Robust Inner Loop Control of 6-DOF Rotary Wing UAV System

Bethany Calvert

May 9, 2023

1 Generalized Plant and Uncertainty and Performance Weighting Functions

The block diagram for the general dynamics, weighting uncertainty, and weighting performance functions is displayed in Fig. 1. The structured uncertainty is multiplicative input uncertainty.

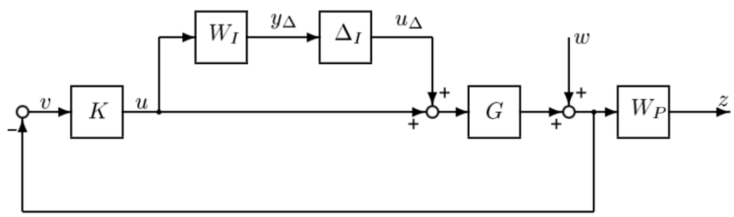

Figure 1: Block diagram structure with structured, multiplicative input uncertainty and weighting performance functions.

The initial weighting functions are as follows:

$$W_P = \frac{\frac{s}{2}+1}{s+.005} * \mathbf{I_{3x3}} \tag{1}$$

$$W_I = \frac{2s+8.704}{s+87.04} * \mathbf{I_{3x3}} \tag{2}$$

The weighting performance function, W_P, is constructed with the parameters set as, $A = .005$, $M = 2$, and $\omega_b* = 1$. Additionally, the weighting uncertainty function, W_I, is made by using the command $makeweight(.1, 50, 2)$.

The generalized plant has the following structure:

$$\begin{bmatrix} u_\Delta \\ z \\ v \end{bmatrix} = \begin{bmatrix} 0 & 0 & W_I \\ W_P G & W_P & W_P G \\ -G & -I & -G \end{bmatrix} \begin{bmatrix} y_\Delta \\ w \\ u \end{bmatrix} \tag{3}$$

Additionally, the input, u, and output, v, vectors are as follows:

$$u = \begin{bmatrix} \delta_{long} \\ \delta_{lat} \\ \delta_{ped} \end{bmatrix}, v = \begin{bmatrix} \theta \\ \phi \\ r_{fb} \end{bmatrix} \tag{4}$$

The lower LFT while $\Delta = 0$ is:

$$\begin{bmatrix} z \\ v \end{bmatrix} = \begin{bmatrix} W_P & W_P G \\ -I & -G \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix} \tag{5}$$

Putting the system into $N - \Delta$ form, the N matrix is:

$$N = \begin{bmatrix} -W_I T_I & -W_I K S_O \\ W_P S_I G & W_P S_O \end{bmatrix} = \begin{bmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{bmatrix} \tag{6}$$

Here, the robust stability (RS) test is N_{11}, and the nominal performance test is N_{22}.

2 H_∞ Controller Synthesis and Nominal Performance

The initial open loop singular value plots of the system plant and its controller are displayed in Fig. 2.

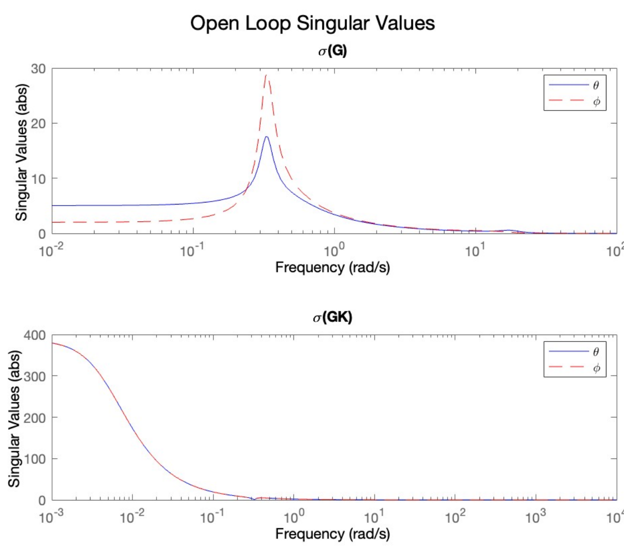

Figure 2: Open Loop Singular Value Plots

Initially, the system is analyzed with no uncertainty, so that the nominal performance of the system can be inspected. Looking at the plot of $\sigma(G)$, the tracking needs to be greatly improved. The initial tracking error is about 18% for the roll angle and 37.5% for the pitch angle up to .1rad/s. Additionally, the nominal plant bandwidth is at about 3.55rad/s. Additionally, the disturbance rejection of the system is suitable. Looking at the plot of $\sigma(GK)$, the tracking error is significantly improved to about 3% for the roll angle and 1.5% for the pitch angle up to .1rad/s. Initially, the controller gains look very high in the low frequency, which could lead to implementation issues with the controller physically. The bandwidth is slightly raised to about 3.74rad/s, but it is still very close to the nominal plant's bandwidth. The disturbance rejection is 2X greater than the rejection in the nominal plant as well.

The RS test is displayed in Fig. 3. This test is a singular value plot of the $N_{11} = -W_I T_I$ term from the prior section.

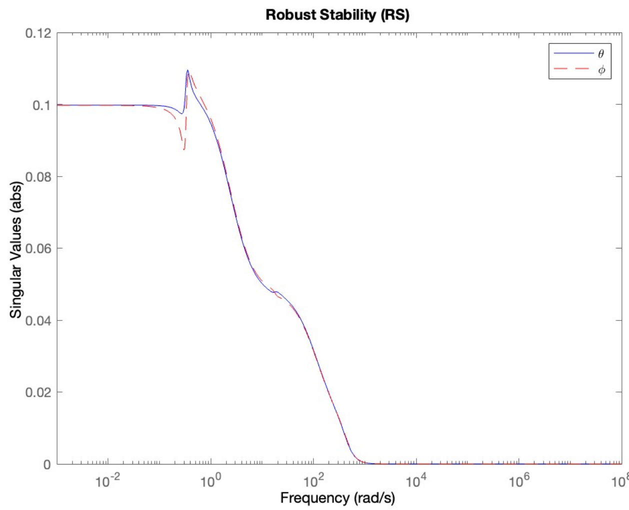

Figure 3: Robust Stability Test

Here, the singular values stay below one. Therefore, the RS condition is satisfied through satisfaction of the small gain theorem (SGT). Further, the maximum singular value of the system N is less than one.

Next, the closed loop transfer function singular values of S_o and the weighting performance function singular values of W_P are plotted in Fig. 4.

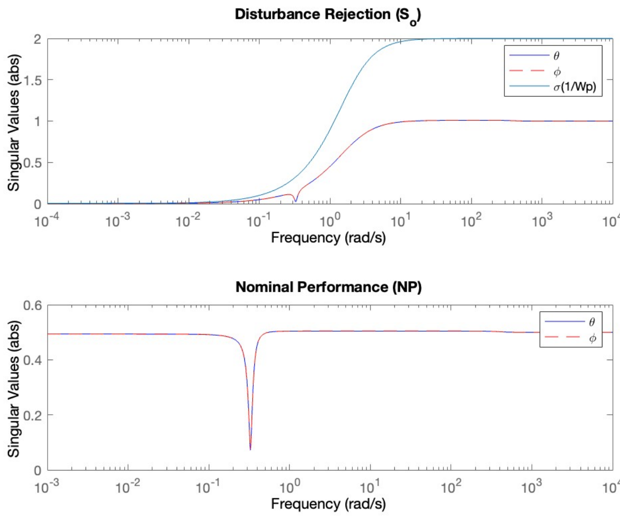

Figure 4: Disturbance Rejection and Nominal Performance Singular Values

First, it is seen that the nominal performance is achieved because S_o is below $1/W_P$, and $N_{22} = W_P S_o$ is below one.

Additionally, the singular values of the CL TF T_o are plotted in Fig. 5. T_o represents the tracking performance of the system.

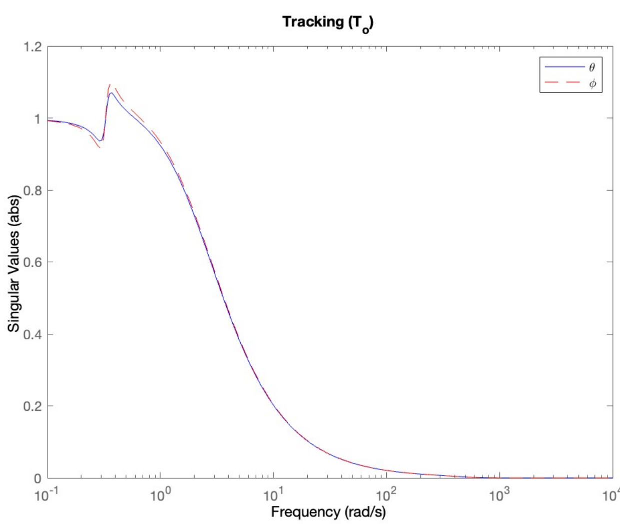

Figure 5: Tracking Performance Singular Values

In this plot, the bandwidth is .322rad/s. Here, we see that the bandwidth is slightly slower in the closed loop, which should allow for the controllers to track the performance requirements successfully in real time.

The step responses of the pitch and roll angles and output disturbances to each output angle are displayed in Fig. 6.

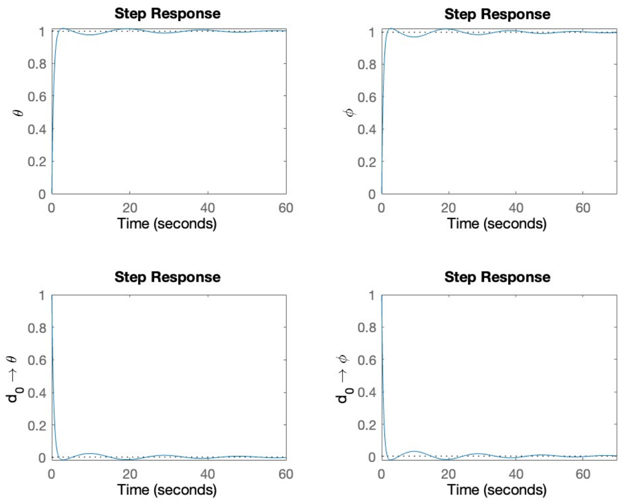

Figure 6: Step Response of Ouput States and Output Disturbances

The top two responses represent the tracking behavior of the pitch and roll following a step input. The responses have a settling time of around 11 and 20 seconds, and overshoot values of around 2%. Overall, this means that the system should track fairly well since the overshoot is low, but it tracks relatively slowly. In the output disturbances, the rise time is extremely fast. Additionally, the settling time is longer/not defined because the controller needs to damp out the oscillations from the disturbance. However, by inspection we see that the disturbance rejection works very fast despite the large overshoot.

3 Robust Performance

Next, a new weighting uncertainty function is introduced:

$$W_I = \frac{s + .2}{.5s + 1} * \mathbf{I_{3x3}} \tag{7}$$

Next, assuming $||\Delta||_\infty = 1$, the upper and lower μ bounds are determined. Using the new uncertainty, the RS and NP are calculated and plotted in Fig. 7.

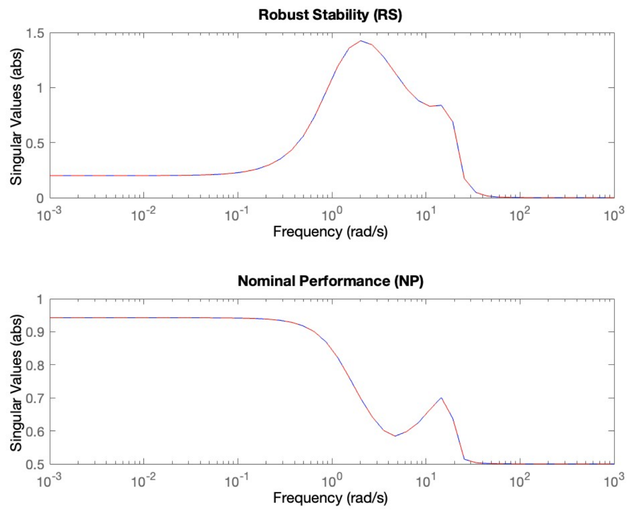

Figure 7: Robust Stability and Nominal Performance with Uncertainty

Here, we can see that the system is not robustly stable, as some of the sigma values are greater than one. In the RS plot, the peak value is at 1.42. This results in a stability margin of $||\Delta||_\infty =$ .704. This means that the system can only tolerate 70% of the current uncertainty permitted in the system before it becomes unstable. The robust performance of the system is plotted in Fig. 8.

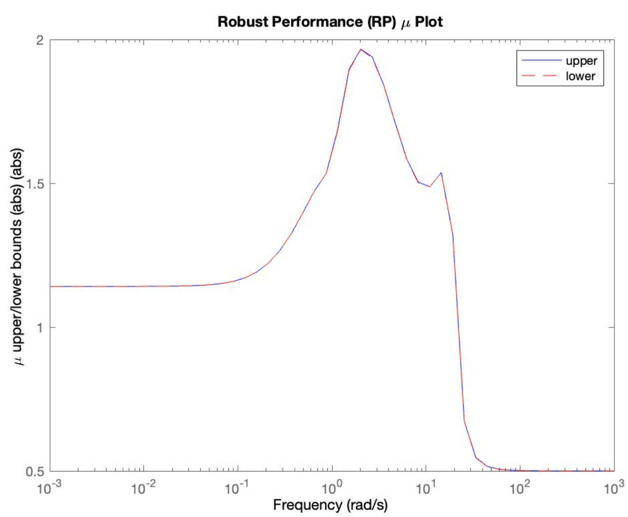

Figure 8: Robust Performance with Uncertainty

Additionally, here the robust performance is not satisfied. The peak value of the RP plot is at 1.96. This results in a stability margin of $||\Delta||_\infty = .51$. Further, only 51% of the current uncertainty can be tolerated while still satisfying the RP criteria. This makes sense as it is a more stringent condition than RS, which was not satisfied either. The step responses of the roll and pitch angles and the corresponding output disturbances are plotted in Fig. 9.

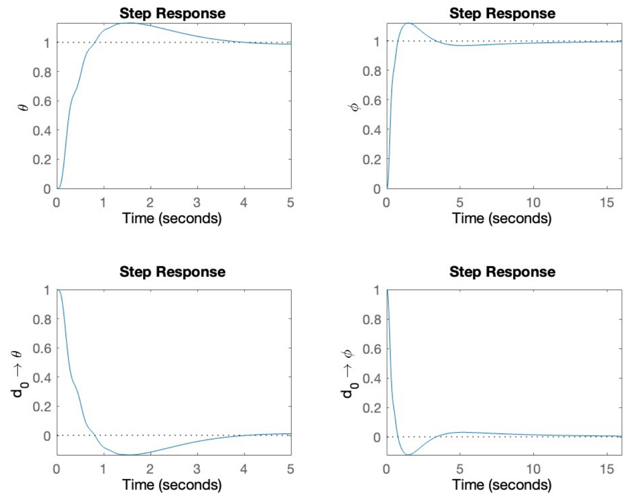

Figure 9: Step Response of Pitch and Roll Angles and Output Disturbances with Uncertainty

Since the results above did not provide RS or RP for the system, iterations on the weighting performance function are completed.

The new weighting performance function is

$$W_P = \frac{\frac{s}{3.2} + .1}{s + .005} * \mathbf{I_{3x3}} \tag{8}$$

Here, we have increased the noise rejection constraint to $M = 3.2$ and move the bandwidth to $w_b^* = .1$. The new RS plot is displayed in Fig. 10.

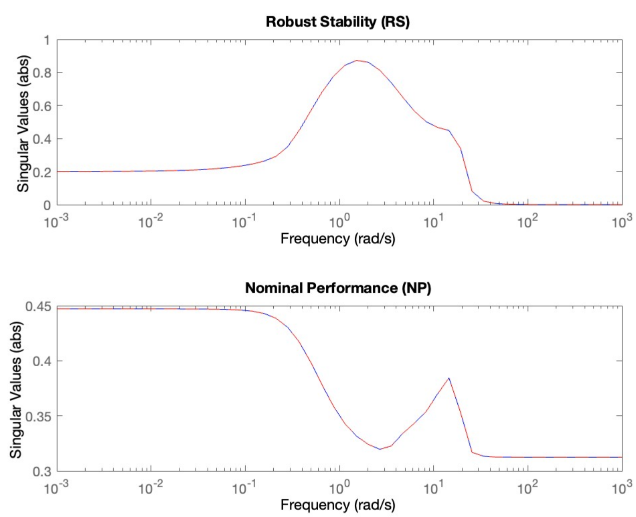

Figure 10: Robust Stability and Nominal Performance with Uncertainty

Now, the system is RS with less stringent weighting performance function requirements. The peak value occurs at .873. Further the stability margin, $||\Delta||_{\infty} = 1.14$. Therefore, including the current amount of uncertainty, the system allows 14% more uncertainty from the current system control design before it induces instability. The RP plot for the system is displayed in Fig. 11.

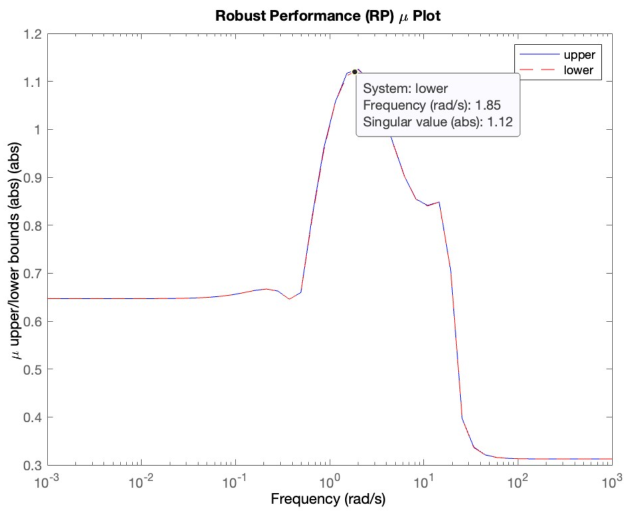

Figure 11: Robust Stability and Nominal Performance with Uncertainty

However, the system still does not adhere to the RP constraints. The peak value of the system is at 1.12. Further the stability margin is $||\Delta||_{\infty} = .893$. Further, the system can only allow for 89% of the uncertainty margin introduced into the system. The step responses for the updated performance function are displayed in Fig. 12.

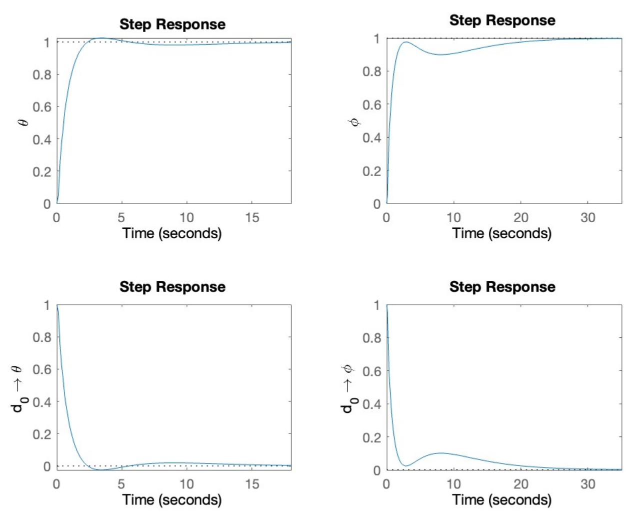

Figure 12: Step Response of Pitch and Roll Angles with Uncertainty Iteration

In these plots, the consequences of designing with less stringent performance parameters becomes clear. The slower bandwidth leads to a much slower response in both the tracking and output disturbance responses.

4 D-K Iteration

As an alternative approach to improve the RP of the controller without sacrificing the performance parameters as much as in the prior section, a DK iteration is implemented during this analysis. The original weighting performance function from Eq. 1 is implemented. The plot of ten iterations is displayed in Fig. 13.

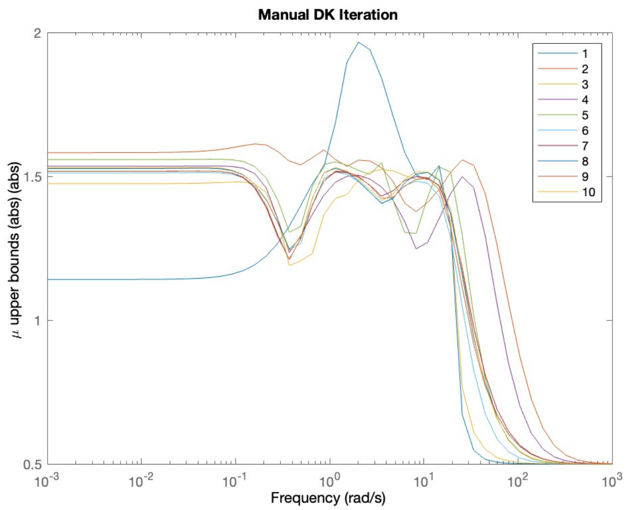

Figure 13: Ten DK-Iterations of RP

Additionally, the robust performance level and H_∞ cost are plotted in Fig. 14.

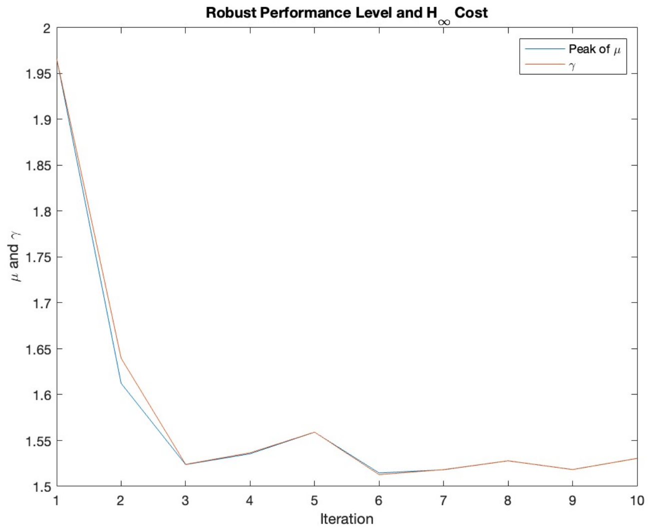

Figure 14: Performance Level and H_∞ Cost Comparison

Looking at this plot, the performance level and cost are lowest at iteration six. Additionally, it is confirmed that enough iterations have been plotted because the cost seems to have leveled out in this plot. The optimal DK iteration (6) RS and NP is plotted in Fig. 15.

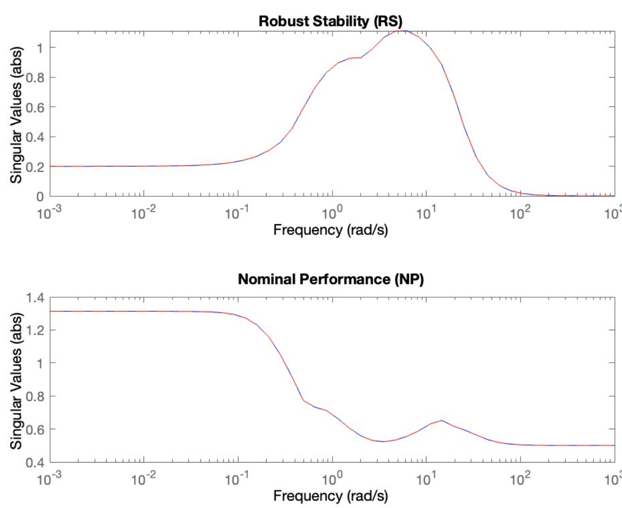

Figure 15: Optimal DK-Iteration RS and NP

The system still is not RS. However, the peak value has been reduced to 1.11. This results in a stability margin of $||\Delta||_\infty = .9$. Therefore, now the system can handle 90% of the proposed uncertainty while still remaining stable. The optimal DK iteration RP is plotted in Fig. 16.

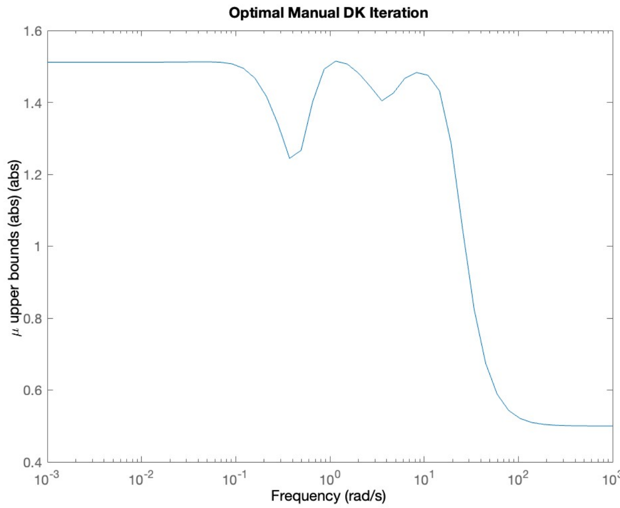

Figure 16: Optimal DK-Iteration RP

In this iteration, the system still does not satisfy the RP requirements. However, the peak value of the RP has been reduced to 1.51. The stability margin is $||\Delta||_\infty = .662$. Therefore, we have increased the RP to this uncertain system by about 15% to 66%. The step response of the tracking and output disturbance parameters is displayed in Fig. 17.

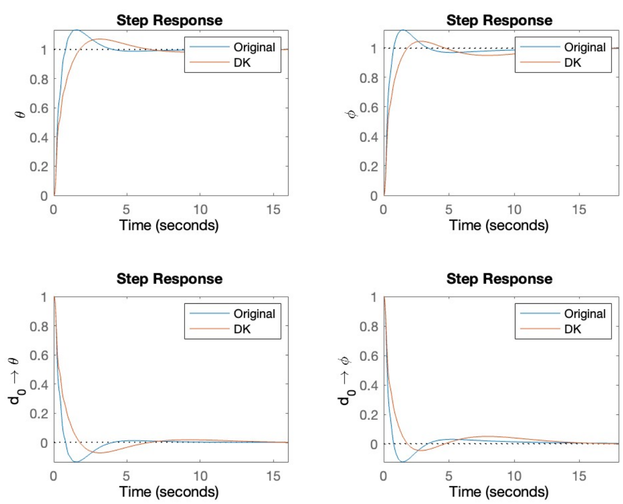

Figure 17: Step Response of Output Disturbances with Uncertainty for DK-Iteration

Here, we can see that the overall, the DK-iteration does make the step response slower. However, these step responses are much improved compared to those in Fig. 12.

5 Conclusion

Overall, this report analyzes a system that does not reach RS and RP without compromise. Once uncertainty is introduced into the system, the performance and stability of the system is not easily achieved. The weighting performance function parameters can be changed so that the system achieves RS and RP. However, sacrificing these parameters very much does slow the response, which physically would lead to problems for a drone during flight. Slow reaction times can cause the drone to crash during flight, as other environmental factors and reactions can be exasperated during these controller settling time periods. An alternative control method for this system is the D-K iteration. Here, the RS and RP can be greatly improved without having to make as great of a sacrifice in the speed of the system control.

6 MATLAB Code

```
clc, clear all, close all, format compact

% Rotorcraft dynamics
A = [-0.1778,zeros(1,4),-9.7807,-9.7807,zeros(1,4);
     0,-0.3104,0,0,9.7807,0,0,9.7807,zeros(1,3);
     -0.3326,-0.5353,zeros(1,4),75.7640,343.86,zeros(1,3);
     0.1903,-0.294,zeros(1,4),172.62,-59.958,zeros(1,3);
     0,0,1,zeros(1,8);zeros(1,3),1,zeros(1,7);
     zeros(1,3),-1,0,0,-8.1222,4.6535,zeros(1,3);
     0,0,-1,zeros(1,3),-0.0921,-8.1222,zeros(1,3);
     zeros(1,6),17.168,7.1018,-0.6821,-0.1070,0;
     0,0,-0.2834,zeros(1,5),-0.1446,-5.5561,-36.674;
     zeros(1,9),2.7492,-11.1120];

B = [zeros(6,3);
     0.0632,3.339,0;             % delta_lat (roll motions)
     3.1739,0.2216,0;            % delta_long (pitch motions)
     zeros(1,3);
     0,0,-74.364;                % delta_ped (yaw motions)
     zeros(1,3)];

C = [zeros(1,4),1,zeros(1,6);    % phi
     zeros(1,5),1,zeros(1,5);    % theta
     zeros(1,9),0,1];            % r_fb

D = zeros(size(C,1),size(B,2));

G = ss(A,B,C,D);

%% A
p = sigmaoptions("cstprefs") ;
p.MagUnits = 'abs' ;

So = minreal(inv(eye(3)+G)) ;
```

```matlab
To = minreal(G*inv(eye(3)+G)) ;
Ti = minreal(G*inv(eye(3)+G)) ;
Si = minreal(inv(eye(3)+G)) ;


% Performance Functions
A = .005 ;
M = 2 ;
wb = 1 ;
wp = tf([1/M wb],[1 wb*A]) ;
Wp = wp*eye(3) ;

% Uncertainty Function
Wi = tf(makeweight(.1,50,2))*eye(3) ;
% Wi = tf([1 .2],[1 50])*eye(3) ;

% Nominal Plant
systemnames = 'G Wp' ;
inputvar = '[w(3); u(3)]' ;
outputvar = '[Wp; -G-w]' ;
input_to_G = '[u]' ;
input_to_Wp = '[w+G]' ;
sysoutname = 'P' ;
sysic ;
P = minreal(ss(P)) ;


%% B: Nominal Performance
n_meas = 3 ;
n_ctrl = 3 ;
w = logspace(-3,3) ;

[K,CL,gamma,info] = hinfsyn(P,n_meas,n_ctrl) ;
So = minreal(inv(eye(3)+G*K)) ;
To = minreal(G*K*inv(eye(3)+G*K)) ;
Ti = minreal(K*G*inv(eye(3)+K*G)) ;
Si = minreal(inv(eye(3)+K*G)) ;

GK = G*K ;


% Open Loop Sigma Values
figure(1)
sgtitle('Open Loop Singular Values')
subplot(2,1,1)
sigma(G(1,1),'b',G(2,2),'r--',p)
hold on
title('\sigma(G)')
```

```matlab
legend('\theta','\phi')

subplot(2,1,2)
sigma(GK(1,1),'b',GK(2,2),'r--',p)
title('\sigma(GK)')
legend('\theta','\phi')

% CL TFs
figure(2)
subplot(2,1,1)
sigma(So(1,1),'b',So(2,2),'r--',inv(Wp),p)
legend('\theta','\phi','\sigma(1/Wp)')
title('Disturbance Rejection (S_o)')

WpSo = Wp*So ;
subplot(2,1,2)
sigma(WpSo(1,1),'b',WpSo(2,2),'r--',p)
legend('\theta','\phi')
title('Nominal Performance (NP)')

figure(3)
sigma(To(1,1),'b',To(2,2),'--r',p)
legend('\theta','\phi')
title('Tracking (T_o)')

figure(4)
WiTi = -Wi*Ti ;
sigma(WiTi(1,1),'b',WiTi(2,2),'--r',p)
legend('\theta','\phi')
title('Robust Stability (RS)')

% CL Step Response and Disturbance Rejection
figure(5)
subplot(2,2,1)
step(To(1,1))
ylabel('\theta')

subplot(2,2,2)
step(To(2,2))
ylabel('\phi')

subplot(2,2,3)
step(So(1,1))
ylabel('d_0 \rightarrow \theta')

subplot(2,2,4)
step(So(2,2))
ylabel('d_0 \rightarrow \phi')
```

```
%% C: Robust Performance
Wi = tf([1 .2],[.5 1])*eye(3) ;

A = .005 ;
M = 2 ;
wb = 1 ;
wp = tf([1/M wb],[1 wb*A]) ;
Wp = wp*eye(3) ;

% Nominal Plant
systemnames = 'G Wp Wi' ;
inputvar = '[ydel(3); w(3); u(3)]' ;
outputvar = '[Wi; Wp; -G-w]' ;
input_to_G = '[u+ydel]' ;
input_to_Wp = '[w+G]' ;
input_to_Wi = '[u]' ;
sysoutname = 'P' ;
sysic ;
P = minreal(ss(P)) ;

n_meas = 3 ;
n_ctrl = 3 ;
w = logspace(-3,3) ;


[K,CL,gamma,info] = hinfsyn(P,n_meas,n_ctrl) ;
So = minreal(inv(eye(3)+G*K)) ;
To = minreal(G*K*inv(eye(3)+G*K)) ;
Ti = minreal(K*G*inv(eye(3)+K*G)) ;
Si = minreal(inv(eye(3)+K*G)) ;

N = [-Wi*Ti Wi*K*So; -Wp*G*Si Wp*So] ;

N_frd = frd(N,w) ;
blk = [1 1; 1 1; 1 1; 3 3] ;
[mu,info] = mussv(N_frd,blk) ;

WiTi_frd = frd(N(1:3,1:3),w) ;
blk_WiTi = [1 1; 1 1; 1 1] ;
[mu_WiTi] = mussv(WiTi_frd,blk_WiTi) ;

WpSo_frd = frd(N(4:6,4:6),w) ;
blk_WpSo = [3 3] ;
[mu_WpSo] = mussv(WpSo_frd,blk_WpSo) ;


% RS and NP
```

```
figure(7)
subplot(2,1,1)
sigma(mu_WiTi(1,1),'b',mu_WiTi(1,2),'--r',p)
title('Robust Stability (RS)')

subplot(2,1,2)
sigma(mu_WpSo(1,1),'b',mu_WpSo(1,2),'--r',p)
title('Nominal Performance (NP)')


% Plot of mu bounds
figure(8)
sigma(mu(1,1),'b',mu(1,2),'--r',p)
ylabel('\mu upper/lower bounds (abs)')
title('Robust Performance (RP) \mu Plot')
legend('upper','lower')


% CL Step Response and Disturbance Rejection
figure(9)
subplot(2,2,1)
step(To(1,1))
ylabel('\theta')

subplot(2,2,2)
step(To(2,2))
ylabel('\phi')

subplot(2,2,3)
step(So(1,1))
ylabel('d_0 \rightarrow \theta')

subplot(2,2,4)
step(So(2,2))
ylabel('d_0 \rightarrow \phi')


%% D: D-K Iteration

% Initialize
D = append(1,1,1,tf(eye(3)),tf(eye(3))) ;

% Initial K-Step
[K,CL,gamma,info] = hinfsyn(D*P*inv(D),n_meas,n_ctrl) ;

% Initial D-Step
Nf = frd(lft(P,K),w) ;
[mu_Nf,mu_Info] = mussv(Nf,blk) ;
```

```
mu_RP(1) = norm(mu_Nf(1,1),inf,1e-6) ;

figure(11)
sigmaplot(mu_Nf(1,1),p)
ylabel('\mu upper bounds (abs)')
title('Manual DK Iteration')

% D-K Iteration Loop
n = 10 ;

for i = 2:n
    % Fit resulting D-scales
    [dsysl,dsysr] = mussvunwrap(mu_Info) ;
    dsysl = dsysl/dsysl(3,3) ;
    di = fitfrd(genphase(dsysl(1,1)),3) ;
    Di = append(di,di,di,tf(eye(3)),tf(eye(3))) ;

    % K-Step
    [Ki,CL,gamma(i),info] = hinfsyn(Di*P*inv(Di),n_meas,n_ctrl) ;

    % D-Step
    Nf = frd(lft(P,Ki),w) ;
    [mu_Nf,mu_Info] = mussv(Nf,blk) ;
    mu_RP(i) = norm(mu_Nf(1,1),inf,1e-6) ;
    if i == 6
        WiTi_frd = frd(Nf(1:3,1:3),w) ;
        blk_WiTi = [1 1; 1 1; 1 1] ;
        [mu_WiTi] = mussv(WiTi_frd,blk_WiTi) ;

        WpSo_frd = frd(Nf(4:6,4:6),w) ;
        blk_WpSo = [3 3] ;
        [mu_WpSo] = mussv(WpSo_frd,blk_WpSo) ;
        Kfinal = Ki ;

        figure(18)
        subplot(2,1,1)
        sigma(mu_WiTi(1,1),'b',mu_WiTi(1,2),'--r',p)
        title('Robust Stability (RS)')

        subplot(2,1,2)
        sigma(mu_WpSo(1,1),'b',mu_WpSo(1,2),'--r',p)
        title('Nominal Performance (NP)')

        figure(12)
        sigma(mu_Nf(1,1),p)
        ylabel('\mu upper bounds (abs)')
        title('Optimal Manual DK Iteration')
    end
```

```
    % Add to Plot
    figure(11)
    hold on
    sigma(mu_Nf(1,1),p)
    ylabel('\mu upper bounds (abs)')
    title('Manual DK Iterations')

end
legend('1','2','3','4','5','6','7','8','9','10')

% Peak of mu vs. iteration
figure(13)
plot(mu_RP)
hold on
plot(gamma)
xlabel('Iteration')
ylabel('\mu and \gamma')
title('Robust Performance Level and H_{\infty} Cost')
legend('Peak of \mu','\gamma')

% CL TFs
So1 = minreal(inv(eye(3)+G*Kfinal)) ;
To1 = minreal(G*Kfinal*inv(eye(3)+G*Kfinal)) ;
GK = G*Kfinal ;


% Closed Loop Step Response and Disturbance Rejection
figure(14)
subplot(2,2,1)
step(To(1,1),To1(1,1))
ylabel('\theta')
legend('Original','DK')

subplot(2,2,2)
step(To(2,2),To1(2,2))
ylabel('\phi')
legend('Original','DK')

subplot(2,2,3)
step(So(1,1),So1(1,1))
ylabel('d_0 \rightarrow \theta')
legend('Original','DK')

subplot(2,2,4)
step(So(2,2),So1(2,2))
ylabel('d_0 \rightarrow \phi')
legend('Original','DK')
```