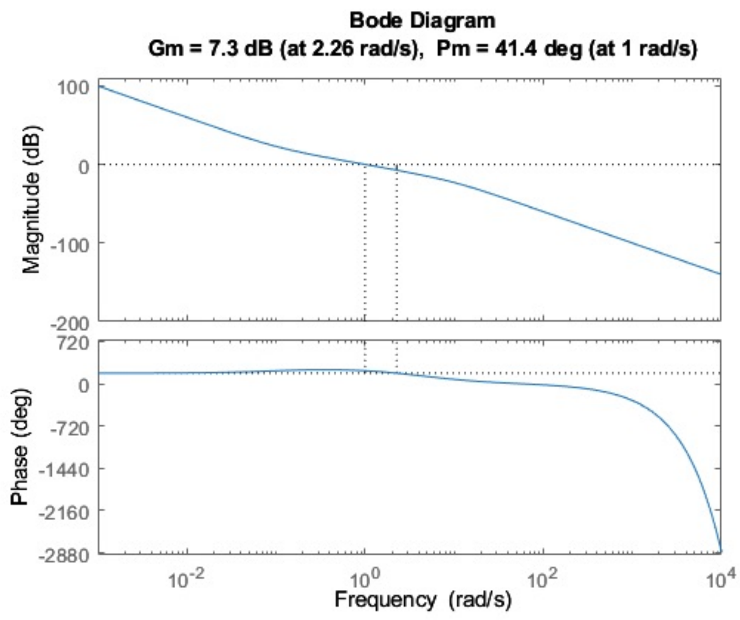
1: 20 Points

Unstable / Non-minimum Phase Plants

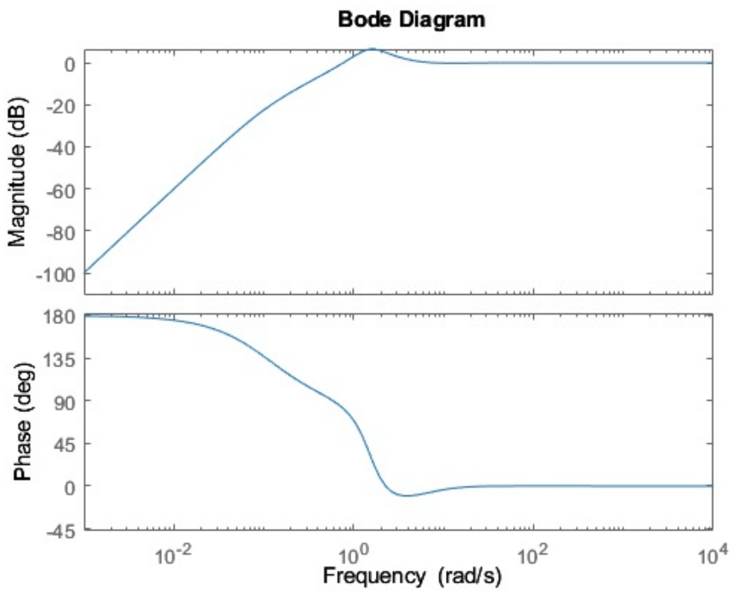
(a) Consider the non-minimum phase plant $G(s) = \frac{-s+3}{(s+2)(s+1000)}$ and the problem of inverse-based controller design. Assuming that the system has a time delay of 5 ms, and that we desire a 40 dB/dec slope starting a decade above and below the crossover frequency, design an inverse-based controller that maximizes the crossover frequency while achieving a gain margin of 6 dB and a phase margin of 40 degrees. Show the stability margin of the final achieved loop shape along with the sensitivity function.

$-2\tan^{-1}\left(\frac{w_c}{3}\right) > -90^\circ + \underset{PM}{40^\circ} \quad \rightarrow w_c < 1.4 \ \text{rad/s}$



Stability Margin

```
%Problem 1a
s = tf('s');
G = (-s+3)/(s+2)*(s+1000);
time_delay = 0.005;
GM = 200;
PM = 200;

G_delay = G * exp(-time_delay * s);
wc = 0.1;
desired_L = 0;
a = 0.1;
b = 10;
while(db(GM) > 6 && PM > 40)
    reference = desired_L;
    L_discretized = (s + a*wc)/(s^2)/(s + b*wc)*exp(-time_delay*s)*(-s+3)/(s+3);
    [magnitude, phase] = bode(L_discretized, wc);
    desired_L = (s + a*wc)/(s^2)/(s + b*wc)*(-s+3)/(s+3)/magnitude;
    [GM, PM, ~, ~] = margin(L_discretized / magnitude);
    wc = wc + 0.1;
end
L = G_delay * minreal(reference/G);
margin(L)
sensitivity = 1 / (1 + L)
bodeplot(sensitivity)
```

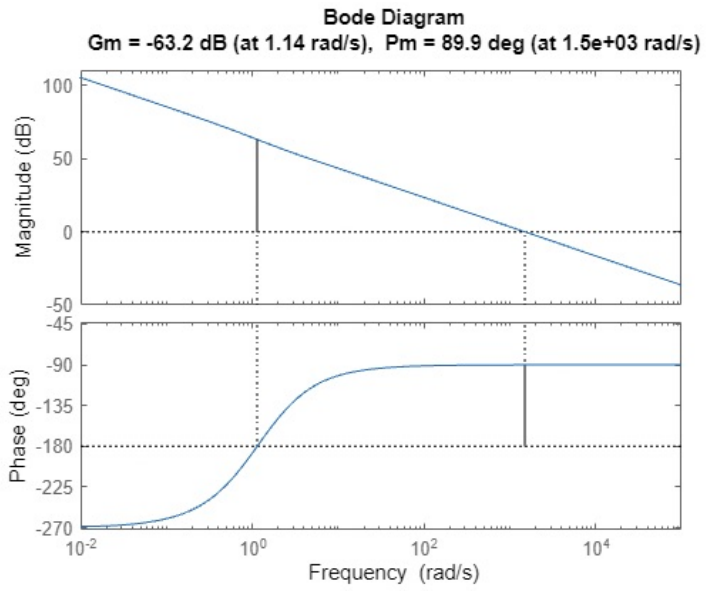


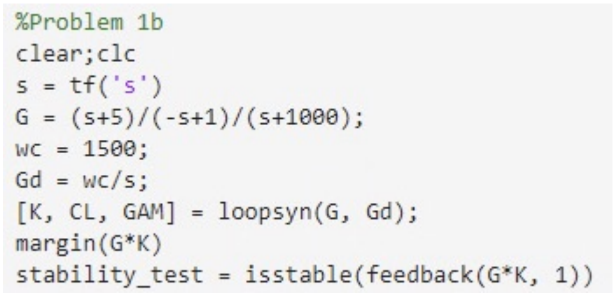
sensitivity bode plot

```
sensitivity =

  A = 
             x1         x2         x3         x4         x5         x6
   x1     -1015     -117.6     -27.37     -7.324          0          0
   x2       128          0          0          0          0          0
   x3         0         16          0          0          0          0
   x4         0          0          4          0          0          0
   x5         0          0          0    0.03125          0          0
   x6         0          0          0          0    0.03125          0
   x7         0  -0.002441    -0.1525    0.03456      7.447      23.44

  B = 
       u1
   x1   0
   x2   0
   x3   0
```

sensitivity function
state space

(b) The inverse-based design breaks down for an unstable plant. Consider the plant $G(s) = \frac{s+5}{(-s+1)(s+1000)}$ <u>without</u> a time delay. Using any method of your choice, design a feedback controller for this system that achieves an open loop crossover of at least 1000 rad/s with a GM$\geq$ 6dB and a PM $\geq$ 30 degrees. Use the margin command to demonstrate success and plot the sensitivity function.





```
%Problem 1b
clear;clc
s = tf('s')
G = (s+5)/(-s+1)/(s+1000);
wc = 1500;
Gd = wc/s;
[K, CL, GAM] = loopsyn(G, Gd);
margin(G*K)
stability_test = isstable(feedback(G*K, 1))
```

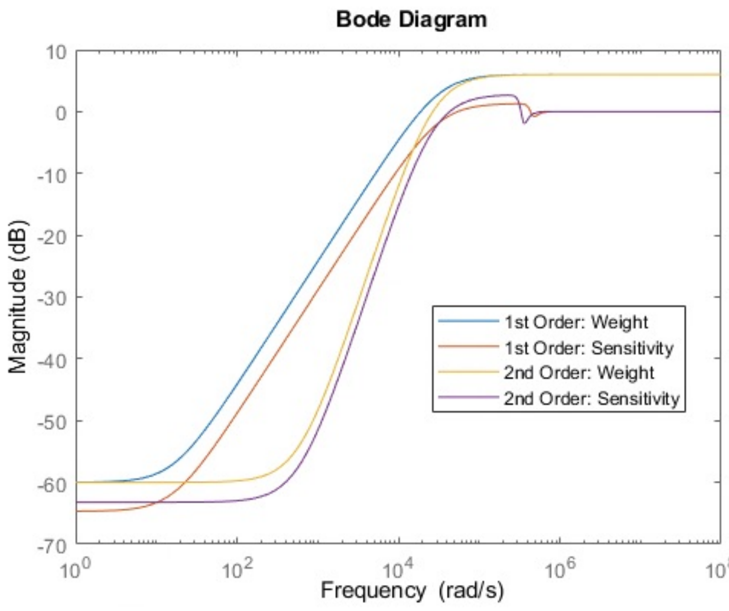
```
stability_test = logical
  1
```

↑ closed loop stability test

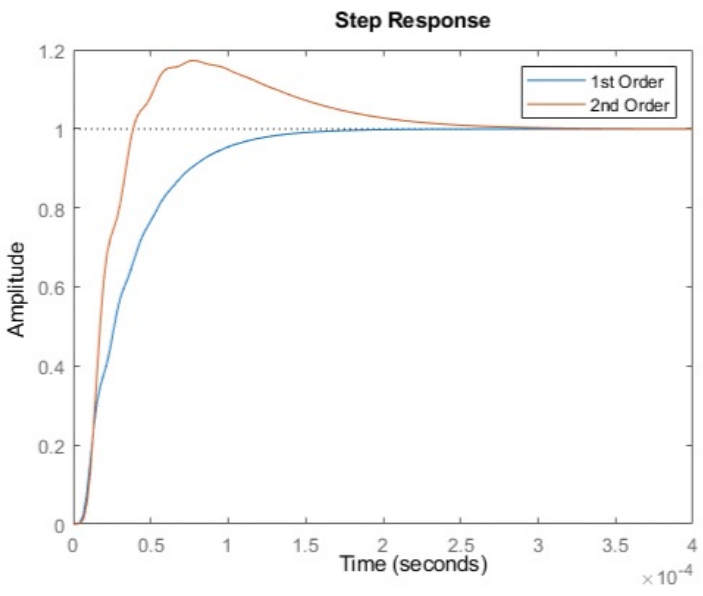
2: 30 Points

Single Stage HDD Control - Mixed Sensitivity Synthesis

(a) Using the single stage HDD model in file HDDModel.m, design a controller that achieves 60 dB low frequency attenuation, a 3 kHz sensitivity zero dB crossing, and a sensitivity peak no greater than 6 dB. Do this for both first and 2nd order sensitivity weights. Submit the following plots: 1.) sensitivity functions for both designs along with desired sensitivity functions; 2.) step response of both closed loop systems. Given the same specs, is there is a limit to the sensitivity crossing frequency for the 2nd order sensitivity weight?



weights and sensitivity



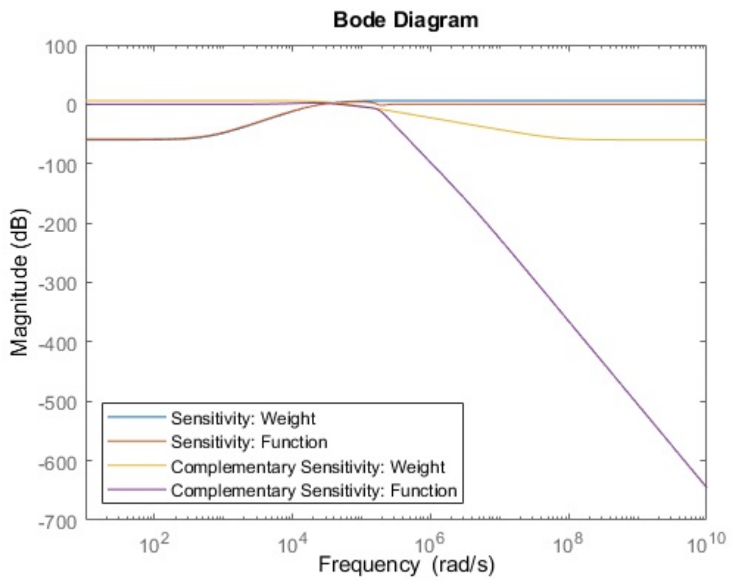
step response

```
gamma_test =
1.1476
```

yes, there is a limit as $\gamma > 1$ at high BW

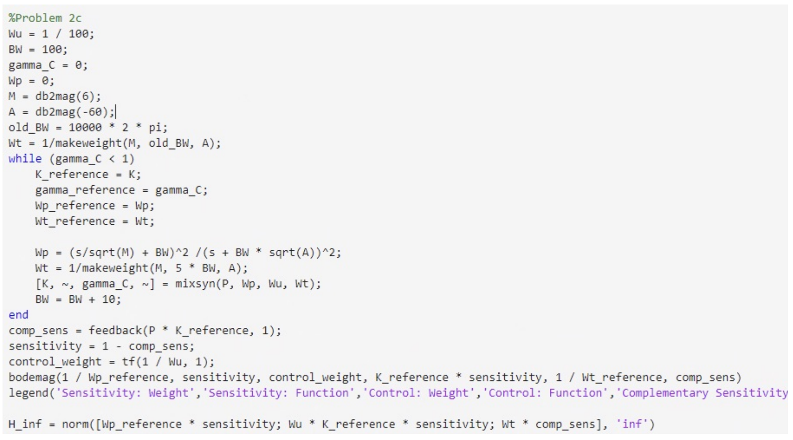
```
%Problem 2a
HDDModel;
s = tf('s')
BW = 3000 * 2 *pi;
M = db2mag(6);
A = db2mag(-60);
Wp_1 = 1/makeweight(A, BW, M);
Wp_2 = (s/sqrt(M) + BW)^2/(s + BW  *sqrt(A))^2;

[K_1, ~, gamma_1, ~] = mixsyn(P, Wp_1, [], []);
[K_2, ~, gamma_2, ~] = mixsyn(P, Wp_2, [], []);

sens_1 = 1 - feedback(P * K_1,1);
sens_2 = 1 - feedback(P * K_2,1);
bodemag(1/Wp_1, sens_1, 1/Wp_2, sens_2);
legend('1st Order: Weight','1st Order: Sensitivity','2nd Order: Weight','2nd Order: Sensitivity','Location','best')

step(1 - sens_1, 1 - sens_2);
legend('1st Order', '2nd Order')

BW_test = 50000;
Wp_test = (s/sqrt(M)+BW_test)^2/(s+BW_test*sqrt(A))^2;
[K_max, ~, gamma_test, ~] = mixsyn(P,Wp_test,[],[]);
gamma_test
```

(b) Now let's add a complementary sensitivity weight to ensure that the high frequency gain is low. For a successful design ($\gamma < 1$), our weights must satisfy $1/W_p + 1/W_T \geq 1$, or else we violate the fundamental constraint $S + T = 1$. Create a complementary sensitivity weight that ensures a 20 dB/dec rolloff at high frequencies with crossover frequency of 10000 Hz. Using the 2nd order performance weight from Part a, perform the mixed sensitivity design and plot the weights against the sensitivity and complementary sensitivity function.

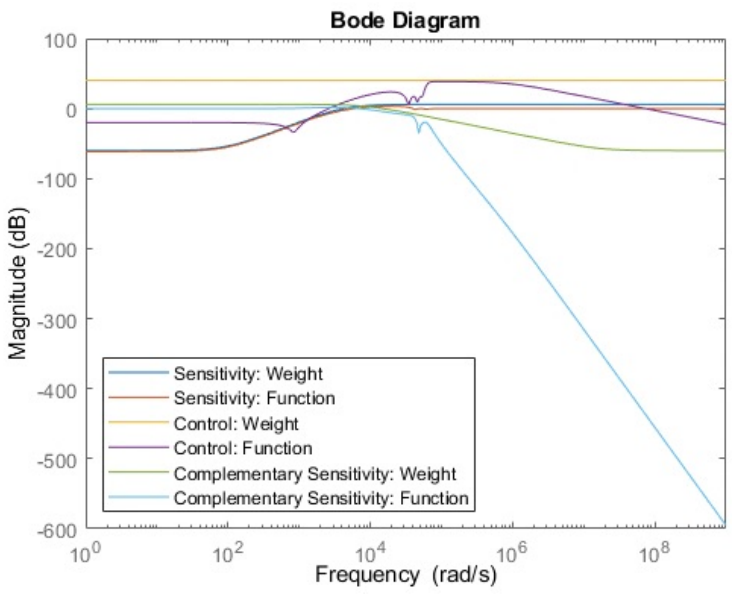


```
%Problem 2b
BW_t = 10000 * 2 * pi;
M_t = db2mag(6);
A_t = db2mag(-60);
W_t = 1/makeweight(M_t, BW_t, A_t);
[K,~,gamma,~] = mixsyn(P, Wp_2, [], W_t);

comp_sens = feedback(P*K,1);
sensitivity = 1 - comp_sens;
bodemag(1 / Wp_2, sensitivity, 1 / W_t, comp_sens)
```

(c) Now let's add a control weight in it's simplest form. Assume that the actuator's saturation limit is 100, and design constant weight $W_u = 1/100$. You will NOT achieve a value of $\gamma < 1$ for this design using the bandwidths from above. Write a script to maximize the bandwidth such that $\gamma < 1$ using the controller weight, a 2nd order performance weight as in Part a, and a 1st order complementary sensitivity weight with crossover 5x higher than the performance weight crossover. Plot the weights vs. their respective sensitivity functions, and compute the final H_∞ norm using the norm command in Matlab.



```
%Problem 2c
Wu = 1 / 100;
BW = 100;
gamma_C = 0;
Wp = 0;
M = db2mag(6);
A = db2mag(-60);
old_BW = 10000 * 2 * pi;
Wt = 1/makeweight(M, old_BW, A);
while (gamma_C < 1)
    K_reference = K;
    gamma_reference = gamma_C;
    Wp_reference = Wp;
    Wt_reference = Wt;

    Wp = (s/sqrt(M) + BW)^2 /(s + BW * sqrt(A))^2;
    Wt = 1/makeweight(M, 5 * BW, A);
    [K, ~, gamma_C, ~] = mixsyn(P, Wp, Wu, Wt);
    BW = BW + 10;
end
comp_sens = feedback(P * K_reference, 1);
sensitivity = 1 - comp_sens;
control_weight = tf(1 / Wu, 1);
bodemag(1 / Wp_reference, sensitivity, control_weight, K_reference * sensitivity, 1 / Wt_reference, comp_sens)
legend('Sensitivity: Weight','Sensitivity: Function','Control: Weight','Control: Function','Complementary Sensitivity

H_inf = norm([Wp_reference * sensitivity; Wu * K_reference * sensitivity; Wt * comp_sens], 'inf')
```



```
H_inf = 1.2956
```

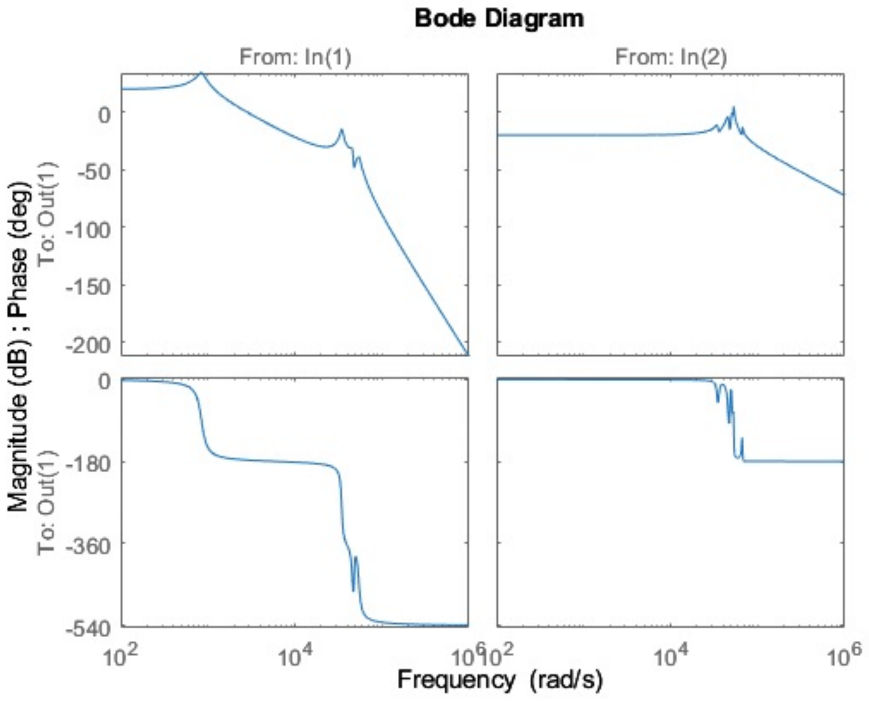
3: 20 points

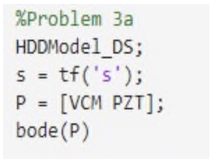
Dueal Stage HDD Control - Mixed Sensitivity Synthesis

(a) The file HDDModel_DS.m includes the plant model for a dual-stage hard disk drive. This includes the single stage model from Problem 2 along with a model of a secondary stage actuator, with the total system output given by

$$y = VCM \cdot u_{VCM} + PZT \cdot u_{PZT}.$$

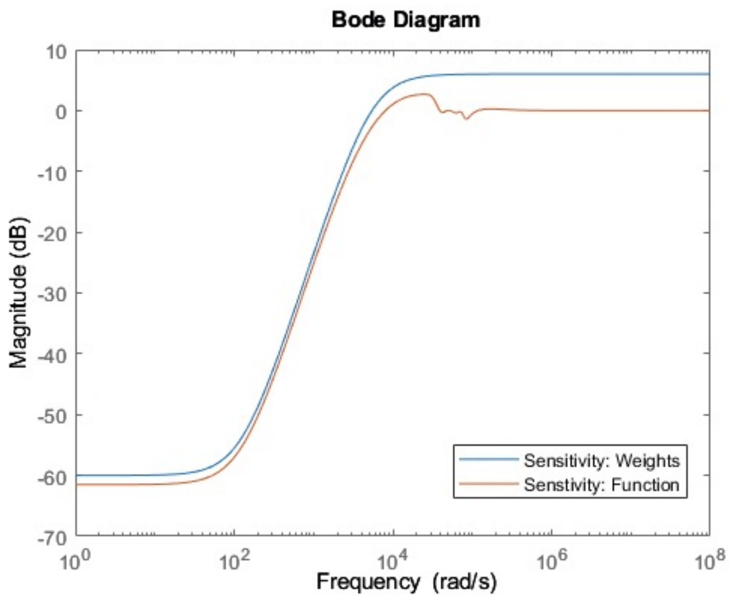
Our goal is to increase system bandwidth using the secondary actuation stage, while respecting the limits of actuator saturation. When loading the file, the two relevant models are VCM and PZT. Create a DISO model in Matlab (think about the dimensions of a 2 input, 1 output system and create a transfer function matrix) and plot the Bode plot.



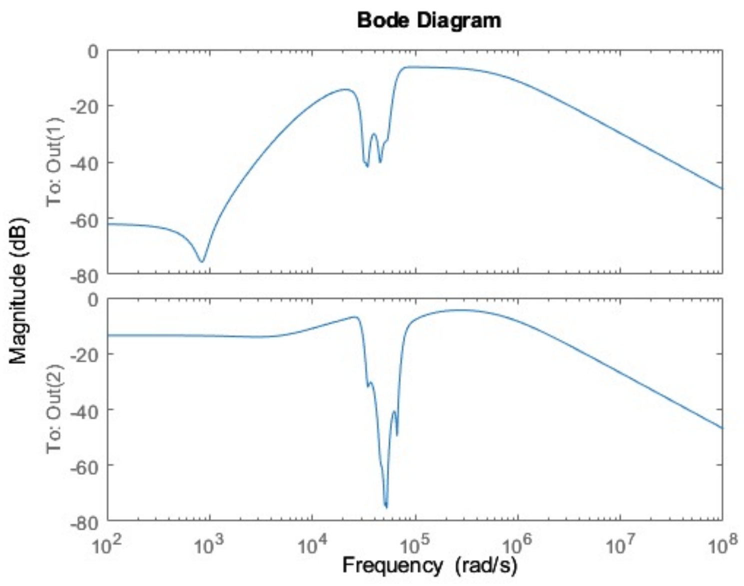


```
%Problem 3a
HDDModel_DS;
s = tf('s');
P = [VCM PZT];
bode(P)
```

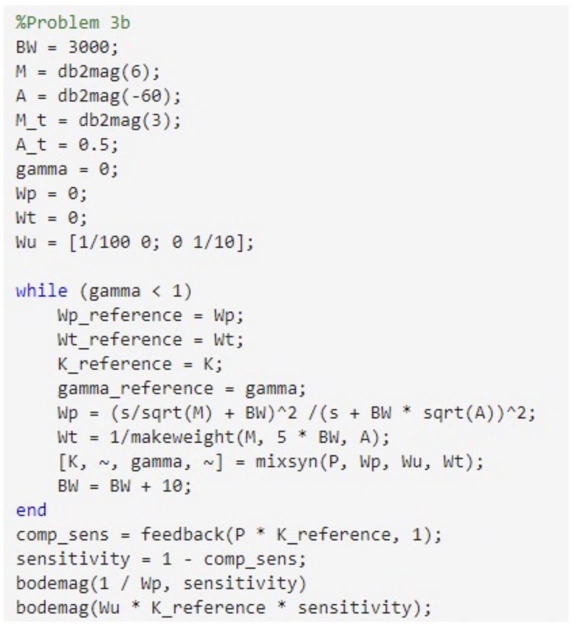
(b) The dual-stage design problem requires us to use a performance weight as in Problem 2a while designing appropriate control weights for the two actuation stages. Assign the VCM an actuation penalty assuming its maximum output is 100, and give the PZT a constant actuator penalty given that its maximum output is 10 (you will need to think about the dimensions when you put this into *mixsyn*). Perform a mixed sensitivity design using a 2nd order performance weight with the same specs as Problem 2(c) in terms of complementary sensitivity weight. As in Problem 2(c), maximize the bandwidth such that $\gamma < 1$. Plot the achieved sensitivity function along with its performance weight and plot the Bode magnitude plot of $W_U KS$ to show that the actuation limits hold.



Sensitivity function and weights



$W_U KS$ plot



```
%Problem 3b
BW = 3000;
M = db2mag(6);
A = db2mag(-60);
M_t = db2mag(3);
A_t = 0.5;
gamma = 0;
Wp = 0;
Wt = 0;
Wu = [1/100 0; 0 1/10];

while (gamma < 1)
    Wp_reference = Wp;
    Wt_reference = Wt;
    K_reference = K;
    gamma_reference = gamma;
    Wp = (s/sqrt(M) + BW)^2 /(s + BW * sqrt(A))^2;
    Wt = 1/makeweight(M, 5 * BW, A);
    [K, ~, gamma, ~] = mixsyn(P, Wp, Wu, Wt);
    BW = BW + 10;
end
comp_sens = feedback(P * K_reference, 1);
sensitivity = 1 - comp_sens;
bodemag(1 / Wp, sensitivity)
bodemag(Wu * K_reference * sensitivity);
```