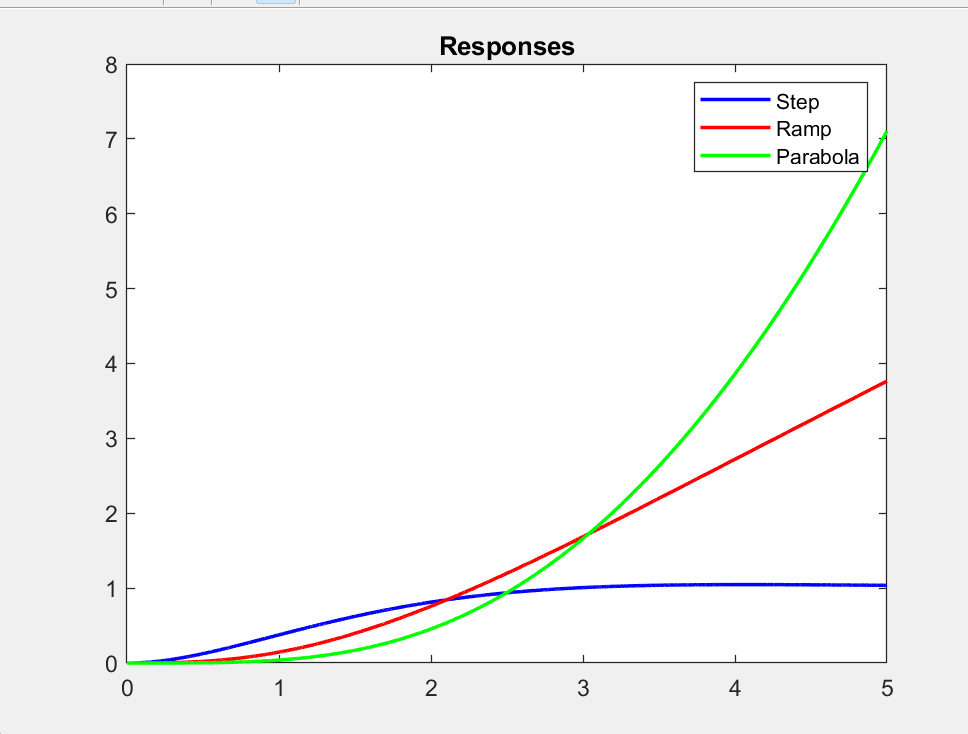
Question 18:

Graphs of inputting different inputs (Step, Ramp, and Parabolic function)  


A diagram of a flowchart

Description automatically generated

A graph with lines and dots

Description automatically generated

18) a) by checking The system’s stability using the function Pole we get the following A white background with black numbers and symbols

Description automatically generated   
which is insufficient to see the stability but if we resort to an online calculator we can see the following  
A screenshot of a computer

Description automatically generated

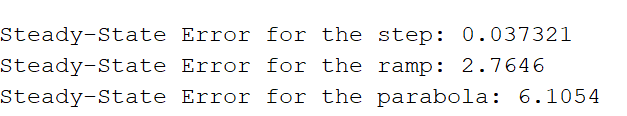
Which means we have poles on the RHP which were too small for Matlab’s rounding errors but it indicates that the system is stable (although it is a very small value it’s trivial to show how it goes to infinity).  
After adding the lead lag controller we get the following poles

A number with numbers and symbols

Description automatically generated with medium confidence

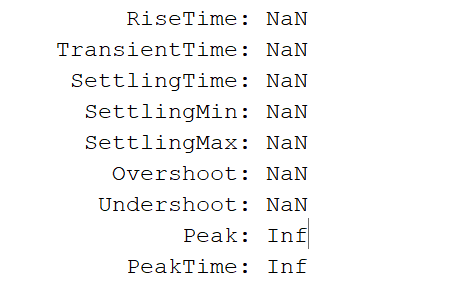
Which is a clear indication that our poles were shifted to the left and that our system is now stable

b) It’s clear to see how the error grows very quickly for any system bigger than the ramp’s as shown in the following



Question 19:

Before the lead compensator we can see that since the system is unstable it’ll never have any of these values as it’s inherently not going to rest.

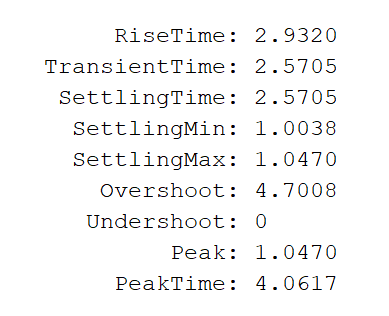


After adding a lead compensator with arbitrary values we get the following

A screenshot of a computer

Description automatically generated

After tuning the zero/pole/gain values we can reach the following values



Question 20:

Kc = 6.7

P1 = 1.8

Z1 = .015

Question 21:

Since the unstable poles were shifted to the left the system becomes stable and achieves the requirements relatively decently

Question 22:

A graph of a graph

Description automatically generated

As mentioned before the system becomes stable as demonstrated by the absence of poles in the RHP

Question 23:

A screenshot of a graph

Description automatically generated

As shown by the GM we can see that we can use a gain up until 40.7 and the gain we used was well below that margain (6.7), in addition we can see that closed loop system is stable which means we have achieved our requirements.

Code for questions 18-23  
  
close all

% Define Laplace variable 's'

s = tf('s'); % Create a transfer function variable

% Lead Compensator Design

Kc = 6.7; % Compensator gain (to reduce overshoot decrease this but it'll increase the error)

z1 = .015; % Lead zero // rise and settling time

p1 = 1.8; % Lead pole (to reduce overshoot increase this)

omega = 88; % wn

Gc = Kc \* ((s + z1) / (s + p1)); % Lead-Lag Compensator

G = tf(0.21 \* omega^2, [1 omega omega^2 0 0]); % System transfer function

sys\_cl = feedback(G \* Gc, 1); % Closed-loop system with compensator

% Step, Ramp, and Parabolic Responses

time = 0:0.001:5; % Finer time resolution for accuracy

stepResponse = step(sys\_cl, time);

rampResponse = lsim(sys\_cl, time, time); % Ramp input

parabolicResponse = lsim(sys\_cl, 0.5 \* time.^2, time); % Parabolic input

% Plot Responses

figure;

plot(time, stepResponse, 'b', 'LineWidth', 1.5);

title('Step Response'); hold on;

plot(time, rampResponse, 'r', 'LineWidth', 1.5);

title('Ramp Response'); hold on;

plot(time, parabolicResponse, 'g', 'LineWidth', 1.5);

title('Responses');legend('Parabola'); legend('Step', 'Ramp', 'Parabola');

%%

% checking the stability of the system

pole(G)

pole(sys\_cl)

%%

% info such as settling time, overshoot, etc..

info = stepinfo(sys\_cl, 'RiseTimeLimits', [0 1], 'SettlingTimeThreshold', 0.05);

disp(info);

% Check Steady-State Error for Step Input

steady\_state\_error = abs(1 - stepResponse(end)); % Steady-state error

disp(['Steady-State Error for the step: ', num2str(steady\_state\_error)]);

steady\_state\_error\_ramp = abs(1 - rampResponse(end)); % Steady-state error

disp(['Steady-State Error for the ramp: ', num2str(steady\_state\_error\_ramp)]);

steady\_state\_error\_parabola = abs(1 - parabolicResponse(end)); % Steady-state error

disp(['Steady-State Error for the parabola: ', num2str(steady\_state\_error\_parabola)]);

%%

figure;

rlocus(sys\_cl);

title('Root Locus with Lead-Lag Compensator');

%%

figure;

margin(sys\_cl);

title('Bode Plot with Lead-Lag Compensator');