1) $\dot{x} = Ax + Bu$

$y = Cx + Du$

$$y(t) = Ce^{At}x(0) + \int_0^t Ce^{A(t-\tau)}Bu(\tau)d\tau + Du(t)$$

a) $u(t) = 0$

$y(t) = Ce^{At}x(0)$ for an initial condition $x(0)$

The initial conditions are $x(0)$ and $\hat{x}(0)$, which correspond to outputs $y_1(t)$ & $y_2(t)$.

$y_1(t) = Ce^{At}x(0)$

$y_2(t) = Ce^{At}\hat{x}(0)$

When the initial state is $\alpha x(0) + \beta\hat{x}(0)$:

$$y(t) = Ce^{At}(\alpha x(0) + \beta\hat{x}(0))$$
$$= \alpha Ce^{At}x(0) + \beta Ce^{At}\hat{x}(0)$$
$$= \alpha y_1(t) + \beta y_2(t)$$

$$\therefore \begin{cases} x(0) \longrightarrow y_1(t) \\ \hat{x}(0) \longrightarrow y_2(t) \end{cases} \Longrightarrow \alpha x(0) + \beta\hat{x}(0) \longrightarrow \alpha y_1(t) + \beta y_2(t)$$

$\therefore$ The output $y(t)$ is linear in the initial state $x(0)$

b) $x(0)=0$

$y(t)=\int_0^t Ce^{A(t-\tau)}Bu(\tau)d\tau + Du(t)$ for an input $u(t)$

Let the inputs $u_1(t)$ and $u_2(t)$ correspond to outputs $y_1(t)$ & $y_2(t)$ respectively

$y_1(t)=\int_0^t Ce^{A(t-\tau)}Bu_1(\tau)d\tau + Du_1(t)$

$y_2(t)=\int_0^t Ce^{A(t-\tau)}Bu_2(\tau)d\tau + Du_2(t)$

When the input is $u(t)=\alpha u_1(t)+\beta u_2(t)$:

$$y(t)=\int_0^t Ce^{A(t-\tau)}B(\alpha u_1(\tau)+\beta u_2(\tau))\,d\tau + D(\alpha u_1(t)+\beta u_2(t))$$

$$=\alpha\left[\int_0^t Ce^{A(t-\tau)}Bu_1(\tau)d\tau + Du_1(t)\right]+\beta\left[\int_0^t Ce^{A(t-\tau)}Bu_2(\tau)d\tau + Du_2(t)\right]$$

$$=\alpha y_1(t)+\beta y_2(t)$$

$$\therefore \begin{cases} u_1(t)\to y_1(t) \\ u_2(t)\to y_2(t)\end{cases} \Longrightarrow \alpha u_1(t)+\beta u_2(t)\longrightarrow \alpha y_1(t)+\beta y_2(t)$$

$\therefore$ The output $y(t)$ is linear in the input $u(t)$ when $x(0)=0$.

2) a) $u(t)=\delta(t)$

$x(0)=0$

$$h(t)=\int_0^t Ce^{A(t-\tau)}B\delta(\tau)\,d\tau + D\delta(t)$$

$$=Ce^{At}B+D\delta(t)$$

$\therefore$ The impulse response of a system with $u(t)=\delta(t)$ and $x(0)=0$ is $h(t)=Ce^{At}B+D\delta(t)$

4) a)

$$A = \begin{bmatrix} -a_0 - a_1 & a_1 \\ a_2 & -a_2 \end{bmatrix} = \begin{bmatrix} -3 & 2 \\ 1 & -1 \end{bmatrix}$$

$$B = \begin{bmatrix} b_0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}$$

$$K = \begin{bmatrix} k_1 & k_2 \end{bmatrix}$$

$$A - BK = \begin{bmatrix} -3 & 2 \\ 1 & -1 \end{bmatrix} - \underbrace{\begin{bmatrix} 0.5 \\ 0 \end{bmatrix}}_{2\times 1} \underbrace{\begin{bmatrix} k_1 & k_2 \end{bmatrix}}_{1\times 2}$$

$$= \begin{bmatrix} -3 & 2 \\ 1 & -1 \end{bmatrix} - \begin{bmatrix} 0.5k_1 & 0.5k_2 \\ 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} -3 - 0.5k_1 & 2 - 0.5k_2 \\ 1 & -1 \end{bmatrix}$$

$$\boxed{A - BK = \begin{bmatrix} -3 - 0.5k_1 & 2 - 0.5k_2 \\ 1 & -1 \end{bmatrix}}$$

$$k_r B = k_r \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.5k_r \\ 0 \end{bmatrix}$$

$$\boxed{k_r B = \begin{bmatrix} 0.5k_r \\ 0 \end{bmatrix}}$$

b) $|\lambda I - (A-BK)| = \begin{vmatrix} \lambda+3+0.5k_1 & -2+0.5k_2 \\ -1 & \lambda+1 \end{vmatrix} = (\lambda+3+0.5k_1)(\lambda+1) - (2-0.5k_2)$

$$= \lambda^2 + 3\lambda + 0.5k_1\lambda + \lambda + 3 + 0.5k_1 - 2 + 0.5k_2$$

$$= \lambda^2 + (4+0.5k_1)\lambda + (1+0.5k_1+0.5k_2)$$

$$2\zeta_0\omega_0 = 4+0.5k_1 \rightarrow 0.5k_1 = 2\zeta_0\omega_0 - 4 \Rightarrow \boxed{k_1 = 4\zeta_0\omega_0 - 8}$$

$$\omega_0^2 = 1+0.5k_1+0.5k_2 = 1+2\zeta_0\omega_0-4+0.5k_2 = 2\zeta_0\omega_0 - 3 + 0.5k_2$$

$$0.5k_2 = \omega_0^2 - 2\zeta_0\omega_0 + 3$$

$$\boxed{k_2 = 2\omega_0^2 - 4\zeta_0\omega_0 + 6}$$

For $\dot{x} = Ax+Bu,\ y = Cx+Du$:

$$G(s) = C(sI-A)^{-1}B + D$$

For $\dot{x} = (A-BK)x + k_rBr\ ,\ y = Cx + 0$:

$$G(s) = C(sI-(A-BK))^{-1}k_rB$$

$$y_{ss} = G(0)r = r$$

$$G(0) = 1 = C(-(A-BK)^{-1})Bk_r$$

$$= \frac{-1}{1+0.5k_1+0.5k_2} \underbrace{\begin{bmatrix} 0 & 1 \end{bmatrix}}_{1\times2} \underbrace{\begin{bmatrix} -1 & -2+0.5k_2 \\ -1 & -3-0.5k_1 \end{bmatrix}}_{2\times2} \begin{bmatrix} 0.5k_r \\ 0 \end{bmatrix}$$

$$= \frac{-1}{1+0.5k_1+0.5k_2} \underbrace{\begin{bmatrix} -1 & -3-0.5k_1 \end{bmatrix}}_{1\times2} \underbrace{\begin{bmatrix} 0.5k_r \\ 0 \end{bmatrix}}_{2\times1}$$

$$= \frac{-1}{1+0.5k_1+0.5k_2}(-0.5k_r)$$

$$= \frac{0.5k_r}{1+0.5k_1+0.5k_2} = \frac{0.5k_r}{\cancel{1+2\zeta_0\omega_0-4} + \omega_0^2 - \cancel{2\zeta_0\omega_0} + \cancel{3}}$$

$$= \frac{0.5k_r}{\omega_0^2}$$

$$\frac{0.5k_r}{\omega_0^2} = 1 \Rightarrow \boxed{k_r = 2\omega_0^2}$$

Questions 2b, 2c, 3, 4c, and 4d are answered in the section below.

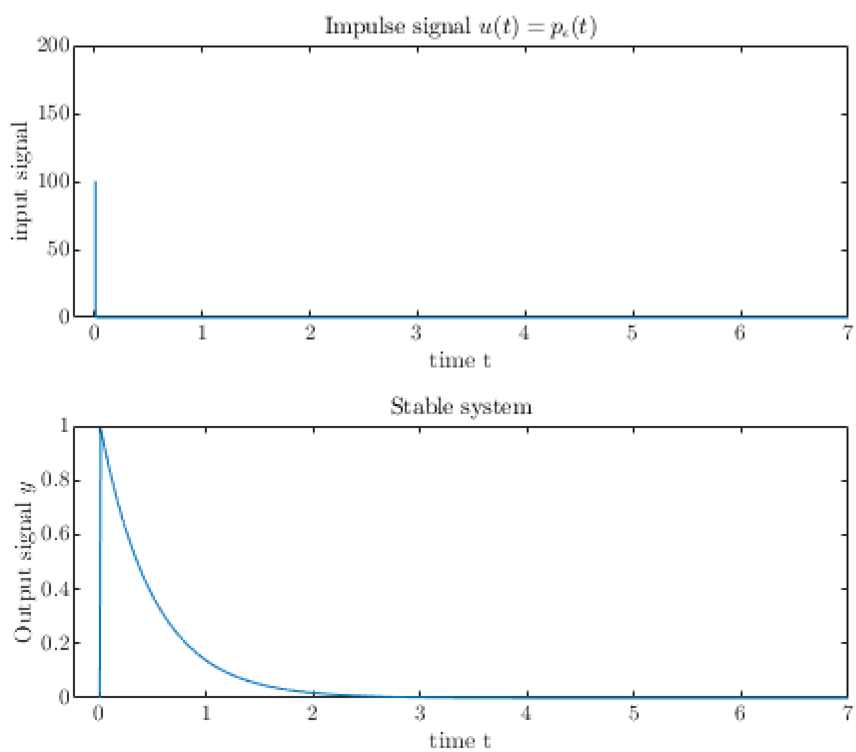
2b)

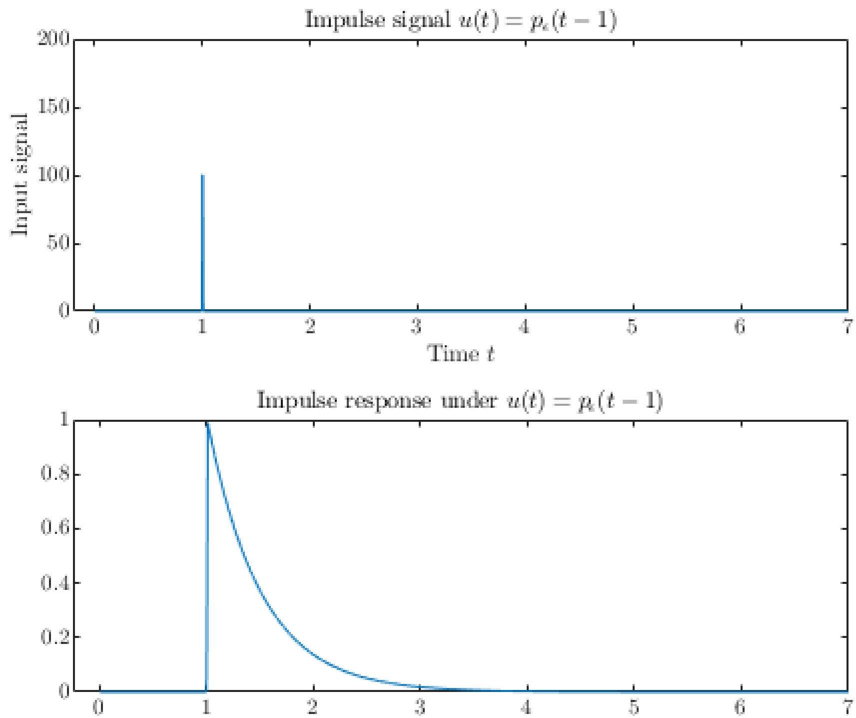
```
-------------------------------------------------------------------------

ECE 171A: Linear Control System Theory
Impulse response - time plot

-------------------------------------------------------------------------

clc;
close all; clear;

% ---------------------------------------------------
% Case 1: u(t) = p_e(t)
% ---------------------------------------------------
% ---------------------------------------------------
%    state-space system
% ---------------------------------------------------
A    = [-1 1; 0, -2];
B    = [0; 1];
C    = [0, 1];
D    = 0;
sys1 = ss(A, B, C, D);   % A stable system

% ---------------------------------------------------
% define the impulse input signal
%   u and simulate the impuse response
% ---------------------------------------------------

% define the impulse signal (which is an approximation)
T   = 7;
t   = 0:.0001:T;
eps = 0.01;
u1  = zeros(size(t));
i   = 1;
while t(i) < eps
    u1(i)= 1/eps;
    i    = i+1;
end

% ---------------------------------------------------
% simulate impulse response
% ---------------------------------------------------
x0   = [0;0];              % zero initial condition
y1   = lsim(sys1,u1,t,x0); % stable system

% ---------------------------------------------------
% plot the input signal and output signal
% ---------------------------------------------------
figure; FontSize = 8;
subplot(2,1,1); plot(t,u1);    % input signal
axis([-0.2,T 0 1/eps*2]);
xlabel('time t','Interpreter','latex');
```

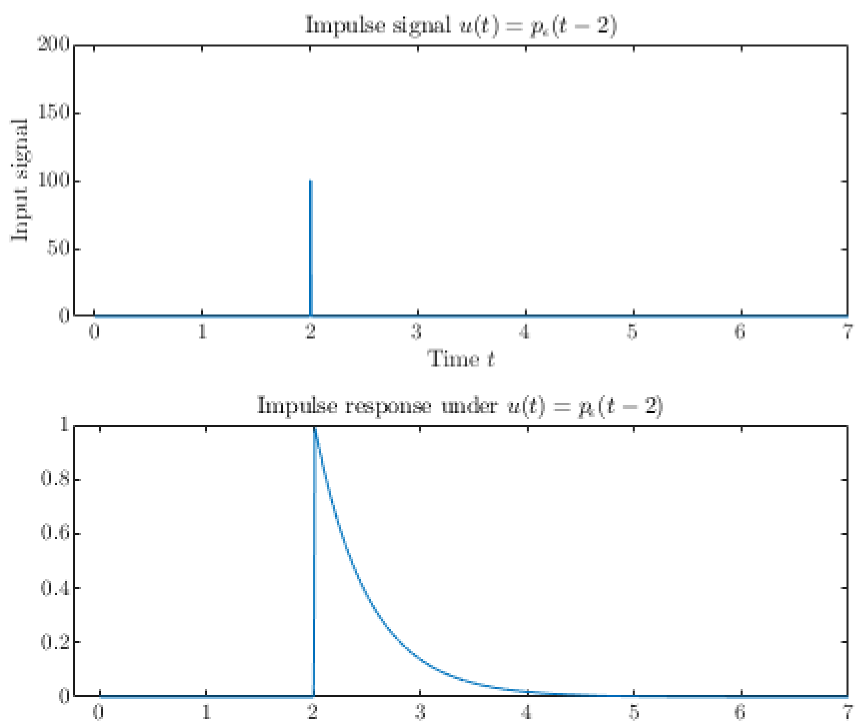
```
ylabel('input signal','Interpreter','latex');
title('Impulse signal $u(t) = p_{\epsilon}(t)$','Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','fontsize',FontSize);

subplot(2,1,2);           % output signal
plot(t,y1);
axis([-0.25, T, 0, 1]);
title('Stable system','Interpreter','latex');
xlabel('time t','Interpreter','latex');
ylabel('Output signal $y$','Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','fontsize',FontSize);


set(gcf,'Position',[100 100 500 400])

% --------------------------------------------------
% Case 2: u(t) = p_e(t-1)
% --------------------------------------------------
tau  = 1;
i    = 1;
u3   = zeros(size(u1));
u3(tau/0.0001:end) = u1(1:end-tau/0.0001+1); % a shift of the input

y3   = lsim(sys1,u3,t,x0);

figure;
subplot(2,1,1); plot(t,u3); % input signal
axis([-0.2 T 0 1/eps*2]);
xlabel('Time $t$','Interpreter','latex');
ylabel('Input signal','Interpreter','latex');
title('Impulse signal $u(t) = p_{\epsilon}(t-1)$','Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','fontsize',FontSize);

subplot(2,1,2); % output signal
plot(t,y3);
axis([-0.25 T 0 1]);
title('Impulse response under $u(t) = p_{\epsilon}
(t-1)$','Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','fontsize',FontSize);

set(gcf,'Position',[100 100 500 400])



% --------------------------------------------------
% Case 3: u(t) p_e(t-2)
% --------------------------------------------------

tau  = 2;
i    = 1;
u4   = zeros(size(u1));
u4(tau/0.0001:end) = u1(1:end-tau/0.0001+1); % a shift of the input

y4   = lsim(sys1,u4,t,x0);
```

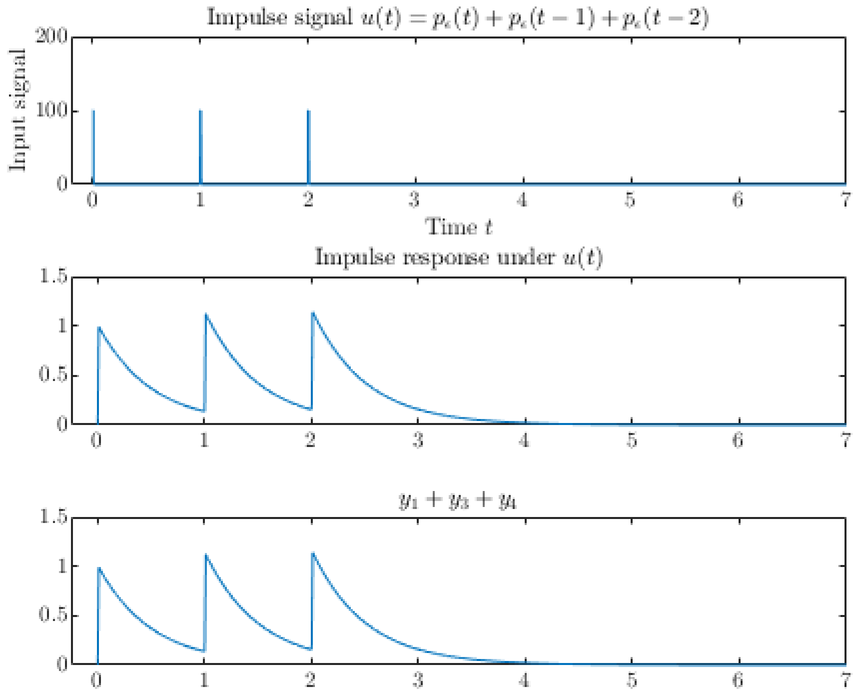
```
figure;
subplot(2,1,1);  plot(t,u4); % input signal
axis([-0.2 T 0 1/eps*2]);
xlabel('Time $t$','Interpreter','latex');
ylabel('Input signal','Interpreter','latex');
title('Impulse signal $u(t) = p_{\epsilon}(t-2)$','Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','fontsize',FontSize);

subplot(2,1,2); % output signal
plot(t,y4);
axis([-0.25 T 0 1]);
title('Impulse response under $u(t) = p_{\epsilon}
(t-2)$','Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','fontsize',FontSize);

set(gcf,'Position',[100 100 500 400])

% -------------------------------------------------
% Case 4: u(t) = p_e(t) + p_e(t-1) + p_e(t-2)
% -------------------------------------------------

% recall that u3 and u4 are shiftted signals of u1
u5 = u1 + u3 + u4;
y5 = lsim(sys1,u5,t,x0);

figure;
subplot(3,1,1);  plot(t,u5);   % input signal
axis([-0.2,T 0 1/eps*2]);
xlabel('Time $t$','Interpreter','latex');
ylabel('Input signal','Interpreter','latex');
title('Impulse signal $u(t) = p_{\epsilon}(t) + p_{\epsilon}(t-1) +
p_{\epsilon}(t-2)$','Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','fontsize',FontSize);


subplot(3,1,2); plot(t,y5);% output signal
axis([-0.25,T 0 1.5]);
title('Impulse response under $u(t)$','Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','fontsize',FontSize);

% let's compute y1+y2+y3 and compare it with y4
subplot(3,1,3);
plot(t,y1+y3+y4);
axis([-0.25,T 0 1.5]);
title('$y_1+y_3+y_4$','Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','fontsize',FontSize);

set(gcf,'Position',[100 100 500 400])

%--------------------------------------------------------------------------
% Observations:
% In case 1, the input surges at time t=0 to 100 and almost immediately
% falls back down to 0. The output exhibits the same behavior at time t=0
```

```
% and surges up to the value y=1 and then exponentially decays to 0
% indicating that the system is stable. In case 2, the input has similar
% behavior to that in case 1 except it is delayed by 1 unit of time, so the
% input's surge from 0 to 100 happens at t=1 instead of t=0. This delay can
% also be seen in the output, which has similar behavior to the output in
% case 1. There is a surge to y=1 at time t=1 and then an exponential decay
% back down to. Case 3 has the same behavior except with a delay of 2 units
% of time. The surges and decreases/decays of the input and output occur at
% and after t=2. All 3 systems are stable. Case 4's input is the sum of the
% inputs in the previous cases, which can be seen on the input graph as 3
% signals (pulses) at times t=0,1, and 2. What's more interesting however,
% is that the output is also the sum of the outputs of the 3 previous
% cases. There is a pulse at time t=0 to y=1 and then an exponential decay
% until another pulse at time t=1 to y=1 and then an exponential decay and
% finally, there is a pulse at time t=2 and a continuous(uninterrupted)
% exponential decay down to 0. This indicates that the system has linearity
% as the sum of individual inputs results in an output that is the sum of
% the outputs of the individual inputs.
%----------------------------------------------------------------------------
```




Impulse signal $u(t) = p_\epsilon(t-1)$
Input signal
Time t
Impulse response under $u(t) = p_\epsilon(t-1)$
0
1
2
3
4
5
6
7
50
100
150
200
0.2
0.4
0.6
0.8

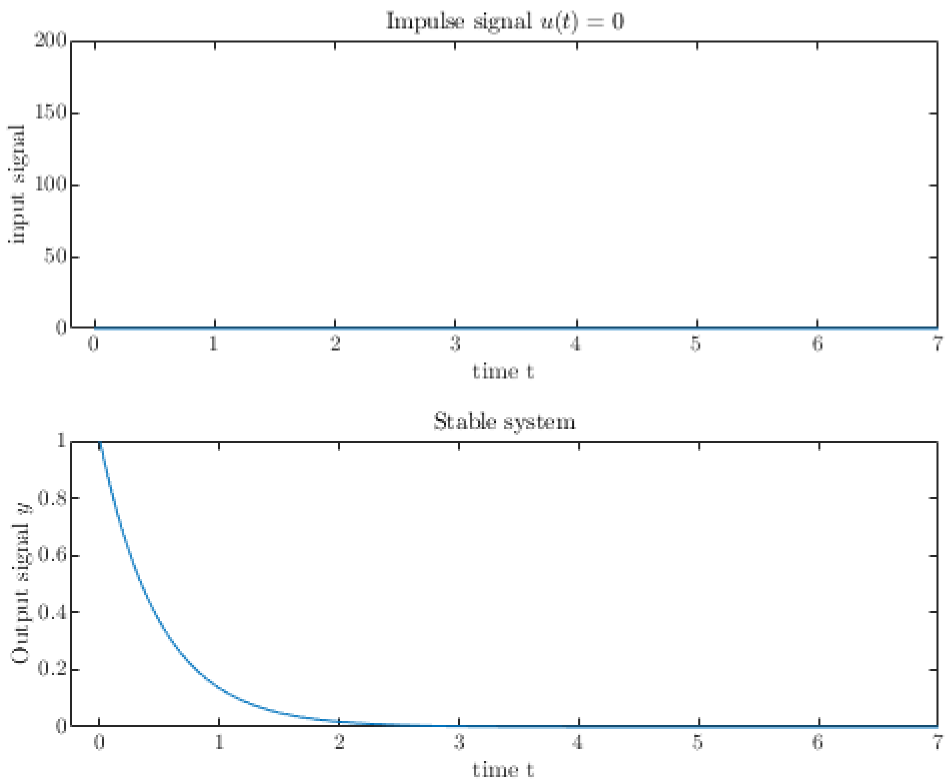

Impulse signal $u(t) = p_\epsilon(t-2)$
Input signal
Time t
Impulse response under $u(t) = p_\epsilon(t-2)$
0
1
2
3
4
5
6
7
50
100
150
200
0.2
0.4
0.6
0.8



2c)

```
u = zeros(size(t));
x0 = [0; 1];
y = lsim(sys1, u, t, x0);

figure; FontSize = 8;
subplot(2,1,1); plot(t,u);     % input signal
axis([-0.2,T 0 1/eps*2]);
xlabel('time t','Interpreter','latex');
ylabel('input signal','Interpreter','latex');
title('Impulse signal $u(t)=0$','Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','fontsize',FontSize);

subplot(2,1,2);           % output signal
plot(t,y);
axis([-0.25, T, 0, 1]);
title('Stable system','Interpreter','latex');
xlabel('time t','Interpreter','latex');
ylabel('Output signal $y$','Interpreter','latex');
set(gca,'TickLabelInterpreter','latex','fontsize',FontSize);

%---------------------------------------------------------------------------
% Comparison:
% There is no pulse in the input signal unlike case 1. It is consistently
% 0. The output signal starts with the value 1 and exponentially decays to
% 0, and there is no pulse in the output signal either. However, just like
```

```
% case 1, the system is stable, so it has the exponential decay in common.
%--------------------------------------------------------------------------
```



Published with MATLAB® R2024b

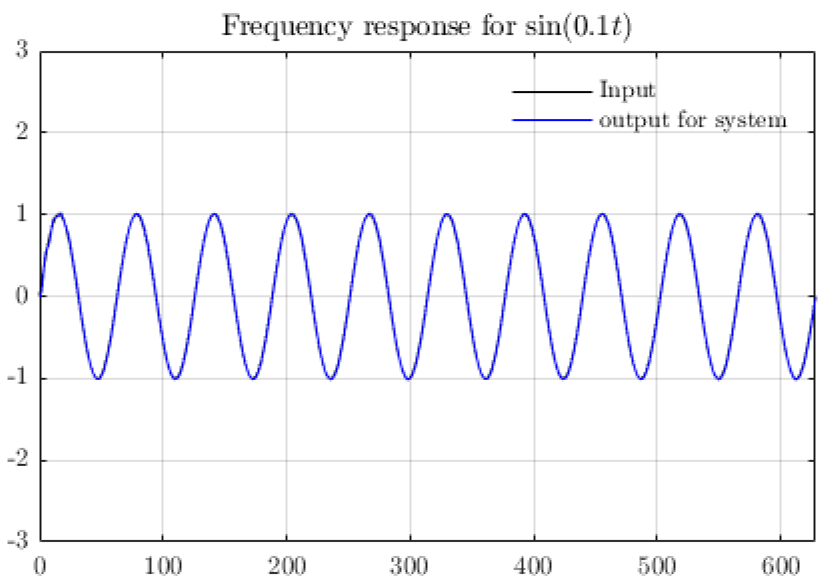
3a)

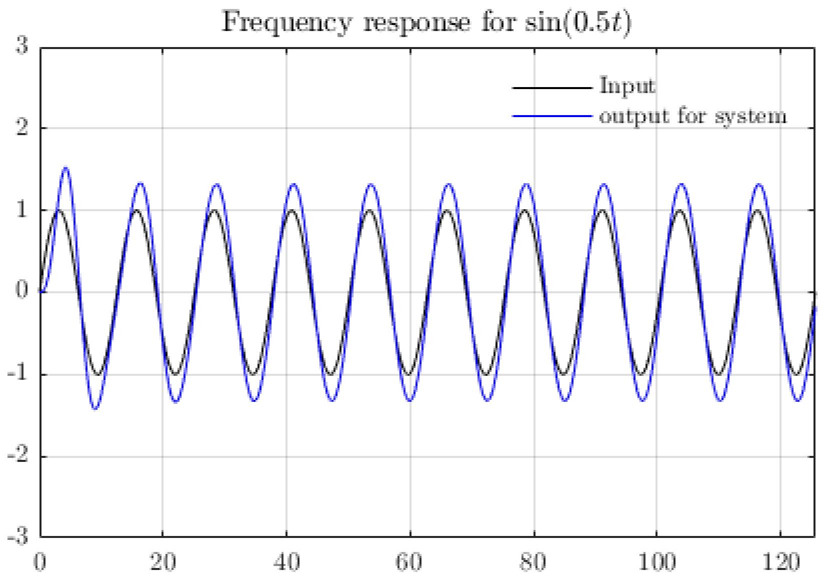
```
% ------------------------------------------------------------------------
%
%  ECE 171A: Linear Control System Theory
%  Frequency response - time plot
%
% ------------------------------------------------------------------------


close all; clear;

% --------------------------------------------------
%    state-space system
% --------------------------------------------------
k = 1;
m = 1;
c = 0.2;
A1   = [0 1; -k/m, -c/m];
B1   = [0; 1/m];
C1   = [1, 0];
D1   = 0;
sys1 = ss(A1, B1, C1, D1);  % stable system 1
x0 = [0; 0];

% --------------------------------------------------
% Case 1: input signal (sin(0.1t))
% --------------------------------------------------

omega = 0.1;
T     = 2*pi/omega*10;
t     = 0:.0001:T;

u1    = sin(omega*t);
y1    = lsim(sys1,u1,t,x0);

figure;  FontSize = 10;
plot(t,u1,'black',t,y1,'b');
h = legend('Input','output for system','Interpreter','latex');
set(h,'box','off')
title('Frequency response for $\sin(0.1t)
$','Interpreter','latex','fontsize',FontSize);
axis([0,T, -3, 3]);
grid on;
set(gca,'TickLabelInterpreter','latex','fontsize',FontSize);
set(gcf,'Position',[100 100 500 300])
print(gcf,'L11_fre2','-painters','-depsc','-r300')
[pks_y, locs_y] = findpeaks(y1, t);
[pks_u, locs_u] = findpeaks(u1, t);
Ay = mean(pks_y);
Au = mean(pks_u);
M1 = Ay / Au;
```

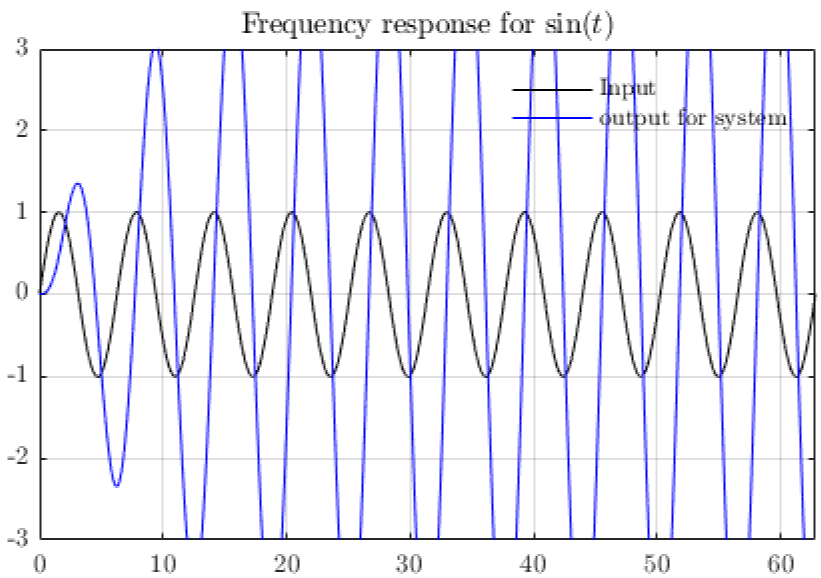
```
n = min(length(locs_y), length(locs_u));
T_vector = locs_y(1:n) - locs_u(1:n);
deltaT = mean(T_vector);
period = 2*pi*omega;
phase1 = (-2*pi*deltaT) / (T/10);

% ------------------------------------------------
% Case 2: input signal (sin(0.5t))
% ------------------------------------------------

omega = 0.5;
T     = 2*pi/omega*10;
t     = 0:.0001:T;

u1    = sin(omega*t);
y1    = lsim(sys1,u1,t, x0);

figure;  FontSize = 10;
plot(t,u1,'black',t,y1,'b');
h = legend('Input','output for system','Interpreter','latex');
set(h,'box','off')
title('Frequency response for $\sin(0.5t)
$','Interpreter','latex','fontsize',FontSize);
axis([0,T, -3, 3]);
grid on;
set(gca,'TickLabelInterpreter','latex','fontsize',FontSize);
set(gcf,'Position',[100 100 500 300])
print(gcf,'L11_fre2','-painters','-depsc','-r300')
[pks_y, locs_y] = findpeaks(y1, t);
[pks_u, locs_u] = findpeaks(u1, t);
Ay = mean(pks_y);
Au = mean(pks_u);
M2 = Ay / Au;
n = min(length(locs_y), length(locs_u));
T_vector = locs_y(1:n) - locs_u(1:n);
deltaT = mean(T_vector);
period = 2*pi*omega;
phase2 = (-2*pi*deltaT) / (T/10);


% ------------------------------------------------
% Case 3: input signal (sin(t))
% ------------------------------------------------

omega = 1;
T     = 2*pi/omega*10;
t     = 0:.0001:T;

u1    = sin(omega*t);
y1    = lsim(sys1,u1,t, x0);

figure;  FontSize = 10;
plot(t,u1,'black',t,y1,'b');
h = legend('Input','output for system','Interpreter','latex');
```

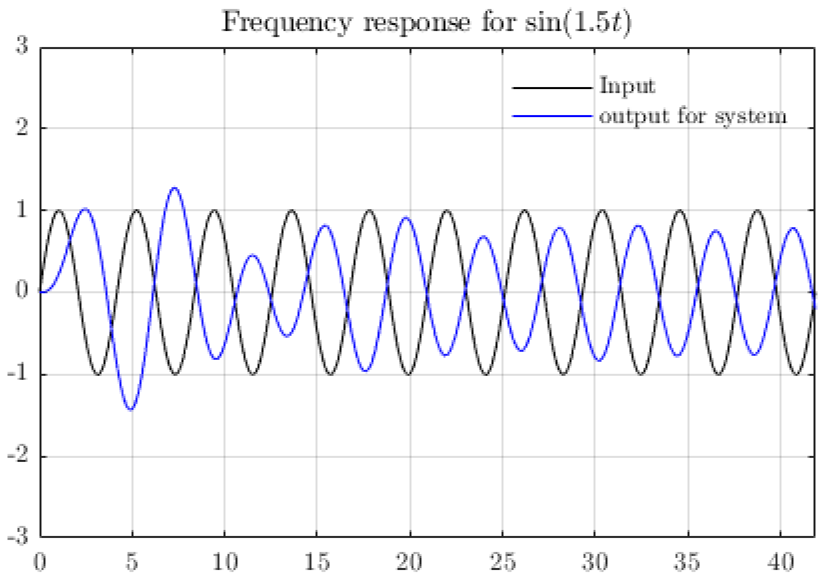
```
set(h,'box','off')
title('Frequency response for $\sin(t)
$','Interpreter','latex','fontsize',FontSize);
axis([0,T, -3, 3]);
grid on;
set(gca,'TickLabelInterpreter','latex','fontsize',FontSize);
set(gcf,'Position',[100 100 500 300])
print(gcf,'L11_fre2','-painters','-depsc','-r300')
[pks_y, locs_y] = findpeaks(y1, t);
[pks_u, locs_u] = findpeaks(u1, t);
Ay = mean(pks_y);
Au = mean(pks_u);
M3 = Ay / Au;
n = min(length(locs_y), length(locs_u));
T_vector = locs_y(1:n) - locs_u(1:n);
deltaT = mean(T_vector);
period = 2*pi*omega;
phase3 = (-2*pi*deltaT) / (T/10);

% --------------------------------------------------
% Case 4: input signal (sin(1.5t))
% --------------------------------------------------

omega = 1.5;
T     = 2*pi/omega*10;
t     = 0:.0001:T;

u1    = sin(omega*t);
y1    = lsim(sys1,u1,t, x0);

figure;  FontSize = 10;
plot(t,u1,'black',t,y1,'b');
h = legend('Input','output for system','Interpreter','latex');
set(h,'box','off')
title('Frequency response for $\sin(1.5t)
$','Interpreter','latex','fontsize',FontSize);
axis([0,T, -3, 3]);
grid on;
set(gca,'TickLabelInterpreter','latex','fontsize',FontSize);
set(gcf,'Position',[100 100 500 300])
print(gcf,'L11_fre2','-painters','-depsc','-r300')
[pks_y, locs_y] = findpeaks(y1, t);
[pks_u, locs_u] = findpeaks(u1, t);
Ay = mean(pks_y);
Au = mean(pks_u);
M4 = Ay / Au;
n = min(length(locs_y), length(locs_u));
T_vector = locs_y(1:n) - locs_u(1:n);
deltaT = mean(T_vector);
period = 2*pi*omega;
phase4 = (-2*pi*deltaT) / (T/10);

% --------------------------------------------------
% Case 5: input signal (sin(2t))
```

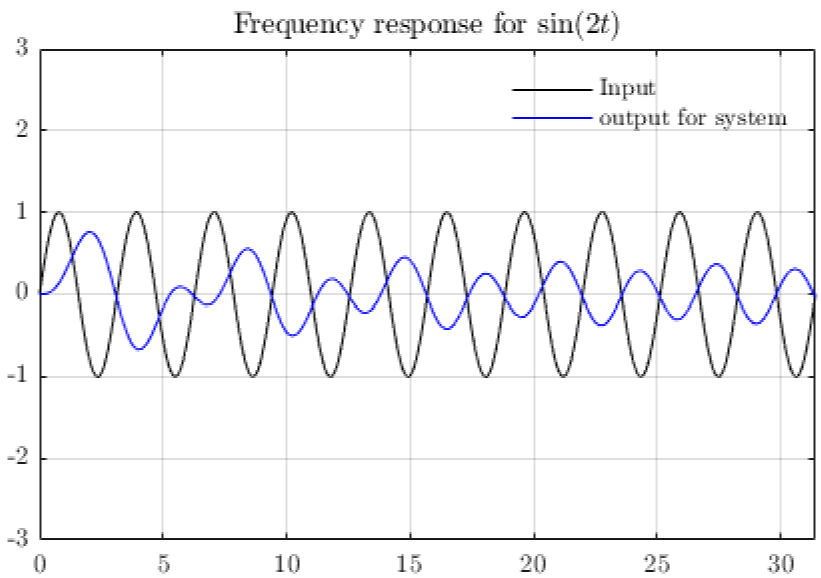
```
% ---------------------------------------------------

omega = 2;
T     = 2*pi/omega*10;
t     = 0:.0001:T;

u1    = sin(omega*t);
y1    = lsim(sys1,u1,t, x0);

figure;  FontSize = 10;
plot(t,u1,'black',t,y1,'b');
h = legend('Input','output for system','Interpreter','latex');
set(h,'box','off')
title('Frequency response for $\sin(2t)
$','Interpreter','latex','fontsize',FontSize);
axis([0,T, -3, 3]);
grid on;
set(gca,'TickLabelInterpreter','latex','fontsize',FontSize);
set(gcf,'Position',[100 100 500 300])
print(gcf,'L11_fre2','-painters','-depsc','-r300')
[pks_y, locs_y] = findpeaks(y1, t);
[pks_u, locs_u] = findpeaks(u1, t);
Ay = mean(pks_y);
Au = mean(pks_u);
M5 = Ay / Au;
n = min(length(locs_y), length(locs_u));
T_vector = locs_y(1:n) - locs_u(1:n);
deltaT = mean(T_vector);
period = 2*pi*omega;
phase5 = (-2*pi*deltaT) / (T/10);
```




Frequency response for sin(0.5t)
Input
output for system


Frequency response for sin(t)
Input
output for system





3b)

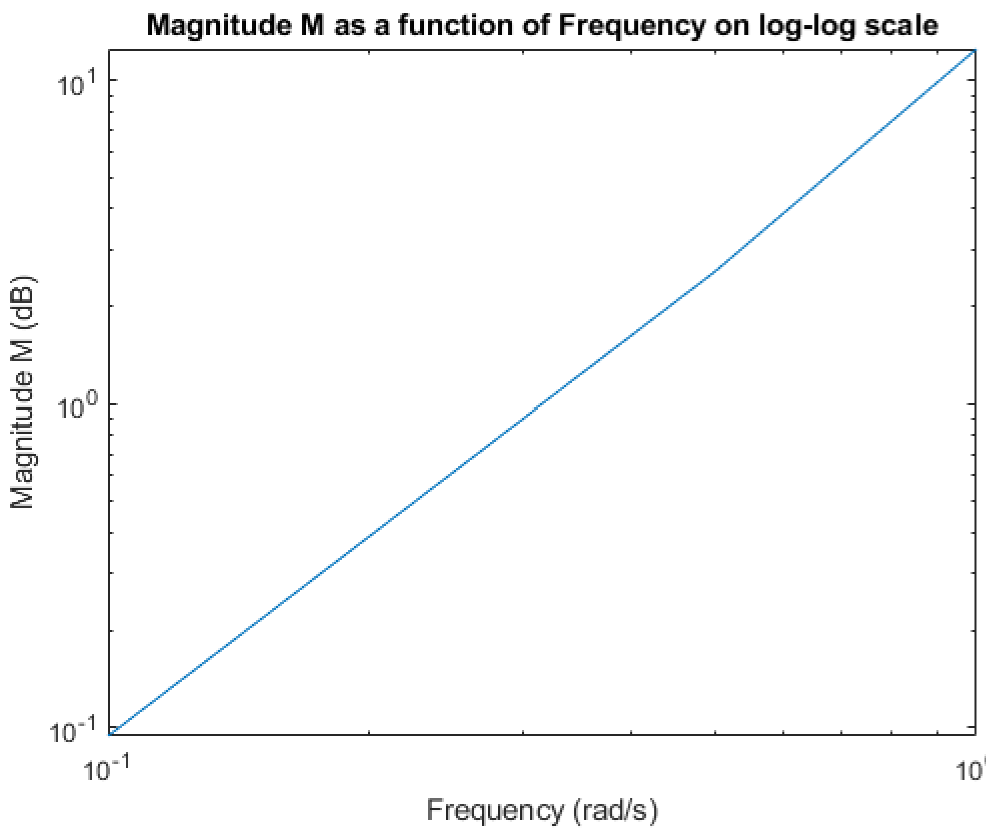
```
mag = [M1; M2; M3; M4; M5];
phi = [phase1; phase2; phase3; phase4; phase5];
omegas = [0.1; 0.5; 1; 1.5; 2];
tabl = table(omegas, 20*log10(mag), rad2deg(phi), 'VariableNames', {'omega',
'Gain M', 'Phase phi'});
disp(tabl);
```

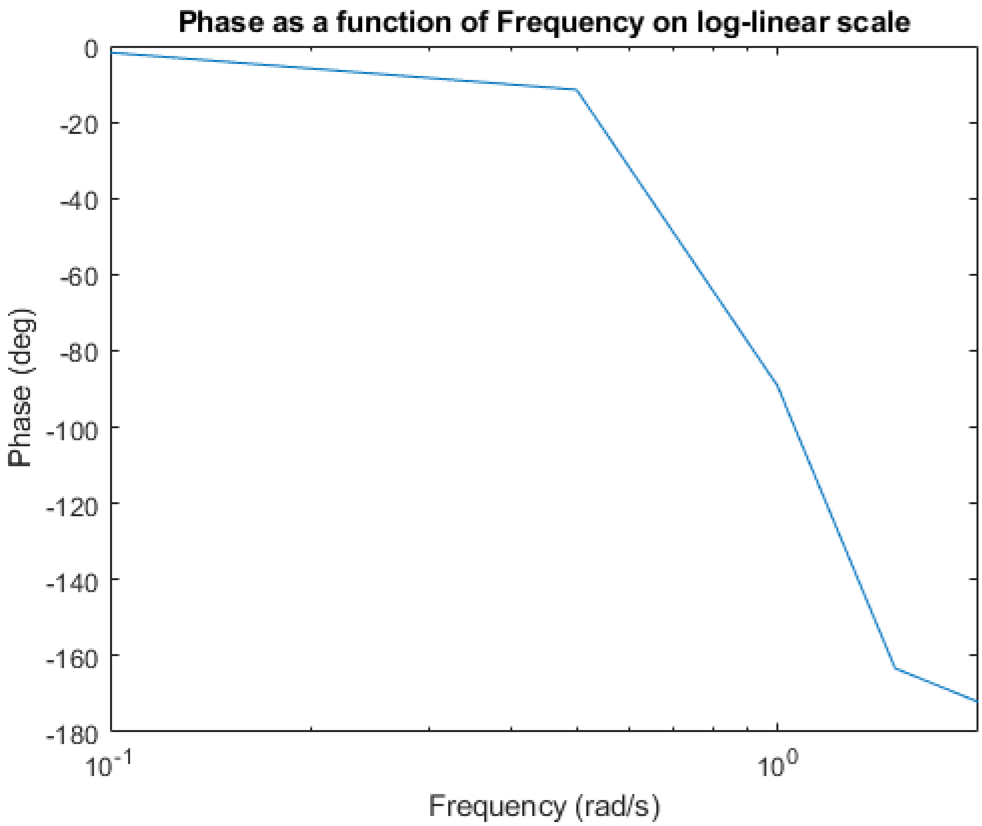
omega	Gain M	Phase phi
0.1	0.093921	-1.7279
0.5	2.5589	-11.483

```
       1        12.516      -89.251
     1.5       -1.6253      -163.26
       2       -8.7952      -172.12
```

3c)

```
figure;
loglog(omegas, 20*log10(mag));
title('Magnitude M as a function of Frequency on log-log scale');
xlabel('Frequency (rad/s)');
ylabel('Magnitude M (dB)');

figure;
semilogx(omegas, rad2deg(phi));
title('Phase as a function of Frequency on log-linear scale');
xlabel('Frequency (rad/s)');
ylabel('Phase (deg)');
```





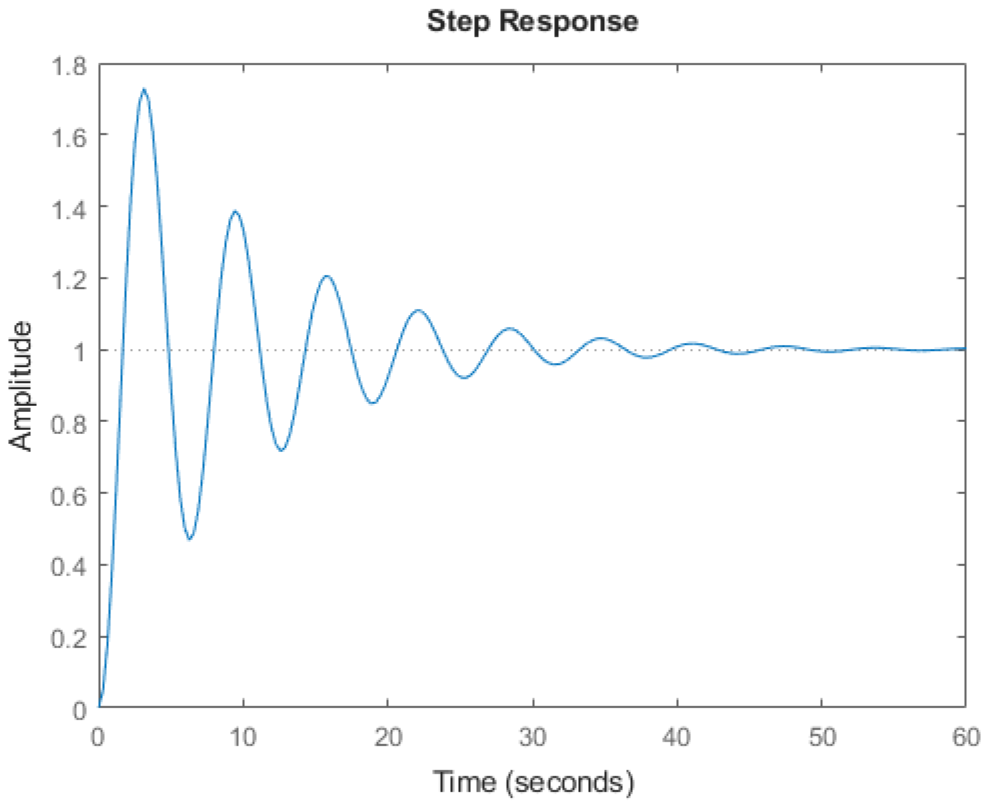
4c)

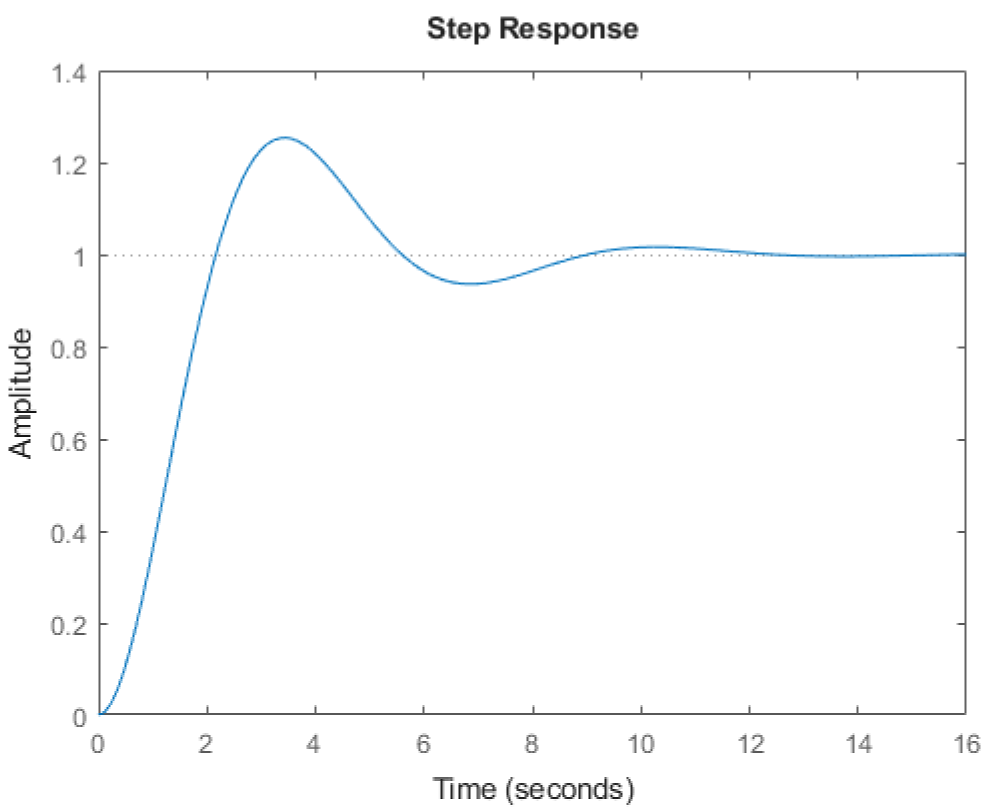
```
omega0 = 1;
%-------------------------------------------------------------------------
% Case 1: damping = 0.1
%-------------------------------------------------------------------------
damping = 0.1;
k1 = (4 * damping * omega0) - 8;
k2 = (2*omega0^2) - (4 * damping * omega0) + 6;
A = [-3 2;1, -1];
B = [0.5;0];
C = [0 1];
D = 0;
K1 = [k1 k2];
kr1 = 2*omega0^2;
sys = ss(A-B.*K1, kr1*B, C, D);
e1 = eig(A-B.*K1);
disp('Eigenvalues (damping = 0.1):');
disp(e1);
fprintf('kr(damping = 0.1): = %d\n', kr1);
disp('K matrix (damping=0.1): ')
disp(K1);
figure;
step(sys);
```

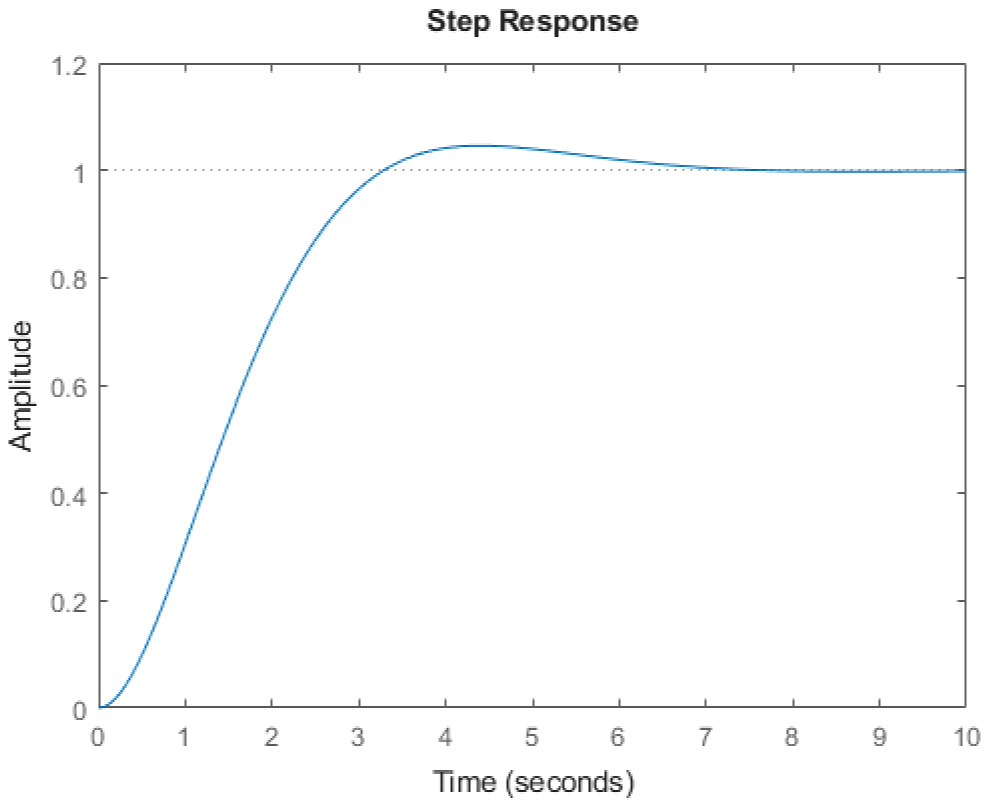
```
%--------------------------------------------------------------------------
% Case 2: damping = 0.4
%--------------------------------------------------------------------------
damping = 0.4;
k1 = (4 * damping * omega0) - 8;
k2 = (2*omega0^2) - (4 * damping * omega0) + 6;
A = [-3 2;1, -1];
B = [0.5;0];
C = [0 1];
D = 0;
K2 = [k1 k2];
kr2 = 2*omega0^2;
sys = ss(A-B.*K2, kr2*B, C, D);
e2 = eig(A-B.*K2);
disp('Eigenvalues (damping = 0.4):');
disp(e2);
fprintf('kr(damping = 0.4): = %d\n', kr2);
disp('K matrix (damping=0.4): ')
disp(K2);
figure;
step(sys);

%--------------------------------------------------------------------------
% Case 3: damping = 0.7
%--------------------------------------------------------------------------
damping = 0.7;
k1 = (4 * damping * omega0) - 8;
k2 = (2*omega0^2) - (4 * damping * omega0) + 6;
A = [-3 2;1, -1];
B = [0.5;0];
C = [0 1];
D = 0;
K3 = [k1 k2];
kr3 = 2*omega0^2;
sys = ss(A-B.*K3, kr3*B, C, D);
e3 = eig(A-B.*K3);
disp('Eigenvalues (damping = 0.7):');
disp(e3);
fprintf('kr(damping = 0.7): = %d\n', kr3);
disp('K matrix (damping=0.7): ')
disp(K3);
figure;
step(sys);

%--------------------------------------------------------------------------
% Case 4: damping = 0.9
%--------------------------------------------------------------------------
damping = 0.9;
k1 = (4 * damping * omega0) - 8;
k2 = (2*omega0^2) - (4 * damping * omega0) + 6;
A = [-3 2;1, -1];
B = [0.5;0];
C = [0 1];
D = 0;
```

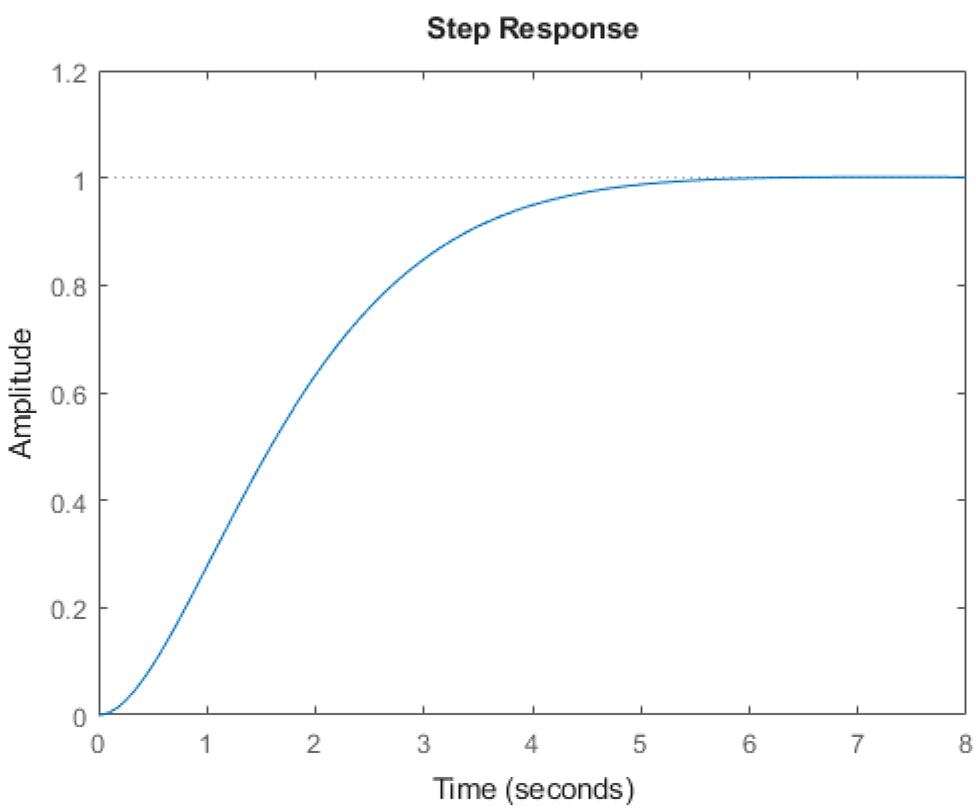
```
K4 = [k1 k2];
kr4 = 2*omega0^2;
sys = ss(A-B.*K4, kr4*B, C, D);
e4 = eig(A-B.*K4);
disp('Eigenvalues (damping = 0.9):');
disp(e4);
fprintf('kr(damping = 0.9): = %d\n', kr4);
disp('K matrix (damping=0.9): ')
disp(K4);
figure;
step(sys);

%--------------------------------------------------------------------------
% Observation:
% The amount of oscillations that the system has to go through before
% reaching steady state decreases as the damping constant increases from
% 0.1 to 0.9.
%--------------------------------------------------------------------------
```

```
Eigenvalues (damping = 0.1):
  -0.1000 + 0.9950i
  -0.1000 - 0.9950i

kr(damping = 0.1): = 2
K matrix (damping=0.1):
   -7.6000    7.6000

Eigenvalues (damping = 0.4):
  -0.4000 + 0.9165i
  -0.4000 - 0.9165i

kr(damping = 0.4): = 2
K matrix (damping=0.4):
   -6.4000    6.4000

Eigenvalues (damping = 0.7):
  -0.7000 + 0.7141i
  -0.7000 - 0.7141i

kr(damping = 0.7): = 2
K matrix (damping=0.7):
   -5.2000    5.2000

Eigenvalues (damping = 0.9):
  -0.9000 + 0.4359i
  -0.9000 - 0.4359i

kr(damping = 0.9): = 2
K matrix (damping=0.9):
   -4.4000    4.4000
```


Step Response
Amplitude
Time (seconds)
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
10
20
30
40
50
60


Step Response
Amplitude
Time (seconds)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
2
4
6
8
10
12
14
16


Step Response
Amplitude
Time (seconds)
0
0.2
0.4
0.6
0.8
1
1.2
0 1 2 3 4 5 6 7 8 9 10


Step Response
Amplitude
Time (seconds)
0
0.2
0.4
0.6
0.8
1
1.2
0 1 2 3 4 5 6 7 8

4d)

```
%--------------------------------------------------------------------------
% I would advise healthcare providers to develop and dose controllers whose
% transient overshoots don't exceed the upper limit of healthy nutrient
% intake in order to not induce any toxicity in the bloodstream and to
% develop pricing models based on individual patient data that show the
% dosage that results in the greatest health gain per dollar spent.
% Socially, there are people with different income levels, but all of them
% should get access to nutrition education so that they can learn how to
% live healthy lifestyles from a young age, and there should be publically
% funded healthcare programs that allow patients to get access to
% healthcare for a minimal cost or maybe even for free. To address the
% problem of each individual having different patterns, the solution is the
% pricing model as mentioned before that is thoroughly simulated in
% multiple scenarios with the patient's parameters in order to find the
% optimal price/dosage.
%--------------------------------------------------------------------------
```

Published with MATLAB® R2024b