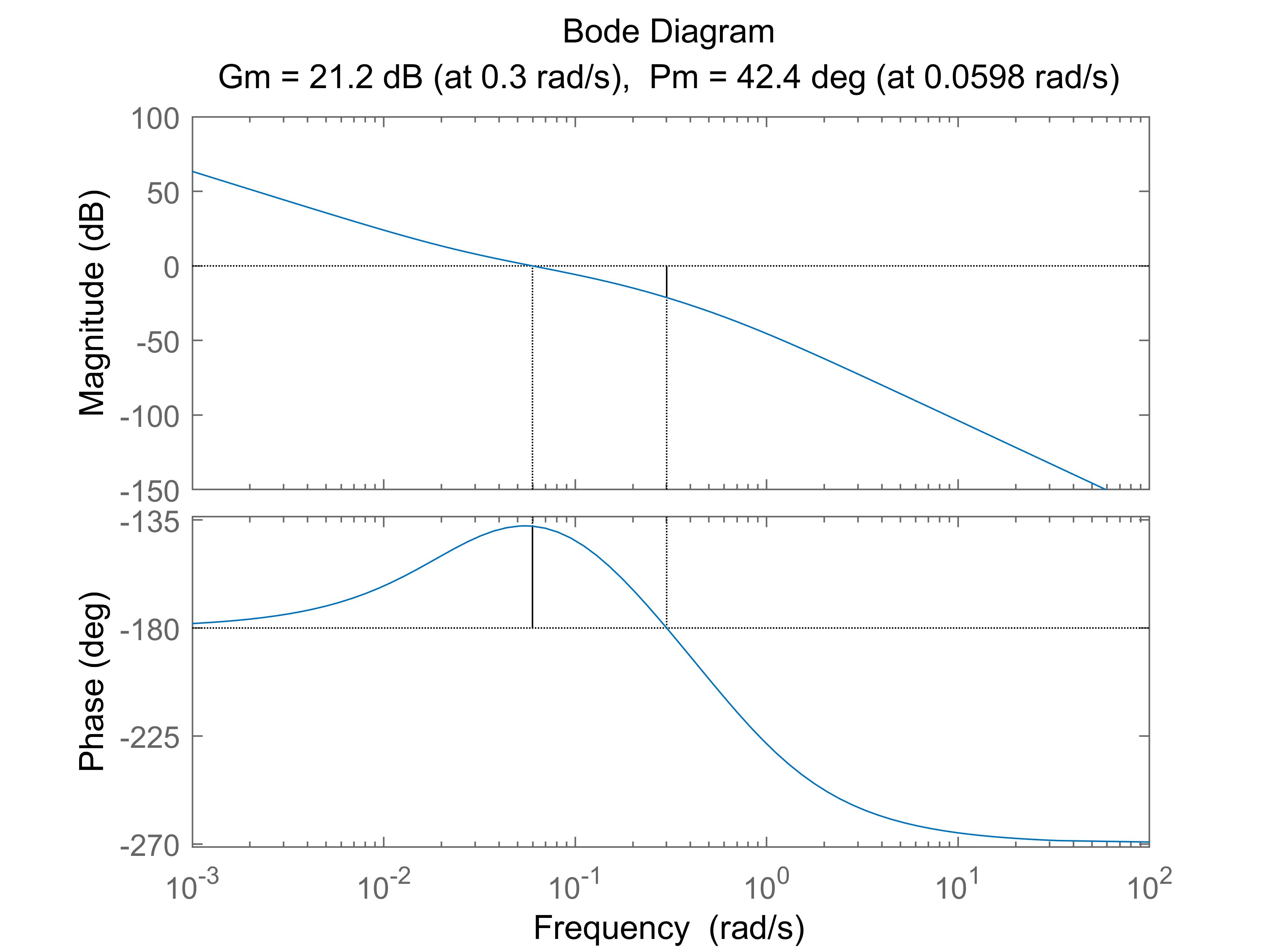
1. **Lead Controller**

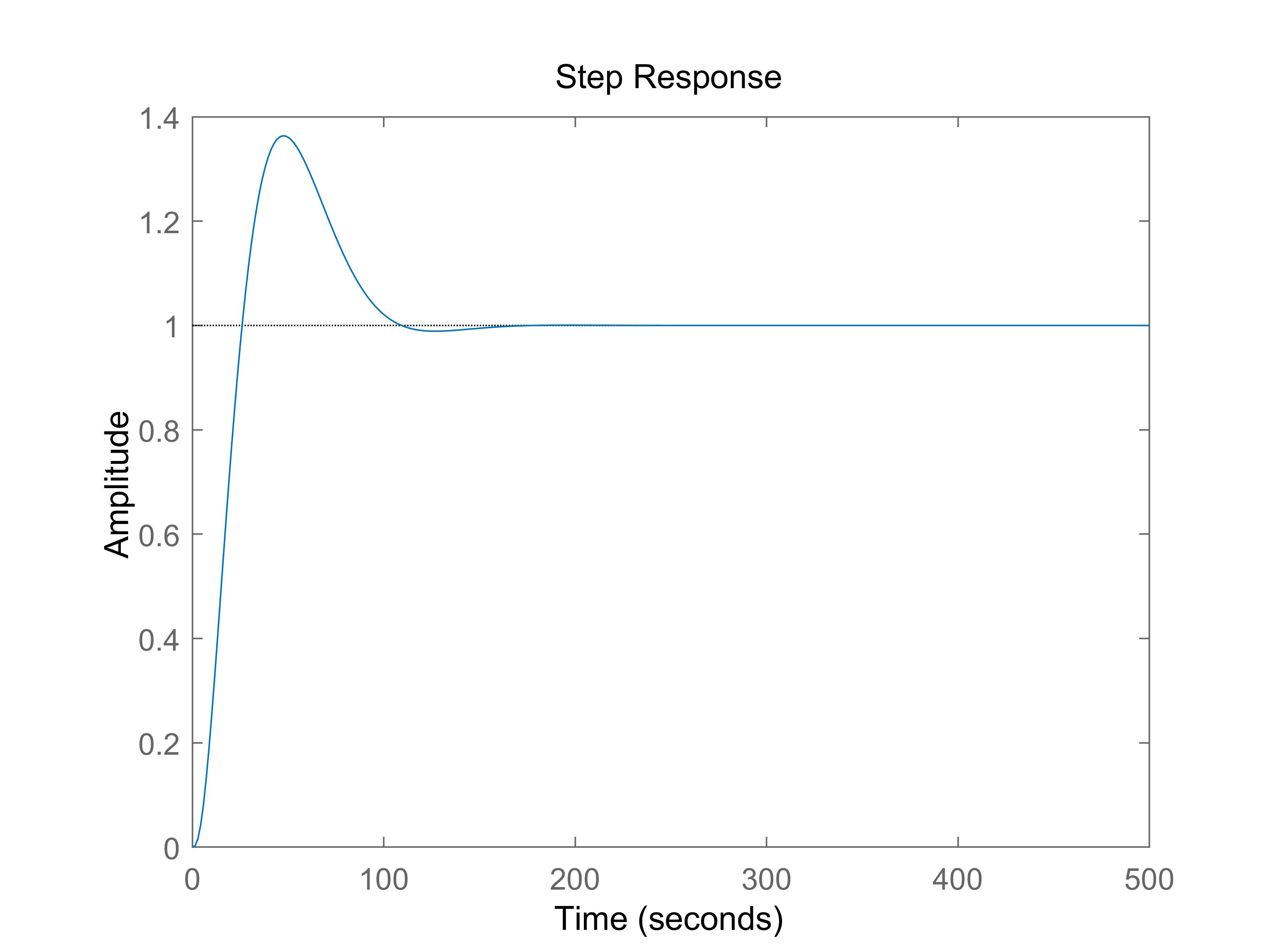
(a) The lead controller is chosen with parameters:

The resulting lead controller is:

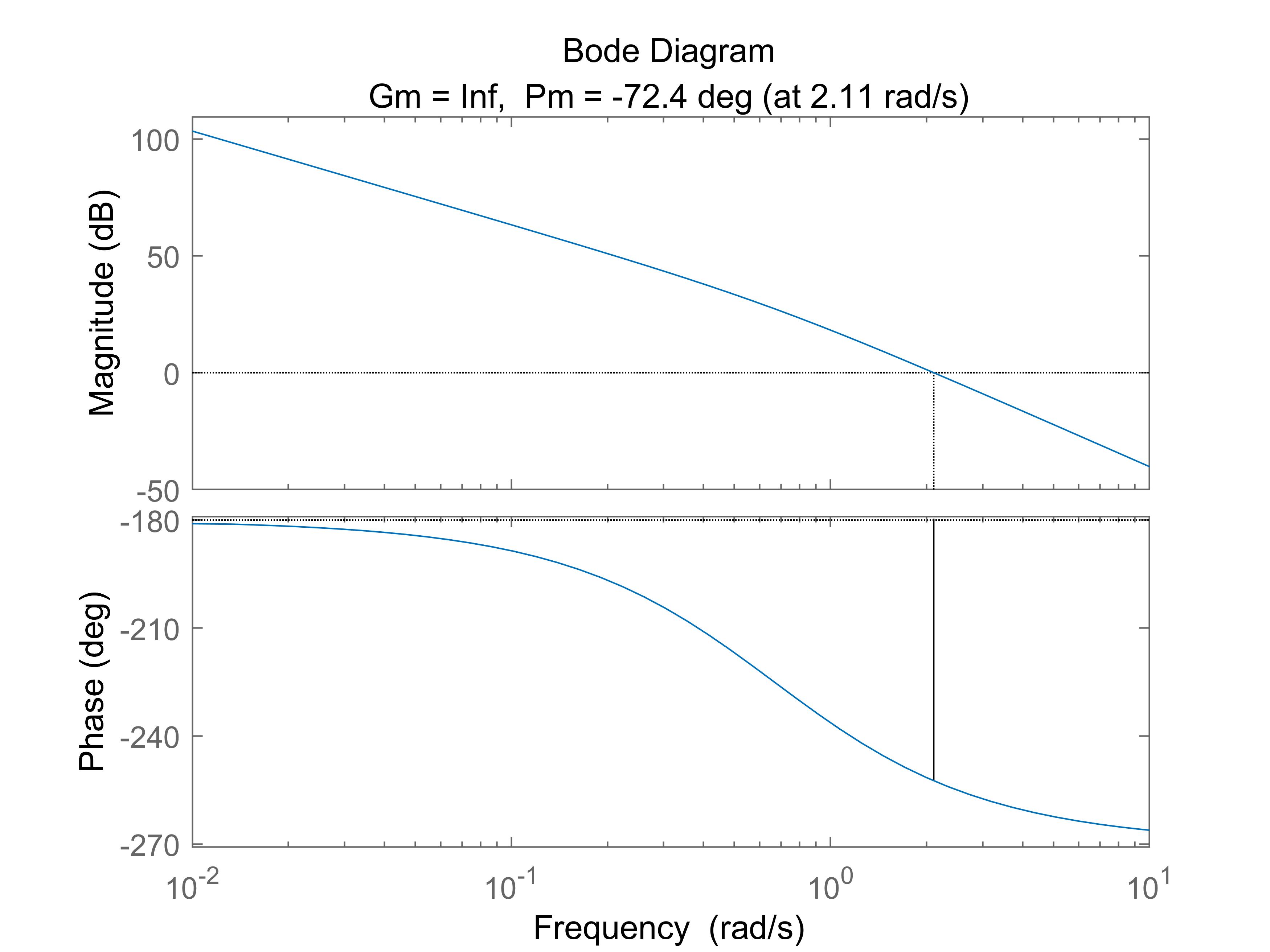
(b) The Bode plot is attached below.



(c) The closed-loop step response is attached below.



(d) It’s not possible to achieve all four specifications. As shown in the open-loop Bode plot of the uncompensated transfer function (attached below), at frequency , the phase is less than . To achieve a phase margin of at least while having the gain crossover frequency of at least , a phase lead of more than has to be introduced at the frequency of at least , which is impossible for a single lead compensator. A single lead compensator can provide a phase lead of only asymptotically. So, requirement III and requirement IV are contradicting and cannot be achieved simultaneously. The lead compensator designed above achieves requirements I, II, and IV.

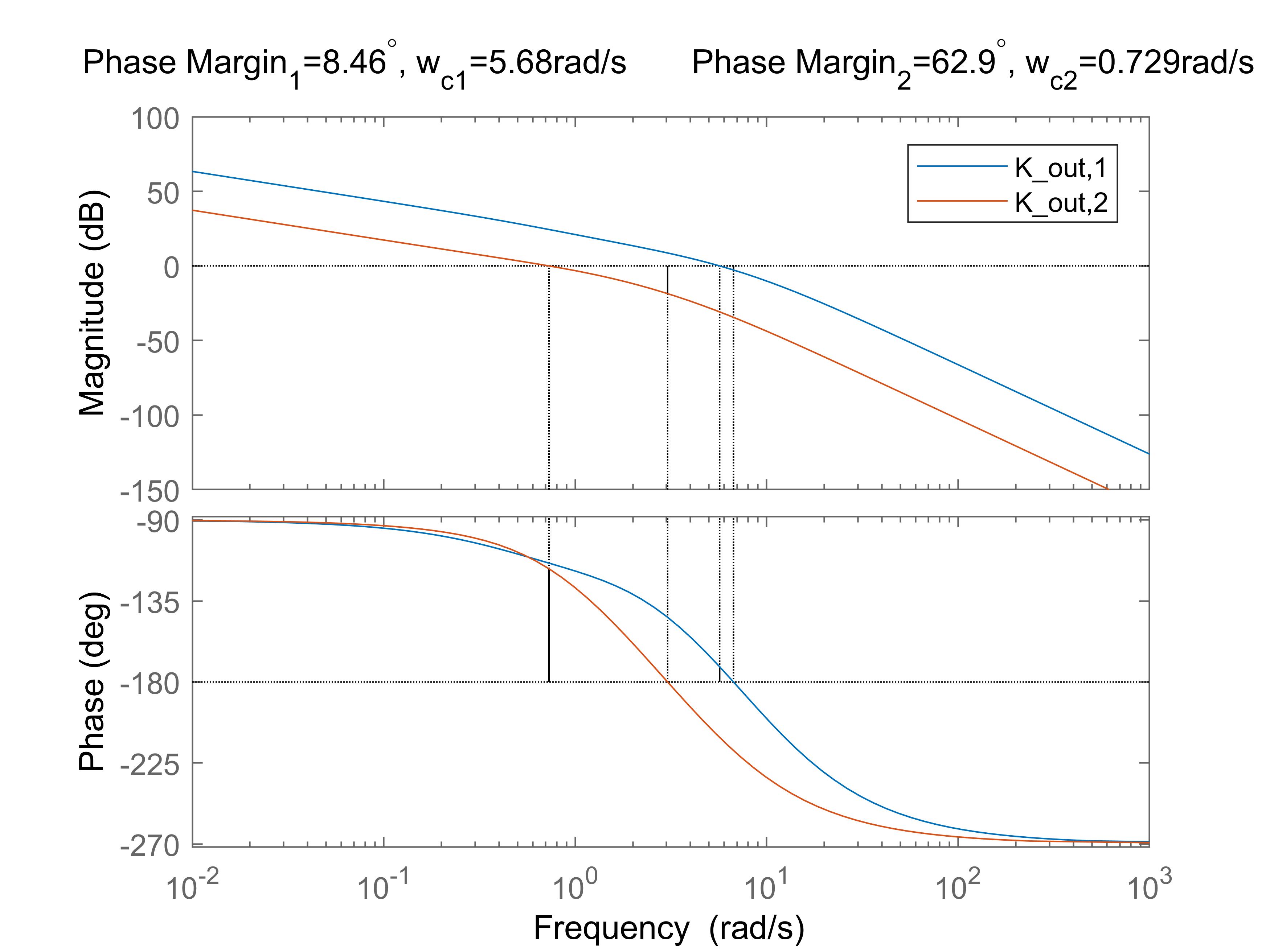


1. **Cascade Control**

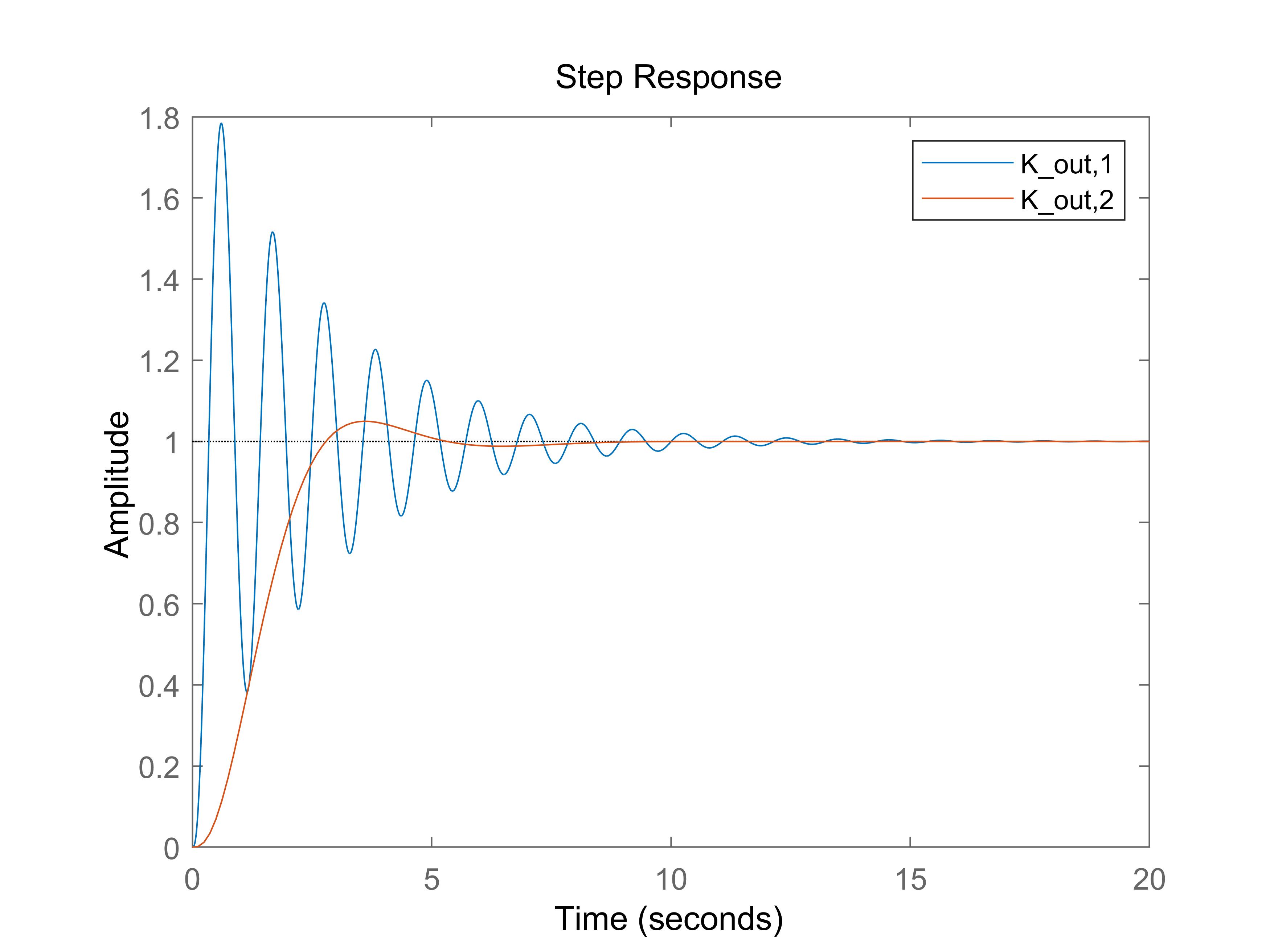
(a)

(b)

(c) The overlaid Bode plots are attached below.



(d) The overlaid closed-loop step responses are attached below.



(e) Using stepinfo(), the table is filled below.

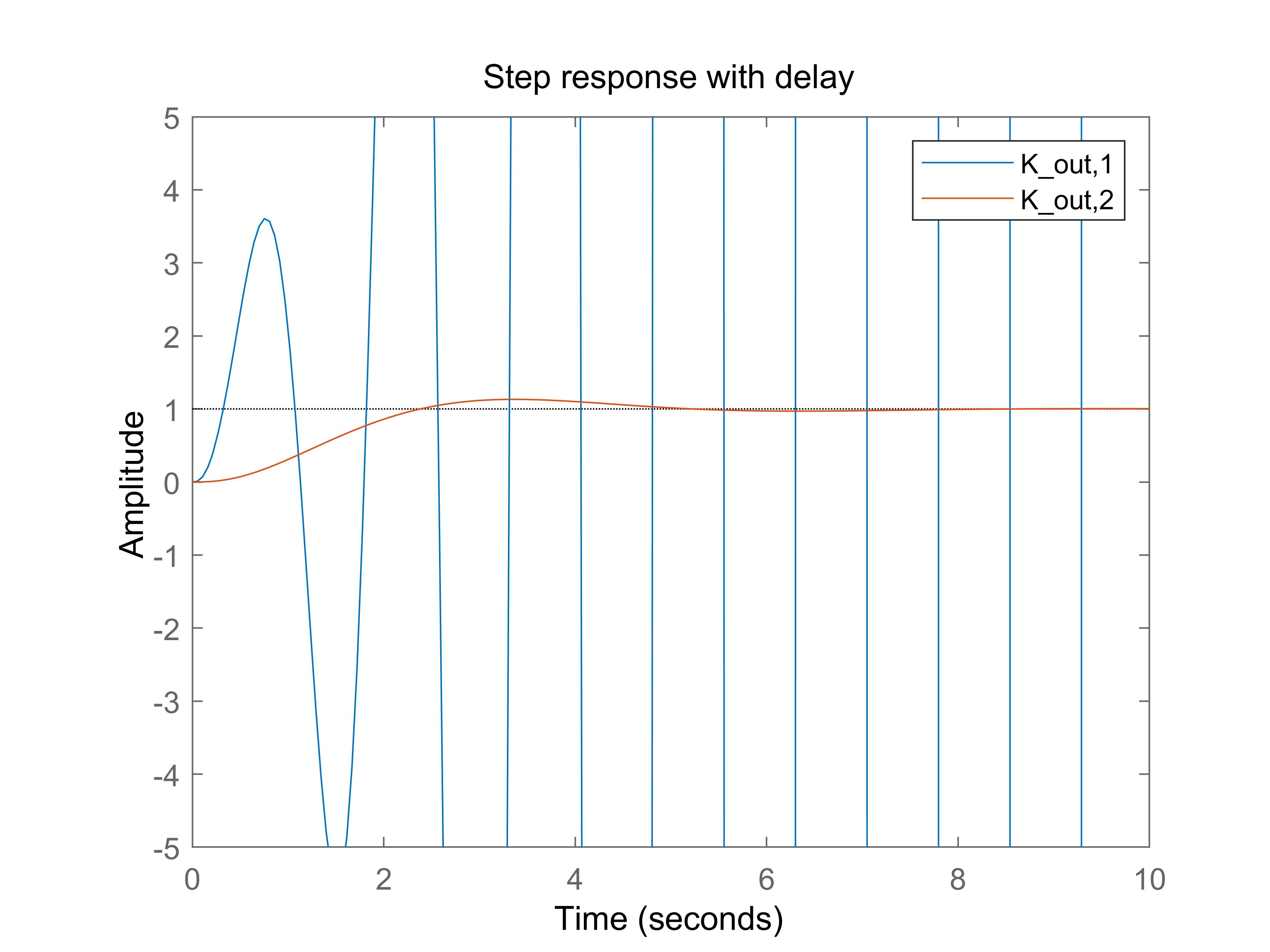
|  |  |  |  |
| --- | --- | --- | --- |
| **Controller** |  |  | No Cascade Control |
| **Rise Time(s)** |  |  |  |
| **% Overshoot** |  |  |  |

1. **Model Mismatch**

(a)

|  |  |  |
| --- | --- | --- |
| **Controller** |  |  |
| **Maximum delay (ms)** |  |  |

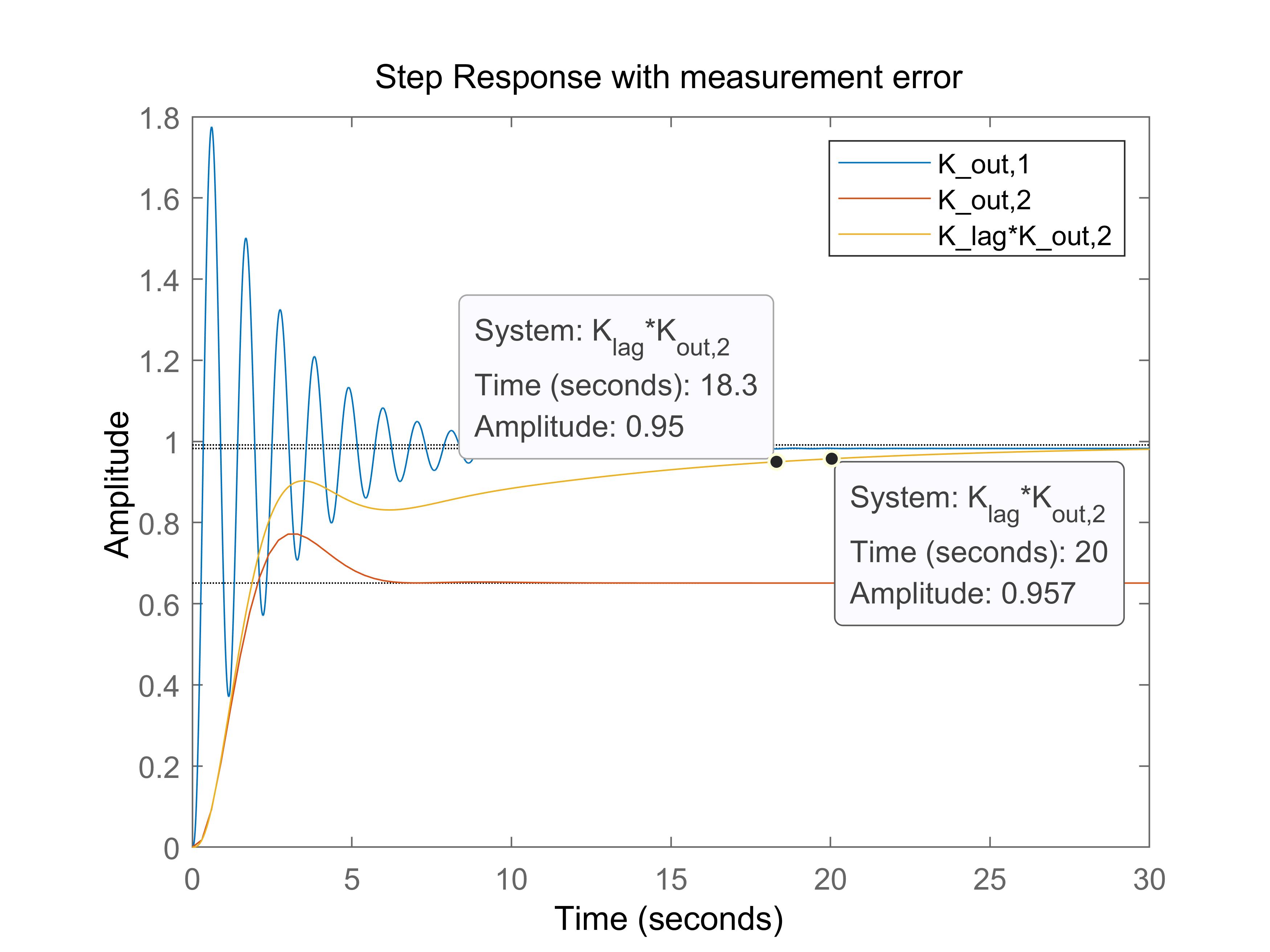
(b) The step responses with delay of is attached below. Since , which exceeds the maximum delay allowed for controller 1, the closed-loop system becomes unstable and the step response diverges. While for the controller 2, it’s within the range of allowed delay. The closed-loop system is still stable and the step response converges. The results of step responses agree with the maximum delay analysis.



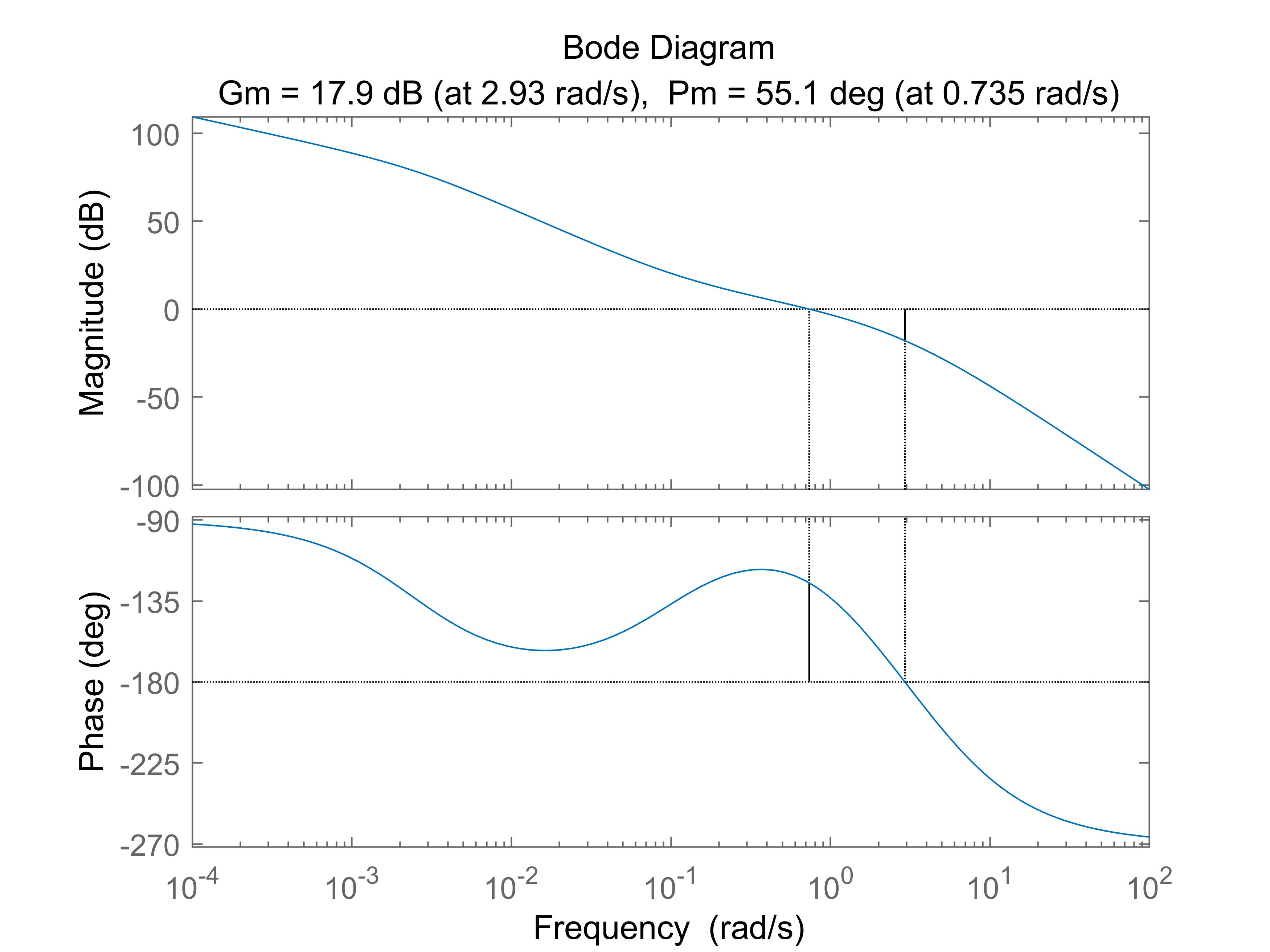
(c)

(d)

(e) The overlaid step responses with measurement error for both uncompensated and compensated plants are attached below. As shown in the plot, the compensated controller 2 rises to within 5cm error at 18.3s. And its steady state error is much smaller than using the controller 2, and the steady state error is even smaller than using the uncompensated controller 1 with much less oscillation.



(f) The Bode plot of is attached below. As shown in the plot, the phase margin is , which is above .



Full MATLAB code:

|  |
| --- |
| %% Lead Compensator  G\_CF = tf(9.81, [1 2/3 0 0]);  K = tf([40/10000 1/10000], [40\*0.15 1]);  margin(G\_CF)  margin(K\*G\_CF)  step(feedback(K\*G\_CF,1),500)  %% Cascade Control  g\_theta = tf(1,[1 0]);  tf\_theta = feedback(5\*g\_theta,1);  G\_no\_theta = tf(9.81, [1 2/3 0]);  G\_out = tf\_theta \* G\_no\_theta;  % Crossover @>=4rad/s  K1 = tf([1 1],[1\*(0.1) 1]);  % bode(K1\*G\_out);  % step(feedback(K1\*G\_out,1));  % stepinfo(feedback(K1\*G\_out,1));  % Phase margin >= 60 degree  K2 = tf([sqrt(3)/20 1/20],[1/3\*sqrt(3) 1]);  % bode(K2\*G\_out);  % step(feedback(K2\*G\_out,1));  % stepinfo(feedback(K1\*G\_out,1));  margin(K1\*G\_out)  hold on  margin(K2\*G\_out)  hold off  title("Phase Margin\_1=8.46^{\circ}, w\_{c1}=5.68rad/s Phase Margin\_2=62.9^{\circ}, w\_{c2}=0.729rad/s")  step(feedback(K1\*G\_out,1),20);  hold on  step(feedback(K2\*G\_out,1),20);  legend('K\_{out,1}','K\_{out,2}')  stepinfo(feedback(K\*G\_CF,1))  stepinfo(feedback(K1\*G\_out,1))  stepinfo(feedback(K2\*G\_out,1))  %% Model Mismatch  % Max delay  [Gm1,Pm1,Wcg1,Wcp1] = margin(K1\*G\_out);  max\_time\_delay1 = deg2rad(Pm1)/Wcp1;  [Gm2,Pm2,Wcg2,Wcp2] = margin(K2\*G\_out);  max\_time\_delay2 = deg2rad(Pm2)/Wcp2;  % Step response with delay  tf\_delay = tf(1,1,'InputDelay',0.2);  step(feedback((K1\*G\_out),tf\_delay),10)  ylim([-5,5])  hold on  step(feedback((K2\*G\_out),tf\_delay),10)  ylim([-5,5])  hold off  title("Step response with delay")  legend('K\_{out,1}','K\_{out,2}')  %S error  syms x  S1\_num = sym2poly(49.05\*((7-4\*sqrt(3))\*x+1));  S1\_den = sym2poly(x\*((7-4\*sqrt(3))\*x+1)\*(x+2/3)\*(x+5) +49.05\*(x+1));  S1 = tf(S1\_num, S1\_den);  S2\_num = sym2poly(49.05\*(sqrt(3)/3\*x+1));  S2\_den = sym2poly(x\*(sqrt(3)/3\*x+1)\*(x+2/3)\*(x+5)+(49.05/20)\*(sqrt(3)\*x+1));  S2 = tf(S2\_num, S2\_den);  %Step response with error  T1 = feedback(K1\*G\_out,1);  tf\_error\_1 = T1 - pi/180\*S1;  T2 = feedback(K2\*G\_out,1);  tf\_error\_2 = T2 - pi/180\*S2;  % Lag Compensator  K\_lag = tf([400 40],[400 1]);  S\_lag\_num = sym2poly(49.05\*(sqrt(3)/3\*x+1)\*(400\*x+1));  S\_lag\_den = sym2poly((sqrt(3)/3\*x+1)\*(400\*x+1)\*x\*(x+2/3)\*(x+5)+(49.05/20)\*(40)\*(sqrt(3)\*x+1)\*(10\*x+1));  S\_lag = tf(S\_lag\_num, S\_lag\_den);  T\_lag = feedback(K\_lag\*K2\*G\_out,1);  tf\_error\_lag = T\_lag - pi/180\*S\_lag;  step(tf\_error\_1,30)  hold on  step(tf\_error\_2,30)  hold on  step(tf\_error\_lag,30)  title("Step Response with measurement error")  legend('K\_{out,1}','K\_{out,2}','K\_{lag}\*K\_{out,2}')  hold off  margin(K\_lag\*K2\*G\_out) |