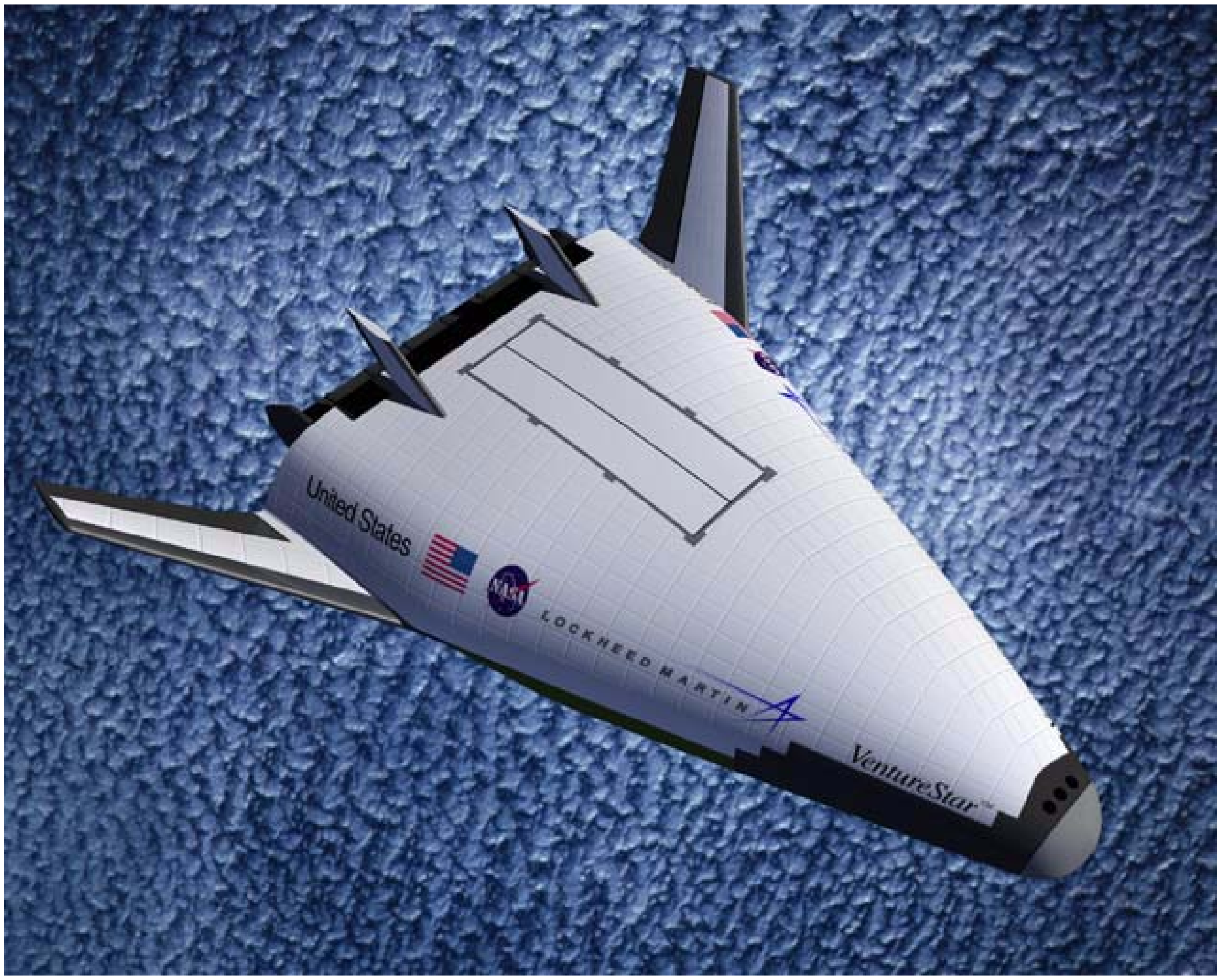
Autonomous landing of an un-powered glider vehicle with fuel sloshing tanks



In this example we shall develop dynamic models, design, analyze and simulate the landing of an un-powered Shuttle type of reentry vehicle as it descends from a high altitude and lands on a runway using an autonomous landing system. The dynamic model is further complicated by the sloshing of the residual reaction control fuel inside two tanks which has a tendency to destabilize the vehicle.

1. Introduction

In this example we are going to analyze and simulate the landing of an un-powered glider vehicle from an altitude of 9000 (feet). We will create longitudinal and lateral rigid-body linear models of the vehicle at a fixed point of the trajectory, combine it with the flight control and the guidance systems, simulate its landing on the runway at see level, and analyze the system stability in frequency domain, both, in pitch and lateral. The flight vehicle parameters for the linear model were 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 vehicle uses elevon and speed-brake for longitudinal control, and it uses aileron and rudder for lateral control. The simulation assumes that the vehicle parameters are constant. It starts when the vehicle is aligned with the runway at an altitude of 9000 ft and ends when it lands at see level, 100 seconds later.

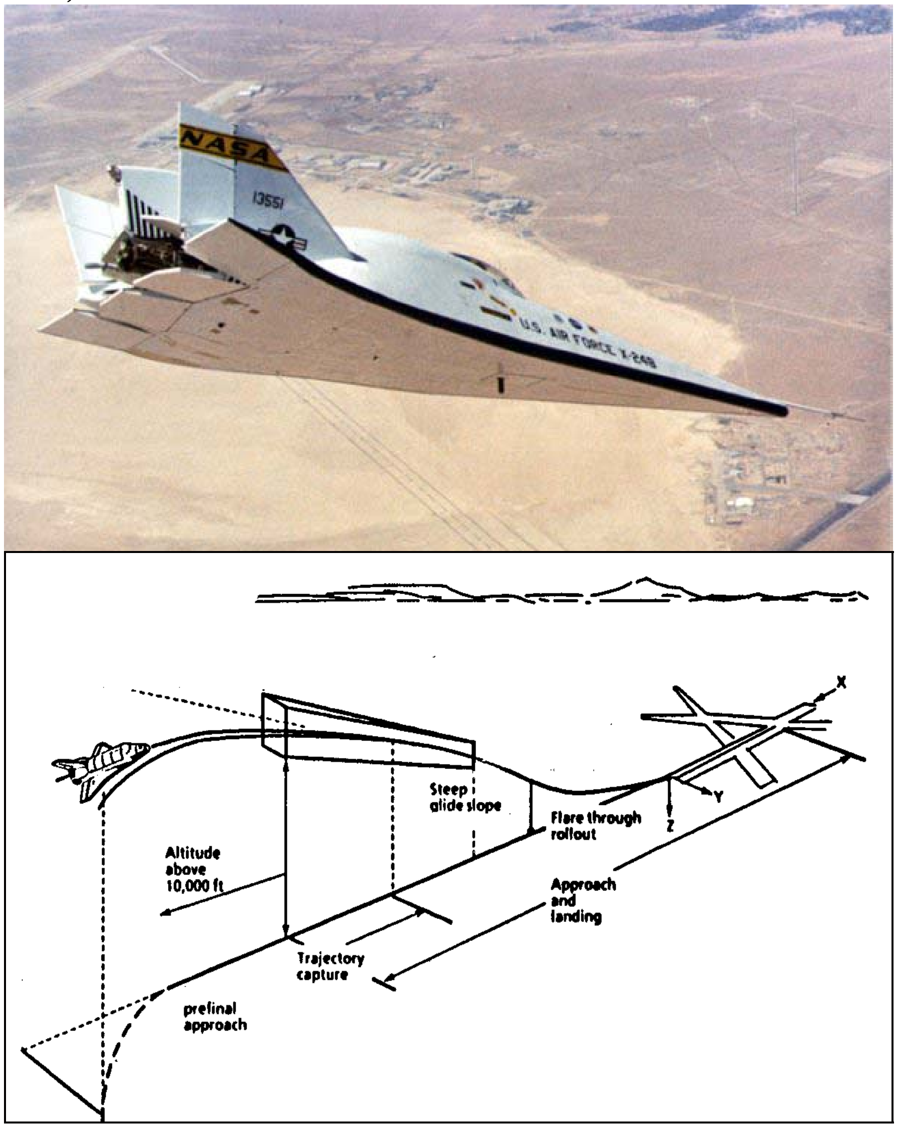
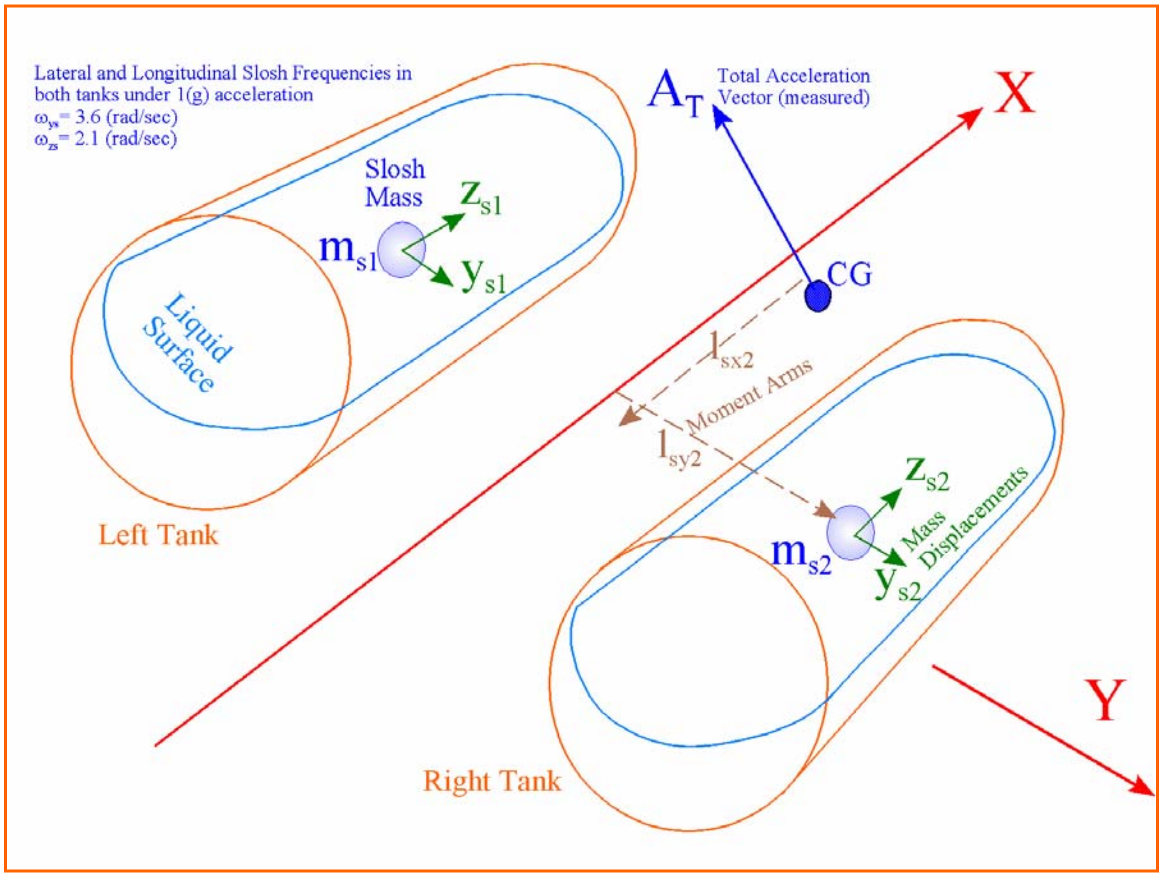


Figure 1 Approach and Landing

2. The Fuel Sloshing Model

The vehicle has two identical fuel tanks symmetrically positioned on the left and the right sides of the vehicle, as shown in figure (2). They contain residual RCS fuel used for attitude control at high altitudes. The fuel in each tank is sloshing in two directions parallel to the surface (perpendicular to the total acceleration vector A_T). The fuel motion generates slosh disturbance forces on the vehicle which interfere with system stability and vehicle 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 different. The longitudinal slosh frequency (oscillating in the x-z plane of the vehicle) is ω_{zs}=2.1 (rad/sec) and the damping is ζ_{zs}=0.008. The lateral slosh frequency (oscillating along the y vehicle direction) is ω_{ys}=3.6 (rad/sec) and the damping is ζ_{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 the un-deflected positions at steady state are 4.5 (ft) to the left and to the right of the vehicle x centerline. The masses are also 7.5 (ft) behind, and 2.3 (ft) above the vehicle CG.



Lateral and Longitudinal Slosh Frequencies in both tanks under 1(g) acceleration
ω_{ys}= 3.6 (rad/sec)
ω_{zs}= 2.1 (rad/sec)
A_T
Total Acceleration Vector (measured)
X
Slosh Mass
m_{s1}
z_{s1}
y_{s1}
CG
Liquid Surface
l_{sx2}
Moment Arms
l_{sy2}
z_{s2}
m_{s2}
y_{s2}
Mass Displacements
Left Tank
Right Tank
Y

Figure 2 Fuel Tank Geometry

"*Autoland.qdr*" to a Matlab state-space m-file "Siso_Anal.m", and saved in directory "\Mat Pitch". It is used by the script file "run.m" to perform frequency domain analysis. The pitch guidance system "*Guidance (Altitude and Speed Control)*" must also be converted to an m-function filename "guidance.m" and saved in the same pitch analysis directory.

4.2 Pitch Flight Control and Guidance Systems

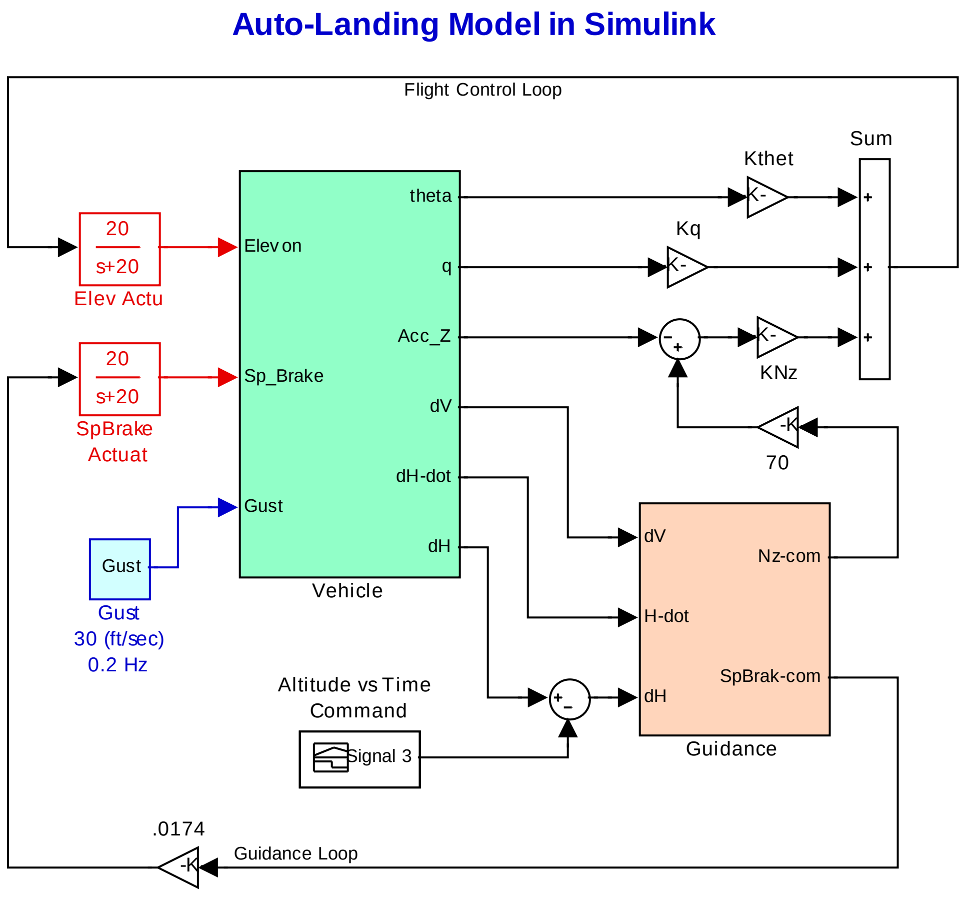


Figure (4.2.1) Pitch flight control and guidance system

The pitch axis flight control and guidance system is shown in figure (4.2.1). The flight control system is designed to control the vehicle normal acceleration Nz. It combines the Nz error signal with the pitch rate q and attitude θ, and drives the Elevon actuator. The Nz-command input to the FCS comes from guidance. There is also a guidance loop that regulates the vehicle altitude by issuing an Nz command to the flight control loop. The guidance consists of two loops, (a) an altitude (h) control loop, and (b) a speed (V) control loop. The altitude control loop is attempting to track a pre-computed altitude reference (h-ref). It is 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 using the speed brake which varies the vehicle drag coefficient.

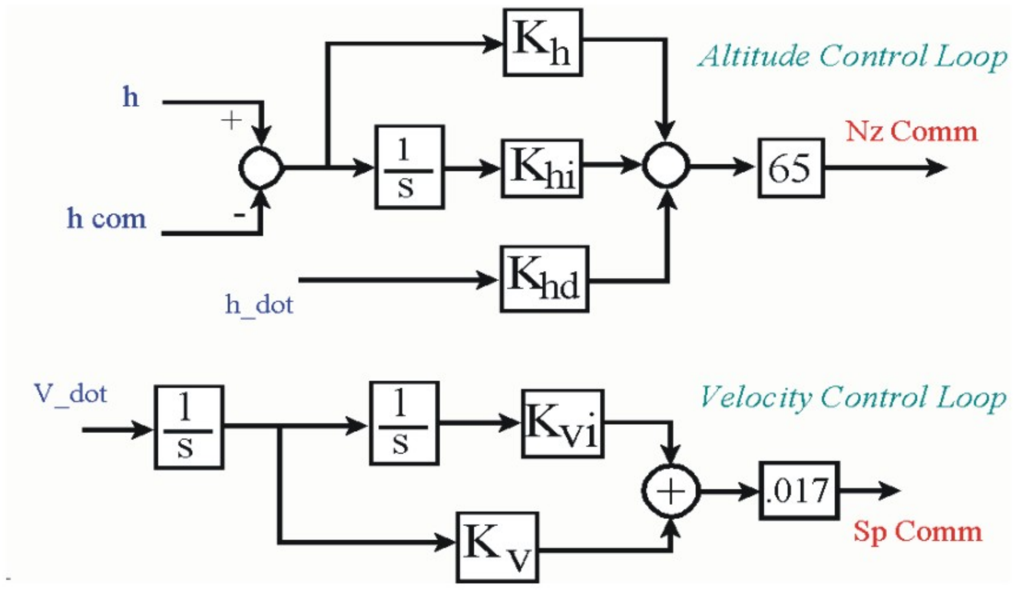


Figure (4.2.2) Altitude and Speed Control Guidance

The pitch vehicle output consists of: pitch attitude (θ), rate (q), angle of attack (α), change in altitude (δh), altitude rate (h-dot), change in velocity (δV), Nz acceleration and slosh mass displacements (z_s) with respect to the tank.

4.3 Pitch Axis Simulation in Simulink

The Simulink model used to simulate the automatic landing of the vehicle in the longitudinal axes is shown in Figure (4.2.1). The Simulink file is "*Autoland.Mdl*" in subdirectory "*Examples\Autoland\Mat Pitch*". It contains the pitch guidance subsystem (guidance.m) and the vehicle subsystem (pitch_vehi3.m). The script file "run.m" is used to load the state-space subsystems into Matlab workspace. The file "run.m" initializes also the vehicle state vector at [θ, q, α, h, δV] = [0.1, -0.08, 0.18, 9000, 50], and the slosh mass displacements and velocities at zero.

The simulation results are shown in figures (4.3.1 to 4.3.3). It starts when the vehicle 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 control surfaces. The Elevon attempts to control its altitude and the Speed-Brake its speed, but of course there is a significant amount of coupling between the two loops. In reality the vehicle parameters, such as dynamic pressure, aerodynamic coefficients, etc, change during flight, but in this example we are using the same linear model during the 100 seconds flight. The purpose of this exercise is to demonstrate the auto-landing modeling and design methodology rather than to present accurate results. In order to obtain more accurate results the analyst should develop a 6-dof simulation with time varying parameters.

4.4 Pitch Axis Open-Loop Stability Analysis

The following diagram in figure (4.4.1) shows the interconnection system used for longitudinal open-loop frequency domain analysis. The control loop is opened at the elevon input. The speed-brake loop is closed. This block diagram does not represent an actual Simulink model. The state-space system for open-loop analysis was created using the Flixan program which combined the vehicle, actuator, and guidance subsystems as described earlier. Its title is: "*Elevon Open-Loop System (Speed-Brake Closed)*", and it was saved in file "*Autoland.Qdr*".

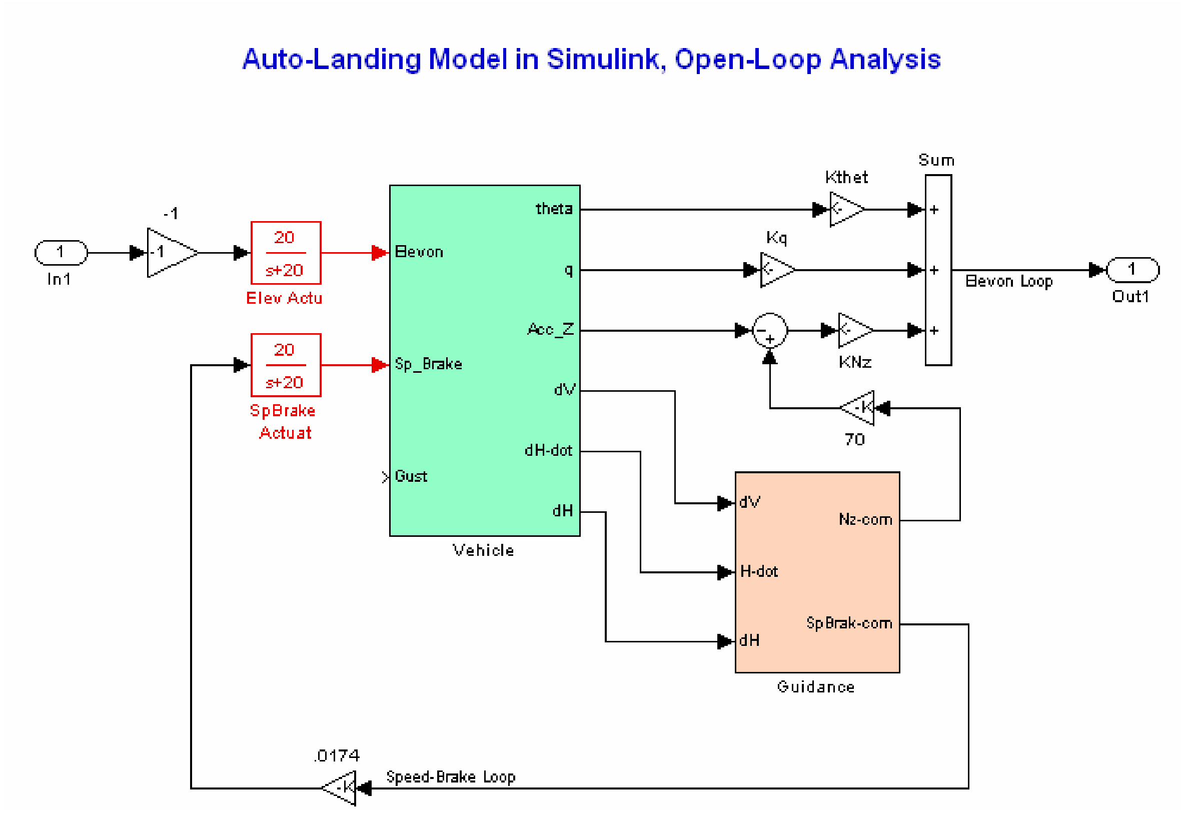


Figure (4.4.1) Open-Loop Interconnection for Elevon Stability Analysis

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 model (which was created earlier) is loaded into Matlab workspace from file "Siso_Anal.m". Matlab calculates the frequency response of the open-loop system and computes the Nichols Chart as shown in figure (4.4.2).

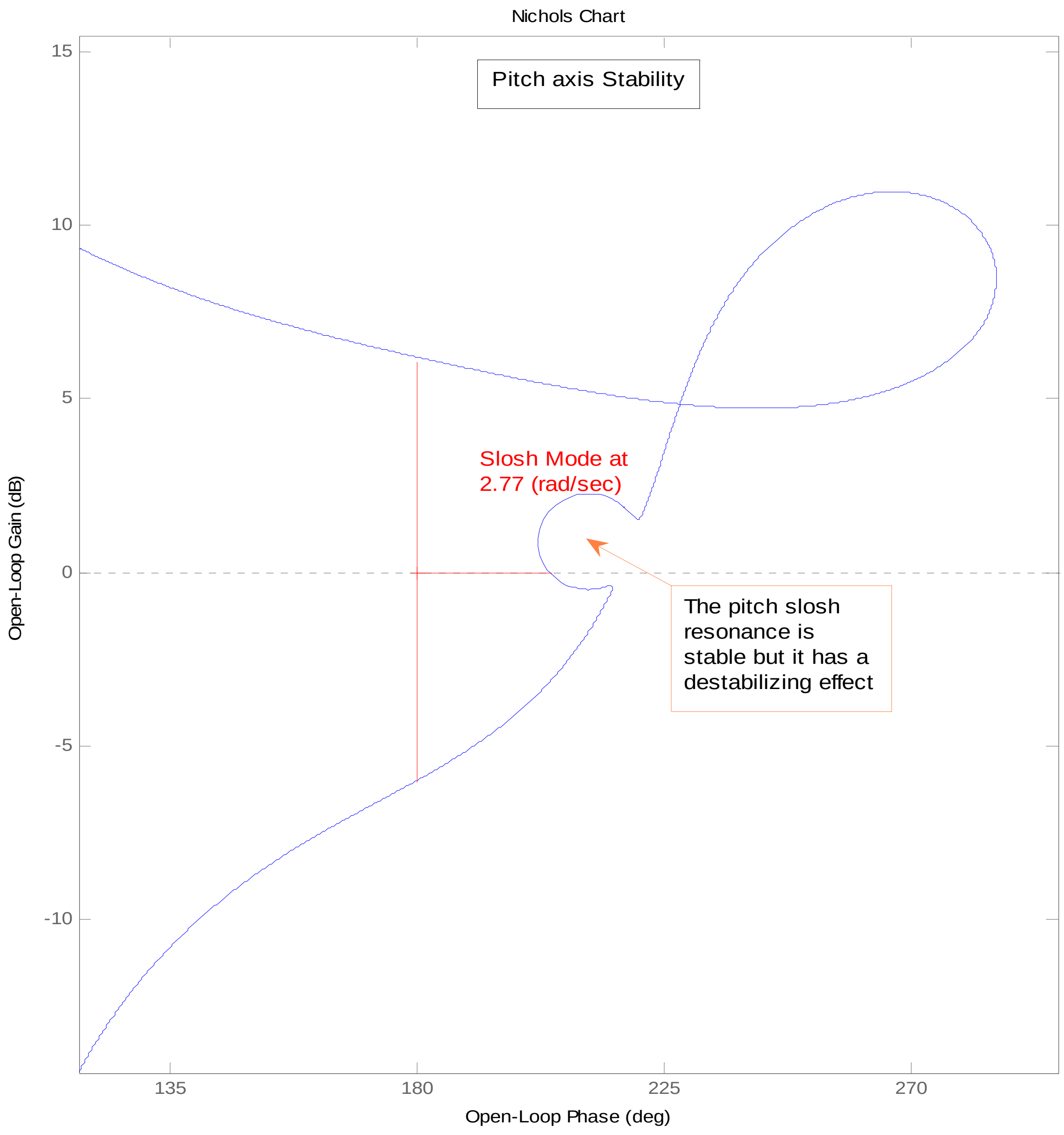


Figure (4.4.2) Nichol's Chart across the Elevon loop is showing that the system has 7 (dB) of gain margin and 32 (deg) of phase margin. The slosh resonance tends to degrade the pitch phase margin

5. Lateral Axes Analysis

The lateral control system uses the aileron and the rudder control surfaces for roll and yaw stabilization. It uses also 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 (β) is directly measurable. The lateral guidance system uses 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 used for LQR state-feedback design "*Automatic Landing of Unpowered Vehicle, Lateral Design Model*", and a simulation model "*Automatic Landing of Unpowered Vehicle, Lateral Axes-2*". Actually, the original coupled vehicle model was modified (not shown here) to change the direction of the wind gust for more effectiveness in the lateral direction. The wind gust directions used in the lateral model are: (el=90, az=90 deg).

<u>Extract the lateral vehicle</u>. To extract the lateral vehicle subsystem (the one used in Matlab simulations) from the fully coupled vehicle system it involves two steps and the creation of an intermediary system. We first use the system modification data "*Automatic Landing of Unpowered Vehicle, Lateral Axes*" which are already saved in file "*Autoland.Inp*" to extract the lateral subsystem from the original vehicle, and in the second step we use the system modification data "*Automatic Landing of Unpowered Vehicle, Lateral Axes-2*" to include some additional outputs in the previous system for monitoring the slosh mass activity inside the tanks.

After selecting the project directory"*\Flixan\ Flight\ Examples\ Autoland*", go to "Analysis Tools", then to "Create State-State Systems", and then to "Extract and Modify Systems". From the filenames selection menu select the input and system filenames, "*Autoland.Inp*" and "*Autoland.Qdr*". From the systems modification titles menu select the title: "*Automatic Landing of Unpowered Vehicle, Lateral Axes*", and click on "Run Input Set". This set of instructions will extract the lateral subsystem from the original vehicle system and save it in file "*Autoland.Qdr*" under the same title "*Automatic Landing of Unpowered Vehicle, Lateral Axis*".

Repeat the program for the second time. Select the same input and system filenames. From the systems modification titles menu select the title: "*Automatic Landing of Unpowered Vehicle, Lateral Axes-2*", and click on "Run Input Set". This set of instructions will use the previously created system and include some additional slosh mass displacement outputs 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 Axis-2*". A similar process is used to extract the lateral design model, and it is not shown here.

<u>Exporting the lateral vehicle model into Matlab.</u> The lateral simulation system "*Automatic Landing of Unpowered Vehicle, Lateral Axis-2*" which is in systems file "*Autoland.qdr*" must now be converted to a state-space format that can be imported into Matlab subdirectory "\Autoland\Mat Lateral" in order to perform Matlab simulations. Run Flixan and select the same project directory "\Autoland". Go to "File", select "Matlab Conversions", and "Export to Matlab". Select the systems filename "Autoland.Qdr". 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 "*Automatic Landing of Unpowered Vehicle, Lateral Axis-2*" and click "OK". Enter the filename "vehi_latan2.m" (without the .m) where the state-space data will be saved as a Matlab function. Repeat the system conversion also for the design model "*Automatic Landing of Unpowered Vehicle, Lateral Design Model*". This system is saved in file "vehi_ldes.m" in the same directory "\Autoland\Mat Lateral".

5.2 Linear Lateral Stability Analysis

The following model in figure (2.5.1) is used to perform open-loop frequency domain stability analysis. This model is in Simulink file "Lat_Open.mdl" and it can be found in directory "*Examples\Autoland\Mat Lateral*". A Matlab lateral analysis m-file script "run_lat.m" is used to load the design model from file "vehi_ldes.m", and the analysis model from file "vehi_latan2.m" which is used for frequency domain analysis and simulations. The script file uses also the design model to synthesize an LQR state-feedback gain matrix (Kg). The script uses the Simulink model "Lat_Open.mdl" (shown below with the aileron loop opened and the rudder loop closed) to create an open-loop state-space system and calculates the Bode and Nichols plots which are used to evaluate the aileron stability margins. The Simulink model can be modified by closing the aileron and opening the rudder loop to evaluate the rudder stability. The Nichols plots in figures (5.2.2) and (5.2.3) show good stability margins on both aileron and rudder loops, and that slosh does not degrade the lateral stability.

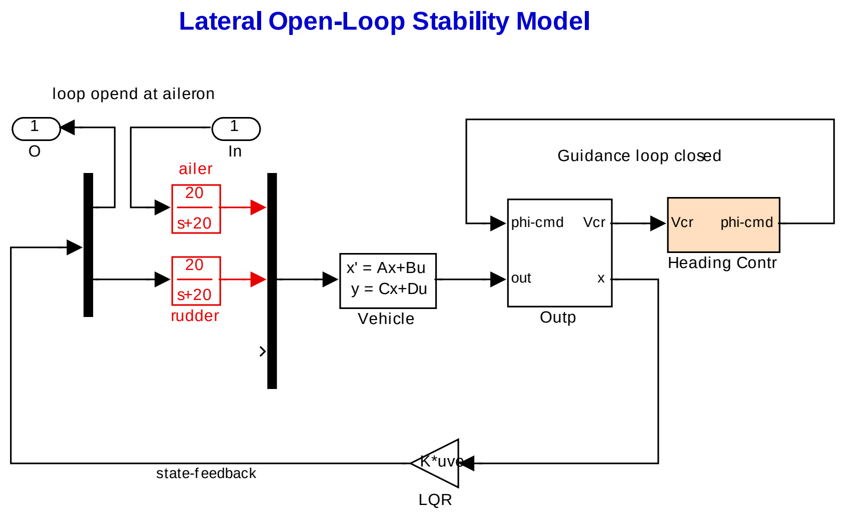


Figure (5.2.1) Simulink model for Open-Loop Analysis. The loop is opened at the aileron input.

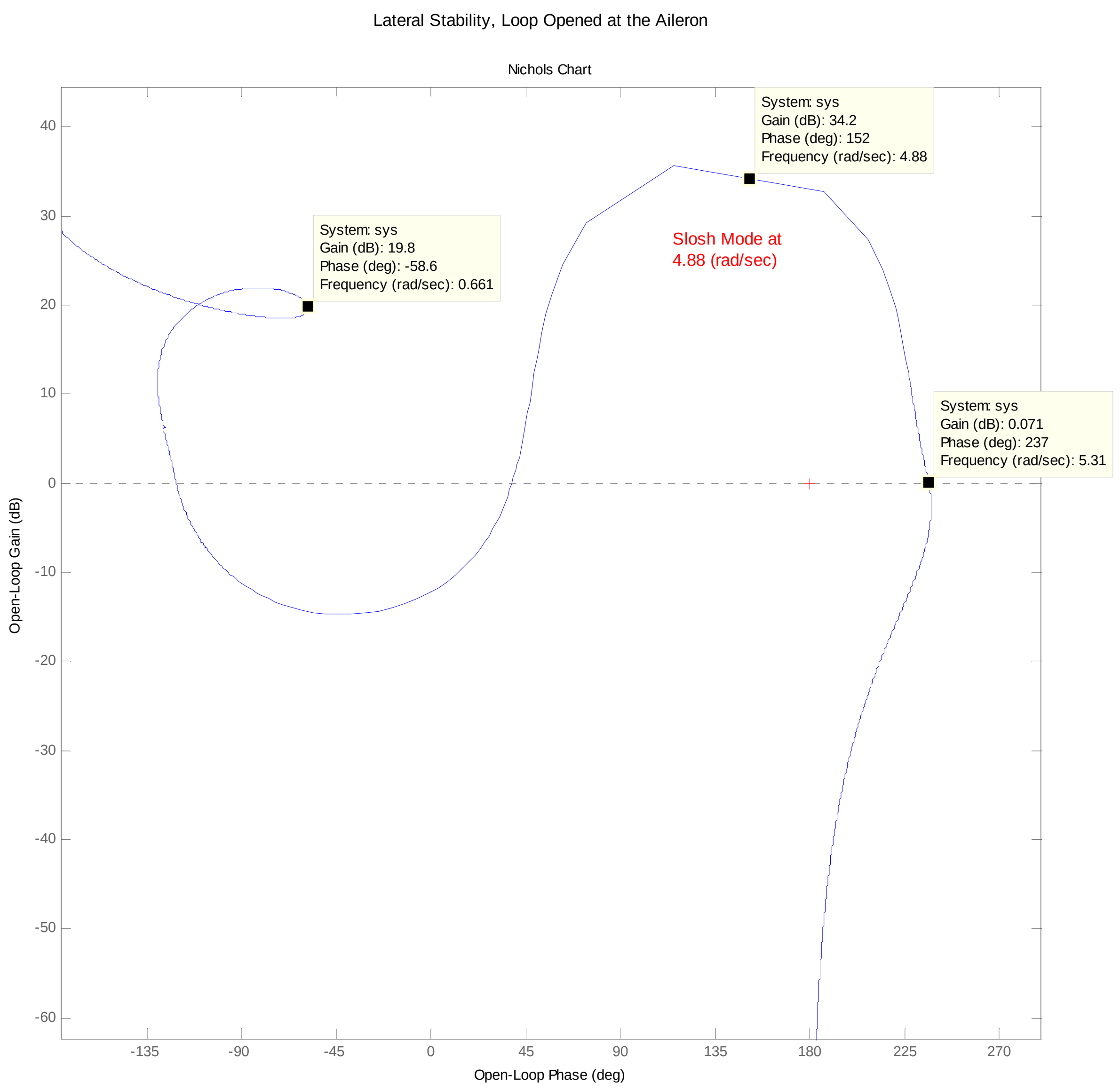


Figure (5.2.2) Lateral stability with the loop opened at the aileron shows a stable slosh resonance at 4.88 (rad/sec)

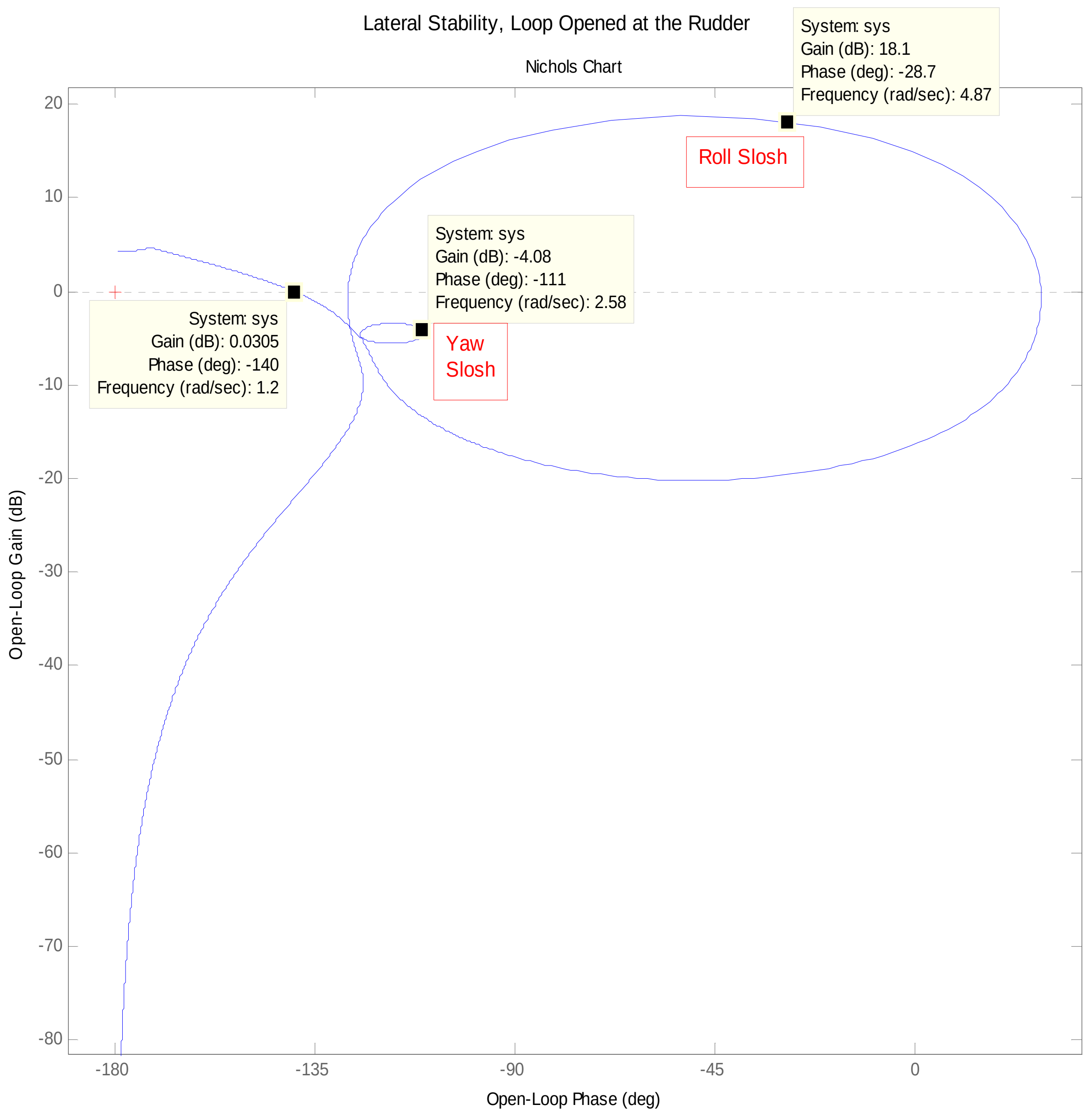


Figure (5.2.3) Lateral stability with the loop opened at the rudder shows two stable slosh resonances at 2.58 and 4.88 (rad/sec)

5.3 Lateral Simulations

The Simulink model used in lateral simulations is in file "Lat_Sim.mdl", in directory "*Examples\Autoland\Mat Lateral*". The vehicle simulation model "vehi_latan2.m" is loaded by running the Matlab script file "run_lat.m". The script calculates also the LQR state-feedback gains. The Simulink model is shown in figure 5.3.1 (a & b) below.

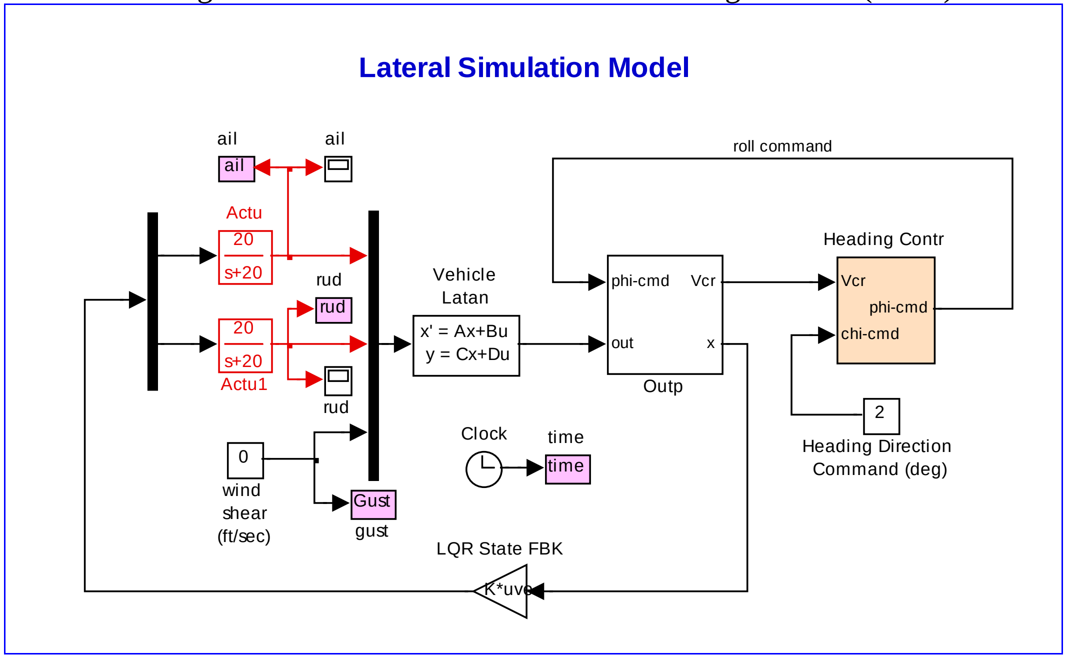


Figure (5.3.1a) The Simulation Inputs are: (a) Heading direction command, and (b) Wind-Gust Velocity (ft/sec). The wind direction is pre-defined.

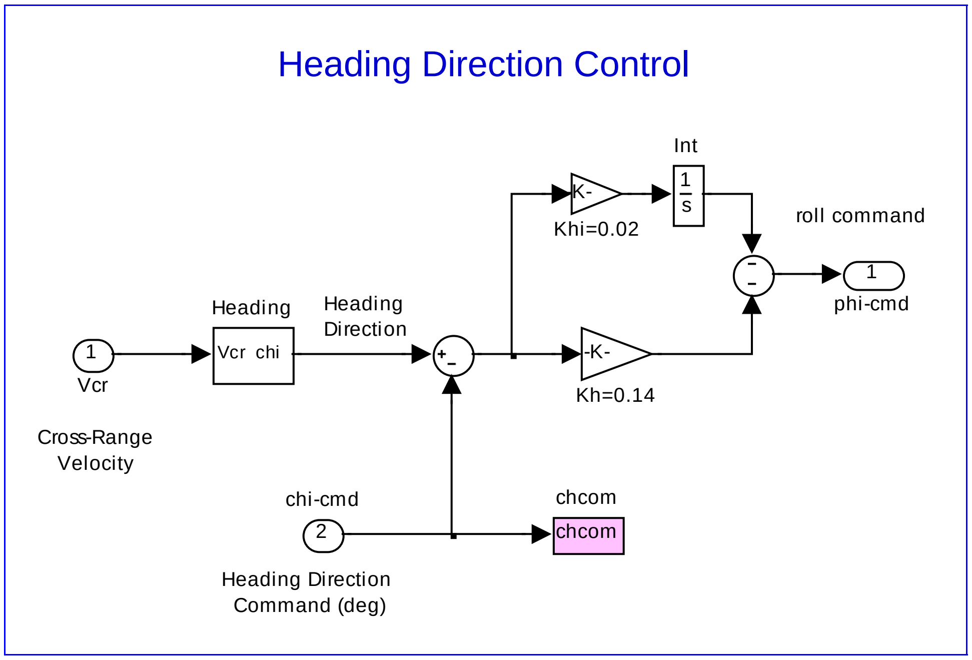


Figure (5.3.1b) PI Heading direction guidance system. The cross-range velocity is converted to heading direction angle (deg). The error generates the vehicle (phi-command) .