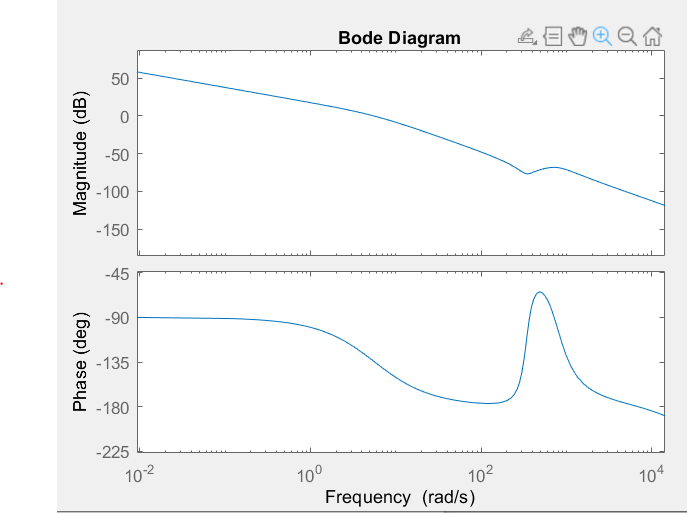
Ratthamnoon Prakitpong

#63205165

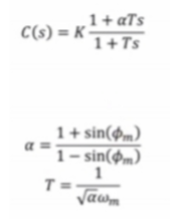
MECH421 Lab 3

1. I obtain this by exporting linsys1 as mat file, then extracting A, B, C, D to plot.



2.

We can use lead lag compensator to satisfy phase margin and cross over frequency requirements. I picked 75 deg as initial pick for phase margin, since it’s a decent bit above 60 deg. I picked 10000 rad/s as cross over frequency as it’s reasonably high. We can use these two numbers to solve for alpha and T.

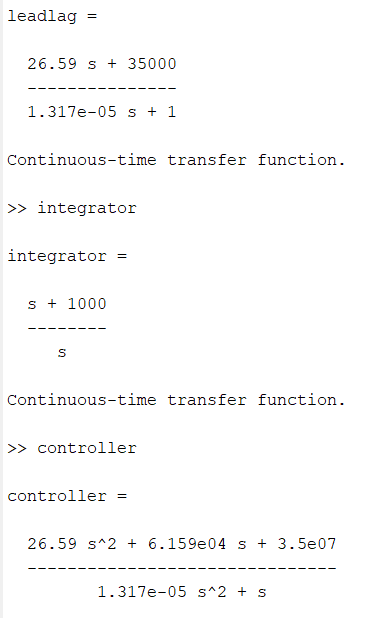


K is the offset that will shift gain plot such that cross over frequency will become approximately 10000 rad/s. I picked K = 35000.

Cascading integrator will make steady state error equals zero.



The resulting controller is:

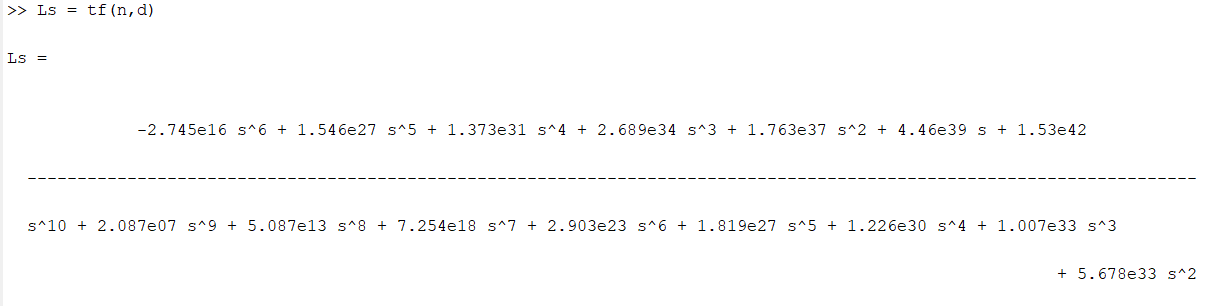


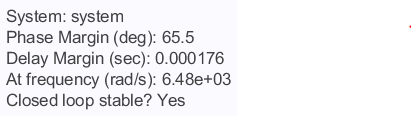
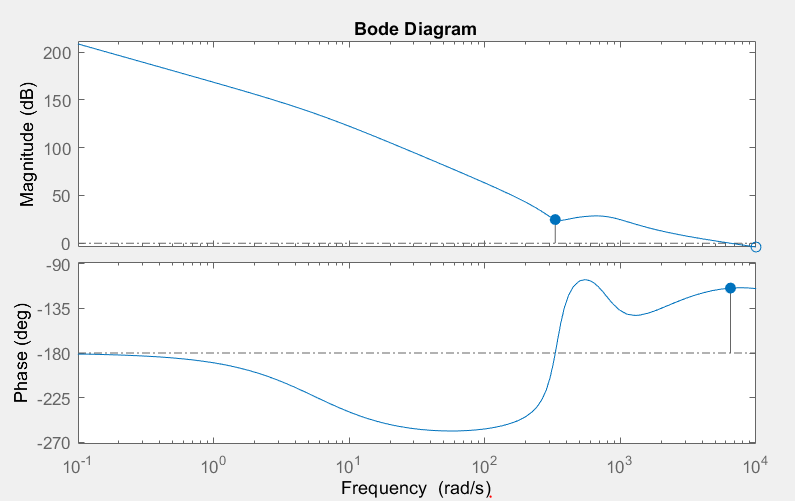
3.

L(s) = C(s)P(s)

Since P(s) is a state space model, L(s) is too. We can convert L(s) to transfer function like shown below.



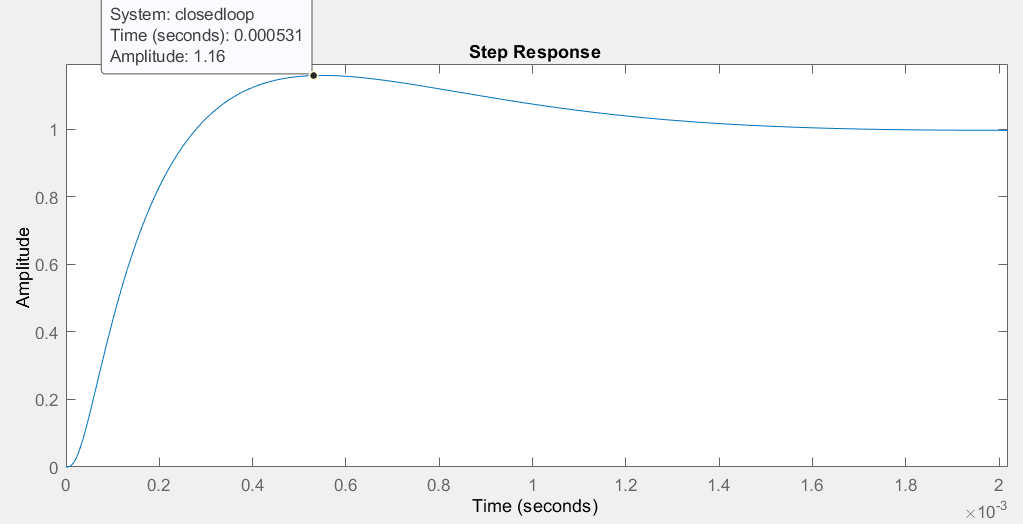




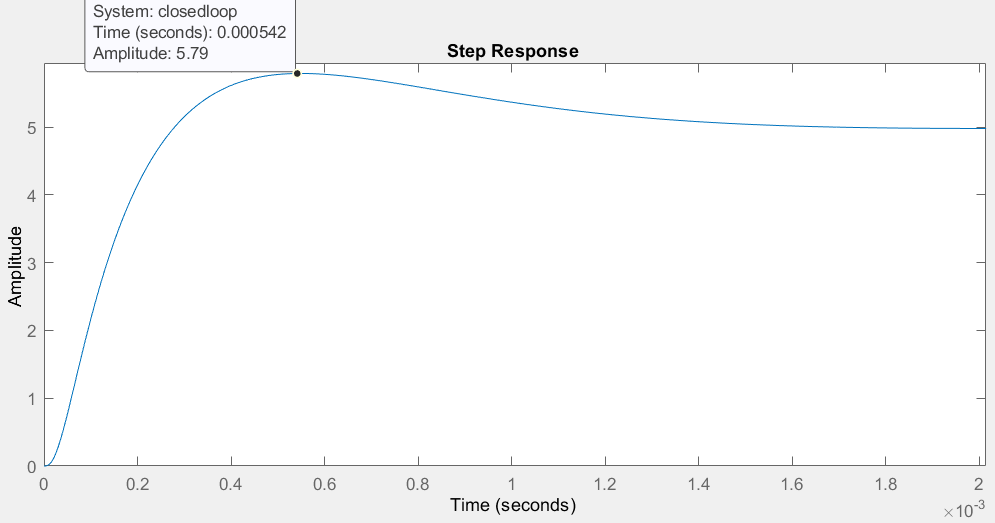
We can see that phase margin of 65.5 deg is more than 60 deg requirement. Cross over frequency is 6480 rad/s which is reasonably high. Therefore, requirements are achieved.

4.

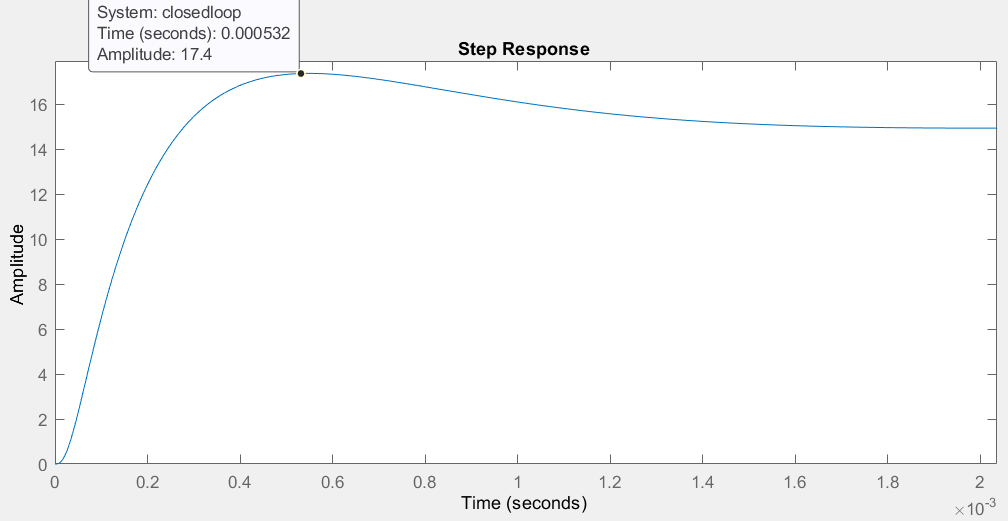
Step = 1:



Step = 5:



Step = 15:



For all:

Rise time: 1.9176e-04 s

Overshoot: 14.6209%

Amplitude is linearly proportional to input size, with no other changes observed.

Appendix: MATLAB code

% q1

load('q1.mat'); % exported from model linearizer

a = LinearAnalysisToolProject.Plots.Variables.Value.A;

b = LinearAnalysisToolProject.Plots.Variables.Value.B;

c = LinearAnalysisToolProject.Plots.Variables.Value.C;

d = LinearAnalysisToolProject.Plots.Variables.Value.D;

system = ss(a,b,c,d);

plot = bodeplot(system);

plot.showCharacteristic('AllStabilityMargins');

p = getoptions(plot);

p.PhaseMatching = 'on';

p.PhaseMatchingFreq = .1;

p.PhaseMatchingValue = -90;

setoptions(plot,p);

% q2

w = 10000;

phi = 75; % deg

phi = phi \* pi/180;

a = (1+sin(phi))/(1-sin(phi));

t = 1/(sqrt(a)\*w);

K = 35000; % trial and error

Ki = w/10;

integrator = tf([1 Ki], [1 0]); % integrator

leadlag = tf([K\*a\*t K], [t 1]); % lead lag compensator

controller = leadlag\*integrator;

system = controller\*system;

% q3

plot = bodeplot(system);

plot.showCharacteristic('AllStabilityMargins');

p = getoptions(plot);

p.PhaseMatching = 'on';

p.PhaseMatchingFreq = .1;

p.PhaseMatchingValue = -90;

setoptions(plot,p);

setoptions(plot,'Xlim',[0.1,10000]);

% q4

closedloop = system/(1+system); % black's formula

step(closedloop);

opt = stepDataOptions('StepAmplitude',5);

step(closedloop, opt);

opt = stepDataOptions('StepAmplitude',15);

step(closedloop, opt);

info = stepinfo(closedloop);