Table of Content

Introduction

Checking the robustness of a transonic missile model with modelling uncertainties in MATLAB/Simulink is the main problem of this assignment. Linearisation can be done by the Taylor expansion of a real or complex function[1]. Linearisation performs about a nominal trajectory. For a nominal input, $\tilde{u}$ the nominal state trajectory is $\tilde{x}_t = f(\tilde{x}_t, \tilde{u}_t, t)$ and nominal output trajectory $\tilde{y}_t$ satisfies with $\tilde{y}_t = g(\tilde{x}_t, \tilde{u}_t, t)$. Expanding nonlinear equations in a multivariate Taylor Series about $(\tilde{x}_t, \tilde{u}_t, t)$ to obtain accurate linear systems by neglecting higher order terms. These steps are followed in MATLAB to identify proper equations to linearise and to have proper equations for systems. In addition, some toolboxes of Simulink such as LTI system box are used.

Part 1: System Modelling

1.0 Flight Dynamics Aspect

Dynamic response of the system depends on the Mach number and altitude. Parameters such as Mach number, altitude and angle of attack are constant in this case. Calculate the size and position of the steering surfaces so that the airframe can perform the anticipated manoeuvres across the flight envelope. In missile application there are two state equations and one output equation for five unknowns at the equilibrium. Parameter that causes a state or output equilibrium point parameterisation and another variable must be imposed. Output or state equilibrium point vectors must be defined[2]. There are three equilibrium analysis applications to be used. Nonlinear systems can be linearised around every equilibrium point by using Jacobean linearisation. Linearised state space models have standard form with 6 terms to calculate. To linearise systems Simulink, numeric and symbolic, and flight envelope methods can be used.

1.1 Parametric Uncertainty

1.1.A Uncertain State Space Model

MATLAB was used to form the uncertain state space model of the given complete open loop dynamics by putting the given variables to matrix forms of actuator, airframe and sensor. State space models were obtained from these matrix forms. Then, state spaces were combined. The same method was followed for nominal and uncertain models. The difference between these systems is uncertain model is found by using uncertain parameters. Graphs, obtained from MATLAB, of nominal and uncertain open loop systems are in the figures given below.

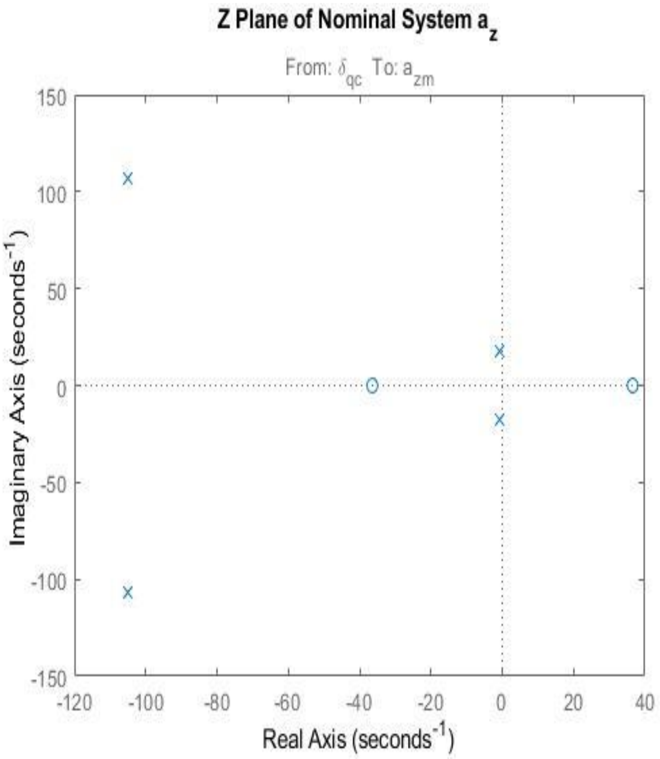

Figure 1 - Poles and Zeros of Nominal system a_{zm}

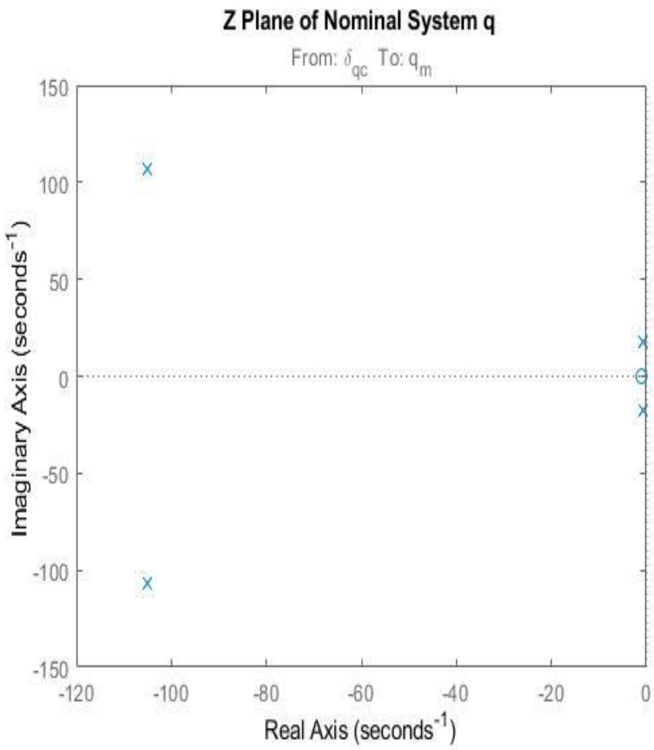

Figure 2 - Poles and Zeros of Nominal System q_m

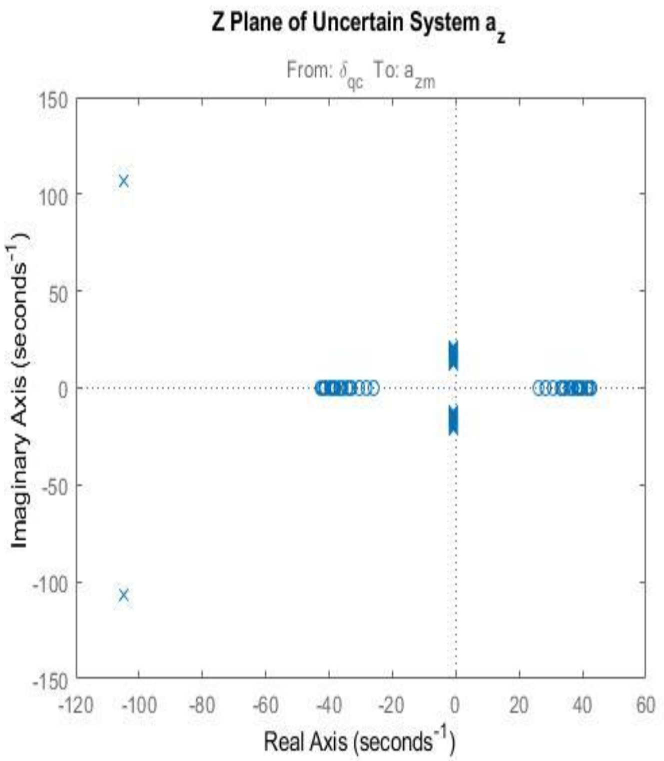

Figure 3 - Poles and Zeros of Uncertain System a_{zm}

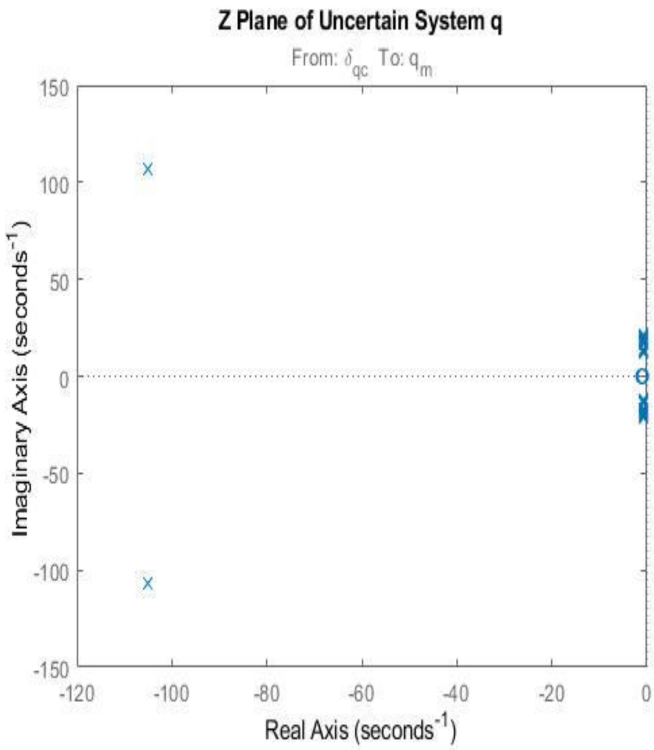

Figure 4 - Poles and Zeros of Uncertain System q_m

From the figures above if poles are considered nominal system that has sensor output $a_{z,\ m}$ has four poles and this means this system is fourth order system and it is the same for nominal system to output q_m because it has four poles likewise. However, they do not have any poles at the origin and this makes them type 0 systems[3]. Furthermore, it can be easily seen both systems are stable because there are no poles in the right-half of Z plane for both cases. It is not easy to calculate order of transfer function of uncertain system for two different cases because it is easy to see poles and zeros coincided more than once and at more than one point. However, it can be said that in both cases uncertain system is stable and it is type 0 system according to reasons for nominal system.

1.1.B LFT Model

Airframe H-Δ LFT was obtained by using MATLAB from the uncertain state space model. "lftdata" command was used to decomposed the uncertainties and it was tuned to LFT on MATLAB. The estimation of expected matrices was made to validate. These matrices compared by using "isequal" command. That means it is successful validation if every "isequal" value is 1, Mathematical verification of codes is below:

$$\begin{bmatrix} \begin{pmatrix}\dot\alpha\\ \dot q\end{pmatrix} \\ \begin{pmatrix} y_{M_\alpha}\\ y_{M_\delta}\end{pmatrix} \\ \begin{pmatrix} n_z\\ q\end{pmatrix}\end{bmatrix} = \begin{bmatrix} \begin{pmatrix} Z_\alpha & 1\\ \bar M_\alpha & M_q\end{pmatrix} & \begin{pmatrix} 0 & 0 & Z_\delta \\ \sqrt{\bar M_\alpha r_{M_\alpha}} & \sqrt{\bar M_\delta r_{M_\delta}} & M_\delta\end{pmatrix} \\ \begin{pmatrix} \sqrt{\bar M_\alpha r_{M_\alpha}} & 0 \\ 0 & 0 \\ A_\alpha & 0 \\ 0 & 1\end{pmatrix} & \begin{pmatrix} 0&0&0\\0&0&\sqrt{\bar M_\delta r_{M_\delta}}\\0&0&A_\delta\\0&0&0\end{pmatrix}\end{bmatrix} \begin{bmatrix} \begin{pmatrix}\alpha\\q\end{pmatrix}\\ \begin{pmatrix}u_{M_\alpha}\\u_{M_\delta}\end{pmatrix}\\ (\delta_m)\end{bmatrix}, \qquad \Delta = \begin{bmatrix}\delta_{M_\alpha} & 0\\ 0 & \delta_{M_\delta}\end{bmatrix}$$

```
C1 = isequal(A_exp, ssLFT.A);
C2 = isequal(B_exp, ssLFT.B);
C3 = isequal(C_exp, ssLFT.C);
C4 = isequal(D_exp, ssLFT.D);
if [C1, C2, C3, C4] == [1, 1, 1, 1]
    fprintf("Succesful Validation")
else
    fprintf("Validation Failed")
end
```

```
ssLFT =

  A =
           alpha        q
   alpha    -1.3        1
   q      -299.3        0

  B =
                 ?          ?  \delta_q
   alpha         0          0   -0.1136
   q         13.15      6.543    -130.9

  C =
         alpha        q
   ?     13.15        0
   ?         0        0
   a_z  -145.7        0
   q         0        1

  D =
               ?          ?  \delta_q
   ?           0          0         0
   ?           0          0     6.543
   a_z         0          0    -11.68
   q           0          0         0

Continuous-time state-space model.

Succesful Validation
```

Figure 5 - Validation of LFT Form

From the figure 5 and mathematical model of LFT shows that LFT model verification is correct that means expected matrices are equal to matrices of H – ΔLFT.

1.2.A Uncertain Inner Loop State Space Model

Negative feedback inner loop gain K_q was calculated to place the transfer function q_c by $a_{z,\,m}$ dominant poles at a damping ratio 0.707 and the loop was scaled by using a gain K_{sc} with place q_{sc} by $a_{z,\,m}$. Zero feedback gain was used on $\delta_{q,\,m}$. Figures obtained from this operation are shown below:

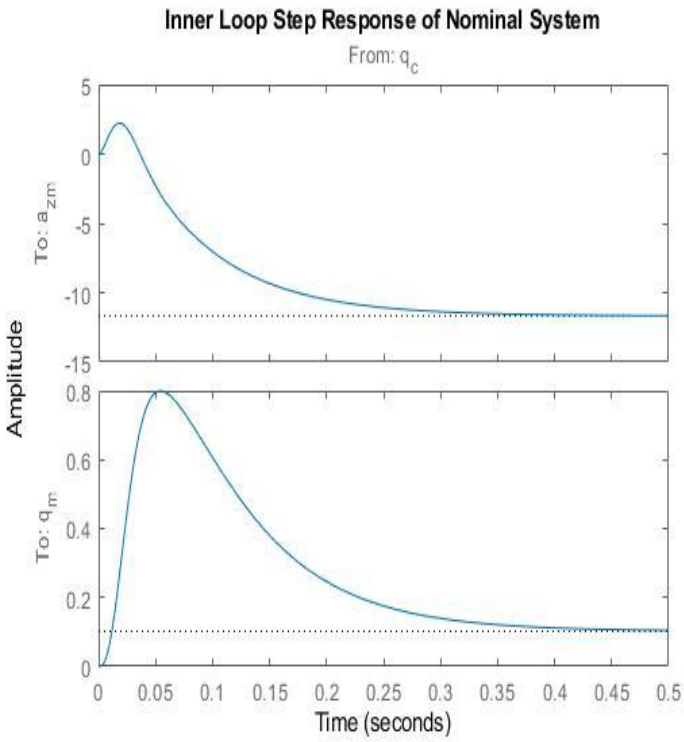

Figure 6 - Inner Loop Step Response of Nominal System

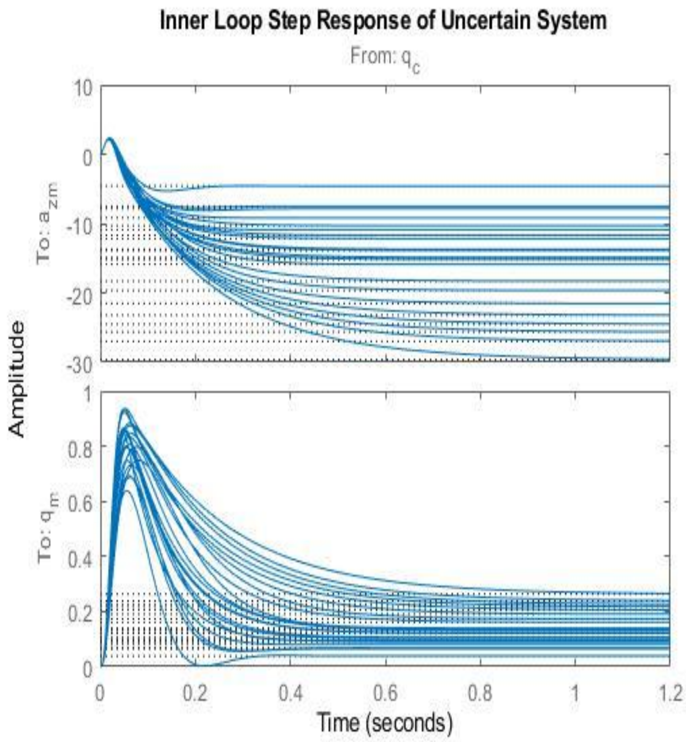

Figure 7 - Inner Loop Step Response of Uncertain System

In four cases (nominal and uncertain, and output of sensor) step response graphs do not have frequent oscillations, they are slightly oscillating in this context it can be said they are relatively smooth. In four graphs, each responses have their own final values according to this it can be understood that they are stable. For more information Root Locus graphs of these cases are shown below:

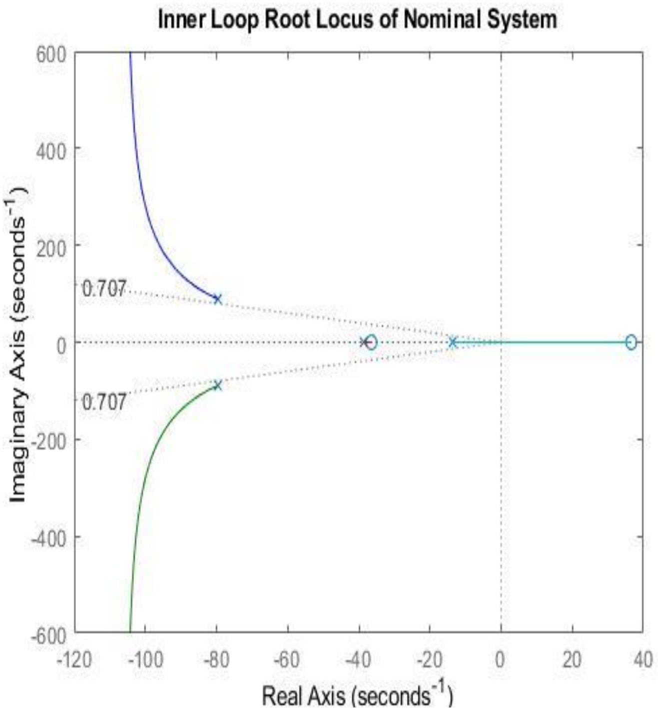

Figure 8 – Inner Loop Root Locus of Nominal System

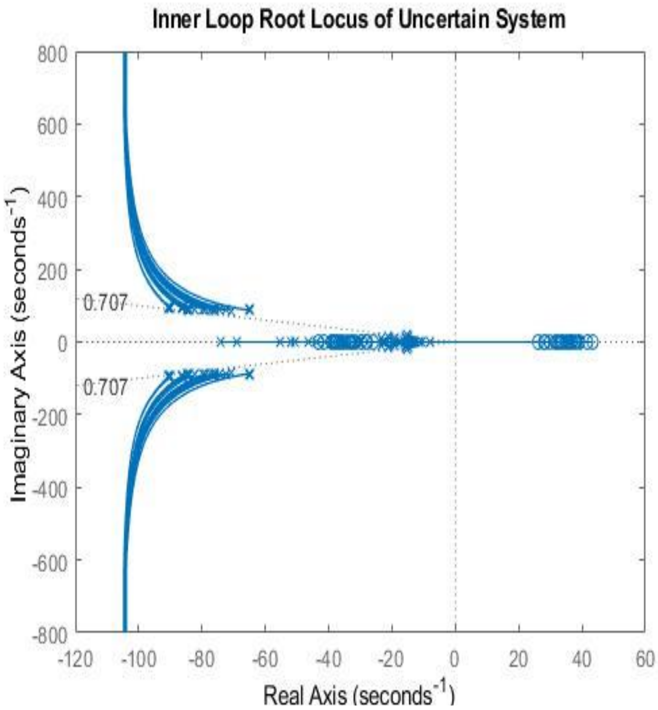

Figure 9 – Inner Loop Root Locus of Uncertain System

The effects the closed loop poles of changing gain values or adding open loop poles and/or zeros on the position can be foreseen by designer by using root locus method. It indicates that which open loop poles and zeros should be improved, so the response coincides the system performance specifications[4]. From the graphs above it can be understood that systems should has no positive real component of poles, so systems are stable for the gain value. Furthermore, there are more than three poles that have imaginary part that means systems should be oscillated by gain.

1.2.B Uncertainty Weight Computation

Multiplicative unstructured uncertainty form was selected to solve sample this uncertain system for a given number of values for uncertain variables and to obtain an array of LTI systems. Singular values graph is shown in the figure below:

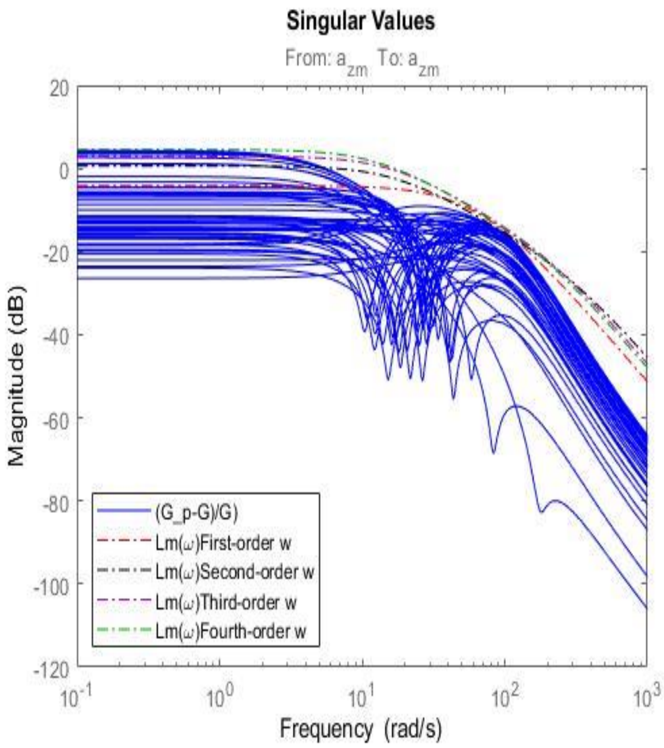

Figure 10 - Singular Values

On the figure above there are some singular values are (almost) zero, then this means singular vectors for these singular values are solutions for the system output in the absence of force at its input. Furthermore, notches on the graph state that there are numerous free modes in this system.

Part 2: Control Design

2 degrees of freedom controller are designed for the purpose of to control normal acceleration $a_{z,\ m}$ via the virtual control signal q_{sc} for inner loop of controlled system in this part of assignment. Inner loop controlled system is defined as $G_{in}(s)$, disturbance rejection controller defined as $K_{dr}(s)$ and $K_{cf}(s)$ denotes the feedforward command following controller.

2.A Preliminary Step

The value of the inner closed loop transfer function with four poles and two zeros designed and obtained in part 1.2.A was verified using MATLAB. The step response of this system is the first graph in shown figure 11.

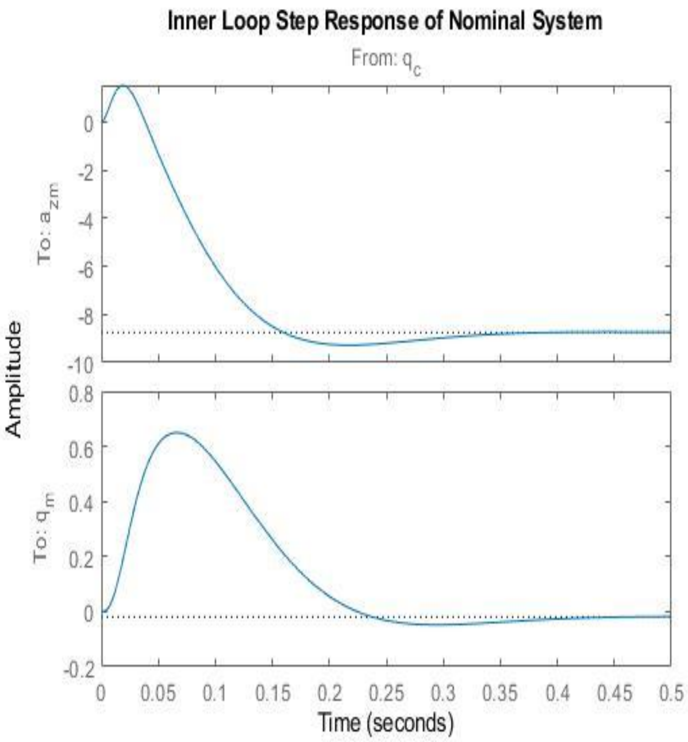

Figure 11 - Inner Loop Step Response of Nominal System

The settling time is almost 0.4 seconds for 2% overshoot. Non-minimum real phase zeros lead to undershoot step response. This is because the size of the undershoot necessarily tends towards the minus infinity as the settling time approaches to zero. In this case it can be seen that non-minimum phase zeros lead to step response undershoot. The desired rotation matrix for the inner loop is generated by the outer loop[5]. Thus, it controls the inner loop and this is the main reason of usage of the outer loop.

2.B Reference Model Computation

To have final closed loop transfer function performs like a second order system is desired aim of the assignment. Necessary functions and calculations were used to find this final transfer function. Obtained transfer function and step information, and step response of this system are shown in the figures 12 and 13 respectively.

```
T_rm =

   -11.03 s + 411.5
  ------------------
  s^2 + 32 s + 411.5

Continuous-time transfer function.


step_information =

  struct with fields:

         RiseTime: 0.1121
    TransientTime: 0.1993
     SettlingTime: 0.2005
      SettlingMin: 0.9031
      SettlingMax: 1.0194
        Overshoot: 1.9414
       Undershoot: 9.2404
             Peak: 1.0194
         PeakTime: 0.2706
```

Figure 12 - Transfer Function and Step Information

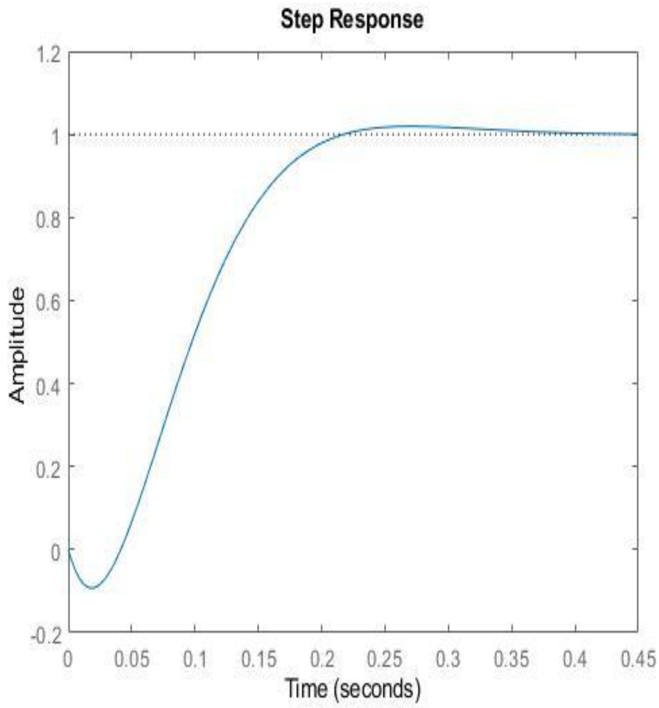

Figure 13 - Step Response

2.C Weighting Filter Selection

S_O, K_{SO} and T_o should be chosen as singular-valued rotational, stable and minimum phase approximations of the inverse closed loop transfer functions. This also targets closed loop sensitivity function. The control times sensitivity constraint KS_t cannot be calculated analytically. First order approximations of S_t and T_c can be used. At the final step, the resulting open loop system is calculated[6]. Singular value graph of the weight filter is shown in figure 13. It can be easily seen that only two singular vectors of singular values can be used as solutions for the system output in the absence of force at its input because only two singular values start from (near) zero.

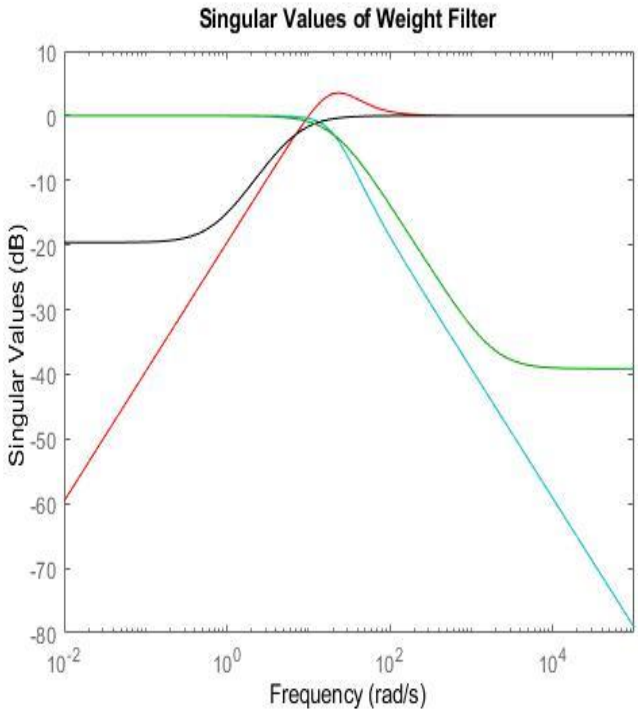

Figure 14 - Singular Values of Weight Filter

2.D Synthesis Block Diagram

Desired reference model transfer function, the inner closed loop, two interface gain matrices and weighting filter matrix were found by using MATLAB. In this part, the main aim is computing the transfer function of $\widetilde{P}(s)$ (orange box) and the weighted transfer matrix P(s) (orange + green box) using MATLAB. Transfer function of $\widetilde{P}(s)$, P(s), and poles and zeros for every case are in the figures given below.

```
tf_full =

  From input "r" to output...
           -11.02 s^2 + 335.1 s + 2836
   z_1:  --------------------------------
        s^3 + 32.72 s^2 + 434.5 s + 296.5

   z_2:  0

   v_1:  0

   v_2:  1

  From input "d" to output...
        -0.9992 s - 6.893
   z_1:  -----------------
           s + 0.7206

   z_2:  0

   v_1:  1

   v_2:  0

  From input "u" to output...

                   4930 s^3 + 2.7e04 s^2 - 6.65e06 s - 4.554e07

   z_1:  -----------------------------------------------------------------

        s^5 + 210.7 s^4 + 2.295e04 s^3 + 5.651e05 s^2 + 7.002e06 s

                                                             + 4.761e06


        90.63 s + 1885
   z_2:  --------------
           s + 1885

                    -4934 s^2 + 7022 s + 6.607e06
   v_1:  ---------------------------------------------------
        s^4 + 210 s^3 + 2.28e04 s^2 + 5.487e05 s + 6.607e06

   v_2:  0

Continuous-time transfer function.
```

Figure 15 - Transfer Function of Orange Box

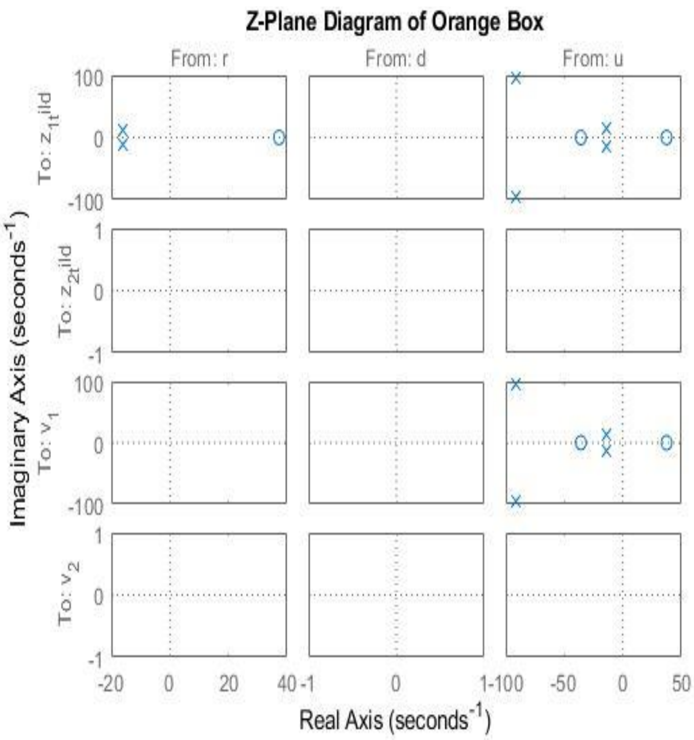

Figure 16 - Poles and Zeros of Orange Box

```
tf_tilda =

  From input "r" to output...
              -11.03 s + 411.5
   z_1_tild:  -----------------
             s^2 + 32 s + 411.5

   z_2_tild:  0

   v_1:  0

   v_2:  1

  From input "d" to output...
   z_1_tild:  -1

   z_2_tild:  0

   v_1:  1

   v_2:  0

  From input "u" to output...
                       4934 s^2 - 7022 s - 6.607e06
   z_1_tild:  ---------------------------------------------------
              s^4 + 210 s^3 + 2.28e04 s^2 + 5.487e05 s + 6.607e06

   z_2_tild:  1

                   -4934 s^2 + 7022 s + 6.607e06
   v_1:  ---------------------------------------------------
         s^4 + 210 s^3 + 2.28e04 s^2 + 5.487e05 s + 6.607e06

   v_2:  0

Continuous-time transfer function.
```

Figure 17 - Transfer Function of Orange + Green Box

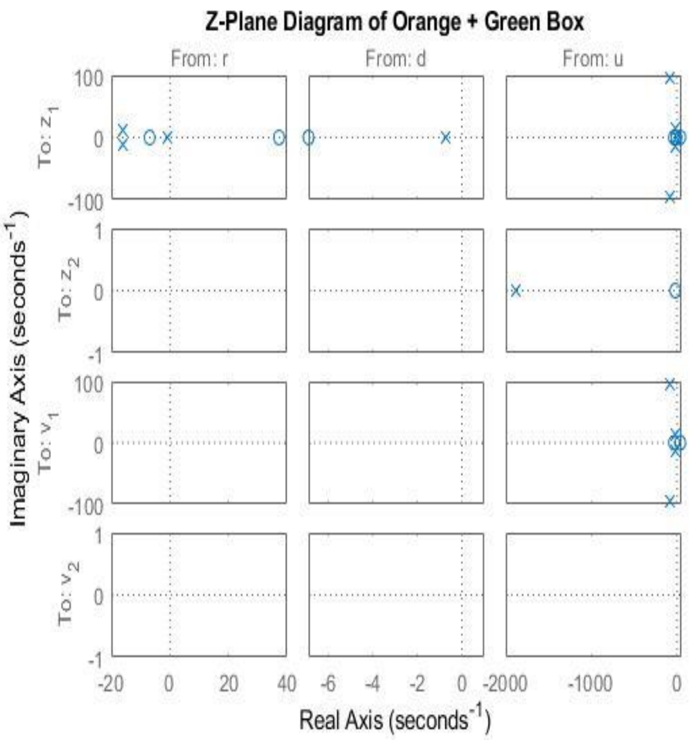

Figure 18 - Poles and Zeros of Orange + Green Box

As it can be understood from the figures above, every system has different transfer functions for every input. However, $\widetilde{P}(s)$ has the same output transfer functions for input u. There are no poles which have positive real part, so both systems are stable.

2.E Controller Synthesis

A full order H_∞ controller $K_{FO}(s)$ for the standard form was designed and K_{dr} and K_{cf} were obtained as taking first and second elements of K_{FO} respectively. The order of the minimal controllers is 6 K_{dr} for and 8 for K_{cf}. They were obtained from transfer functions of K_{dr} and K_{cf} individually by using MATLAB.

```
tf_Dr =

  -0.3441 s^5 - 3963 s^4 - 8.249e05 s^3 - 8.889e07 s^2 - 2.137e09 s
                                                           - 2.57e10
  -------------------------------------------------------------------
  s^6 + 2100 s^5 + 4.289e05 s^4 + 4.559e07 s^3 + 1.253e09 s^2 + 1.656e10 s
                                                           + 1.13e10

Continuous-time transfer function.

tf_Cf =

   3.459 s^7 + 7357 s^6 + 1.68e06 s^5 + 1.988e08 s^4 + 8.805e09 s^3
                            + 1.992e11 s^2 + 2.359e12 s + 1.175e13
  ---------------------------------------------------------------------
  s^8 + 2132 s^7 + 4.965e05 s^6 + 6.018e07 s^5 + 2.888e09 s^4 + 7.542e10 s^3
                            + 1.057e12 s^2 + 7.177e12 s + 4.65e12

Continuous-time transfer function.

order_Dr =

     6

order_Cf =

     8
```

Figure 19 – Transfer Functions and Order of Minimal Controllers

From input r there are two z̃ outputs and similarly from input d there are two z̃ outputs. Thus, there are four singular values graphs in total to compare performance of inputs. For input r, one of outputs has literally no singular values. However, the other output has two singular values one starts from 0 dB and descents and the other one starts from (near) -15 dB, peaks at 0 dB and descents. From this visualisation, there are two singular vectors which can be used during lack of input for output case $\widetilde{z_1}$. On the other hand, there are no singular values and vectors to consider any situation for output case $\widetilde{z_1}$. In second case, there are two outputs for input d similarly to input r. Nevertheless, it has 2 singular values for each output and the pair of each output performs the same performance. One singular value starts from 0 dB and descents until nearly -1 dB and ascents to 0 dB. The other one starts from -20 dB peaks at almost 0 dB and descents. This visualisation is the same as for the both cases of outputs and it can be seen in the figure 19. According to given information about each case it can be said that from input r there are total two singular vectors for both outputs. However, there are total four singular vectors for both cases from input d in the same frequency interval. Thus, the cases of d can be performed by two singular vectors during lack of input for both outputs, but the cases of r cannot do the same because there are no singular values for the second output. Thence, performance of input d is better than performance of input r.

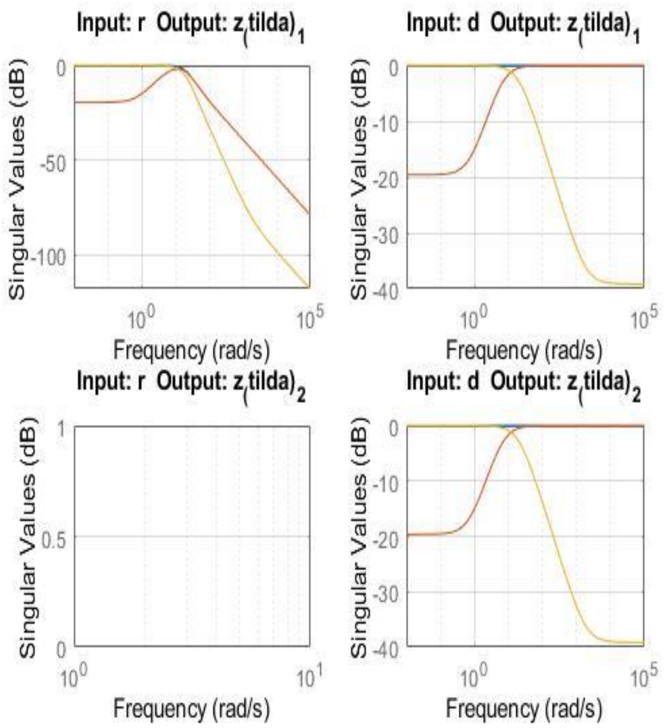

Figure 20 - Singular Values of Input r and d to Outputs $\widetilde{z_1}$ and $\widetilde{z_2}$

2.F Controller Implementation

K_{cf} and K_{dr} were calculated from K_{FO} which is calculated by H_∞ control. K_{cf}' and K_{dr}' were calculated by a function of K_{cf} and K_{dr} and taking care of negative feedback. Their transfer functions, order and reduced order are shown in figure 20 and 21 relatively.

```
K_cf_prime =

  -3.459 s^13 - 1.462e04 s^12 - 1.861e07 s^11 - 7.04e09 s^10 - 1.487e12 s^9

          - 1.898e14 s^8 - 1.549e16 s^7 - 7.688e17 s^6 - 2.446e19 s^5

          - 5.102e20 s^4 - 6.89e21 s^3 - 5.604e22 s^2 - 2.213e23 s

                                                          - 1.328e23

  -------------------------------------------------------------------------

   0.3441 s^13 + 4696 s^12 + 9.443e06 s^11 + 3.835e09 s^10 + 8.406e11 s^9

           + 1.098e14 s^8 + 9.147e15 s^7 + 4.645e17 s^6 + 1.532e19 s^5

           + 3.353e20 s^4 + 4.839e21 s^3 + 4.292e22 s^2 + 1.944e23 s

                                                           + 1.195e23

Continuous-time transfer function.

K_dr_prime =

  -0.3441 s^5 - 3963 s^4 - 8.249e05 s^3 - 8.889e07 s^2 - 2.137e09 s

                                                           - 2.57e10

  ------------------------------------------------------------------------

  s^6 + 2100 s^5 + 4.289e05 s^4 + 4.559e07 s^3 + 1.253e09 s^2 + 1.656e10 s

                                                           + 1.13e10
```

Figure 21 - Transfer Functions of Kcf' and Kdr'

Transfer functions of K_{cf}' and K_{dr}' are quite complicated and contains many of higher order terms.

```
order_K_cf_prime =

    13


order_K_df_prime =

     6


reduced_order_K_cf =

     3


reduced_order_K_dr =

     3
```

Figure 22 - Order and Reduced Order of Kcf’ and Kdr’

Order of K_{cf}’ and K_{dr}’ are reduced to 3 from 13 and 6 respectively. Z plane of full and reduced controllers are shown in the figures below. According to poles and zeros of each, every one of them is stable because none of them has poles have positive real part. They are also type 0 system because none of them has zero at the origin. However, there is an exception, one of zeros of K_{cf}’ is at the origin and this makes this system type 1 system.

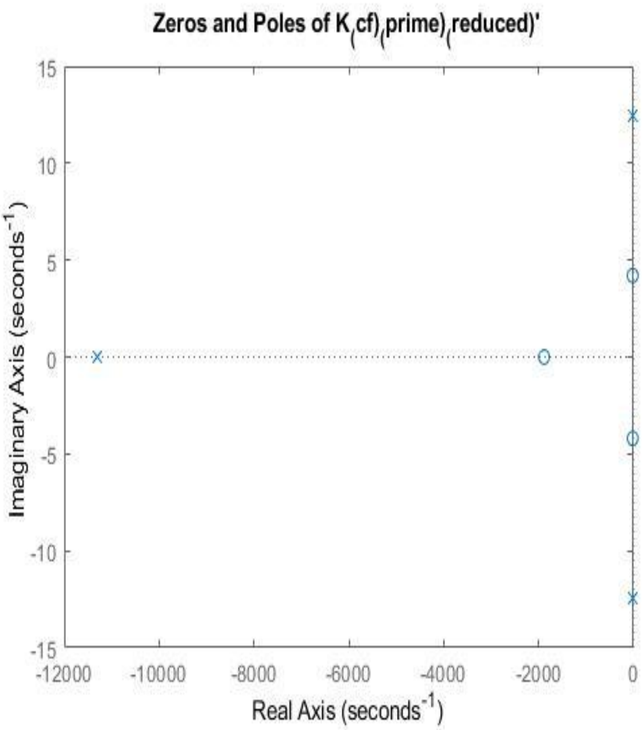

Figure 23 – Zeros and Poles of Kcf’ Reduced

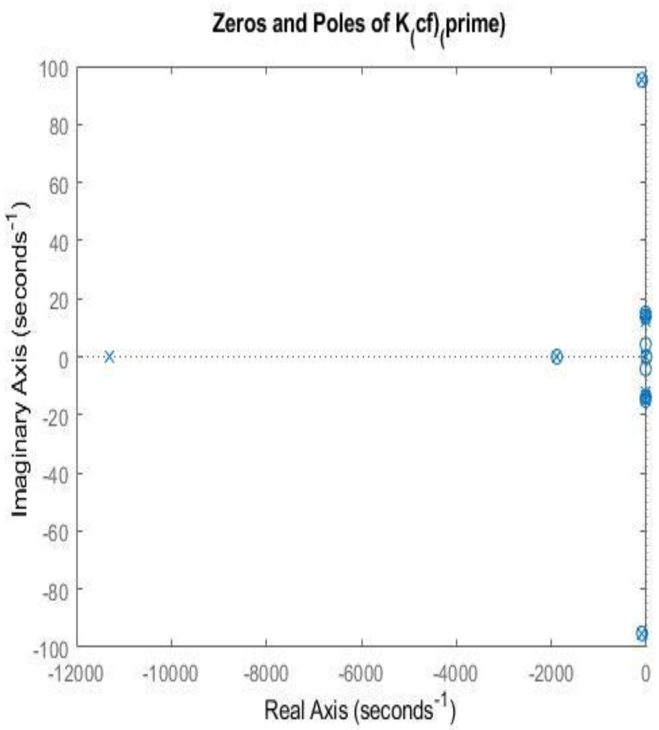

Figure 24 - Zeros and Poles of Kcf’

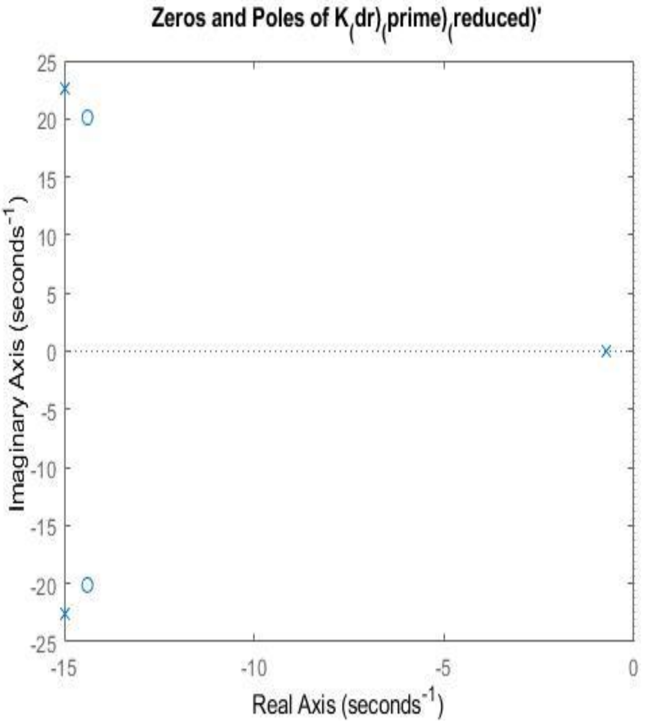

Figure 25 - Zeros and Poles of Kdr' Reduced

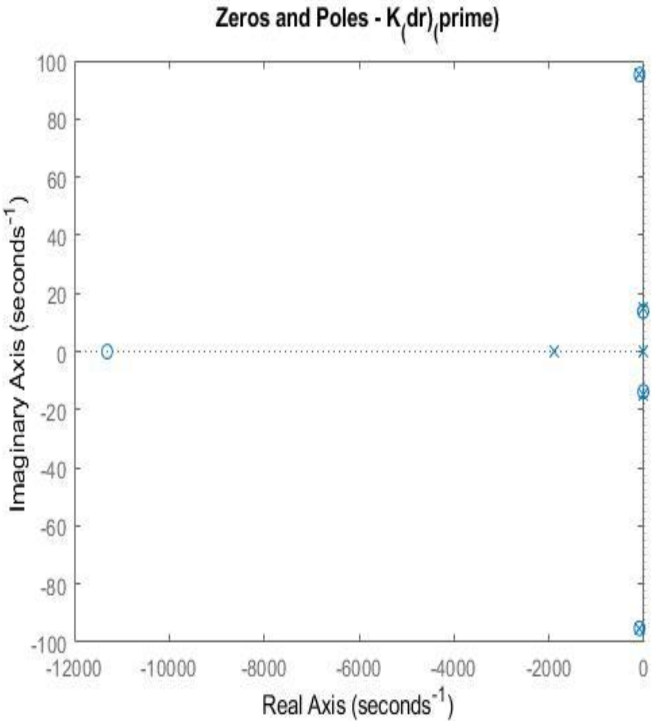

Figure 26 - Zeros and Poles of Kdr'

There are no singular values of reduced K_{cf}' and reduced K_{dr}', but singular values of K_{cf}' and K_{dr}' start from (near) 0 dB. That means singular vectors of these singular values can be used when a lack of input. However, there is no singular values for reduced K_{cf}' and reduced K_{dr}' to mention about characteristic of singular vectors. Thus, performances of full controllers are better than performances of reduced controllers. Singular values graph is shown in figure 26.

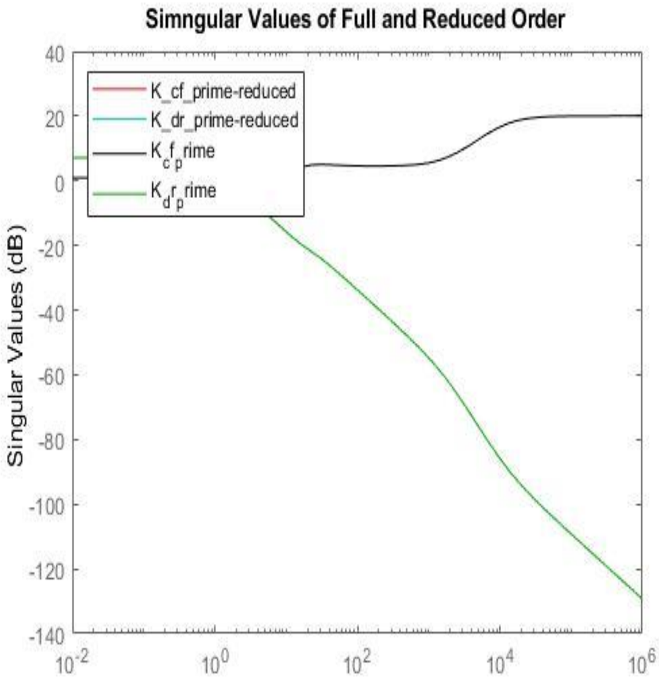

Figure 27 - Singular Values of Full and Reduced Order Controllers

2.G Controller Simulation

A Simulink model was created according to steps mentioned in assignment file. After having this model, model takes some of the parameters from previous parts because of this reason "main.m" file should be run first to have results from this Simulink model. The rest of parameters are determined from main's the assignment file again such as $a_{z,c}$ and $a_{z,d}$. According to these parameters, wanted response graphs are shown below:

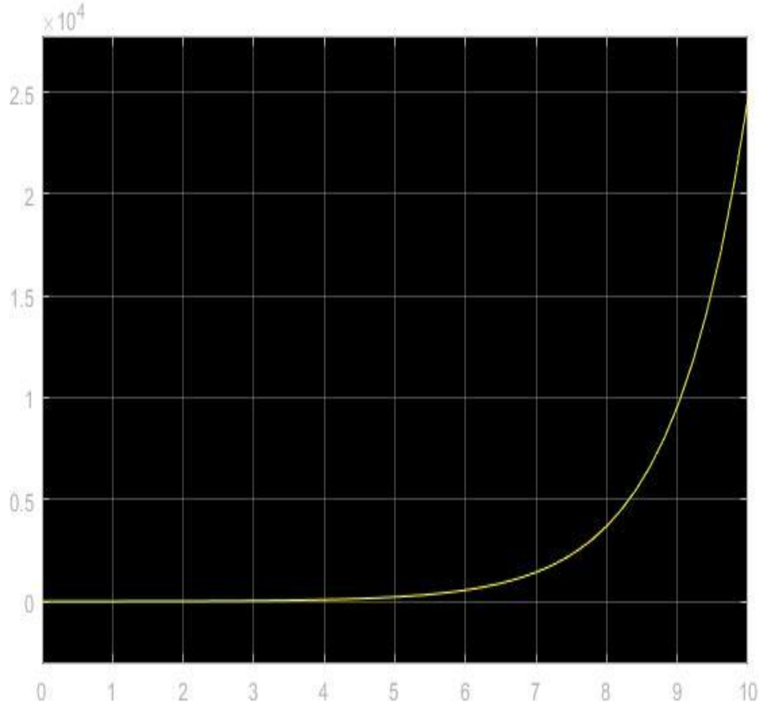

Figure 28 - Acceleration Step $a_{z,m}$

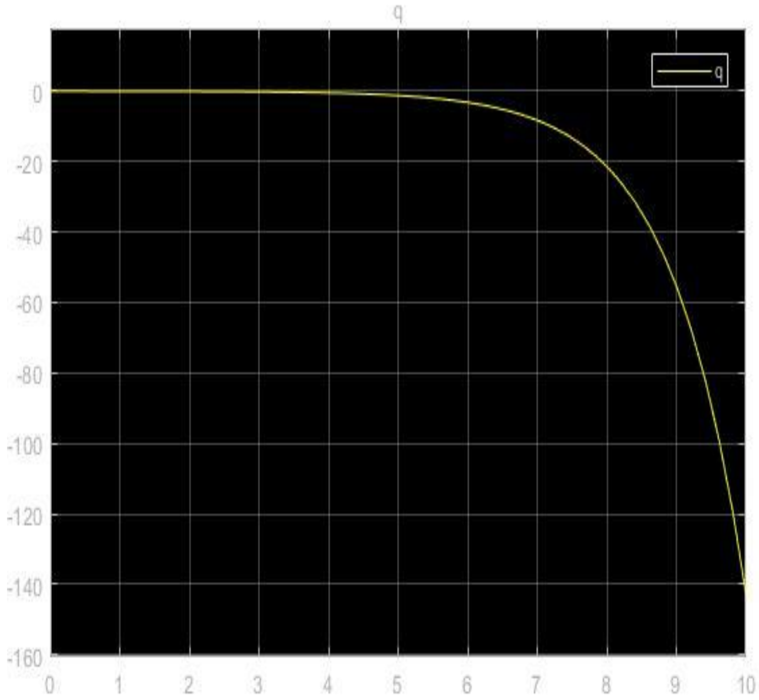

Figure 29 - Pitch Rate Step q

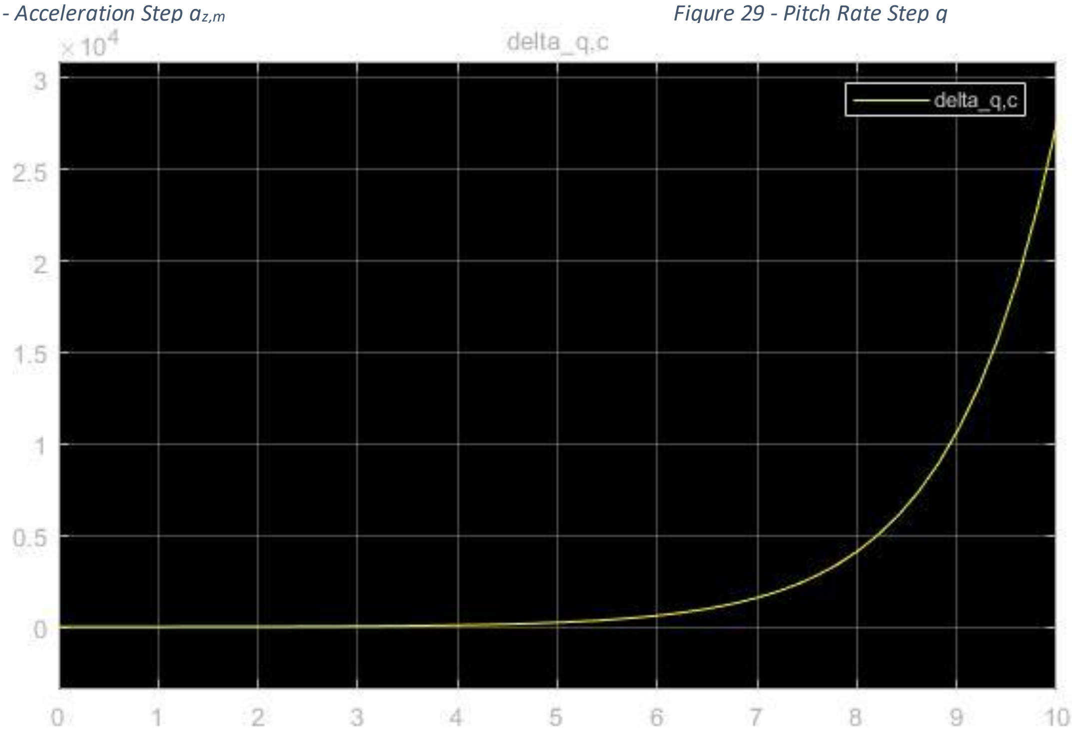

Figure 30 - Control Signal Step $\delta_{q,c}$

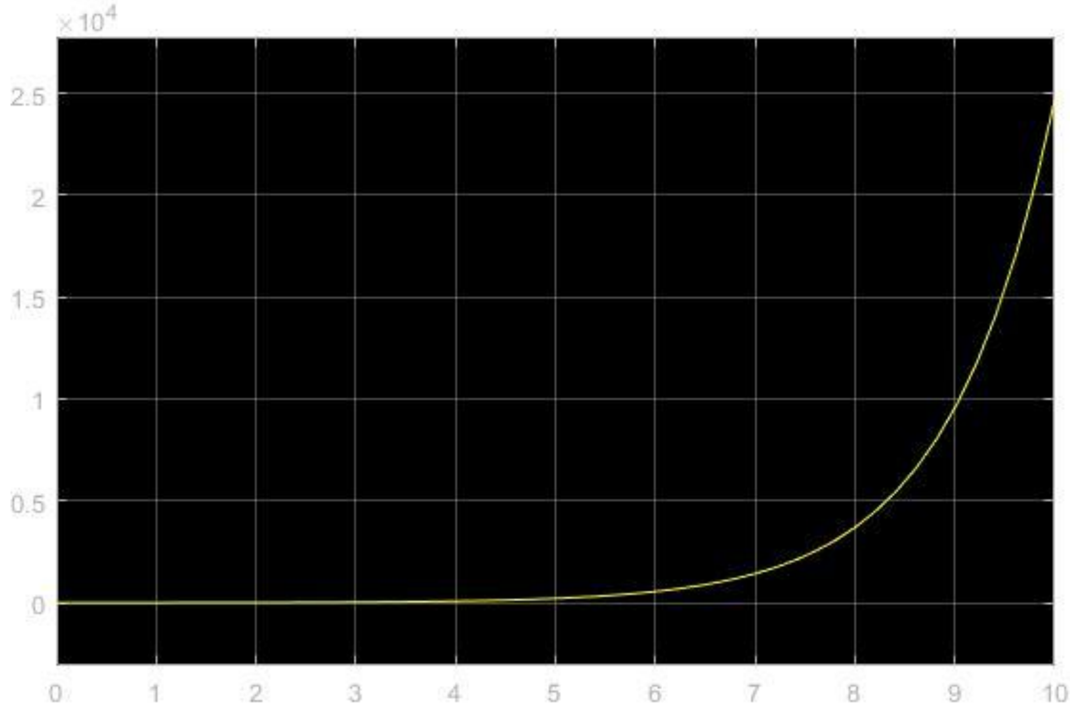

Figure 31 - Derivative of Control Signal Step $\dot{\delta}_{q,c}$

In this case step command was 1g, so it is expected that missile should accelerate. It can be easily seen in figure 29 that missile is accelerating as expected. This acceleration should cause some changes on control surfaces (pitch rate in this case). According to acceleration, pitch rate should decrease to save current altitude of missile and figure 30 approves this statement. It is decreasing to stabilize missile at the current altitude because acceleration means more force more speed and more lift pitch rate is showing a negative trend to save current conditions. Control signal send commands where pitch rate should show this trend and naturally an increase in control signal should be observed over time and it can be seen in figure 30. Derivative of control signal should show the same trend as control signal but slightly smaller because derivative functions minimize the actual functions.

2.H Robustness Analysis

To verify gain and phase margins input of the actuator was chosen as linear analysis point of open loop input and there are two open loop outputs were chosen for linear analysis point. These points are outputs of sensor. From this linear analysis bode diagrams for each case were obtained. This bode diagrams are shown in figure 32.

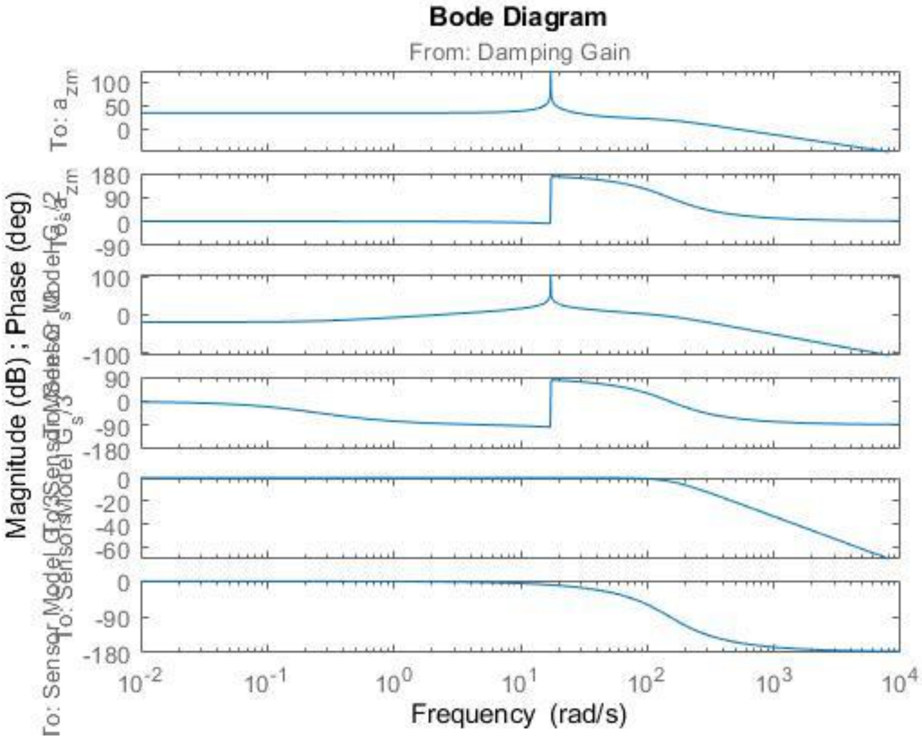

Figure 32 - Bode Diagram from Linear Analysis

According to previous modules, it can be understood that the systems for all three outputs of sensor are stable from its gain margin and phase margin.

Part 3: Adaptive Control Insights

3.A

Adaptive control differs from robust control in terms of past knowledge of limits on uncertain and/or time-varying parameters. It does not any past knowledge about it. If the changes are within certain limits, it is robust control that ensures the control ensures that the control law does not need to be changed while control law is changed by adaptive control. Differences between robust control and adaptive control in terms of pros and cons are as followed:

Robust control can deal with non-minimum phase system and has freedom to choose identification and control design. However, it is difficult to analyse closed loop stability by using robust control. It is not implemented for different weather conditions[7].

Adaptive control has higher stability as much as high speed of reaction. It is also applied easily. However, it requires appropriate system[7].

3.B

Adaptive controller has better performance in closed loop systems than robust controller. A feedback can be added on the system to have closed loop function to use adaptive controller instead of robust controller. Implementation of adaptive controller is as follow:

A controller should be designed to drive plant response to mimic ideal response and the reference model, controller structure and adjustment law can be chosen by the designer. Plant for adaptive control system is not SISO system because it has better performance on MIMO systems. Baseline control is subtracted by adaptive component to have controller for adaptive control system. Furthermore, there are reference model and tracking error dynamics on adaptive control implementation. Coupling between tracking error dynamics and the parameter estimation error dynamics should be cancelled according to adaptive law[8].

References

[1] H. Khalil, ' ', "A classical book in Nonlinear control."

[2] S. Theodoulis, "Robust Flight Control Lecture 1.1: Modeling of Aerospace Systems."

[3] "Transient and Steady-State Response Analyses."

[4] Katsuhiko. Ogata, *Modern control engineering*. Prentice-Hall, 2010.

[5] D. Xia, L. Cheng, and Y. Yao, "A Robust Inner and Outer Loop Control Method for Trajectory Tracking of a Quadrotor," 2017, doi: 10.3390/s17092147.

[6] S. Theodoulis, "Grumman F-14 Tomcat Robust Flight Control Lecture 3.4: Applying í µí°» ∞ Control."

[7] F. Behrooz, N. Mariun, M. H. Marhaban, M. A. M. Radzi, and A. R. Ramli, "Review of control techniques for HVAC systems-nonlinearity approaches based on fuzzy cognitive maps," *Energies*, vol. 11, no. 3, Feb. 2018, doi: 10.3390/en11030495.

[8] "Adaptive Control : MIAC, MRAC, and Concurrent Learning."