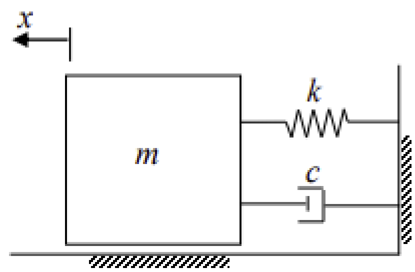
Consider the spring-mass-damper system shown.




$k$ = 1000 N/m, $m$ = 20 kg, damping ratio $\xi$ = 0.05 + 26/200

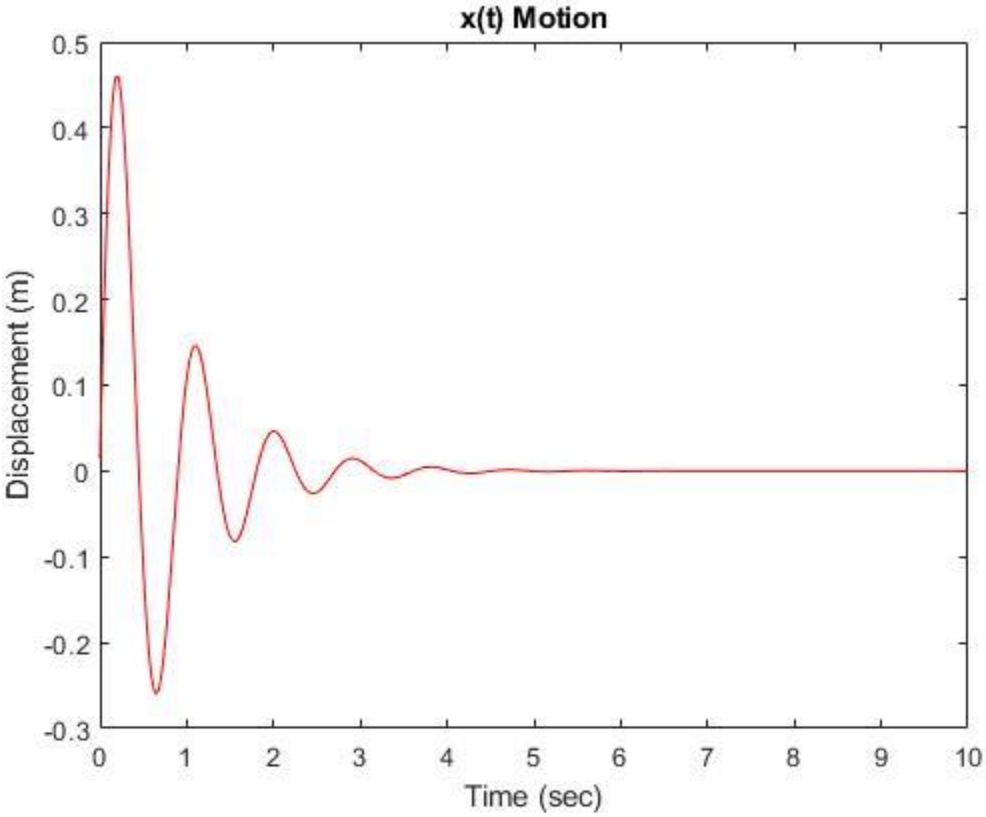
The system undergoes free vibrations with initial conditions: $xo$ = 0.1 m, $\dot{x}o$ = 4 m/s.

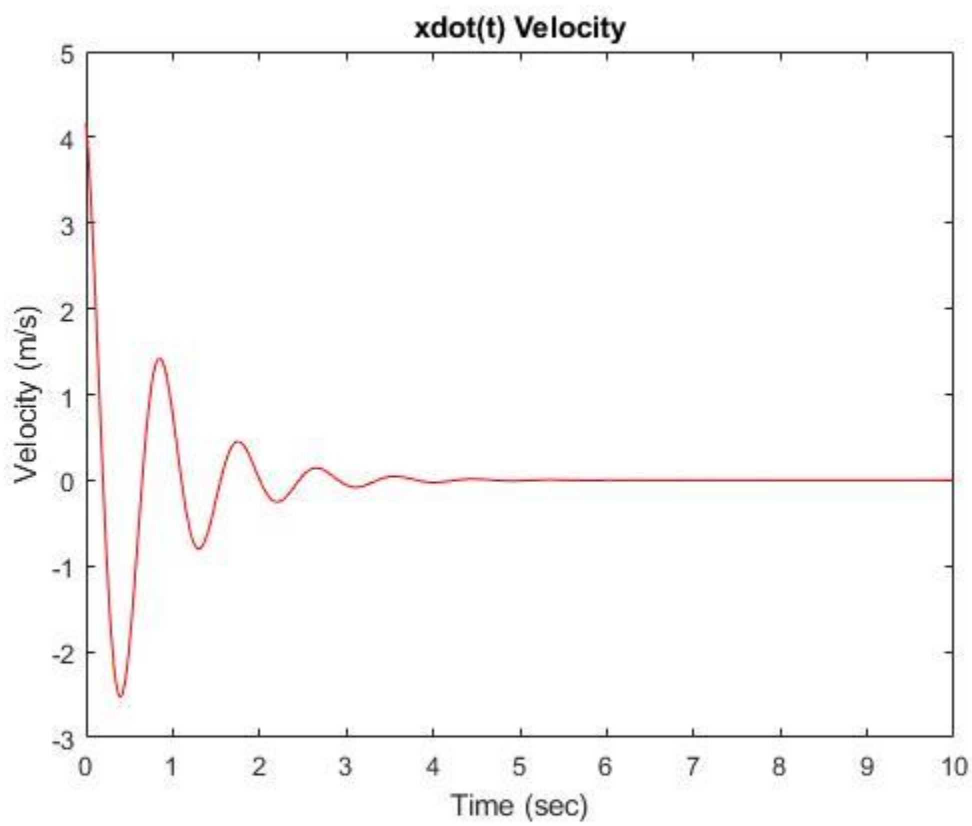
Write a MATLAB program to calculate $x(t)$, $\dot{x}(t)$ and $E(t)$ for 0 < $t$ < 10 sec. (Hint: Take simulation step size as 0.01 sec or smaller.)

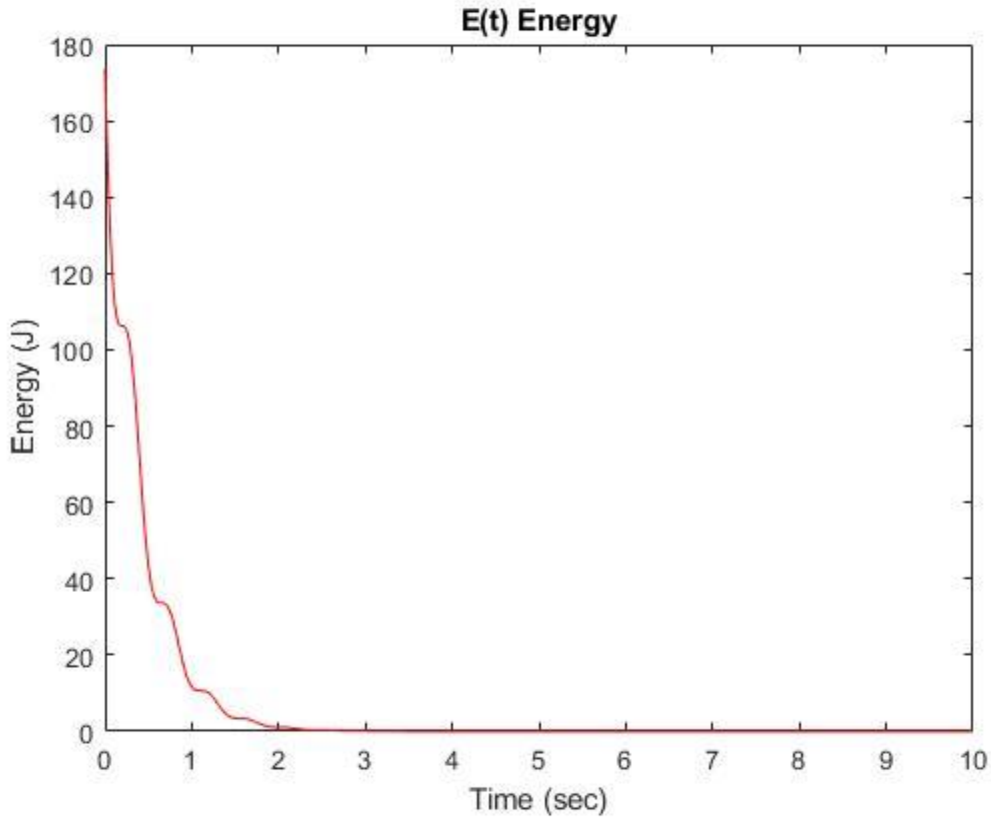
a) Plot time history of $x(t)$, $\dot{x}(t)$ and $E(t)$ for 0 < $t$ < 10 sec.

b) Determine how much time it takes for 98% of the initial energy of the system to be dissipated. Find your result within 0.01 sec accuracy.

**a.)**


x(t) Motion
Displacement (m)
Time (sec)
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0
1
2
3
4
5
6
7
8
9
10


xdot(t) Velocity
Velocity (m/s)
Time (sec)
5
4
3
2
1
0
-1
-2
-3
0
1
2
3
4
5
6
7
8
9
10




**b.)** 1.54 sec takes for 98% of the initial energy of the system to be dissipated.

## MATLAB SCRIPT

```
syms t k m ksi x0 xdot0 xt xdott Et xt_equation xdott_equation Et_equation;

k       = 1000;              % N/m
m       = 20;                % kg
ksi     = 0.05 + 26/200;
x0      = 0.1;               % m
xdot0   = 4;                 % m/s


wn      = sqrt(k/m);                                 % rad/s
wd      = wn*sqrt(1-ksi^2);                          % rad/s
A       = sqrt( x0^2 + ((xdot0+ksi*wn*x0)/wd)^2 );   % meter
fi      = atan2(((xdot0+ksi*wn*x0)/wd),x0) ;         % degree
fi_rad  = deg2rad(fi);                               % radian


xt_equation      = A.*exp(-ksi.*wn.*t).*sin(wd.*t+fi_rad);    % motion equation in meter

xdott_equation   = A*exp(-ksi*wn.*t).* ...                    % velocity in m/s
                (wd.*cos(wd.*t+fi_rad)-ksi.*wn*sin(wd.*t+fi_rad)) ;

Et_equation      = 0.5.*k.*A.^2.*exp(-2.*ksi.*wn.*t).* ...    % energy equation in J
                ((1+ksi.^2).*(sin(wd.*t+fi_rad).^2) + (1-ksi.^2).*(cos(wd.*t+fi_rad)).^2 ...
                -2.*ksi.*sqrt(1-ksi.^2).*sin(wd.*t+fi_rad).*cos(wd.*t+fi_rad)  );



t       = 0:0.01:10 ; % time interval in second, step size 0.01 sec



figure

xt      = A.*exp(-ksi.*wn.*t).*sin(wd.*t+fi_rad);

plot(t, xt, 'r-') ;
title({sprintf('x(t) Motion')})
xlabel('Time (sec)')
ylabel('Displacement (m)')

figure

xdott   = A*exp(-ksi*wn.*t).*(wd.*cos(wd.*t+fi_rad)-ksi.*wn*sin(wd.*t+fi_rad)) ;

plot(t, xdott, 'r-') ;
title({sprintf('xdot(t) Velocity')})
xlabel('Time (sec)')
ylabel('Velocity (m/s)')

figure

Et      = 0.5.*k.*A.^2.*exp(-2.*ksi.*wn.*t).* ...                % energy equation
        ((1+ksi.^2).*(sin(wd.*t+fi_rad).^2) + (1-ksi.^2).*(cos(wd.*t+fi_rad)).^2 ...
        -2.*ksi.*sqrt(1-ksi.^2).*sin(wd.*t+fi_rad).*cos(wd.*t+fi_rad)  );

plot(t, Et, 'r-') ;
title({sprintf('E(t) Energy')})
xlabel('Time (sec)')
ylabel('Energy (J)')

```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


syms energy_ratio n percent_ratio T time_decay_98;

energy_ratio = exp((4*pi*ksi)/(1-ksi^2));

percent_ratio = 100/2;                       % 98%

n = log(percent_ratio)/log(energy_ratio);    % n is cycle value for 98% energy decay

T = (2*pi)/wd;                               % Damped Period in second/cycle

time_decay_98 = T*n;                         % time in second for 98% energy decay

fprintf("%g sec takes for 98%% of the initial energy of the system to be dissipated.", round(time_decay_98,2));

```