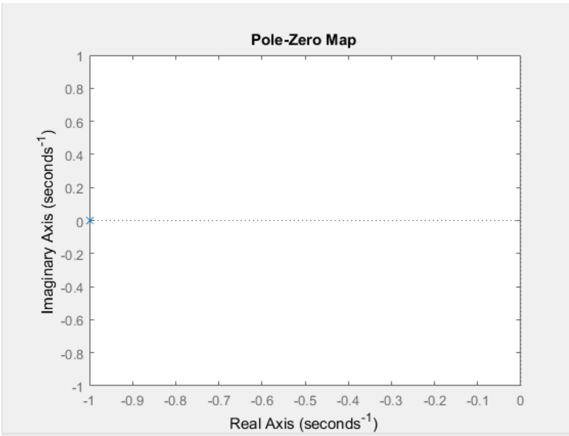
System Theory

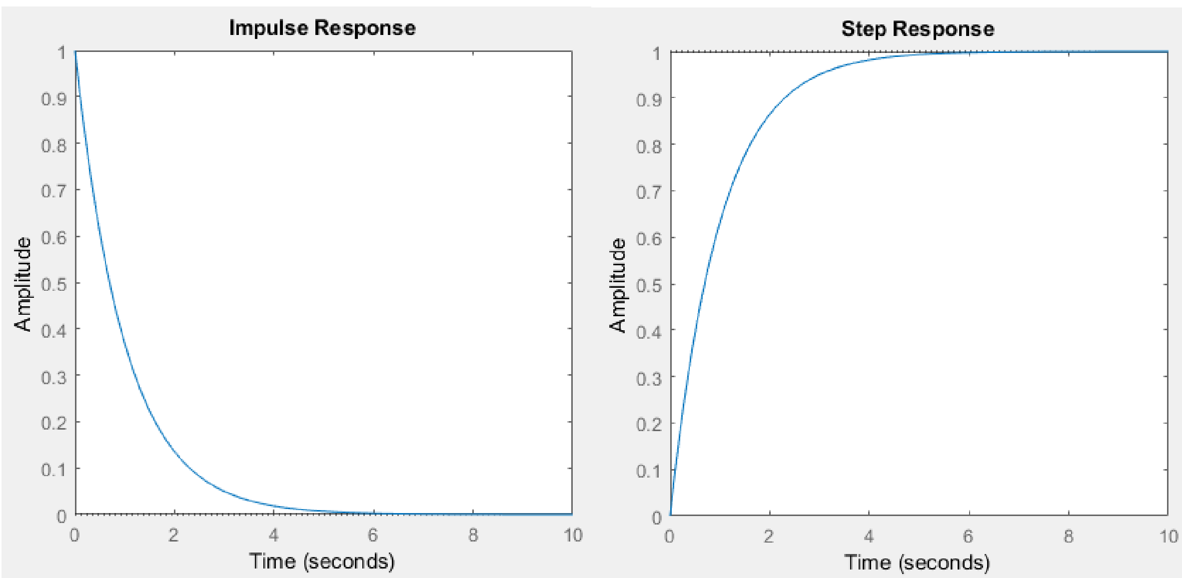
2. Stability analysis and simulation of transfer function

Poles are the roots of the corresponding denominators

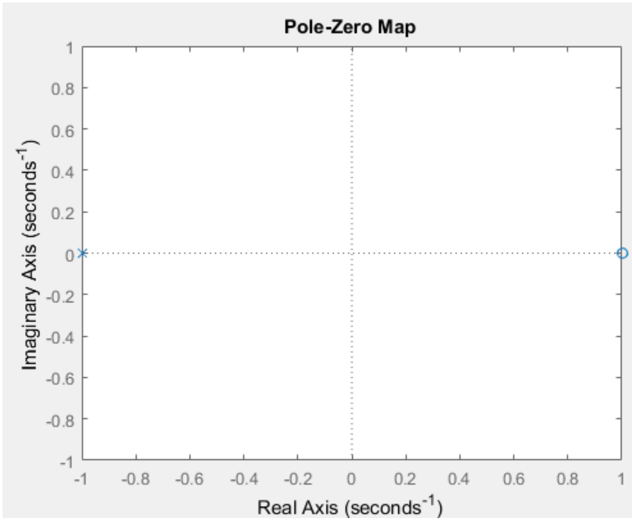
(1) $H1 = \frac{1}{s+1}$; pole1=-1;



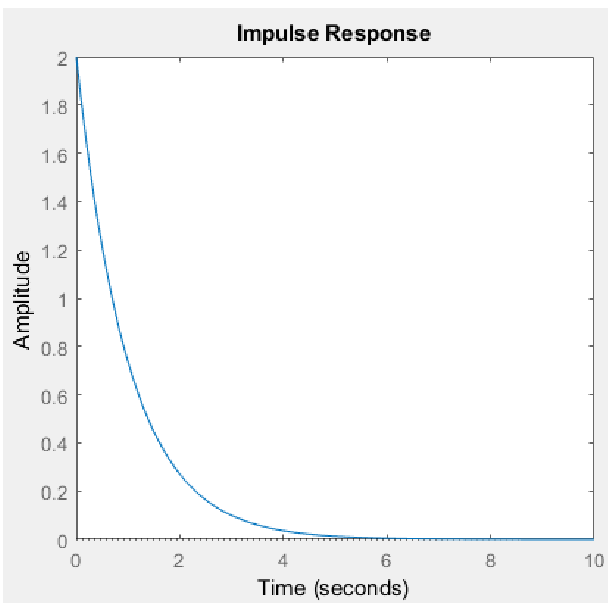
The pole lies in the left plane. The system is asymptotically stable.

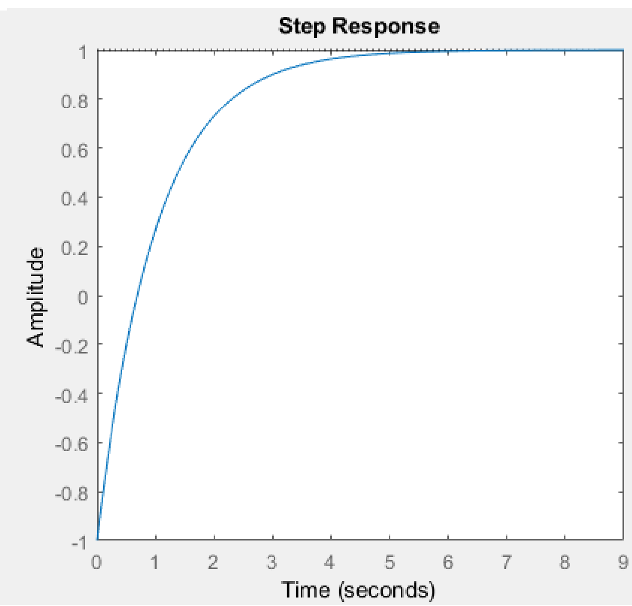


(2) $H_1(s) = \frac{1-s}{1+s}$, pole2 = -1



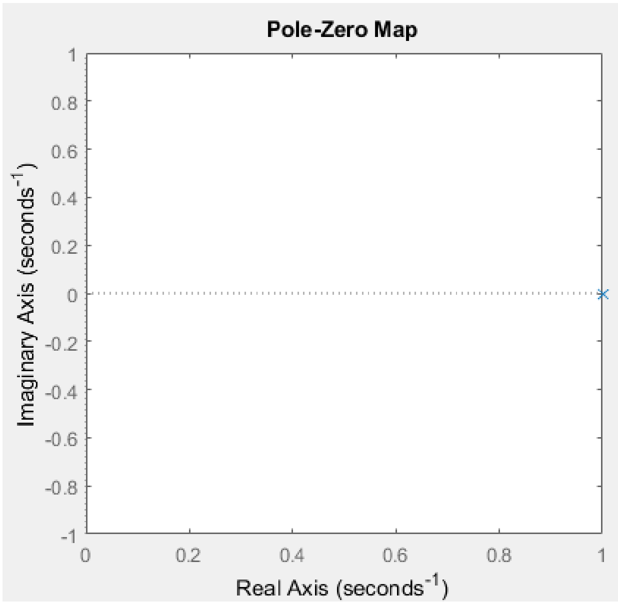
The pole locates in the left side of the plane. It is therefore asymptotically stable.

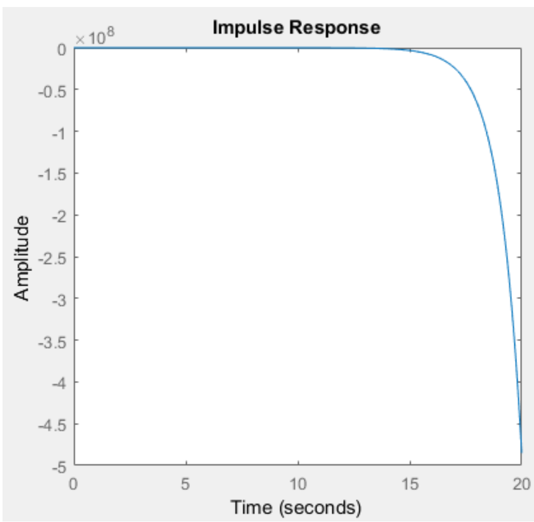


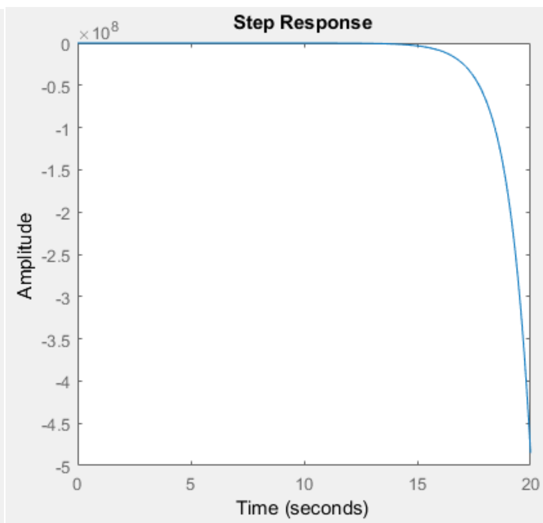


(3) $H_3(s) = \frac{1}{1-s}$; pole3=1

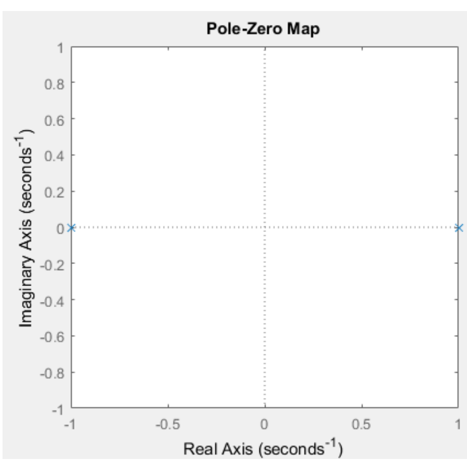
The pole lies in the right plane, and the system is unstable.



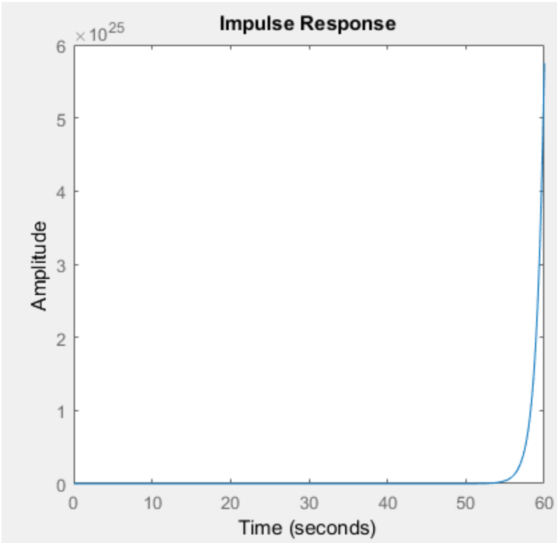


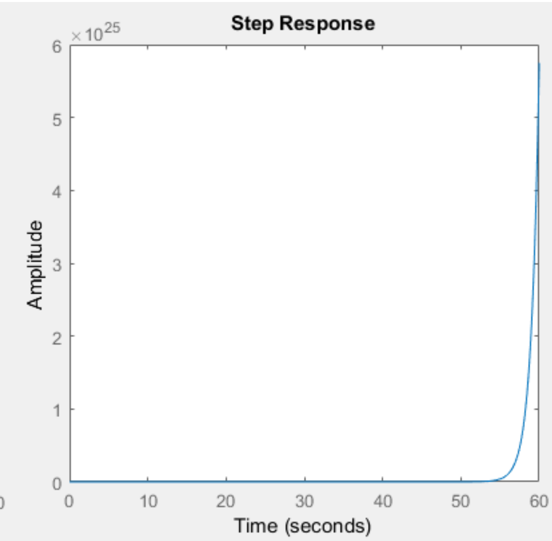


(4) $H_4(s) = \frac{1}{(s+1)(s-1)}$; pole4=1 and -1



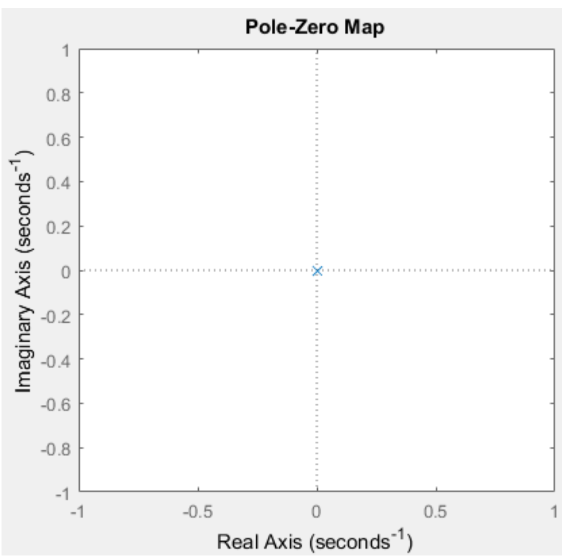
The system is unstable because one pole is greater than 0.

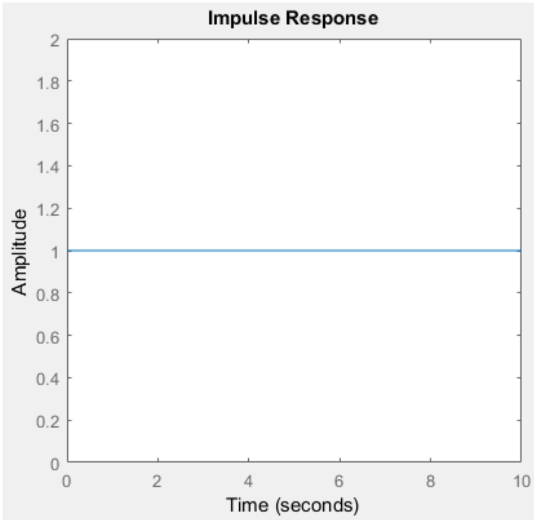


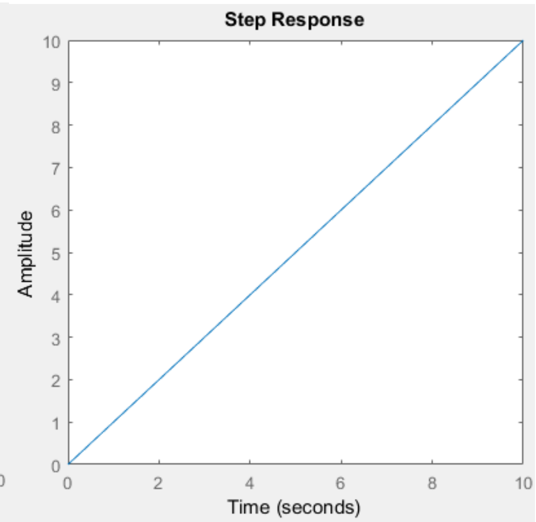


(5) $H_5(s) = \frac{1}{s}$; pole5=0

The system is marginally stable. The pole lies on the imaginary axis.

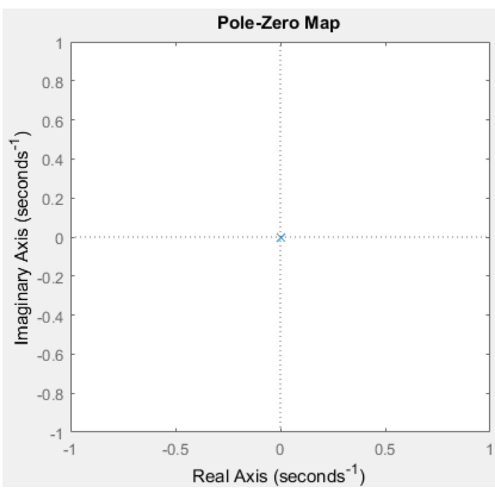


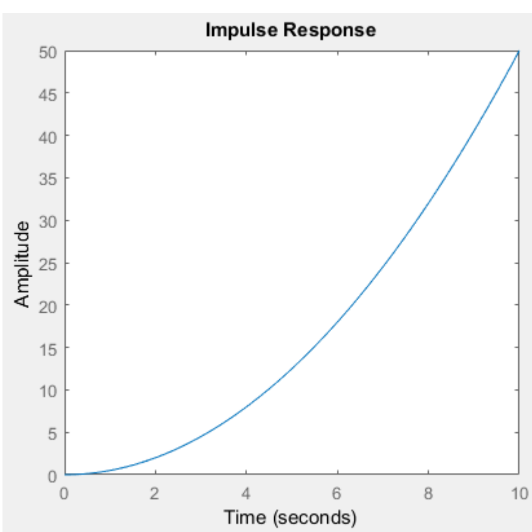


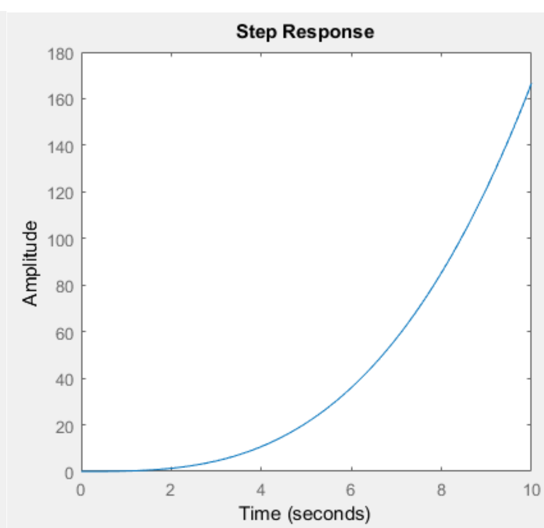


(6) $H_6(s) = \frac{1}{s^3}$; there are three poles, pole6=0,0 and 0

All three poles lies on the imaginary axis, the system is unstable.

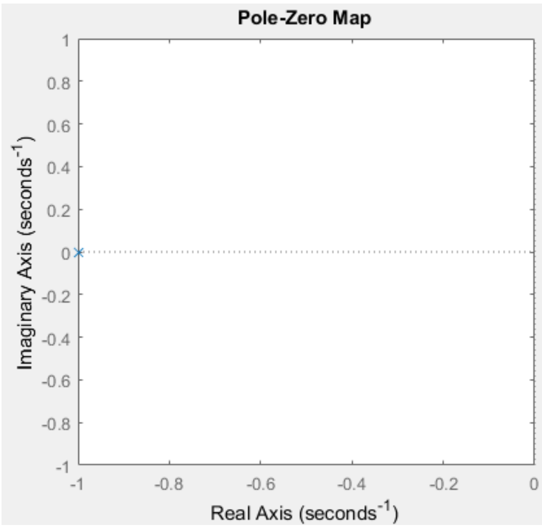


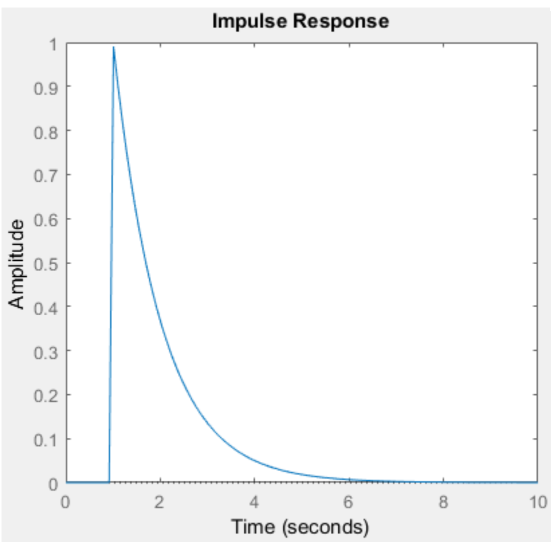


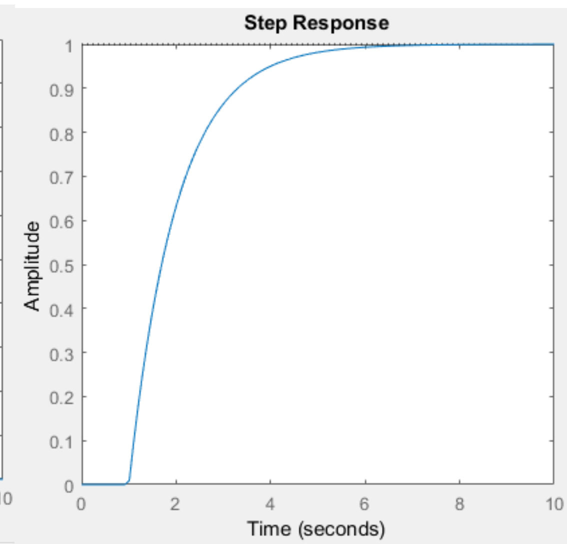


(7) $H_7(s) = \frac{e^{-s}}{s+1}$; pole7=-1

Pole locates in the left plane. The system is asymptotically stable.

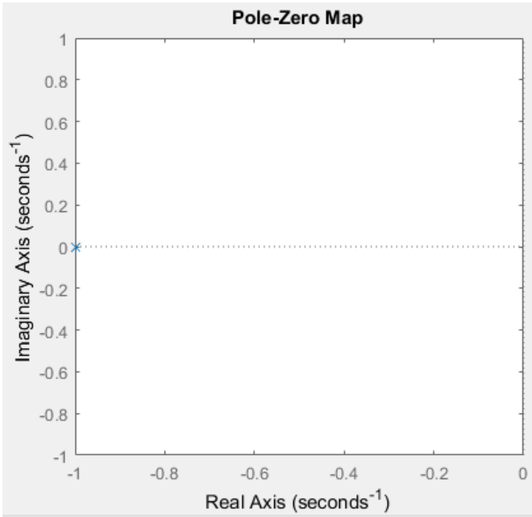


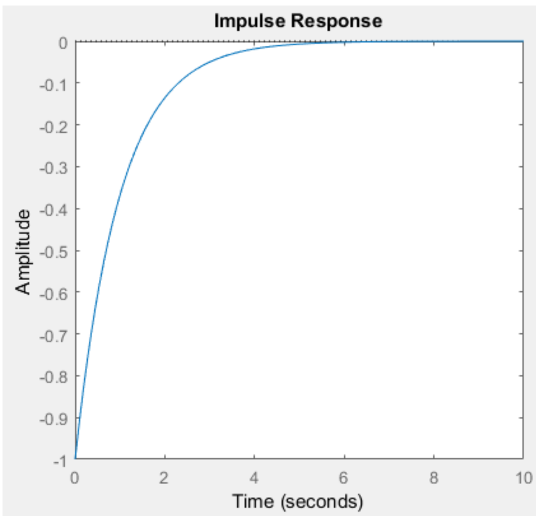


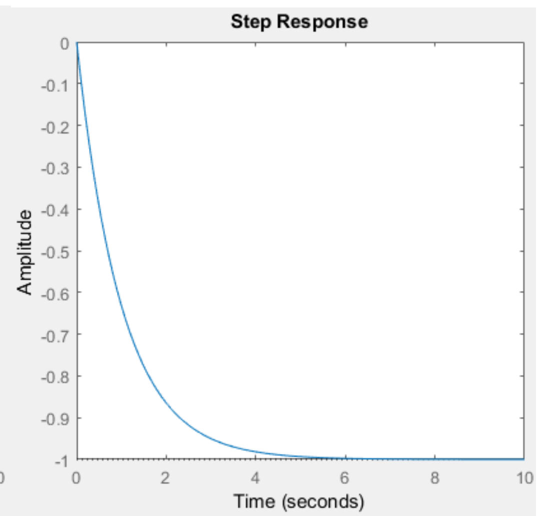


(8) $H_8(s) = -\frac{1}{s+1}$; pole8=-1

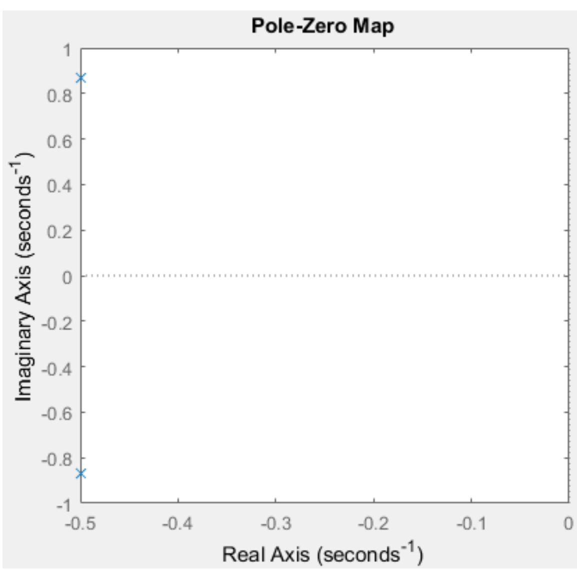
The system is asymptotically stable, because the pole lies in the left plane of the P-Z plot.



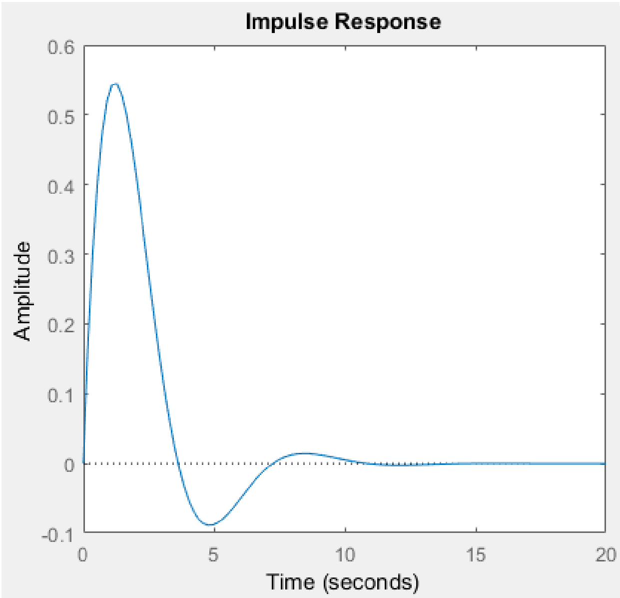


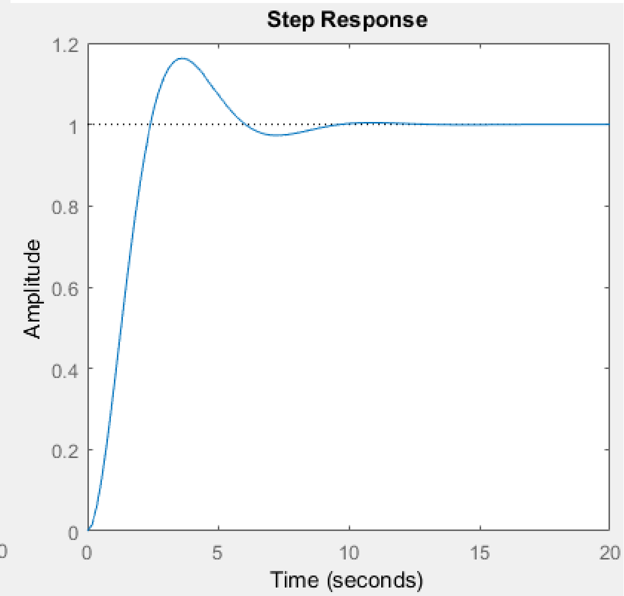


(9) $H_9(s) = \frac{1}{s^2+s+1}$; there are two poles, pole9=$-\frac{1}{2} \pm \frac{\sqrt{3}}{2}i$



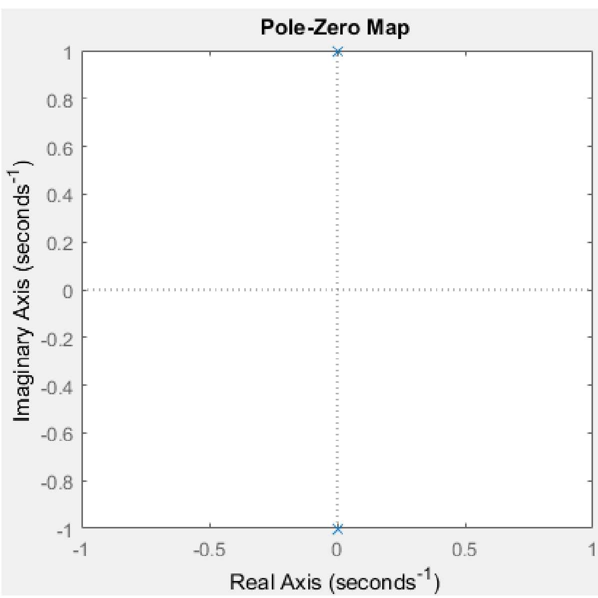
The system is stable, both poles locate in the left plane of the plot.

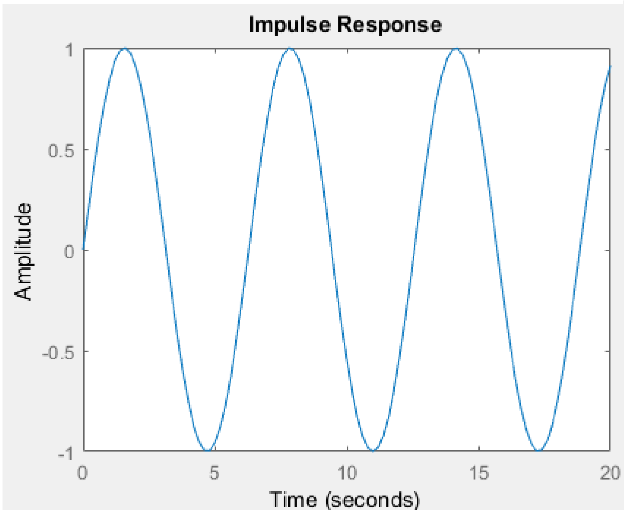


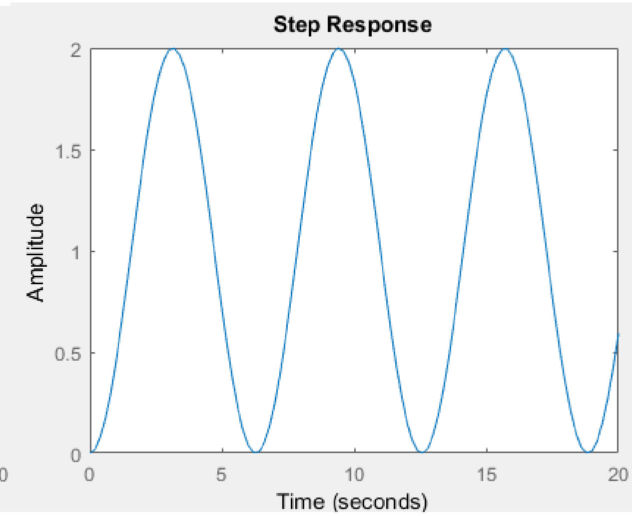


(10) $H_{10}(s) = \frac{1}{s^2+1}$; there are two poles, pole10=$\pm i$

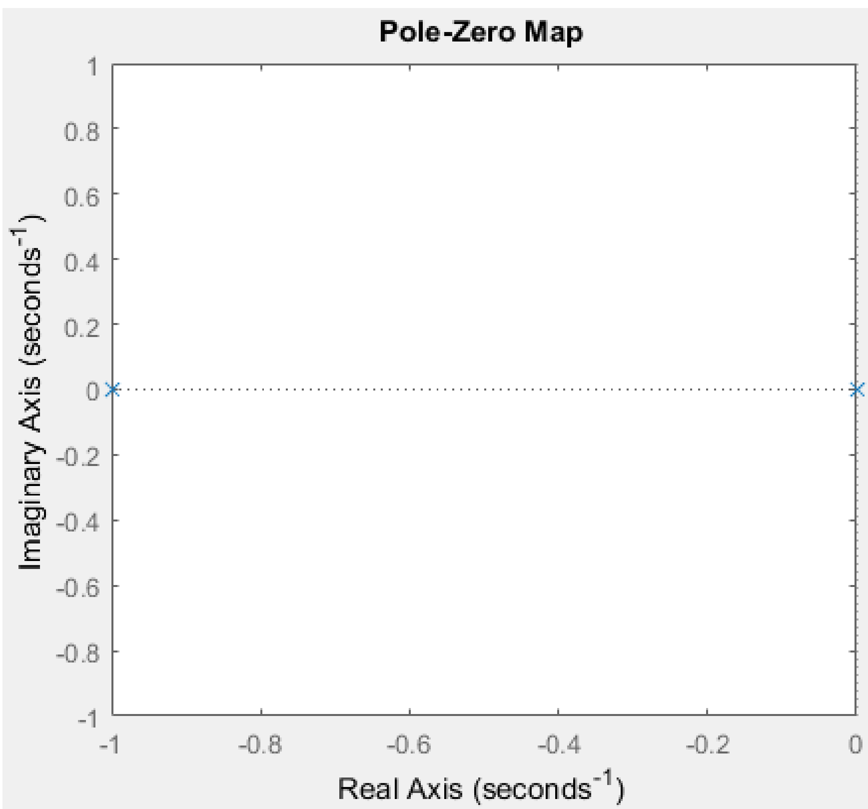
Both poles lie on the imaginary axis, and they are not equal to each other. Therefore, the system is marginally stable.



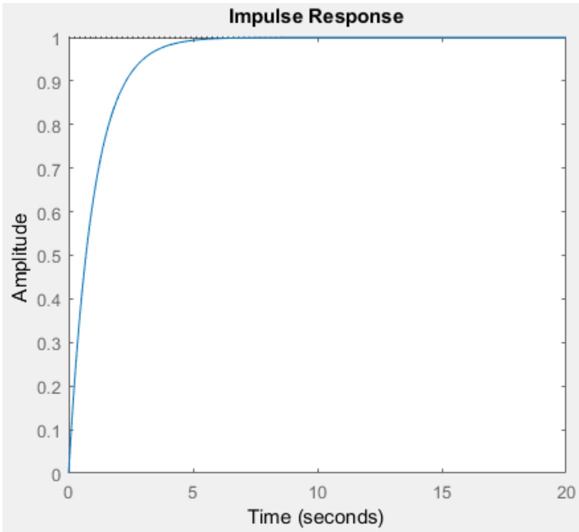


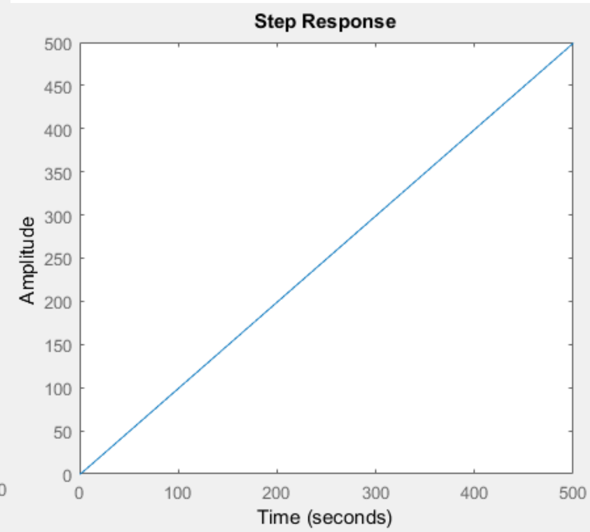


(11) $H_{11}(s) = \frac{1}{(s+1)s}$; two poles, pole11=0 and -1



The system is stable, because one pole lies in the left plane and another one lies on the imaginary axis.





3. Stability analysis of state space model

a. $A = \begin{bmatrix} 0 & 1 \\ 0 & -2 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$

The stability property of the state-space model is determined by the eigenvalues of A

det(sI-A)=det$\begin{bmatrix} s & -1 \\ 0 & s+2 \end{bmatrix}$=s(s+2)=0

s=0 and -2. One pole is 0, which lies on the imaginary axis. Another pole is -2, which is in the left half plane. Therefore, the system is marginally stable.

C=$[0 \quad 0]$; D=$[0]$,

Since C and D is zero, the output is always zero.

b. The transfer function $H(s) = \frac{y(s)}{u(s)} = C(sI - A)^{-1}B + D = C\frac{adj(sI-A)}{\det(sI-A)}B + D$

The pole of this transfer function is still the eigenvalues of the matrix sI-A. The eigenvalues are the same as the state space model.

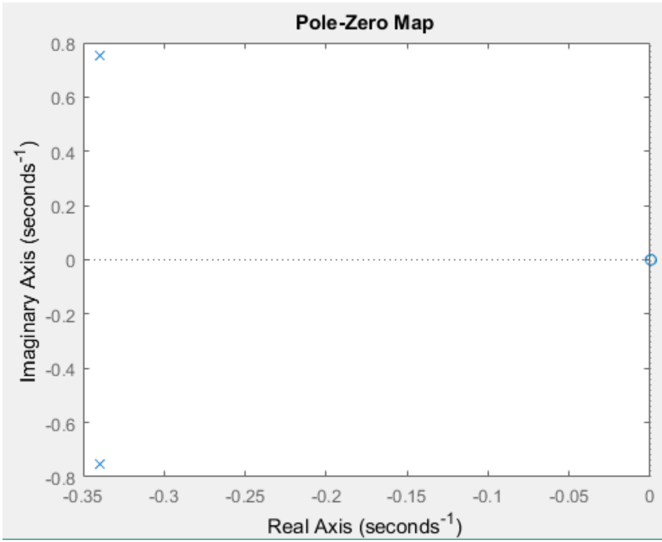
4. tuning and simulation of a feedback control system

The transfer function of the process $H_{pm}(s) = \frac{1}{s}$

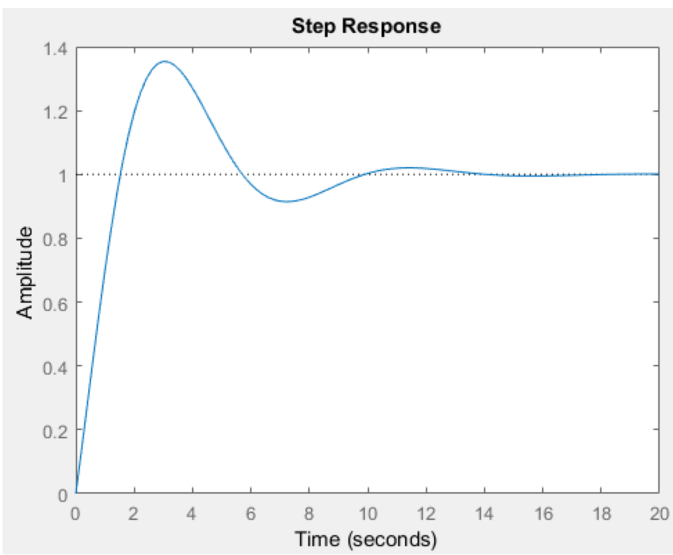
The PI controller transfer function is $H_c(s) = K_p + \frac{K_p}{T_i \cdot s}$;

The controller was tuned as Kp=0.68, Ti=1, based on Ziegler-Nichols' method. The close loop system is $\frac{s}{s^2+0.68s+0.68}$

poles' plot is as below



The setpoint response is shown as beow:



5. Discretization of a continuous model

$A = \begin{bmatrix} 0 & 1 \\ 0 & -2 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, C=$[0 \quad 0]$; D=$[0]$,

Using the 'c2d' function, we get the discrete state space model.

The character matrix Ad can be retrieved by 'ssdata': $Ad = \begin{bmatrix} 1 & 0.0906 \\ 0 & 0.8187 \end{bmatrix}, B = \begin{bmatrix} 0.0047 \\ 0.0906 \end{bmatrix}$, C=$[0 \quad 0]$; D=$[0]$,

The eigenvalues of the discretized system is 1 and 0.8187. This system is unstable because both poles locate in the right half plane, which is different with the continuous model.

6. Symbolic linearization with Matlab symbolic toolbox

Manually linearization:

$$\begin{bmatrix} \Delta \dot{h}_1 \\ \Delta \dot{h}_2 \end{bmatrix} = \left.\begin{bmatrix} \frac{\partial f_1}{\partial h_1} & \frac{\partial f_1}{\partial h_2} \\ \frac{\partial f_2}{\partial h_1} & \frac{\partial f_2}{\partial h_2} \end{bmatrix}\right|_0 \cdot \begin{bmatrix} \Delta h_1 \\ \Delta h_2 \end{bmatrix} + \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} \end{bmatrix} \cdot \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} -\frac{K_{v1}}{A_1} \cdot \sqrt{\frac{\rho g}{G}} \cdot \frac{1}{2\sqrt{h_1}} , & 0 \\ \frac{K_{v1}}{A_2} \sqrt{\frac{\rho g}{G}} \cdot \frac{1}{2\sqrt{h_1}} , & \frac{K_{v2} u_2}{A_2} \cdot \sqrt{\frac{\rho g}{G}} \cdot \sqrt{\frac{1}{2h_2}} \end{bmatrix}}_{A} + \underbrace{\begin{bmatrix} \frac{K_p}{A_1} , & 0 \\ 0 , & -\frac{K_{v2}}{A_2} \cdot \sqrt{\frac{\rho g h_2}{G}} \end{bmatrix}}_{B} \cdot \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \end{bmatrix}$$

$$\begin{bmatrix} \Delta y_1 \\ \Delta y_2 \end{bmatrix} = \begin{bmatrix} \frac{\partial g_1}{\partial h_1} & \frac{\partial g_1}{\partial h_2} \\ \frac{\partial g_2}{\partial h_1} & \frac{\partial g_2}{\partial h_2} \end{bmatrix} \cdot \begin{bmatrix} \Delta h_1 \\ \Delta h_2 \end{bmatrix} + \begin{bmatrix} \frac{\partial g_1}{\partial u_1} & \frac{\partial g_1}{\partial u_2} \\ \frac{\partial g_2}{\partial u_1} & \frac{\partial g_2}{\partial u_2} \end{bmatrix} \cdot \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{C} \cdot \begin{bmatrix} \Delta h_1 \\ \Delta h_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}}_{D} \cdot \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \end{bmatrix}$$

Matlab output

```
[ -(Kv1*g*rho)/(2*A1*G*((g*h1*rho)/G)^(1/2)),                                    0]
[  (Kv1*g*rho)/(2*A2*G*((g*h1*rho)/G)^(1/2)), -(Kv2*g*rho*u2)/(2*A2*G*((g*h2*rho)/G)^(1/2))]

[ Kp/A1,                          0]
[     0, -(Kv2*((g*h2*rho)/G)^(1/2))/A2]

[ 1, 0]
[ 0, 1]

[ 0, 0]
[ 0, 0]
```

Optimization

See Matlab script