψ-integral and $\delta_{aileron}$ and δ_{rudder} can easily be included in the LQR optimization because they are all measurable for feedback.

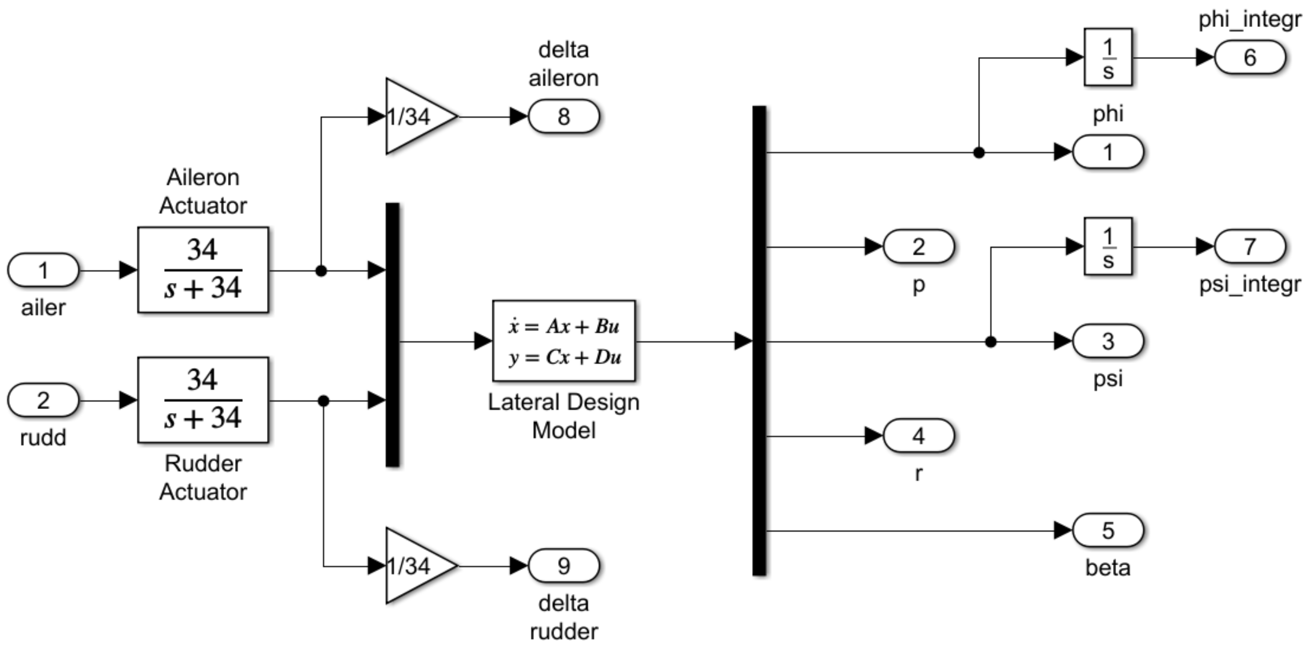


Figure 6.3.8 Augmented Lateral Design Plant for LQR Control Design

Figure 6.3.8 shows the augmented plant for the Roll/ Yaw LQR control design. The interconnection dataset below combines the 5 subsystems together and generates the augmented system, which is: "*Augmented Lateral Design Model*". The sequence of the states, however, is not the same as the outputs sequence and it is modified by reordering the states to "*Augmented Lateral Design Model-2*" 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)
-------------------------------------------------------------------------------
```

```
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*" performs the LQR control design using the plant "*Augmented Lateral Design Model-2*". It uses the C matrix for 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
```

6.3.3.2 Lateral Simulation

The closed-loop simulation model "*Lateral_Sim.mdl*" in Figure 6.3.9 is located in folder "*Control Analysis\LQG\ Examples\Missile Control Design\Lateral LQR*" and it is used to analyze the system's response to gusts and to heading commands. The 9-state-feedback loop is closed 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). The heading direction is $\xi=\psi+\beta$. The heading error is converted into a simultaneously applied roll and yaw attitude command which makes the vehicle to perform a coordinated roll/ yaw turn with minimal β-transient.

In Figure 6.3.10 the vehicle is commanded to perform a 10^o increment in ξ which is achieved by a coordinated roll and yaw command to minimize the β-transients because they are undesirable at high Q_{bar}. In Figure 6.3.11 the missile is excited by a lateral wind-gust velocity impulse along the –y direction and it responds by rotating the aerosurfaces.

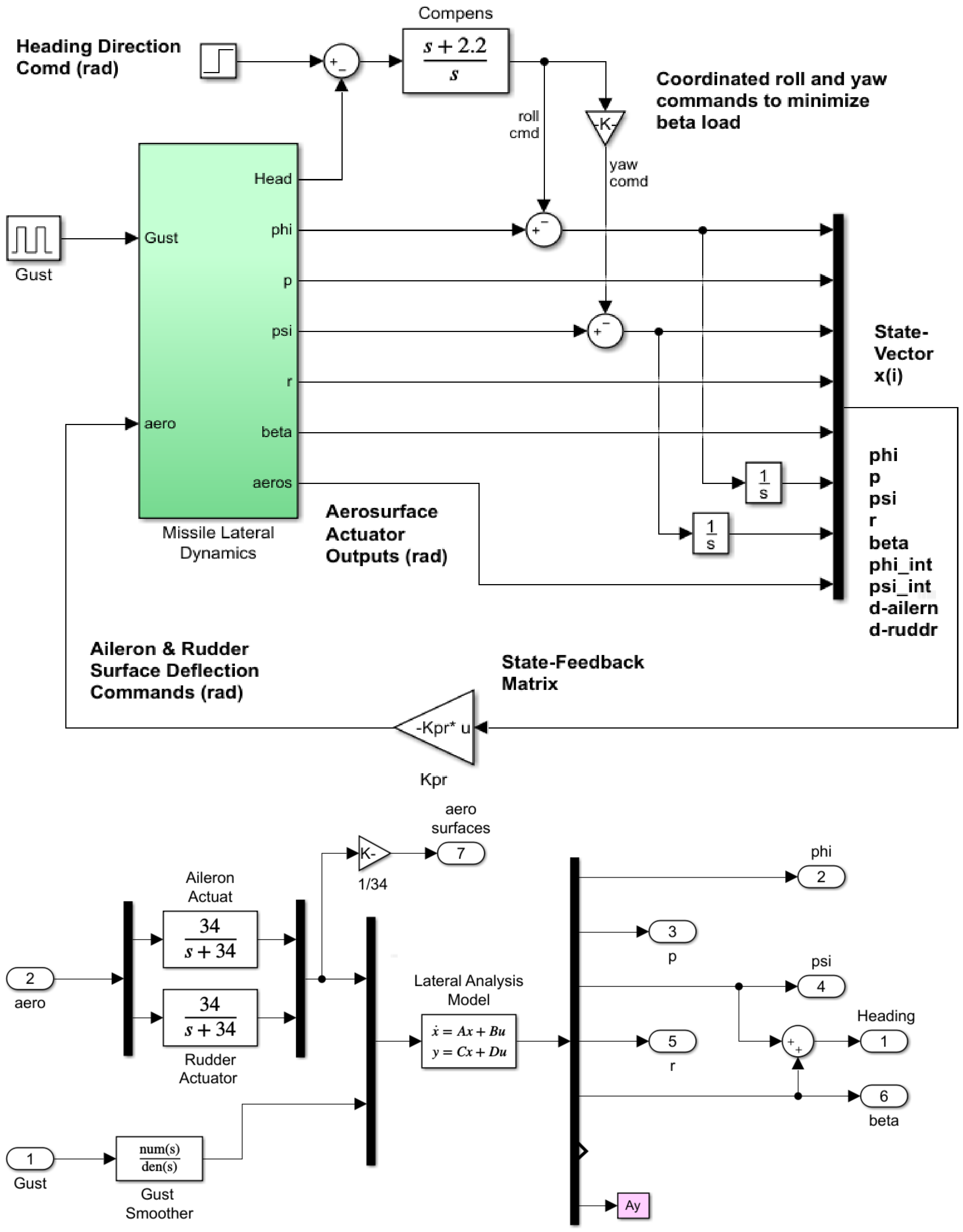


Figure 6.3.9 Roll and Yaw Axes Closed-Loop Simulation Model "Lateral_Sim.mdl"

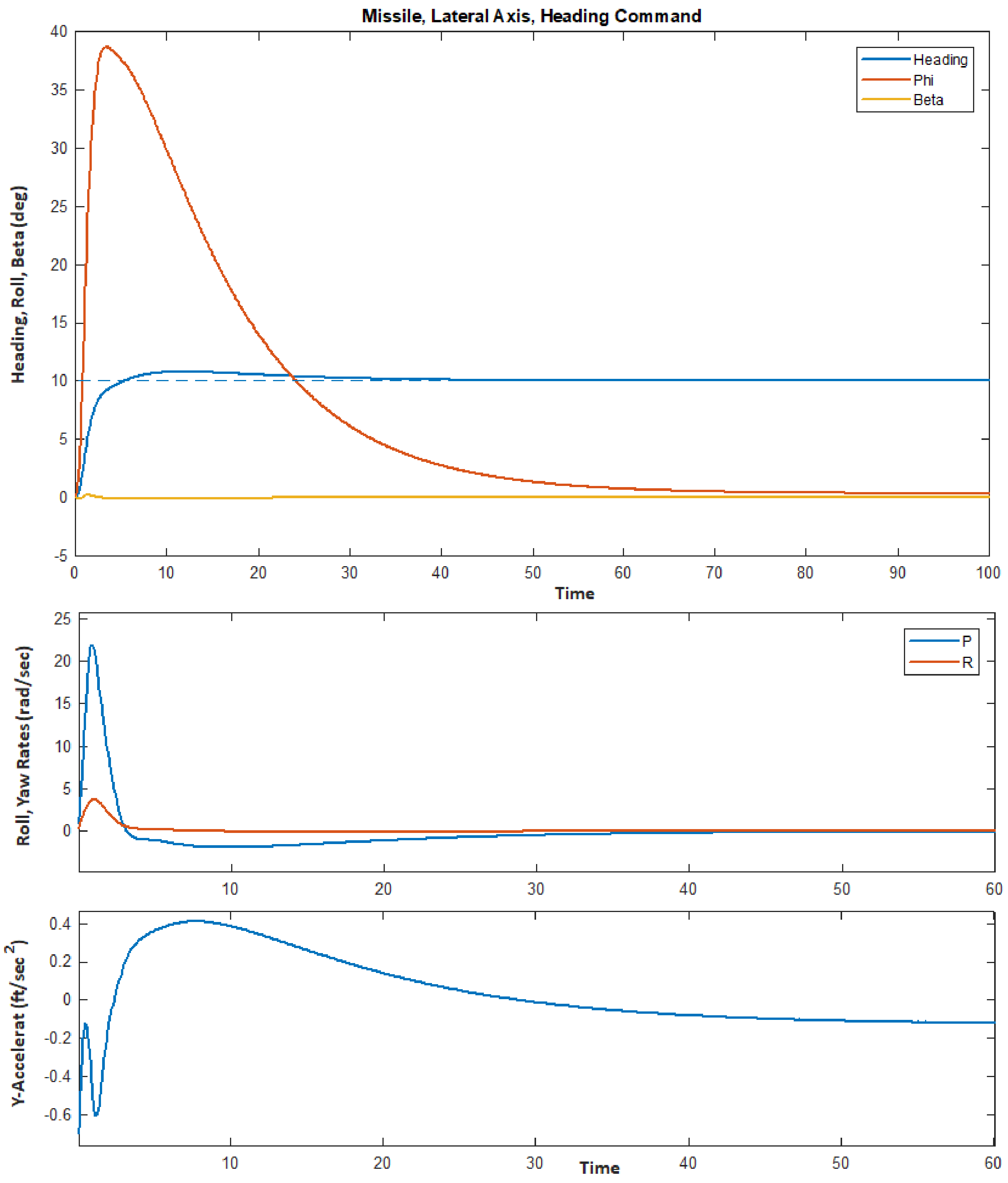


Figure 6.3.10 Missile Responds to 10^o Heading Command. Performs Coordinated Roll/ Yaw Maneuver (mostly roll) to Change its Heading. Beta Transient is minimized by the Coordinated Roll/ Yaw Turn

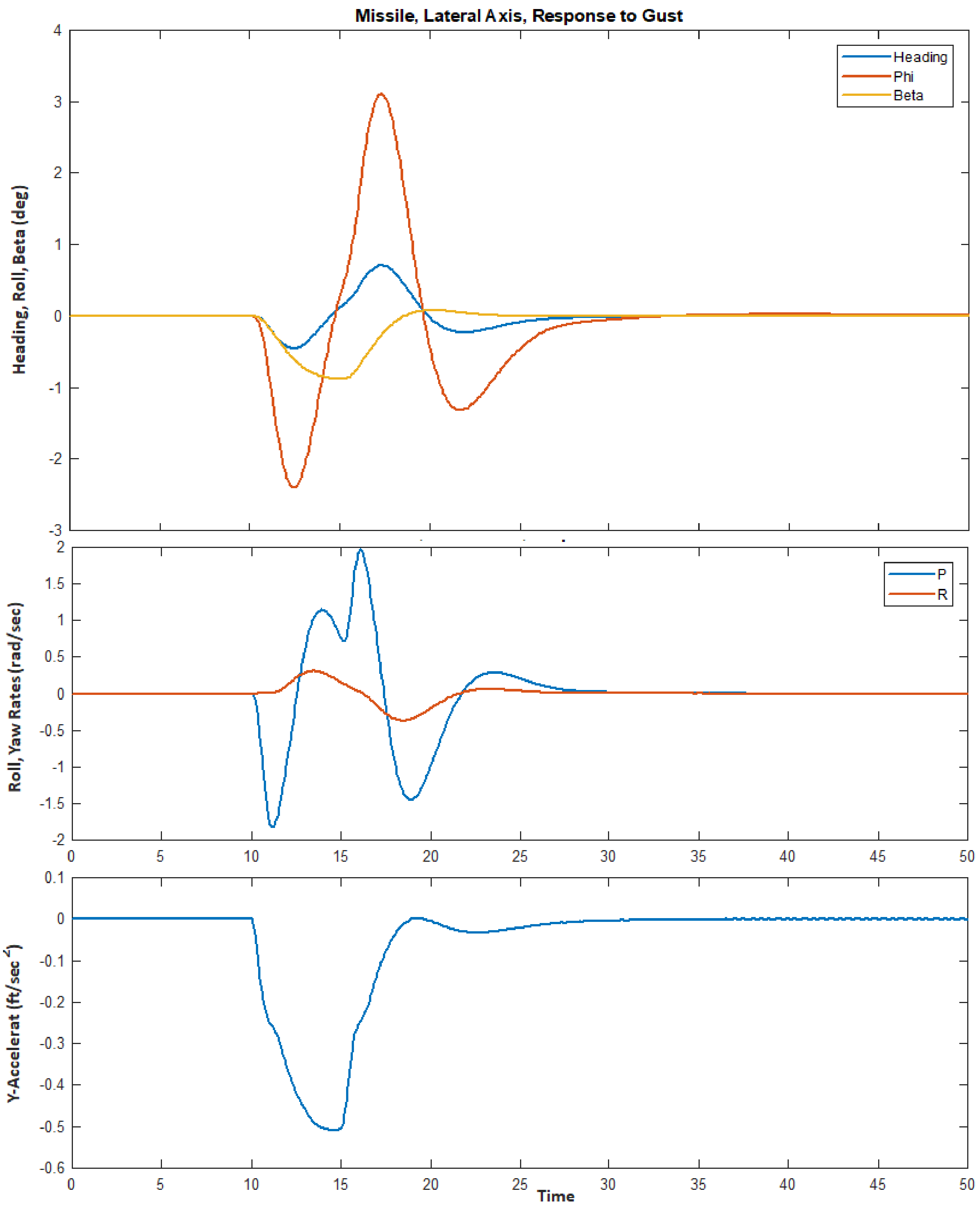
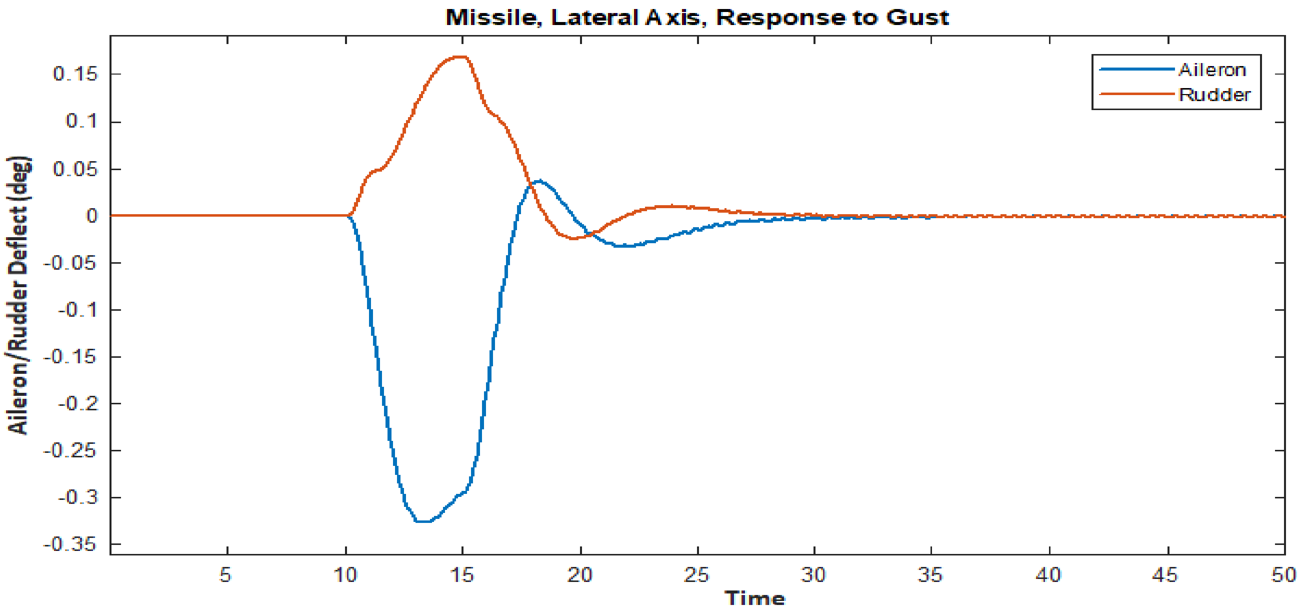


Figure 6.3.11 Missile is excited by Lateral Wind-Gust from the right side, along –Y, causing –Y Acceleration. It also causes Negative Roll and Positive Yaw due to the Vertical Stabilizer. Beta is Initially Negative because it does not see the Wind due to (NoWind Alpha/ Beta) Definition in the Data. The aileron and Rudder accordingly respond to counteract the Vehicle Roll and Yaw Rates



6.3.3.3 Roll/ Yaw Stability Analysis

The system stability is analyzed in the frequency domain by calculating the Nichols plot from the open-loop Simulink model "*Open_Lateral.mdl*" shown in Figure 6.3.12. The script file "frequ.m" calculates the frequency response across one of the loops, the one which is opened while the other loop is closed. The aileron is opened and the rudder is closed in this case to analyze roll axis stability. The model is modified to check the yaw loop stability by opening the rudder and closing the aileron loops. The model includes the two actuators and low-pass filters. The loop is broken between the low-pass filter output and the corresponding actuator input. Figure 6.3.13 shows the LQR control system's stability in the Roll and Yaw directions.

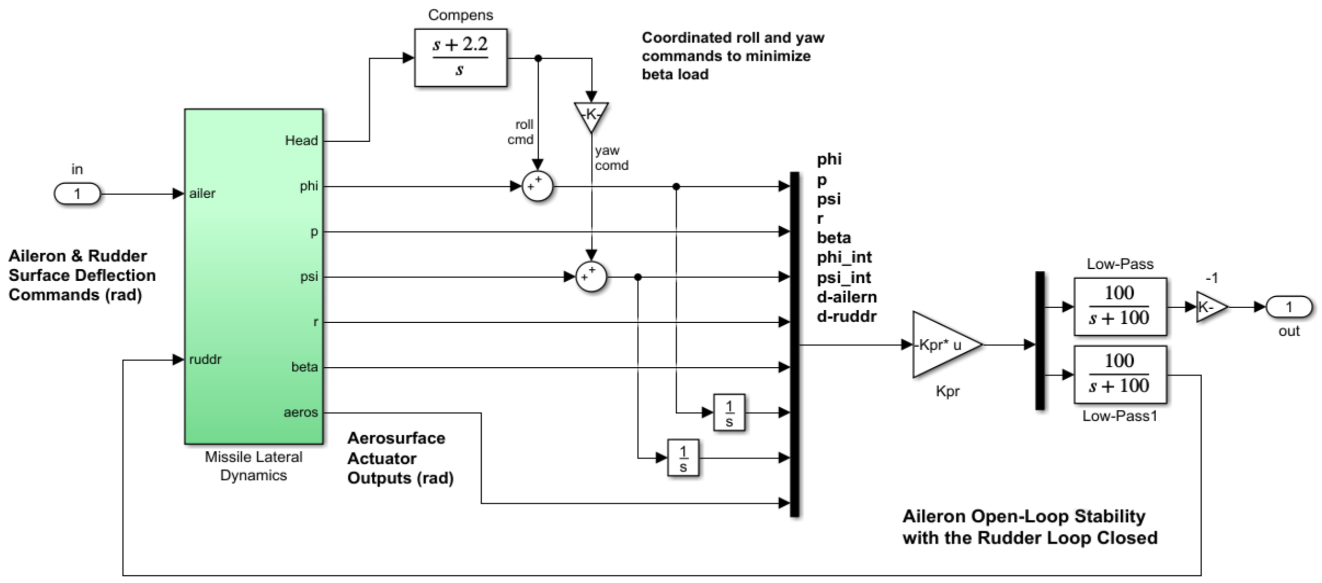


Figure 6.3.12 Open-Loop Model "Open_Lateral.mdl" used for Roll and Yaw Stability Analysis

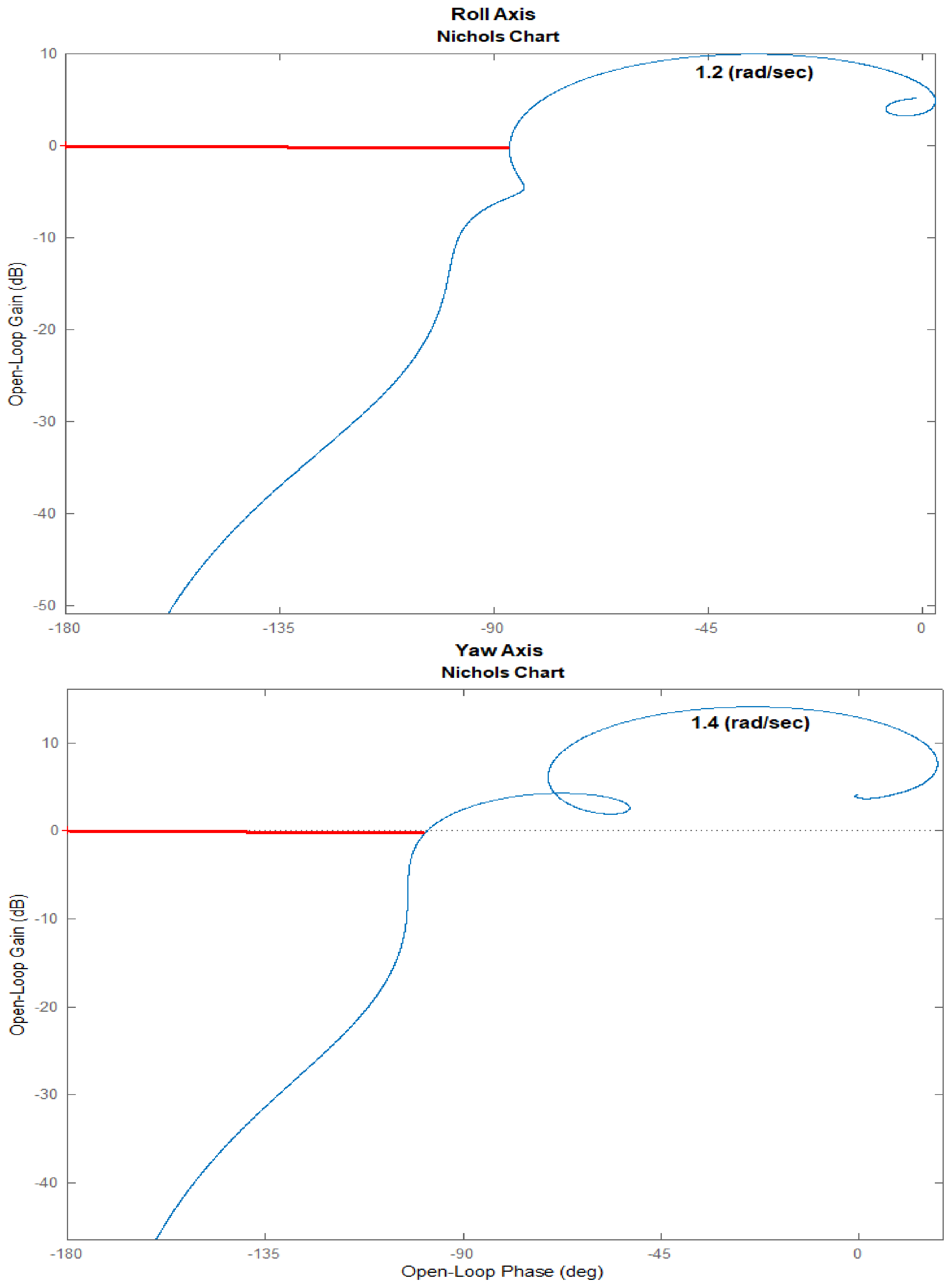


Figure 6.3.13 Nichols Plots Showing Stability of the LQR Control System in the Roll and Yaw Axes. The System also has a Short-Period Modes at 1.2 and 1.4 (rad/sec)