EML 6934-Robust Control Project

Submitted to: Dr.Benjamin Dickinson, University of Florida

Submitted by: Prajwal Gowdru Shanthamurthy

December 11, 2020

Part 1 - Autopilot Design

The open-loop poles of A matrix are:

$$\begin{bmatrix} -1.2895 + 21.8316i \\ -1.2895 - 21.8316i \\ -1.0695 + 9.0870i \\ -1.0695 - 9.0870i \\ -0.5920 + 0.0000i \end{bmatrix}$$

Of these, $(-1.2895 + 21.8316i, -1.2895 - 21.8316i)$ correspond to Dutch Roll Mode; $(-1.0695 + 9.0870i, -1.0695 - 9.0870i)$ correspond to Short Period Mode; the real pole, -0.5920 corresponds to Roll Subsidence Mode.
For the given system, the Dutch Roll Mode is the fastest (most negative real part of poles), meaning oscillations in the rolling-yawing motion die down faster than oscillations in the longitudinal motion(i.e. AoA and Pitch Rate: Short Period Mode).
All three modes are stable.

The extracted short period dynamics state-space model (A_{sp},B_{sp},C_{sp},D_{sp}) has the following poles: $\begin{bmatrix} -1.0700 + 9.0923i \\ -1.0700 - 9.0923i \end{bmatrix}$ which are approximately the same as the original short period mode poles.

Control Objective:

The states available for feedback are vertical acceleration (A_z) and Pitch Rate,q. The plant receives inputs from an actuator whose dynamics is not entirely known, except that it's a second order under-damped system of unit steady state gain.

The control objective is to design an autopilot (controller + observer) such that the closed-loop compensator's output A_z tracks the desired $A_{z_{cmd}}$ while satisfying the frequency and time domain requirements mentioned in the problem statement. The actuator saturation requirements are to be met to the end of A_z tracking $A_{z_{cmd}}$ generated by the Guidance problem in Part 2.

The LQ Controller and Observer Penalty matrices are to be tuned to minimize Miss Distance in the in the integrated autopilot-guidance system while satisfying the other design specifications.

Robust Servo Mechanism is a tracking controller that can be used to asymptotically track general reference signals even in the presence of plant uncertainties or external disturbances of a particular form. For this problem, we use RSMDM meant to track a step reference signal (nothing but a PI controller). A Leuenberger observer is used to estimate the full state for use in the LQR state feedback law.

Integrating the Autopilot with Guidance:

The Autopilot takes $A_{z_{cmd}}$ as the input and generates A_z (the actual vertical acceleration). A_z is fed in as the acceleration of the Missile perpendicular to the LOS vector into the ProNav dynamics. The ProNav Guidance law in turn generates $A_{z_{cmd}}$ based on the current positions and velocities of the missile and the target.

Assumption made: It should be noted that A_z is assumed to be aligned with the perpendicular to the LOS direction of the guidance problem. A_z being the vertical acceleration in the inertial frame, this assumption is reasonable only when we're dealing with small LOS angles.

Tuning Process:

a) Tuning the Controller

For the sake of designing the controller, we assume full-state feedback.

LQR is used for computing the control gains. Tuning the LQR penalties gives us a handle on the time response and frequency response characteristics of the closed-loop system (Design Specifications in the given problem). For this problem, we fix the control penalty R at 1. In the state penalty matrix, Q we vary the integral error penalty *(qq =logspace(-5,-2,50))*; while keeping all other entries of Q at zero, get the frequency response (Nyquist plots of the loop gain at the plant input) and observe the how well the design requirements are met.The integrated autopilot-guidance closed-loop is also simulated to identify actuator saturation and miss distance. The performance of the controller in this exercise is plotted onto "design charts" to identify the design which gives the minimum miss without violating the design requirements.

Important Note: Once we bring the observer into the picture in the next stage of design, the values of the design parameters change. Although, the behavior of the full-state-feedback closed-loop system can be recovered using Loop Transfer Recovery (LTR), we do not want to make the Observer Gains too large in an attempt to make the dynamic compensator meet the same margins, since that would cause other issues (Sensor noise amplification to give an example).

In view of this, we try to be generous in satisfying the design margins while tuning the

LQR. *In the code I have made sure the tuning process picks an integral error penalty for which each design specification is satisfied with a 10 percent safety margin of the original requirement.*

The MATLAB code has been written to automate the tuning i.e. it picks the best design among the different penalties which are tried out without us having to inspect anything. Although, we don't necessarily have to rely on design charts for picking the design point, it provides an excellent visual aid to facilitate the tuning process.

Note that the spike in $A_{z_{cmd}}$ at the end of the integration time interval is due to a numerical issue that arises in the ProNav Guidance law on account of the relative distance between the missile and the target (which appears in the denominator) taking on very small values. It can therefore be ignored since real life scenarios won't run into this issue due to effects like IMU seeker noise various other factors

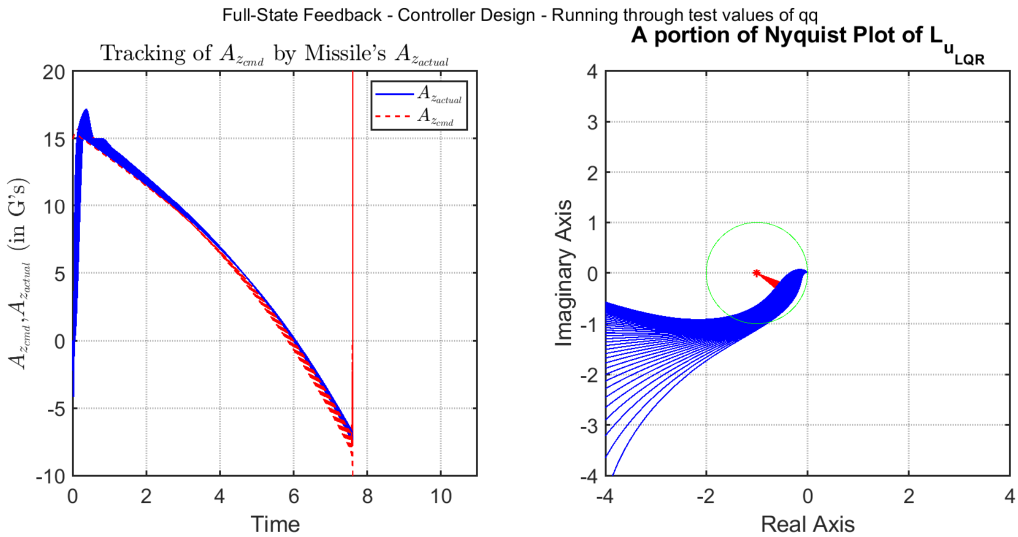
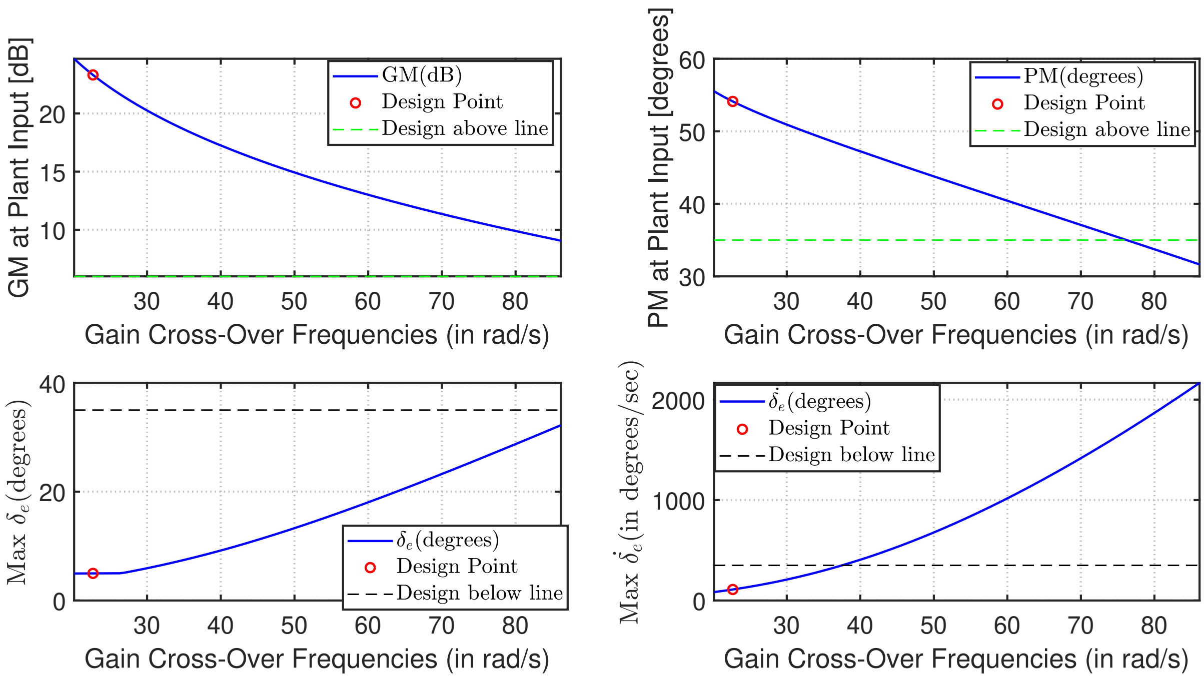
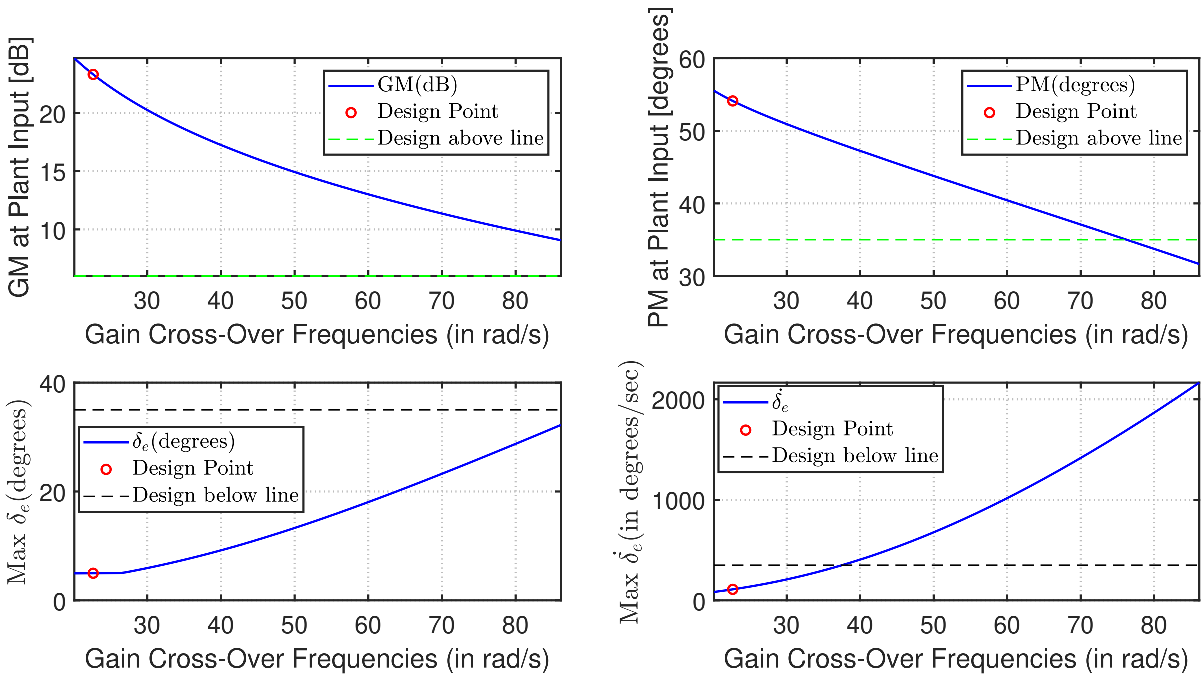


Figure 1: $A_{z_{cmd}}$ tracking and Nyquist Plots for different integral error penalties


GM at Plant Input [dB]
GM(dB)
Design Point
Design above line
Gain Cross-Over Frequencies (in rad/s)
PM at Plant Input [degrees]
PM(degrees)
Design Point
Design above line
Gain Cross-Over Frequencies (in rad/s)
Max δ_e(degrees)
δ_e(degrees)
Design Point
Design below line
Gain Cross-Over Frequencies (in rad/s)
Max $\dot{\delta}_e$(in degrees/sec)
$\dot{\delta}_e$(degrees)
Design Point
Design below line
Gain Cross-Over Frequencies (in rad/s)


GM at Plant Input [dB]
GM(dB)
Design Point
Design above line
Gain Cross-Over Frequencies (in rad/s)
PM at Plant Input [degrees]
PM(degrees)
Design Point
Design above line
Gain Cross-Over Frequencies (in rad/s)
Max δ_e(degrees)
δ_e(degrees)
Design Point
Design below line
Gain Cross-Over Frequencies (in rad/s)
Max $\dot{\delta}_e$(in degrees/sec)
$\dot{\delta}_e$
Design Point
Design below line
Gain Cross-Over Frequencies (in rad/s)

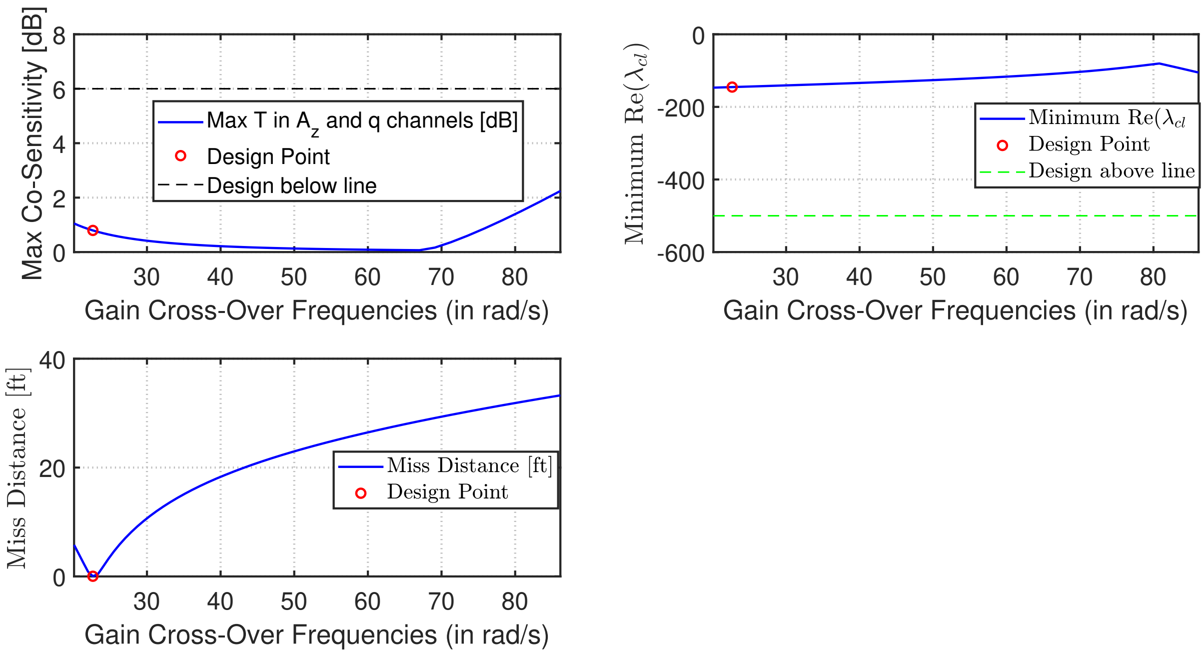
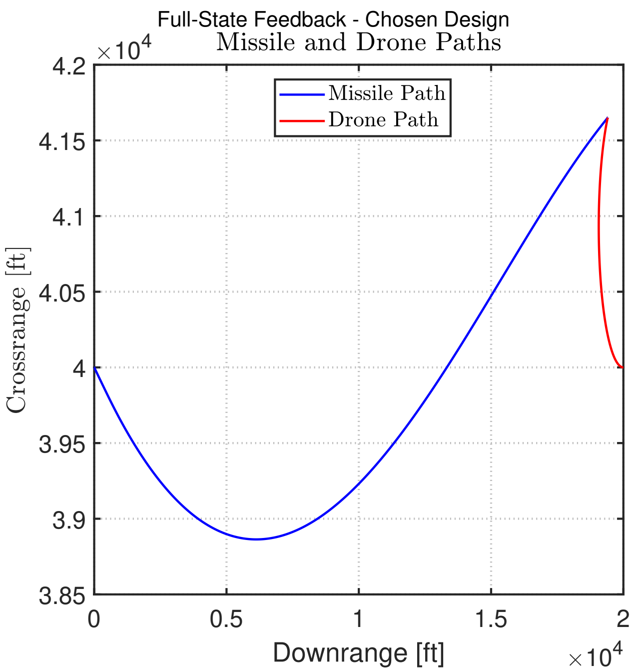
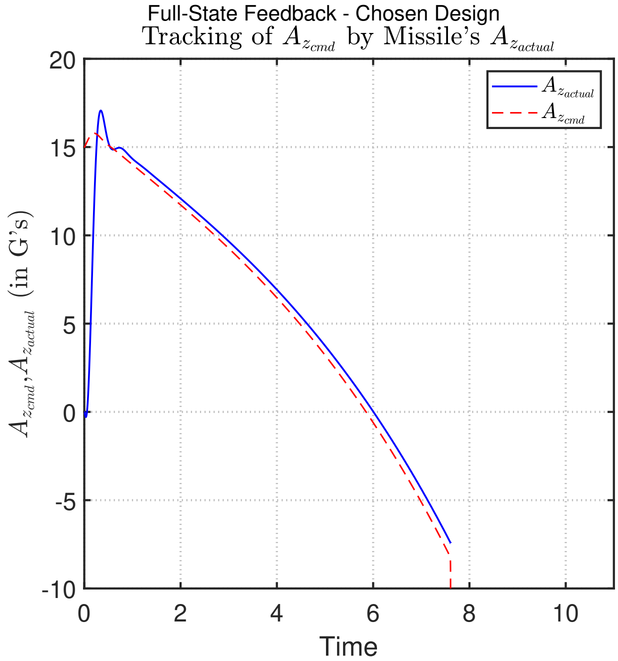
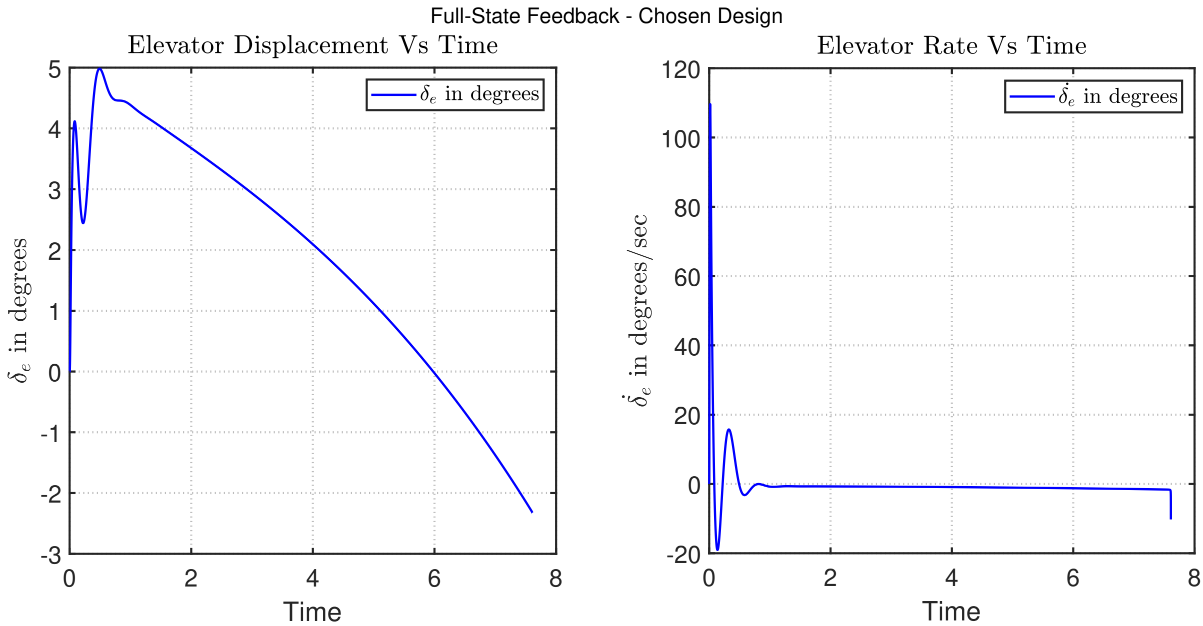


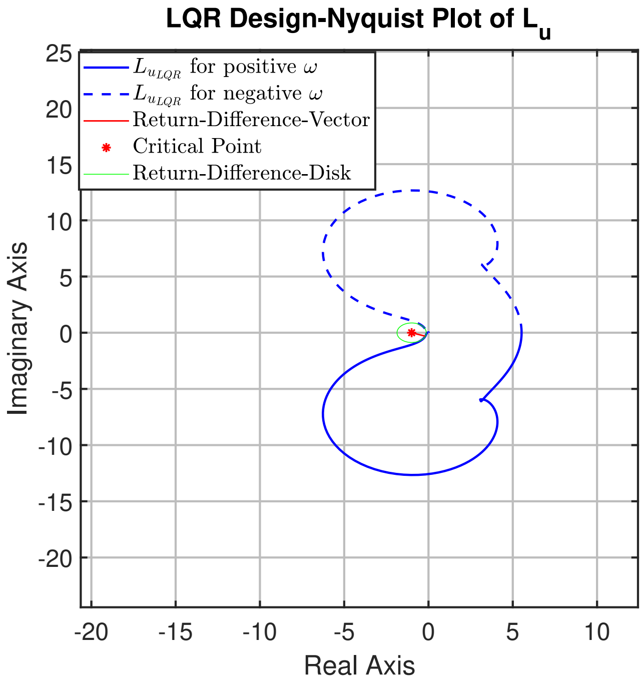
Figure 2: Design Charts (One for each specification)

Chosen Integral Error Penalty: qq = 1.7575e-05;
Miss Distance = 0.0349 ft




Full-State Feedback - Chosen Design
Tracking of $A_{z_{cmd}}$ by Missile's $A_{z_{actual}}$
$A_{z_{actual}}$
$A_{z_{cmd}}$
$A_{z_{cmd}}, A_{z_{actual}}$ (in G's)
Time


Full-State Feedback - Chosen Design
Elevator Displacement Vs Time
δ_e in degrees
δ_e in degrees
Time
Elevator Rate Vs Time
$\dot{\delta}_e$ in degrees
$\dot{\delta}_e$ in degrees/sec
Time



Discussion on the Nyquist Plot: The open loop system has all poles in the LHP. So, the loop gain at the plant input has one all poles in the LHP except one integrator pole at the origin. So, we'll need to go around the origin with Γ_N while plotting hte Nyquist curve.

The Nyquist plot should have zero encirclements of the critical point (-1+0j) for the closed-loop system to be stable. As can be seen from the plot, it does in fact do so.

Given that closed-loop stability is satisfied, we now check the other frequency domain design criteria.

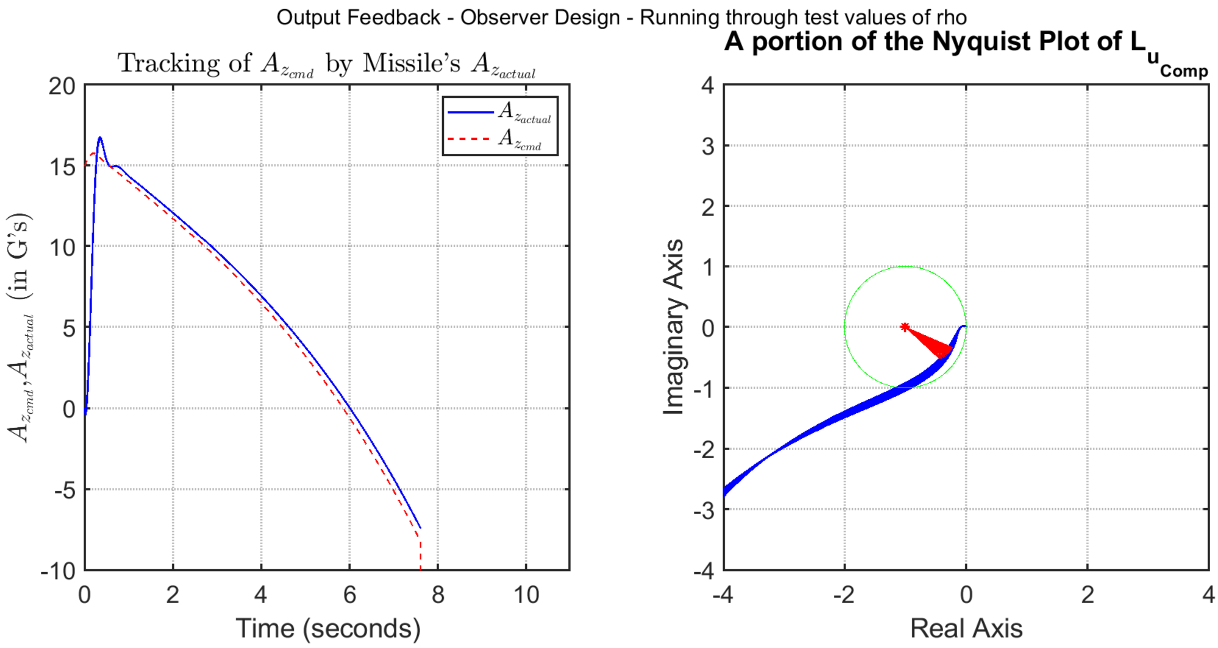
Caveat: Given that the Nyquist plot has multiple phase cross-over frequencies, we should not rely on the 'margin' MATLAB command to find the gain margin since it could give this value with respect to a phase cross-over frequency which is not so critical to stability of the closed-loop system. The code uses phase margin from the command 'margin' but employs an adhoc algorithm to find the gain margin that actually matters.

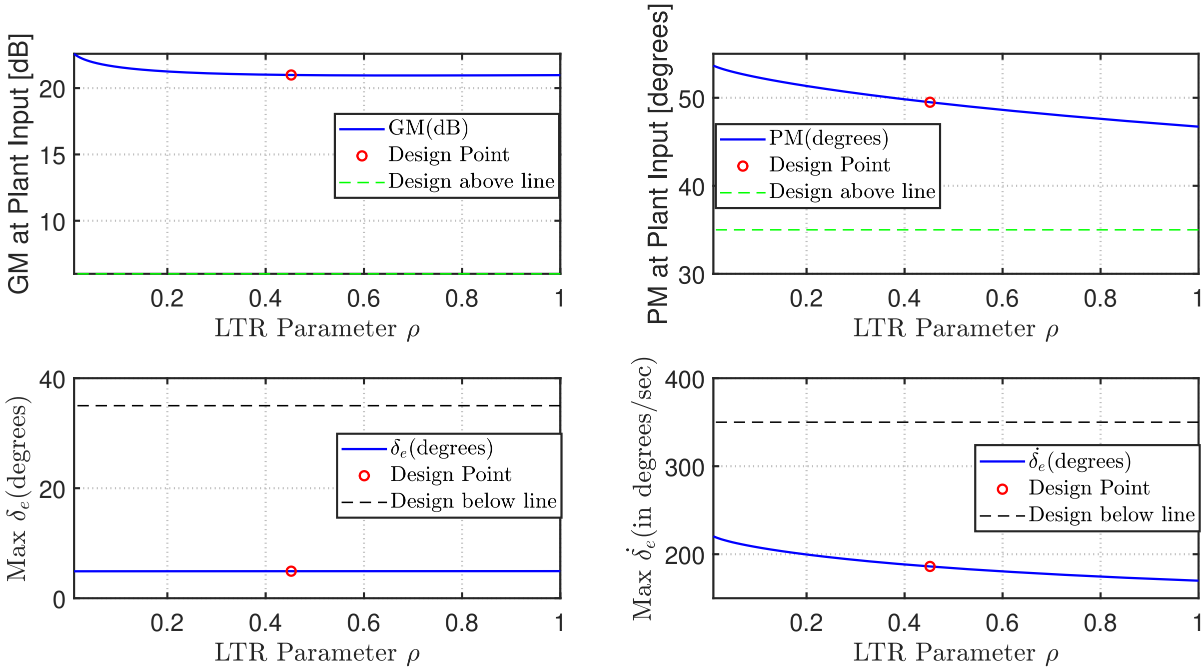
NOTE : The same discussion points on the Nyquist plot hold for the dynamic compensator as well

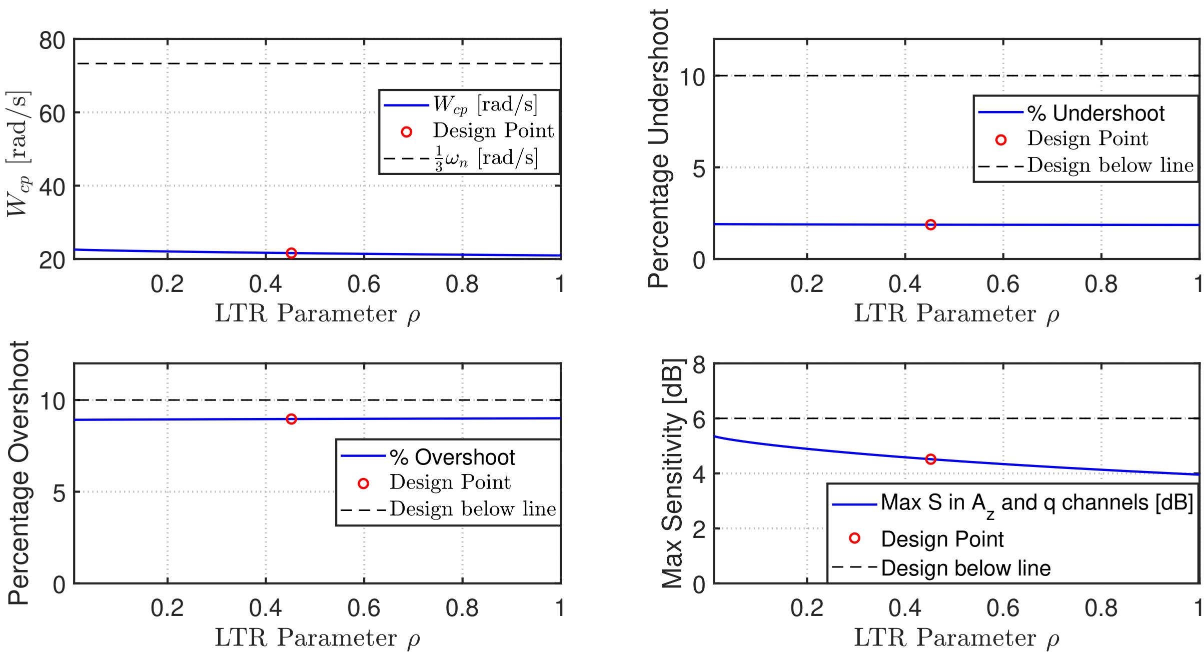
b) Tuning the observer: LQR is used on the Observer Design Model Matrices to arrive at the Observer Gain Matrix. The penalties are chosen as follows:

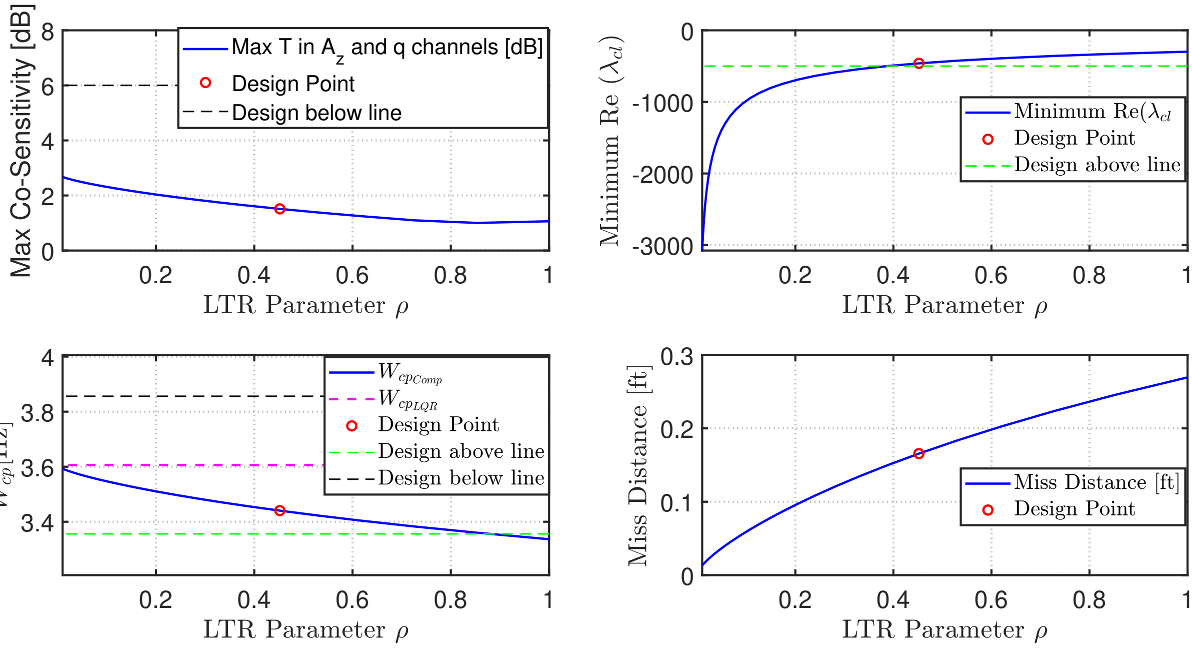
$$Q_e = Q_o + \frac{1}{\rho}(\tilde{B}\tilde{B}^T)$$

where Q_o was chosen as a 3×3 identity matrix and ρ is the LTR parameter varied as *logspace(0,-2,30)*. Lower values of ρ result in more loop recovery. But we need to make sure that it does not get so low as to violate the design specification on the minimum value of closed loop poles.



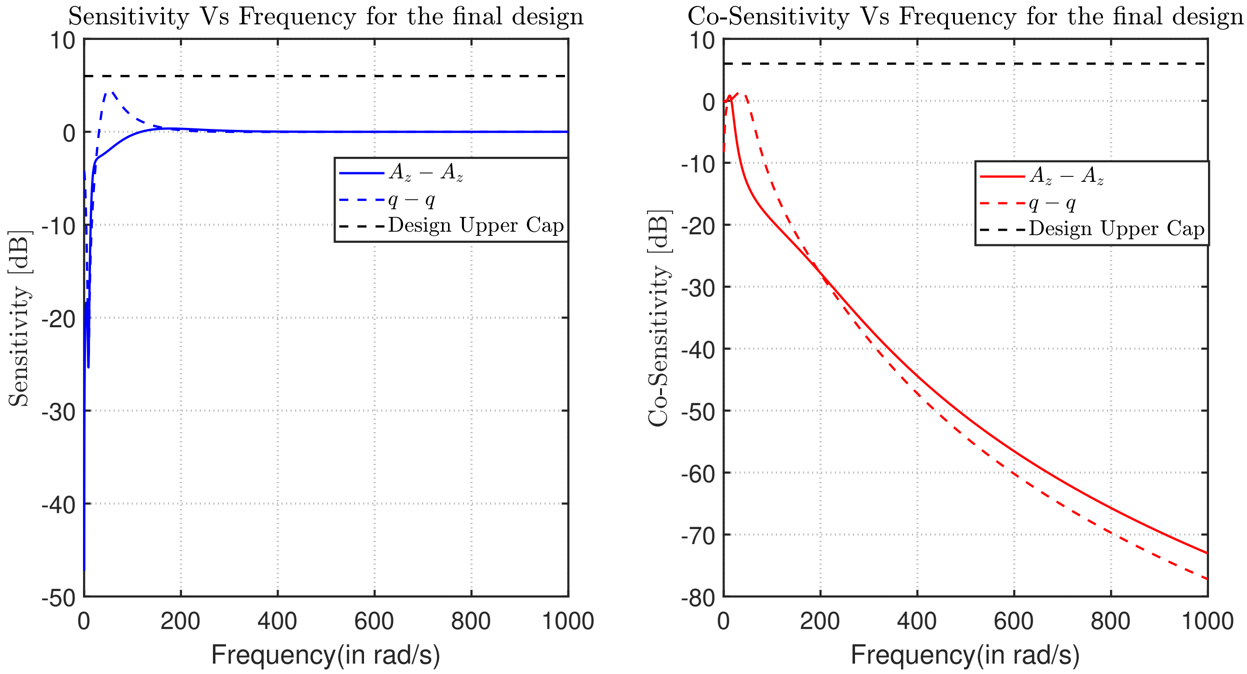

GM at Plant Input [dB]
GM(dB)
Design Point
Design above line
LTR Parameter ρ
PM at Plant Input [degrees]
PM(degrees)
Design Point
Design above line
LTR Parameter ρ
Max δ_e(degrees)
δ_e(degrees)
Design Point
Design below line
LTR Parameter ρ
Max $\dot{\delta}_e$(in degrees/sec)
$\dot{\delta}_e$(degrees)
Design Point
Design below line
LTR Parameter ρ

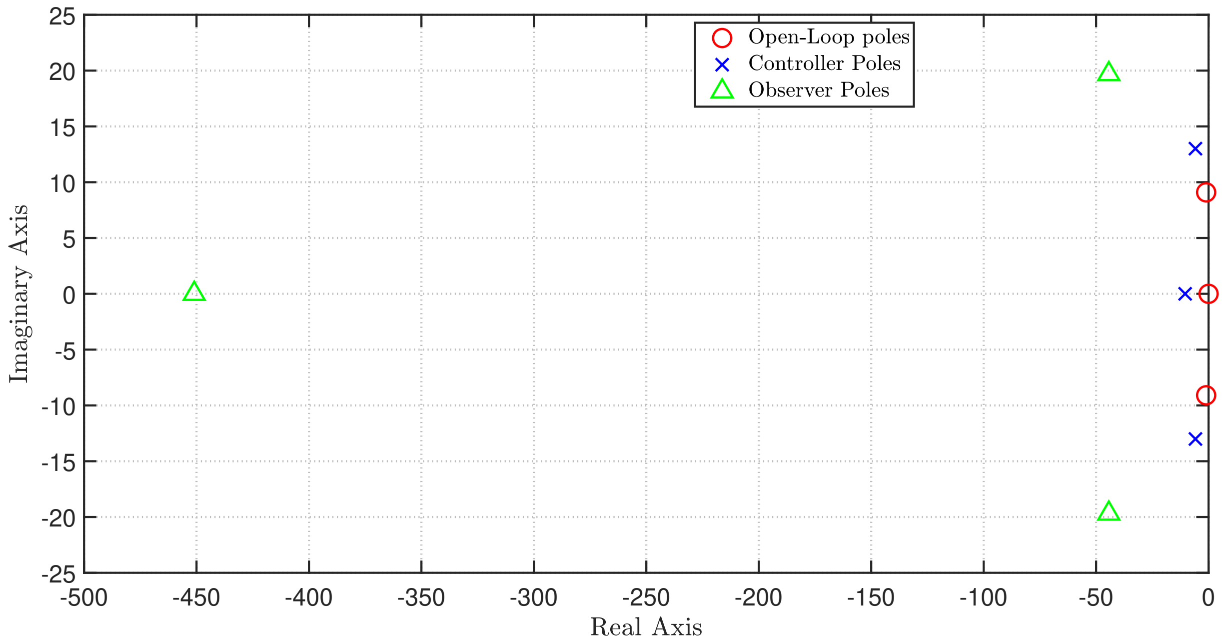

W_{cp} [rad/s]
W_{cp} [rad/s]
Design Point
$\frac{1}{3}\omega_n$ [rad/s]
LTR Parameter ρ
Percentage Undershoot
% Undershoot
Design Point
Design below line
LTR Parameter ρ
Percentage Overshoot
% Overshoot
Design Point
Design below line
LTR Parameter ρ
Max Sensitivity [dB]
Max S in A_z and q channels [dB]
Design Point
Design below line
LTR Parameter ρ



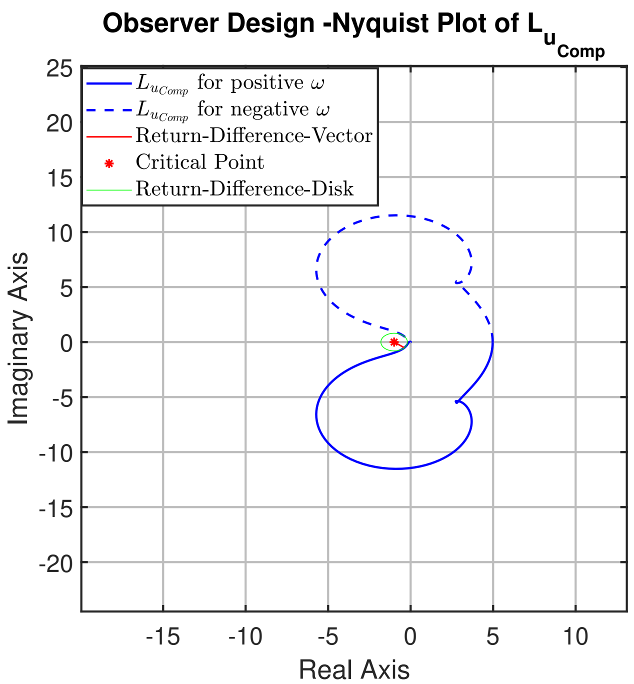
Chosen $\rho = 0.4520$

Miss Distance = 0.0353 ft

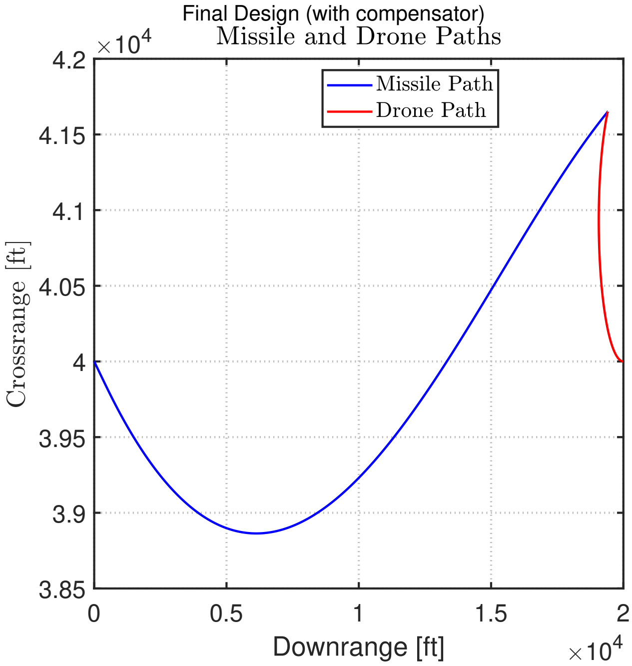




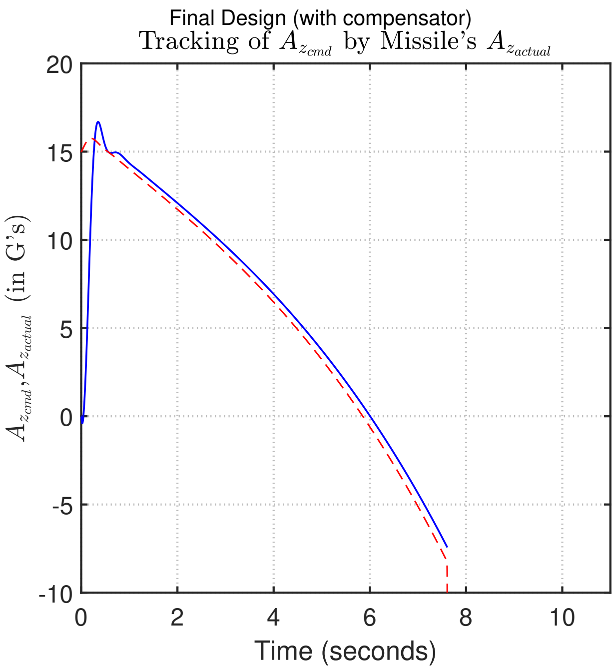
Open-Loop poles
Controller Poles
Observer Poles
Imaginary Axis
Real Axis



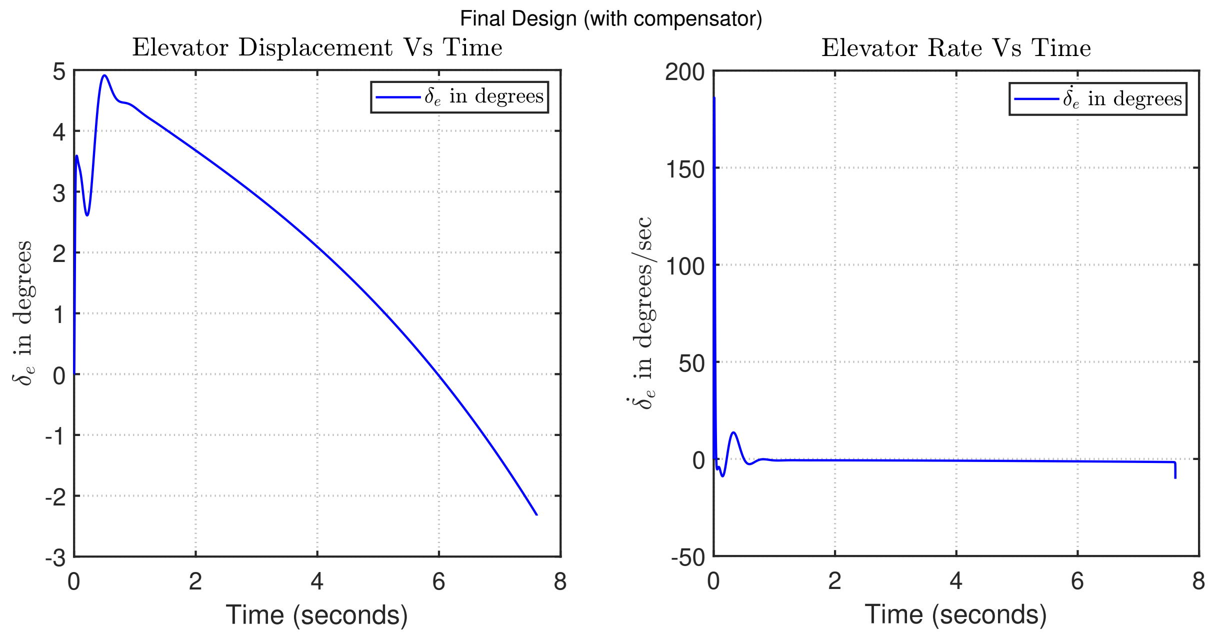
Observer Design -Nyquist Plot of $L_{u_{Comp}}$
$L_{u_{Comp}}$ for positive ω
$L_{u_{Comp}}$ for negative ω
Return-Difference-Vector
Critical Point
Return-Difference-Disk
Imaginary Axis
Real Axis

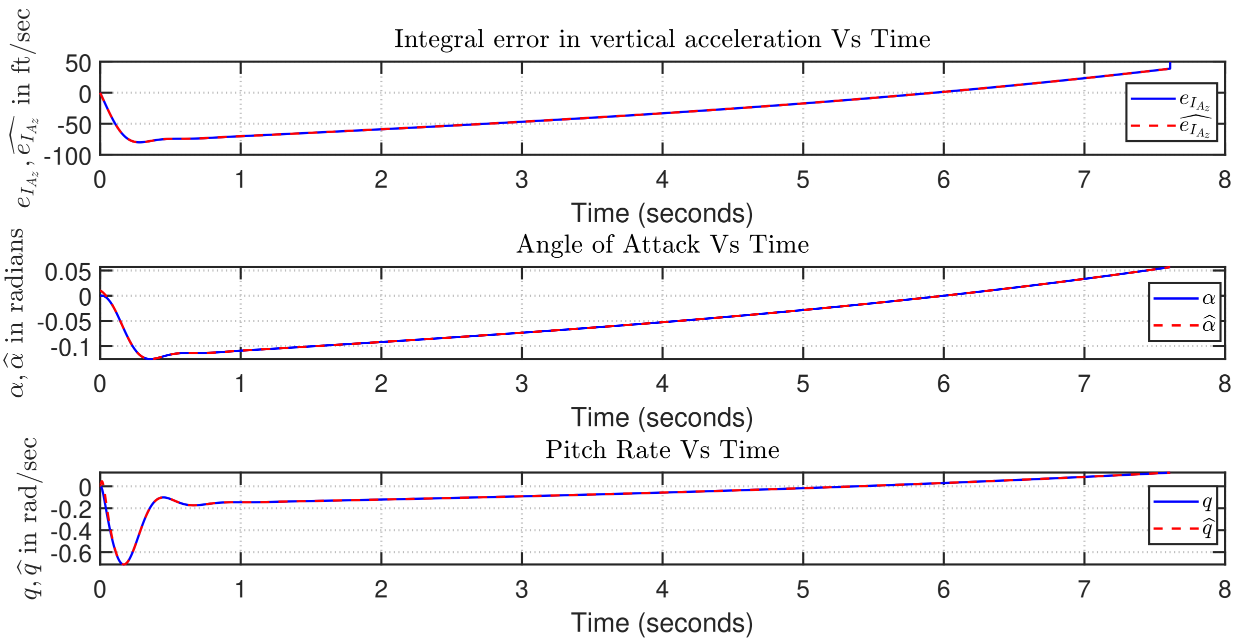


Final Design (with compensator)
Missile and Drone Paths
×10^4
4.2
4.15
4.1
4.05
4
3.95
3.9
3.85
Crossrange [ft]
Missile Path
Drone Path
0
0.5
1
1.5
2
Downrange [ft]
×10^4



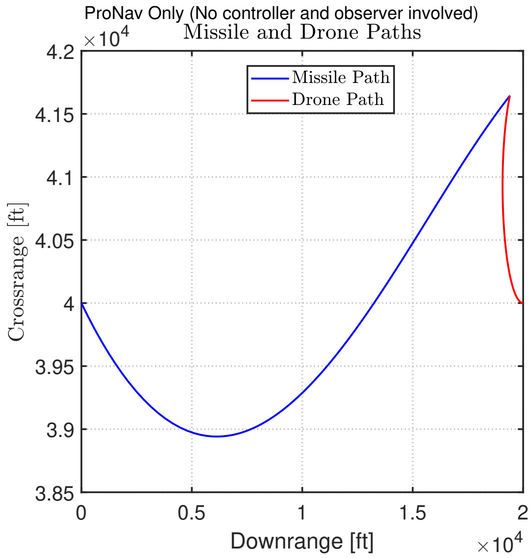
Final Design (with compensator)
Tracking of $A_{z_{cmd}}$ by Missile's $A_{z_{actual}}$
20
15
10
5
0
-5
-10
$A_{z_{cmd}}, A_{z_{actual}}$ (in G's)
0
2
4
6
8
10
Time (seconds)

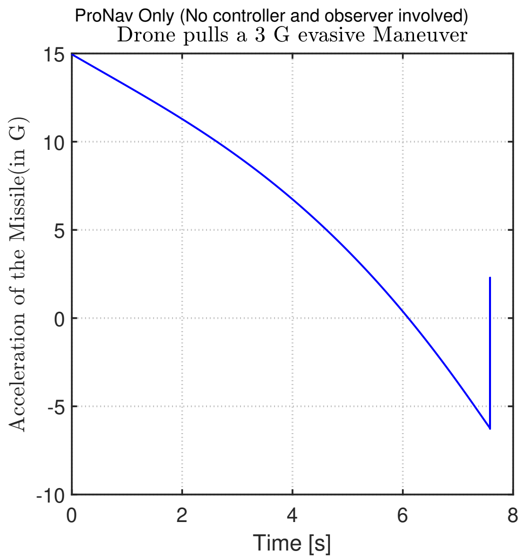

Final Design (with compensator)
Elevator Displacement Vs Time
δ_e in degrees
Time (seconds)
Elevator Rate Vs Time
$\dot{\delta}_e$ in degrees
$\dot{\delta}_e$ in degrees/sec
Time (seconds)


Integral error in vertical acceleration Vs Time
$e_{I_{A_z}}, \widehat{e_{I_{A_z}}}$ in ft/sec
$e_{I_{A_z}}$
$\widehat{e_{I_{A_z}}}$
Time (seconds)
Angle of Attack Vs Time
$\alpha, \widehat{\alpha}$ in radians
α
$\widehat{\alpha}$
Time (seconds)
Pitch Rate Vs Time
$q, \widehat{q}$ in rad/sec
q
$\widehat{q}$
Time (seconds)

Part 2- Guidance

Guidance Only: Without the Autopilot : Without the closed loop compensator in the homing loop, the miss distance is actually zero. However, Matlab's solver will break as range goes to zero because of reasons previously mentioned.





To summarize:

The results achieved for the closed-loop compensator included in the homing loop are as follows:
Percentage Overshoot = 8.9634
Percentage Undershoot = 1.8708
Gain Margin at Plant Input = 20.9888 dB
Phase Margin at Plant Input = 49.4963 degrees
Maximum Elevator deflection = 4.9117 degrees
Maximum Elevator Rate = 186.0951 degrees per second
Miss Distance = 0.0353 ft

A few other design specifications can be seen from the design charts.