Problem 4

11 January 2022 09:59 AM

4A)(i) For EPP system using CDM method

EQ_HW2...

78/80

28/1/22 10:09 PM EQ_HW2_P4_A1.m 1 of 3

```
%Assignment #2 P4(A)(i)-Constant-Ry Ductility & Residual Deformation Spectrum For EPP↙
system
%Central Difference Scheme
clc
fid = fopen('El Centro Ground Motion data.txt') ;  % open the text file
S = textscan(fid,'%s');   % text scan the data
fclose(fid) ;      % close the file
S = S{1} ;
a_g = cellfun(@(x)str2double(x), S);  % convert the cell array to double
% Remove NaN's which were strings earlier
a_g(isnan(a_g))=[];
col = 2;
count = 0;
temp_arr =[];
temp_row = [];
for i = 1:length(a_g)
    if count == col
        temp_arr = [temp_arr;
                    temp_row];
        count = 0;
        temp_row = [];
    end
    temp_row = [temp_row,a_g(i)];
    count = count +1;
end
temp_arr = [temp_arr;
            temp_row];
a_g = temp_arr(:,2:end);
a_g=a_g.*9.81;
clear temp_arr temp_row S;
% Creating Time axis with zero padding of 20 sec
t=zeros(length(a_g),1);
for i=2:length(a_g)+(20/0.02)
    t(i)=t(i-1)+0.02;
end
del_t=0.005;
dt=0.005; % Time step for EPP analysis
% Refning the time axis with dt=0.005
t1=0:0.005:51.180;
% Adding zero padding to the given Earthquake excitation data
a_g=[a_g;zeros((20/0.02),1)]; % appneding the a_g vector with zeros for the next 20↙
sec.
% interpolating the acceleration values within the refined time range
a_g1=interp1(t,a_g,t1);
Ry=[1,2,4,6,8]; % Array containg the Yield Strength reduction factors
Tn=0.02:0.02:50;
meu=zeros(length(Tn),5); %Matrix to store the ductility demands for each Tn against↙
given Ry
u_r=zeros(length(Tn),5); %Matrix to store the residual displacement for each Tn↙
againts given Ry
for j=1:length(Ry)
    for x=1:length(Tn)
        %Producing System response data for Equivalent Linear Elastic system
        Z=0.05; %Damping ratio
        m=1; %Considering unit mass
```

28/1/22 10:09 PM EQ_HW2_P4_A1.m 2 of 3

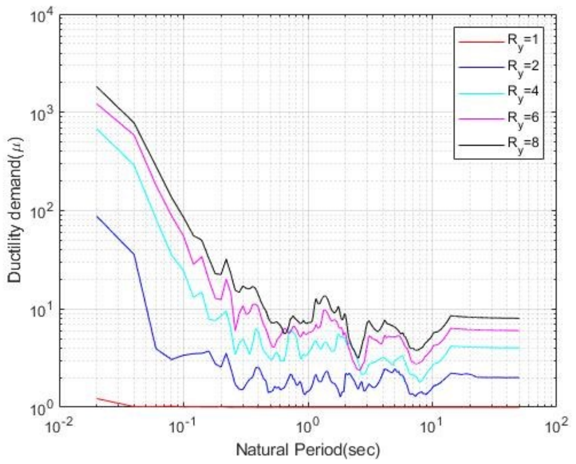
```
        Wn=(2*pi)/Tn(x); %Natural Frequency
        k=m*Wn^2; %Linear elastic Stiffness
        Wd=Wn*sqrt(1-Z^2); %Damped Natural Frequency
        %Defining Parameters required A,B,C,D & A1,B1,C1,D1
        A=exp(-Z*Wn*del_t)*((Z/sqrt(1-Z^2))*sin(Wd*del_t)+cos(Wd*del_t));
        B=exp(-Z*Wn*del_t)*(sin(Wd*del_t)/Wd);
        C=(((2*Z)/(Wn*del_t))+exp(-Z*Wn*del_t)*(((1-2*Z^2)/(Wd*del_t)-(Z/sqrt(1-Z^2)))↙
*sin(Wd*del_t)-(1+((2*Z)/(Wn*del_t)))*cos(Wd*del_t)))/Wn^2;
        D=(1-((2*Z)/(Wn*del_t))+exp(-Z*Wn*del_t)*(((2*Z^2-1)/(Wd*del_t))*sin(Wd*del_t)↙
```

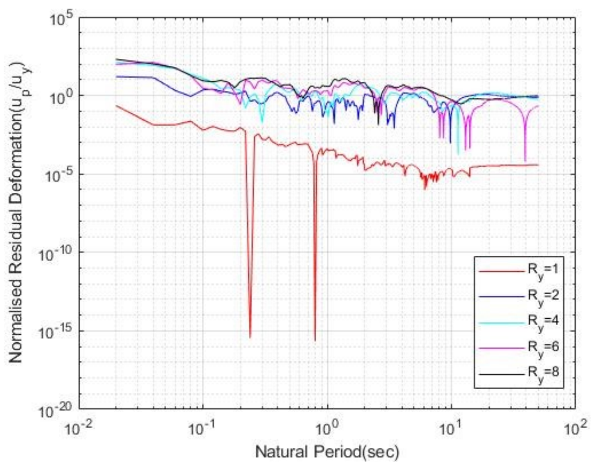
```
+((2*Z)/(Wn*del_t))*cos(Wd*del_t)))/Wn^2;
        A1=-exp(-Z*Wn*del_t)*((Wn/sqrt(1-Z^2))*sin(Wd*del_t));
        B1=exp(-Z*Wn*del_t)*(cos(Wd*del_t)-(Z/sqrt(1-Z^2))*sin(Wd*del_t));
        C1=((-1/del_t)+exp(-Z*Wn*del_t)*(((Wn/(sqrt(1-Z^2)))+(Z/(del_t*sqrt(1-Z^2))))
*sin(Wd*del_t)+(cos(Wd*del_t)/del_t)))/Wn^2;
        D1=(1-exp(-Z*Wn*del_t)*((Z/sqrt(1-Z^2))*sin(Wd*del_t)+cos(Wd*del_t)))/
(Wn^2*del_t);
        u=zeros(length(a_g1),1); %Initialising displacement response vector of the
SDOF system
        v=zeros(length(a_g1),1); %Initialising velocity response vector of the SDOF
system
        acc=zeros(length(a_g1),1);
        for i=1:length(a_g1)-1
            u(i+1)=A*u(i)+B*v(i)-C*a_g1(i)-D*a_g1(i+1);
            v(i+1)=A1*u(i)+B1*v(i)-C1*a_g1(i)-D1*a_g1(i+1);
            acc(i+1)=-a_g1(i+1)-2*Z*Wn*v(i+1)-Wn^2*u(i+1);
        end
        a_t=a_g1+acc;
        %plot(t(1:1560),u);
        umax=max(abs(u));
        f_0=k*umax; %Max. Force for system to remain Linear Elastic
        % Performing Inelastic Response Analysis
        u_epp=zeros(length(a_g1),1);
        v_epp=zeros(length(a_g1),1);
        a_epp=zeros(length(a_g1),1);
        %Initial calculations:
        fs=zeros(length(a_g1),1);
        fs(1)=k*u_epp(1);
        ry=Ry(j); %Yield Strength reduction factor
        fy=f_0/ry; %yield strength of the system
        a_epp(1)=(-m*a_g1(1)-2*m*Z*Wn*v_epp(1)-fs(1))/m; %Initial acceleration
        u_0=u_epp(1)-dt*v_epp(1)+0.5*dt^2*a_epp(1);
        k_hat=(m/dt^2)+((m*Z*Wn)/dt); %effective stiffness
        a=(m/dt^2)-((m*Z*Wn)/dt); %Integration parameter
        b=(2*m)/dt^2; %Integration parameter
        p_hat=0;du=0;fst=0;
        for i=1:length(a_g1)-1
            if i==1
                p_hat=-m*a_g1(1)-a*u_0-fs(1)+b*u_epp(1);
                u_epp(2)=p_hat/k_hat;
                v_epp(1)=(u_epp(2)-u_0)/(2*dt);
                a_epp(1)=(u_epp(2)-2*u_epp(1)+u_0)/dt^2;
            else
                p_hat=-m*a_g1(i)-a*u_epp(i-1)-fs(i)+b*u_epp(i);
                u_epp(i+1)=p_hat/k_hat;
                v_epp(i)=(u_epp(i+1)-u_epp(i-1))/(2*dt);
```



```
                a_epp(i)=(u_epp(i+1)-2*u_epp(i)+u_epp(i-1))/dt^2;
            end
            du=u_epp(i+1)-u_epp(i);
            fst=fs(i)+k*du;
            if abs(fst)>fy
                fs(i+1)=sign(fst)*fy;
            else
                fs(i+1)=fst;
            end
        end
        u_m=max(abs(u_epp)); %Maximum displacement
        meu(x,j)=abs(u_m/(fy/k));
        %Finding the residual displacement
        u_r(x,j)=abs(u_epp(end)-fs(end)/k)/(fy/k);
    end
end
%Plotting Ductility demand spectra
figure(1)
title('Constant-Ry Ductility Spectrum');
loglog(Tn,meu(:,1),'red');
hold on
loglog(Tn,meu(:,2),'blue');
hold on
loglog(Tn,meu(:,3),'cyan');
hold on
loglog(Tn,meu(:,4),'magenta');
hold on
loglog(Tn,meu(:,5),'black');
xlabel('Natural Period(sec)');
ylabel('Ductility demand(\mu)');
legend('R_{y}=1','R_{y}=2','R_{y}=4','R_{y}=6','R_{y}=8');
grid on
%Plotting Residual Displacement Spectra
figure(2)
title('Constant-Ry Normalised Residual Deformation Spectra');
loglog(Tn,u_r(:,1),'red');
hold on
```

```
loglog(Tn,u_r(:,2),'blue');
hold on
loglog(Tn,u_r(:,3),'cyan');
hold on
loglog(Tn,u_r(:,4),'magenta');
hold on
loglog(Tn,u_r(:,5),'black');
xlabel('Natural Period(sec)');
ylabel('Normalised Residual Deformation(u_{p}/u_{y})');
legend('R_{y}=1','R_{y}=2','R_{y}=4','R_{y}=6','R_{y}=8');
grid on
```





4A) (ii) For BP system using CDM method

EQ_HW2...

28/1/22 10:09 PM EQ_HW2_P4_A2.m 1 of 3

```
%Assignment #2 P4(A)(ii)-Constant-Ry Ductility & Residual Deformation Spectrum For BP
system
%Central Difference Scheme
clc
fid = fopen('El Centro Ground Motion data.txt') ;  % open the text file
S = textscan(fid,'%s');    % text scan the data
fclose(fid) ;       % close the file
S = S{1} ;
a_g = cellfun(@(x)str2double(x), S);  % convert the cell array to double
% Remove NaN's which were strings earlier
a_g(isnan(a_g))=[];
col = 2;
count = 0;
temp_arr =[];
temp_row = [];
for i = 1:length(a_g)
    if count == col
        temp_arr = [temp_arr;
                    temp_row];
        count = 0;
        temp_row = [];
    end
    temp_row = [temp_row,a_g(i)];
    count = count +1;
end
```

```
temp_arr = [temp_arr;
            temp_row];
a_g = temp_arr(:,2:end);
a_g=a_g.*9.81;
clear temp_arr temp_row S;
% Creating Time axis with zero padding of 20 sec
t=zeros(length(a_g),1);
for i=2:length(a_g)+(20/0.02)
    t(i)=t(i-1)+0.02;
end
del_t=0.005;
dt=0.005; % Time step for EPP analysis
% Refning the time axis with dt=0.005
t1=0:0.005:51.180;
% Adding zero padding to the given Earthquake excitation data
a_g=[a_g;zeros((20/0.02),1)]; % appneding the a_g vector with zeros for the next 20↙
sec.
% interpolating the acceleration values within the refined time range
a_g1=interp1(t,a_g,t1);
Ry=[1,2,4,6,8]; % Array containg the Yield Strength reduction factors
Tn=0.02:0.02:50;
meu=zeros(length(Tn),5); %Matrix to store the ductility demands for each Tn against↙
given Ry
u_r=zeros(length(Tn),5); %Matrix to store the residual displacement for each Tn↙
againts given Ry
for j=1:length(Ry)
    for x=1:length(Tn)
        %Producing System response data for Equivalent Linear Elastic system
        Z=0.05; %Damping ratio
        m=1; %Considering unit mass
```

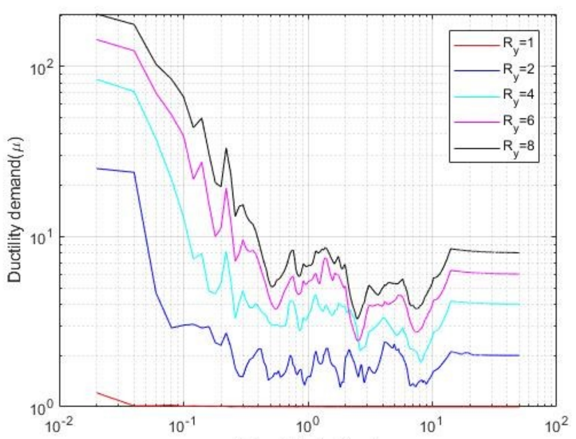


```
        Wn=(2*pi)/Tn(x); %Natural Frequency
        k=m*Wn^2; %Linear elastic Stiffness
        Wd=Wn*sqrt(1-Z^2); %Damped Natural Frequency
        %Defining Parameters required A,B,C,D & A1,B1,C1,D1
        A=exp(-Z*Wn*del_t)*((Z/sqrt(1-Z^2))*sin(Wd*del_t)+cos(Wd*del_t));
        B=exp(-Z*Wn*del_t)*(sin(Wd*del_t)/Wd);
        C=(((2*Z)/(Wn*del_t))+exp(-Z*Wn*del_t)*(((1-2*Z^2)/(Wd*del_t)-(Z/sqrt(1-Z^2)))↙
*sin(Wd*del_t)-(1+((2*Z)/(Wn*del_t)))*cos(Wd*del_t)))/Wn^2;
        D=(1-((2*Z)/(Wn*del_t))+exp(-Z*Wn*del_t)*(((2*Z^2-1)/(Wd*del_t))*sin(Wd*del_t)↙
+((2*Z)/(Wn*del_t))*cos(Wd*del_t)))/Wn^2;
        A1=-exp(-Z*Wn*del_t)*((Wn/sqrt(1-Z^2))*sin(Wd*del_t));
        B1=exp(-Z*Wn*del_t)*(cos(Wd*del_t)-(Z/sqrt(1-Z^2))*sin(Wd*del_t));
        C1=((-1/del_t)+exp(-Z*Wn*del_t)*(((Wn/(sqrt(1-Z^2)))+(Z/(del_t*sqrt(1-Z^2))))↙
*sin(Wd*del_t)+(cos(Wd*del_t)/del_t)))/Wn^2;
        D1=(1-exp(-Z*Wn*del_t)*((Z/sqrt(1-Z^2))*sin(Wd*del_t)+cos(Wd*del_t)))/↙
(Wn^2*del_t);
        u=zeros(length(a_g1),1); %Initialising displacement response vector of the↙
SDOF system
        v=zeros(length(a_g1),1); %Initialising velocity response vector of the SDOF↙
system
        acc=zeros(length(a_g1),1);
        for i=1:length(a_g1)-1
            u(i+1)=A*u(i)+B*v(i)-C*a_g1(i)-D*a_g1(i+1);
            v(i+1)=A1*u(i)+B1*v(i)-C1*a_g1(i)-D1*a_g1(i+1);
            acc(i+1)=-a_g1(i+1)-2*Z*Wn*v(i+1)-Wn^2*u(i+1);
        end
        a_t=a_g1+acc;
        %plot(t(1:1560),u);
        umax=max(abs(u));
        f_0=k*umax; %Max. Force for system to remain Linear Elastic
        % Performing Inelastic Response Analysis
        u_epp=zeros(length(a_g1),1);
        v_epp=zeros(length(a_g1),1);
        a_epp=zeros(length(a_g1),1);
        %Initial calculations:
        fs=zeros(length(a_g1),1);
        fs(1)=k*u_epp(1);
        ry=Ry(j); %Yield Strength reduction factor
        fy=f_0/ry; %yield strength of the system
        a_epp(1)=(-m*a_g1(1)-2*m*Z*Wn*v_epp(1)-fs(1))/m; %Initial acceleration
        alpha=0.05;
        k_lin=alpha*k;
        k_epp=(1-alpha)*k;
        fy_epp=(1-alpha)*fy;
        fs(1)=(k_lin+k_epp)*u_epp(1); %Initial resistive force
        u_0=u_epp(1)-dt*v_epp(1)+0.5*dt^2*a_epp(1);
        k_hat=(m/dt^2)+((m*Z*Wn)/dt); %effective stiffness
        a=(m/dt^2)-((m*Z*Wn)/dt); %Integration parameter
        b=(2*m)/dt^2; %Integration parameter
        p_hat=0;du=0;f_epp=0;f_lin=0;
        for i=1:length(a_g1)-1
            if i==1
                p_hat=-m*a_g1(1)-a*u_0-fs(1)+b*u_epp(1);
```

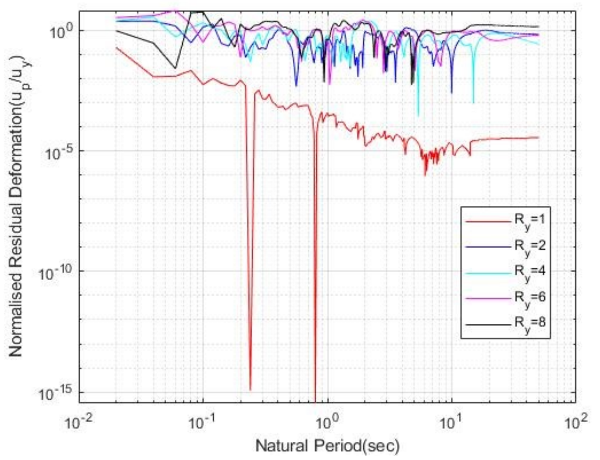
```
            u_epp(2)=p_hat/k_hat;
            v_epp(1)=(u_epp(2)-u_0)/(2*dt);
```



```
            a_epp(1)=(u_epp(2)-2*u_epp(1)+u_0)/dt^2;
        else
            p_hat=-m*a_g1(i)-a*u_epp(i-1)-fs(i)+b*u_epp(i);
            u_epp(i+1)=p_hat/k_hat;
            v_epp(i)=(u_epp(i+1)-u_epp(i-1))/(2*dt);
            a_epp(i)=(u_epp(i+1)-2*u_epp(i)+u_epp(i-1))/dt^2;
        end
        du=u_epp(i+1)-u_epp(i);
        f_lin=f_lin+k_lin*du;
        f_epp=f_epp+k_epp*du;
        if abs(f_epp)>fy_epp
            f_epp=sign(f_epp)*fy_epp;
        end
        fs(i+1)=f_lin+f_epp;
    end
    u_m=max(abs(u_epp)); %Maximum displacement
    meu(x,j)=abs(u_m/(fy/k));
    %Finding the residual displacement
    u_r(x,j)=abs(u_epp(end)-fs(end)/k)/(fy/k);
  end
end
%Plotting Ductility demand spectra
figure(1)
title('Constant-Ry Ductility Spectrum');
loglog(Tn,meu(:,1),'red');
hold on
loglog(Tn,meu(:,2),'blue');
hold on
loglog(Tn,meu(:,3),'cyan');
hold on
loglog(Tn,meu(:,4),'magenta');
hold on
loglog(Tn,meu(:,5),'black');
xlabel('Natural Period(sec)');
ylabel('Ductility demand(\mu)');
legend('R_{y}=1','R_{y}=2','R_{y}=4','R_{y}=6','R_{y}=8');
grid on
%Plotting Residual Displacement Spectra
figure(2)
title('Constant-Ry Normalised Residual Deformation Spectra');
loglog(Tn,u_r(:,1),'red');
hold on
loglog(Tn,u_r(:,2),'blue');
hold on
loglog(Tn,u_r(:,3),'cyan');
hold on
loglog(Tn,u_r(:,4),'magenta');
hold on
loglog(Tn,u_r(:,5),'black');
xlabel('Natural Period(sec)');
ylabel('Normalised Residual Deformation(u_{p}/u_{y})');
legend('R_{y}=1','R_{y}=2','R_{y}=4','R_{y}=6','R_{y}=8');
grid on
```



Natural Period(sec)



4A)(iii) For BEL system using CDM method

EQ_HW2...


```
%Assignment #2 P4(A)(i)-Constant-Ry Ductility & residual Deformation Spectrum For BEL system
%Central Difference Scheme
clc
fid = fopen('El Centro Ground Motion data.txt') ;  % open the text file
S = textscan(fid,'%s');   % text scan the data
fclose(fid) ;      % close the file
S = S{1} ;
a_g = cellfun(@(x)str2double(x), S);  % convert the cell array to double
% Remove NaN's which were strings earlier
a_g(isnan(a_g))=[];
col = 2;
count = 0;
temp_arr =[];
temp_row = [];
for i = 1:length(a_g)
    if count == col
        temp_arr = [temp_arr;
                    temp_row];
        count = 0;
        temp_row = [];
    end
    temp_row = [temp_row,a_g(i)];
    count = count +1;
end
temp_arr = [temp_arr;
            temp_row];
a_g = temp_arr(:,2:end);
a_g=a_g.*9.81;
clear temp_arr temp_row S;
% Creating Time axis with zero padding of 20 sec
t=zeros(length(a_g),1);
for i=2:length(a_g)+(20/0.02)
    t(i)=t(i-1)+0.02;
end
del_t=0.005;
dt=0.005; % Time step for EPP analysis
% Refning the time axis with dt=0.005
t1=0:0.005:51.180;
% Adding zero padding to the given Earthquake excitation data
a_g=[a_g;zeros((20/0.02),1)]; % appneding the a_g vector with zeros for the next 20 sec.
% interpolating the acceleration values within the refined time range
a_g1=interp1(t,a_g,t1);
Ry=[1,2,4,6,8]; % Array containg the Yield Strength reduction factors
Tn=0.02:0.02:50;
meu=zeros(length(Tn),5); %Matrix to store the ductility demands for each Tn against given Ry
u_r=zeros(length(Tn),5); %Matrix to store the residual displacement for each Tn agaiints given Ry
for j=1:length(Ry)
    for x=1:length(Tn)
        %Producing System response data for Equivalent Linear Elastic system
        Z=0.05; %Damping ratio
        m=1; %Considering unit mass
```

```
        Wn=(2*pi)/Tn(x); %Natural Frequency
        k=m*Wn^2; %Linear elastic Stiffness
        Wd=Wn*sqrt(1-Z^2); %Damped Natural Frequency
        %Defining Parameters required A,B,C,D & A1,B1,C1,D1
        A=exp(-Z*Wn*del_t)*((Z/sqrt(1-Z^2))*sin(Wd*del_t)+cos(Wd*del_t));
        B=exp(-Z*Wn*del_t)*(sin(Wd*del_t)/Wd);
        C=(((2*Z)/(Wn*del_t))+exp(-Z*Wn*del_t)*(((1-2*Z^2)/(Wd*del_t)-(Z/sqrt(1-Z^2)))↙
*sin(Wd*del_t)-(1+((2*Z)/(Wn*del_t)))*cos(Wd*del_t)))/Wn^2;
        D=(1-((2*Z)/(Wn*del_t))+exp(-Z*Wn*del_t)*(((2*Z^2-1)/(Wd*del_t))*sin(Wd*del_t)↙
+((2*Z)/(Wn*del_t))*cos(Wd*del_t)))/Wn^2;
        A1=-exp(-Z*Wn*del_t)*((Wn/sqrt(1-Z^2))*sin(Wd*del_t));
        B1=exp(-Z*Wn*del_t)*(cos(Wd*del_t)-(Z/sqrt(1-Z^2))*sin(Wd*del_t));
        C1=((-1/del_t)+exp(-Z*Wn*del_t)*(((Wn/(sqrt(1-Z^2)))+(Z/(del_t*sqrt(1-Z^2))))↙
*sin(Wd*del_t)+(cos(Wd*del_t)/del_t)))/Wn^2;
        D1=(1-exp(-Z*Wn*del_t)*((Z/sqrt(1-Z^2))*sin(Wd*del_t)+cos(Wd*del_t)))/↙
(Wn^2*del_t);
        u=zeros(length(a_g1),1); %Initialising displacement response vector of the↙
SDOF system
        v=zeros(length(a_g1),1); %Initialising velocity response vector of the SDOF↙
system
        acc=zeros(length(a_g1),1);
        for i=1:length(a_g1)-1
            u(i+1)=A*u(i)+B*v(i)-C*a_g1(i)-D*a_g1(i+1);
            v(i+1)=A1*u(i)+B1*v(i)-C1*a_g1(i)-D1*a_g1(i+1);
            acc(i+1)=-a_g1(i+1)-2*Z*Wn*v(i+1)-Wn^2*u(i+1);
        end
        a_t=a_g1+acc;
        %plot(t(1:1560),u);
        umax=max(abs(u));
        f_0=k*umax; %Max. Force for system to remain Linear Elastic
        % Performing Inelastic Response Analysis
        u_epp=zeros(length(a_g1),1);
        v_epp=zeros(length(a_g1),1);
        a_epp=zeros(length(a_g1),1);
        %Initial calculations:
        fs=zeros(length(a_g1),1);
        fs(1)=k*u_epp(1);
        ry=Ry(j); %Yield Strength reduction factor
        fy=f_0/ry; %yield strength of the system
        a_epp(1)=(-m*a_g1(1)-2*m*Z*Wn*v_epp(1)-fs(1))/m; %Initial acceleration
        alpha=0.05;
        k_lin=alpha*k;
        k_bel=(1-alpha)*k;
        fy_bel=(1-alpha)*fy;
        fs(1)=(k_lin+k_bel)*u_epp(1); %Initial resistive force
        u_0=u_epp(1)-dt*v_epp(1)+0.5*dt^2*a_epp(1);
        k_hat=(m/dt^2)+((m*Z*Wn)/dt); %effective stiffness
        a=(m/dt^2)-((m*Z*Wn)/dt); %Integration parameter
        b=(2*m)/dt^2; %Integration parameter
        p_hat=0;du=0;f_lin=0;f_bel=0;
        for i=1:length(a_g1)-1
            if i==1
                p_hat=-m*a_g1(1)-a*u_0-fs(1)+b*u_epp(1);
                u_epp(2)=p_hat/k_hat;
                v_epp(1)=(u_epp(2)-u_0)/(2*dt);
```

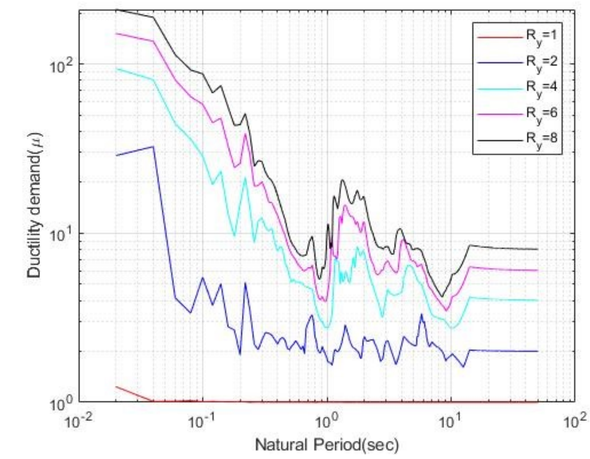


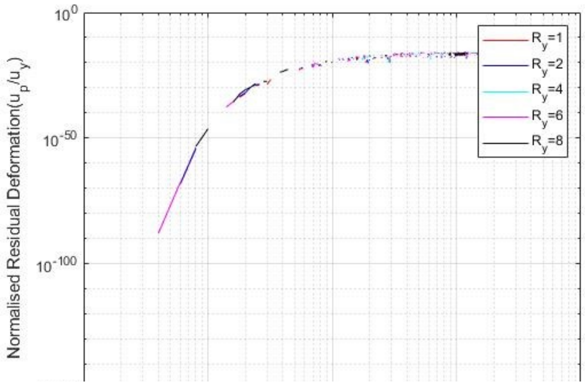
```
                a_epp(1)=(u_epp(2)-2*u_epp(1)+u_0)/dt^2;
            else
                p_hat=-m*a_g1(i)-a*u_epp(i-1)-fs(i)+b*u_epp(i);
                u_epp(i+1)=p_hat/k_hat;
                v_epp(i)=(u_epp(i+1)-u_epp(i-1))/(2*dt);
                a_epp(i)=(u_epp(i+1)-2*u_epp(i)+u_epp(i-1))/dt^2;
            end
            du=u_epp(i+1)-u_epp(i);
            if abs(u_epp(i+1))>(fy/k)
                f_bel=sign(u_epp(i+1))*fy_bel;
                f_lin=k_lin*u_epp(i+1);
            else
                f_bel=k_bel*u_epp(i+1);
                f_lin=k_lin*u_epp(i+1);
            end
                fs(i+1)=f_lin+f_bel;
        end
        u_m=max(abs(u_epp)); %Maximum displacement
        meu(x,j)=abs(u_m/(fy/k));
        %Finding the residual displacement
        if abs(u_epp(end))>(fy/k)
```

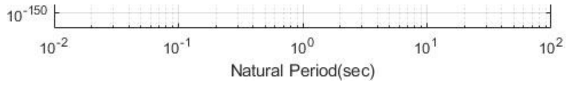
```
            if u_epp(end)<0
                u_r(x,j)=abs(u_epp(end)+((abs(fs(end))-fy)/k_lin)+(fy/k));
            else
                u_r(x,j)=abs(u_epp(end)-((fs(end)-fy)/k_lin)-(fy/k));
            end
        else
            u_r(x,j)=abs(u_epp(end)-fs(end)/k)/(fy/k);
        end
    end
end
%Plotting Ductility demand spectra
figure(1)
title('Constant-Ry Ductility Spectrum');
loglog(Tn,meu(:,1),'red');
hold on
loglog(Tn,meu(:,2),'blue');
hold on
loglog(Tn,meu(:,3),'cyan');
hold on
loglog(Tn,meu(:,4),'magenta');
hold on
loglog(Tn,meu(:,5),'black');
xlabel('Natural Period(sec)');
ylabel('Ductility demand(\mu)');
legend('R_{y}=1','R_{y}=2','R_{y}=4','R_{y}=6','R_{y}=8');
grid on
%Plotting Residual Displacement Spectra
figure(2)
title('Constant-Ry Normalised Residual Deformation Spectra');
semilogx(Tn,u_r(:,1),'red');
hold on
semilogx(Tn,u_r(:,2),'blue');
hold on
semilogx(Tn,u_r(:,3),'cyan');
```



```
hold on
semilogx(Tn,u_r(:,4),'magenta');
hold on
semilogx(Tn,u_r(:,5),'black');
xlabel('Natural Period(sec)');
ylabel('Normalised Residual Deformation(u_{p}/u_{y})');
legend('R_{y}=1','R_{y}=2','R_{y}=4','R_{y}=6','R_{y}=8');
grid on
```







Comments:
The Ductility Spectra for the EPP system has the highest values in the acceleration sensitive region i.e. at low natural periods followed by the BEL system and then the BP system.
Also at higher natural periods, the ductility demand becomes almost equal to the Yield Strength Reduction factor (Ry).
Lastly for Ry=1, which basically means a perfectly linear elastic behaviour, the ductility demand is zero, since the maximum displacement is same as the yield displacement.
For all cases of Ry in BEL systems, the residual deformation is practically zero since it is an Elastic system and always comes back to its original configuration at the end of the motion.

4B)(i) [**Note:** In part B, I have used a single script for finding the plots for the 4 cases of Ry. I have just changed the values of Ry(rest everything is unchanged) and ran the script to generate the plots for the 4 values of Ry. Also this script has used functions which are also given after the main script.]

EQ_HW2...


```
%Assignment #2 P4(A)(i)-Constant-Ry Ductility & residual Deformation Spectrum For BEL
system
%Central Difference Scheme
clc
fid = fopen('El Centro Ground Motion data.txt') ;  % open the text file
S = textscan(fid,'%s');   % text scan the data
fclose(fid) ;      % close the file
S = S{1} ;
a_g = cellfun(@(x)str2double(x), S);  % convert the cell array to double
% Remove NaN's which were strings earlier
a_g(isnan(a_g))=[];
col = 2;
count = 0;
temp_arr =[];
temp_row = [];
for i = 1:length(a_g)
    if count == col
        temp_arr = [temp_arr;
                    temp_row];
        count = 0;
        temp_row = [];
    end
    temp_row = [temp_row,a_g(i)];
    count = count +1;
end
temp_arr = [temp_arr;
            temp_row];
a_g = temp_arr(:,2:end);
a_g=a_g.*9.81;
clear temp_arr temp_row S;
% Creating Time axis with zero padding of 20 sec
t=zeros(length(a_g),1);
for i=2:length(a_g)+(20/0.02)
    t(i)=t(i-1)+0.02;
end
del_t=0.005;
dt=0.005; % Time step for EPP analysis
% Refning the time axis with dt=0.005
t1=0:0.005:51.180;
% Adding zero padding to the given Earthquake excitation data
a_g=[a_g;zeros((20/0.02),1)]; % appneding the a_g vector with zeros for the next 20
sec.
% interpolating the acceleration values within the refined time range
a_g1=interp1(t,a_g,t1);
Ry=4; % Array containg the Yield Strength reduction factors
Tn=0.02:0.02:50;
meu_epp=zeros(length(Tn),1); %Matrix to store the ductility demands for each Tn
against given Ry for EPP system
u_r_epp=zeros(length(Tn),1); %Matrix to store the residual displacement for each Tn
against given Ry for EPP system
meu_bp=zeros(length(Tn),1);  %%Matrix to store the ductility demands for each Tn
against given Ry for BP system
u_r_bp=zeros(length(Tn),1); %Matrix to store the residual displacement for each Tn
againts given Ry for BP system
meu_bel=zeros(length(Tn),1);  %%Matrix to store the ductility demands for each Tn
```

```
against given Ry for BEL system
u_r_bel=zeros(length(Tn),1); %Matrix to store the residual displacement for each Tn↙
againts given Ry for BEL system
for x=1:length(Tn)
        %Producing System response data for Equivalent Linear Elastic system
        Z=0.05; %Damping ratio
        m=1; %Considering unit mass
        Wn=(2*pi)/Tn(x); %Natural Frequency
        k=m*Wn^2; %Linear elastic Stiffness
        Wd=Wn*sqrt(1-Z^2); %Damped Natural Frequency
        %Defining Parameters required A,B,C,D & A1,B1,C1,D1
        A=exp(-Z*Wn*del_t)*((Z/sqrt(1-Z^2))*sin(Wd*del_t)+cos(Wd*del_t));
        B=exp(-Z*Wn*del_t)*(sin(Wd*del_t)/Wd);
        C=(((2*Z)/(Wn*del_t))+exp(-Z*Wn*del_t)*(((1-2*Z^2)/(Wd*del_t)-(Z/sqrt(1-Z^2)))↙
*sin(Wd*del_t)-(1+((2*Z)/(Wn*del_t)))*cos(Wd*del_t)))/Wn^2;
        D=(1-((2*Z)/(Wn*del_t))+exp(-Z*Wn*del_t)*(((2*Z^2-1)/(Wd*del_t))*sin(Wd*del_t)↙
+((2*Z)/(Wn*del_t))*cos(Wd*del_t)))/Wn^2;
        A1=-exp(-Z*Wn*del_t)*((Wn/sqrt(1-Z^2))*sin(Wd*del_t));
        B1=exp(-Z*Wn*del_t)*(cos(Wd*del_t)-(Z/sqrt(1-Z^2))*sin(Wd*del_t));
        C1=((-1/del_t)+exp(-Z*Wn*del_t)*(((Wn/(sqrt(1-Z^2)))+(Z/(del_t*sqrt(1-Z^2))))↙
*sin(Wd*del_t)+(cos(Wd*del_t)/del_t)))/Wn^2;
        D1=(1-exp(-Z*Wn*del_t)*((Z/sqrt(1-Z^2))*sin(Wd*del_t)+cos(Wd*del_t)))/↙
(Wn^2*del_t);
        u=zeros(length(a_g1),1); %Initialising displacement response vector of the↙
SDOF system
        v=zeros(length(a_g1),1); %Initialising velocity response vector of the SDOF↙
system
        acc=zeros(length(a_g1),1);
        for i=1:length(a_g1)-1
            u(i+1)=A*u(i)+B*v(i)-C*a_g1(i)-D*a_g1(i+1);
            v(i+1)=A1*u(i)+B1*v(i)-C1*a_g1(i)-D1*a_g1(i+1);
            acc(i+1)=-a_g1(i+1)-2*Z*Wn*v(i+1)-Wn^2*u(i+1);
        end
        a_t=a_g1+acc;
        %plot(t(1:1560),u);
        umax=max(abs(u));
        f_0=k*umax; %Max. Force for system to remain Linear Elastic
        % Performing Inelastic Response Analysis
        fy=f_0/Ry; %yield strength of the system
        %Defining parameters for EPP system
        u_epp=zeros(length(a_g1),1);
        v_epp=zeros(length(a_g1),1);
        a_epp=zeros(length(a_g1),1);
        fs_epp=zeros(length(a_g1),1);
        [meu_epp(x,1),u_r_epp(x,1)]=ElastoPlastic(m,Z,Wn,dt,a_g1,k,fy,u_epp,v_epp,↙
a_epp,fs_epp);
        %Defining parameters for BP system
        u_bp=zeros(length(a_g1),1);
        v_bp=zeros(length(a_g1),1);
        a_bp=zeros(length(a_g1),1);
        fs_bp=zeros(length(a_g1),1);
        [meu_bp(x,1),u_r_bp(x,1)]=BilinearPlastic(m,Z,Wn,dt,a_g1,k,fy,u_bp,v_bp,a_bp,↙
fs_bp);
        %Defining parameters for BEL system
        u_bel=zeros(length(a_g1),1);
```



```
        v_bel=zeros(length(a_g1),1);
        a_bel=zeros(length(a_g1),1);
        fs_bel=zeros(length(a_g1),1);
        [meu_bel(x,1),u_r_bel(x,1)]=BilinearElastic(m,Z,Wn,dt,a_g1,k,fy,u_bel,v_bel,↙
a_bel,fs_bel);
end
%Plotting Ductility demand spectra
figure(1)
title('Constant-Ry(=2) Ductility Spectrum for different systems');
loglog(Tn,meu_epp(:,1),'red');
hold on
loglog(Tn,meu_bp(:,1),'blue');
hold on
loglog(Tn,meu_bel(:,1),'black');
hold on
xlabel('Natural Period(sec)');
ylabel('Ductility demand(\mu)');
legend('EPP system','BP system','BEL system');
grid on
%Plotting Residual Displacement Spectra
figure(2)
```

```
title('Constant-Ry(=2) Normalised Residual Deformation Spectra for different
systems');
loglog(Tn,u_r_epp(:,1),'red');
hold on
loglog(Tn,u_r_bp(:,1),'blue');
hold on
loglog(Tn,u_r_bel(:,1),'black');
hold on
xlabel('Natural Period(sec)');
ylabel('Normalised Residual Deformation(u_{p}/u_{y})');
legend('EPP system','BP system','BEL system');
grid on
```

The Functions for EPP, BP and BEL systems are given:

ElastoPla...

BilinearPl...

BilinearEl...

29/1/22 7:19 PM ElastoPlastic.m 1 of 1

```
function [meu,u_r] = ElastoPlastic(m,Z,Wn,dt,a_g1,k,fy,u_epp,v_epp,a_epp,fs_epp)
%Inelastic SDOF response analysis for EPP system using CDM
fs_epp(1)=k*u_epp(1);
a_epp(1)=(-m*a_g1(1)-2*m*Z*Wn*v_epp(1)-fs_epp(1))/m; %Initial acceleration
u_0=u_epp(1)-dt*v_epp(1)+0.5*dt^2*a_epp(1);
k_hat=(m/dt^2)+((m*Z*Wn)/dt); %effective stiffness
a=(m/dt^2)-((m*Z*Wn)/dt); %Integration parameter
b=(2*m)/dt^2; %Integration parameter
p_hat=0;du=0;fst=0;
for i=1:length(a_g1)-1
    if i==1
        p_hat=-m*a_g1(1)-a*u_0-fs_epp(1)+b*u_epp(1);
        u_epp(2)=p_hat/k_hat;
        v_epp(1)=(u_epp(2)-u_0)/(2*dt);
        a_epp(1)=(u_epp(2)-2*u_epp(1)+u_0)/dt^2;
    else
        p_hat=-m*a_g1(i)-a*u_epp(i-1)-fs_epp(i)+b*u_epp(i);
        u_epp(i+1)=p_hat/k_hat;
        v_epp(i)=(u_epp(i+1)-u_epp(i-1))/(2*dt);
        a_epp(i)=(u_epp(i+1)-2*u_epp(i)+u_epp(i-1))/dt^2;
    end
    du=u_epp(i+1)-u_epp(i);
    fst=fs_epp(i)+k*du;
    if abs(fst)>fy
        fs_epp(i+1)=sign(fst)*fy;
    else
        fs_epp(i+1)=fst;
    end
end
u_m=max(abs(u_epp)); %Maximum displacement
meu=abs(u_m/(fy/k));
%Finding the residual displacement
u_r=abs(u_epp(end)-fs_epp(end)/k)/(fy/k);
end
```

29/1/22 7:18 PM BilinearPlastic.m 1 of 1

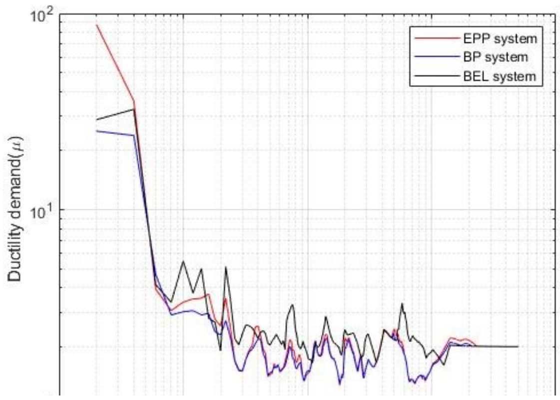
```
function [meu,u_r] = BilinearPlastic(m,Z,Wn,dt,a_g1,k,fy,u_bp,v_bp,a_bp,fs_bp)
%Inelastic SDOF response analysis for BP system using CDM
a_bp(1)=(-m*a_g1(1)-2*m*Z*Wn*v_bp(1)-fs_bp(1))/m; %Initial acceleration
alpha=0.05;
k_lin=alpha*k;
k_epp=(1-alpha)*k;
fy_epp=(1-alpha)*fy;
fs_bp(1)=(k_lin+k_epp)*u_bp(1); %Initial resistive force
u_0=u_bp(1)-dt*v_bp(1)+0.5*dt^2*a_bp(1);
k_hat=(m/dt^2)+((m*Z*Wn)/dt); %effective stiffness
a=(m/dt^2)-((m*Z*Wn)/dt); %Integration parameter
b=(2*m)/dt^2; %Integration parameter
p_hat=0;du=0;f_epp=0;f_lin=0;
for i=1:length(a_g1)-1
    if i==1
        p_hat=-m*a_g1(1)-a*u_0-fs_bp(1)+b*u_bp(1);
        u_bp(2)=p_hat/k_hat;
        v_bp(1)=(u_bp(2)-u_0)/(2*dt);
```

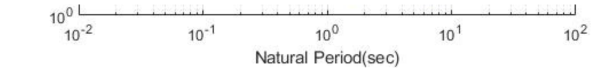
```
        a_bp(1)=(u_bp(2)-2*u_bp(1)+u_0)/dt^2;
    else
        p_hat=-m*a_g1(i)-a*u_bp(i-1)-fs_bp(i)+b*u_bp(i);
        u_bp(i+1)=p_hat/k_hat;
        v_bp(i)=(u_bp(i+1)-u_bp(i-1))/(2*dt);
        a_bp(i)=(u_bp(i+1)-2*u_bp(i)+u_bp(i-1))/dt^2;
    end
    du=u_bp(i+1)-u_bp(i);
    f_lin=f_lin+k_lin*du;
    f_epp=f_epp+k_epp*du;
    if abs(f_epp)>fy_epp
        f_epp=sign(f_epp)*fy_epp;
    end
    fs_bp(i+1)=f_lin+f_epp;
end
u_m=max(abs(u_bp)); %Maximum displacement
meu=abs(u_m/(fy/k));
%Finding the residual displacement
u_r=abs(u_bp(end)-fs_bp(end)/k)/(fy/k);
end
```

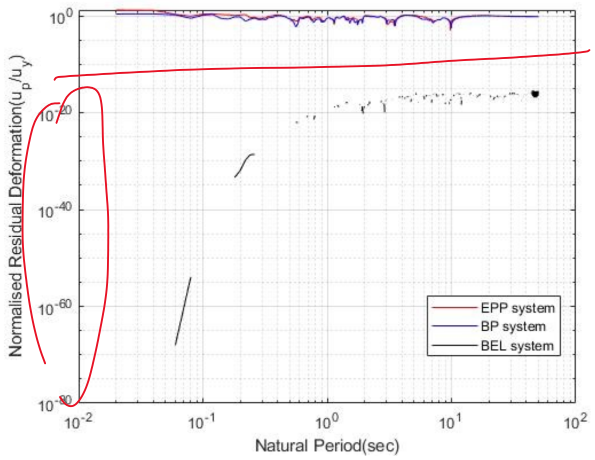
29/1/22 7:18 PM BilinearElastic.m 1 of 1

```
function [meu,u_r] =BilinearElastic(m,Z,Wn,dt,a_g1,k,fy,u_bel,v_bel,a_bel,fs_bel)
%Inelastic SDOF response analysis for BEL system using CDM
a_bel(1)=(-m*a_g1(1)-2*m*Z*Wn*v_bel(1)-fs_bel(1))/m; %Initial acceleration
alpha=0.05;
k_lin=alpha*k;
k_bel=(1-alpha)*k;
fy_bel=(1-alpha)*fy;
fs_bel(1)=(k_lin+k_bel)*u_bel(1); %Initial resistive force
u_0=u_bel(1)-dt*v_bel(1)+0.5*dt^2*a_bel(1);
k_hat=(m/dt^2)+((m*Z*Wn)/dt); %effective stiffness
a=(m/dt^2)-((m*Z*Wn)/dt); %Integration parameter
b=(2*m)/dt^2; %Integration parameter
p_hat=0;du=0;f_lin=0;f_bel=0;
for i=1:length(a_g1)-1
    if i==1
        p_hat=-m*a_g1(1)-a*u_0-fs_bel(1)+b*u_bel(1);
        u_bel(2)=p_hat/k_hat;
        v_bel(1)=(u_bel(2)-u_0)/(2*dt);
        a_bel(1)=(u_bel(2)-2*u_bel(1)+u_0)/dt^2;
    else
        p_hat=-m*a_g1(i)-a*u_bel(i-1)-fs_bel(i)+b*u_bel(i);
        u_bel(i+1)=p_hat/k_hat;
        v_bel(i)=(u_bel(i+1)-u_bel(i-1))/(2*dt);
        a_bel(i)=(u_bel(i+1)-2*u_bel(i)+u_bel(i-1))/dt^2;
    end
    du=u_bel(i+1)-u_bel(i);
    if abs(u_bel(i+1))>(fy/k)
        f_bel=sign(u_bel(i+1))*fy_bel;
        f_lin=k_lin*u_bel(i+1);
    else
        f_bel=k_bel*u_bel(i+1);
        f_lin=k_lin*u_bel(i+1);
    end
    fs_bel(i+1)=f_lin+f_bel;
end
u_m=max(abs(u_bel)); %Maximum displacement
meu=abs(u_m/(fy/k));
if abs(u_bel(end))>(fy/k)
    if u_bel(end)<0
        u_r=abs(u_bel(end)+((abs(fs_bel(end))-fy)/k_lin)+(fy/k));
    else
        u_r=abs(u_bel(end)-((fs_bel(end)-fy)/k_lin)-(fy/k));
    end
else
    u_r=abs(u_bel(end)-fs_bel(end)/k)/(fy/k);
end
end
```

(i) For Ry=2

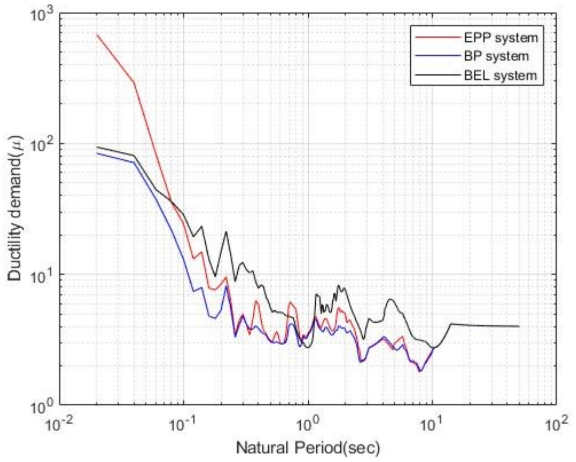


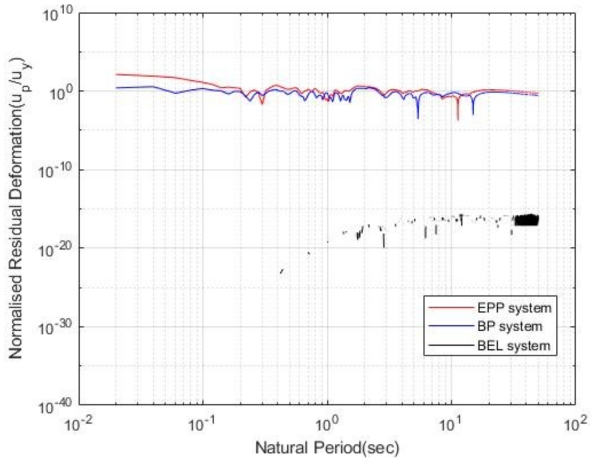




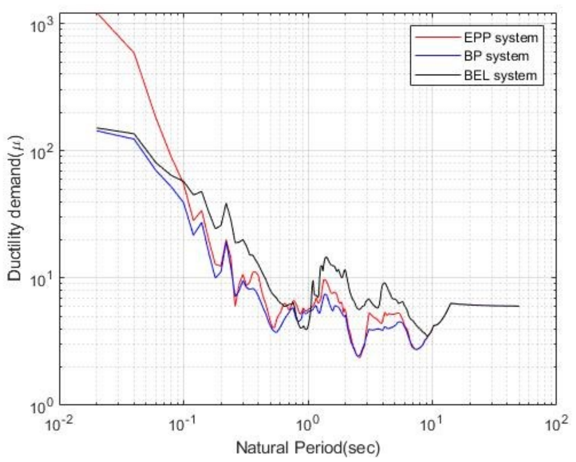
use y lim

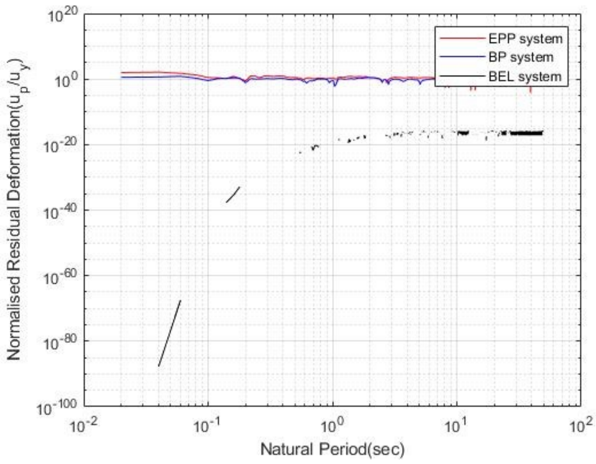
(ii) For Ry=4



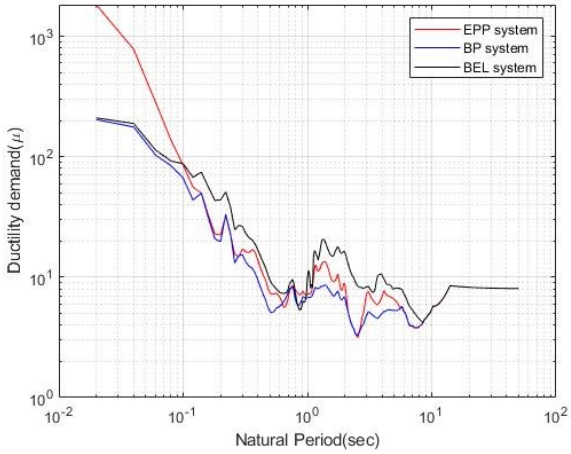


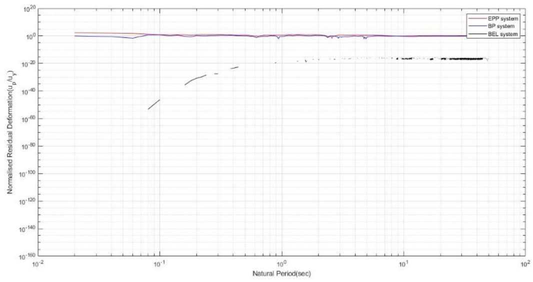
(iii) For Ry=6





(iv) For Ry=8





Comments:

For each of the ductility demand plots, in velocity sensitive region, we observe that the EPP and BP systems behave in a similar manner with their values almost close to one another. The BEL system on the other hand shows a slightly higher ductility demand in this region.
In the acceleration sensitive region, as we have previously discussed, the EPP system shows quite a higher value of ductility demand as compared to the other two.
The 3 systems converge to the respective values of Ry near the higher values of natural period.
In all these cases for the Normalized permanent deformations, the BEL system as usual shows practically zero residual deformations, as it comes back to its original configuration as the motion ends.