Course Project

Mustafa Poonawala (6109-7936)
December 11, 2020

1 SUMMARY

In this project an output feedback system to track the vertical acceleration of an MRAAM is designed and used in a True proportional navigation guidance law to intercept a target drone with minimum miss distance. The missile's short period dynamics are used for this purpose. Robust servomechanism theory and Luenberger observer design is used in the design. The project is divided in 2 parts, namely A and B. Part A consists of designing the closed loop system with the design requirements mentioned, using full state feedback first, and then using output feedback with Acceleration and pitch rate as the outputs.

In part B, the designed closed loop system is then incorporated in the guidance law and minimum miss distance is evaluated. At this time changes in design are made if required. To aid in the design process, design charts are created at all significant points.

2 PART A

- Open loop plant:

  The open loop plant's A and B matrices are given in the problem statement. The states of the open loop plant are $[\alpha, \beta, p, q, r]$ and the inputs are the control surface deflections $[\delta_a, \delta_e, \delta_r]$.

Following are the open loop matrices provided.

$$A = \begin{bmatrix} -1.57 & 0 & 0 & 1 & 0 \\ 0 & -0.5 & 0.17 & 0 & -1 \\ -21.13 & -2876.7 & -2.1 & -0.14 & -0.05 \\ -82.92 & -11.22 & -0.01 & -0.57 & 0 \\ -0.19 & -11.86 & -0.01 & 0 & -0.57 \end{bmatrix} \tag{2.1}$$

$$B = \begin{bmatrix} 0 & -0.1 & 0 \\ -0.07 & 0 & 0.11 \\ -1234.7 & -30.49 & -1803.2 \\ -4.82 & -119.65 & -7 \\ 14.84 & 0.27 & -150.58 \end{bmatrix} \tag{2.2}$$

We look at the open loop plant eigen values. We know from theory that with $[\delta_a, \delta_e, \delta_r]$ as the inputs and $[\alpha, \beta, p, q, r]$ as the states we are going to see the short period, roll subsidence and dutch roll modes only. With the thrust velocity and thrust inputs not in the picture, we won't be seeing the Phugoid modes. Furthermore, the dutch roll and short period are oscillatory in nature so we look at the eigen values and we can see 2 complex conjugate pairs. The 5th eigen value is the roll subsidence. The dutch roll, short period and roll subsidence modes respectively:
$-1.2895 + 21.8316i$
$-1.2895 - 21.8316i$

$-1.0695 + 9.0870i$
$-1.0695 - 9.0870i$

$-0.5920 + 0.0000i$

- The short period: These modes depicts the oscillatory nature of α and q. It is excited by the elevator input δ_e
- The roll subsidence: This is the dominant mode for lateral dynamics since it is the slowest one. It is generally stable critically damped.
- The dutch roll: These modes are the ones depicting the oscillatory nature of the roll and yaw motion. (p, q)

Upon extracting the states for short period modes with respect to δ_e we get the short period dynamics. And the approximate short period modes are: $-1.0700+9.0923i, -1.0700-9.0923i$. This is close to the values of the open loop short period modes listed above, and thus we use them for our design.

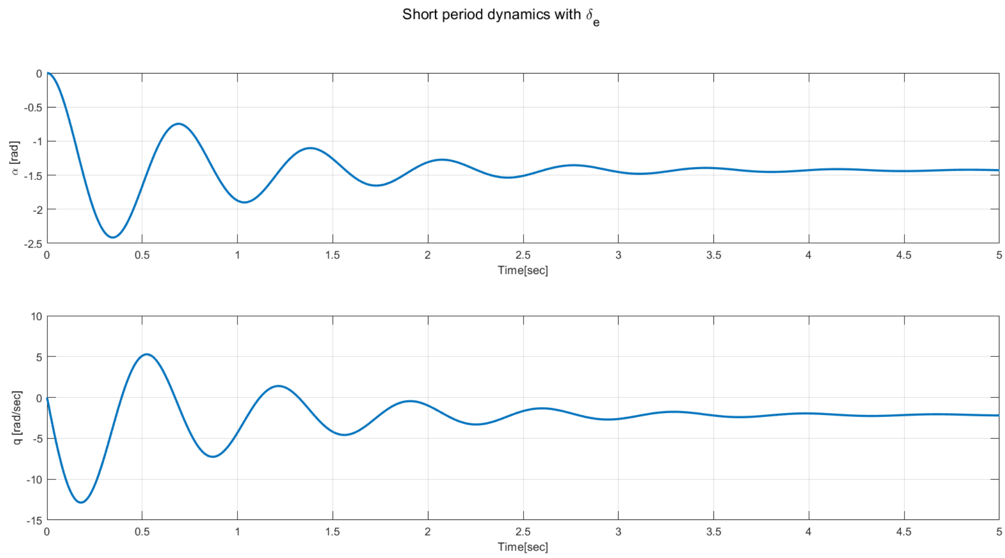

Figure 2.1: Actual Short Period Dynamics

3 Full state feedback design:

We use Robust servomechanism design for designing a full state feedback controller, to track a step reference. Since we do not have Actuator state feedback, we do not include them in our design model.
Following is the design model formulation:

$$\begin{bmatrix} \dot{e} \\ \dot{\xi} \end{bmatrix} = \begin{bmatrix} 0 & C_{Az} \\ 0 & A \end{bmatrix} \begin{bmatrix} e \\ \xi \end{bmatrix} + \begin{bmatrix} D_{Az} \\ B \end{bmatrix} \mu \tag{3.1}$$

Where ξ is $\dot{x}$, μ is $\dot{u}$ and e is $Az - Az_{reg}$
Comparing the above equation with:

$$\dot{z} = \tilde{A}z + \tilde{B}\mu \tag{3.2}$$

We use the $\tilde{A}$ and $\tilde{B}$ matrices above in matlab's "lqr" solver and get the gains. Those gains regulate the above plant i.e (**??**). We use those gains to formulate our control law as follows.

$$u = -K_{ei} \int (Az - Az_{reg}) - K_\alpha \alpha - K_q q \tag{3.3}$$

Once we have our gains we go on to building the analysis model which includes the actuator dynamics and create the closed loop model to track the reference. Since we have not accounted for the actuator dynamics in calculating the gains, we will depart from ideal LQR properties, but will still be able to track reference asymptotically due to the integrator.

As we have done earlier, we must create design charts of our design parameters, by going over a bunch of penalty values on the e_I term of our design model. This not only helps in deciding the design point, but also allows us to develop more intuition about the effects of penalties.

Following is an evaluation of the design requirements and the effect of e_I penalty in satisfying each of them. As a general outline, we can say that increasing the penalty will give larger LQR gains which will give quicker closed loop response, but this will cost us robustness. So there is a tradeoff in robustness and time domain response quality.

1. 6dB Gain margin and 35degree phase margin: Increasing the gain e_I penalty will reduce the gain margin on account of loss of robustness as the delay of the Actuator takes more and more effect as we increase the speed of the controller. So this requirement puts a limit on the penalty we can impose.

2. Maximum fin displacement of 35degrees and maximum fin displacement rate of 350deg/sec : These requirements again puts a limit on how much penalty we can impose, as higher penalty results in more aggressive control.

3. Loop gain at input crossover frequency less than 1/3 the actuator natural frequency: This requirement basically defines a limit on the speed of the controller relative to the speed of the Actuator, thus, again putting a limit on the penalty value. (The actuator bandwidth turns out to be 1/3rd of its natural frequency.)

4. % Undershoot and % Overshoot: % Undershoot increases monotonically with controller aggressiveness. % Overshoot increases to a point and then decreases to 0 and then again increases as we approach instability.

Based on the design charts and the relevant design requirements, we choose a design point that is much better than the required as we would loose some of those properties while going from full state feedback to output feedback. Though, we might have an intuition of the trends, quantifying them is not simple, and thus certain amount of trial and error is necessary. That is why design charts are useful.

3.1 Design summary

The impression we get from the design requirements is that there is no strict time domain requirement such as rise time or settling time. This gives the idea that an aggressive controller is not required. Furthermore the requirement of " Compensator loop gain crossover frequency within 0.25Hz of the LQR loop gain crossover frequency" suggests that the observer has to be quite fast relative to the LQR controller. But there is a limit on the speed of the observer imposed by the requirement that real part of the closed loop Eigen values should be greater than -500. Therefore, we are inclined to choose a very slow controller and a relatively fast observer. The design charts are as shown below with the design point marked.

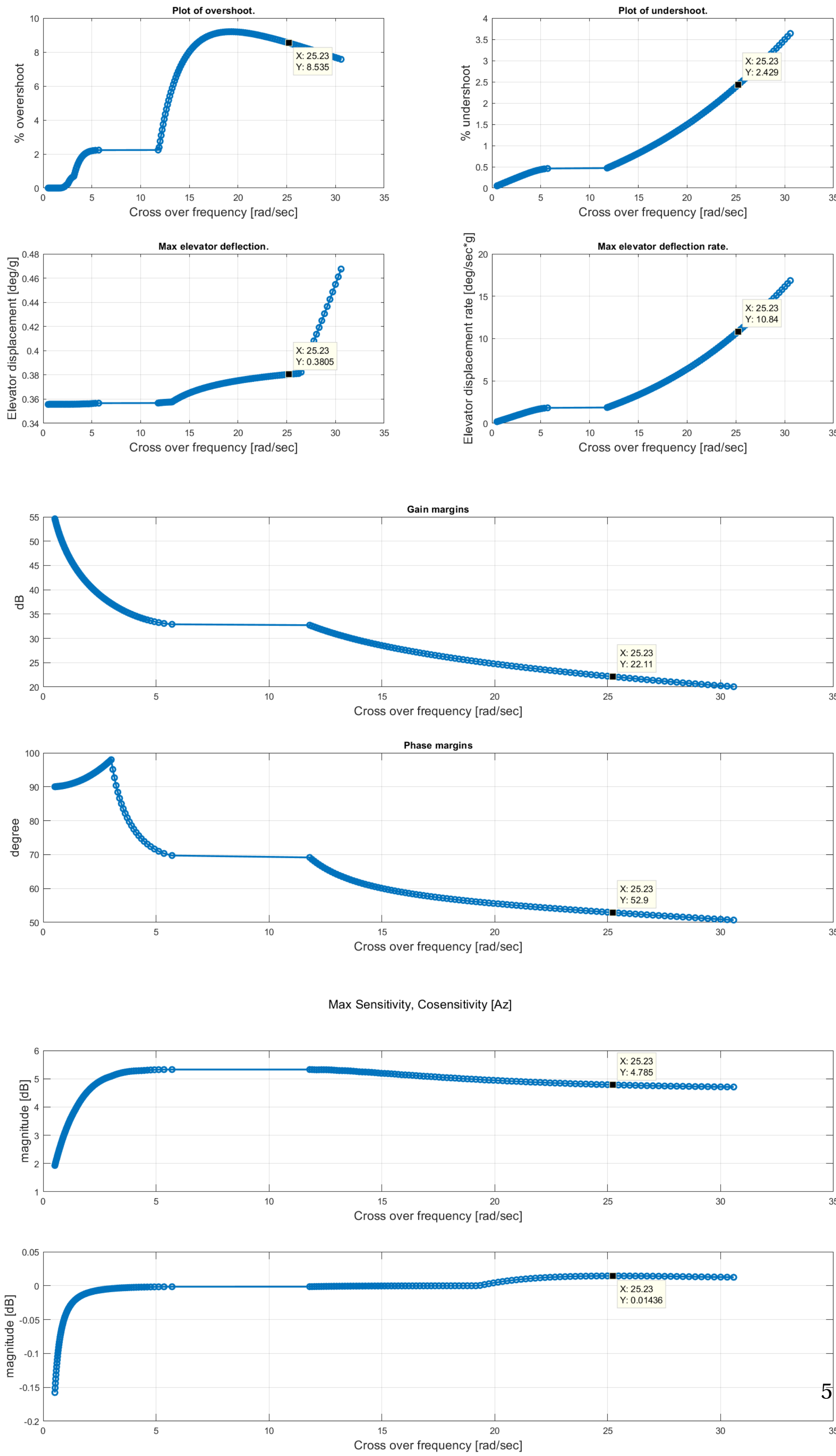

Figure 3.1: Design chart full state feedback

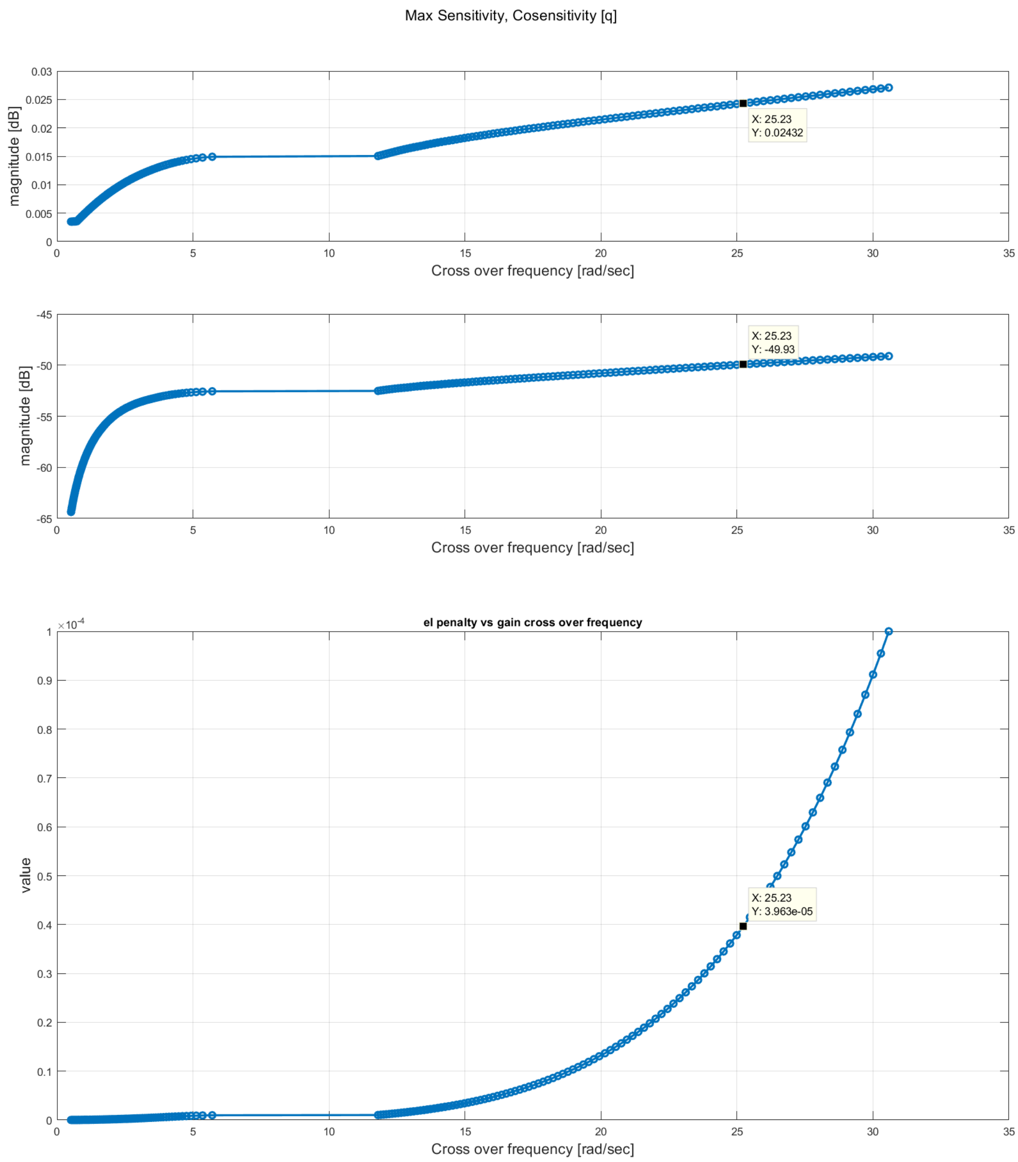

Figure 3.2: Design chart full state feedback

4 DYNAMIC COMPENSATOR WITH OBSERVER:

We use the extended design model dynamics to deisgn an observer:

$$\begin{bmatrix} \dot{e_I}^{Az} \\ \dot{\alpha} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & C_{reg} \\ 0 & A_p \end{bmatrix} \begin{bmatrix} e_I^{Az} \\ \alpha \\ q \end{bmatrix} + \begin{bmatrix} D_{reg} \\ B_p \end{bmatrix} u + \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} r \tag{4.1}$$

$$y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e_I^{Az} \\ \alpha \\ q \end{bmatrix} \tag{4.2}$$

where:

$$\begin{bmatrix} 0 & C_{reg} \\ 0 & A_p \end{bmatrix} = \tilde{A} \tag{4.3}$$

$$\begin{bmatrix} D_{reg} \\ B_p \end{bmatrix} = \tilde{B} \tag{4.4}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \tilde{C} \tag{4.5}$$

Using the property of duality of controllability and observability we use $\tilde{A}^T$ and $\tilde{C}^T$ in the lqr solver in matlab and get the L gains. We again iterate over a bunch of values of penalties W,V and plot design charts.

It should be noted that the large difference at one instant between points is due to a complex pole pair, such that the magnitude "peak" for that complex pole pair barely crosses the 0dB line for those gains, and thus it is not an error. Upon observing the design charts, we can see that the "Compensator loop gain crossover frequency within 0.25Hz of the LQR loop gain crossover frequency" is the deciding factor for the design point, as we meet all other requirements easily. Accordingly a design point is chosen and its charecteristics are listed:

1. Overshoot: 9.1996%
2. Undershoot: 1.5848%
3. Gain margin: 14.4838dB
4. Phase margin: 45.8837 degrees
5. Loop gain cross over frequency: 19.45rad/sec
6. Maximum Sensitivity Cosensitivity [Az,q]
   - Az: [4.7,0.01238]dB
   - q: [0.027,0.01238]dB
7. Difference between compensator and LQR cross over frequencies: 0.2069Hz
8. Most negative eigen compensator value: -344.89

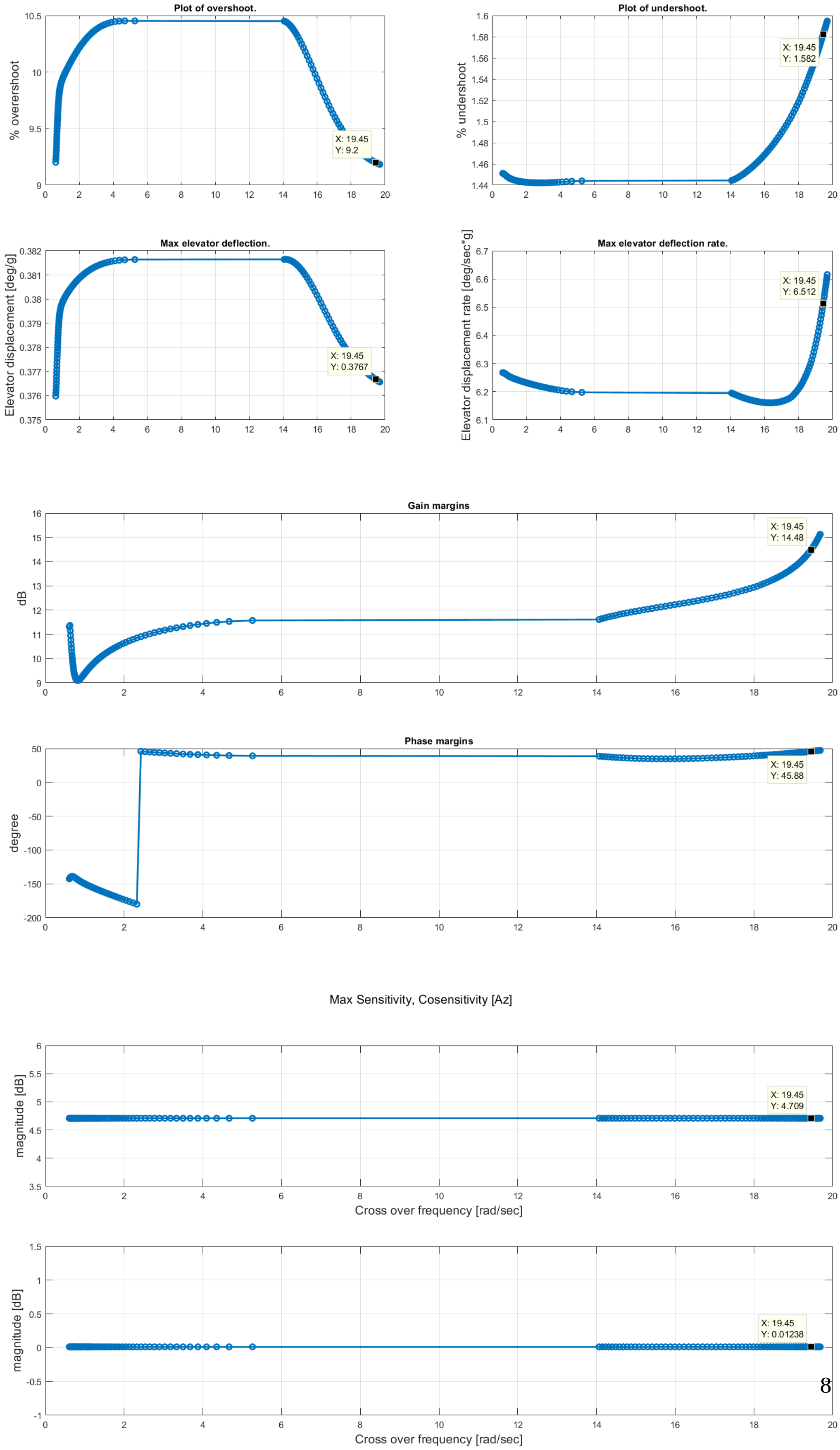

Figure 4.1: Design chart output feedback

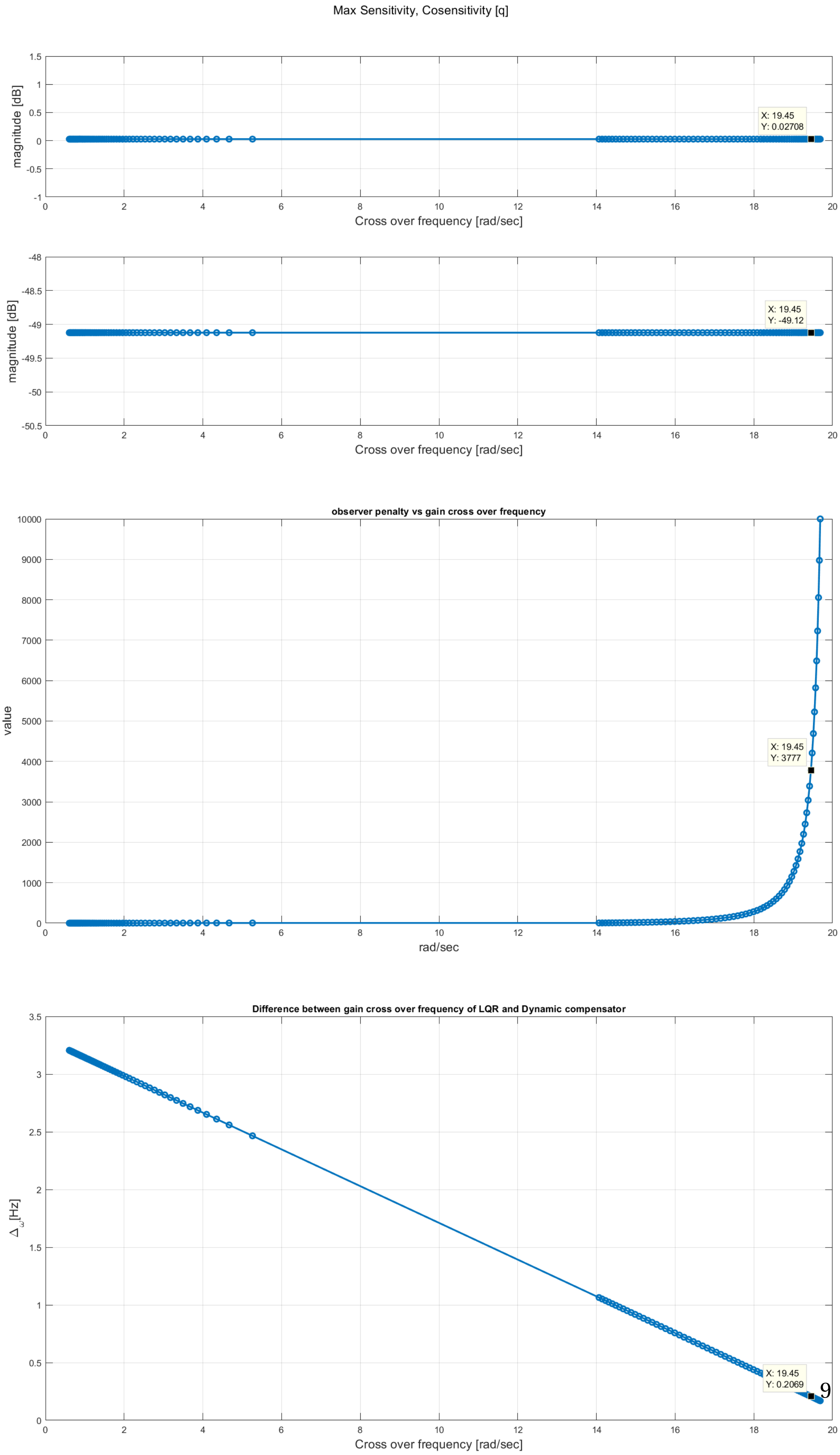

Figure 4.2: Design chart output feedback

5 GUIDANCE (PART B)

Now, that we have our closed loop plant designed, we incorporate it into the guidance script. We set up the initial conditions as shown in the script attached to this report. Upon simulation we use the "a_{true}" command i.e the desired acceleration for collision directly as a reference to our plant. We then use the actual acceleration achieved by the plant for the next step of simulation. We add the closed loop states in the nlinpronav.m function and let the ode solver integrate our plant as well.

An important thing to note is that we use the "a_{true}" directly as reference to our plant. The following 2 points justify it.

1. The model we are using is linearized at level flight, and therefore for small flight path angles the acceleration would be vertical in the inertial frame. Now, it is not garunteed that we are flying level while pursuing the target, but since we are using the same linearized model, so it suggests that the missile's acceleration is vertical in the inertial frame.

2. The line of sight angle initially is 0 degrees since we are on a head on collision track. We also know that true proportional navigation tries to keep the line of sight angle constant, which in our case is 0. Even though due to the maneuvers the line of sight angle is not going to be constant, it does not vary much. It varies maximum 7 degrees in our case, and therefore the "a_{true}" command can be considered vertical.

3. The above 2 points allow us to assume "a_{true}" and plant acceleration are aligned.

We simulate the guidance script after integrating the plant and through a little trial and error reach a suitable navigation constant that does not cause violation of our 20g acceleration limit,35degree elevator displacement limit and 300 degree elevator displacement rate limit.

A small minimum distance of 0.019ft was found but due to singularities in the system, the control commands would blow up, therefore a larger miss distance point was selected. Following are the plots: The actual and estimated states converge very quickly but the plots can be examined through the script.

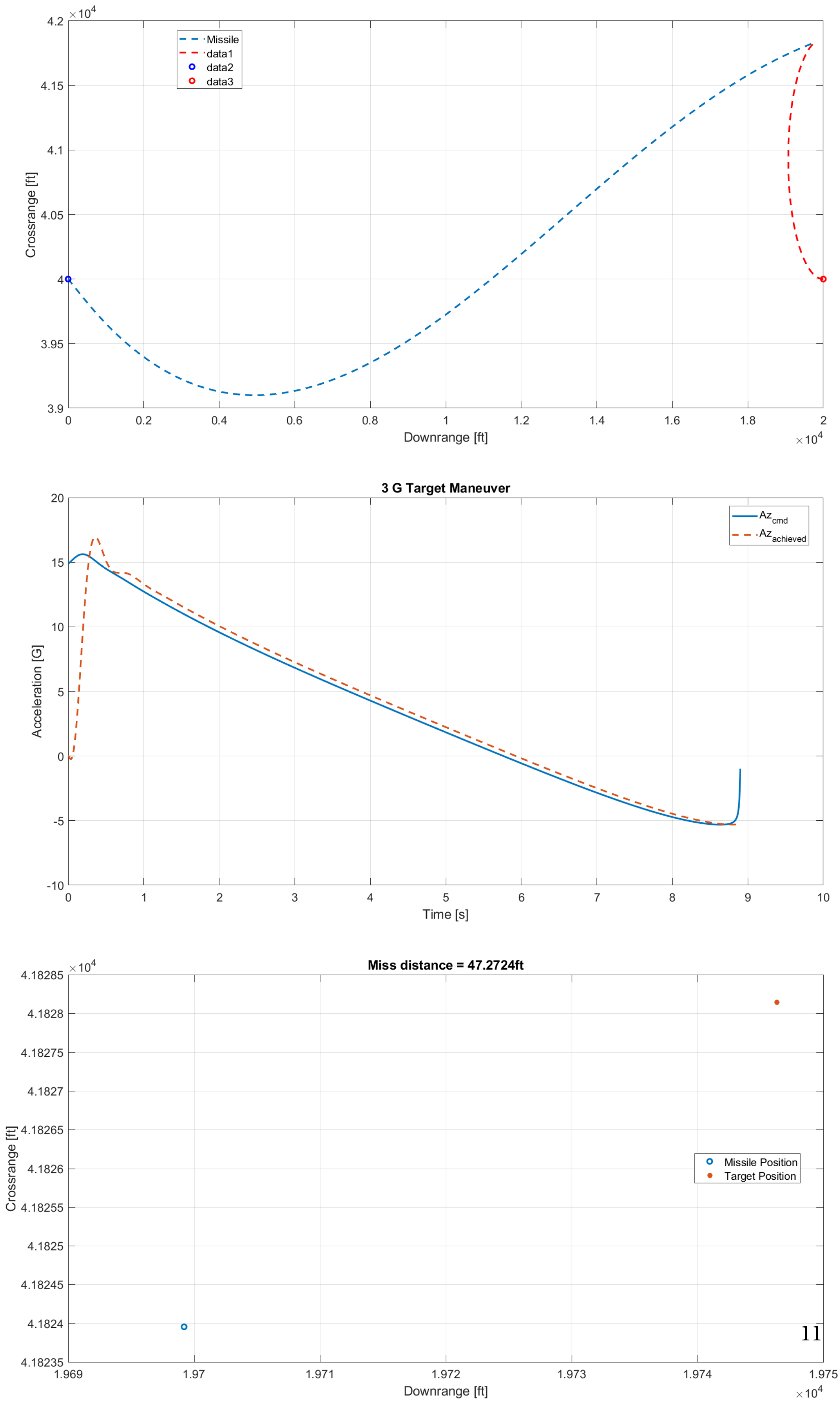

Figure 5.1: Guidance

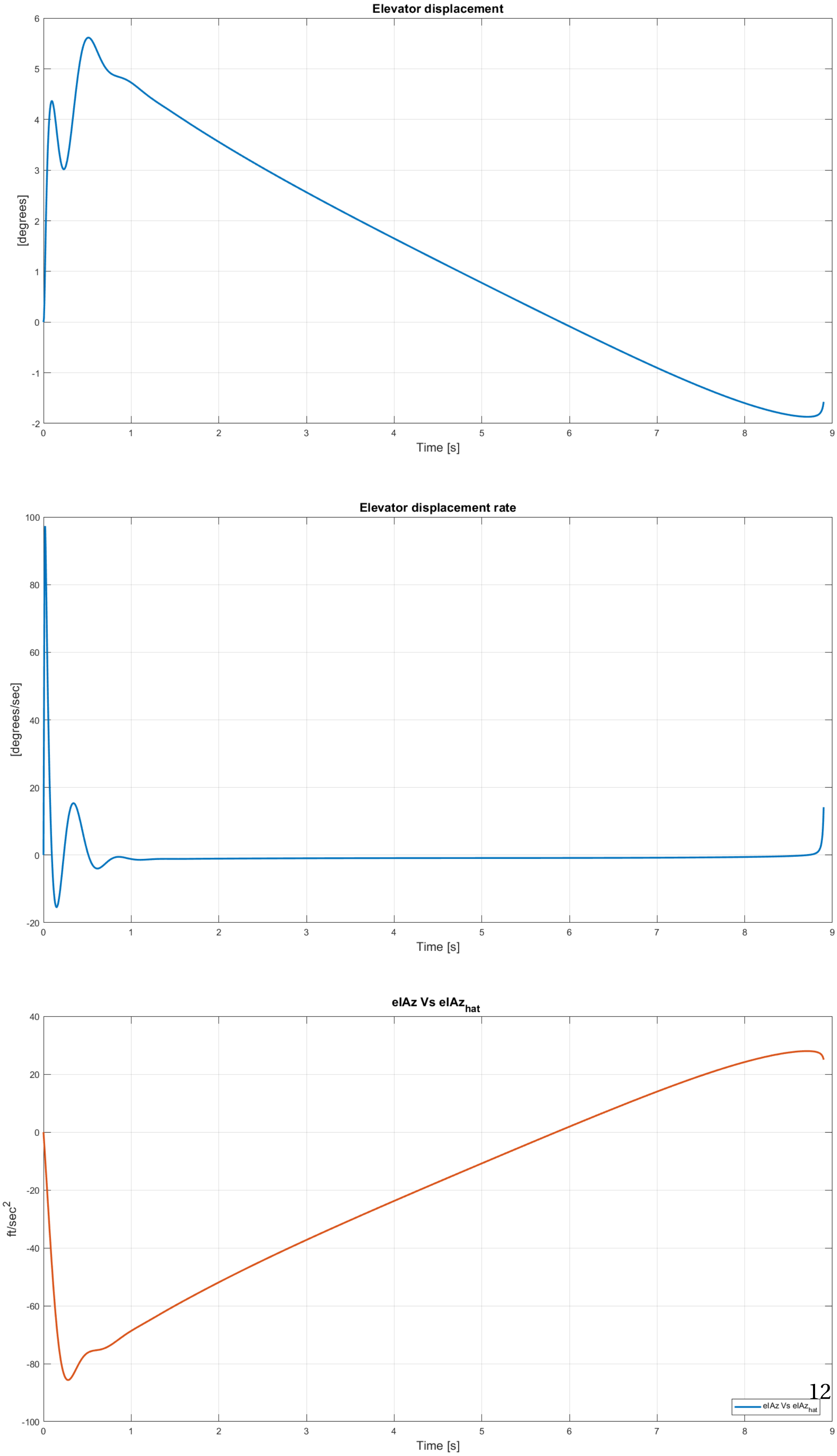

Figure 5.2: Guidance

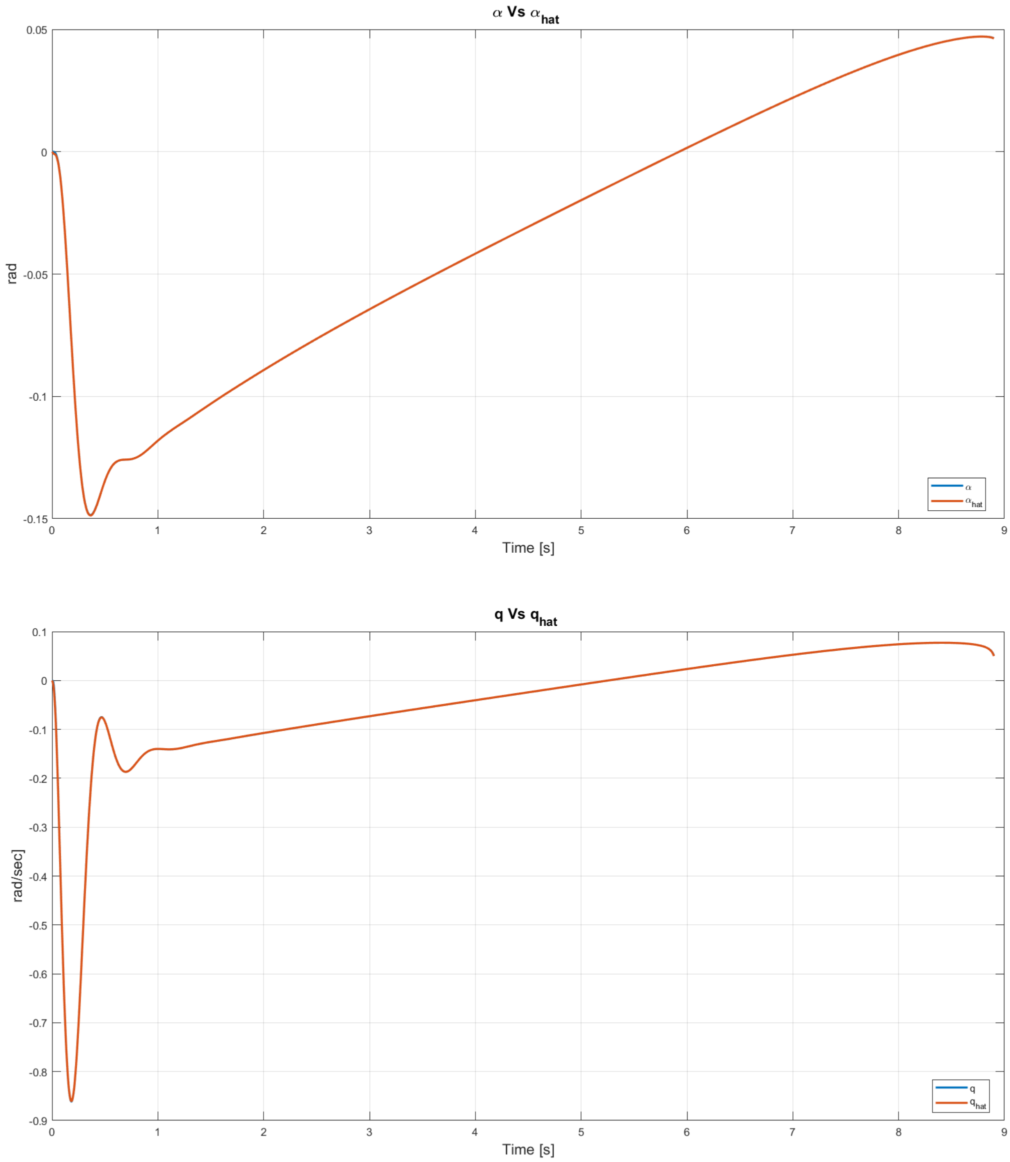

Figure 5.3: Guidance

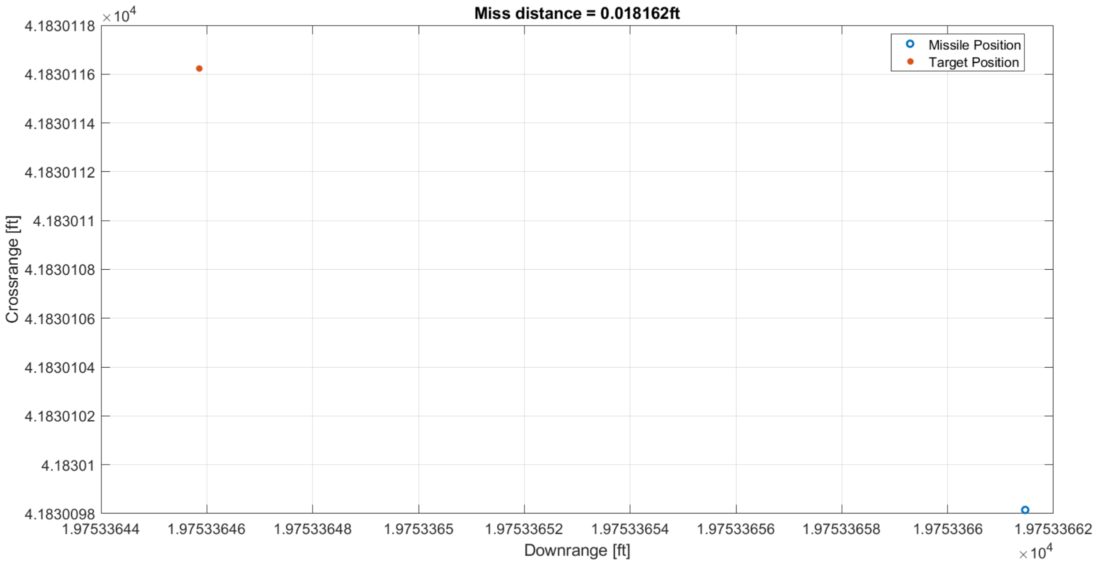

Figure 5.4: Better miss distance