

# Autonomous Landing of an Un-Powered Glider Vehicle with Fuel Sloshing Tanks



In this example we will develop dynamic models, design, analyze and simulate the autonomous landing of an un-powered Shuttle type of reentry vehicle that descends from high altitude and lands on a runway using an autonomous landing guidance system. This vehicle uses Elevon and Speed-Brake for longitudinal control, and Aileron and Rudder for lateral control. The dynamic model is further complicated by the sloshing of the residual reaction control propellant inside two tanks that has the tendency to destabilize the vehicle.

# 1. Introduction

In the analysis that follows we will analyze and simulate the autonomous landing of an un-powered rigid-body glider vehicle starting from an altitude of 9000 (feet) and a speed of 500 (feet/sec). We will create longitudinal and lateral dynamic models of the vehicle at a fixed point of the trajectory, combine it with the flight control and guidance systems, analyze the system stability in frequency domain, in both pitch and lateral, and simulate it landing on the runway, at sea level. The flight vehicle parameters are obtained from a point mass trajectory, and they correspond to a flight condition where the vehicle is at an altitude of 2700 (feet), speed 500 (ft/sec), dynamic pressure 210 (psf), and at Mach 0.3. The simulation assumes that the vehicle parameters are constant. It begins when the vehicle is aligned with the runway at an altitude of 9000 feet and it ends 100 seconds later at zero altitude.

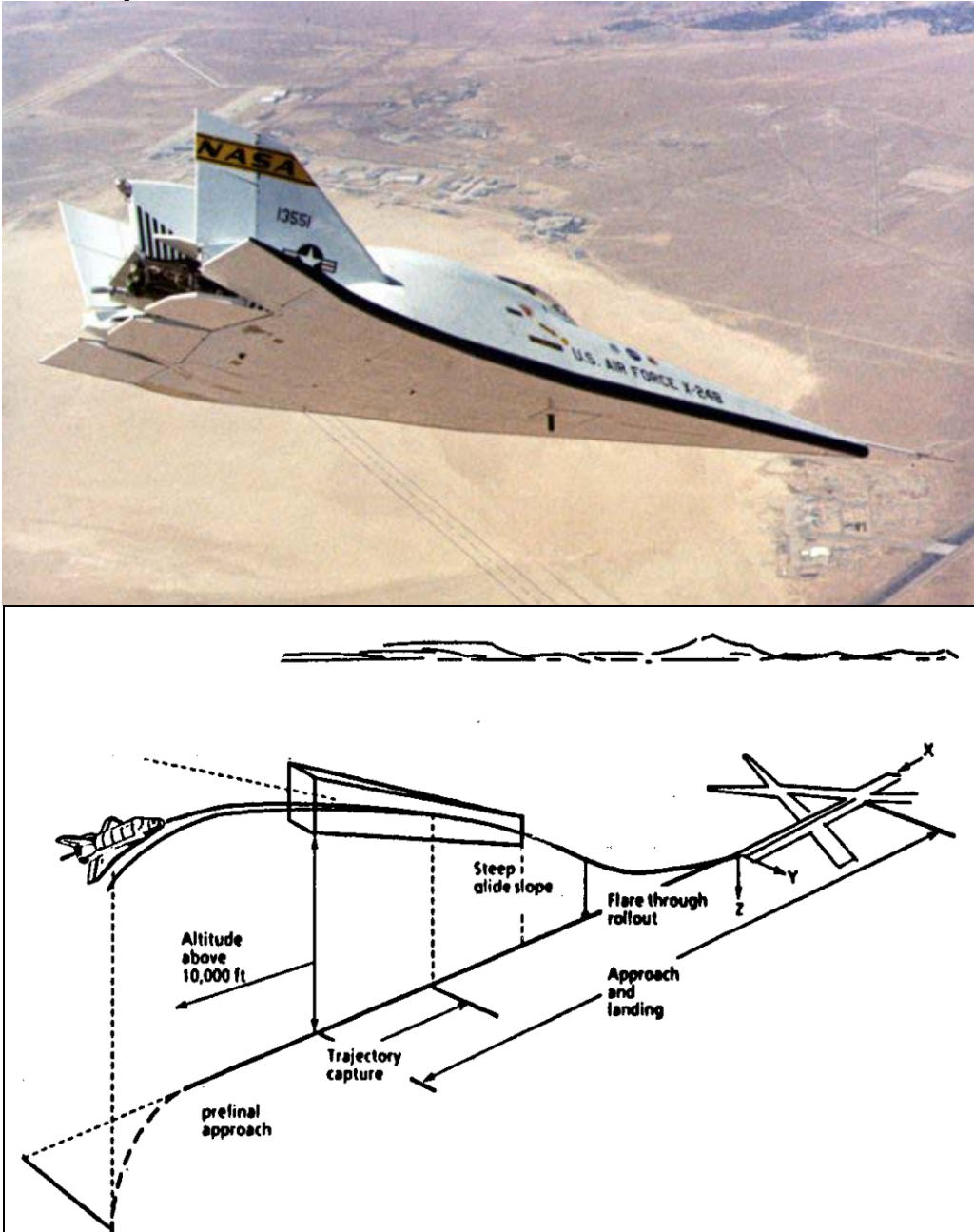


Figure 1 Approach and Landing

## 2. Propellant Sloshing Model

The vehicle has two identical propellant tanks which are symmetrically positioned on the left and right sides of the vehicle, as shown in Figure 2. They contain residual RCS fuel used for attitude control while in space. The propellant in each tank is sloshing in two directions parallel to the liquid surface, which is perpendicular to the total acceleration vector  $A_T$ , and the slosh mass motion generates disturbance forces on the vehicle that degrade the system stability and performance. The slosh forces are oscillatory and low damped. Because of the elongated parabolic shape of the tanks in this configuration, the frequencies in the longitudinal and lateral directions are not the same. The longitudinal slosh frequency (which is oscillating in the vehicle x-z plane) is  $\omega_{zs}=2.1$  (rad/sec) and the damping coefficient is  $\zeta_{zs}=0.008$ . The lateral slosh frequency (oscillating along the y vehicle direction) is  $\omega_{ys}=3.6$  (rad/sec) and the damping is  $\zeta_{ys}=0.004$ . The slosh frequencies in the input data correspond to 1-g acceleration loading. The actual frequencies in the model are higher because the total sensed acceleration  $A_T$  is 1.55(g). The slosh masses are 202 (slugs) each, and their undeflected positions at steady state are  $\pm 4.5$  (feet) to the left and the right of the vehicle x centerline. The slosh masses are also 7.5 (ft) behind, and 2.3 (ft) above the vehicle CG.

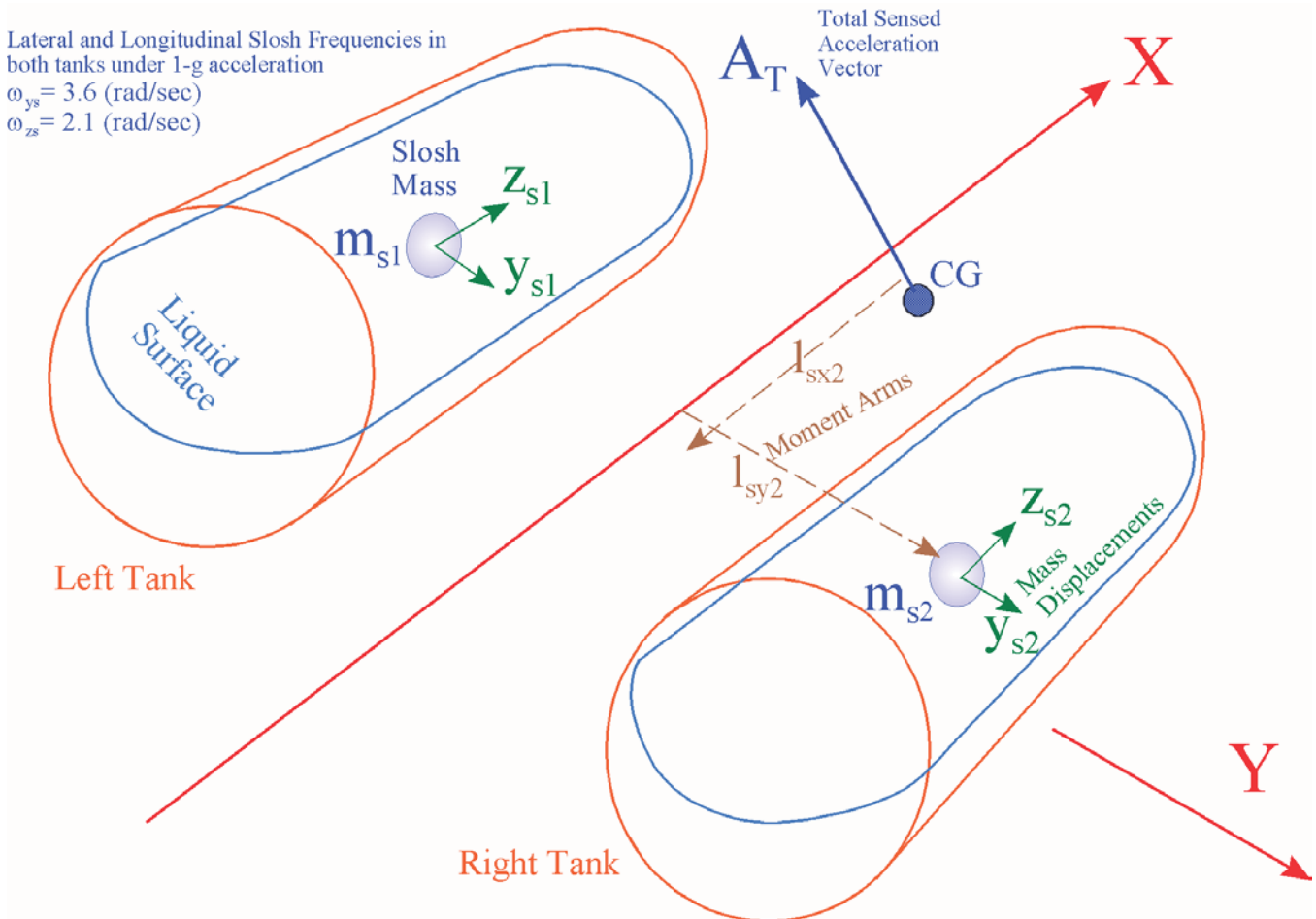


Figure 2 Geometry of the Propellant Tanks

### 3. Data Files

The data files used in this example are located in directory: “\Flixan\Examples\ Autoland”. The input data file “Autoland.Inp” includes the flight vehicle input data, the actuator models, the flight control, guidance, and other model interconnection data. The output file “Autoland.Qdr” contains the vehicle and other related state-space models which are exported into Matlab. The title of the flight vehicle data is “Automatic Landing of Unpowered Vehicle”, and it is processed by the vehicle modeling program to generate the vehicle state-space system under the same title. The vehicle parameters were obtained from a point-mass trajectory. From this system we construct separate pitch and lateral subsystems which are exported into Matlab and used in simulations and frequency domain analysis.

For longitudinal analysis the input file includes two sets of state-space system reconditioning data. The first set extracts the longitudinal system from the original coupled (pitch & lateral) system that was created by the flight vehicle program. Its title is “Automatic Landing of Unpowered Vehicle, Pitch Axis”. The second set of system reconditioning data “Automatic Landing of Unpowered Vehicle, Pitch, New Outputs with Slosh” further modifies the extracted pitch system by including additional outputs. The reason is because additional outputs are needed for control, such as: velocity, altitude, and altitude rate which are not available in the first model. We also need outputs in order to monitor the slosh mass displacements relative to the tanks. The input file “Autoland.Inp” also includes three system interconnection data sets, (a) for generating a system that has only the flight control loop closed “Vehicle Landing, with Closed Pitch Flight Control Loop (Open Guidance)”, (b) a system that has both, the flight control and the guidance loops closed “Vehicle Landing, with Closed Loop Pitch Guidance”, and (c) a system used for open-loop pitch stability analysis “Elevon Open-Loop System (Speed-Brake Closed)”.

For lateral analysis the input data file includes three sets of state-space system reconditioning data: (a) a set “Automatic Landing of Unpowered Vehicle, Lateral Design Model” that extracts a lateral subsystem from the original vehicle system and it is used for state-feedback LQR design, (b) a data-set used for generating a simulation system “Automatic Landing of Unpowered Vehicle, Lateral Axes” used in Matlab, and (c) a similar set that modifies the previous system (b) that includes also the slosh mass displacement outputs “Automatic Landing of Unpowered Vehicle, Lateral New Outputs with Slosh”. The input file includes also a batch set for fast processing the data sets and generating the systems in batch mode. Its title is “Batch set for calculating Pitch and Lateral models of the Auto-Landing Vehicle” and it is located at the top of file “Autoland.Inp”. The state-space systems generated by the Flixan program are saved in the systems file “Autoland.Qdr”. Some of the systems are also converted to Matlab format for further analysis in subdirectories: “\Mat Pitch” and “\Mat Lateral”.

### 4. Longitudinal Analysis

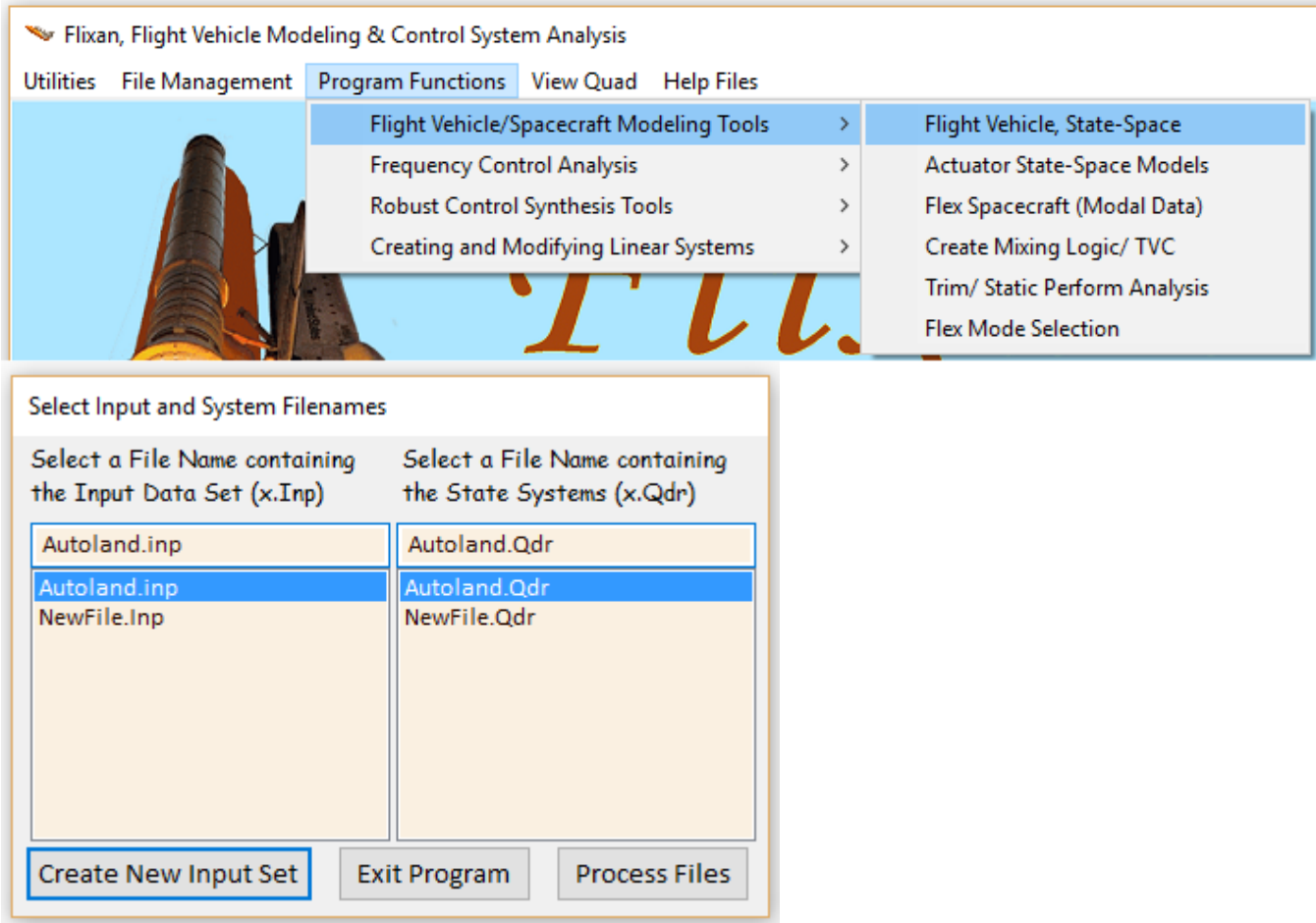
The longitudinal control and guidance systems consist of two loops which are strongly coupled together and control the vehicle altitude and the speed of descent. The flight control system uses the Elevon and Speed-Brake aerosurfaces.

#### 4.1 Using Flixan to Create the Longitudinal Models

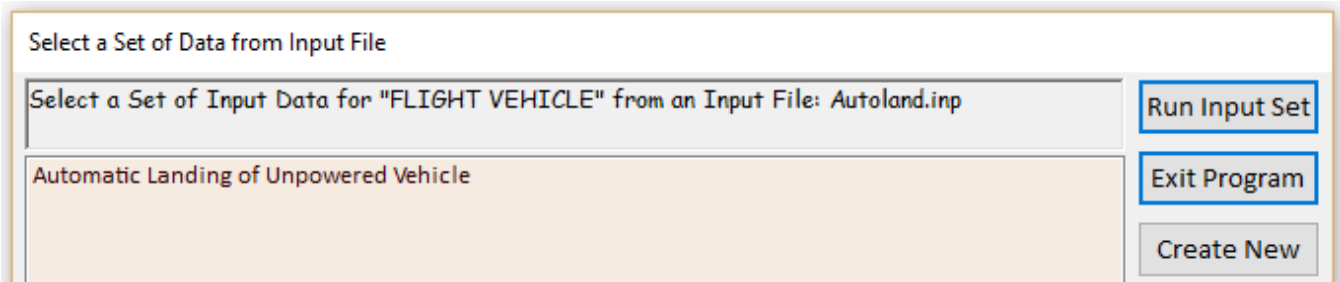
Although the entire input file can be processed in batch mode, for tutorial purposes, in the following sections we shall demonstrate the processing of each data-set separately by running the corresponding Flixan utilities.

## Vehicle Model

To process the vehicle data and generate the vehicle state-space model, you must first start the Flixan program, go to the project directory “*Flixan\Examples\Autoland*”, and from the main menu go to “*Flight Vehicle/ Spacecraft Modeling Tools*” and then “*Flight Vehicle, State-Space Modeling*”. From the following filename selection menu, select the filenames for the vehicle input data “*Autoland.Inp*” and the systems file “*Autoland.Qdr*” and click on “*Process Files*”.



The program opens the input data file and searches for a flight vehicle data set. It finds only one set and shows its title “*Automatic Landing of Unpowered Vehicle*” in the following menu. Select the title and click on “*Run Input Set*” to process the vehicle data. Then the program provides the following dialog containing the vehicle parameters from file “*Autoland.Inp*”.



Flight Vehicle Parameters

**Vehicle System Title**  
Automatic Landing of Unpowered Vehicle

**Number of Vehicle Effectors**  
Gimbaling Engines or Jets. Include Tail-Wags-Dog? 0 WITH TWD WITHOUT TWD  
Rotating Control Surfaces. Include Tail-Wags-Dog? 4 WITH TWD WITHOUT TWD  
Reaction Wheels? 0  
Single Gimbal CMGs? 0

**Momentum Control Devices**  
Include a 3-axes Stabilized Double Gimbal CMG System? Yes No

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

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

**Number of Modes**  
Structure Bending 0  
Fuel Sloshing: 2

Reaction Wheels Single Gimbal CMGs Double Gimbal CMG System Slewing Appendages Gyros Accelerometer Aero Sensors Fuel Slosh Flex Modes User Notes  
Mass Properties Trajectory Data Gust/ Aero Paramet. Aero Force Coeffs Aero Moment Coeffs Control Surfaces Gimbal Engines/ RCS External Torques

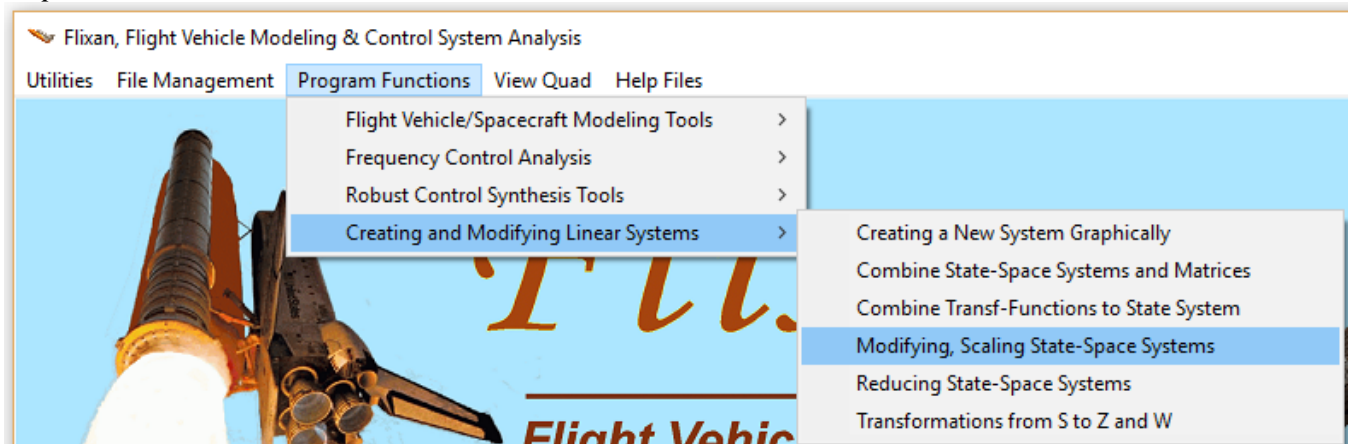
**Vehicle Mass Properties**

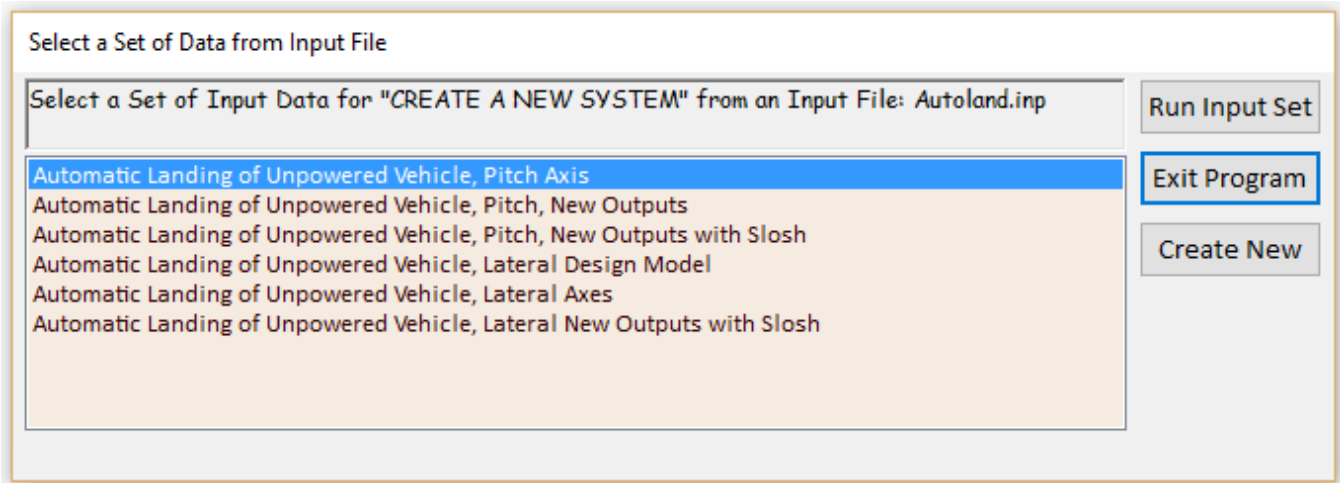
Moments/ Products of Inertia (slg-ft <sup>2</sup> )		Location of Center of Gravity (ft)		Vehicle Mass in (slugs)	
Ixx	169500.0	Ixy	0.000000	Xcg	-62.67000
Iyy	1978800.	Ixz	131.0000	Ycg	0.000000
Izz	1975300.	Iyz	0.000000	Zcg	1.290000
				Accelerat. due to Gravity (ft/sec <sup>2</sup> )	32.17000
				Earth Radius (Re) in (feet)	0.2089600E+08

Click on “Run” and the program will generate the state-space system for this vehicle configuration and it will save it in the systems file “Autoland.Qdr” under the same title as the input data title. It will also include the comments below the title and the definitions of the vehicle inputs, states, and outputs below the state-space matrices.

### Extract the Pitch Subsystem

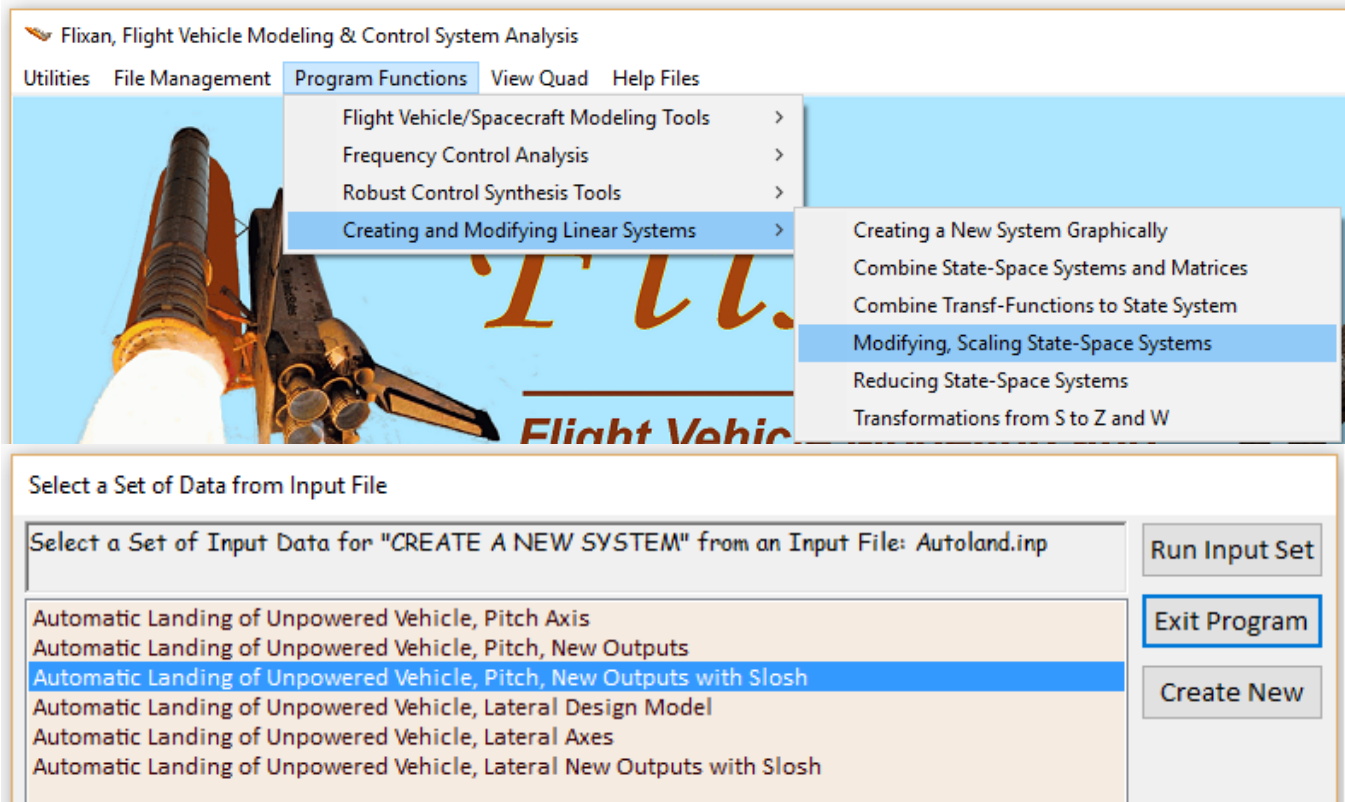
To extract the pitch subsystem from the previously created coupled vehicle using Flixan, you must first select the project directory “\Flixan\ Examples\ Autoland”, and from the main menu select “Program Functions”, “Creating and Modifying Linear Systems”, and then “Modifying/ Scaling Systems” to run the system modification program. Then you must select the filenames for the vehicle input and system files “Autoland.Inp” and “Autoland.Qdr”, as before. The following menu shows the titles of the system reconditioning sets that already exist in file “Autoland.Inp”. Select the input data-set “Automatic Landing of Unpowered Vehicle, Pitch Axis” which provides instructions on how to extract the pitch subsystem, and click on “Run Input Set”. This utility will extract the pitch subsystem from the original vehicle system and save it in file “Autoland.Qdr” under the same title “Automatic Landing of Unpowered Vehicle, Pitch Axis”.





### Augmenting the Pitch Subsystem Outputs

The previously generated pitch subsystem is missing some important outputs and for simulation purposes we must introduce them in the pitch model. The additional outputs are used for implementing the guidance control loop and for monitoring the activity of the sloshing mass inside the propellant tanks. The additional outputs come from states and from output derivatives of the previously extracted pitch subsystem. The systems modification program is used again for the second time and it will augment the pitch subsystem “*Automatic Landing of Unpowered Vehicle, Pitch Axis*”. The data-set containing the instructions on how to extract the new pitch system from the previous and to introduce the new outputs is also included in the input file. Its title is: “*Automatic Landing of Unpowered Vehicle, Pitch, New Outputs with Slosh*”. The processing of this data-set is very similar to that described in the previous section. The modified pitch subsystem will be saved in file “*Autoland.Qdr*” using the same title.



### Actuator Transfer Function:

The systems file “Autoland.Qdr” includes a simple actuator subsystem for the aerosurfaces. They are two  $20/(s+20)$  transfer functions for the Elevon and for the Speed-Brake aerosurfaces, already converted into state-space form and implemented in a single subsystem. Its title is: “Control Surface Actuators  $20/(s+20)$ ”. The conversion from transfer-function to state-space form is not described here. The batch set prevents it from getting destroyed when running the batch because there is a “Retain System” statement included in the batch.

### Pitch Guidance

The pitch guidance system is shown in Figure 3. It is implemented in file “Autoland.Inp” by transfer function interconnection data, consisting of transfer functions, gains, and definitions of the system inputs and outputs. Its title is “Guidance (Altitude and Speed Control)”, and must be converted to state-space form in order to be combined with the vehicle system.

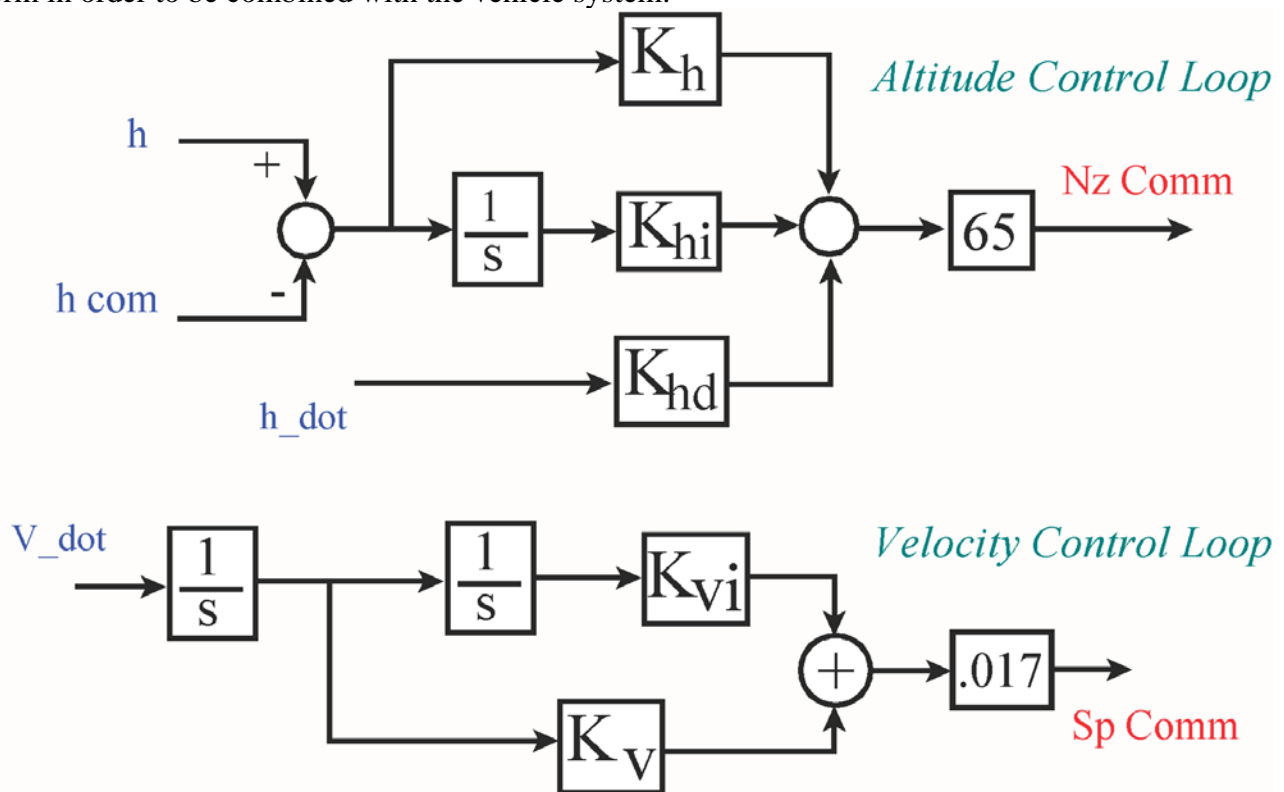
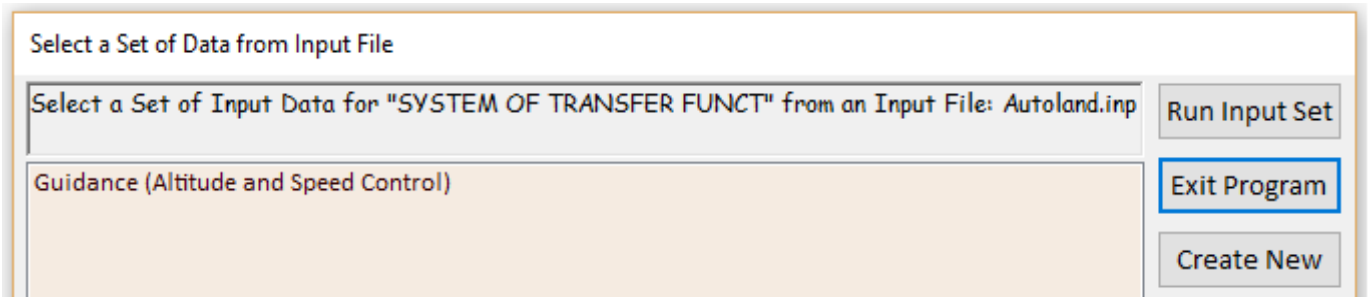
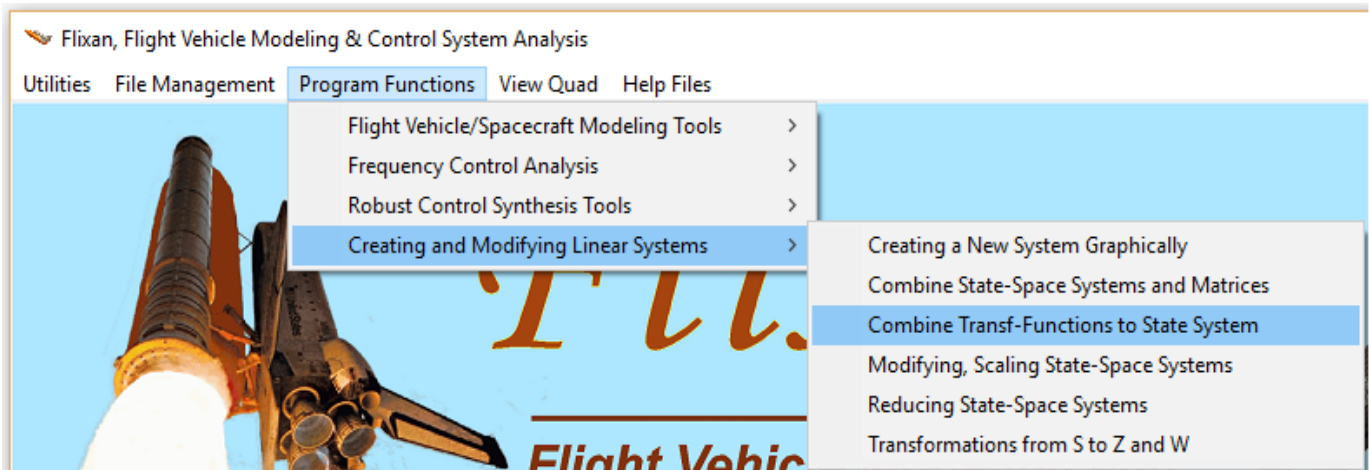


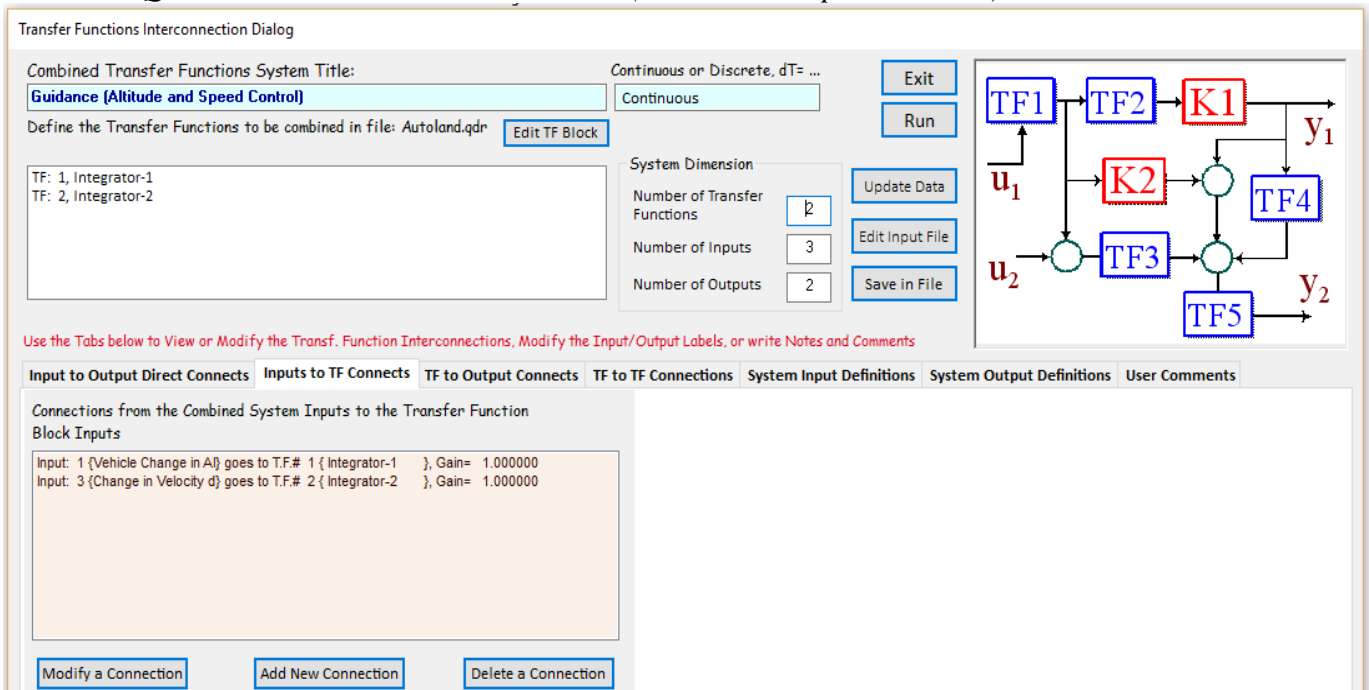
Figure 3 Altitude and Speed Control Guidance System

To run the transfer function program, start Flixan and select the project directory. From the Flixan main menu, go to “Program Functions”, “Creating and Modifying Linear Systems”, and then “Combine Transfer-Functions to Create State Systems”, as shown below. Then you must select the input and system filenames “Autoland.Inp” and “Autoland.Qdr”, as before. The following menu shows the transfer-function data-sets that are saved in the input file. There is only one, select the title “Guidance (Altitude and Speed Control)” and click on “Run Input Set”.





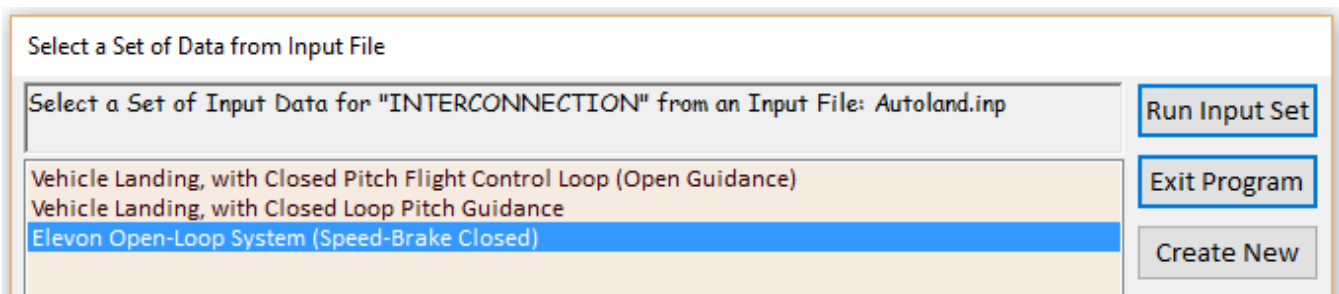
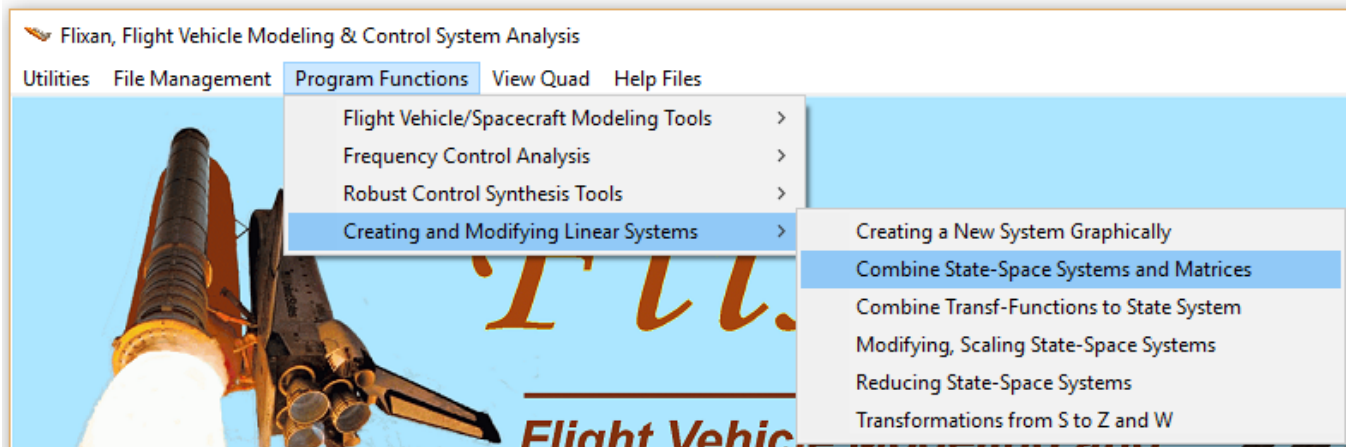
The following transfer function interconnection dialog appears. It has tabs that show the interconnection data between the transfer functions, the system inputs, and the outputs. Click on “Run” and the program will process the transfer-function data, convert it into a state-space system, and save it in the systems file “*Autoland.Qdr*” under the same title: “*Guidance (Altitude and Speed Control)*”.



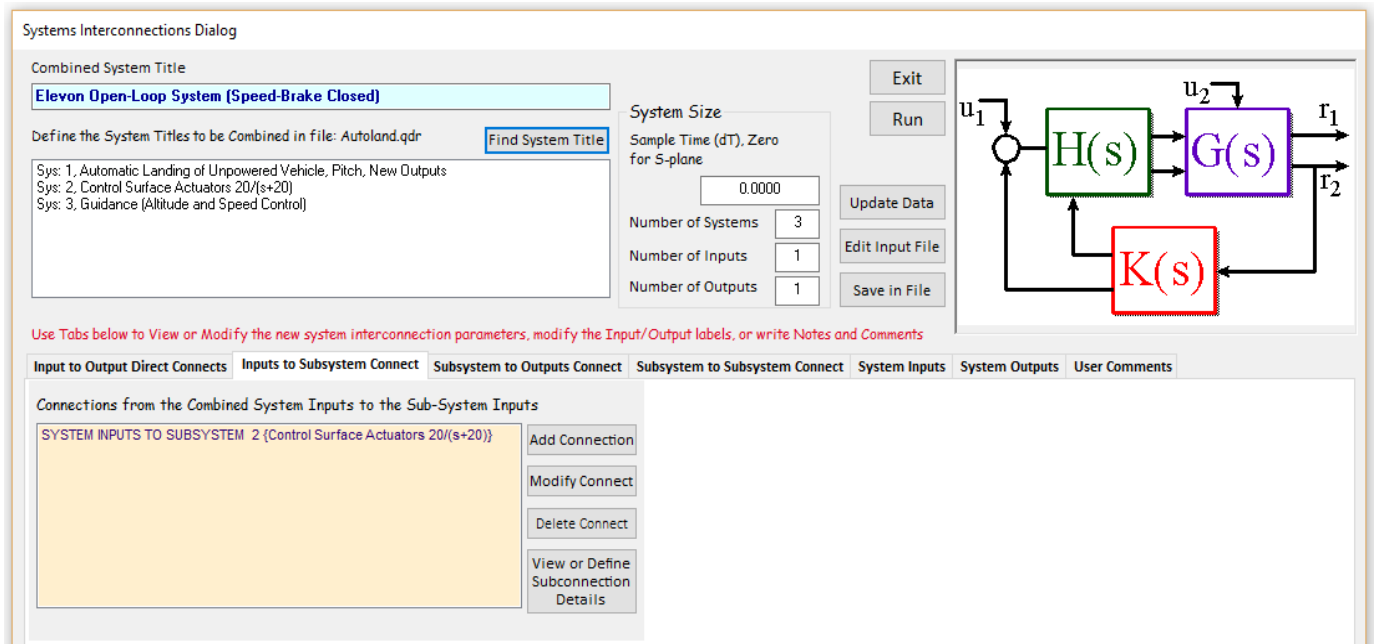
## Creating an Open-Loop System for Stability Analysis

We must also create an open-loop system to be used for analyzing stability margins in the frequency domain. This system consists of three subsystems which are already saved in the systems file and the control loop is opened at the Elevon. The three subsystems to be combined are: (a) the actuator model “*Control Surface Actuators 20/(s+20)*”, consisting of two first order transfer-functions, (b) the previously created guidance subsystem, and (c) the pitch vehicle subsystem “*Automatic Landing of Unpowered Vehicle, Pitch, New Outputs*” (an earlier version that includes the slosh states but not the slosh outputs). The title of the subsystem interconnections set in file “*Autoland.Inp*” is “*Elevon Open-Loop System (Speed-Brake Closed)*”.

To run the systems combination program, you must first start the Flixan program and select the project directory “*Flixan\Examples\Autoland*”. From the Flixan main menu, go to “*Program Functions*”, “*Creating and Modifying Linear Systems*”, and then “*Combine State-Space Systems and Matrices*”, as shown below. Select the input and system filenames “*Autoland.Inp*” and “*Autoland.Qdr*” as before. The following menu shows the systems interconnection data-sets which are included in the input file. Select the title “*Elevon Open-Loop System (Speed-Brake Closed)*” and click on “*Run Input Set*”.



The following systems interconnection dialog appears. It has tabs that show the interconnection data between the three subsystems, their inputs and outputs. Click on “*Run*” and the program will process the data, combine the three systems into a single state-space system, and save it in the systems file “*Autoland.Qdr*” under the same title: “*Elevon Open-Loop System (Speed-Brake Closed)*”.

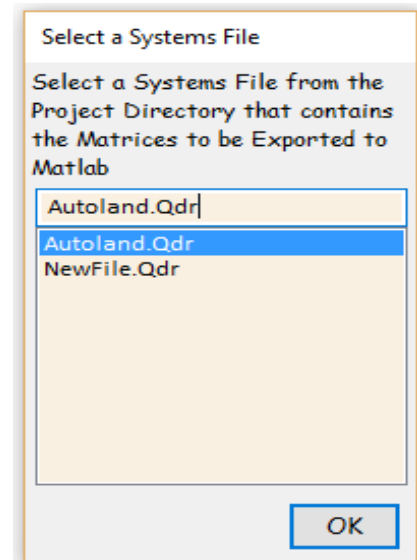
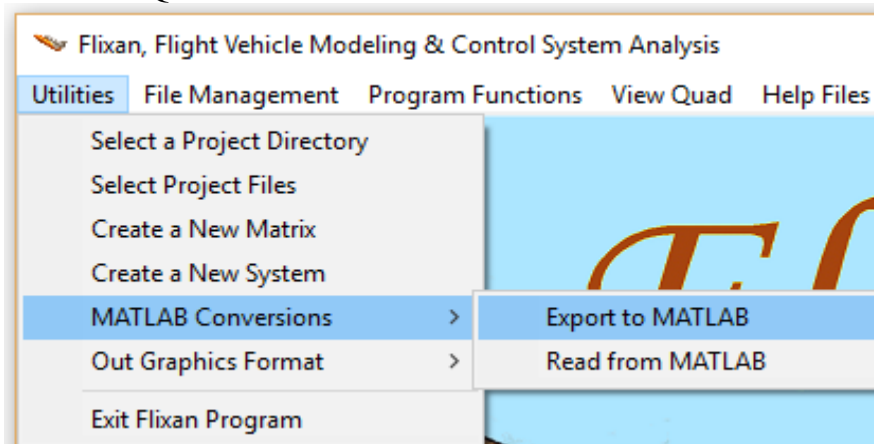


## Additional Systems

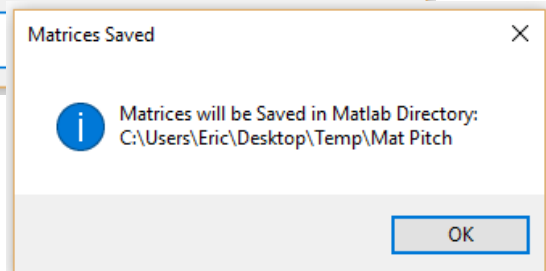
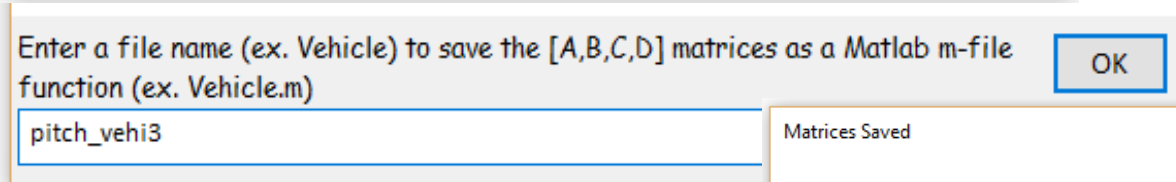
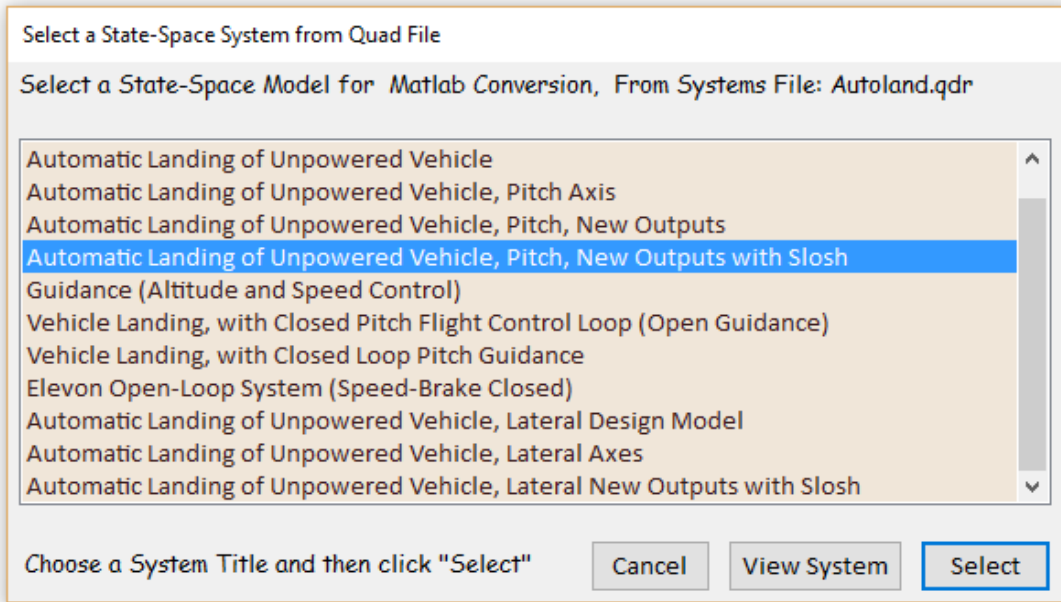
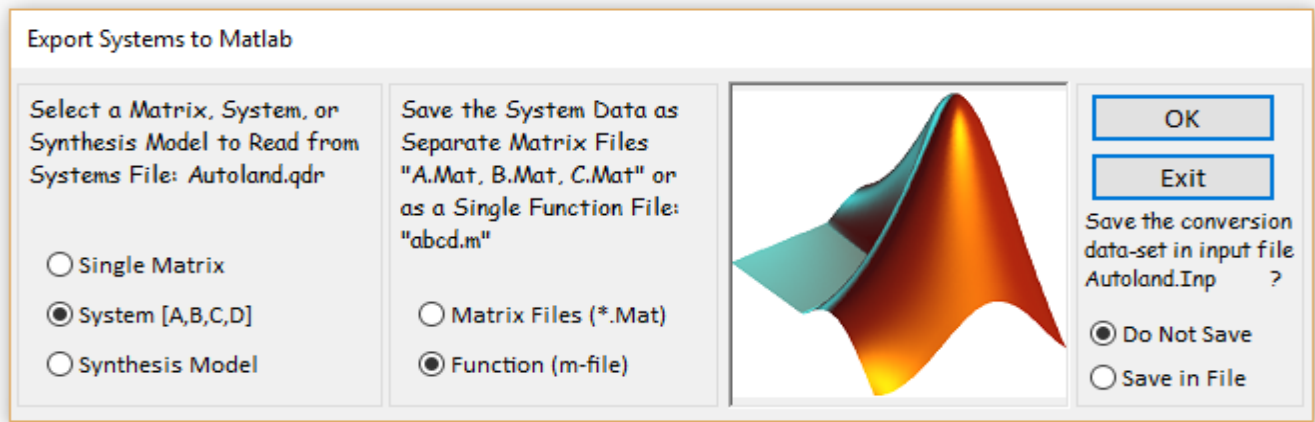
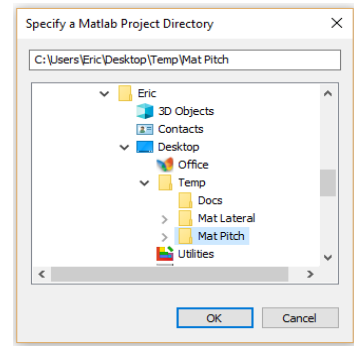
The files “Autoland.Inp” and “Autoland.Qdr” contain interconnections and data for two additional closed-loop systems “Vehicle Landing, with Closed Pitch Flight Control Loop (Open Guidance)” and “Vehicle Landing, with Closed Loop Pitch Guidance”. They are generated by Flixan systems combination program and used for simulation purposes. These systems, however, will not be discussed here because the simulation models consist of subsystems connected together via Simulink models.

## Exporting Pitch Subsystems to Matlab

The pitch simulation system “Automatic Landing of Unpowered Vehicle, Pitch, New Outputs with Slosh” which is saved in systems file “Autoland.qdr” must also be converted to a function format that can be loaded into Matlab subdirectory “\Autoland\Mat Pitch” to be used for further analysis and simulations in Matlab. Start Flixan and from the top menu select “Utilities”, “Matlab Conversions”, and then “Export to Matlab”, as shown below. From the systems filename selection menu, select “Autoland.Qdr” and click “OK”.



Select the directory where the Matlab analysis will be performed and the system function will be saved in “\Examples\Autoland\Mat Pitch”. Use the “Export to Matlab” dialog below to define the type of conversion, a “System (A,B,C,D)” to become a “Function m-file”. The following menu contains the titles of systems in file “Autoland.Qdr”. Select the title “Automatic Landing of Unpowered Vehicle, Pitch, New Outputs with Slosh”. You must also enter the filename “pitch\_vehi3.m” (without the .m) of the Matlab state-space m-function.



## Export the Open-Loop Pitch Analysis System to Matlab

In a similar fashion we repeat the process of converting the stability analysis model “*Elevon Open-Loop System (Speed-Brake Closed)*” which is also saved in the systems file “*Autoland.Qdr*”. This system is converted to a Matlab state-space m-file “*Siso\_Anal.m*” that is also saved in directory “*\Autoland\Mat Pitch*”. It will be used by the script file “*run.m*” to perform frequency response analysis. The pitch guidance system “*Guidance (Altitude and Speed Control)*” is also be converted to an m-function filename “*guidance.m*” and saved in the same subdirectory for pitch analysis using Matlab.

### 4.2 Pitch Flight Control and Guidance Systems

Figure 4 shows the longitudinal flight control and guidance system. The flight control system is designed to control the vehicle normal acceleration  $N_z$  by varying the angle of attack. It combines the  $N_z$  error signal with the pitch rate  $q$  and attitude  $\theta$ , and drives the Elevon actuator. The  $N_z$ -command input to the FCS comes from guidance. There is also a loop that controls the vehicle altitude by issuing an  $N_z$  command to the flight control system. The guidance consists of two loops: an altitude  $h$  control loop, and a speed  $V$  control loop. The altitude control loop is attempting to track a pre-calculated altitude reference ( $h$ -ref) versus time. It uses a PID controller that combines altitude error, integral of altitude error, and altitude rate of change ( $h$ -dot). The speed control loop is a PD controller that regulates the vehicle speed by modulating the speed brake that varies the vehicle drag coefficient.

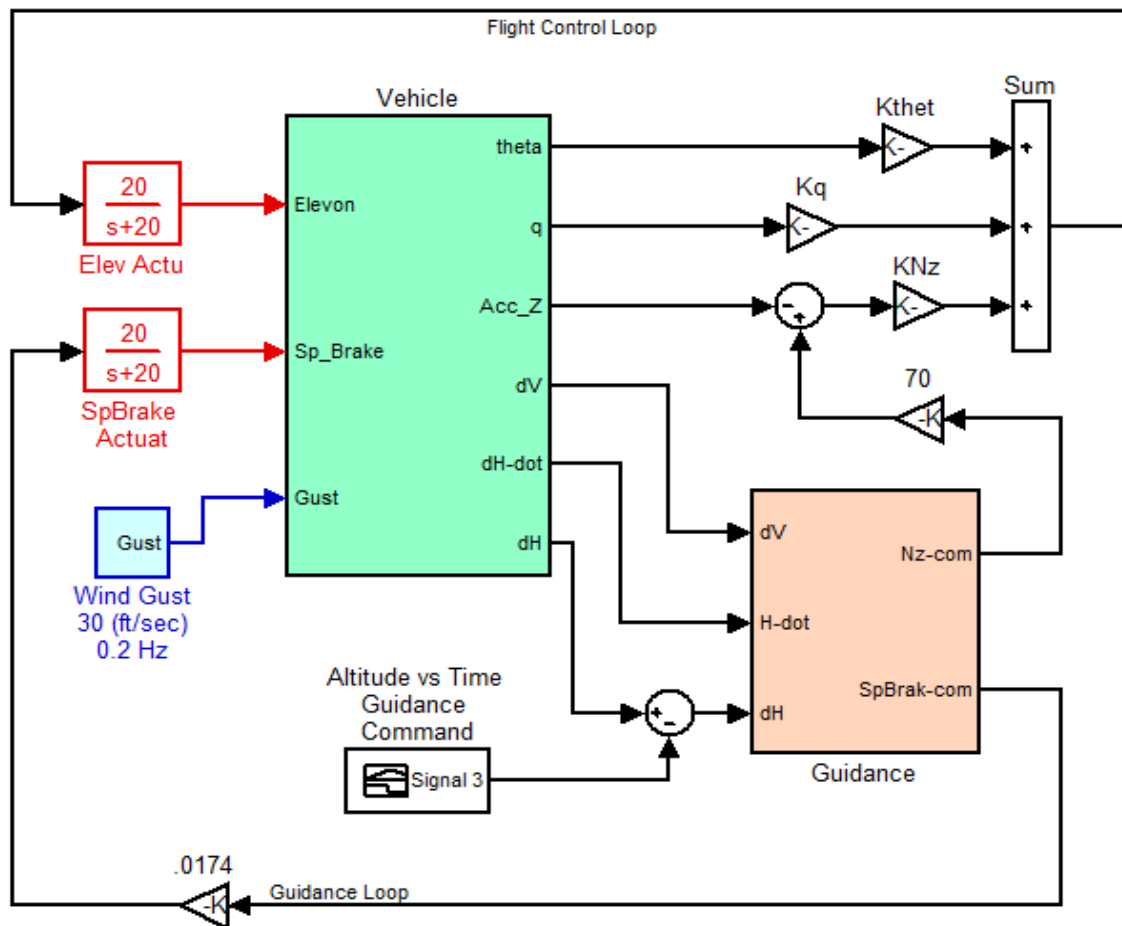


Figure 4 Pitch Flight Control and Guidance System

The pitch vehicle output consists of the following variables: pitch attitude  $\theta$ , rate  $q$ , angle of attack  $\alpha$ , variation in altitude  $\delta h$ , altitude rate  $\dot{h}$ , change in velocity  $\delta V$ , normal acceleration  $N_z$ , and the displacements of the slosh masses  $z_s$  relative to the tank.

### 4.3 Pitch Axis Simulation in Simulink

The Simulink model for simulating the automatic landing of this reentry glider in the longitudinal axes is implemented in file “Autoland.Mdl”, in subdirectory “Examples\Autoland\Mat Pitch”, and it is shown in Figure 4. It includes the pitch guidance subsystem “guidance.m” and the vehicle subsystem “pitch\_veh3.m” that were exported to Matlab. The script file “run.m” is used to load the state-space subsystems to Matlab workspace. The vehicle state vector is also initialized at  $[\theta, q, \alpha, h, \delta V] = [0.1, -0.08, 0.18, 9000, 50]$ , and the slosh mass displacements and velocities at zero. The simulation results are shown in Figures (5.1 to 5.3). It starts when the flight direction is aligned with the runway at an altitude of 9000 ft. Since the vehicle does not have thrust, the only controls available in the pitch axis are the Elevon and the Speed-Brake aerosurfaces. The Elevon controls the vehicle altitude by varying the angle of attack and the Speed-Brake controls speed, but of course there is a significant amount of coupling between the two loops. In real life the vehicle parameters, such as dynamic pressure, aerodynamic coefficients, etc, vary during flight, but in this example we use the invariant Flixan derived linear model during the 100 seconds flight. The purpose of this exercise is to demonstrate the auto-landing modeling and design methodology. For better accuracy we should be using a 6-DOF simulation with time varying parameters. The inputs to the Simulink model are: altitude versus time command, and a noisy wind gust of approximately 20 (ft/sec) amplitude and 0.2 Hz bandwidth.

Figure 5.1 shows the altitude starting at 9000 (feet) and ending at zero at landing. The velocity starts at 550 (ft/sec), that is, 500 (ft/sec) nominal plus 50 (ft/sec) initial ( $\delta V$ ). During landing, 100 sec later, the velocity drops to 455 (ft/sec). The pitch attitude ( $\theta$ ) initially drops to -10 deg but it maintains a slightly positive attitude during landing. The angle of attack ( $\alpha$ ) reaches 20 (deg) at around 42 seconds and then it drops to less than 4 (degrees) during landing.

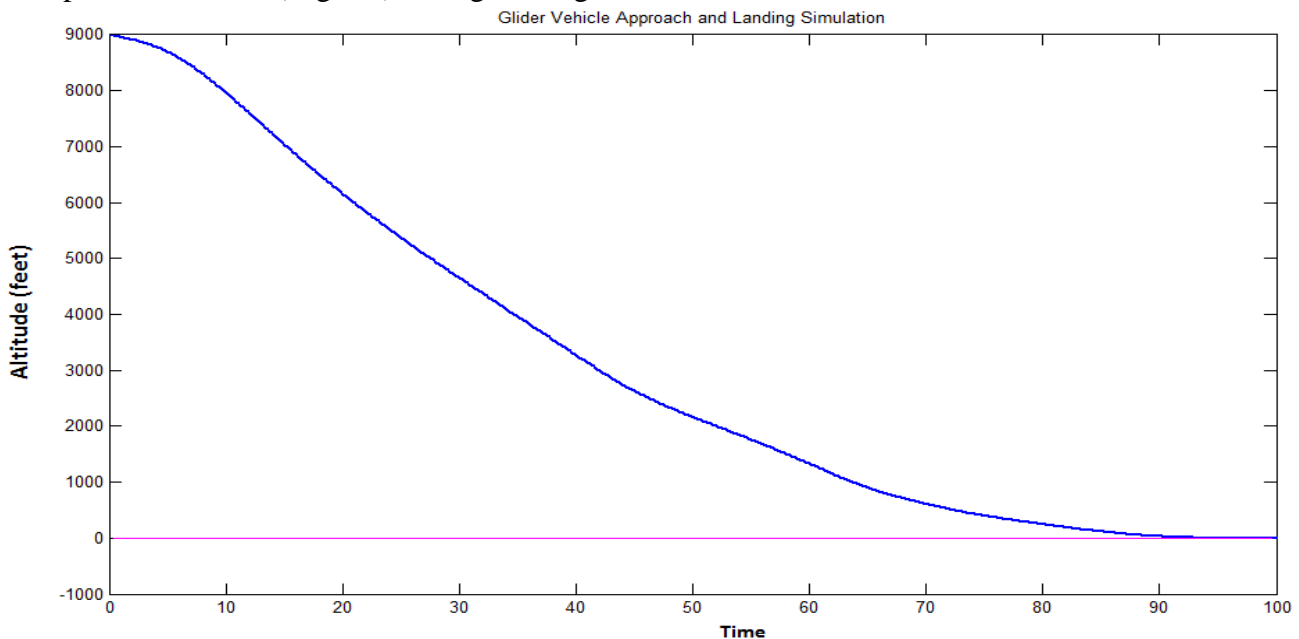


Figure 5.1 Altitude in (feet) versus Time in (sec)

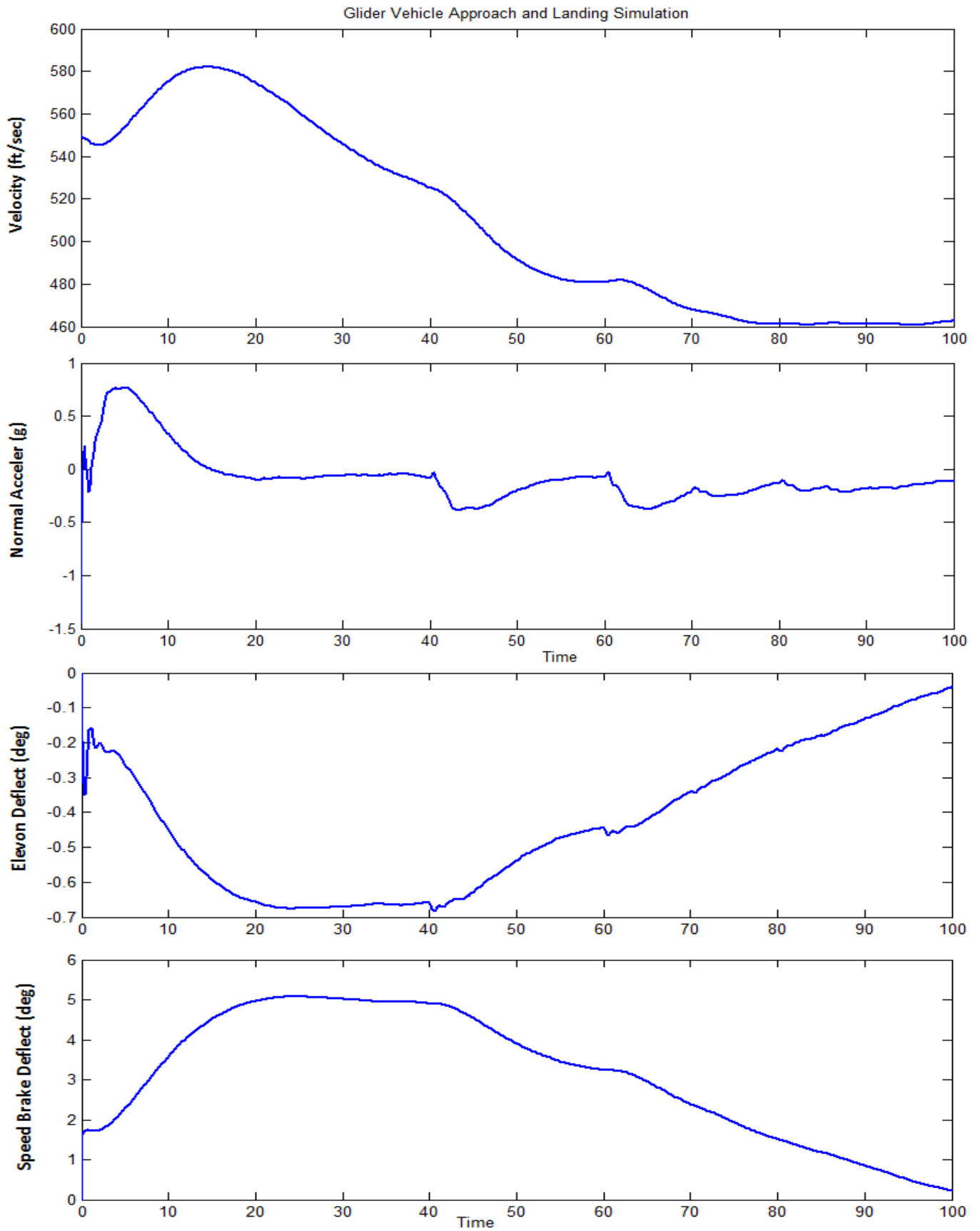


Figure 5.2 Vehicle Speed (feet/sec), Normal Acceleration, Elevon and Speed-Brake Deflections in (deg)

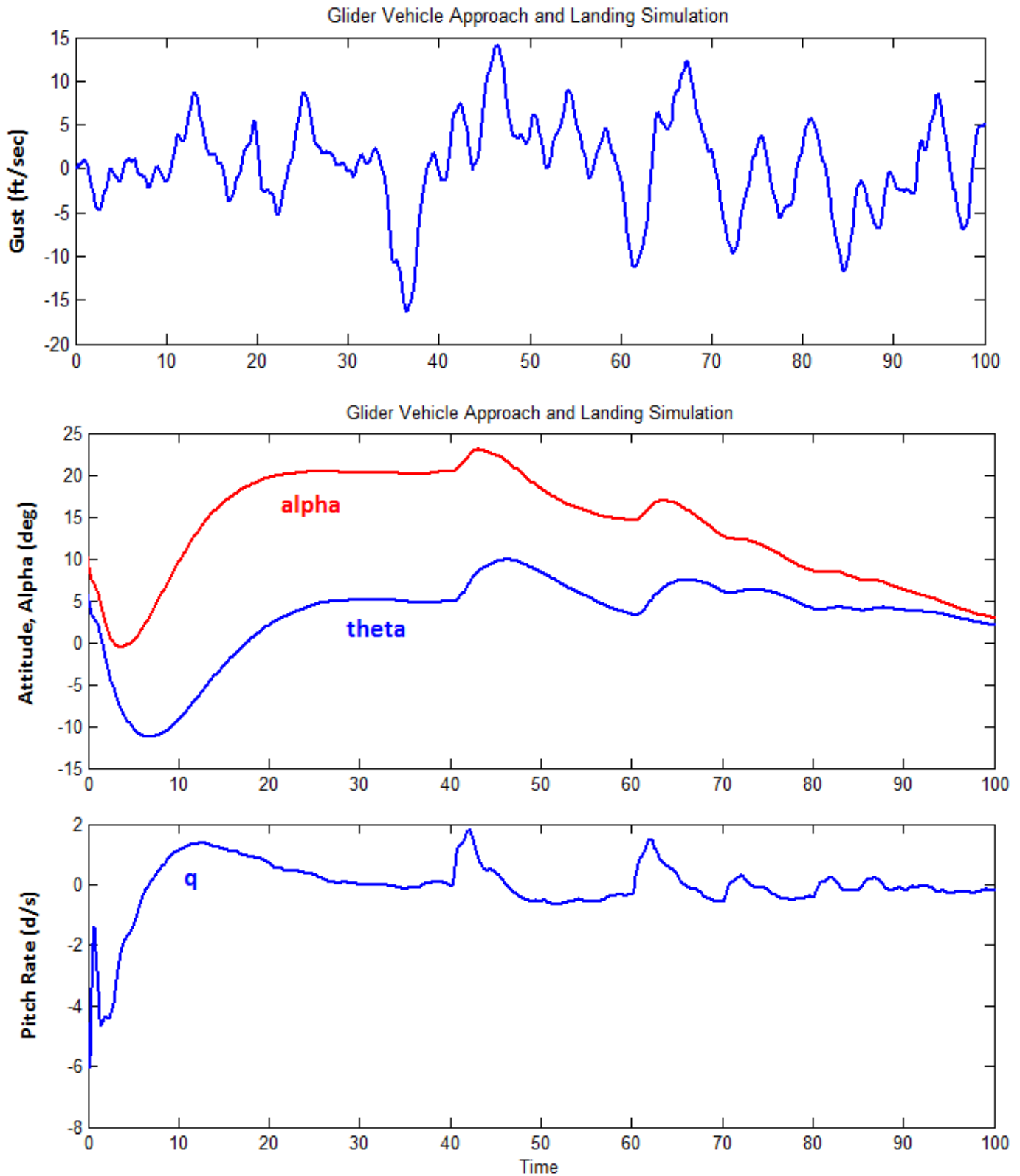


Figure 5.3 Wind Gust velocity (ft/sec), alpha, theta in (deg), and pitch rate (d/s)



#### 4.4 Open-Loop Stability Analysis in Pitch

The Simulink model “*OpenLoop.Mdl*” in Figure 6 is used for the longitudinal axes open-loop frequency domain analysis. The control loop is opened at the Elevon input, and the Speed-Brake loop is closed. The guidance and vehicle subsystems are the same as those used in the simulation model in section 4.2. However, a state-space system for open-loop analysis was also created by the Flixan program which combined the vehicle, actuator, and guidance subsystems as already described. It was saved in file “*Autoland.Qdr*” and its title is: “*Elevon Open-Loop System (Speed-Brake Closed)*”. It was also exported for Matlab analysis in file “*siso\_anal.m*”.

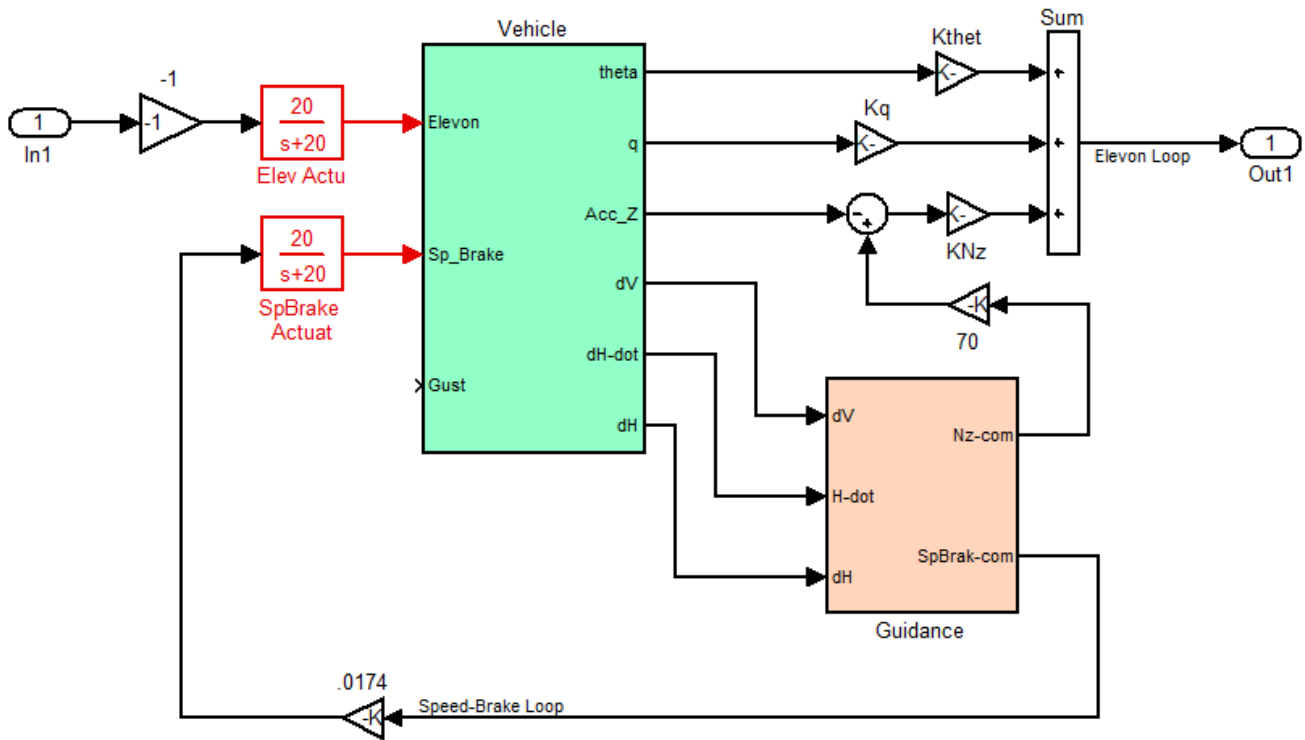
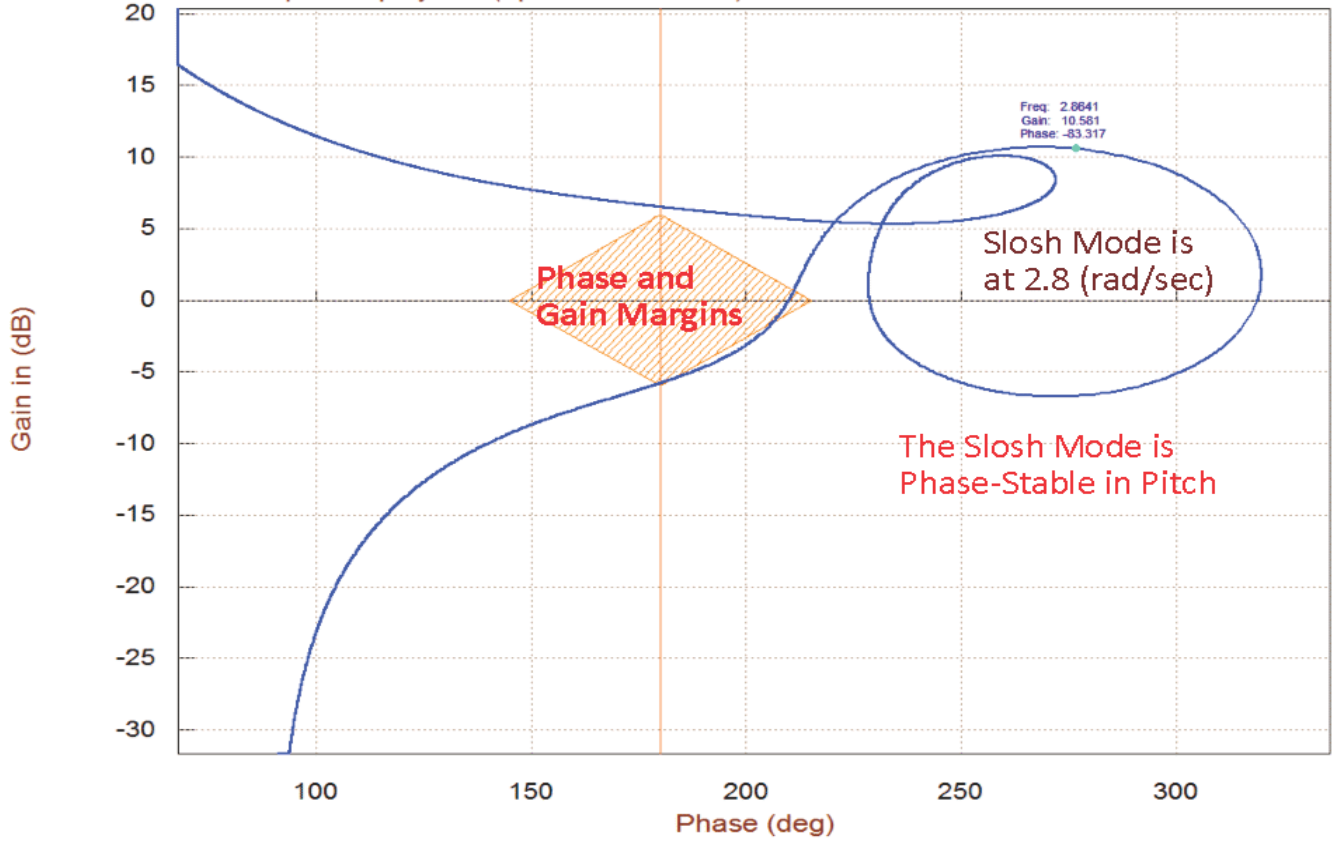


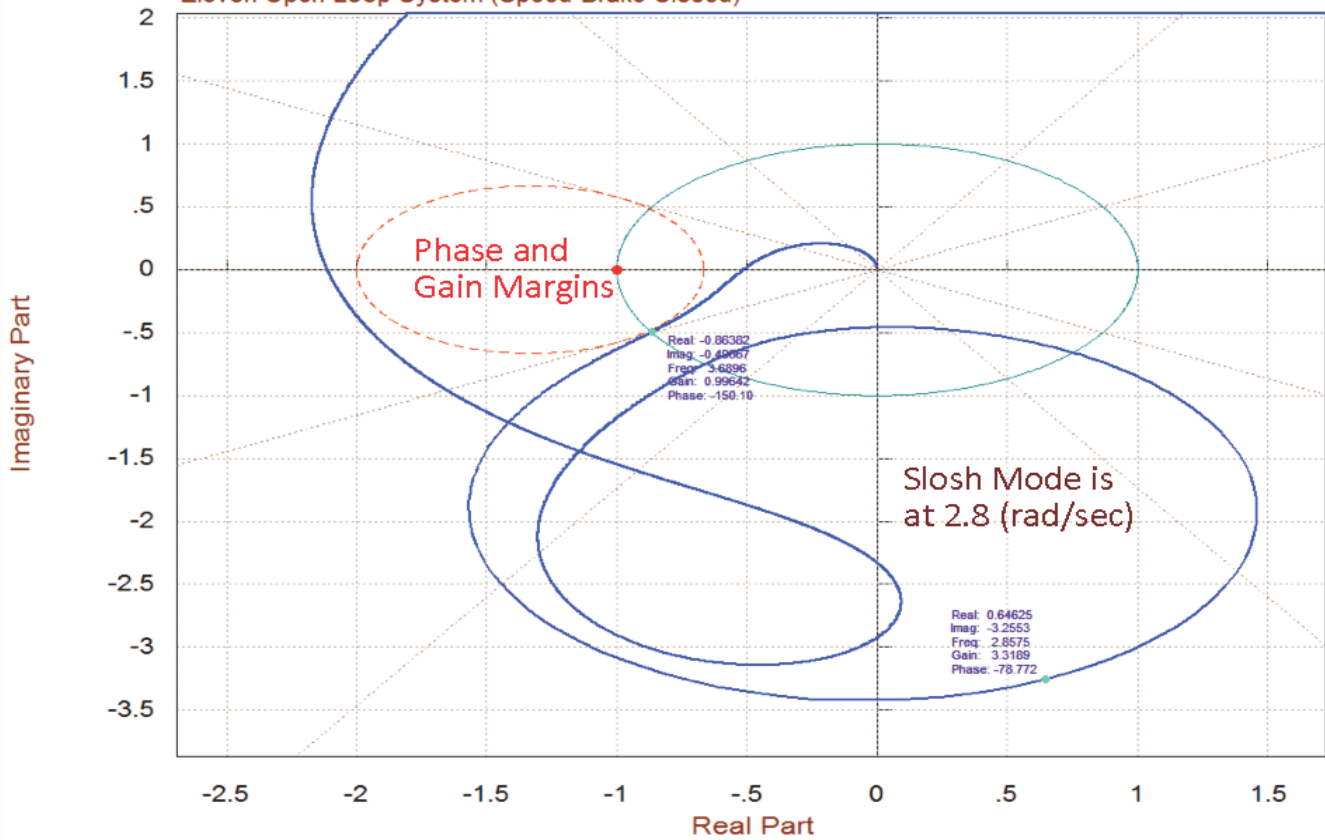
Figure 6 Open-Loop Interconnection Diagram for Elevon Stability Analysis

A script file “*run.m*” in subdirectory “*\Mat\_Pitch*” is used to load the state-space systems, initialize the Simulink models, and perform frequency domain analysis. The open-loop system is loaded into Matlab workspace from file “*siso\_anal.m*” which calculates and plots the frequency response. The same frequency response analysis is also performed by the Flixan frequency response utility program, shown by the Nichols and Nyquist plots in Figure 7. They demonstrate that the Elevon control loop has 6 dB of Gain Margin and a Phase Margin of 30 degrees.

Nichols Plot for: Outp( 1)-Flight Control to Actuator / Inpt( 1)-Elevon Actuator , of:  
Elevon Open-Loop System (Speed-Brake Closed)



Nyquist Plot for: Outp( 1)-Flight Control to Actuator / Inpt( 1)-Elevon Actuator , of:  
Elevon Open-Loop System (Speed-Brake Closed)



## 5. Lateral Axes Analysis

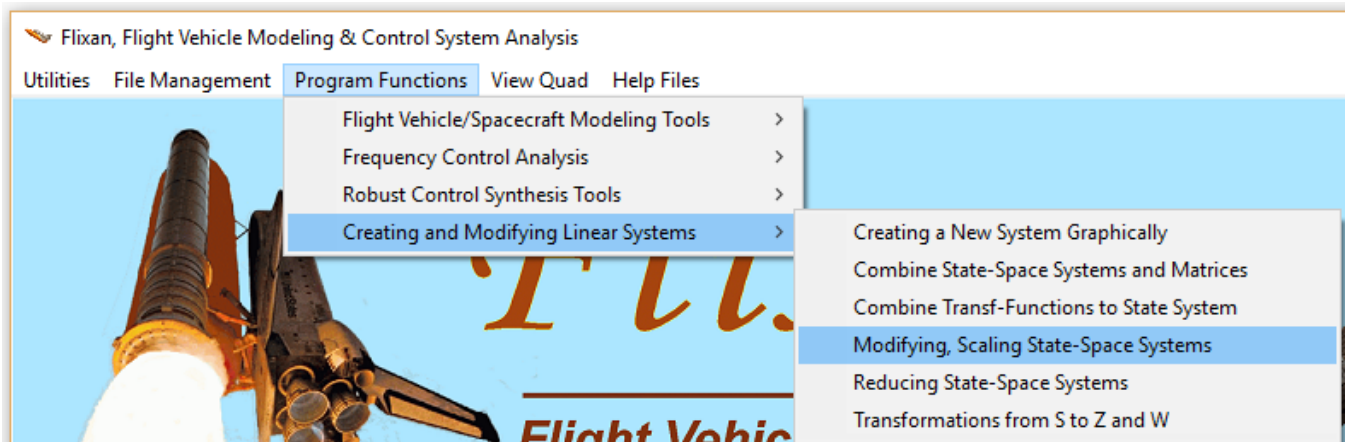
The lateral control system uses the Aileron and the Rudder control surfaces for roll and yaw stabilization. It also uses a state-feedback LQR controller from the five vehicle states ( $\phi$ ,  $p$ ,  $\psi$ ,  $r$ , and  $\beta$ ) to the aileron and rudder control inputs. The control system assumes that the sideslip angle  $\beta$  is directly measurable. The lateral guidance system applies roll attitude commands ( $\phi$ -cmd) to control the flight direction.

### 5.1 Using the Flixan Program to Create the Lateral Models

In the lateral axes we are using two roll/yaw state-space models which are extracted from the original vehicle model “Automatic Landing of Unpowered Vehicle”. A design model “Automatic Landing of Unpowered Vehicle, Lateral Design Model” that is used for LQR state-feedback design, and a simulation model “Automatic Landing of Unpowered Vehicle, Lateral New Outputs with Slosh”.

#### Extracting the Lateral Vehicle Model

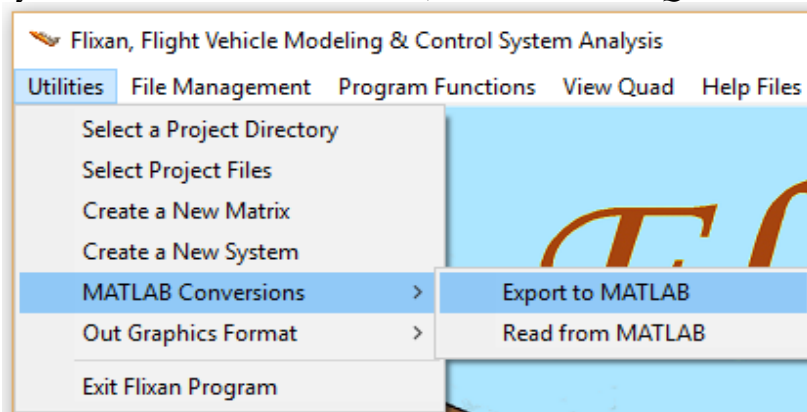
To extract the lateral vehicle subsystem used in the Matlab simulations from the fully coupled vehicle system, it requires two steps using system modification data-sets and the creation of an intermediary system. We first use the system modification set “Automatic Landing of Unpowered Vehicle, Lateral Axes” to extract the lateral subsystem from the original fully coupled vehicle, and in the second step we use the system modification set “Automatic Landing of Unpowered Vehicle, Lateral New Outputs with Slosh” to introduce some additional outputs in the previous system for monitoring the slosh mass activity inside the tanks. Running the system modification utility was already described when extracting the pitch subsystem. Select the project directory “\Flixan\ Examples\ Autoland” and from the main menu select “Program Functions”, “Creating and Modifying Linear Systems”, and then “Modifying/ Scaling Systems”. Select also the input and system filenames: “Autoland.Inp” and “Autoland.Qdr”. From the menu that shows the titles of systems modification sets select the title: “Automatic Landing of Unpowered Vehicle, Lateral Axes”, and click on “Run Input Set”. This set provides instructions on how to extract the lateral subsystem from the original vehicle system and it will save it in file “Autoland.Qdr” under the same title.



Repeat the process for the second time to create the lateral simulation model. From the systems modification titles menu select the title: “Automatic Landing of Unpowered Vehicle, Lateral New Outputs with Slosh”, and click on “Run Input Set”. This set of system reconstruction instructions will add some additional slosh mass displacement outputs to the previously generated system which are extracted from the system state-vector. The modified system will be saved in file “Autoland.Qdr” under the title “Automatic Landing of Unpowered Vehicle, Lateral New Outputs with Slosh” and it will also be converted to Matlab format. A similar process is used to extract the lateral design model.

### Exporting the Lateral Vehicle Model to Matlab

The lateral simulation system “Automatic Landing of Unpowered Vehicle, Lateral New Outputs with Slosh” which is saved in systems file “Autoland.qdr” must also be converted to a state-space format that can be loaded into the Matlab folder “\Autoland\Mat Lateral” for analysis. From Flixan main menu select “Utilities”, “Matlab Conversions”, and then “Export to Matlab”, as shown below. From the systems filename selection menu, select “Autoland.Qdr” and click “OK”.

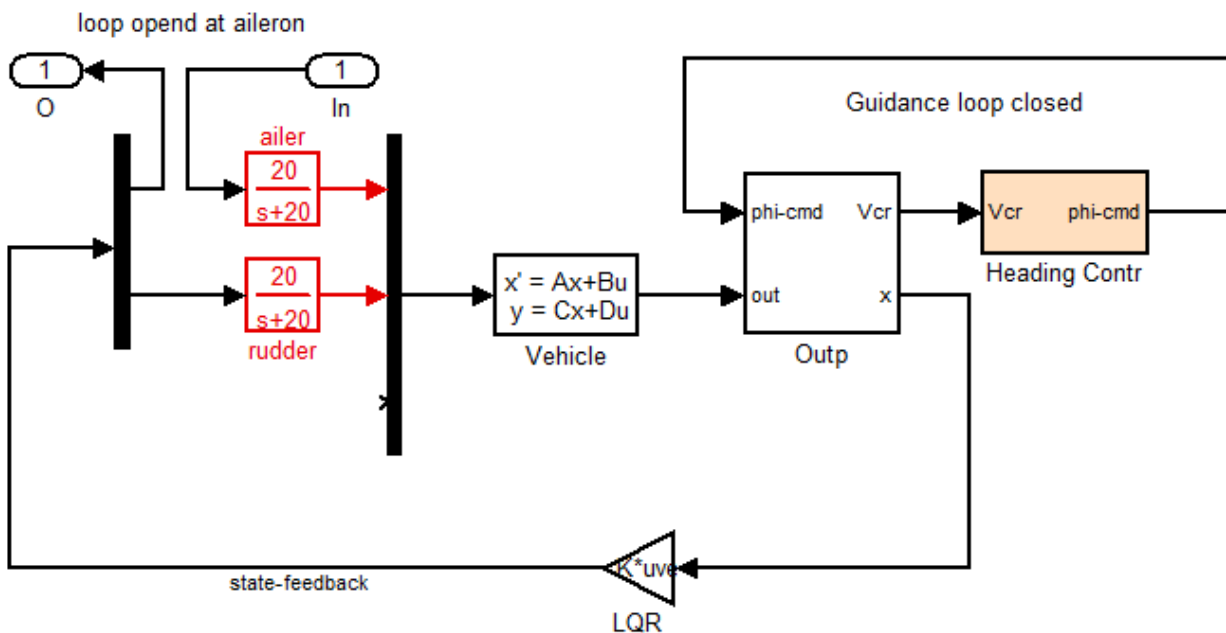


Select the directory where the Matlab analysis will be performed “\Examples\Autoland\Mat Lateral”. From the “Export to Matlab” dialog select “System (A, B, C, D)” and “Function m-file”. From the menu that contains the titles of the systems in file “Autoland.Qdr” select the title of the system to be converted “Automatic Landing of Unpowered Vehicle, Lateral New Outputs with Slosh” and click “OK”. Enter also the m-filename “vehi\_latan2.m” (without the .m) where the state-space matrices will be saved as a Matlab function. Repeat the conversion also for the design system “Automatic Landing of Unpowered Vehicle, Lateral Design Model”. This system is saved in file “vehi\_ldes.m” in the same subdirectory “\Autoland\Mat Lateral”.

## 5.2 Lateral Stability Analysis

The model in Figure 8 is used to perform open-loop frequency response stability analysis. This model is in Simulink file “*Lat\_Open.mdl*” in subdirectory “*Examples\Autoland\Mat Lateral*”. The Matlab m-file “*run\_lat.m*” is used to load the lateral design model from file “*vehi\_ides.m*” and the analysis model from file “*vehi\_latan2.m*” to be used in simulation and frequency response analysis. The m-file also uses the design model to synthesize an LQR state-feedback gain matrix  $K_g$ , and the Simulink model “*Lat\_Open.Mdl*” (shown in Figure 8 with the aileron loop opened and the rudder loop closed) to create an open-loop system and calculate its Bode and Nichols plots for analyzing the aileron stability margins.

The Simulink model can be modified by closing the aileron and opening the rudder loop to analyze the rudder stability. The Nichols plots in Figures 9.1 & 9.2 demonstrate an acceptable stability margin in the aileron loop, and a somewhat degraded stability in the rudder loop.



**Figure 8 Simulink Model for Open-Loop Analysis. The loop is opened at the aileron input.**

The analysis shows that both sash resonances are present in the lateral directions. They are at different frequencies, as described, because the tank is not symmetric but elongated in the x-direction producing a longitudinal frequency of 2.1 (rad/sec) at 1g that becomes 2.63 (rad/sec) at 1.5g. The side frequency in the y-direction is 3.6 (rad/sec) at 1g and it becomes 4.87 (rad/sec) at 1.5g. The results show that in the aileron loop both sash resonances are present and they appear to be stable. In the yaw (rudder) loop, however, the longitudinal sash resonance is pointing towards instability, which is not satisfactory. The design modification is left for the reader, but one way of fixing this problem is to reduce the rudder gain by increasing the yaw penalty in the R matrix in the LQR design. This is acceptable since we are not directly controlling yaw but only roll. The flight direction is controlled by applying roll commands from guidance.

### Aileron Opened Rudder Closed Stability Analysis

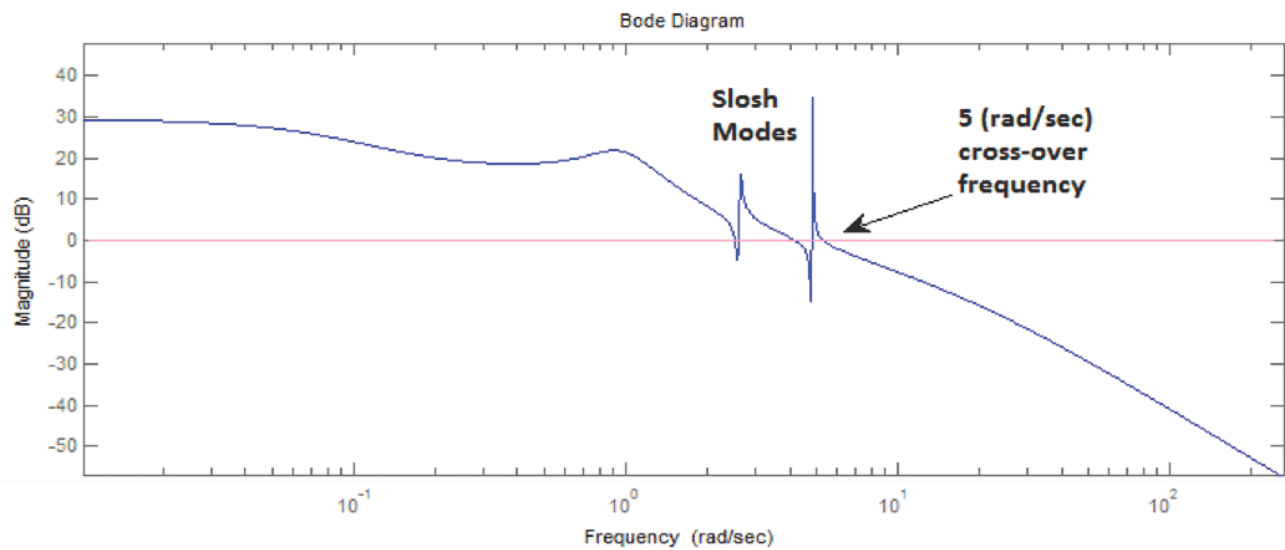
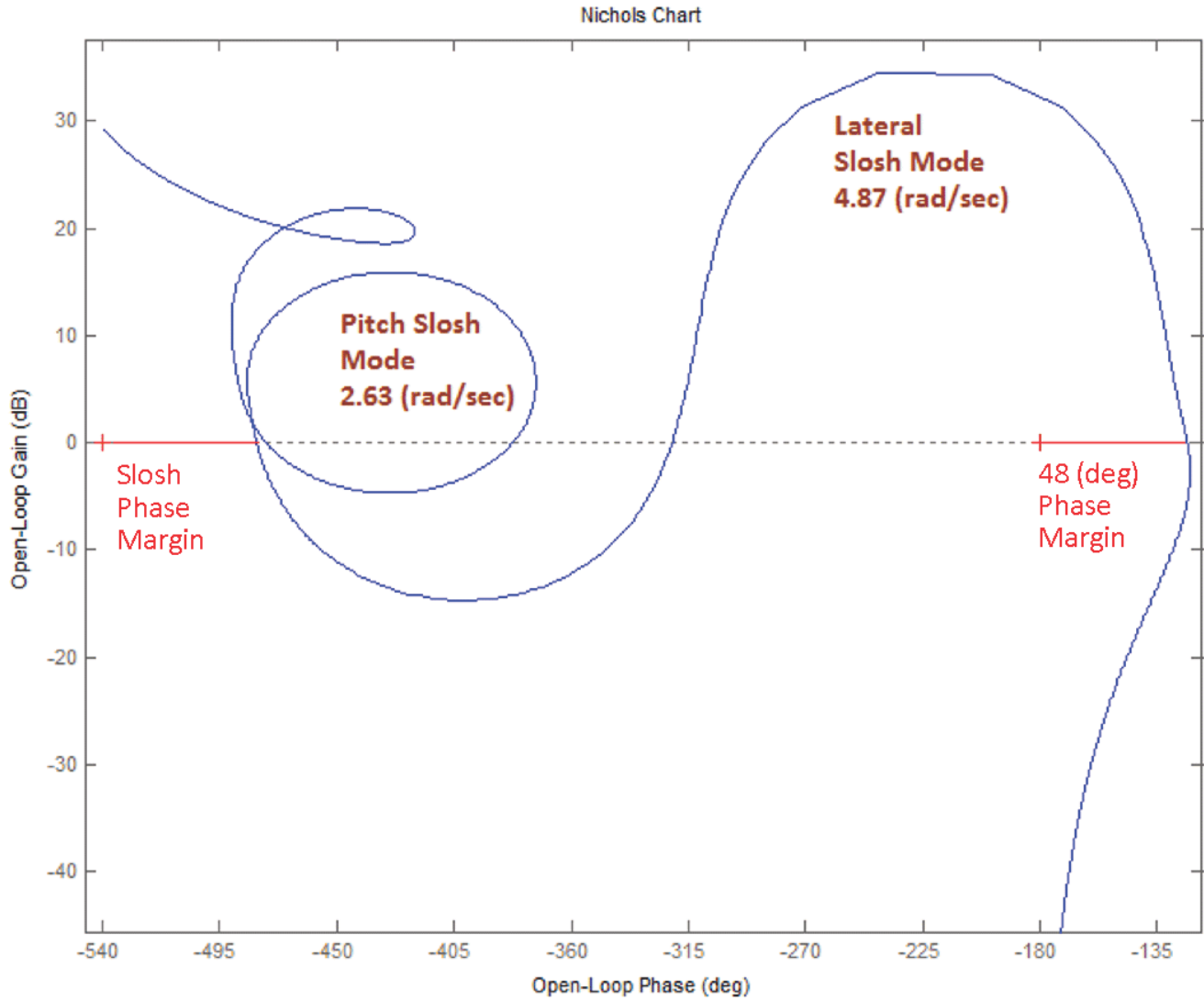


Figure 9.1 Lateral Stability Analysis diagram with the loop opened at the Aileron shows Stable Slos Resonances at 2.63 and 4.87 (rad/sec)

## Rudder Loop Opened, Aileron Loop Closed Stability Analysis

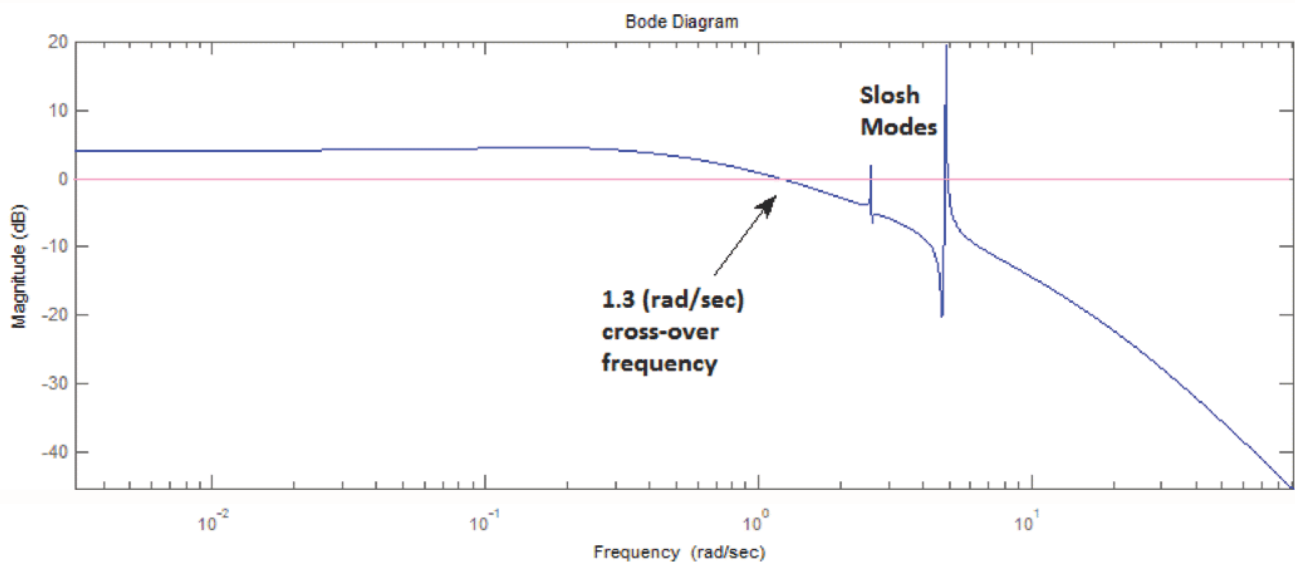
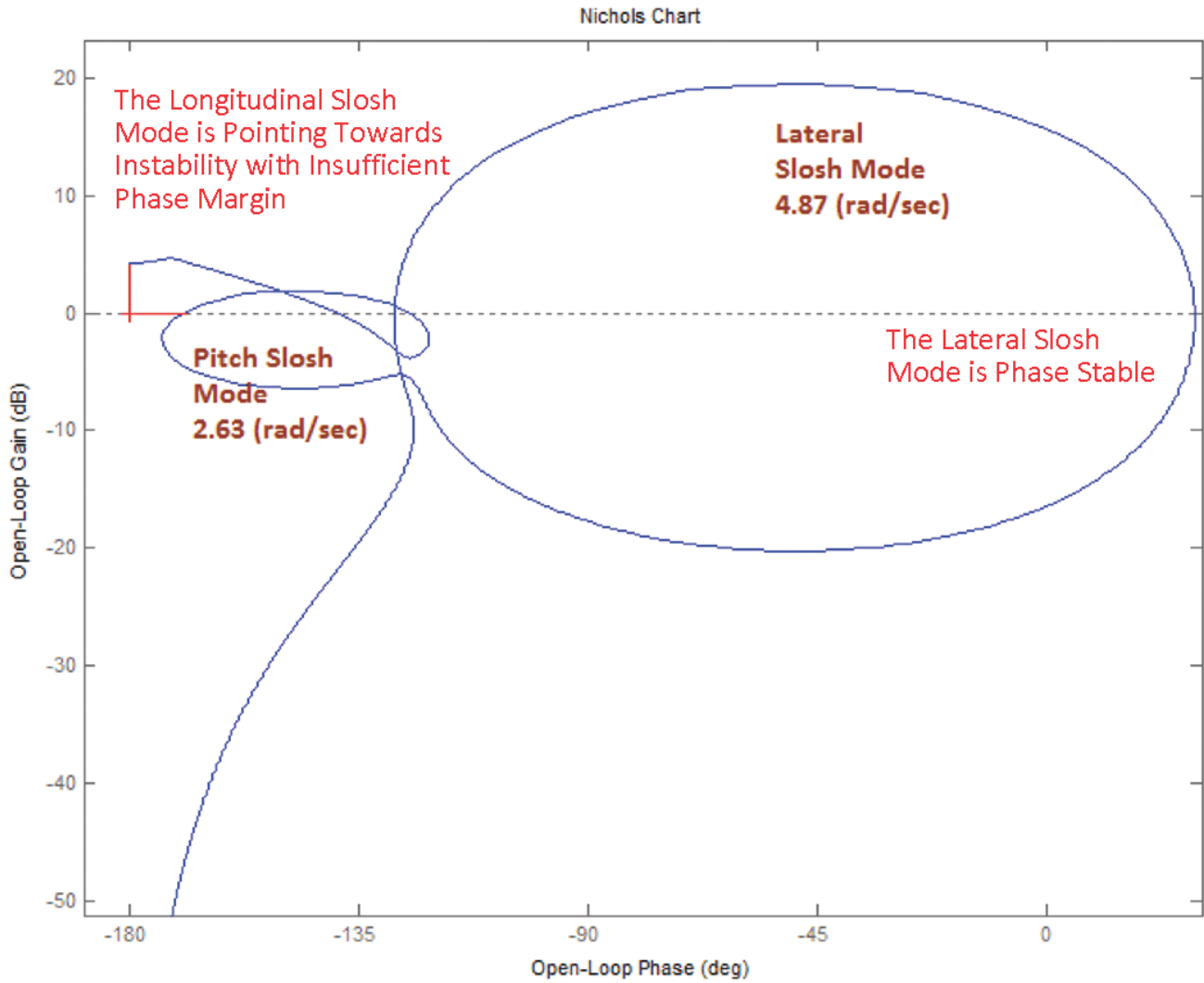


Figure 9.2 Lateral Stability Analysis with the loop opened at the Rudder shows two Slos Resonances at 2.63 and 4.87 (rad/sec). The longitudinal resonance which is at lower frequency is pointing towards the critical point.

### 5.3 Lateral Simulations

The Simulink model used in lateral simulations is in file “*Lat\_Sim.Mdl*” in subdirectory “\Autoland\Mat Lateral”. The Matlab script file “*run\_lat.m*” loads the vehicle simulation model “*Vehi\_Latan2.m*” into Matlab, calculates the LQR state-feedback gains, and also initializes the state-vector in the simulation.

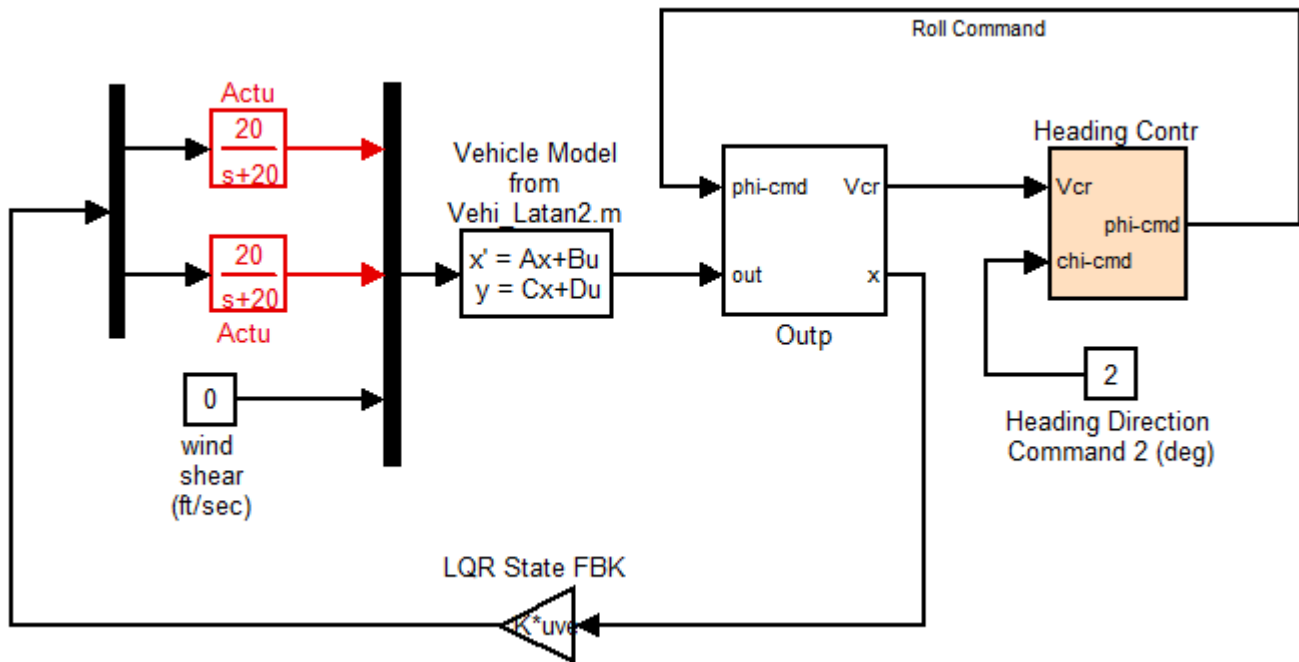


Figure 10.1 The Simulation Inputs are: (a) Heading direction command, and (b) Wind-Gust Velocity (ft/sec). The wind direction is defined in the vehicle data.

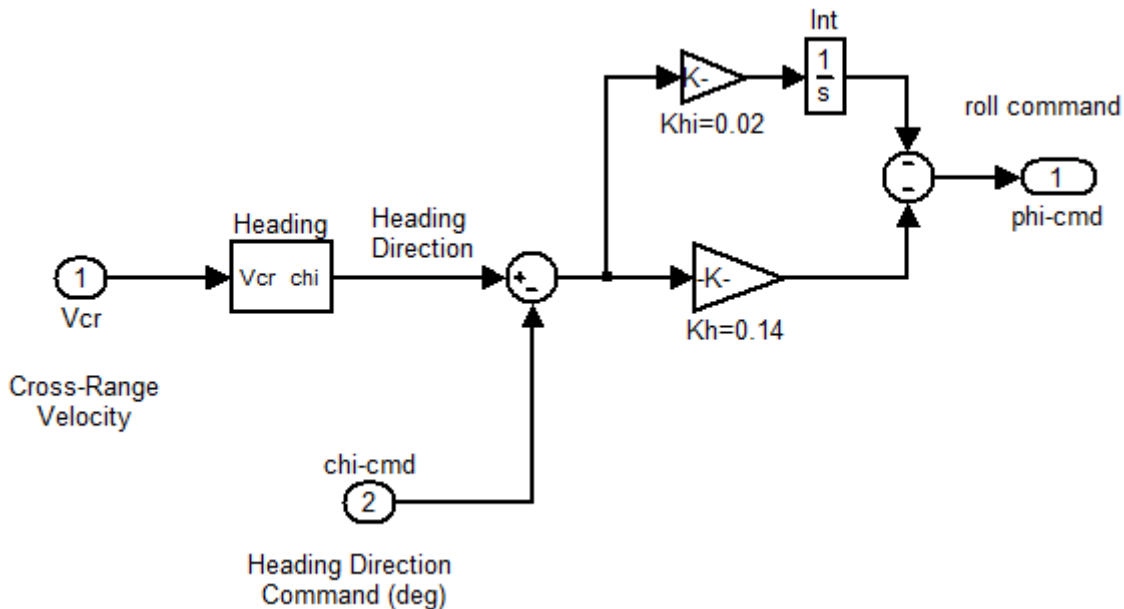


Figure 10.2 Heading Direction Guidance System; The Cross-Range Velocity is converted to Heading Direction Angle (deg). The error generates the roll command (phi-cmd)



Figures (11.1 to 11.4) show the lateral system response to a 2 (deg) change in yaw direction. The vehicle uses roll control to maneuver in yaw. There is no wind gust during this maneuver.

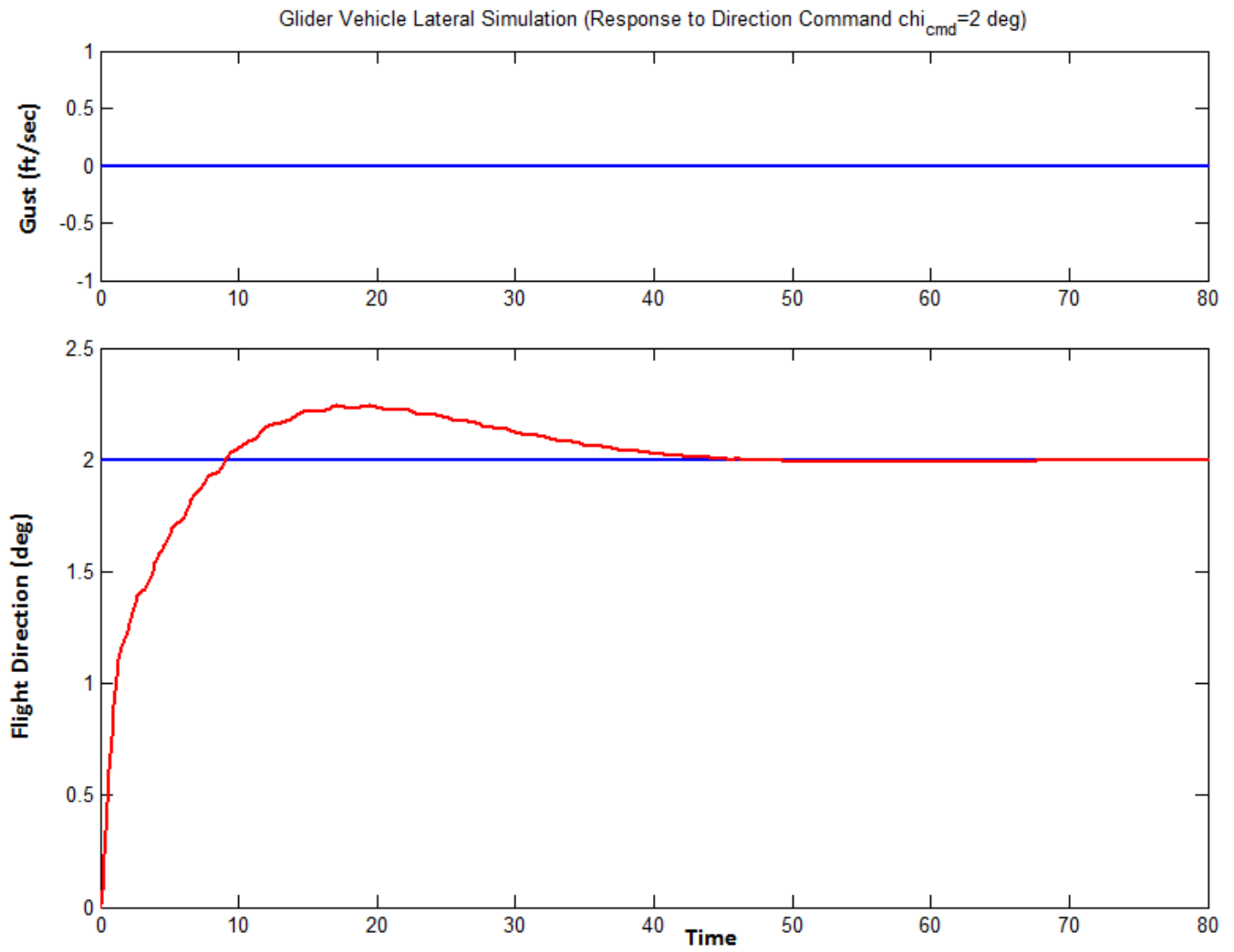


Figure 11.1 Flight Direction Response to a 2 (deg) change command in Flight Direction

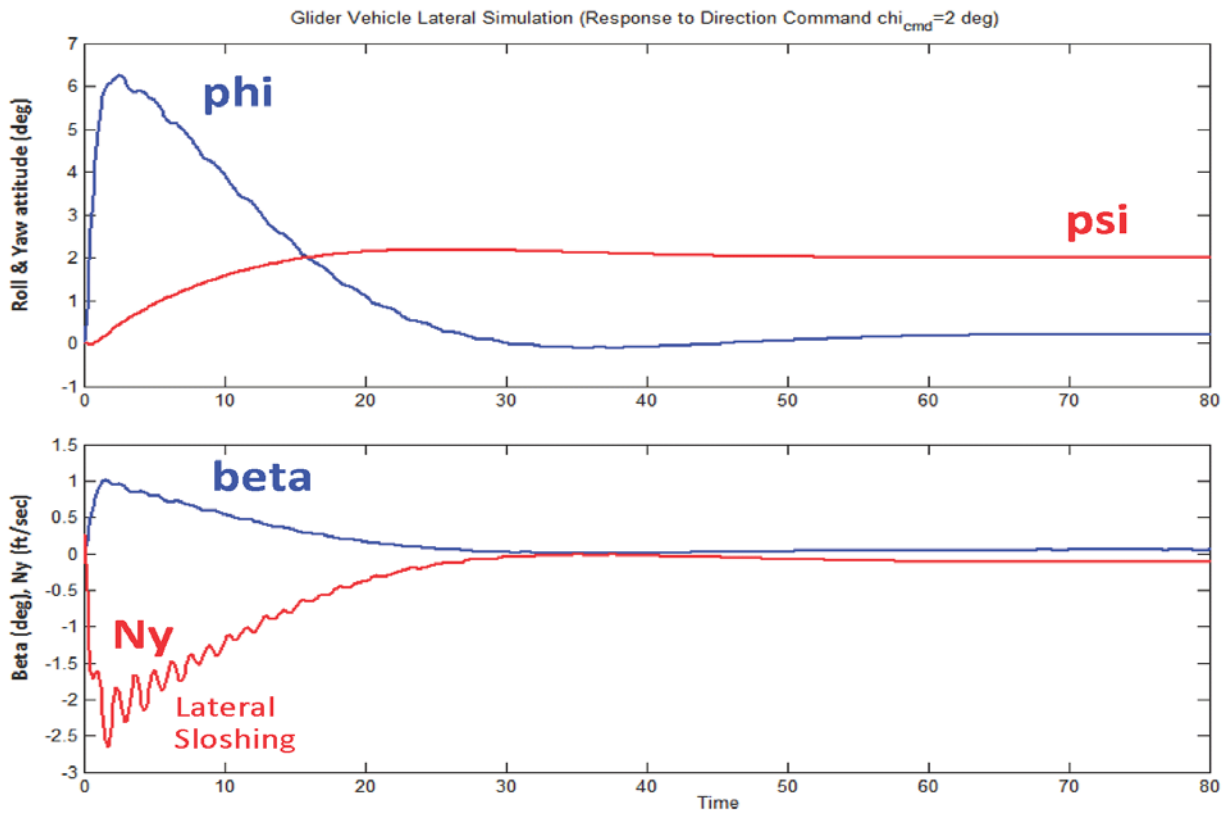


Figure 11.2 Vehicle rolls to achieve 2 (deg) Change in Yaw. The lateral accelerometer senses the fuel motion along y

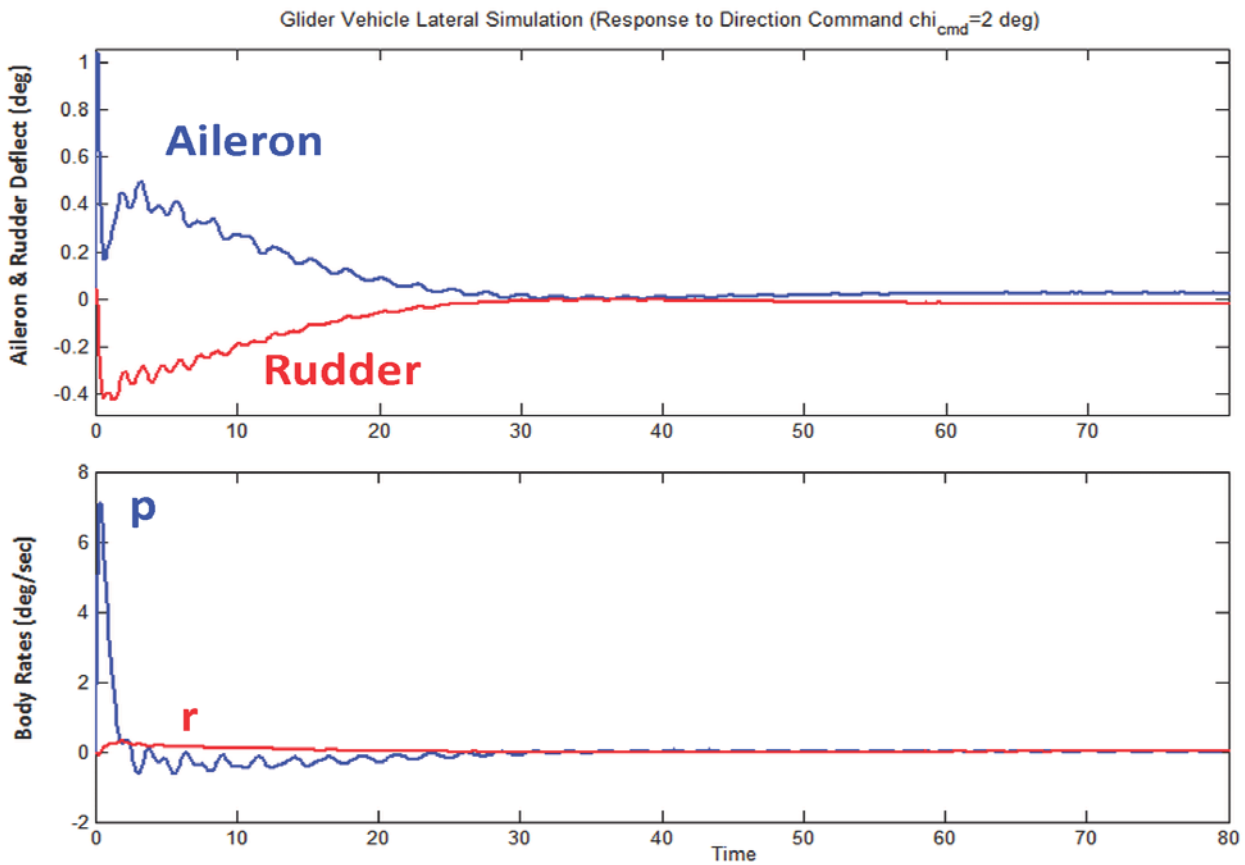
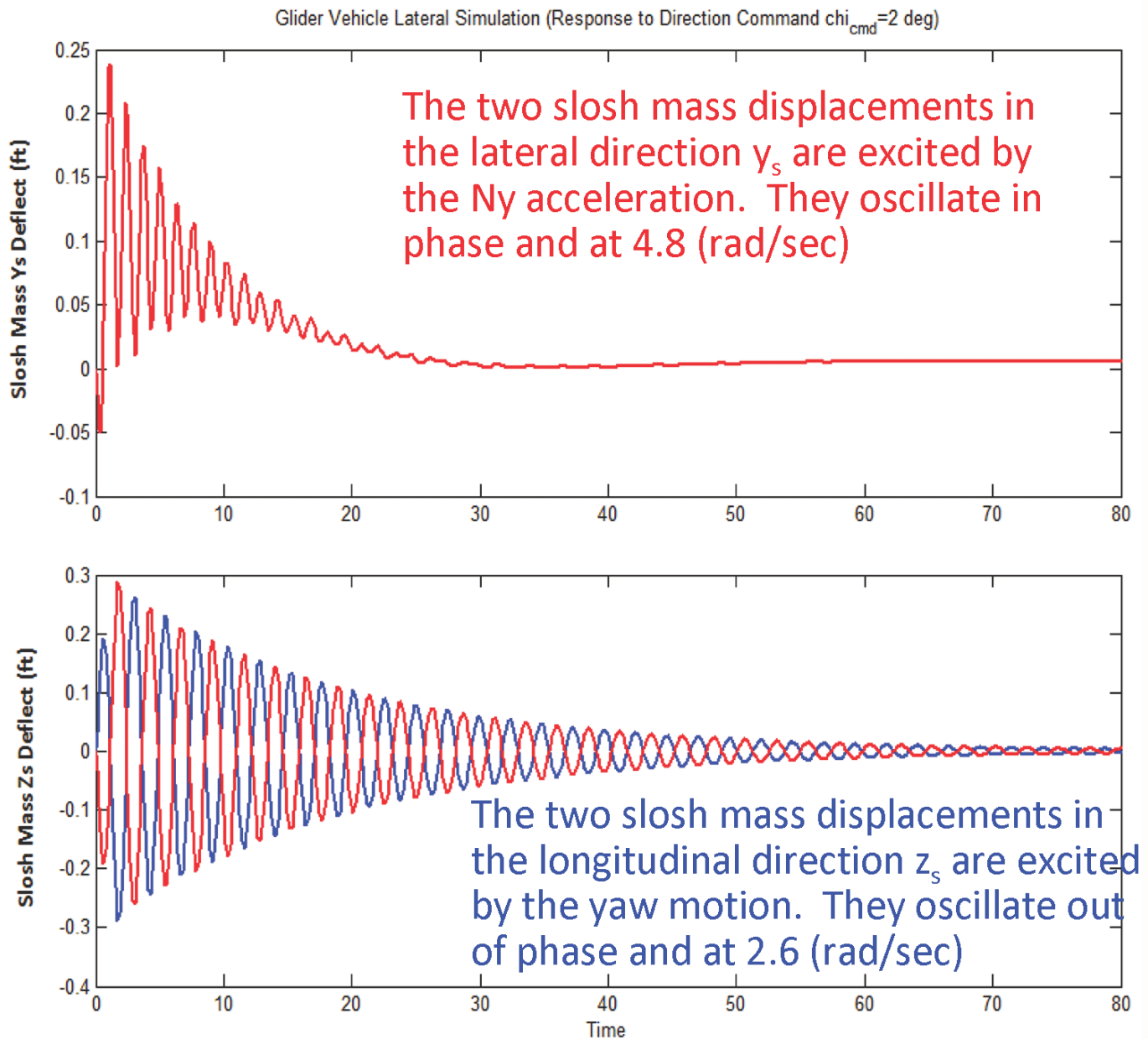


Figure 11.2 Aileron and Rudder deflections, Roll and Yaw Body Rates



**Figure 11.4** Negative  $N_y$  acceleration causes the slosh masses to deflect and oscillate in the  $+y_s$  direction relative to the tank. The yaw maneuver also causes the slosh masses to oscillate anti-symmetrically in the  $z_s$  direction. Note that the  $z_s$  slosh direction is not aligned with the vehicle  $z$  axis but it is on the  $x$ - $z$  plane, perpendicular to the acceleration vector  $A_T$  (see Figure 2), and closer to the vehicle  $x$  axis

Figures (12.1 to 12.4) show the lateral system response to a 30 (ft/sec) wind gust step coming from a direction that excites the vehicle pitch and lateral motion. The wind-gust generates a positive  $\beta$  and a negative lateral acceleration  $N_y$ . The negative acceleration causes a drift in the yaw direction. It also excites sloshing in the y-direction that is measured by the accelerometer. Almost immediately the vehicle responds to the gust by deflecting the aileron to roll towards the wind and to compensate against the yaw drift. The rudder also responds to correct the yaw error. The wind disturbance also excites sloshing motion in the longitudinal direction.

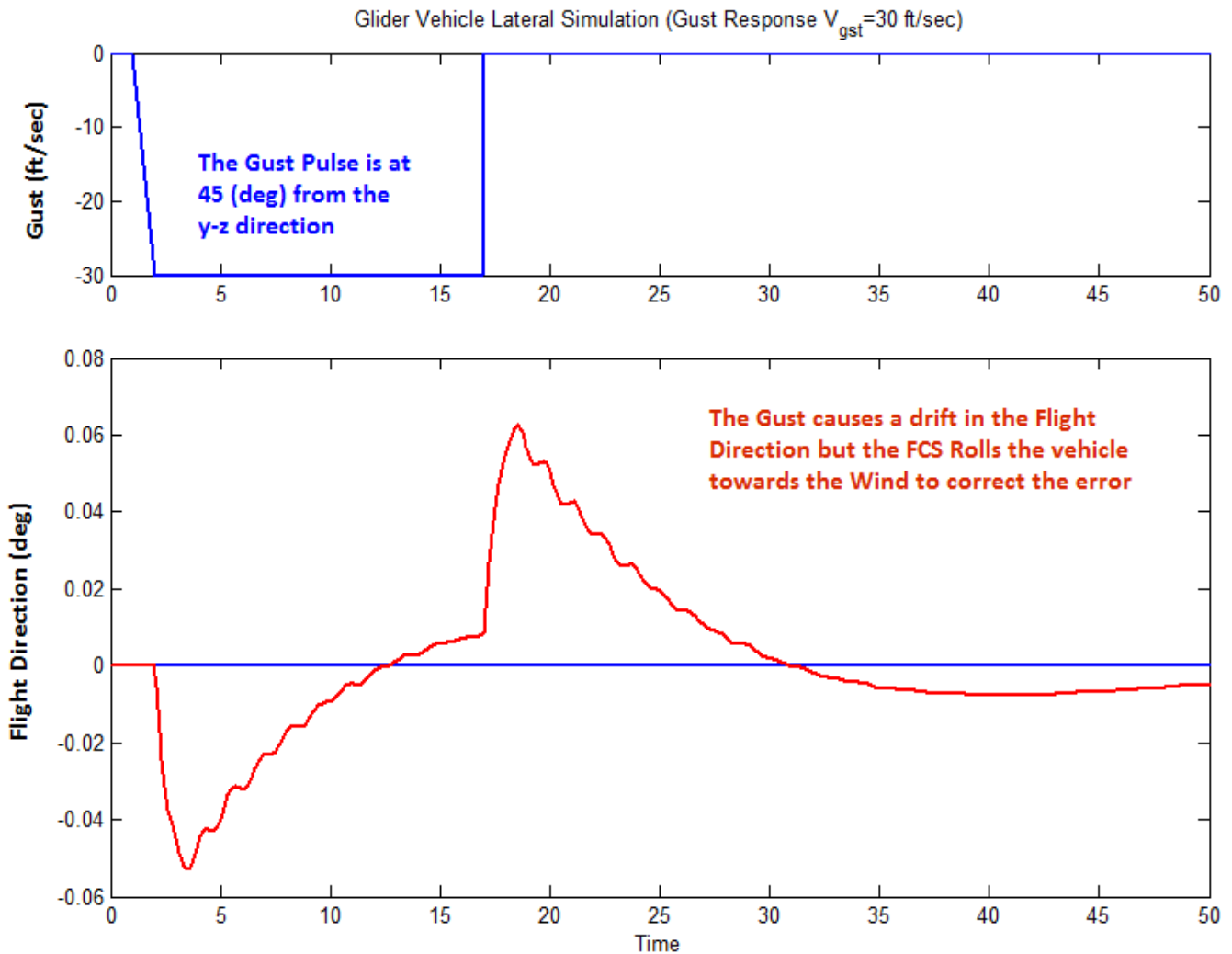


Figure 12.1 The Wind Gust causes a drift in the vehicle direction but the FCS corrects the error by rolling the vehicle towards the wind.

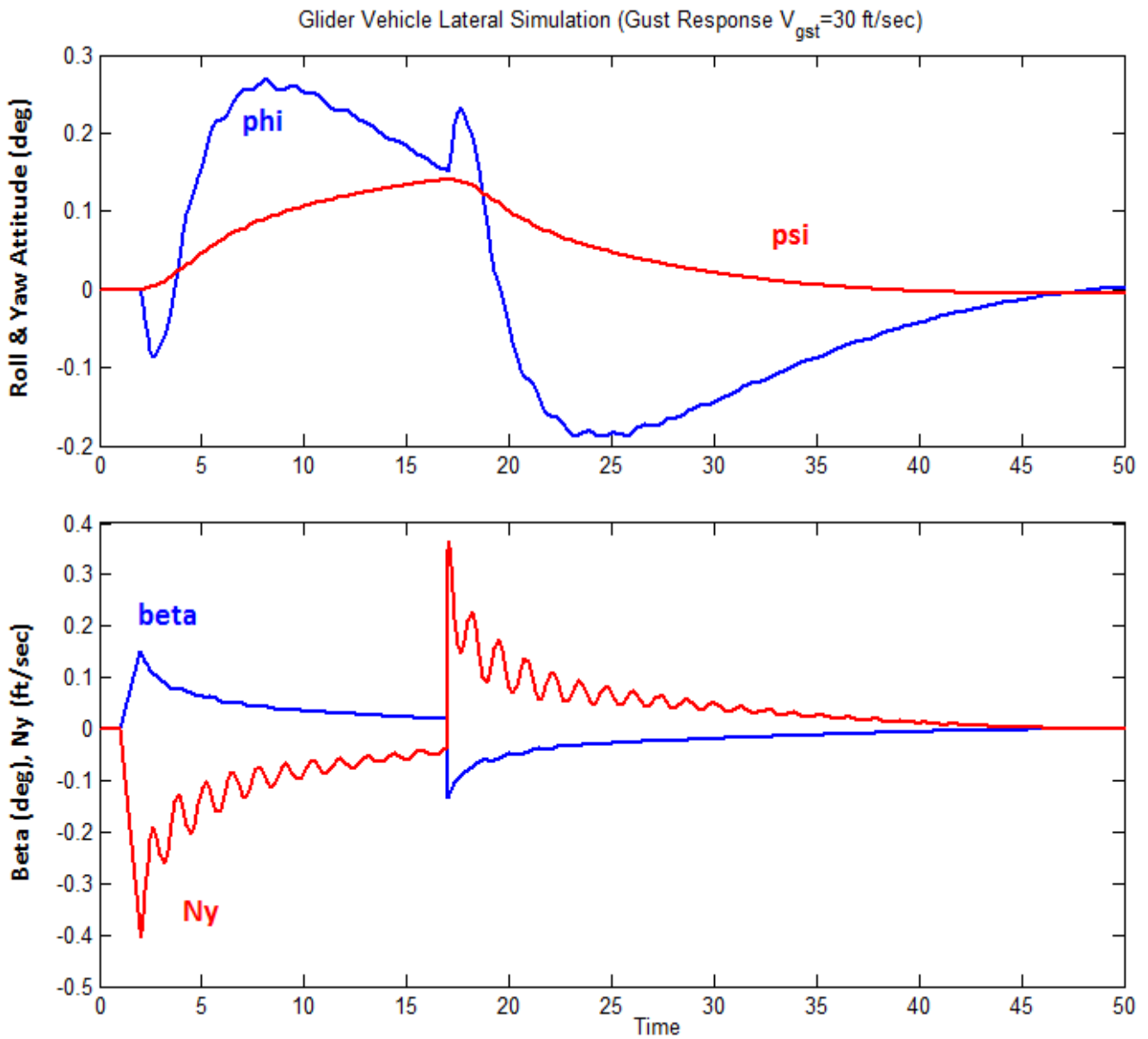
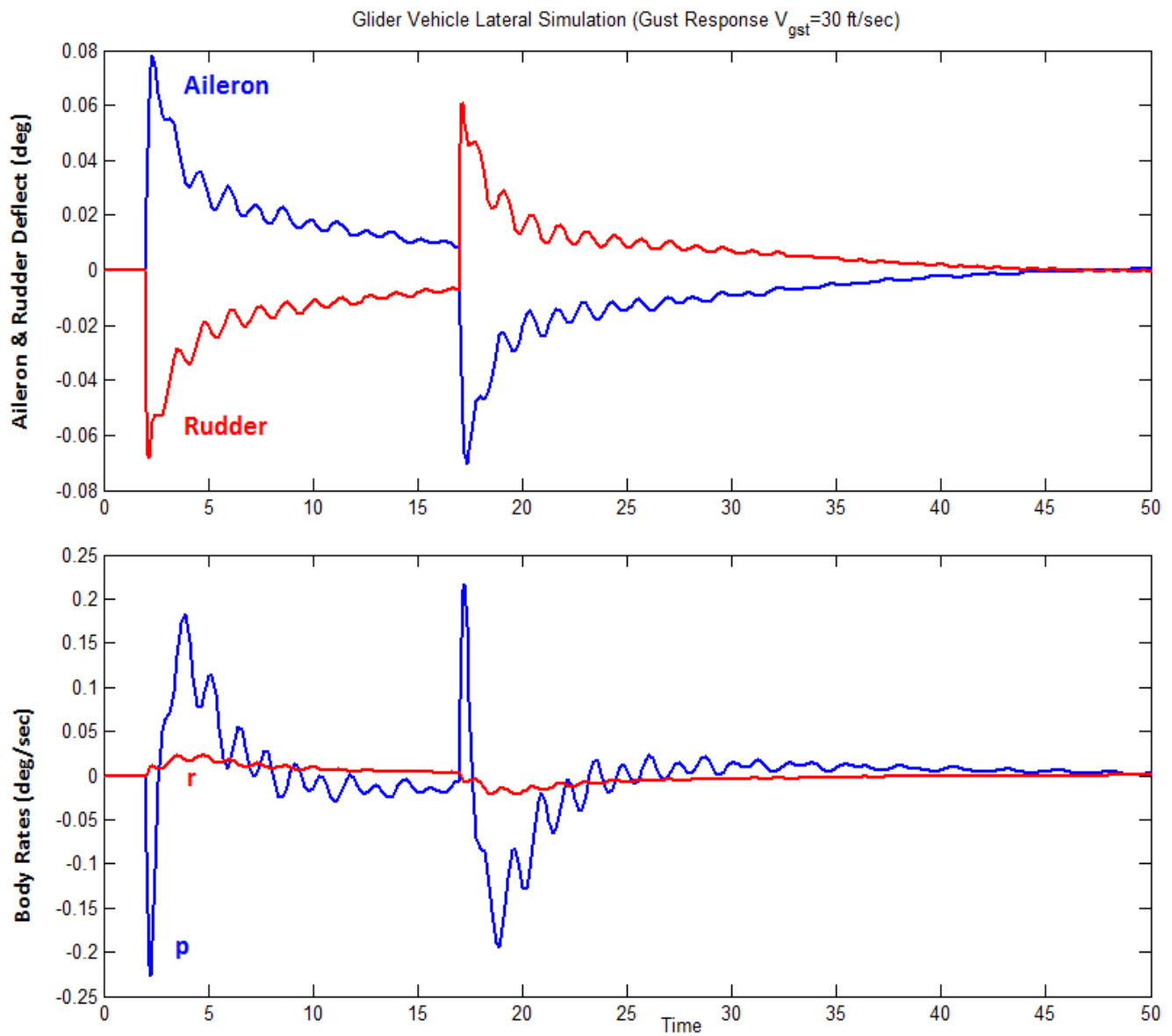
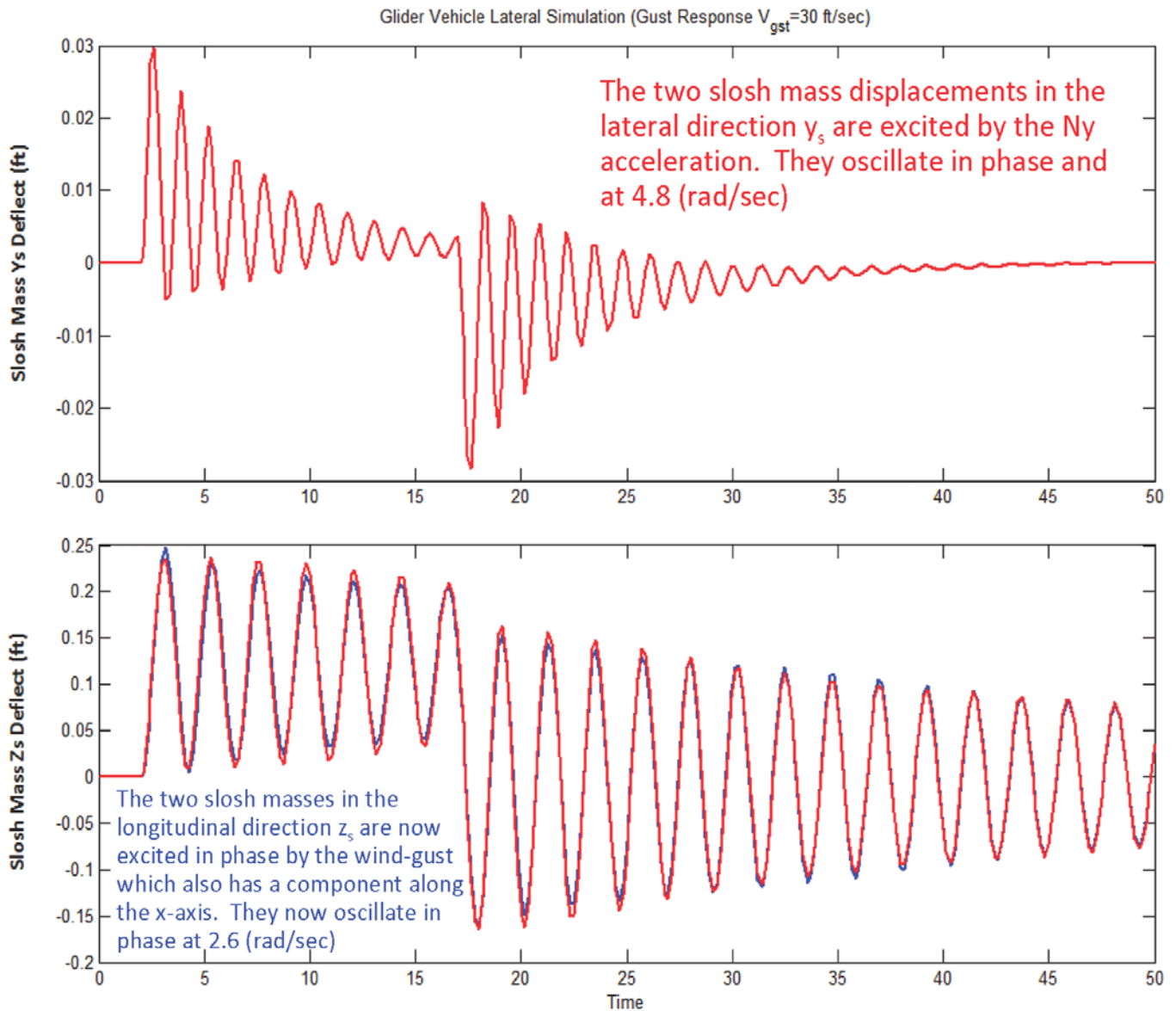


Figure 12.2 Initial Roll is Negative due to Wind hitting the Vertical Rudder; then the vehicle rolls Positive to correct the yaw drift and to counteract the Wind Disturbance. The lateral sloshing forces are detected by the Ny accelerometer.



**Figure 12.3** Aileron and Rudder deflections in (deg) correct the vehicle Roll due to gust and also the Yaw drift. Initially the Wind Gust causes a negative Roll that is corrected with a positive aileron deflection. The negative Rudder deflection compensates against the Yaw drift due to Gust.



**Figure 12.4** Negative  $N_y$  acceleration due to wind causes the slosh masses to deflect and oscillate in-phase along the  $+y_s$  direction relative to the tank at 4.8 (rad/sec). The slosh masses are also excited symmetrically along the longitudinal tank directions  $z_s$ . The reason is because the wind-gust direction has x and z components that excite the slosh modes also in the longitudinal direction. Note that the  $z_s$  slosh direction is not aligned with the vehicle z axis but it is on the x-z plane, perpendicular to the acceleration vector  $A_T$  (see Figure 2), and closer to the vehicle x axis.