```
% Austin Stoker 15 Oct 2013,
% This program integrates earthquake accleration data to find the velocity
% and displacement histories of the earth quake measurement
% It also takes the acceleration history and creates response spectra for
% 2%,5%,20% damping

close all
clearvars
clc
addpath('UseThisData');
load Northridge.mat
load ImperialValley.mat
load SanFernando.mat
load Kobe.mat
dtList=[.02,.01,.01,.01];
QuakeNames(1,:)='Northridge ';
QuakeNames(2,:)='Imperial   ';
QuakeNames(3,:)='SanFernando';
QuakeNames(4,:)='Kobe       ';
NumPeriods=1000;
MaxPeriod=30;%seconds
MaxA=0;
MinA=0;
MaxT=0;
DampingList=[.02,.05,.2];
DampingColors=['r','k','g'];
FilterOrder=20;

doPlot=true;
useFilteredSpectra=false;
doSavePlots=true;

PlotPrefix='Results';
SaveType='jpg';

numQuakes=size(QuakeNames,1);
numDampings=size(DampingList,2);

for k=1:numQuakes
    clearvars -except dtList QuakeNames NumPeriods MaxPeriod DampingList DampingCo
    switch k
        case 1
            curQuakeData=Northridge;
        case 2
            curQuakeData=ImperialValley;
        case 3
            curQuakeData=SanFernando;
        case 4
            curQuakeData=Kobe;
    end

    dt=dtList(k);

    numPoints=size(curQuakeData,2);
    t=0:dt:dt*(numPoints -1);

    figure(1)
    subplot(2,2,k)
    plot(t,curQuakeData)
    title(QuakeNames(k,:))
    BigT=dt*numPoints;

    m=[0 0 1 1 0 0];
```

```
f=[0 0.001 0.002 0.99 0.999 1]; %Change based on ratio with nyquist
b=fir2(FilterOrder,f,m); %forward and back filter, 100th order log approximati
                         %lower order has more rounded corners
amf=filtfilt(b,1,(curQuakeData-mean(curQuakeData)));%forward and back, -mean f
v=cumsum(amf)*dt; %cumulative sum is digital integration /50 for timestep or .
vmf=filtfilt(b,1,(v-mean(v)));
d=cumsum(vmf)*dt;
dmf=filtfilt(b,1,(d-mean(d)));
amfMax=max(abs(amf));
vmfMax=max(abs(vmf));
dmfMax=max(abs(dmf));
unfilteredMax=max(abs(curQuakeData));
BigA=max(curQuakeData);
LittleA=min(curQuakeData);

if doPlot==true;
    figure(k+1);
    set(figure(k+1), 'Position', [0 0 1200 1200])
    subplot(4,2,1)
    plot(t,dmf);
    title(strcat(QuakeNames(k,:),' Ground Displacement (Max=',num2str(dmfMax),
    xlabel('time (s)');
    ylabel('Displacement (cm)');

    subplot(4,2,3)
    plot(t,vmf);
    title(strcat(QuakeNames(k,:),' Ground Velocity (Max=',num2str(vmfMax),')')
    ylabel('Velocity (cm/s)');
    xlabel('time (s)');

    subplot(4,2,5)
    plot(t,amf);
    title(strcat(QuakeNames(k,:),' Ground Acceleration (Max=',num2str(amfMax),
    ylabel('Acceleration (cm/s^2)');
    xlabel('time (s)');

    subplot(4,2,7)
    plot(t,curQuakeData);
    title(strcat(QuakeNames(k,:),' Unfiltered Accleration (Max=',num2str(unfil
    ylabel('Acceleration (cm/s^2)');
    xlabel('time (s)');

end
if useFilteredSpectra==true
    acc=amf;
else
    acc=curQuakeData;
end
dt2=dt*dt;
Dmax=zeros(1,numDampings);
Vmax=zeros(1,numDampings);
Amax=zeros(1,numDampings);
PAmax=zeros(1,numDampings);
PVmax=zeros(1,numDampings);
for p=1:numDampings
    Damping=DampingList(p);
    Dmax=zeros(1,numDampings);
    Vmax=zeros(1,numDampings);
    Amax=zeros(1,numDampings);
    PAmax=zeros(1,numDampings);
    PVmax=zeros(1,numDampings);
    for i=2:NumPeriods
        P(i)=i*MaxPeriod/NumPeriods;
        w=2*pi/P(i);
        wd=w*Damping;
```

```
            Den=w*w/4+wd/dt+1/dt2;
            D=zeros(numPoints,1);
            V=zeros(numPoints,1);
            A=zeros(numPoints,1);
            Dmax(i)=0;
            Vmax(i)=0;
            Amax(i)=0;
            A(i-1)=acc(i-1);
            for n=2:numPoints
                D(n)=(acc(n)/4+(1/dt2+wd/dt)*D(n-1)+(1/dt+wd/2)*(V(n-1))+A(n-1)/4)
                A(n)=4*(D(n)-D(n-1))/dt2-4*V(n-1)/dt-A(n-1);
                V(n)=V(n-1)+dt*(A(n)+A(n-1))/2;
                if abs(D(n))>Dmax(i)
                    Dmax(i)=abs(D(n));
                end
                if abs(V(n))>Vmax(i)
                    Vmax(i)=abs(V(n));
                end
                if abs(A(n))>Amax(i)
                    Amax(i)=abs(A(n));
                end
            end
        PVmax(i)=Dmax(i)*w;
        PAmax(i)=Dmax(i)*w*w;
        end



        DmaxSS=Dmax(end);
        VmaxSS=Vmax(end);
        AmaxSS=Amax(end);

        if doPlot==true;
            subplot(4,2,2);
            plot(P,Dmax,DampingColors(p));
            hold on
            ylabel('Displacement (cm)');
            xlabel('Period T (s)');
            title(strcat(QuakeNames(k,:),' Displacement Spectra  (SS=',num2Str(Dma

            subplot(4,2,4);
            plot(P,Vmax,DampingColors(p));
            hold on
            ylabel('Velocity (cm/s)');
            xlabel('Period T (s)');
            title(strcat(QuakeNames(k,:),' Veloecity Spectra  (SS=',num2Str(VmaxSS

            subplot(4,2,6);
            plot(P,Amax,DampingColors(p));
            hold on
            ylabel('Acceleration (cm/s^2)');
            xlabel('Period T (s)');
            title(strcat(QuakeNames(k,:),' Acceleration Spectra  (SS=',num2Str(Ama

            subplot(4,2,8);
            plot(P,PAmax,DampingColors(p));
            hold on
            ylabel('Acceleration (cm/s^2)');
            xlabel('Period T (s)');
            title(strcat(QuakeNames(k,:),' Psuedo Acceleration Spectra'));
        end
    end
    DampingStrings(:,:)=num2str(DampingList');
    legend(DampingStrings);
    if doSavePlots==true
```

```
        saveas(figure(k+1),strcat(PlotPrefix,QuakeNames(k,:),'.',SaveType),SaveTyp
    end
    if BigT>MaxT
        MaxT=BigT;
    end
    if BigA>MaxA
        MaxA=BigA;
    end
    if LittleA<MinA
        MinA=LittleA;
    end
end
for j=1:numQuakes
    figure(1)
    subplot(2,2,j)
    xlim([0,MaxT]);
    ylim([MinA-50,MaxA+50]);
    xlabel('Time (s)');
    ylabel('Acceleration (cm/s^2)');

end
set(figure(1), 'Position', [0 0 1200 600])
if doSavePlots==true
    saveas(figure(1),strcat(PlotPrefix,'QuakeComparison','.',SaveType),SaveType)
end
```

Published with MATLAB® 7.10