Robust Inner Loop Control of a 6-DOF Rotary Wing Unmanned Aircraft System

MCEN 6228: Robust Multivariable Control Final Project Report

Mohammed Adib Oumer
mohammed.oumer@colorado.edu
University of Colorado Boulder

Abstract

In this project, I designed an H_∞ controller using the dynamics model of a 6-DOF Rotary Wing Unmanned Aircraft System with the objective of ensuring robust command tracking in the presence of structured multiplicative input uncertainty.

1 Introduction

For this project, I synthesized a series of inner loop controllers using H_∞ and μ-synthesis techniques for an 11-state rotary wing micro air vehicle shown in Figure 1 and analyzed their robustness properties. The state vector is given by $x = (u, v, p, q, \phi, \theta, a_s, b_s, w, r, r_{fb})^T$. The variables u, v, w are the vehicle translational velocities along the vehicle's longitudinal, lateral and vertical body axis, p, q, r are the vehicle roll, pitch and yaw rate respectively, ϕ and θ are the roll and pitch attitude, a_s and b_s are the rotor dynamics states (orientation of the rotor relative to the body) and r_{fb} is the washed out yaw rate, a filtering state associated with onboard yaw rate feedback. The control input vector is $u = (\delta_{long}, \delta_{lat}, \delta_{ped})^T$, corresponding to the longitudinal cyclic, the lateral cyclic and the tail rotor inputs. The outputs available for feedback are the pitch and roll attitude θ and ϕ, and the washed out yaw rate r_{fb}. If you are having a difficult time using only these outputs, you are allowed to include the pitch and roll rates q and p as well. The equations of motion linearized about the hover were provided beforehand.

In this project, I first formulated the control problem, design the controller for the nominal plant and finally tested robustness of the controller accounting for a structured multiplicative input uncertainty.

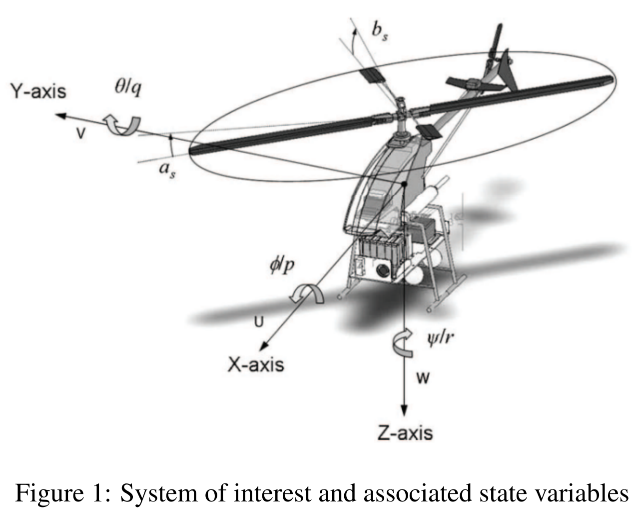

Figure 1: System of interest and associated state variables

2 Problem formulation

In this part, I formulated the generalized plant in terms of the block diagram with the relevant uncertainty and performance weighting functions to achieve the desired command tracking behaviour. This is shown in Figure 2.

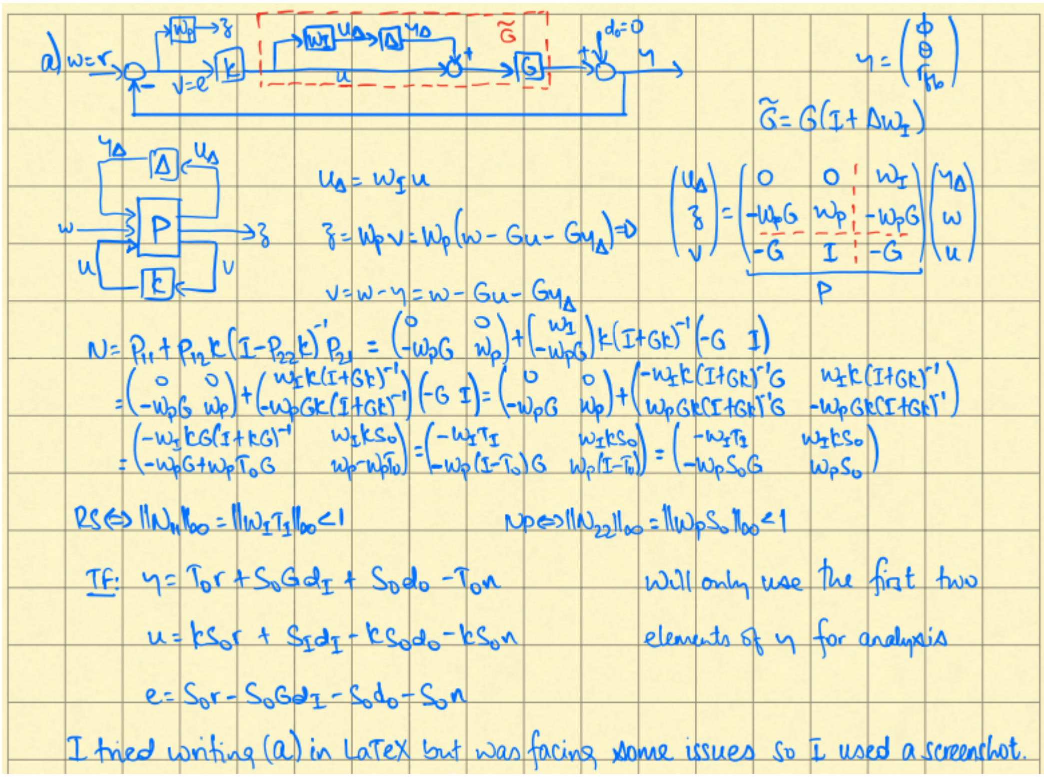

Figure 2: Generalized plant

3 Controller Synthesis

In this part, I synthesized an H_∞ controller for the nominal plant. Its performance was then analyzed using singular value plots and relevant step responses. The nominal plant (without the uncertainty components) is given by Equation 1.

$$\begin{pmatrix} z \\ v \end{pmatrix} = \begin{pmatrix} W_p & -W_pG \\ I & -G \end{pmatrix} \begin{pmatrix} w \\ u \end{pmatrix} \tag{1}$$

I was able to set up the nominal plant and use performance weighting function $w_p(s) = \frac{s/M + w_b}{s + w_b \times A}$ where $M = 2$, $w_b = 1$ and $A = 0.005$ across all three channels to synthesize the controller using MATLAB's `hinfsyn` command. M is related to the noise rejection/transient behaviour of the controller, w_b is the bandwidth and A is related to tracking error ($\frac{1}{A}$x attenuation). $\gamma = 0.5$ from the controller synthesis.

Figure 3 shows the open loop transfer function singular value plots of G and GK. We know from the objective that we want reference tracking (i.e. minimize tracking error) and from (a) above, we saw that the nominal performance is tested via minimization of $\overline{\sigma}(S_o)$ at low frequencies. From class, we know that the open loop equivalent of this constraint is maximizing $\underline{\sigma}(GK)$ (i.e. $\underline{\sigma}(GK) \gg 1$ or $\underline{\sigma}(GK) \gg 0dB$). This behaviour is seen on the open loop singular value plot where we see larger gain at lower frequencies in GK compared to G.

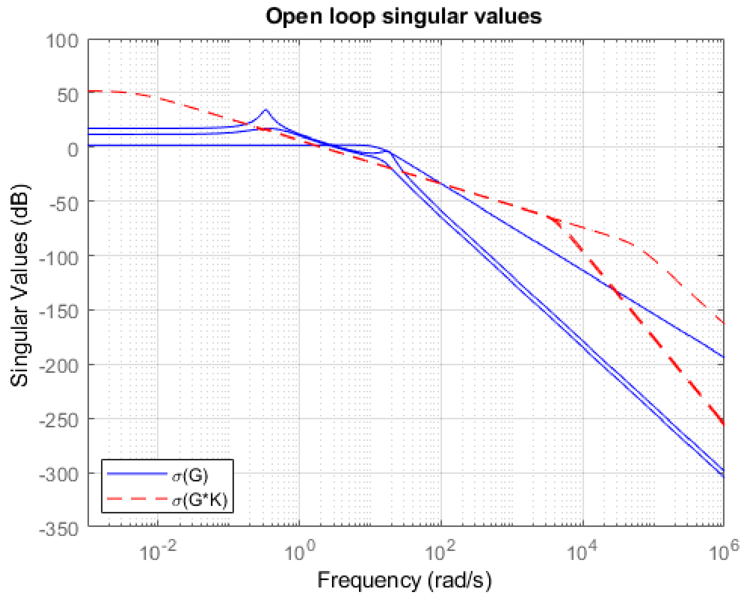

Figure 3: Open Loop Singular Value Plots

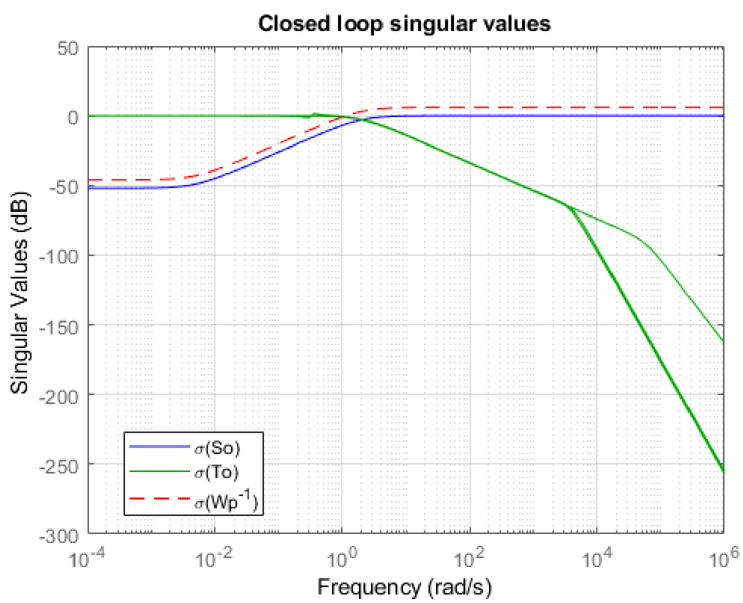

Figure 4: Closed Loop Singular Value Plots

Figure 4 shows the closed loop transfer function singular value plots of $S_o = (I + GK)^{-1}$, $T_o = GK(I + GK)^{-1}$ and $1/|w_p(s)|$. $1/|w_p(s)|$ is an upper bound on S_o and its maximum value is also an upper bound on T_o, which is exactly what we see in the plot. We also wanted S_o to be small at low frequencies, which is what we see. Moreover for reference tracking, $\overline{\sigma}(T_o) = \underline{\sigma}(T_o)$ as is evident in the plot. Hence, this plot verifies that nominal performance is satisfied with the designed controller.

The transfer function $r = w \rightarrow y$ is T_o and $d_o \rightarrow y$ is S_o (which is the same as our objective $w \rightarrow e = v$). We are only interested on the first two elements of y (i.e. ϕ and θ) for the performance evaluation. We plotted step responses using the appropriate transfer functions for the relevant

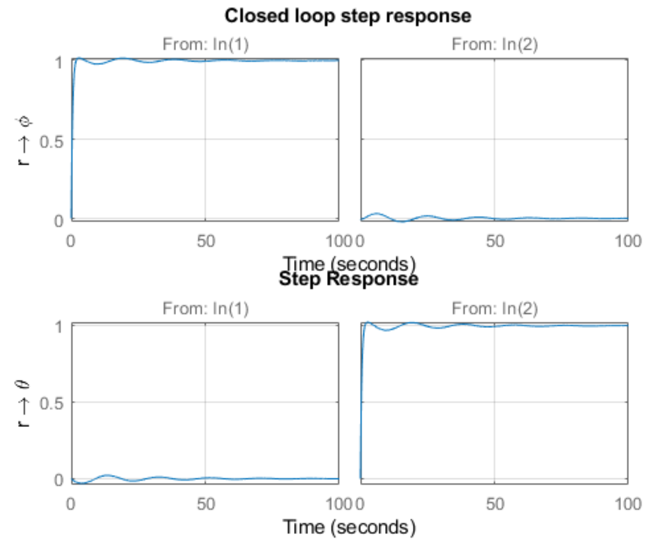

Figure 5: Step responses of $w \rightarrow y$

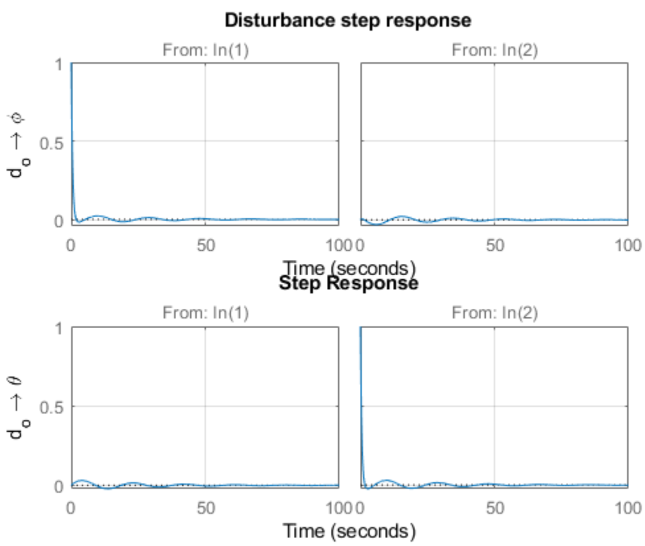

Figure 6: Step responses of $d_o \rightarrow y$

elements. Figure 5 shows $w \rightarrow y$ and Figure 6 shows $d_o \rightarrow y$. In both cases, the transfer function with respect to the third element of w or d_o was always zero so I chose not to show it. For better visualization, I selectively show the interesting step responses from each transfer function in Figures 7 and 8. Figure 7 clearly shows that we are able to track the reference as needed ($r_1 = \phi_{ref} \rightarrow \phi$ and $r_2 = \theta_{ref} \rightarrow \theta$) with minimal tracking error. The bandwidth is set at $1rad/s$. The step responses showed $t_r \approx 3s$ which is an acceptable value for the given case (compared to $1.8s$ expected if the system was second order). The maximum overshoot was around 2%. Figure 8 shows that we are able to attenuate/reject the output disturbance sufficiently (and since the transfer function is the same as $w \rightarrow e$, we have successfully minimized the error signal). However it seems that the settling time is longer in both cases, which perhaps indicates that the M value is accommodating a bit of flexibility on how fast the system reaches final value.

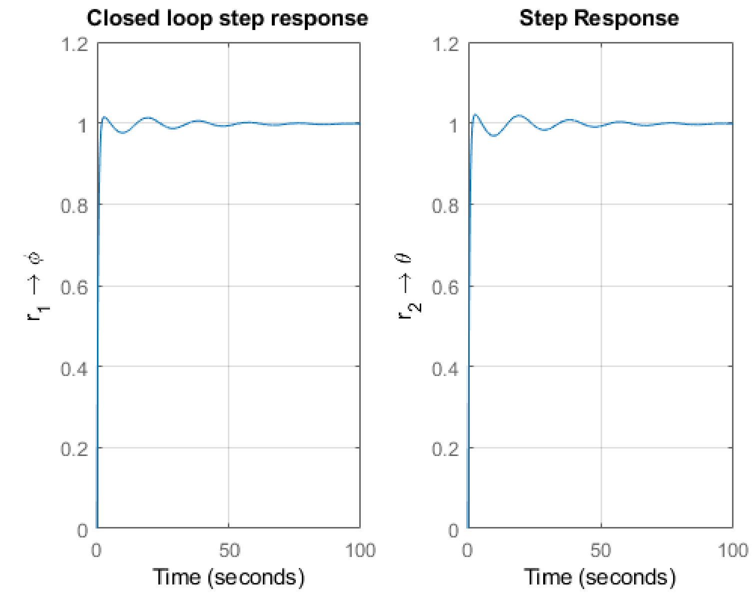

Figure 7: Selected closed loop step responses of $w \to y$

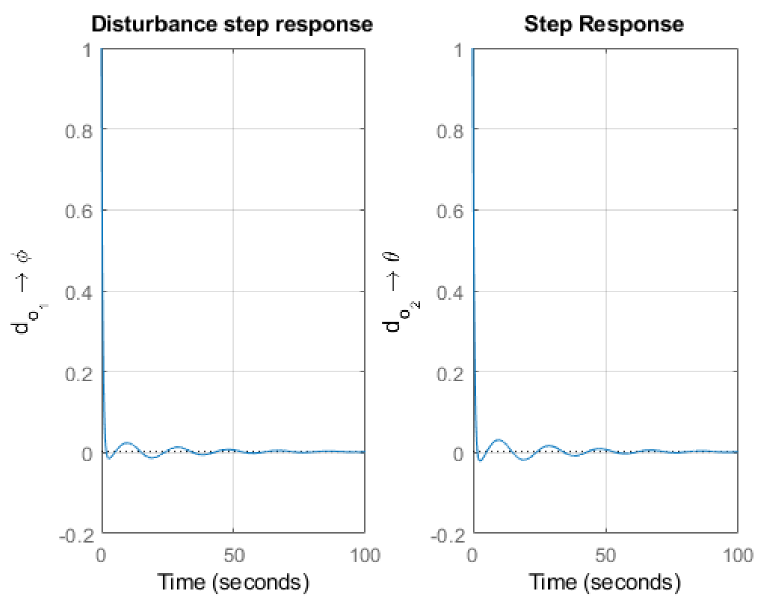

Figure 8: Selected disturbance step responses of $d_o \to y$

4 Initial Robustness Tests

After designing the controller, I tested for robust stability and robust performance using the $N - \Delta$ form shown in Figure 2. I already showed how the nominal performance is satisfied in the previous section using H_∞ norm of N_{22}.

For the robust stability, since we are considering a structured multiplicative input uncertainty, the structured singular value $\mu(N_{11}) = \mu(W_i T_i)$ was used. Computing the actual value is an NP-problem so it was approximated by using MATLAB to compute the lower bound in the spectral radius and

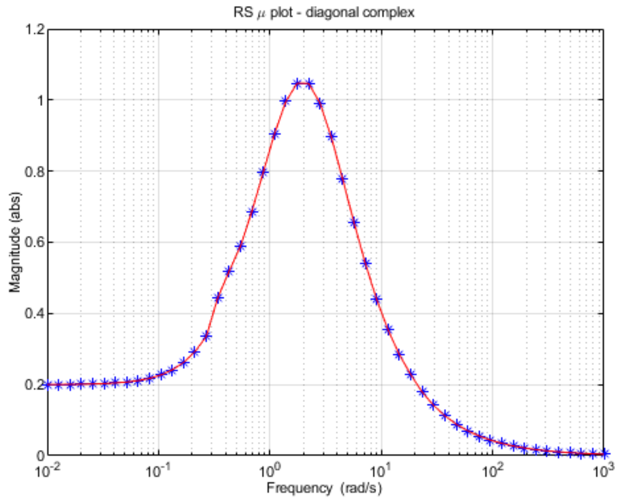

Figure 9: Robust stability μ plot

the upper bound in the spectral norm (i.e. $\rho(N_{11}) \leq \mu(N_{11}) \leq \overline{\sigma}(N_{11})$). For the controller system to be robustly stable, $\mu < 1$ for all frequencies. The uncertainty weight considered was a diagonal structured complex uncertainty with relative weight of $w_i(s) = \frac{s+0.2}{0.5s+1}$ across all channels. An assumption here was that $\|\Delta\|_\infty < 1$. Figure 9 shows the robust stability plot where the lower and upper bounds overlap. Since the values are not always less than 1 over all frequencies, the system is not robustly stable. The stability margin is given by $1/\mu_{peak} = 1/1.05 = 0.952$.

Robust performance is measured using $\mu(N)$ and $\mu < 1$ for robust performance. Figure 10 shows the robust performance plot where the lower and upper bounds overlap. Again we see that since the values are not always less than 1 over all frequencies, the system does not satisfy robust performance. The performance margin is given by $1/\mu_{peak} = 1/1.55 = 0.645$.

Since we do not satisfy both robust stability and robust performance, I tried to achieve both using another iteration with modified performance and uncertainty weights. After some try and error, I was

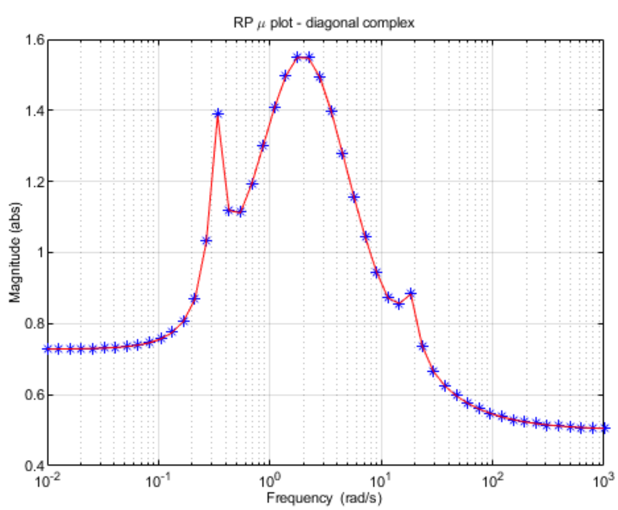

Figure 10: Robust performance μ plot

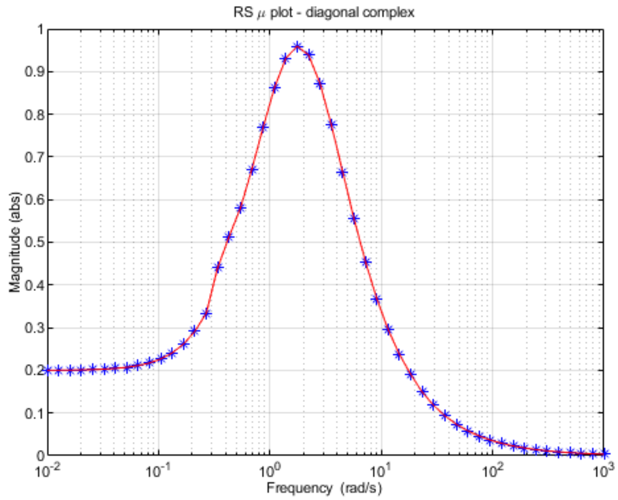

Figure 11: Improved robust stability μ plot

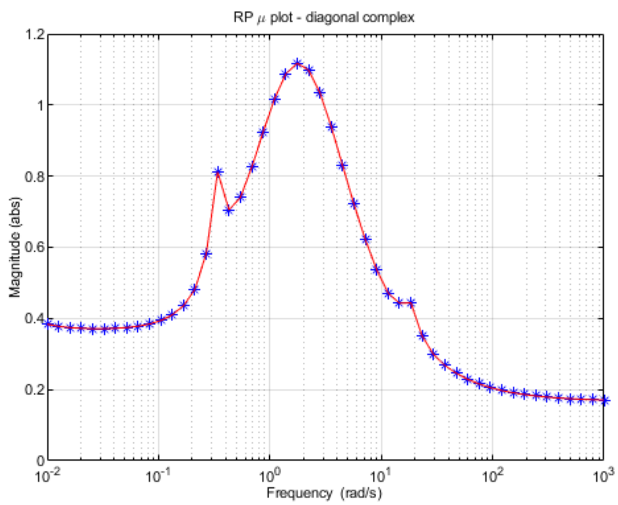

Figure 12: Improved robust performance μ plot

able to achieve robust stability using updated $w_i(s) = \frac{s+0.2}{0.6s+1}$ in all channels, as shown in Figure 11. The improved stability margin in this case was $1/\mu_{peak} = 1/0.957 = 1.045$. I also adjusted to $w_p(s) = \frac{s/M+w_b}{s+w_b \times A} = \frac{s/6+0.3}{s+0.3\times 0.005}$ (increased M and decreased w_b compared to first iteration) which reduced the peak of $\mu(N)$ with improved performance margin $1/\mu_{peak} = 1/1.12 = 0.893$, as shown in Figure 12. Yet, robust performance is not achieved so I will use the D-K iteration technique in the next section to achieve robust performance. Here, the controller K and plant G have not changed so all the transfer functions do not change and thus, the step response plots in this second iteration are still the same as ones shown in Figures 5-8. The difference here is that I increased M and decreased w_b while keeping A the same. This means that we can allow a more lax transient behaviour such as with rise time while keeping tracking error expectations the same. Therefore, our expectations here are more relaxed compared to the first iteration.

5 D-K Iteration

The idea here is to minimize the peak value of $\mu(N)$ that is generally given as a collective minimization problem over a scaling matrix D and controller K. The minimization over each variable is convex, which leads to an optimal solution but the joint minimization problem is not necessarily convex. Therefore, we use iteration of minimization over each variable D and K to find an optimal controller. There are two approaches of getting the controller using this μ-synthesis approach on MATLAB after closing the upper LFT: automatically using `musyn` function and manually using manual settings. This was done by building up on the result and parameters of the second iteration from the previous section.

5.1 Automatic D-K Iteration

Figure 13 shows the upper bound of the structured singular value of the system obtained after closing the lower LFT (to get N) using the controller obtained from the automatic D-K iteration. As we can see the peak is 0.816, leading to performance margin of 1.225, which is much better than the previous versions discussed above and it also satisfies robust performance over all frequencies since the peak is less than 1.

I then computed the open loop and closed loop transfer functions using the controller I obtained. The singular value magnitude plots for each are shown in Figures 14 and 15 respectively. From both plots, we can see the same characteristics that were identified as in ''Controller Synthesis'' section. In Figure 15, we can see that the peak of $1/|w_p(s)|$ is higher than the one in Figure 4 because of the increase in M and the bandwidth value (measured by looking at frequency at which the singular value is $0dB$) is 0.3 as expected from decreasing w_b. Thus, this controller has satisfied nominal performance as well.

Figures 16 and 17 show the step responses for the closed loop transfer function (as shown earlier in Figures 5 and 6). Figures 18 and 19 show the step responses for the selected relevant closed loop transfer function and disturbance for better visualization (as shown earlier in Figures 7 and 8). In all cases, we can see zero tracking error, maximum of 4% overshoot percentage, effective disturbance attenuation and faster settling times as desired.

5.2 Manual D-K Iteration

Here, I used manual settings to implement the D-K iteration. I used third order fit for the scaling factors and I was able to only go for about 6 iterations without issue. For any iteration past 6, I was getting errors about the `hinfsyn` function not finding any stabilizing controllers, perhaps mainly because the scaling matrix was not fit properly. In any case, for the initial iteration, I started with an identity matrix as the scaling matrix and implemented the D-K iteration by first finding a controller

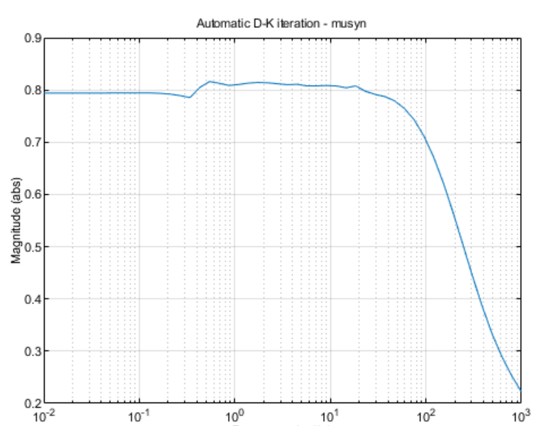

Figure 13: Robust performance μ plot - automatic D-K

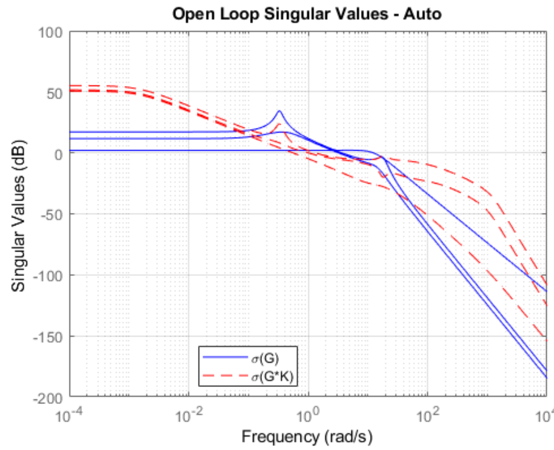

Figure 14: Open Loop Singular Values

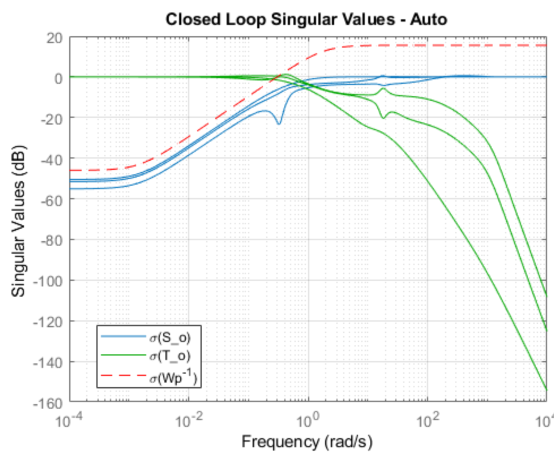

Figure 15: Closed Loop Singular Values

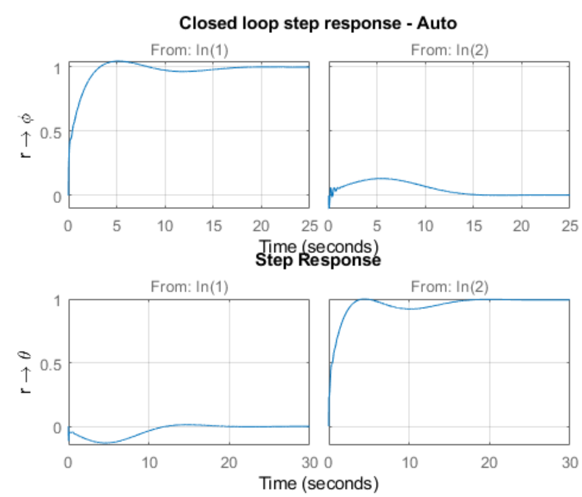

Figure 16: Closed Loop Step Responses

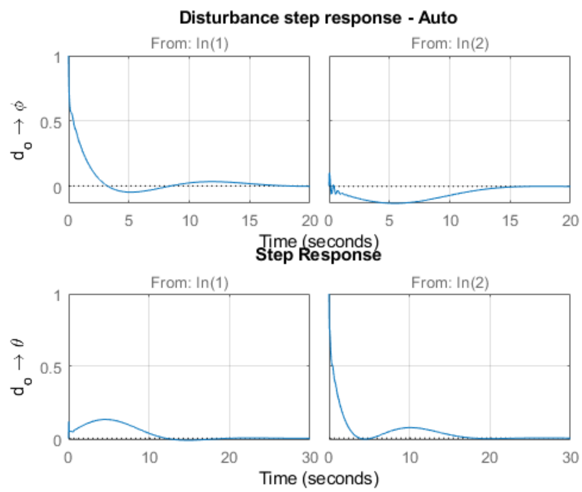

Figure 17: Disturbance Step Responses

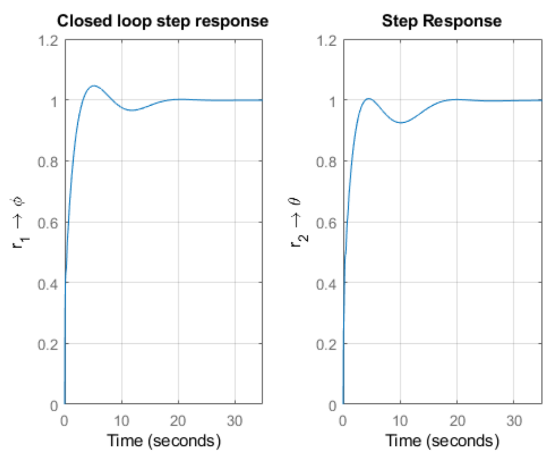

Figure 18: Relevant Closed Loop Step Responses

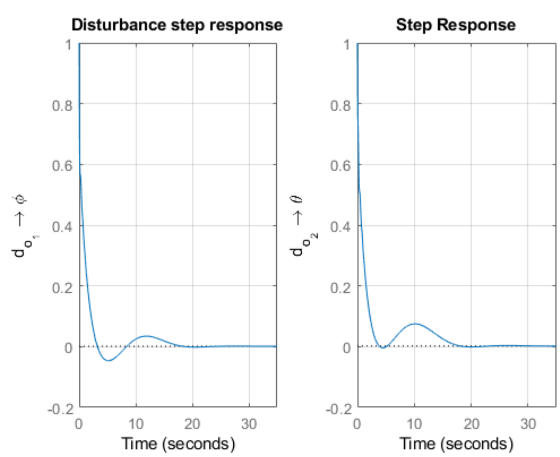

Figure 19: Relevant Disturbance Step Responses

and using that controller to find the next scaling matrix D. The robust performance μ plot for the first and final (sixth) iteration is shown in Figure 20. Figure 21 shows the peak of μ as compared to the γ value obtained via the `hinfsyn` function. From here, we can see that the peak of μ is generally upper bounded by γ as expected but seems to surpass it as iterations go on. The peak from the final iteration was 0.983 leading to a performance margin of 1.017, which is worse than the automatic case. This could be perhaps because the fit for the scaling was made to be third order (and larger orders were stopping the iteration even before the sixth iteration) or perhaps because the peak of μ seems to increase while γ decreases as the iteration increases and iteration is stopped once peak of μ is larger than γ. Regardless, the final iteration still satisfies robust performance with peak of μ being less than 1. Using the controller from the final iteration, I produced several plots. Figures 22 and 23 show the open loop and closed loop transfer function singular value plots. They exhibit the same characteristics as the ones shown before and we can confirm that nominal performance is satisfied. Figures 24-27 show the step responses similar to all the ones before. Here, we have zero tracking error, maximum of 6% overshoot percentage, effective disturbance attenuation and faster settling times as desired.

Finally, one major observation is the tradeoff between robust performance and high bandwidth in the presence of the uncertainty. Earlier in ''Initial Robustness Test'' section, we saw that the robust performance index was improved when I reduced the bandwidth. Comparing Figures 15 and 23, one can see that the former has a relatively smaller bandwidth (≈ 0.2rad/s), longer rise time and better robust performance and the latter has a relatively larger bandwidth (≈ 0.32rad/s), faster rise time and relatively worse robust performance.

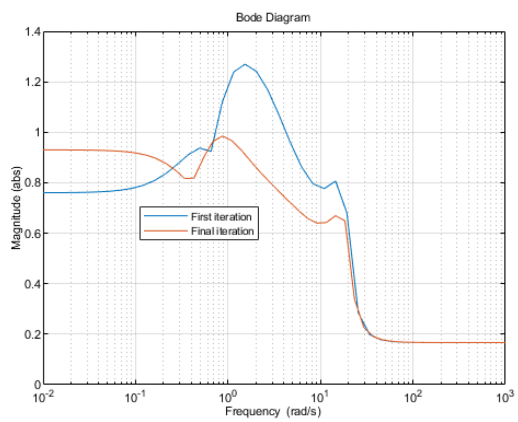

Figure 20: Robust performance μ plot - manual D-K

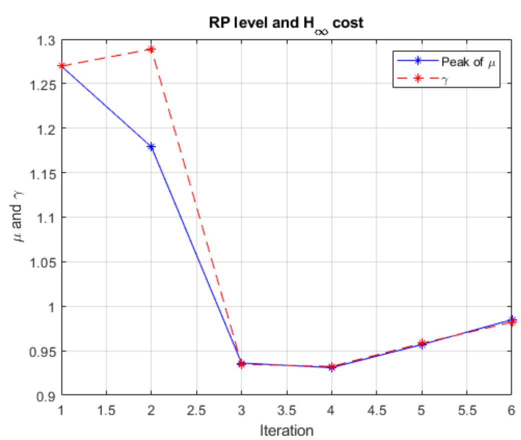

Figure 21: Peak of μ and γ vs Iteration

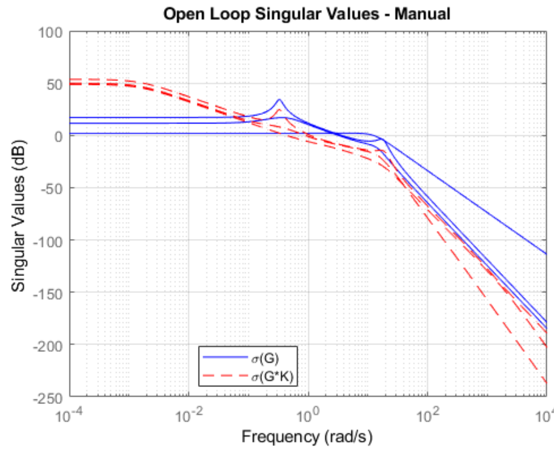

Figure 22: Open Loop Singular Values

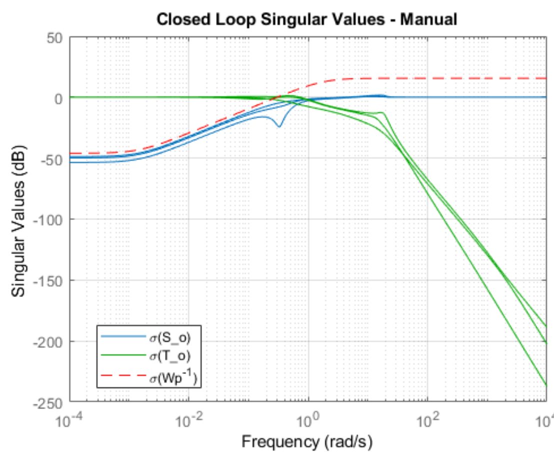

Figure 23: Closed Loop Singular Values

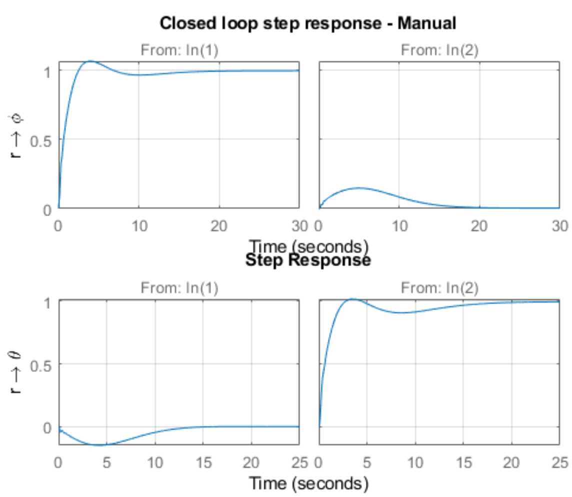

Figure 24: Closed Loop Step Responses

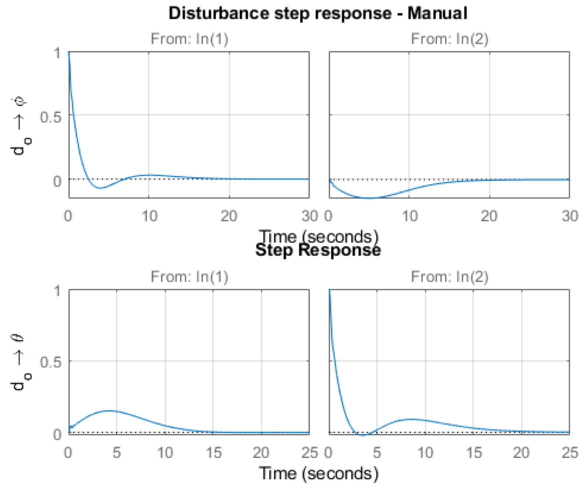

Figure 25: Disturbance Step Responses

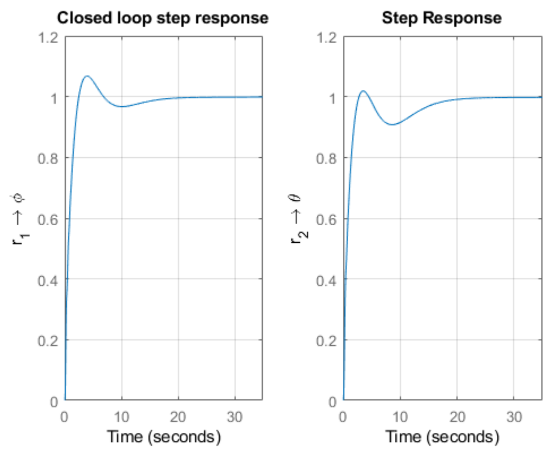

Figure 26: Relevant Closed Loop Step Responses

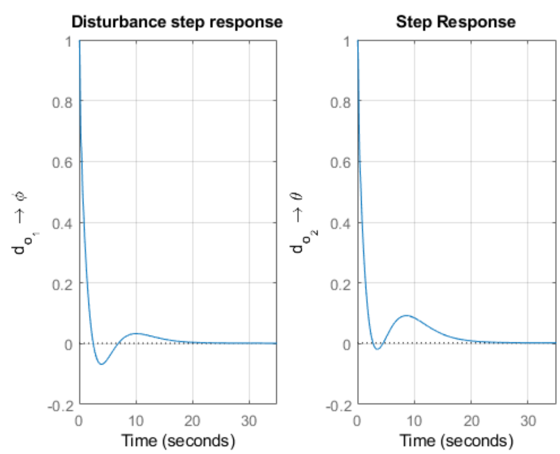

Figure 27: Relevant Disturbance Step Responses