Example20

$$G_p(s) = \frac{1}{s(s+1)},$$

For a sampling period of $T = 1$ s.

Write MATLAB program for finding G(z) and C(kT).

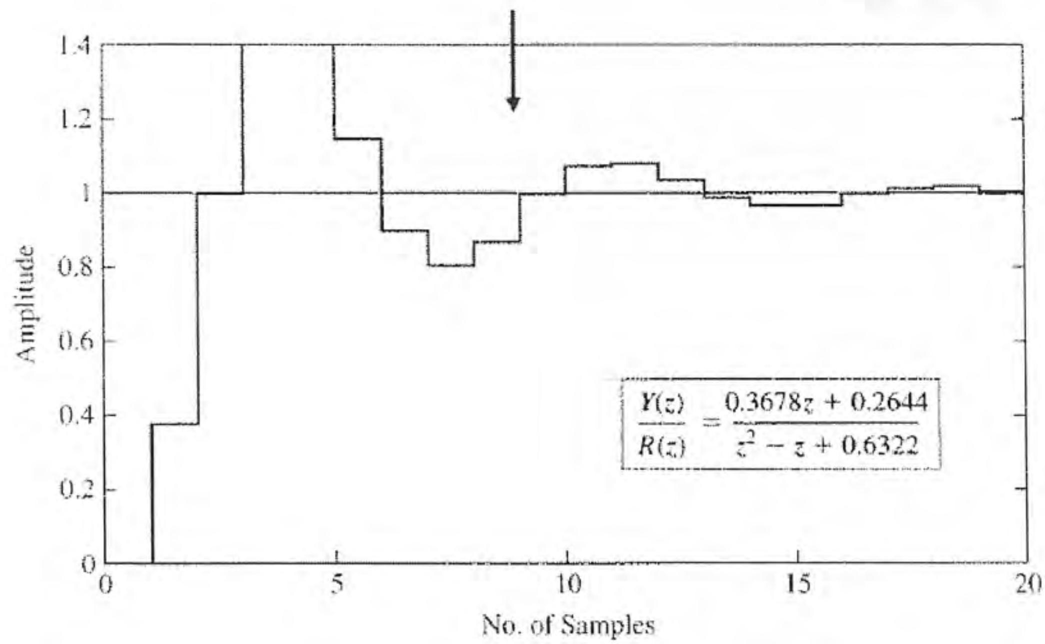
```
% This script converts the transfer function
% Gp(s) = 1/s(s+1) to a discrete-time system
% with a sampling period of T=1 sec.
%
num=[1]; den=[1 1 0]; sysc=tf(num,den);
T=1;
[sysd]=c2d(sysc,T,'zoh');
```

Sampling time → T=1; Zero-order hold → 'zoh'; Output:

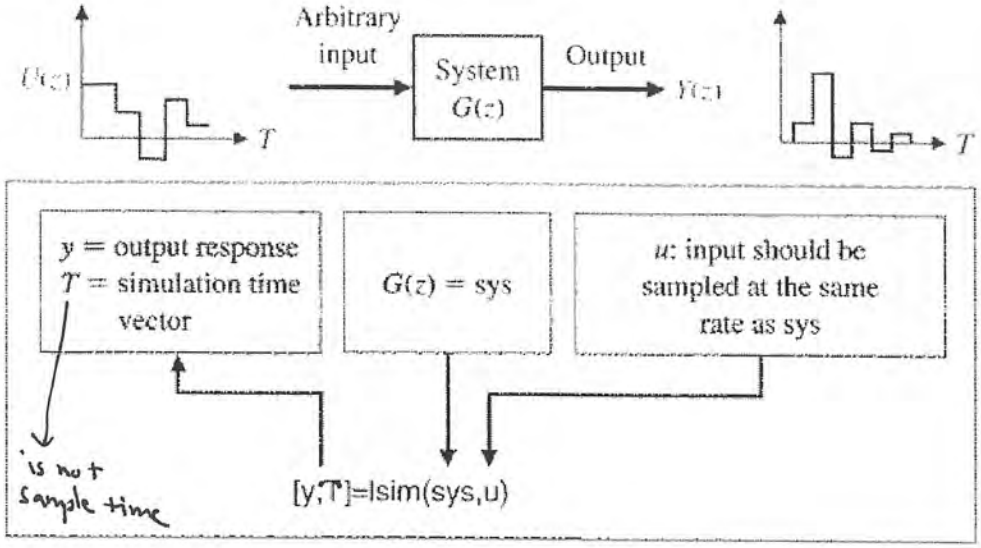
```
>>
Transfer function:
 0.3679 z + 0.2642
---------------------
z^2 - 1.368 z + 0.3679

Sampling time: 1
```

The functions step, impulse, and lsim are used for simulation

```
% This script generates the unit step response, y(kT),
% for the sampled data system given in Example
%
num=[1]; den=[1 1 0];
sysc=tf(num,den);
sysd=c2d(sysc,1,'zoh');
sys=feedback(sysd,[1]);
T=[0:1:20]; step(sys,T)
```



We can use step input, or, impulse or any arbitrary input.



Example21

Write MATLAB program for plotting R-L.

the process was given by

$$G(z) = \frac{0.3678(z + 0.7189)}{(z - 1)(z - 0.3680)}$$

The compensator is selected to be

$$D(z) = \frac{K(z - 0.3678)}{z + 0.2400},$$

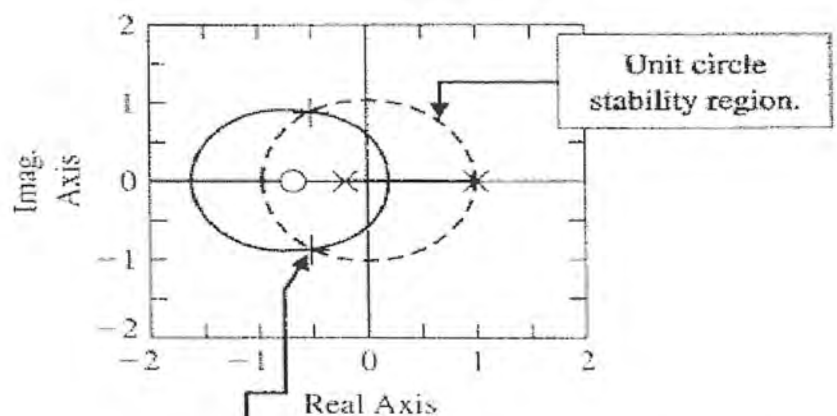
$$G(z)D(z) = K\frac{0.3678(z + 0.7189)}{(z - 1)(z + 0.2400)}.$$

```
% This script generates the root locus for
% the sampled data system
%
%     K(0.3678)(z+0.7189)
%     ---------------------------
%       (z-1)(z+0.2400)
%
num=[0.3678 0.2644]; den=[1.0000 -0.7600 -0.2400]; sys=tf(num,den);
rlocus(sys);hold on
x=[-1:0.1:1];y=sqrt(1-x.^2);            <---- Plot unit circle.
plot(x,y,'--',x,-y,'--')
```

Or Using the statement below for plotting unit-circle

uc= exp(j*[0:0.01:2*pi]);

plot(real(uc),imag(uc));




```
>>rlocfind(sys)
Select a point in the graphics window

selected_point =
 -0.4787 + 0.8530i

ans =
 4.6390
```

Determine K at the unit circle boundary.

$K = 4.639$

* Example22

Root Locus Based Controller Design Using MATAB

In this Example we will show how the MATLAB platform can be utilized to design a controller using root locus technique.

Consider the closed loop discrete control system as shown in Figure 1. Design a digital controller such that the closed loop system has zero steady state error to step input with a reasonable dynamic performance. Velocity error constant of the system should at least be 5. ← or equal

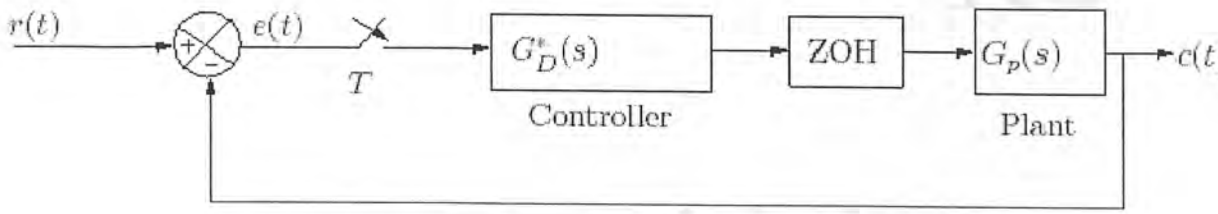


Figure 1: A discrete time control system

$$G_p(s) = \frac{10}{(s+1)(s+2)}, \qquad T = 0.1 \text{ sec}$$

$$\begin{aligned} G_{h0}G_p(z) &= Z\left[\frac{1-e^{-0.1s}}{s}\frac{10}{(s+1)(s+2)}\right] \\ &= (1-z^{-1})Z\left[\frac{10}{s(s+1)(s+2)}\right] \\ &\cong \frac{0.04528(z+0.9048)}{(z-0.9048)(z-0.8187)} \end{aligned}$$

The MATLAB script to find out $G_{h0}G_p(z)$ is as follows.

```
>> s=tf('s');
>> Gp=10/((s+1)*(s+2));
>> GhGp=c2d(Gp,0.1,'zoh');
```

or

```
>> n=10;
>> d=[1 3 2]
>> sys=tf(n,d)
>> sysd=c2d(sys,0.1,'zoh')
```

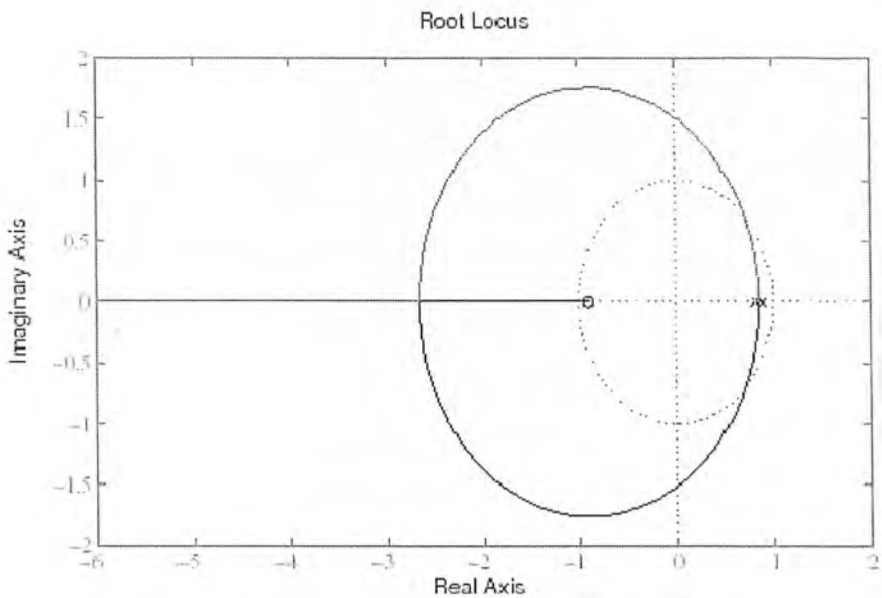


Figure 2: Root locus of the uncompensated system

The root locus of the uncompensated system (without controller) is shown in Figure 2 for which the MATLAB command is

>> rlocus(GhGp)

or

>> rlocus(sysd)

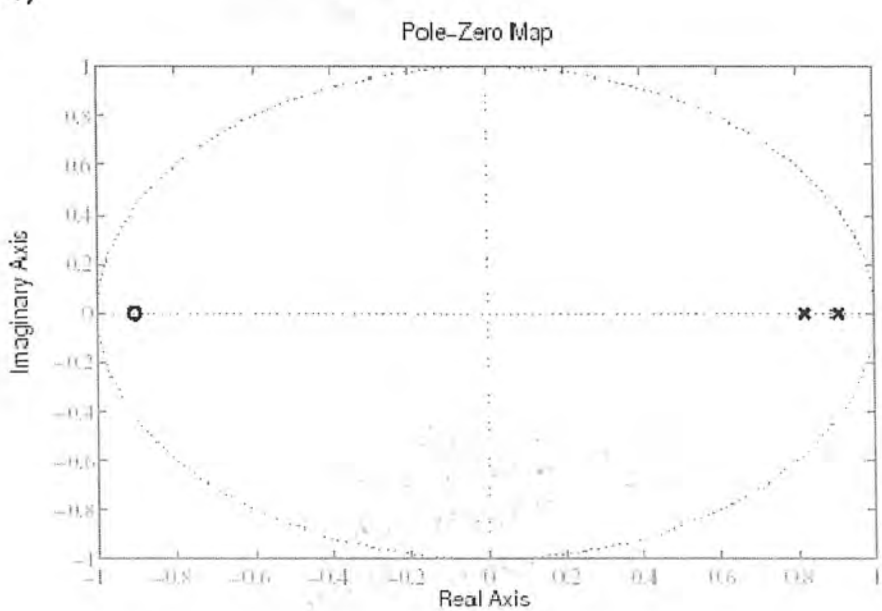


Figure 3: Pole zero map of the uncompensated system

Pole zero map of the uncompensated system is shown in Figure 3 which can be generated using the MATLAB command

>> pzplot(GhGp)

One of the design criteria is that the closed loop system should have a zero steady state error for unit step input. Thus a PI controller is required which has the following transfer function in z-domain when backward rectangular integration is used.

$$G_D(z) = K_p + \frac{K_i T}{z-1} = \frac{K_p z - (K_p - K_i T)}{z-1}$$

The parameter K_i can be designed using the velocity error constant requirement.

$$k_v = \frac{1}{T}\lim_{z\to 1}(z-1)G_D(z)G_{h0}G_p(z) = 5K_i \geq 5$$

Above condition will be satisfied if $K_i \geq 1$. Let us take $K_i = 1$. With $K_i = 1$, the characteristic equation becomes

$$(z-1)(z-0.9048)(z-0.8187) + 0.004528(z+0.9048) + 0.04528K_p(z-1)(z+0.9048) = 0$$

or,

$$1 + \frac{0.04528K_p(z-1)(z+0.9048)}{z^3 - 2.724z^2 + 2.469z - 0.7367} = 0$$

Now, we can plot the root locus of the compensated system with K_p as the variable parameter. The MATLAB script to plot the root locus is as follows.

```
>> z=tf('z',0.1);

>> Gcomp=0.04528*(z-1)*(z+0.9048)/(z^3 - 2.724*z^2 + 2.469*z - 0.7367);

>> zero(Gcomp);

>> pole(Gcomp);

>> rlocus(Gcomp)
```

The zeros of the system are 1 and - 0.9048 and the poles of the system are 1.0114 ± 0.1663 i and 0.7013 respectively. The root locus plot is shown in Figure 4.

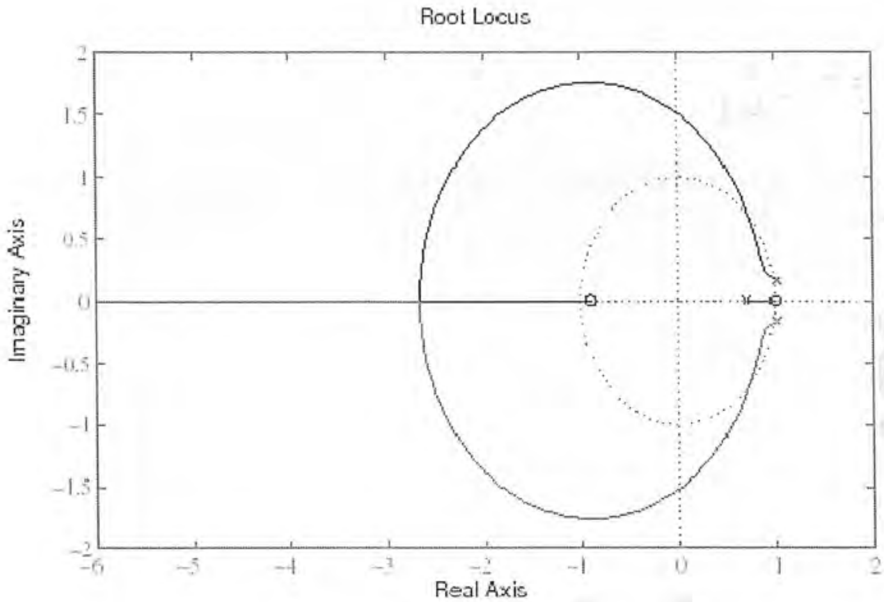


Figure 4: Root locus of the system with PI controller

It is clear from the figure that the system is stable for a very small range of K_p. The stable portion of the root locus is zoomed in Figure 5.

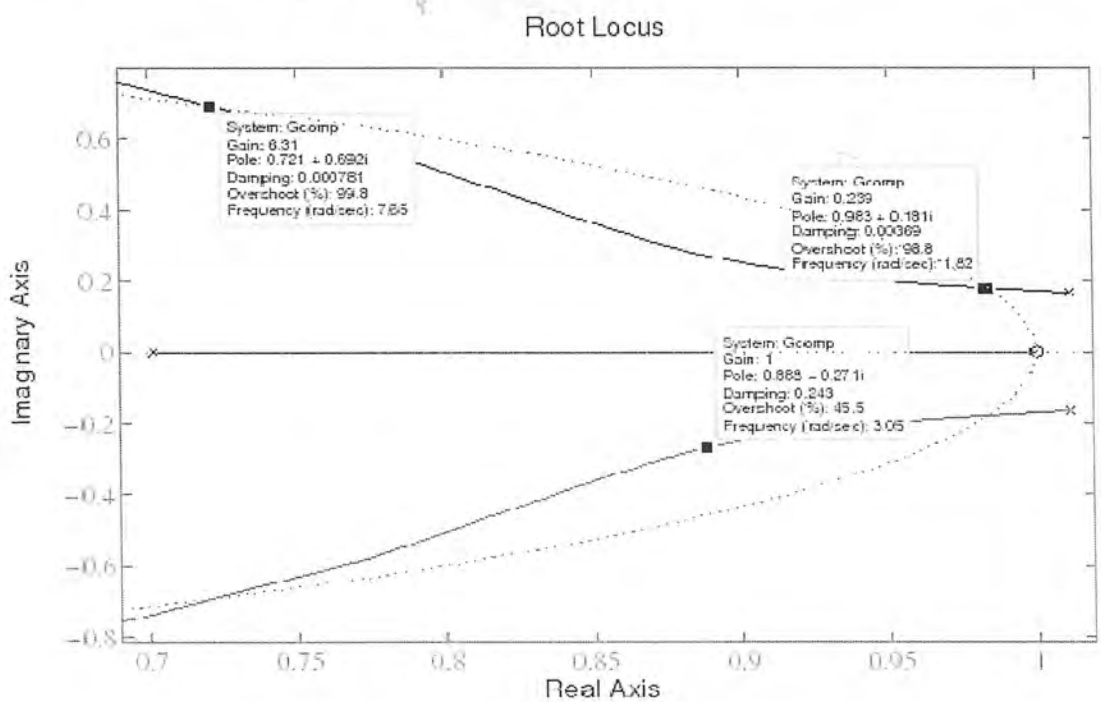


Figure 5: Root locus of the system with PI controller