EE324 Problem Sheet 7

Aseer Israr Ansari, 18D070040

March 13, 2021

Q1

The open loop transfer function of the system is $G(s) = 1/((s+3)(s+4)(s+12))$.

(a) for z=0.01, to get a damping ratio of 0.2, we need to take k=665.5

Figure 1: Root Locus

(b) For z=0.01, to obtain undamped natural frequency of 8 rads/s, we need to take k = 951.8 , and to obtain undamped natural frequency of 9 rads/s, we need to take k = 1327

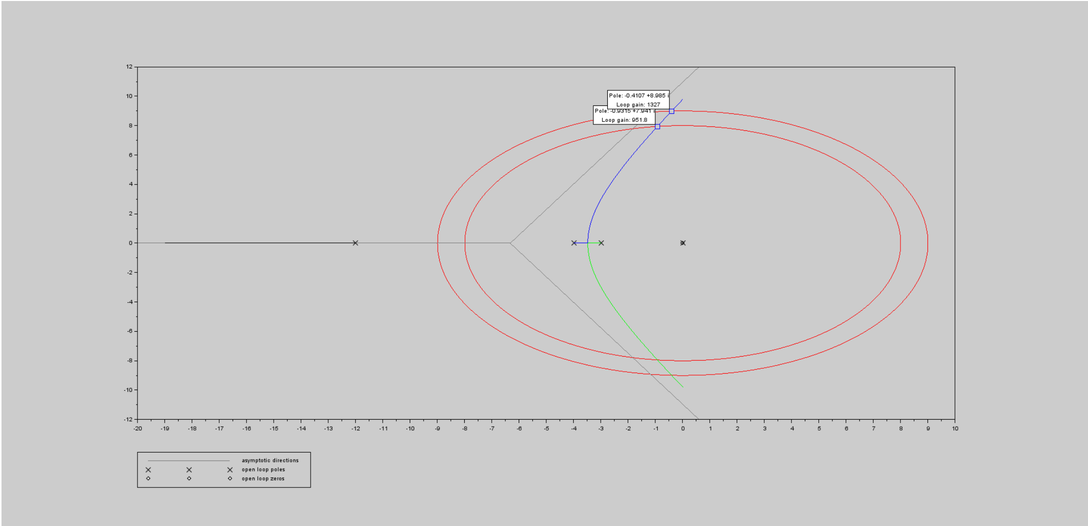


Figure 2: Root Locus

(c) The root locus for different values of z is given in Figure 3

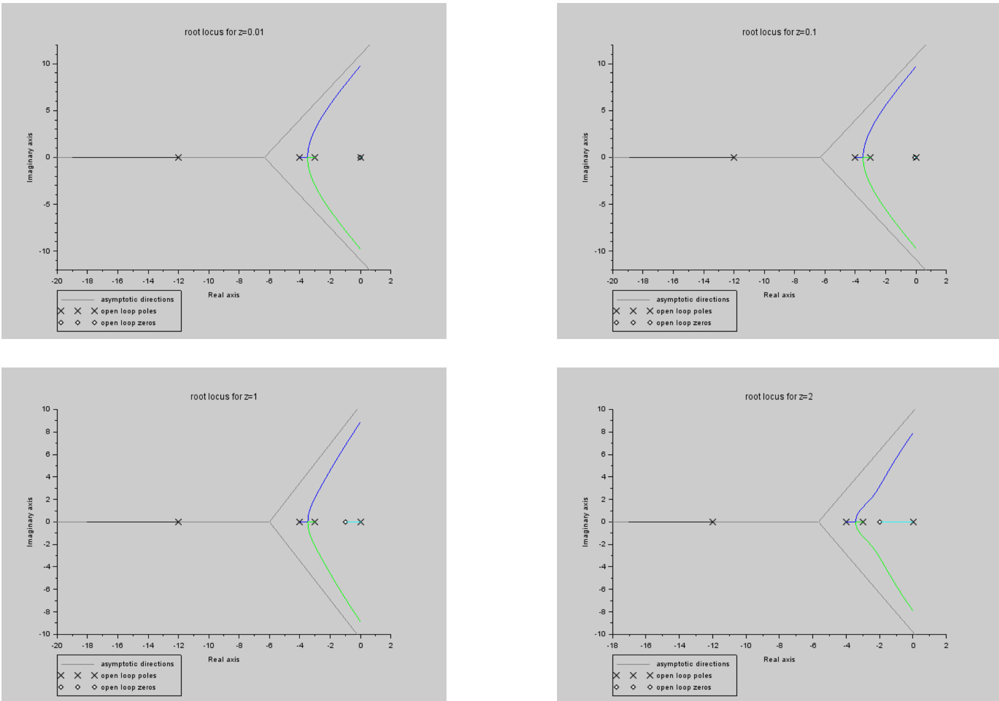


Figure 3: Root Locus for different values of z

(d)

Scilab Code

```
clc
clear

s = poly(0,'s')
G = 1/((s+3)*(s+4)*(s+12))


//Part A
figure
z = 0.01;
tan_phi = 0.2/(sqrt(1-0.2^2));
y = -10:0.01:10;
x = -1*abs(y)*tan_phi;
plot(x,y, 'r')
G_a = G*(s+z)/s;
evans(G_a,kpure(G_a))


//Part B
figure
w1 = 8;
w2 = 9;
t = 0:0.001:2*%pi;
x1 = w1*cos(t);
y1 = w1*sin(t);
x2 = w2*cos(t);
y2 = w2*sin(t);
plot(x1,y1,'r',x2,y2,'r')
z = 0.01;
G_b = G*(s+z)/s;
evans(G_b,kpure(G_b))


//Part C
z_arr = [2 1 0.1 0.01]
for z = z_arr
     G_c = G*(s+z)/s;
    figure,evans(G_c,kpure(G_c))
    title('root locus for z=' + string(z))
end
```

Q2

The open loop transfer function of the system is $G(s) = 1/(s^2 + 3s + 2)$.

(a) constant gain K to achieve 10% OS in the closed-loop is 4.385

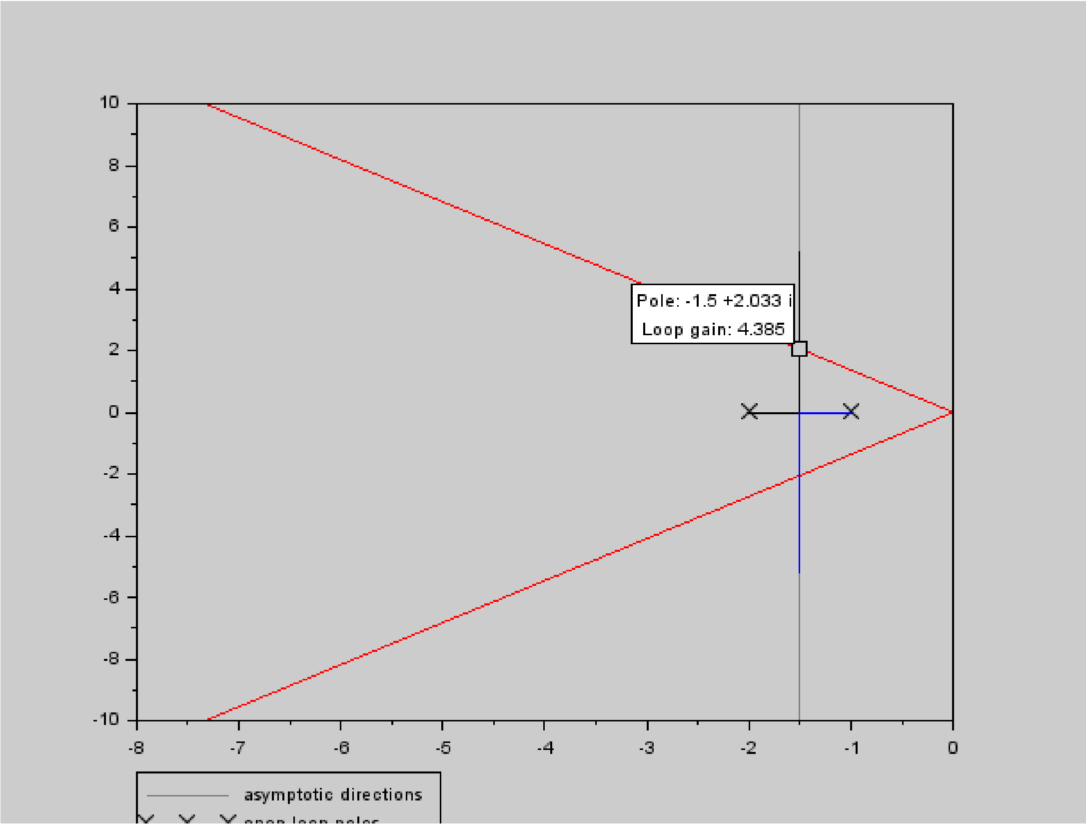


Figure 4: Root Locus

(b) steady-state error for k=4.385 is 0.31. after adding a lag-compensator with pole at s= -0.01 and zero at s=-0.2, steady state error is 0.022.

(c) The unit step response for different locations of pole and zero of the lag compensator, but with the same ratio are give in Figure 5. As the pole is moved away from origin, the settling time decreases.

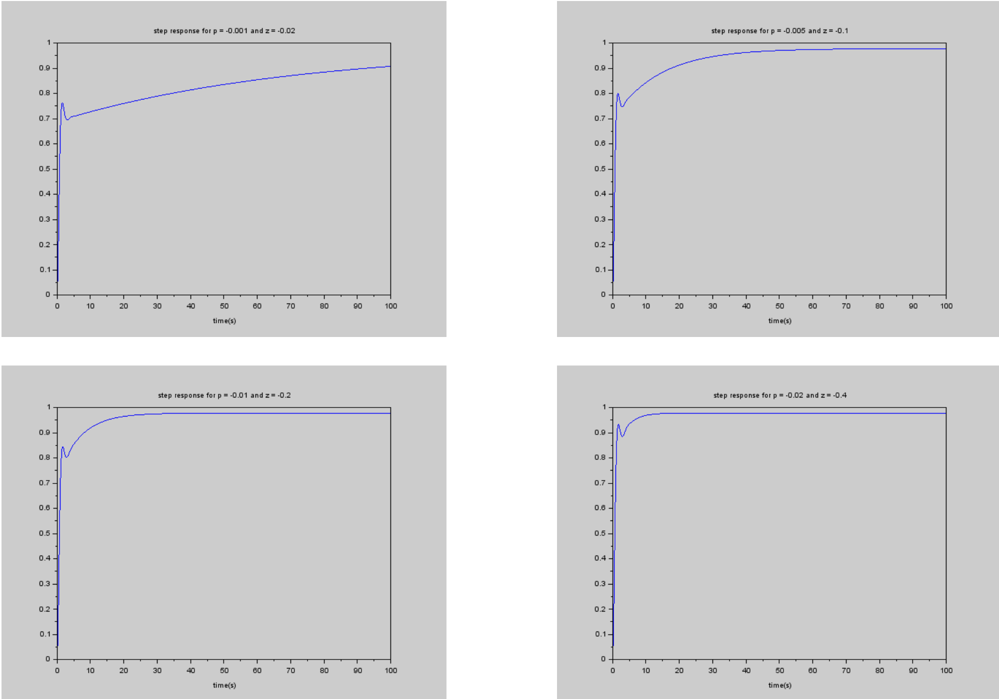


Figure 5: unit step response for different locations of pole and zero of the lag compensator
.

Scilab Code

```
clc
clear

s = poly(0,'s');
G = 1/(s^2+3*s+2);


// Part A
figure
os = 0.1;
tan_phi = -log(os)/(%pi);
y = -10:0.01:10;
x = -1*abs(y)*tan_phi;
plot(x,y, 'r')
evans(G,kpure(G))

k = 4.385;  //obtained from graph
// Part B
sse = 1/(1+ k*horner(G,0))
disp('steady state error = ' + string(sse))

p = 0.01;
z = 20*p;
G_c = G*(s+z)/(s+p);
sse = 1/(1+ k*horner(G_c,0))
disp('steady state error with lag compensator= ' + string(sse))
```

```
//Part c
p_arr = [0.02 0.01 0.005 0.001];
t = 0:0.1:1000;
for p = p_arr
    z = 20*p;
    G_d = G*(s+z)/(s+p);
    G_d = k*G_d/(1+k*G_d)
    sys_G = syslin('c',G_d);
    y = csim('step',t,sys_G);
    figure,plot(t(t<100),y(t<100))
    title('step response for p = ' + string(-p)+ ' and z = ' + string(-z))
    xlabel('time(s)')
    end
```

Q3

The open loop transfer function of the system is $G(s) = 1/(s^2 + 3s + 2)$.

(a) The lead compensator has pole at -10.089, and zero at -4.508, and the proportional gain is 36.022.

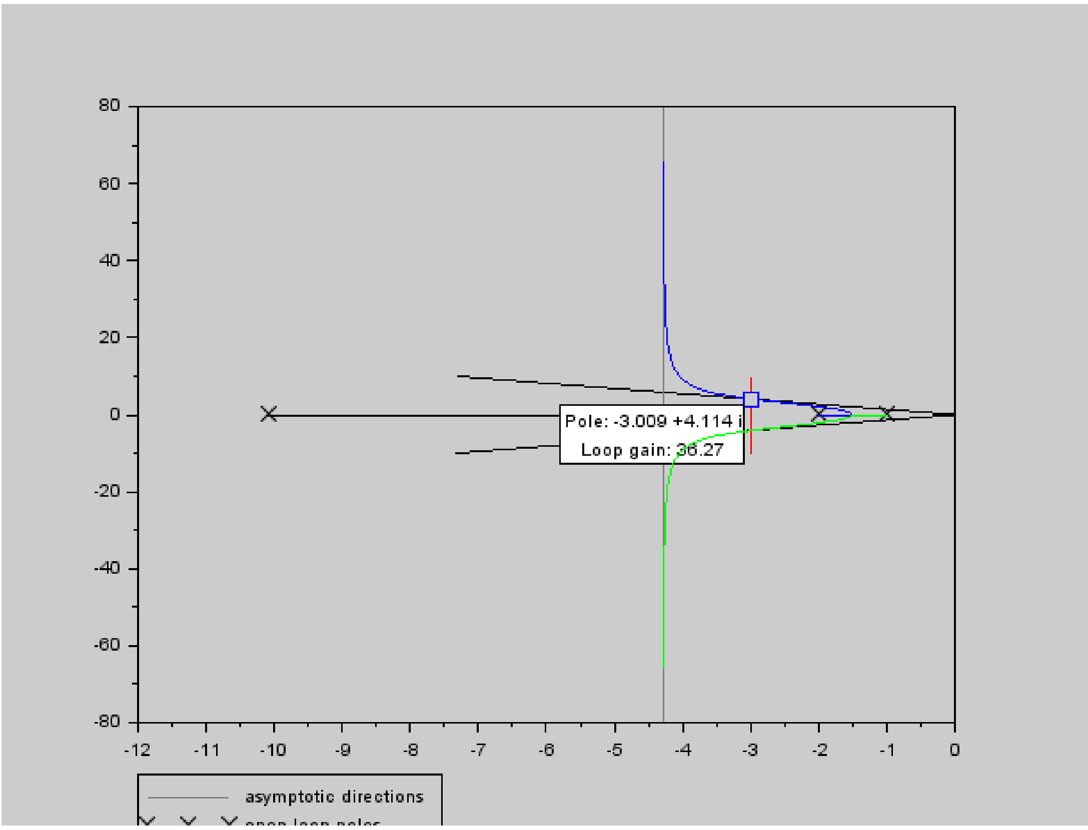


Figure 6: Root Locus

(b) The PD controller has a zero at -7.918, the proportional gain is 3.

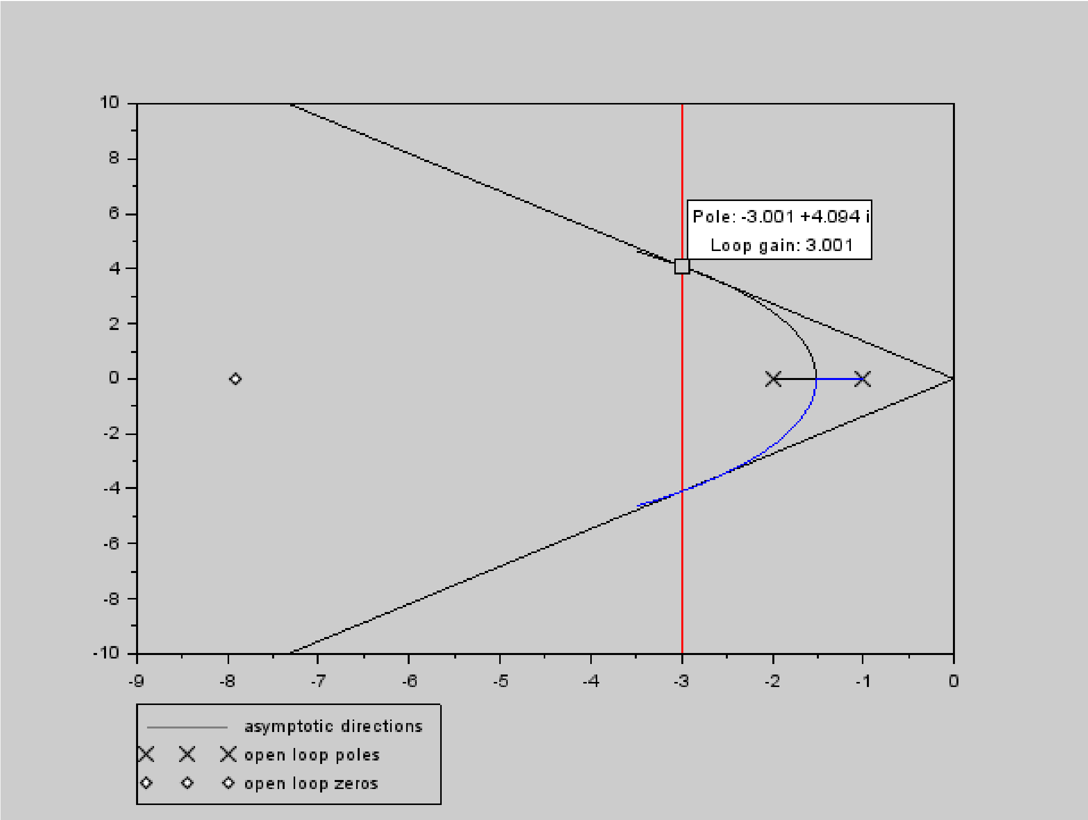


Figure 7: Root Locus

Scilab Code

```
clc
clear

s = poly(0,'s');
G = 1/(s^2+3*s+2);
// Part A

os = 0.1;
k = 4.385;
G_2_c = k*G/(1+k*G);

poles = roots(G_2_c('den'));

t_s = mean(-4./real(poles));
t_s = t_s/2;

x = (-4/t_s);

tan_phi = -log(os)/(%pi);
y = -x/tan_phi;

s_l = x + %i*y;
poles = roots(G('den'));
```

```
theta_zminp = 180;

for i = 1:length(poles)
    temp = poles(i)-s_l;
    [a,arg] = polar(temp);
    arg = real(arg)*180/%pi;
    theta_zminp = theta_zminp + arg;
end

theta_p = 30;
theta_z = theta_p + theta_zminp;
z1 = real(s_l) - imag(s_l)/tan(theta_z*%pi/180);
p1 = real(s_l) - imag(s_l)/tan(theta_p*%pi/180);
disp('Part a')
disp('lead compensator has a pole and zero at ' + string(p1) + ' ' + string(z1) +
    ' respectively')


G1 = G*(s-z1)/(s-p1);
k1 = real(-1/horner(G1,s_l));
disp('the proportional gain is ' + string(k1))

y = -10:0.01:10;
x1 = real(s_l)*ones(1,length(y))
x2 = -1*abs(y)*tan_phi;
figure
plot(x1,y,'r',x2,y,'black')
evans(G1)


// Part B
disp('Part b')
theta_z = theta_zminp;
z2 = real(s_l) - imag(s_l)/tan(theta_z*%pi/180);
disp('Pd controller has a zero at ' + string(z2))
G2 = G*(s-z2);
k2 = real(-1/horner(G2,s_l));
disp('the proportional gain is ' + string(k2))

y = -10:0.01:10;
x1 = real(s_l)*ones(1,length(y))
x2 = -1*abs(y)*tan_phi;
figure
plot(x1,y,'r',x2,y,'black')
evans(G2,4)
```

Q4

(a) For the transfer function $G(s) = 1/(s^2 + 5s + 6)$ the input and output signals for different frequencies is given in Figure 8. The phase response and magnitude response are given in Figure 9

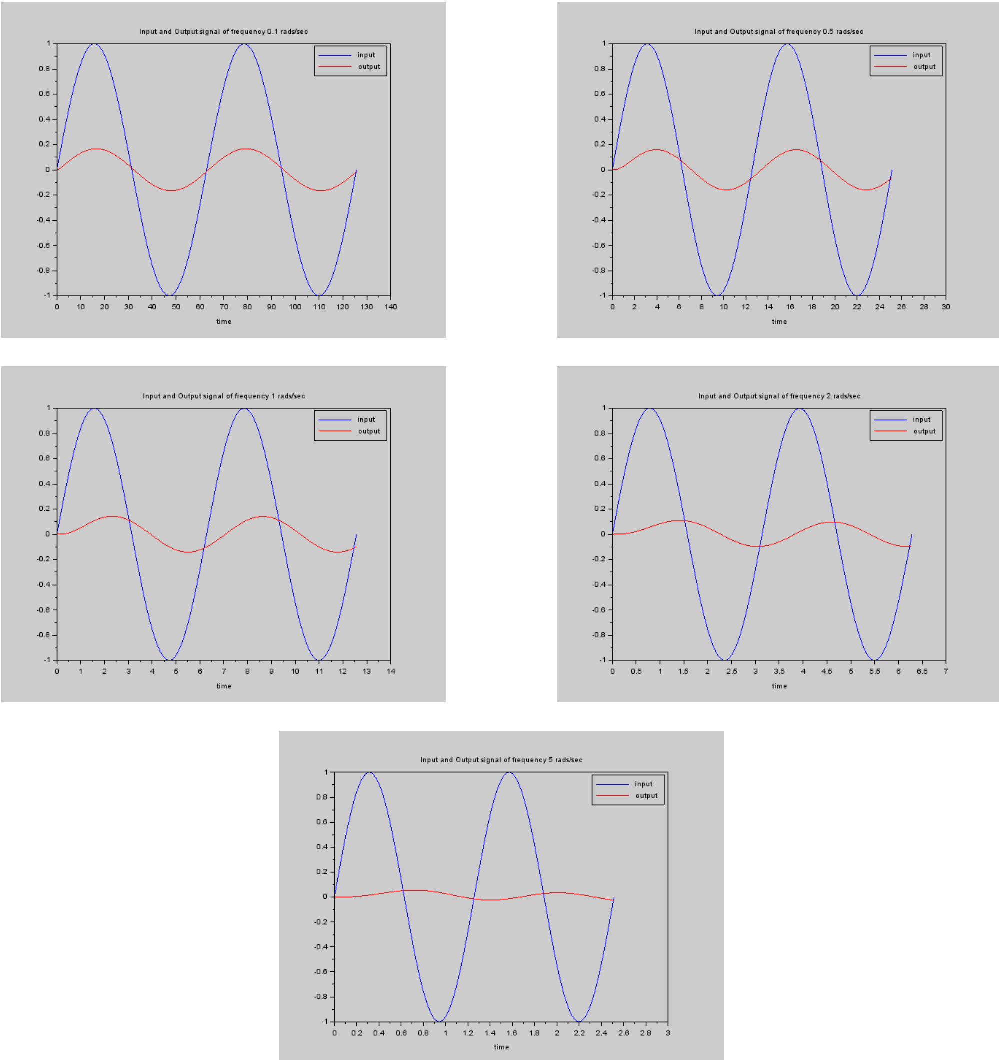
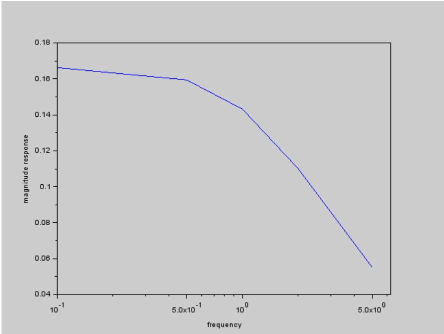


Figure 8: input output signal of different frequencies

.



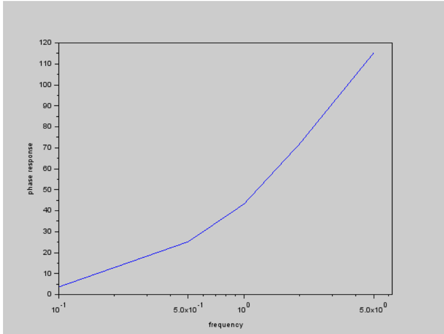


Figure 9: input output signal of different frequencies

.

(b) The frequency measured is in rad/s.

(c) For the transfer function $G(s) = 60/(s^3 + 6s^2 + 11s + 6)$ the input and output signals for different frequencies is given in Figure 10. The phase response and magnitude response are given in Figure 11.// The frequency when the phase angle difference is 180 degrees is 3.32 rad/sec. If the numerator had a negative number, the frequency would have been 0 rad/sec.

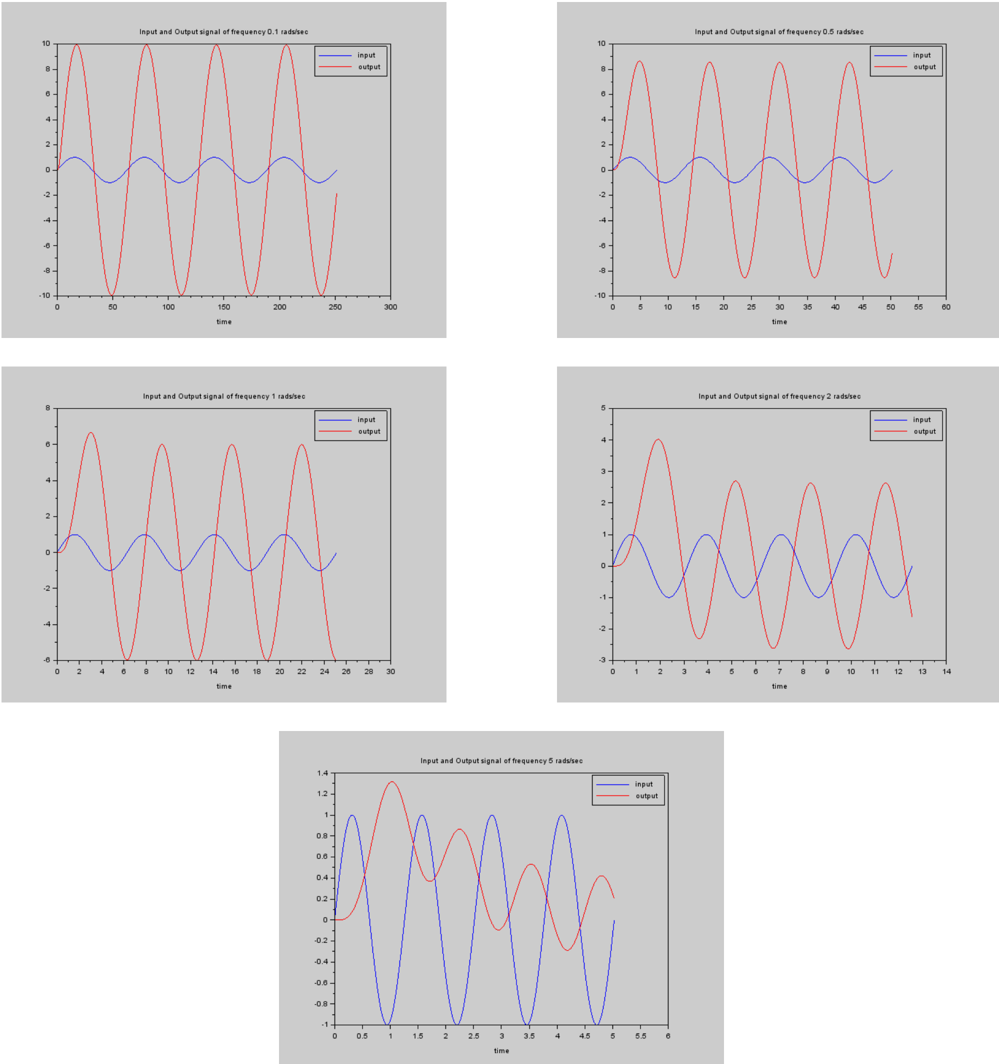


Figure 10: input output signal of different frequencies

.

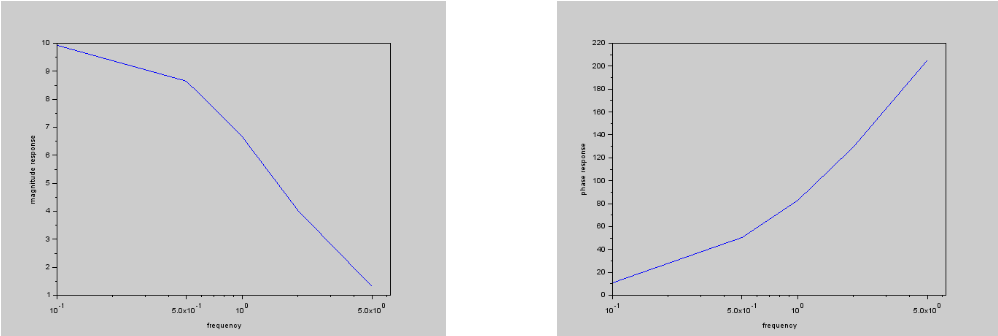


Figure 11: input output signal of different frequencies

.

Scilab Code

```
clc
clear

s = poly(0,'s');

// Part A
G = 1/(s^2+5*s+6);

sys_G = syslin('c',G);


omega_arr = [0.1 0.5 1 2 5];
mag = zeros(1,4)
phase = zeros(1,4)

i = 1
for w = omega_arr
    t = 0:%pi/(50*w):4*%pi/w;
    u = sin(w*t);
    y = csim(u,t,sys_G);
    mag(i) = max(y);
    t1 = t(y == max(y));
    t1 = t1(1);
    t2 = t(u == max(u));
    t2 = t2(1);
    phase(i) = (t1 - t2)*w*180/%pi;
    i = i+1;
    figure
    plot(t,u,'b')
    plot(t,y,'r')
    title('Input and Output signal of frequency ' + string(w) + ' rads/sec')
    xlabel('time')
    legend('input','output')
end

figure,plot('ln',omega_arr,mag)
ylabel('magnitude response')
xlabel('frequency')
figure,plot('ln',omega_arr,phase)
ylabel('phase response')
```

```
xlabel('frequency')



// Part C
G = 60/(s^3+6*s^2+11*s+6);

sys_G = syslin('c',G);


omega_arr = [0.1 0.5 1 2 5];
mag = zeros(1,4)
phase = zeros(1,4)

i = 1
for w = omega_arr
    t = 0:%pi/(50*w):8*%pi/w;
    u = sin(w*t);
    y = csim(u,t,sys_G);
    mag(i) = max(y);
    t1 = t(y == max(y));
    t1 = t1(1);
    t2 = t(u == max(u));
    t2 = t2(1);
    phase(i) = (t1 - t2)*w*180/%pi;
    i = i+1;
    figure
    plot(t,u,'b')
    plot(t,y,'r')
    title('Input and Output signal of frequency ' + string(w) + ' rads/sec')
    xlabel('time')
    legend('input','output')
end

figure,plot('ln',omega_arr,mag)
ylabel('magnitude response')
xlabel('frequency')
figure,plot('ln',omega_arr,phase)
ylabel('phase response')
xlabel('frequency')

w = poly(0,'w')
G_w = horner(G,%i*w);
w1 = roots(imag(G_w('den')));
w1 = w1(w1>0);
disp('frequency when the phase angle difference is 180 degrees is ' + string(w1))
```