EAST WEST UNIVERSITY

PROJECT

Course Title: Introduction to Control Systems

Course Code: ETE420

Submitted By:

Parveen Akther Diti

2016-2-55-024

Submitted to:

S. M. Raiyan Chowdhury

Lecturer, Dept of ECE

Date: 07.01.2021

Project Name: Antenna Azimuth Position Control System.

Objective: A position control system converts a position input command to a position output response. Position control systems find widespread applications in antennas, robot arms, and computer disk drives.

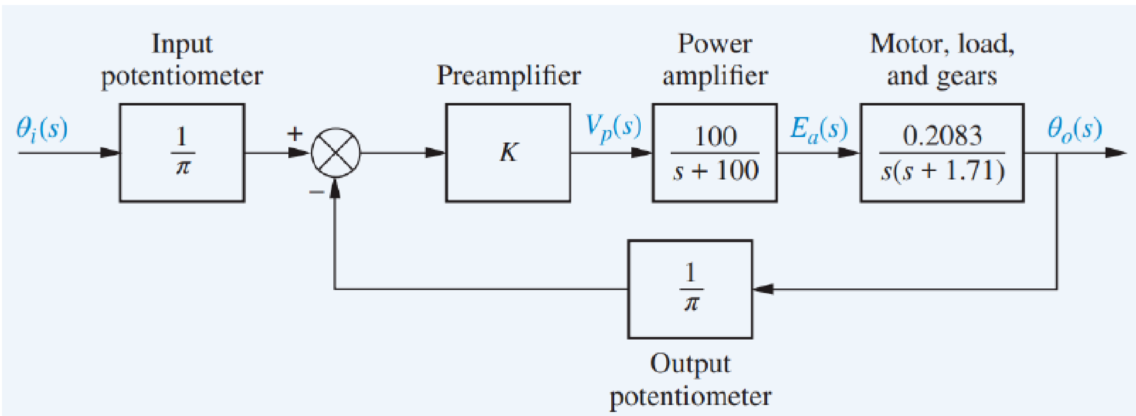
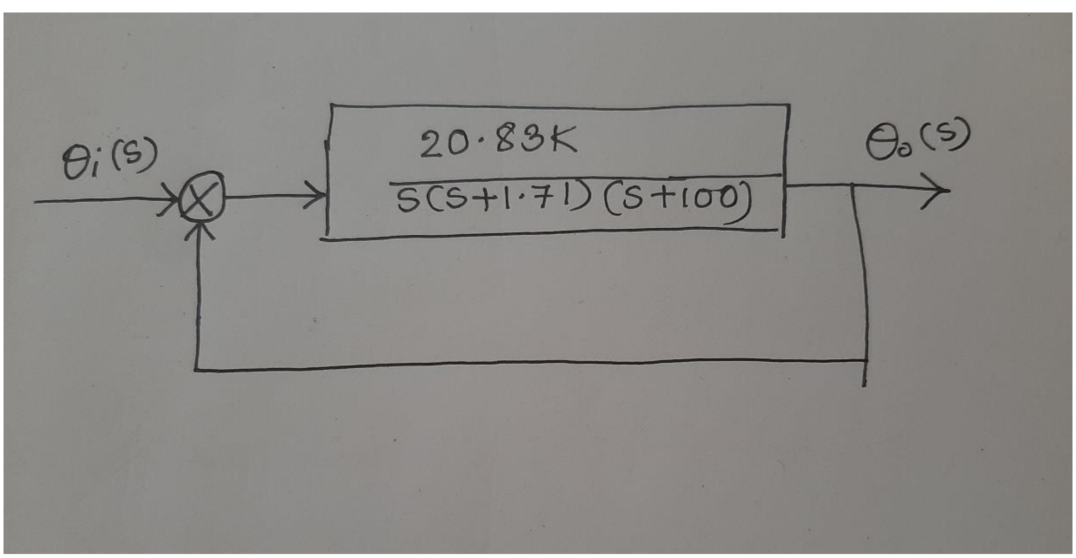


Figure 4: Schematic of Antenna Azimuth Control System

- Problem of this statement are given below:

Problem no: 01

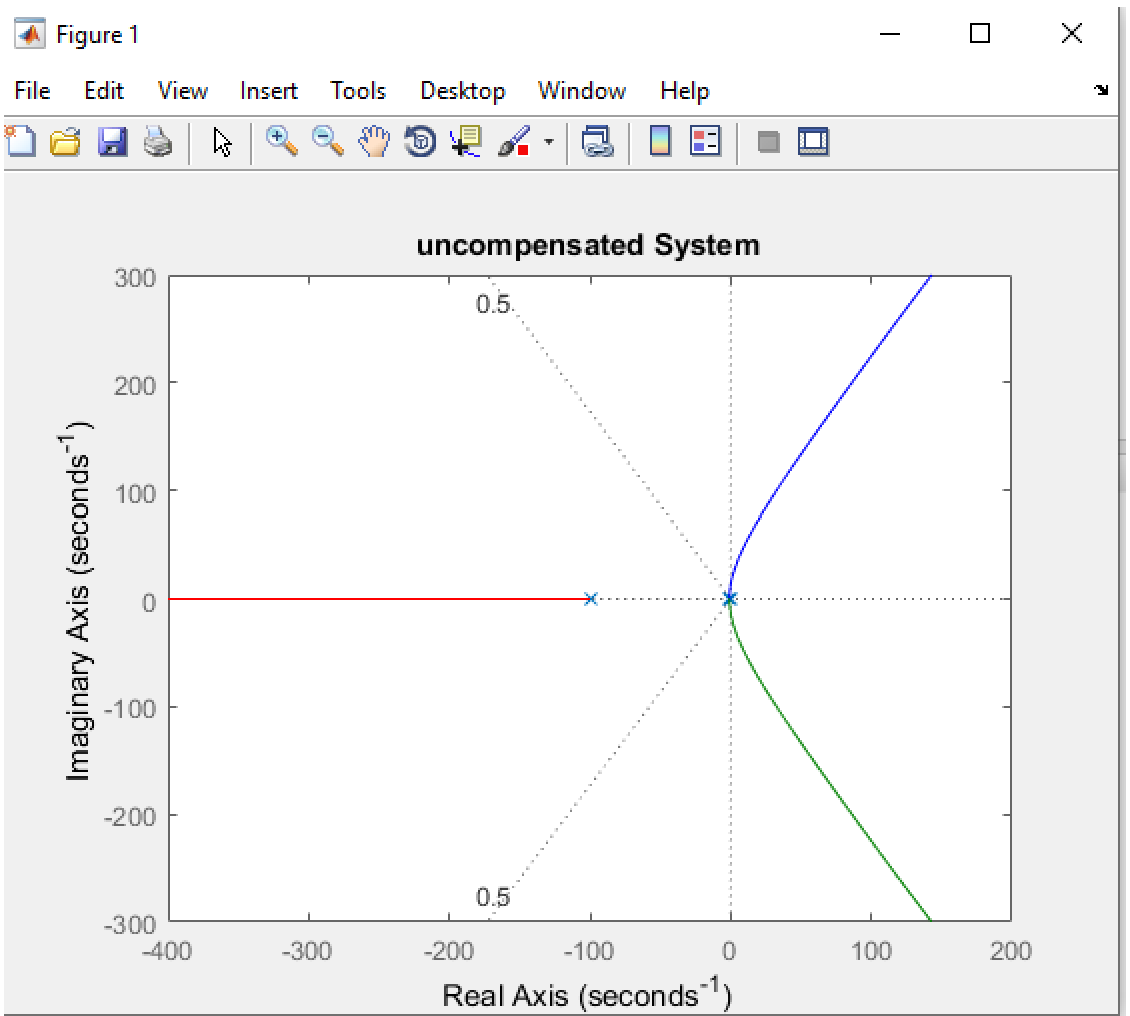
- **Unity Feedback:**



Problem no: 02

Code:

```
K=1; R=4;
s = tf('s'); % Define Laplace Variable
Gs = 20.83*K/(s*(s + 1.71)*(s +100)); % Define OL TF, G(s)
rlocus(Gs); % Drawing the RL of G(s)
sgrid(0.5, 0); % Drawing a line along zeta=0.5 at the s=plane
title('uncompensated System');
saveas(gcf, 'c.png');
% sgrid() is a function in the Matlab tool box. The variables in the sgrid
% command are the zeta term (0.5 corresponds to a overshoot of 16%)
% and the w_n term (natural freq.) respectively.]
```

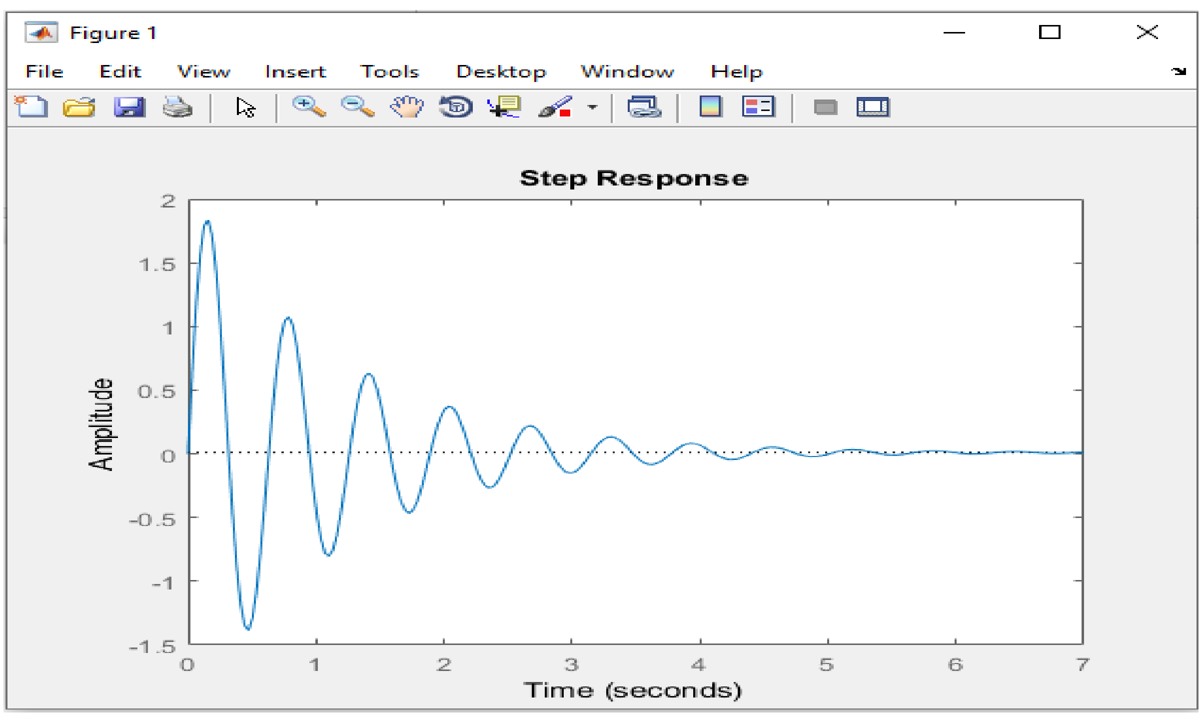


Problem no: 03

Code:

```
clc;
clear all;
close all;
Jm = 0.01;
Dm = 0.1;
k=1;
num = 20.83*k;
den=[ 1*(1+1.71)*(1+100)];
sys = tf([20.83 k],[1 * 1 1.71 * 1 100]);
SysInfo=stepinfo(sys)
step(sys);
```

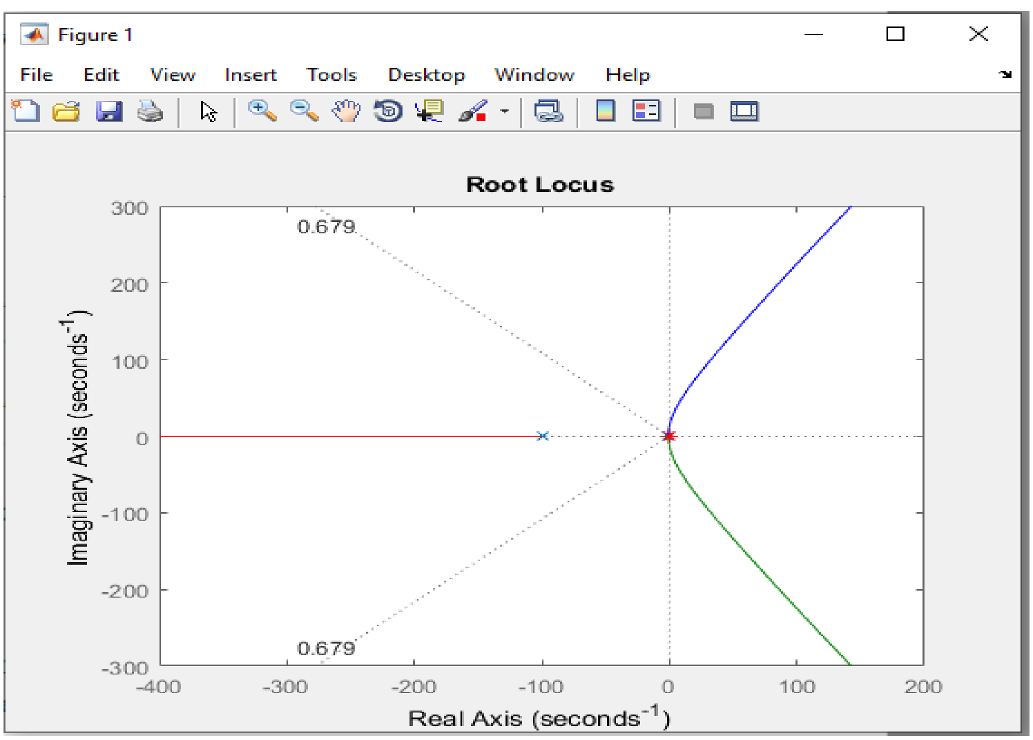
```
SysInfo =

  struct with fields:

        RiseTime: 4.0073e-04
    SettlingTime: 4.6160
     SettlingMin: -1.3864
     SettlingMax: 1.8371
       Overshoot: 1.8271e+04
      Undershoot: 1.3864e+04
            Peak: 1.8371
        PeakTime: 0.1571
```



Code:

```
j = sqrt(-1); K = 1; % Define j as imaginary unit
OS = 18.271 ; Ts = 230.8 ; % Given Specifications
sp_1= -(4/Ts)*( 1 - j*pi/log(1/OS) ); % Equiv s=point of the given data
sp_2= conj(sp_1);
poles = [sp_1, sp_2]' % These poles are the points that we want
zeta= cos(pi-angle(sp_1)); % Find zeta
s = tf('s'); % Define Laplace variable
Gs = 20.83*K/(s*(s+1.71)*(s+100)); %Open loop transfer function
figure(1)
rlocusplot(Gs); hold on; % Plotting the Root Locus
plot(real(poles), imag(poles), 'r*') % Plotting the desired poles
sgrid(zeta, 0);
title('Root Locus');
saveas(gcf, 'z.png')
```

```
poles =

  -0.0173 + 0.0187i
  -0.0173 - 0.0187i
```

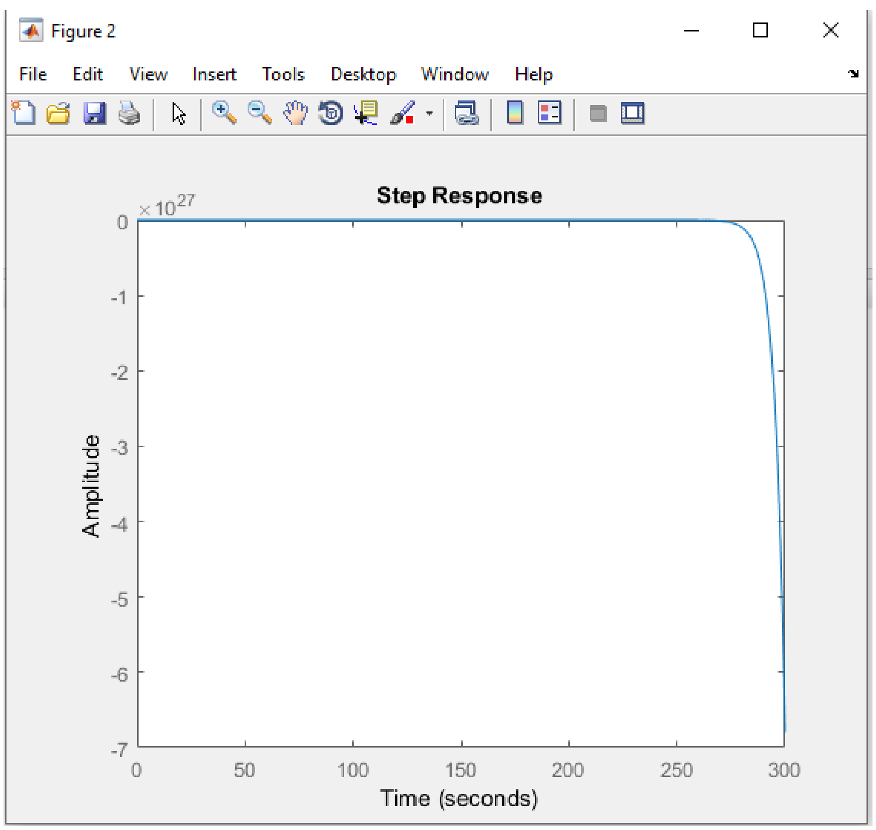


Problem no: 04

Code:

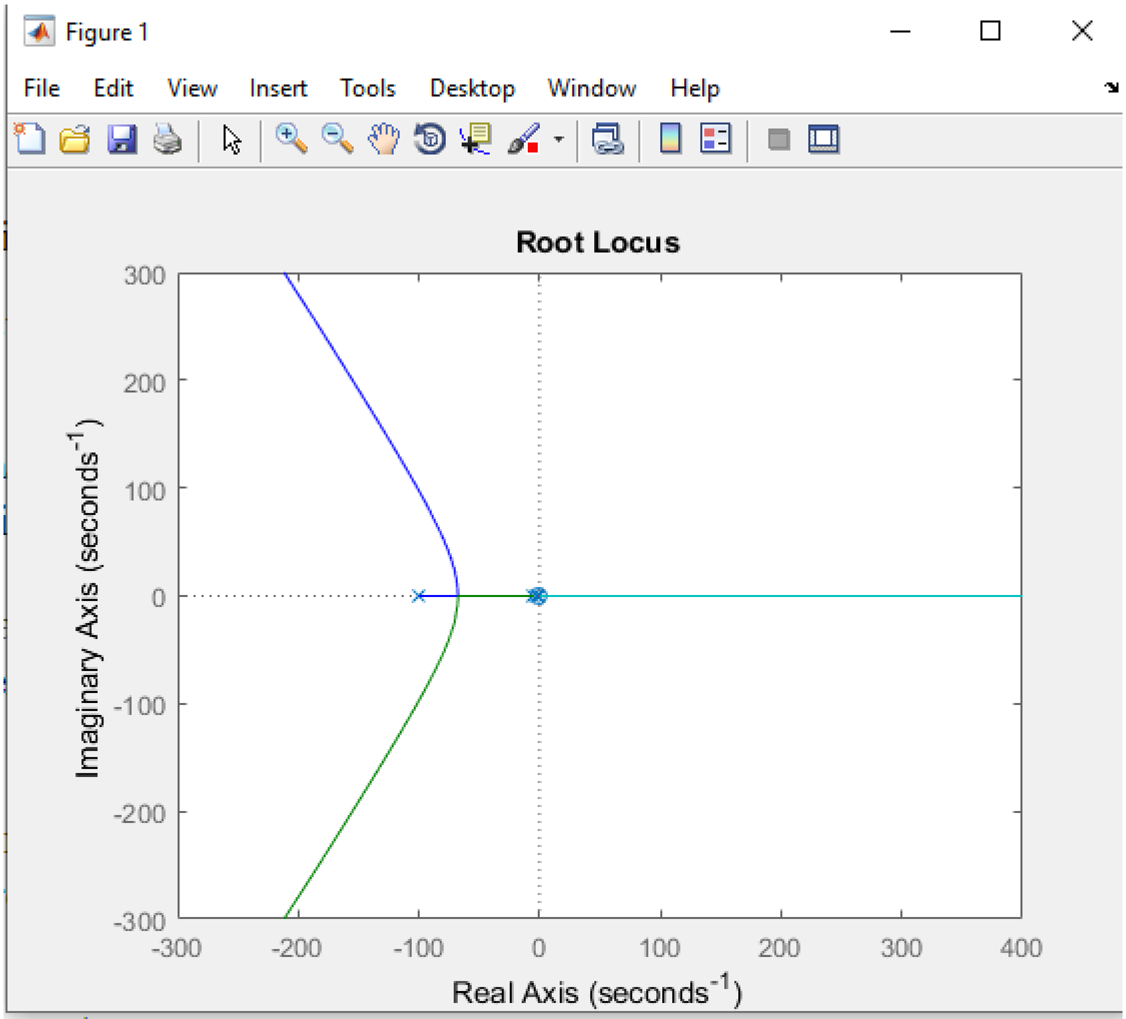
```
clc;
clear all;
s = tf('s');
K =1;
G = 20.83*K/(s*(s + 1.71)*(s +100)); % open loop transfer function
LEAD = - 8.71217612*(s+0.8)/(s+4.257164); %Lead compensator calculated
sys = G*LEAD % Function already compensated
CL = feedback(sys,1);
figure(1);
rlocus(CL);
figure();
step(CL);
```

```
sys =

         -181.5 s - 145.2
  ------------------------------
  s^4 + 106 s^3 + 604 s^2 + 728 s

Continuous-time transfer function.
```



Problem no: 05

Root Locus:



Problem no: 06

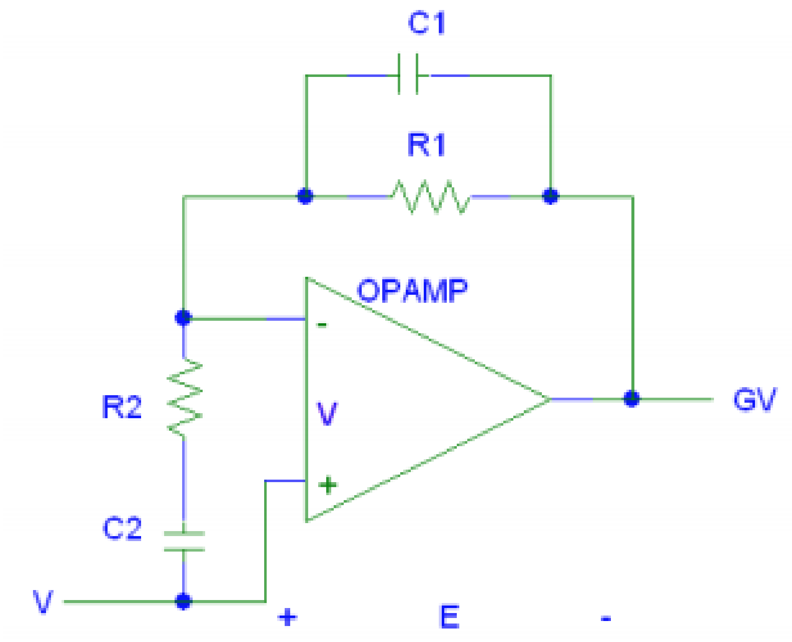


Fig: Lag-lead compensator using an op-amp circuit.