Delft University of Technology

SC42145 Robust Control

PART 3 Robust Analysis and Controller Design

1 Part 3.1

A control system is robust if it is insensitive to differences between the actual system and the model of the system which was used to design the controller. These differences are referred to as model/plant mismatch or simply model uncertainty. The key idea in the $\mathcal{H}_\infty$ robust control paradigm we use is to check whether the design specifications are satisfied even for the "worst case" uncertainty.
Our approach is then as follows [1]:
1. Determine the uncertainty set: find a mathematical representation of the model uncertainty.
2. Check Robust stability (RS): determine whether the system remains stable for all plants in the uncertainty set.
3. Check Robust performance (RP): if RS is satisfied, determine whether the performance specifications are met for all plants in the uncertainty set.
This approach may not always achieve optimal performance. In particular, if the worst-case plant rarely or never occurs, other approaches, such as optimizing some average performance or using adaptive control, may yield better performance.
In this part of the assignment we start from the nominal model and derive a perturbed (uncertain) model, followed by an analysis of the robustness associated with the previous controller design. We let Δ denote a normalized perturbation with $\mathcal{H}_\infty$ norm less than 1. Including this uncertainty in the model results in a (infinite) set of models describing the behaviour of the floating wind turbine.
We are first going to add input and output multiplicative uncertainty to the nominal model describing the dynamical behaviour between the inputs β and τ_e and the outputs w_r and z. Then we derive a mathematical expression for the new generalised plant with performance and uncertainty weights. We then interpret the uncertainty weights and plot the singular values of the uncertain transfer matrix describing the dynamic behaviour between the inputs and the outputs. Then we use the mixed-sensitivity controller found in Part 2 and the formulated generalized plant found here and check for nominal stability (NS) using the generalized Nyquist criterion, nominal performance (NP), robust stability (RS) and robust performance (RP) using the Structured Singular Value (SSV). We then relate the value of μ for NP relate to the $\mathcal{H}_\infty$ norm of the objective function in the mixed-sensitivity design.

1.1 QUESTION 1

The performance weights W_p and W_u that were used for the mixed-sensitivity controller and that will be used here again are:

$$W_p = \begin{bmatrix} \frac{\frac{s}{1.8}+0.8\pi}{s+0.8\pi*10^{-4}} & 0 \\ 0 & 0.2 \end{bmatrix}, \; W_u = \begin{bmatrix} 0.01 & 0 \\ 0 & \frac{5*10^{-3}s^2+7*10^{-4}s+5*10^{-5}}{s^2+14*10^{-4}s+10^{-6}} \end{bmatrix}$$

where W_u are the controller weights and W_p are the error weights.
The weight on the input uncertainty is defined as W_i=diag(W_{i1}, W_{i2}) where W_{i1}, W_{i2} are the uncertainty weights on control inputs β and τ_e respectively.
The weight on the output uncertainty is defined as W_o=diag(W_{o1}, W_{o2}) where W_{o1}, W_{o2} are the uncertainty weights on plant outputs w_r and z respectively.
Therefore the input and output uncertainty weights are:

$$W_i = \begin{bmatrix} W_{i1} & 0 \\ 0 & W_{i2} \end{bmatrix} = \begin{bmatrix} \frac{\frac{s}{16\pi}+0.3}{\frac{s}{64\pi}+1} & 0 \\ 0 & \frac{\frac{s}{16\pi}+0.3}{\frac{s}{64\pi}+1} \end{bmatrix}, \; W_o = \begin{bmatrix} W_{o1} & 0 \\ 0 & W_{o2} \end{bmatrix} = \begin{bmatrix} \frac{0.05s+0.2}{0.01s+1} & 0 \\ 0 & \frac{0.05s+0.2}{0.01s+1} \end{bmatrix}$$

Using these weights we draw the block diagram of with the controller, the floating wind turbine, the input and output uncertainty weights, and the performance weights, as shown in Figure 1.

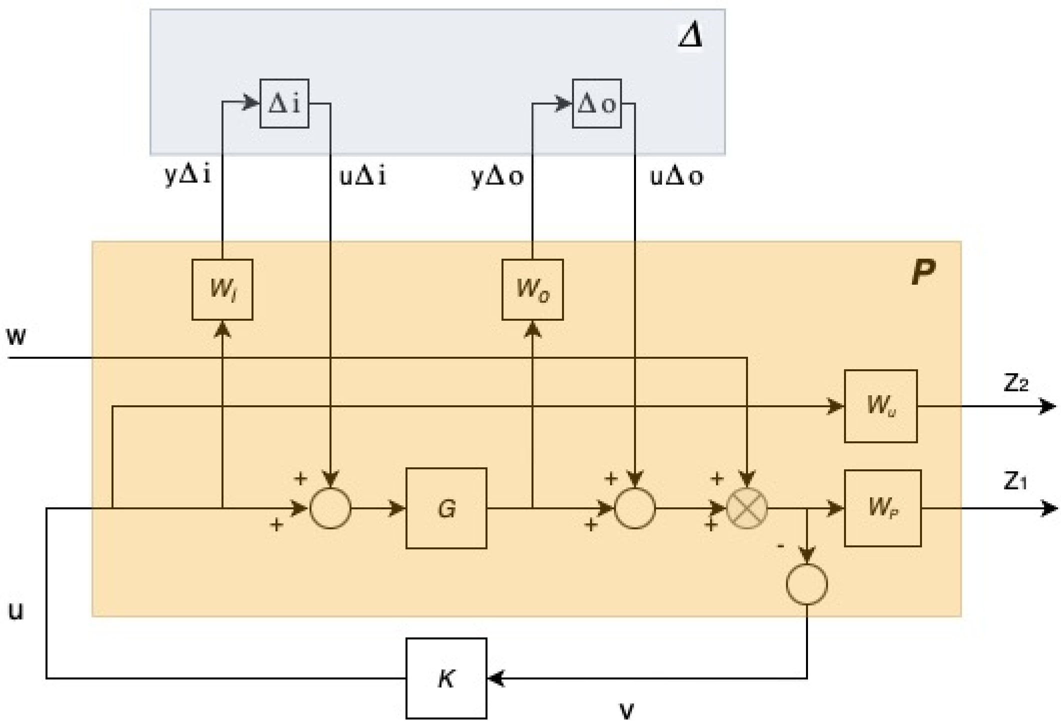

Figure 1: Generalized Plant

G is the nominal plant, which is the same plant used in Part 2 of this project. The outputs of the uncertainty weights are considered to leave the plant and enter the respective perturbation blocks Δ. Therefore, $y_{\Delta i}$ and $y_{\Delta o}$ are the outputs from the uncertainty weights and $u_{\Delta i}$ and $y_{\Delta o}$ are the outputs from the perturbation blocks Δ_i and Δ_o and inputs to our generalized plant respectively. The feedback loop has unity gain and is in negative feedback with the open-loop. The performance signals for the control inputs and output errors are z_2 and z_1 respectively. The derivation for all these expressions will be covered in Question 2. w is the exogenous input to our generalized plant which is r in the case of reference tracking and d in the case of output disturbance rejection. For disturbance rejection, the step input is multiplied with the pre-defined disturbance transfer function G_d.

1.2 QUESTION 2

The new generalised plant is a mapping between the inputs to the plant, and the outputs of the plant. The inputs to the plant are the 2 uncertainty inputs from $u_{\Delta i}$, 2 uncertainty inputs from $u_{\Delta o}$, 2 external inputs at w, and 2 control inputs at u. The outputs of the plant are 2 uncertainty weight outputs $y_{\Delta i}$, 2 uncertainty weight outputs $u_{\Delta o}$, 2 controller weight outputs at z_2, 2 error weight outputs at z_1 and 2 outputs at v, which is the input to the controller. Therefore, the generalized plant P has 10 outputs and 8 inputs which means it should have dimensions of 10x8.
We will derive each output with respect to the input matrix. The input matrix is given by

$$\begin{bmatrix} u_{\Delta i} \\ u_{\Delta o} \\ w \\ u \end{bmatrix}$$

and the output matrix is given by

$$\begin{bmatrix} y_{\Delta i} \\ y_{\Delta o} \\ z_1 \\ z_2 \\ v \end{bmatrix}$$

The first row of generalized plant P will be the mapping between the input matrix and $y_{\Delta i}$. Since we cannot go from $u_{\Delta i}, u_{\Delta o}, w$ to $y_{\Delta i}$ without leaving the plant, all these elements are 0. We can however 'travel' from u to $y_{\Delta i}$ by passing through W_i. Therefore the first row of P is [0, 0, 0, W_i].
The second row of generalized plant P will be the mapping between the input matrix and $y_{\Delta o}$. Since we cannot go from $u_{\Delta o}$ and w to $y_{\Delta o}$ without leaving the plant, both these elements are 0. We can however 'travel' from u to $y_{\Delta o}$ and from $u_{\Delta i}$ to $y_{\Delta o}$ by passing through G and then W_o. Therefore the second row of P is [W_oG, 0, 0, W_oG].
The third row of generalized plant P will be the mapping between the input matrix and z_1. We can travel from all 4 of the plant inputs to z_1. All these inputs have to go through W_p to get to z_1. However, $u_{\Delta i}$ and u have to additionally travel through the nominal plant G. Therefore the third row of P is [W_pG, W_p, W_p, W_pG].
The fourth row of generalized plant P will be the mapping between the input matrix and z_2. It is pretty clear that we can get to z_2 from u only without leaving the plant. Hence all the other corresponding elements are 0. Therefore the fourth row of P is [0, 0, 0, W_u].
The fifth row of generalized plant P will be the mapping between the input matrix and input to the controller v. Since the negative sign is inside the plant, all the outputs to v will be negated. We can travel from w and $u_{\Delta o}$ just by going through the negative sign. Hence they both are -I where I is a 2x2 Identity Matrix. We can travel from $u_{\Delta i}$ and u by first going through G and then the negative sign. Therefore the fifth row of P is [-G, -I, -I, -G].
To summarize, the relation between the input and outputs of the new generalized plant are translated by the matrix P, which is:

$$P = \begin{bmatrix} 0 & 0 & 0 & W_i \\ W_oG & 0 & 0 & W_oG \\ W_pG & W_p & W_p & W_pG \\ 0 & 0 & 0 & W_u \\ -G & -I & -I & -G \end{bmatrix}$$

Therefore,

$$\begin{bmatrix} y_{\Delta i} \\ y_{\Delta o} \\ z_1 \\ z_2 \\ v \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & W_i \\ W_oG & 0 & 0 & W_oG \\ W_pG & W_p & W_p & W_pG \\ 0 & 0 & 0 & W_u \\ -G & -I & -I & -G \end{bmatrix} \begin{bmatrix} u_{\Delta i} \\ u_{\Delta o} \\ w \\ u \end{bmatrix}$$

1.3 QUESTION 3

Uncertainty in the plant model may have several origins. Therefore, there are several different ways in which uncertainty can affect the system performance. The model provided to us suggests that the uncertainty is lumped in the form of input and output uncertainty. This is of the form

$$G_p(s) = (1 + W_O(s)\Delta_O(s))G(s)(1 + W_I(s)\Delta_I(s)) \quad ||\Delta_I||_\infty < 1,\ ||\Delta_O||_\infty < 1 \tag{1}$$

Here Δ_I and Δ_O can be any stable transfer function which has $\mathcal{H}_\infty < 1$. This approach is used to

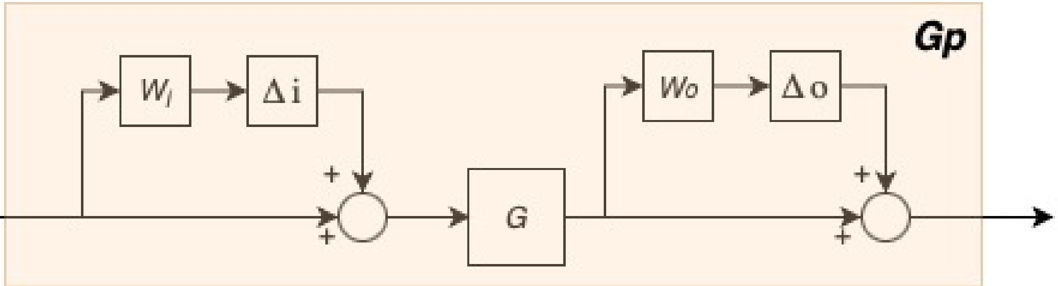

Figure 2: Perturbed Plant G_P with input and output uncertainties

quantify unmodelled dynamics uncertainty. At frequencies where $|W_I(jw)| > 1$ or $|W_O(jw)| > 1$, the uncertainty exceeds 100% and the Nyquist curve may pass through the origin. At these frequencies we do not know the phase of the plant, and we allow for zeros crossing from the left to the right-half plane. For this plant at these frequencies the input has no effect on the output, so control has no effect.

The usual procedure to obtain weights for complex uncertainties are to select $W_I(w) \geqslant l_I(w), \forall w$ or $W_O(w) \geqslant l_O(w), \forall w$ where $l_I(w) = \max_{G_P \in \Pi} \bar{\sigma}(G^{-1}(G_P - G)(jw))$ and $l_O(w) = \max_{G_P \in \Pi} \bar{\sigma}((G_P - G)G^{-1}(jw))$ [2]. Output uncertainty is frequently less restrictive than input uncertainty in terms of control performance.

To represent unmodelled dynamics we usually use a simple multiplicative weight of the form

$$w_I(s) = \frac{\tau s + r_0}{(\tau / r_\infty)s + 1}$$

where r_0 is the relative uncertainty at steady-state, $(1/\tau)$ is approximately the frequency at which relative uncertainty reaches 100%, and r_∞ is the magnitude of the weight at high frequency. Typically, the uncertainty $|w|$, associated with each input, is at least 10% at steady-state($r_0 < 0.1$), and it increases at higher frequencies to account for neglected or uncertain dynamics (typically, $r_\infty > 2$). This type of diagonal uncertainty is always present and that makes it all the more important to select appropriate weights and account for these uncertainties. This uncertainty weight should be designed such that all kinds of uncertainty in the model can satisfy $W_I(w) \geqslant l_I(w), \forall w$ or $W_O(w) \geqslant l_O(w), \forall w$. Some examples of uncertainty models that should taken into consideration are neglected delays, neglected lag, uncertain pole in frequency and time constant forms, neglected resonance, neglected dynamics, neglected RHP-Zero, etc.

Starting with the input uncertainty weight, the transfer function we are provided with is

$$W_i = \begin{bmatrix} W_{i1} & 0 \\ 0 & W_{i2} \end{bmatrix} = \begin{bmatrix} \frac{\frac{s}{16\pi}+0.3}{\frac{s}{64\pi}+1} & 0 \\ 0 & \frac{\frac{s}{16\pi}+0.3}{\frac{s}{64\pi}+1} \end{bmatrix}$$

Both the uncertainty weights are the same, thus analysing any one of them serves the purpose. The Bode plot of the $\frac{\frac{s}{16\pi}+0.3}{\frac{s}{64\pi}+1}$ is shown in Figure 3. Comparing the given weight to the generalized unmodelled dynamics uncertainty weight, we can see that $\tau = 16\pi \approx 50.26 rad/s$, $r_\infty = 4$, $r_0 = 0.3$. From the bode plot we can see that this weight has a gain of 4 at higher frequencies, a relative uncertainty of 0.3, and it crosses the point 10^0(100% uncertainty) at approximately 50.2 rad/s.

Initializing uncertainty perturbation blocks using **ultidyn** command we plot the bode of $w_I \Delta_I$ and

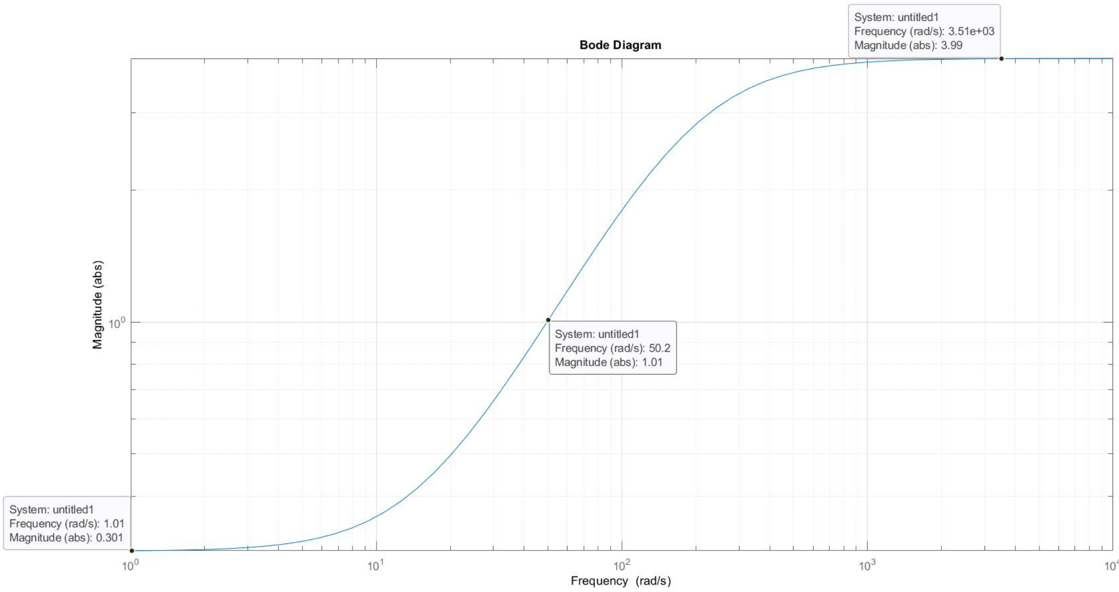

Figure 3: Bode Plot of Input Uncertainty Weight

notice that the weights more or less seem to restrict the perturbations to a band of gains. This is shown in Figure 4. Also observe that although the uncertainty weights are same, the perturbations are different and so this leads to different random bands every time.

For the output uncertainty weight, the transfer function we are provided with is

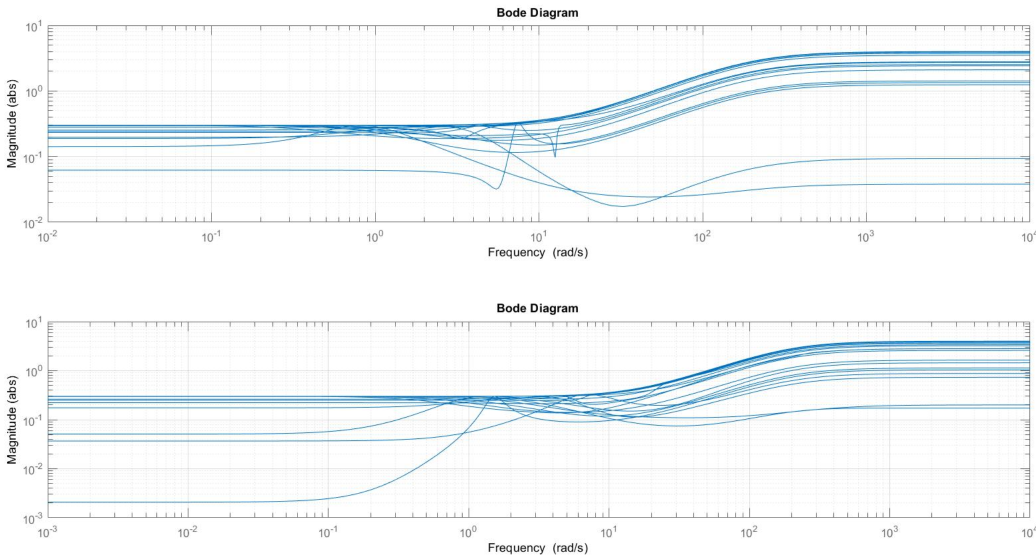

Figure 4: Bode Plot of $W_I \Delta_I$

$$W_o = \begin{bmatrix} W_{o1} & 0 \\ 0 & W_{o2} \end{bmatrix} = \begin{bmatrix} \frac{0.05s+0.2}{0.01s+1} & 0 \\ 0 & \frac{0.05s+0.2}{0.01s+1} \end{bmatrix}$$

Both the uncertainty weights are the same, thus analysing any one of them serves the purpose. The Bode plot of the $\frac{0.05s+0.2}{0.01s+1}$ is shown in Figure 5. Comparing the given weight to the generalized unmodelled dynamics uncertainty weight, we can see that $\tau = 20 rad/s$, $r_\infty = 5$, $r_0 = 0.2$. From the bode plot we can see that this weight has a gain of 5 at higher frequencies, a relative uncertainty of 0.2, and it crosses the point 10^0(100% uncertainty) at approximately 20 rad/s.

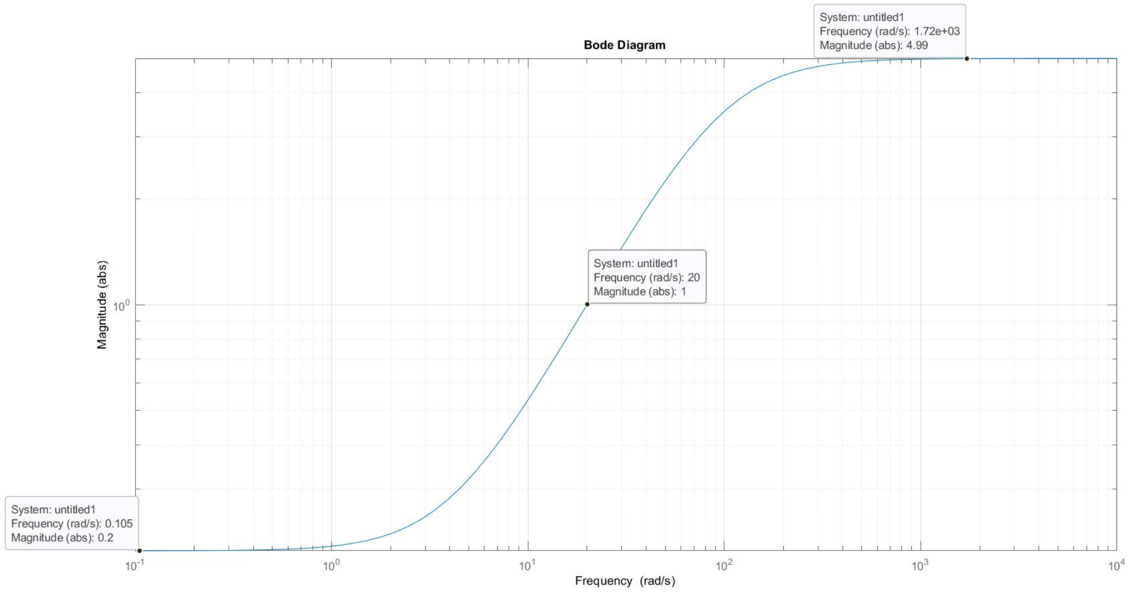

Figure 5: Bode Plot of Output Uncertainty Weight

Initializing uncertainty perturbation blocks using **ultidyn** command we plot the bode of $w_O \Delta_O$ and notice that the weights more or less seem to restrict the perturbations to a band of gains. This is shown in Figure 6. Also observe that although the uncertainty weights are same, the perturbations are different and so this leads to different random bands every time.

We not calculate the perturbed plant G_P and plot its singular values, as shown in Figure 7. We

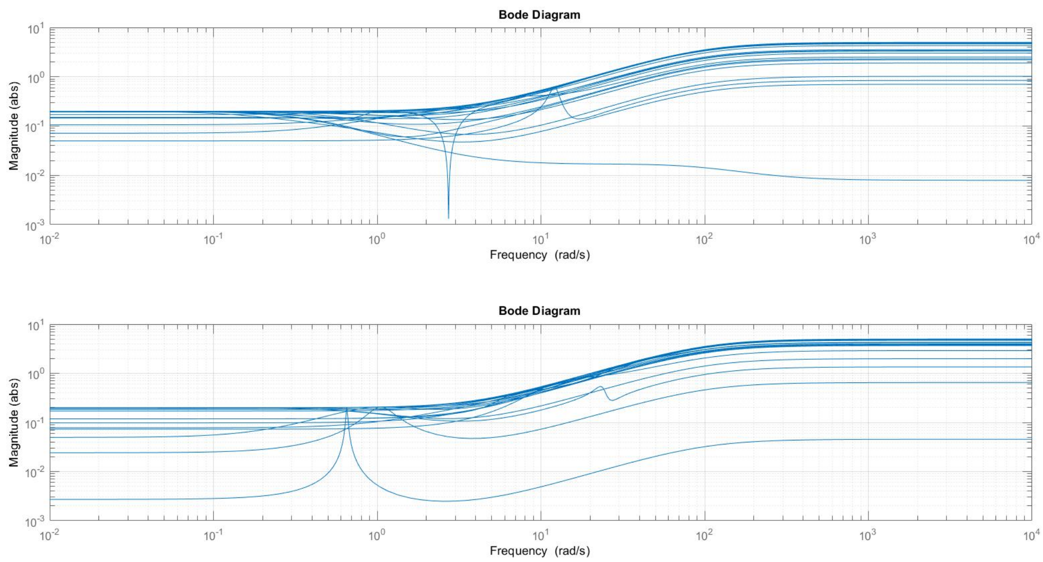

Figure 6: Bode Plot of $W_O \Delta_O$

can see that the frequencies at which the uncertainty bands start diverging correspond to the 100% uncertainty value τ discussed previously.

Relating this to the inputs β and τ_e and outputs w_r and z, we know that singular values represent the amplification from the inputs to the outputs over various frequencies. We can thus see that at higher frequencies there is a larger uncertainty and less gain in the output, while the lower frequencies have much lesser uncertainties and high gain. Thus, the plant we are presented with creates high uncertainties in the outputs beyond frequencies 50.26 rad/s and 20 rad/s where the inputs β and

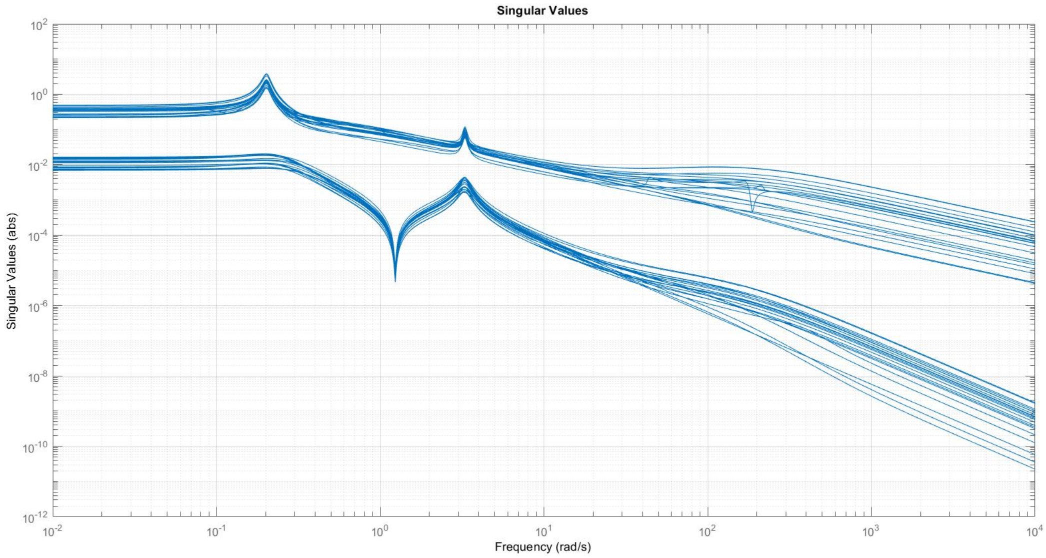

Figure 7: Singular Values of Perturbed Plant

τ_e become more uncertain and start to diverge beyond 50.26 and the outputs β and z become more uncertain and start to diverge beyond 20 rad/s.

1.4 QUESTION 4

Now that we have defined the uncertainty and performance blocks, we can augment the generalized plant code from the nominal design of Part 2 to include the extra inputs and outputs as shown in Question 1. The resultant generalized plant we obtain P is as derived in Question 2. This is directly programmed into our MATLAB code as a matrix. This is as shown in the Figure below. As mentioned previously P has 10 outputs and 8 inputs, and is thus a 10x8 matrix.

$$P = \begin{bmatrix} 0 & 0 & 0 & W_i \\ W_oG & 0 & 0 & W_oG \\ W_pG & W_p & W_p & W_pG \\ 0 & 0 & 0 & W_u \\ -G & -I & -I & -G \end{bmatrix}$$

```
P1=[[0 0 0 0 0 0;0 0 0 0 0 0] Wi];
P2=[Wo*G [ 0 0 0 0;0 0 0 0] Wo*G];
P3=[Wp*G Wp Wp Wp*G];
P4=[zeros(2,2) [0 0 0 0;0 0 0 0] Wu];
P5=[-G -eye(2) -eye(2) -G];
P=[P1;P2;P3;P4;P5];
P=minreal(ss(P),0.0001);
```

Figure 8: Programming the Generalized Plant in MATLAB

1.5 QUESTION 5

1.5.1 Nominal Stability(NS)

The first task is to check for Nominal Stability(NS) using the Generalized Nyquist Criterion.
Since this has been covered extensively in the mixed-sensitivity design, it will be concise and to the point. The open loop is defined as L=G*K. Based on the Cauchy's argument principle, for the closed-loop system be internally stable, the generalized Nyquist plot of det(I + L) should encircle origin N=Z-P times clock wise. Z is the number of the unstable closed loop poles and P is the number of the unstable open loop poles [3].
In our case, we have an unstable pole pair at [0.0083 ± 3.4260i]. It is clear that Z=0 since the number of the unstable closed loop poles is 0, and P=2 since the number of the unstable open loop poles are 2. Thus the generalized Nyquist plot of det(I + L) should encircle origin N=0-2 times clock wise, which is 2 times anti-clock wise.
By using the MATLAB command nyquist, the generalized Nyquist plot of det(I+L) is shown below:
As seen above, the Nyquist plot encircles origin 2 times anti-clock wise, which affirms that the closed-

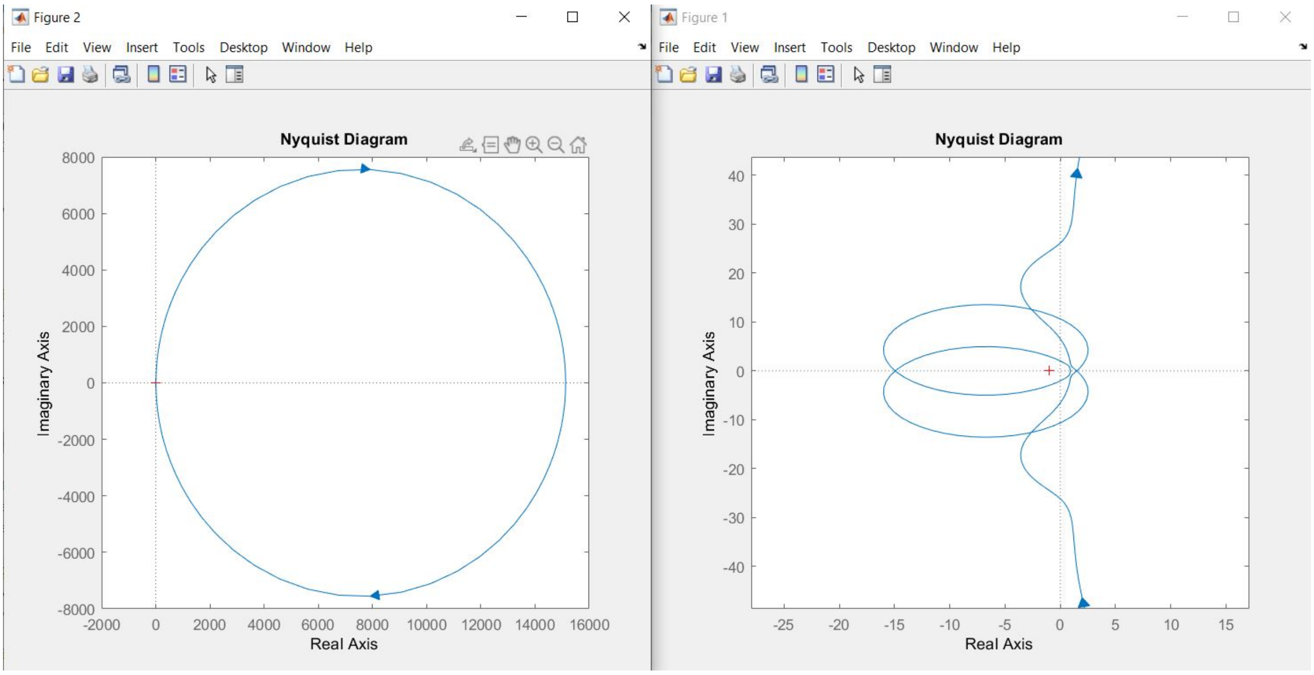

Figure 9: Nyquist plot of det(I+L) and its zoomed plot

loop system is internally stable, and thus Nominally Stable.
Another test for NS is to check the Eigen values of N matrix(Closed-Loop Relation between inputs and outputs), and if all the Eigen Values are in the LHP, the system in NS [1].
This too, worked in our case, where our largest eigen value was -0.0106.

1.5.2 Nominal Performance(NP)

The underlying principle behind checking for nominal performance(NP) is to ensure that the performance weights bound the closed-loop Sensitivity for all frequencies. From the mixed-sensitivity design we know that if $W_pS < 1\forall w$ and $KW_uS < 1\forall w$, and in addition the system is Nominally Stable, we can declare that we have achieved Nominal Performance.

N matrix is defined as the relation between the inputs(reference/disturbance input and perturbation inputs) and the outputs when the controller is included in the plnat(i.e. closed-loop). In MATLAB we find this N matrix by using the command $\mathbf{lft(P, K)}$ where P is the new generalized plant and K is the mixed-sensitivity controller we designed previously. lft is the Linear Fractional Transformation (LFT) or the star product of both its input argument models.

The N matrix is structured such that it can be split into 4 parts $N_{11}, N_{12}, N_{21}, N_{22}$ by defining each block as the relation between the corresponding input and output. Thus, N_{11} is the closed-loop 'pathway' from the perturbation inputs to the outputs of the uncertainty weights without leaving the block. Similarly, N_{22} is the closed-loop 'pathway' from the external input(disturbance, reference) to the performance weight outputs without leaving the block. From the block in Figure 1, we can see that this term does not depend on the uncertainty or perturbations, but just on the performance weights, the controller, and the nominal plant. In fact the 2 terms in this block are W_pS and KW_uS which are exactly what we covered in the mixed-sensitivity design. Thus, we can achieve nominal performance if $||N_{22}||_\infty < 1$ and if N is nominally stable.

In this case however, we attempt to show this by using SSV(Structured Singular Values μ). We will calculate the bounds on these SSVs and after normalising them, show that they are lesser than 1, hence achieving nominal performance. The structured singular value is a function which provides a generalization of the singular value, $\bar{\sigma}$, and the spectral radius, ρ. The SSV does not just depend only on the frequency data model input, but also the structure of the perturbations(Δ).

Therefore, we achieve NP if

$$\bar{\sigma}(N_{22}) = \mu_{\Delta_P} < 1,\ \forall w,\ and\ NS \tag{2}$$

where Δ_P is always a full complex matrix. We must first convert our N model to the frequency domain over a certain frequency range. Our N_{22} here is a 4x2 matrix(2x2 for error weight output, 2x2 for controller weight output) and the perturbation block structure in this case is 2x4. Using the **mussv** command, we get the bound on the SSV and then find the infinity norm. Using this procedure, the peak value of the μ synthesis of N_{22} was found to be 0.7992. Since this is lesser than 1, and N is NS, we can confirm that NP has been achieved. This is confirmed from the plot in Figure 10.

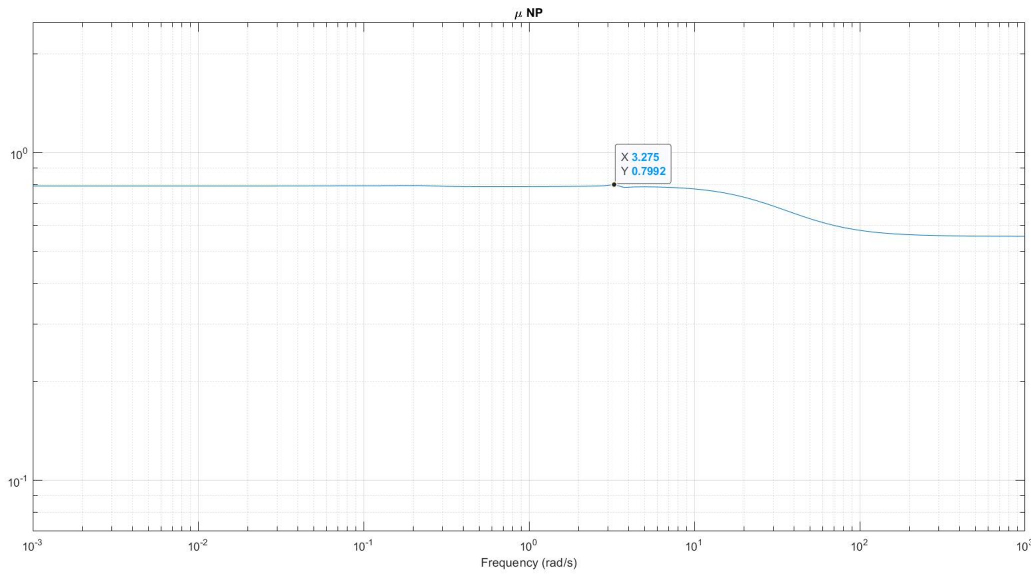

Figure 10: Variation of μ with frequency for NP

1.5.3 Robust Stability(RS)

The concepts of robustness, robust stability and robust performance arise due to the existence of uncertainties and perturbations in the model.
Similar to the way we placed a bound on the performance weights with respect to the sensitivity for NP, here we placed a bound on the complementary sensitivity function with respect to the uncertainty weights. Therefore the requirement is that $||W_I T||_\infty < 1$, and this can be seen from Figure 11. A broad way of understanding why we check N_{11} only for RS is that the actual definition of RS is that RS is achieved if $F_u(N, \Delta)$ is stable for all Δ, $||\Delta||_\infty \leqslant 1$ and NS. Here,

$$F_u(n, \Delta) \triangleq N_{22} + N_{21}\Delta(I - N_{11}\Delta)^{-1}N_{12}$$

For F_u to be stable, N_22 needs to be stable, which is proven by NS. N_{21}, Δ and N_{12} are assumed to be stable, which means that the stability depends only on the stability of $(I - N_{11}\Delta)^{-1}$, and thus N_{11} [4].

Therefore, we achieve RS if

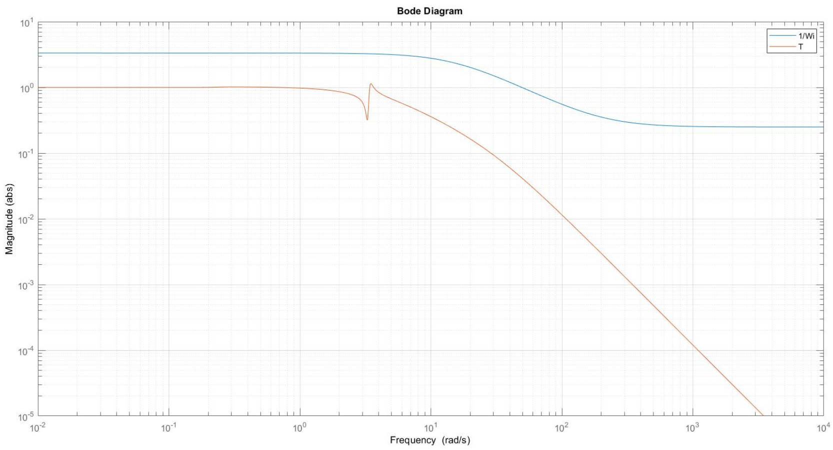

Figure 11: Input Uncertainty Weight Bounds the Complementary Sensitivity Function

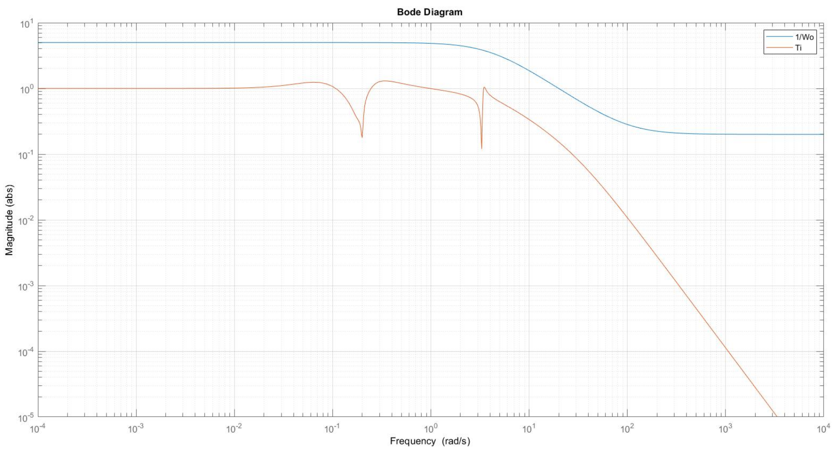

Figure 12: Output Uncertainty Weight Bounds the Complementary Sensitivity Function

$$\mu_\Delta(N_{11}) < 1,\ \forall w,\ and\ NS \tag{3}$$

where Δ is a block-diagonal matrix. Procedure wise, the computation of SSV is very similar to that done for NP. The differences here are that the frequency model whose bounds we find is now N_{11} instead of N_{22}. This is the relation between the inputs to the N block, $u_{\Delta I}$ and $u_{\Delta O}$, and the outputs to the perturbation blocks $y_{\Delta I}$ and $y_{\Delta O}$. When combined N_{11} becomes a 4x4 diagonal matrix.

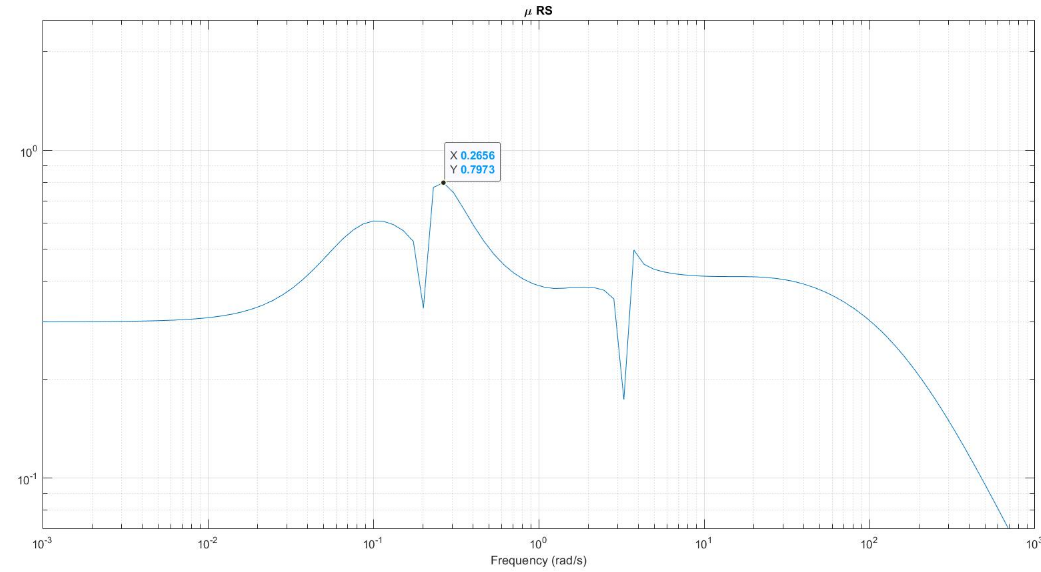

Figure 13: Variation of μ with frequency for RS

Correspondingly, the perturbation structure is also taken as a 4x4 diagonal. Once again we find the bounds, and normalise them with respect to infinity. Using this procedure, the peak value of the μ synthesis of N_{11} was found to be 0.7973. Since this is lesser than 1, and N is NS, we can confirm that RS has been achieved. This is confirmed from the plot in Figure 12.

1.5.4 Robust Performance(RP)

Robust performance can be seen as putting together everything done for NP, NS AND RS.
For RP, the condition on the function F_u now becomes [1],

$$||F_u(N, \Delta)||_\infty < 1 \forall \Delta, \ ||\Delta||_\infty \leqslant 1 \ and \ NS$$

This implies that we now consider the whole N structure while finding the Δ-norm. For Robust Performance, we require that [1]

$$|W_p S| + |W_u KS| + |W_I T| + |W_O T_i| < 1, \forall w$$

Therefore, we achieve RP if

$$\mu_{\hat{\Delta}}(N) < 1, \ \forall w, \hat{\Delta} = [\Delta \ 0; 0 \ \Delta_P] \ and \ NS \tag{4}$$

As it can be seen this can be seen as a putting together of the conditions covered for NS, NP and RS. For the SSV, we consider the full frequency response structure N, which includes N_{11} used for RS and N_{22} used for NP. All this is in addition to NS of course. This makes the dimension of N to be 8x6(since N_{11} is 4x4 and N_{22} is 4x2). The dimensions of the perturbation block structure for this SSV is a combination of the ones used for RS and NP and it has dimensions of 6x8. Once again we find the bounds, and normalise them with respect to infinity. Using this procedure, the peak value of the μ synthesis of N_{11} was found to be 1.7183. Since this is not lesser than 1, and even though N is NS, we can see that RP has not been achieved. Figure 15 shows the variation the normalized SSV bounds with frequency.

Seeing these results we can conclude that the system with uncertainties which we have is Nominally

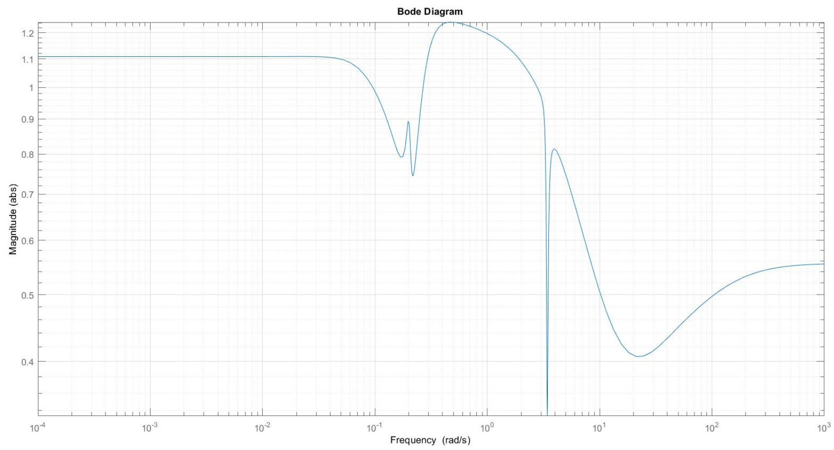

Figure 14: Bode plot of $|W_p S| + |W_u KS| + |W_I T| + |W_O T_i|$

Stable, achieves Nominal Performance, is Robust Stable, but does not achieve the required Robust Performance. Figure 16 combines all the μ values in a single plot.

To achieve RP, we need to reduce the penalties on the weighting matrices. If we reduce the penalty

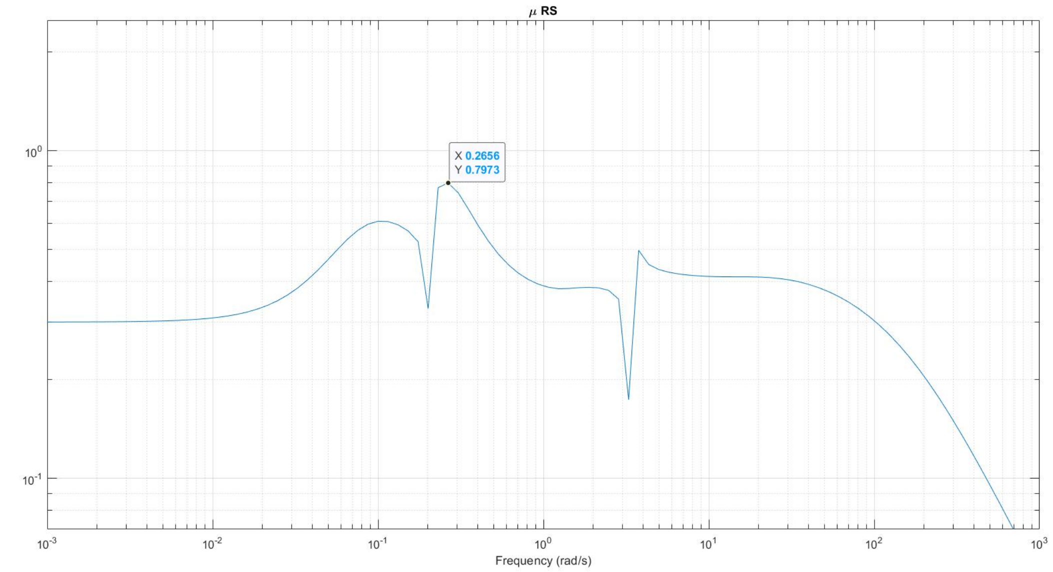

Figure 15: Variation of μ with frequency for RP

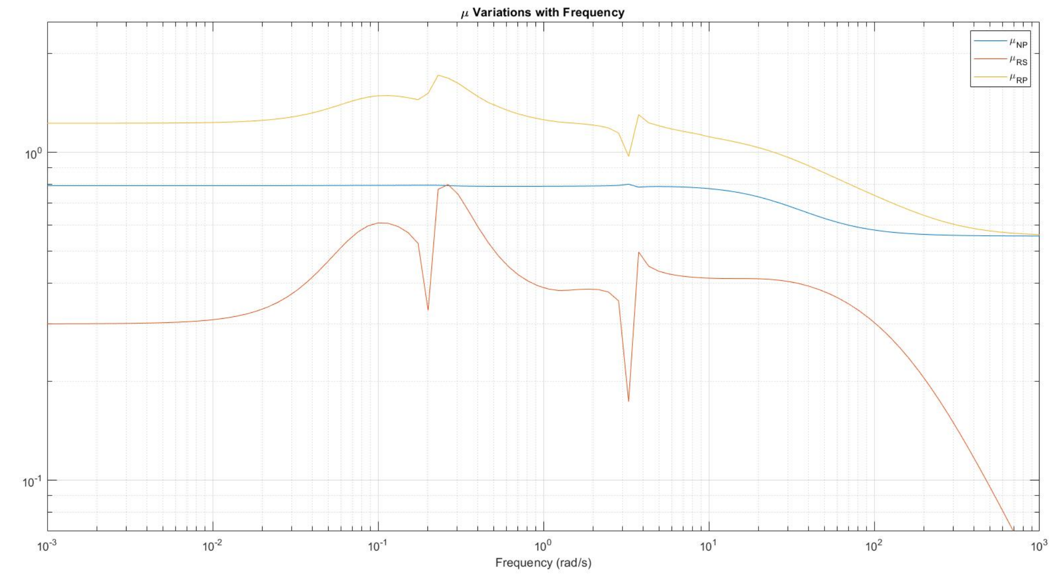

Figure 16: Variation of μ with frequency for NP, RS and RP

imposed by W_p, we find that we can achieve RP well below 1, and this will result in a robust, but overall poorly performing controller.

1.6 QUESTION 6

The μ synthesis for checking nominal performance NP is done on the N_{22} term for the N matrix. N_{22} is the term of the N matrix corresponding to the closed-loop relation between the input channel at the plant output and the outputs at the performance weightsz_1 and z_2.
In the $\mathcal{H}_\infty$ framework used to design the mixed-sensitivity controller, our task was to ensure that $||N||_\infty < 1$ and we did this by ensuring that $||W_pS||_\infty < 1$ and $||W_uKS||_\infty < 1$ by appropriately designing weights W_p and W_u and then checking the peak of the $\mathcal{H}_\infty$ norm [4]. In this case, we find

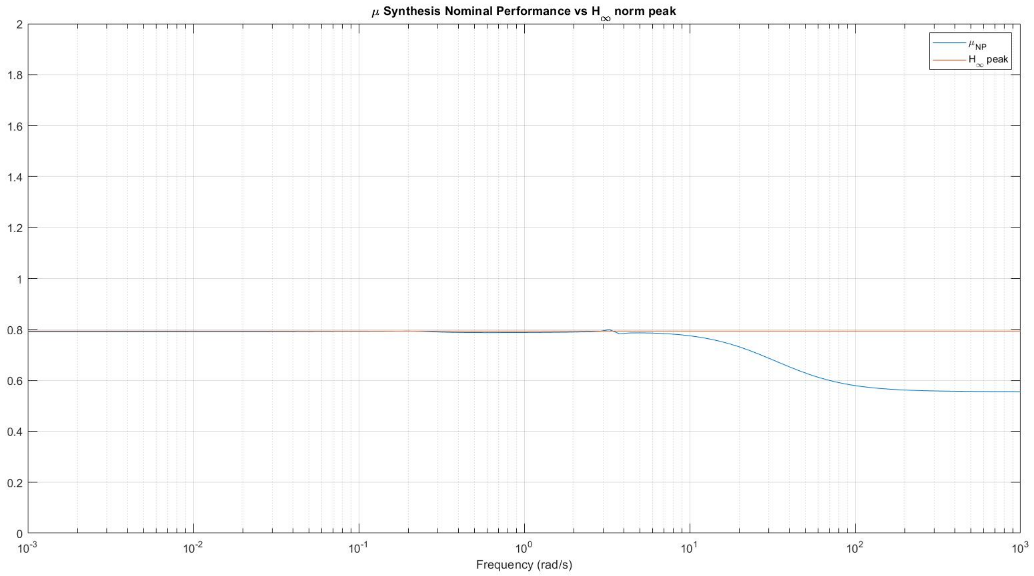

Figure 17: Comparison of μ Synthesis for NP vs Mixed-Sensitivity Design

the peak by normalizing(infinity norm) the Structured Singular Values(SSV) of N_{22} matrix. Since the N_{22} matrix in this case is the N matrix used in the mixed-sensitivity design, we expect that both the synthesis methods will return the same peak and that both will be less than 1.
And indeed, the μ value for NP was found to be 0.79.
The **hinfnorm** norm value for the mixed-sensitivity design was found to be 0.79, as expected. The Figure shows the variation of μ values with frequency, and the horizontal line corresponds to the peak of the $\mathcal{H}_\infty$ norm.

2 Part 3.2

The structured singular value μ is a powerful tool for the analysis of robust performance with a given controller as shown in the previous part. Therefore, we need to seek μ-synthesis to find the controller that minimizes a certain μ-condition. At present there is no direct method to synthesize a μ-optimal controller. However, for complex perturbations a method known as DK -iteration is available. It combines $\mathcal{H}_\infty$-synthesis and μ-analysis, and often yields good results. In this part, the D-K iteration is applied to synthesize a controller that achieves robust stability (RS) and robust performance (RP). The comparison of the time-domain simulations between the mixed-sensitivity controller and the synthesized controller is proceeded with and the relevant frequency-domain analysis explains the differences in the results.

2.1 Question 1

The starting point of the DK-iteration is the upper bound on μ in terms of the scaled singular value:

$$\mu(N(K)) \leq \min_{D} \bar{\sigma}(DN(K)D^{-1}) \tag{5}$$

with D satisfying

$$\hat{\Delta}D = D\hat{\Delta} \tag{6}$$

where $\hat{\Delta}$=diag(Δ,Δ_p)=diag(Δ_i, Δ_o, Δ_p).The size of Δ_p is 2×2.

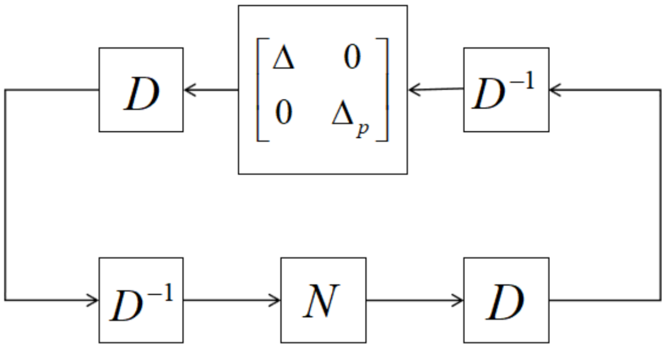

Figure 18: Block diagram of D matrix

The idea is to find the controller that minimizes the peak value over frequency of this upper bound, namely

$$\min_{K}(\min_{D \in \mathcal{D}} ||DN(K)D^{-1}||_{\infty}) \tag{7}$$

by alternating between minimizing $||DN(K)D^{-1}||_{\infty}$ with respect to either K or D (while holding the other fixed). To start the iterations, one selects an initial stable rational transfer matrix D(s) with appropriate structure. The identity matrix is often a good initial choice for D provided the system has been reasonably scaled for performance. The DK-iteration then proceeds as follows [4]:

- **K-step**. Synthesize an $\mathcal{H}_{\infty}$ controller for the scaled problem, $\min_K ||DN(K)D^{-1}||_{\infty}$ with fixed D(s).
- **D-step**. Find $D(j\omega)$ to minimize at each frequency $\bar{\sigma}(DND^{-1}(j\omega))$ with fixed N.
- Fit the magnitude of each element of $D(j\omega)$ to a stable and minimum phase transfer function D(s) and go to Step 1.
- Stop iterating when, for a given iteration step k, $|D_{k-1} - D_k|$ and $|K_{k-1} - K_k|$ are less than a specified tolerance level.

One fundamental problem with this approach is that although each of the minimization steps (K-step and D-step) are convex, joint convexity is not guaranteed. Therefore, the iterations may converge to a local optimum. However, practical experience suggests that the method works well in most cases.

2.2 Question 2

We use MATLAB command **dksyn** to synthesize a robust controller via D-K iteration. The D-K iteration procedure is an approximation to μ-synthesis control design. The objective of μ-synthesis is to minimize the structure singular value μ of the corresponding robust performance problem associated with the uncertain system p_{un}. The uncertain system p_{un} is an open-loop interconnection containing known components including the nominal plant model P and the perturbation block Δ. We use MATLAB command $\mathbf{p_{un}} = \mathbf{lft}(\mathbf{\Delta}, \mathbf{P})$ to get the uncertain system. **dksyn** automates the D-K iteration procedure and the options object **dksynOptions** allows us to customize its behavior. The following is a list of what occurs during a single, complete step of the D-K iteration [3].

- (In the first iteration, this step is skipped.) The μ calculation (from the previous step) provides a frequency-dependent scaling matrix, D_f. The fitting procedure fits these scalings with rational, stable transfer function matrices. (In the first iteration, this step is skipped.) The rational $\hat{D}$ is absorbed into the open-loop interconnection for the next controller synthesis. Using either the previous frequency-dependent D's or the just-fit rational $\hat{D}$, an estimate of an appropriate value for the $\mathcal{H}_\infty$ norm is made. This is simply a conservative value of the scaled closed-loop $\mathcal{H}_\infty$ norm, using the most recent controller and either a frequency sweep (using the frequency-dependent D's) or a state-space calculation (with the rational D's).

- (The first iteration begins at this point.) A controller is designed using $\mathcal{H}_\infty$ synthesis on the scaled open-loop interconnection. If you set the **DisplayWhileAutoIter** field in **dksynOptions** to 'on', the following information is displayed:
  - The progress of the γ-iteration is displayed.
  - The singular values of the closed-loop frequency response are plotted.
  - You are given the option to change the frequency range. If you change it, all relevant frequency responses are automatically recomputed.
  - You are given the option to rerun the $\mathcal{H}_\infty$ synthesis with a set of modified parameters if you set the **AutoIter** field in **dksynOptions** to 'off'. This is convenient if, for instance, the bisection tolerance was too large, or if maximum gamma value was too small.

- The structured singular value of the closed-loop system is calculated and plotted. item An iteration summary is displayed, showing all the controller order, as well as the peak value of μ of the closed-loop frequency responses.

- The choice of stopping or performing another iteration is given.

Subsequent iterations proceed along the same lines without the need to reenter the iteration number. A summary at the end of each iteration is updated to reflect data from all previous iterations. This often provides valuable information about the progress of the robust controller synthesis procedure. The same procedure is adopted here is test the NS,NP,RS and RP using the structured singular values. The results are collected in Table 1:

Table 1: NS,NP,RS and RP of the system

Check	Criteria	Value
NS	N internally stable (max(real(λ(N)))<0))	$-8.5864 * 10^{-04}$
NS	$\mu\Delta_p(N_{22}) < 1$	0.8794
RS	$\mu\Delta(N_{11}) < 1$	0.5797
RS	$\mu\hat{\Delta}(N) < 1$	1.1401

Since all the criteria are satisfied (with $\mu\hat{\Delta}(N)$=1.1401, we can conclude the RS), the DK iteration is successful. The criterion values of μ are shown in Figure 19.

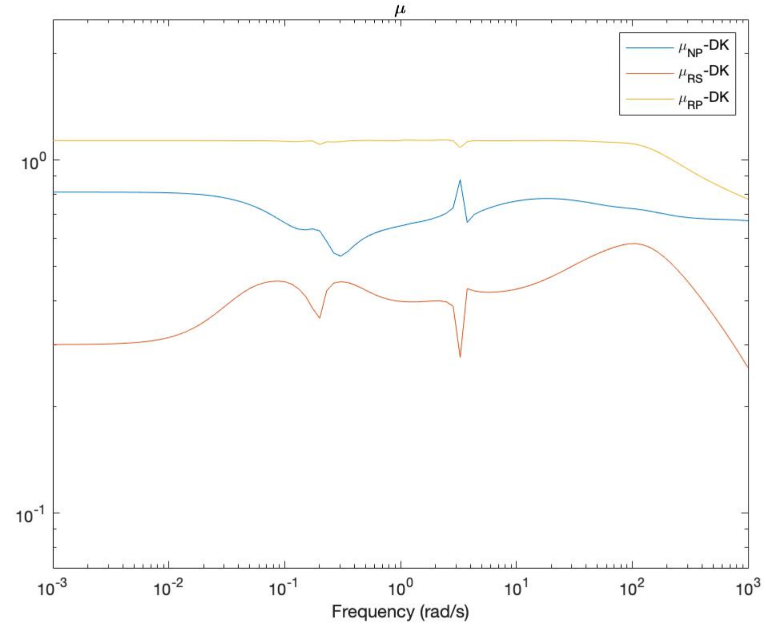

Figure 19: μ values for NP, RS and RP

2.3 Question 3

For the last part, we perform time-domain simulations for reference tracking and disturbance rejection with step input using the vert last obtained controller from DK iterations and mixed sensitivity controller designed in the second report. Figure 20 shows the time domain performances of reference tracking with two controllers respectively. The statistical information of the performances are collected in Table 2 . As can be seen, the controller designed with DK iteration gives a better settling time and robustness in both stability and performance, than the mixed sensitivity controller regarding the first output. When we focus on the performance of the second output of ζ, the mixed sensitivity controller provides a slightly better performance in terms of settling time and overshoot, but both of them have steady state error. Figure 21 shows the time domain performances of disturbance rejection with two controllers respectively. By observation, we find that the response with the controller designed with DK iteration is also better than the mixed sensitivity controller. The DK controller gives robustness in both stability and performance, especially in the second output ζ, where the fluctuation is clearly flattened out. Similar to the reference tracking, there are steady state errors for both loop transfer functions.

The frequency-domain bode plots of two open-loop systems are shown in Figure 24. As we can see, when applying controller synthesized by DK iteration to the plant, the open loop of each channel would have higher gain at low frequency, compared to the open loop with controller designed by mixed-sensitivity. Besides, the DK controller also has a similarly good noise rejection at high frequencies, as compared to the mixed sensitivity controller. Besides, the plot of sensitivity and complementary sensitivity of the closed-loop system with two controllers are shown in Figure 22 and Figure 23. By comparison, we can observe that the magnitude of sensitivity for DK controller, of the output ω_r at low frequency is slightly lower and has a slightly lower peak. For output ζ, there is no obvious difference of two sensitivities at low frequency. From the figures, we can find that the bandwidth of the system with the DK controller is slightly higher for the output ω_r, and the difference is not obvious when it comes to output ζ. Overall, based on all of these frequency domain analysis, the DK controller has a better tracking and disturbance rejection from a frequency response perspective.

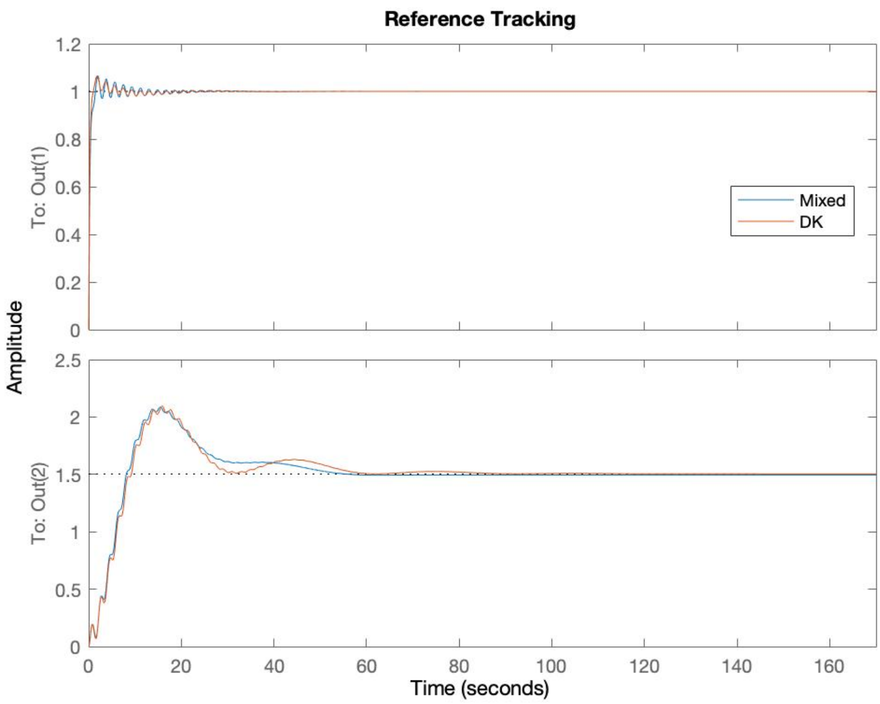

Figure 20: Reference tracking performance with two controllers

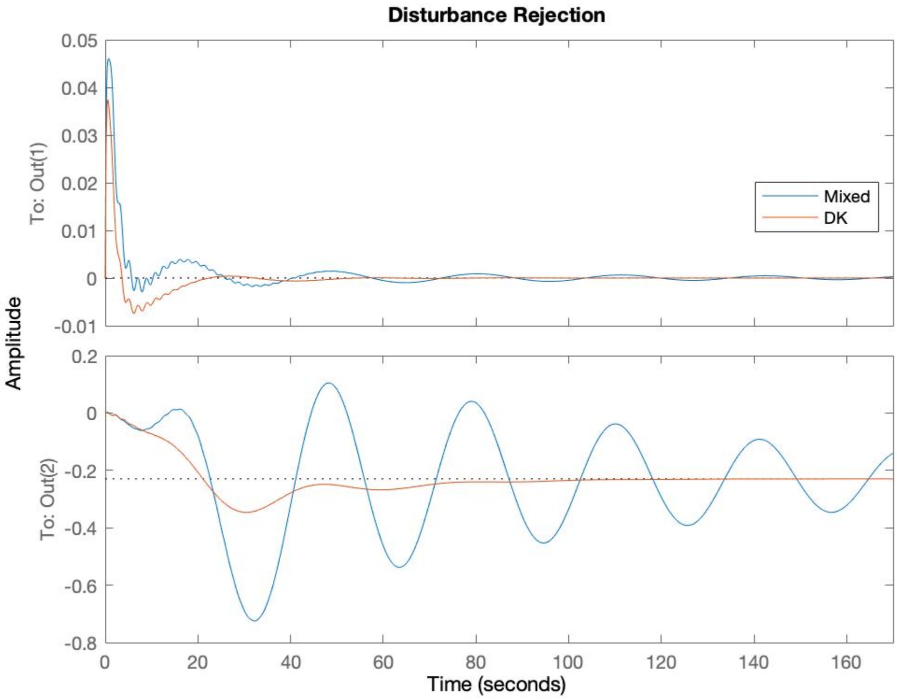

Figure 21: Disturbance rejection performance with two controllers

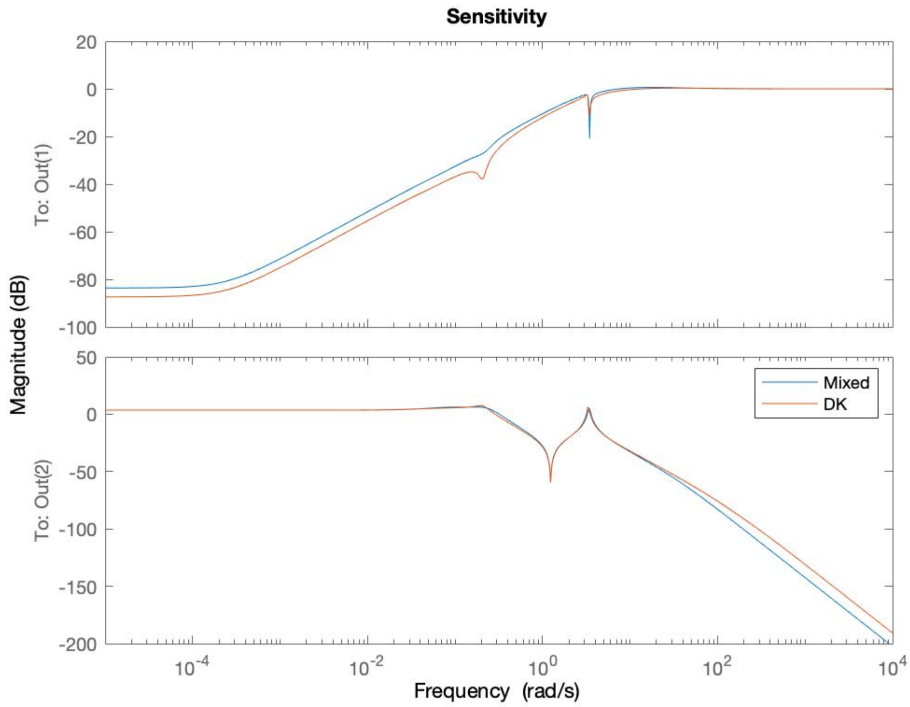

Figure 22: Sensitivity of the closed loop system with two controllers

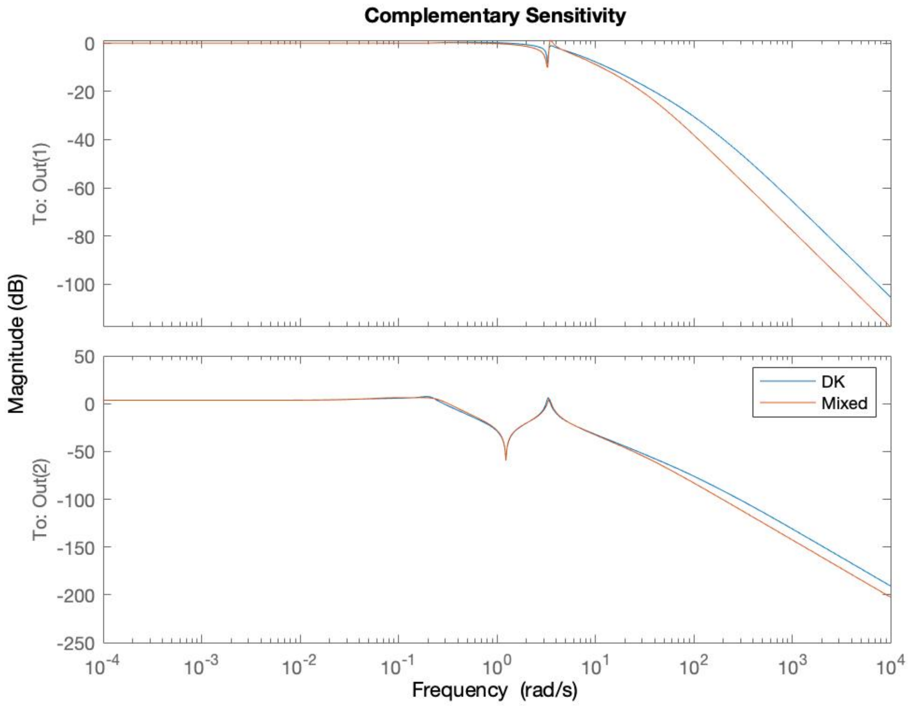

Figure 23: Complementary sensitivity of the closed loop system with two controllers

Table 2: Statistical information about the time domain performance with two controllers

Controller**(Reference tracking)**	Settling time (s)	Overshoot (%)[Peak]	Rise time (s)
Mixed(output ω_r)	10.3408	6.5772[1.0657]	0.6066
Mixed(output ζ)	**49.7440**	**38.7938[2.0867]**	**7.0787**
DK(output ω_r)	**5.7811**	**6.3495[1.0634]**	**0.4728**
DK(output ζ)	55.3156	39.7764[2.0954]	7.3866
Controller**(Disturbance rejection)**	Settling time (s)	Overshoot (%)	Peak
Mixed(output ω_r)	66.7380	$1.3377 * 10^5$	0.0460
Mixed(output ζ)	390.2331	216.0349	0.7249
DK(output ω_r)	**19.6290**	$\mathbf{4.5259 * 10^4}$	**0.0374**
DK(output ζ)	**102.0099**	**51.0720**	**0.3465**

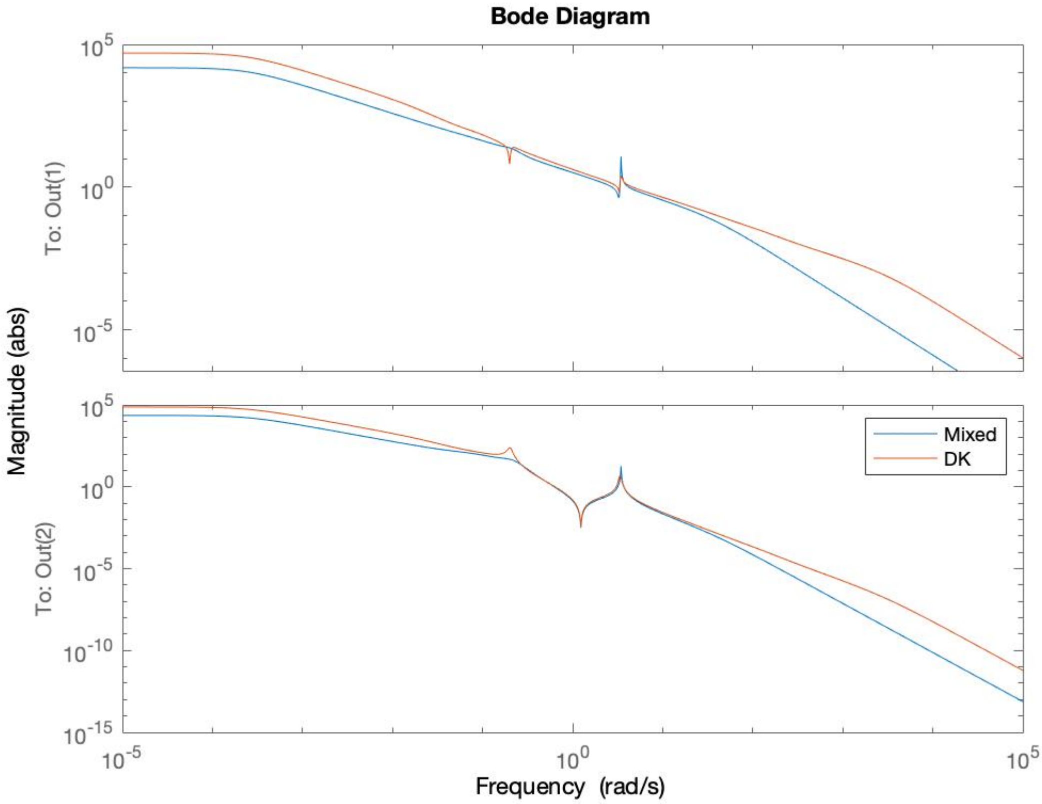

Figure 24: Bode magnitude plot of open loop transfer function with two controllers

Comparing the size of the controllers in frequency domain, it can be noticed that the DK controller has a larger order than the mixed-sensitivity controller. The DK controller has one of its transfer functions having a denominator with order 46, while the largest of the mixed-sensitivity controller has order of 8, which means that the DK controller will be more costly to implement in terms of required computational power.

References

[1] I. P. Siguard Skogestad, *Multivariable Feedback Control-Analysis and design.* John Wiley and Sons, Chichester, England, 2nd Edition, 2005.

[2] J. van Wingerden, *Lecture Slides on Robust and Multivariable Control [SC4015],.* Technische Universiteit Delft, 2020.

[3] G. Meinsma, *Unstable and non-proper weights in robust control.* Automatica, 1995.

[4] D. J. C. G. K. P. Balas, G. J., *μ-Analysis and Synthesis Toolbox User's Guide.* MathWorks, Natick, Mass., 1993.