حل معادله حرکت در فضای حالت
به کمک نرم‌افزار MATLAB

کاوه کرمی

استادیار سازه دانشگاه کردستان، دانشکده مهندسی، گروه عمران

E-mail: ka.karami@uok.ac.ir

الف- حوضه پیوسته زمانی

معادله دینامیک حرکت یک سیستم n درجه آزاد از رابطه زیر به دست می‌آید:

$$\mathbf{m}\ddot{\mathbf{x}}+\mathbf{c}\dot{\mathbf{x}}+\mathbf{k}\mathbf{x}=-\mathbf{m}\ell\ddot{\mathbf{x}}_{\mathbf{g}}+\mathbf{B}_{\mathbf{u}}\mathbf{u}_{\mathbf{c}} \tag{۱}$$

که در آن $\mathbf{m}$، $\mathbf{c}$ و $\mathbf{k}$ به ترتیب ماتریس‌های جرم، میرایی و سختی با ابعاد $n\times n$ می‌باشند. پاسخ‌های دینامیکی سازه شامل جابجایی، سرعت و شتاب به ترتیب با بردارهای $\mathbf{x}$، $\dot{\mathbf{x}}$ و $\ddot{\mathbf{x}}$ با ابعاد $n\times 1$ نشان داده شده است. سازه تحت اثر شتاب پایه $\ddot{\mathbf{x}}_{\mathbf{g}}$ با ابعاد 1×1 قرار می‌گیرد. بردار ℓ با ابعاد $n\times 1$ بردار تاثیر شتاب زمین در جرم‌های سازه است. همچنین نیروی کنترل $\mathbf{u}_{\mathbf{c}}$ با ابعاد $r\times 1$ نیز به سازه اعمال می‌شود. ماتریس $\mathbf{B}_{\mathbf{u}}$ با ابعاد $n\times r$ محل درجات آزادی که نیروهای کنترل در آن‌ها وارد می‌گردد را نشان می‌دهد. رابطه (۱) را می‌توان به صورت زیر نوشت:

$$\begin{aligned}\dot{\mathbf{x}}&=\dot{\mathbf{x}}\\ \ddot{\mathbf{x}}&=-\mathbf{m}^{-1}\mathbf{k}\mathbf{x}-\mathbf{m}^{-1}\mathbf{c}\dot{\mathbf{x}}-\ell\ddot{\mathbf{x}}_{\mathbf{g}}+\mathbf{m}^{-1}\mathbf{B}_{\mathbf{u}}\mathbf{u}_{\mathbf{c}}\end{aligned} \tag{۲}$$

فرم ماتریسی رابطه (۲) به صورت زیر در می‌آید:

$$\begin{Bmatrix}\dot{\mathbf{x}}\\ \ddot{\mathbf{x}}\end{Bmatrix}_{2n\times 1}=\begin{bmatrix}\mathbf{O}_{n\times n} & \mathbf{I}_{n\times n}\\ -\mathbf{m}^{-1}\mathbf{k} & -\mathbf{m}^{-1}\mathbf{c}\end{bmatrix}_{2n\times 2n}\begin{Bmatrix}\mathbf{x}\\ \dot{\mathbf{x}}\end{Bmatrix}_{2n\times 1}+\begin{Bmatrix}\mathbf{O}_{n\times 1}\\ -\ell\end{Bmatrix}_{2n\times 1}\ddot{\mathbf{x}}_{\mathbf{g}}+\begin{bmatrix}\mathbf{O}_{n\times r}\\ \mathbf{m}^{-1}\mathbf{B}_{\mathbf{u}}\end{bmatrix}_{2n\times r}\mathbf{u}_{\mathbf{c}} \tag{۳}$$

در رابطه بالا ماتریس‌های $\mathbf{O}$ و $\mathbf{I}$ به ترتیب ماتریس‌های صفر و واحد می‌باشند. فرض می‌شود که بردار نیروی کنترل $\mathbf{u}_{\mathbf{c}}$ به صورت زیر تعریف می‌گردد:

$$\mathbf{u_c} = \mathbf{G}\begin{Bmatrix}\mathbf{x}\\ \dot{\mathbf{x}}\end{Bmatrix}_{2n\times 1} = \begin{bmatrix}\mathbf{G_k} & \mathbf{G_c}\end{bmatrix}_{r\times 2n}\begin{Bmatrix}\mathbf{x}\\ \dot{\mathbf{x}}\end{Bmatrix}_{2n\times 1} \tag{۴}$$

که در آن ماتریس $\mathbf{G}$ با ابعاد $r\times 2n$ ماتریس بهره[1] نام دارد. ماتریس‌های $\mathbf{G_k}$ و $\mathbf{G_c}$ با ابعاد $r\times n$ از جنس ماتریس‌های سختی و میرایی می‌باشند. با جایگذاری رابطه (۴) در رابطه (۳) خواهیم داشت:

$$\begin{Bmatrix}\dot{\mathbf{x}}\\ \ddot{\mathbf{x}}\end{Bmatrix}_{2n\times 1} = \begin{bmatrix}\mathbf{O}_{n\times n} & \mathbf{I}_{n\times n}\\ -\mathbf{m}^{-1}\mathbf{k} & -\mathbf{m}^{-1}\mathbf{c}\end{bmatrix}_{2n\times 2n}\begin{Bmatrix}\mathbf{x}\\ \dot{\mathbf{x}}\end{Bmatrix}_{2n\times 1} + \begin{Bmatrix}\mathbf{O}_{n\times 1}\\ -\ell\end{Bmatrix}_{2n\times 1}\ddot{\mathbf{x}}_\mathbf{g} + \begin{bmatrix}\mathbf{O}_{n\times n} & \mathbf{O}_{n\times n}\\ \mathbf{m}^{-1}\mathbf{B_u}\mathbf{G_k} & \mathbf{m}^{-1}\mathbf{B_u}\mathbf{G_c}\end{bmatrix}\begin{Bmatrix}\mathbf{x}\\ \dot{\mathbf{x}}\end{Bmatrix}_{2n\times 1} \tag{۵}$$

در نتیجه می‌توان رابطه (۵) به صورت زیر ساده نوشت:

$$\begin{Bmatrix}\dot{\mathbf{x}}\\ \ddot{\mathbf{x}}\end{Bmatrix}_{2n\times 1} = \begin{bmatrix}\mathbf{O}_{n\times n} & \mathbf{I}_{n\times n}\\ -\mathbf{m}^{-1}(\mathbf{k}-\mathbf{B_u}\mathbf{G_k}) & -\mathbf{m}^{-1}(\mathbf{c}-\mathbf{B_u}\mathbf{G_c})\end{bmatrix}_{2n\times 2n}\begin{Bmatrix}\mathbf{x}\\ \dot{\mathbf{x}}\end{Bmatrix}_{2n\times 1} + \begin{Bmatrix}\mathbf{O}_{n\times 1}\\ -\ell\end{Bmatrix}_{2n\times 1}\ddot{\mathbf{x}}_\mathbf{g} \tag{۶}$$

با تعریف پارامترهای زیر

$$\mathbf{A_c} = \begin{bmatrix}\mathbf{O}_{n\times n} & \mathbf{I}_{n\times n}\\ -\mathbf{m}^{-1}(\mathbf{k}-\mathbf{B_u}\mathbf{G_k}) & -\mathbf{m}^{-1}(\mathbf{c}-\mathbf{B_u}\mathbf{G_c})\end{bmatrix}_{2n\times 2n} \qquad \mathbf{B_c} = \begin{Bmatrix}\mathbf{O}_{n\times 1}\\ -\ell\end{Bmatrix}_{2n\times 1} \qquad \mathbf{q} = \begin{Bmatrix}\mathbf{x}\\ \dot{\mathbf{x}}\end{Bmatrix}_{2n\times 1} \tag{۷}$$

می توان معادلات حرکت را به صورت زیر نوشت:

$$\begin{aligned}\dot{\mathbf{q}} &= \mathbf{A_c}\mathbf{q} + \mathbf{B_c}\ddot{\mathbf{x}}_\mathbf{g}\\ \mathbf{y} &= \mathbf{C}\mathbf{q} + \mathbf{D}\ddot{\mathbf{x}}_\mathbf{g}\end{aligned} \tag{۸}$$

معادله (۸) را معادله حرکت سیستم در فضای حالت[2] و در حوضه زمانی پیوسته[3] می‌نماند. یکی از ویژگی‌های مهم فضای حالت آن است که معادله دیفرانسیل خطی مرتبه دوم ناهمگن با ضرایب ثابت (رابطه ۱) را به یک معادله دیفرانسیل خطی مرتبه اول ناهمگن با ضرایب ثابت (رابطه ۸) تبدیل می‌کند. ماتریس $\mathbf{A_c}$ با ابعاد $2n\times 2n$ ماتریس سیستم نام دارد؛ زیرا شامل ویژگی‌های سیستم از جمله سختی و میرایی است. همچنین ماتریس‌های $\mathbf{B_c}$، $\mathbf{C}$ و $\mathbf{D}$ به ترتیب با ابعاد $2n\times 1$، $m\times 2n$ و $m\times 1$ ماتریس‌های تشکیل دهنده فضای حالت است. بردار $\mathbf{y}$ با ابعاد $m\times 1$ خروجی سیستم را نشان می‌دهد. با توجه به نوع خروجی سیستم ماتریس‌های $\mathbf{C}$ و $\mathbf{D}$ به صورت زیر تشکیل می‌گردد:

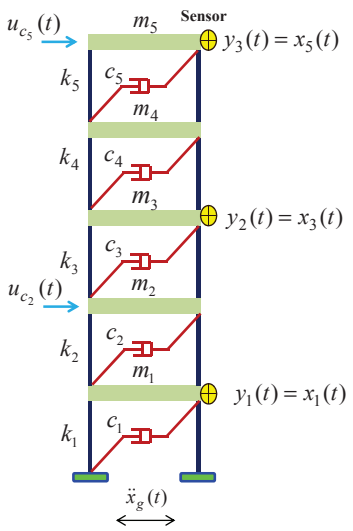
خروجی شتاب	خروجی سرعت	خروجی جابجایی	
$\mathbf{C} = \begin{bmatrix}-\mathbf{C_S}\mathbf{m}^{-1}(\mathbf{k}-\mathbf{B_u}\mathbf{G_k}) & -\mathbf{C_S}\mathbf{m}^{-1}(\mathbf{c}-\mathbf{B_u}\mathbf{G_c})\end{bmatrix}$ $\mathbf{D} = -\mathbf{C_S}\ell$	$\mathbf{C} = \begin{bmatrix}\mathbf{O}_{m\times n} & \mathbf{C_S}\end{bmatrix}$ $\mathbf{D} = \mathbf{O}_{m\times 1}$	$\mathbf{C} = \begin{bmatrix}\mathbf{C_S} & \mathbf{O}_{m\times n}\end{bmatrix}$ $\mathbf{D} = \mathbf{O}_{m\times 1}$	(۹)

[1] Gain Matrix
[2] State Space
[3] Continues Time Domain

ماتریس $\mathbf{C_S}$ با ابعاد $m \times n$ محل نصب حسگرها را مشخص می‌کند. معادله حرکت به فرم معادله (۸) را می‌توان با استفاده از برنامه 'DynamicResponseAgUcG.m' در محیط MATLAB حل کرد. برای این منظور مثال زیر در نظر گرفته می‌شود.

مثال: شکل (۱) یک ساختمان پنج طبقه را نشان می‌دهد؛ که مشخصات سازه‌ای آن در جدول (۱) آمده است. در این ساختمان از سه عدد حسگر برای اندازه‌گیری خروجی‌های مربوط به جابجایی طبقات اول، سوم و پنجم استفاده شده است. همچنین در طبقات دوم و پنجم نیروهای کنترلی به سازه اعمال می‌گردد. ضریب بهره مربوط به نیروهای کنترل در طبقات دوم و سوم به صورت زیر است.

$$\mathbf{G_k} = \begin{bmatrix} 0 & -0.1k_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.15k_5 \end{bmatrix}_{2\times5} \qquad \mathbf{G_c} = \begin{bmatrix} 0 & -c_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -c_5 \end{bmatrix}_{2\times5}$$



شکل ۱. ساختمان پنج طبقه

جدول ۱

پارامترهای سازه‌ای ساختمان پنج طبقه

St. No	Mass (ton)	Stiffness (kN/m)
1	12	**22000**
2	12	**20000**
3	12	**17800**
4	11	**16000**
5	10	**14300**

$\xi = 5\%$

ورودی‌های این برنامه که کاربر باید آن را وارد نماید شامل موارد زیر است:

```
%%%
```

است که با استفاده از دستور diag به صورت قطری تشکیل می‌گردد. k0 بردار سختی طبقات است. xi ضریب میرایی سیستم را مشخص می‌کند. st0 برابر با گام زمانی شتاب نگاشت مورد استفاده است. اگر بخواهیم از گام زمانی کوچکتر یا بزرگتری استفاده کنیم توسط st تعیین می‌گردد. mv و rv به ترتیب بردار مکان خروجی‌ها (محل نصب حسگرها) و مکان ورودی‌ها (محل اعمال نیروی‌های کنترل) در درجات آزادی مربوطه را نشان می‌دهد.

مدت زمان زلزله T ، بردار زمان t و تعداد طبقات n نیز با استفاده از دستور زیر محاسبه می‌شود:

```
T=(length(Xg)-1)*st0;
t=0:st:T;
n=length (m);
```

در دستور زیر به کمک تابع XG درون‌یابی انجام گرفته و شتاب زلزله در هر گام زمانی st تعیین می‌گردد؛ و نهایتا در شتاب $g=9.807\ m/s^2$ نیز ضرب شده است.

```
xg(1:((T/st)+1))=0;
for i=1:length(xg)
    xg(i)=XG(Xg,st0,(i-1)*st)*9.807;
end
```

با استفاده از تابع‌های StiffnessMatrix و DampingMatrix به ترتیب ماتریس‌های سختی و میرایی تشکیل می‌شود.

```
k=StiffnessMatrix(k0);
c=DampingMatrix(m ,k,xi);
```

تابع StiffnessMatrix ماتریس سختی را بر اساس رابطه زیر تشکیل می‌دهد:

$$\begin{cases} \mathrm{K}_{(i,i)} = k_i + k_{i+1} & i \neq n_m \\ \mathrm{K}_{(i,i)} = k_i & i = n_m \\ \mathrm{K}_{(i,j)} = -k_{Max(i,j)} & |i-j| = 1 \\ \mathrm{K}_{(i,j)} = 0 & |i-j| > 1 \end{cases} \qquad (۱۰)$$

که در آن $\mathrm{K}_{(i,j)}$ نشان دهنده (i,j)اُمین درایه ماتریس سختی می‌باشد. همچنین k_i سختی طبقه iاُم است. تابع DampingMatrix نیز ماتریس میرایی را براساس رابطه زیر تشکیل می‌دهد:

$$\begin{aligned} &\mathbf{M} = \mathbf{\Phi}^T \mathbf{m} \mathbf{\Phi} \\ &\mathbf{\Phi}^T \mathbf{c} \mathbf{\Phi} = 2\xi \boldsymbol{\omega} \mathbf{M} \quad \Rightarrow \quad \mathbf{c} = 2\xi \left(\mathbf{\Phi}^T\right)^{-1} \boldsymbol{\omega} \mathbf{M} \mathbf{\Phi}^{-1} \end{aligned} \qquad (۱۱)$$

که در رابطه بالا، $\mathbf{\Phi}$ و $\boldsymbol{\omega}$ به ترتیب ماتریس اشکال مودی و ماتریس فرکانس سیستم می‌باشند. پارامتر ξ ضریب میرایی سیستم است. ماتریس‌های Gk و Gc نیز باید کاربر آن را مشخص نماید.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Gk=zeros(length(rv),n);
Gk(1,2)=-0.1*k0(2);
Gk(2,5)=-0.15*k0(5);
Gc=zeros(length(rv),n);
Gc(1,2)=-1*(-c(1,2));
Gc(2,5)=-1*c(5,5);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

تابع StateSpaceAgUcG ماتریس‌های $\mathbf{A_c}$، $\mathbf{B_c}$، $\mathbf{C}$ و $\mathbf{D}$ را با توجه به نوع خروجی out تشکیل می‌دهد.

```
[Ac,Bc,C,D]=StateSpaceAgUcG(m,k,c,Gk,Gc,rv,mv,out);
```

دستور SS سیستم sysc را در فضای حالت و در فضای پیوسته زمانی تشکیل می‌دهد

```
sysc=ss(Ac,Bc,C,D);
```

به طوری که

Ac=sysc.a
Bc=sysc.b
C=sysc.c
D=sysc.d

دستور زیر شرایط اولیه (بردار حالت اولیه که شامل جابجایی اولیه و سرعت اولیه است) مسئله را تعیین می‌کند:

```
q0(1:2*n,1)=0;
```

با استفاده از دستور زیر معادلات حرکت سیستم در فضای حالت محاسبه می‌گردد:

```
[y]=lsim(sysc,xg,t,q0);
```

دستور lsim تاریخچه زمانی پاسخ دینامیکی یک سیستم خطی را تحت اثر یک ورودی دلخواه شبیه سازه می‌کند. خروجی این دستور همان y یا بردار خروجی سیستم است. برای مقایسه سازه کنترل شده با سازه کنترل نشده باید پاسخ سازه را در حالتی که نیروی کنترل صفر است (حالت بدون کنترل) y0 نیز تعیین کرد.

```
Gk0=zeros(length(rv),n);
Gc0=zeros(length(rv),n);
[Ac0,Bc0,C0,D0]=StateSpaceAgUcG(m,k,c,Gk0,Gc0,rv,mv,out);
sysc0=ss(Ac0,Bc0,C0,D0);
[y0]=lsim(sysc0,xg,t,q0);
```

پاسخ سازه در حالتی که خروجی جابجایی باشد با استفاده از دستور زیر رسم می‌گردد:

```
if strcmp(out,'dis')==1
    LB=min(min(y0*100));
    UB=max(max(y0*100));
    for i=1:length(mv)
        subplot(length(mv),1,i),plot(t,y0(:,length(mv)-i+1)*100,':k')
        hold on
        plot(t,y(:,length(mv)-i+1)*100,'k','linewidth',1.5)
        if i==length(mv)
            xlabel('time [sec]','FontSize',8)
            legend('Uncontrolled','Controlled','orientation','horizontal')
        end
        ylabel('dis[cm]','FontSize',8)
        axis([0 T LB UB])
    end
end
```

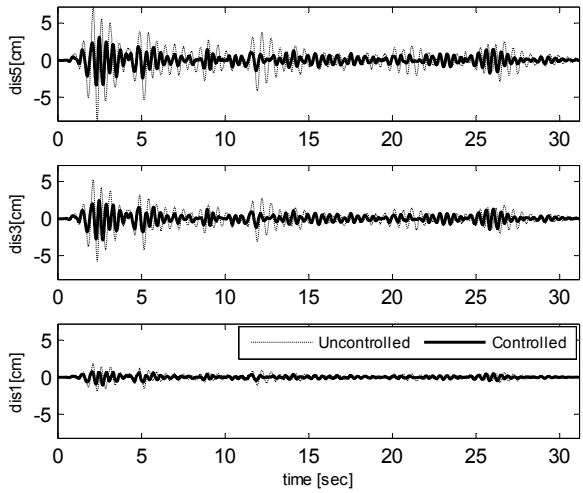
پاسخ سازه در حالتی که خروجی سرعت باشد با استفاده از دستور زیر رسم می‌گردد:

```
if strcmp(out,'vel')==1
    LB=min(min(y0*100));
    UB=max(max(y0*100));
    for i=1:length(mv)
        subplot(length(mv),1,i),plot(t,y0(:,length(mv)-i+1)*100,':k')
        hold on
        plot(t,y(:,length(mv)-i+1)*100,'k','linewidth',1.5)
        if i==length(mv)
            xlabel('time [sec]','FontSize',8)
            legend('Uncontrolled','Controlled','orientation','horizontal')
        end
        ylabel('vel[cm/s]','FontSize',8)
        axis([0 T LB UB])
    end
end
```

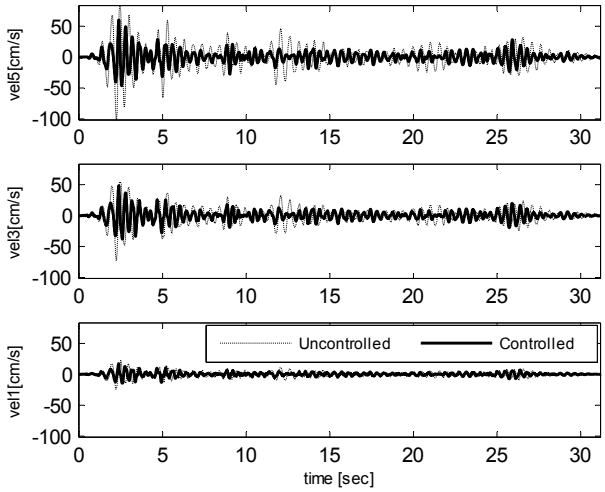
پاسخ سازه در حالتی که خروجی شتاب باشد با استفاده از دستور زیر رسم می‌گردد:

```
if strcmp(out,'acc')==1
    LB=min(min(y0/9.807));
    UB=max(max(y0/9.807));
    for i=1:length(mv)
        subplot(length(mv),1,i),plot(t,y0(:,length(mv)-i+1)/9.807,':k')
        hold on
        plot(t,y(:,length(mv)-i+1)/9.807,'k','linewidth',1.5)
        if i==length(mv)
            xlabel('time [sec]','FontSize',8)
            legend('Uncontrolled','Controlled','orientation','horizontal')
        end
        ylabel('acc[g]','FontSize',8)
        axis([0 T LB UB])
    end
end
```

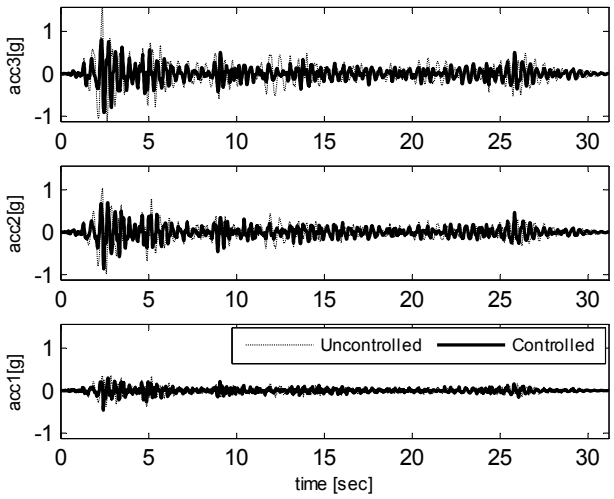
پاسخ‌های سازه به صورت زیر می باشد:



شکل۲. تاریخچه زمانی پاسخ جابجایی در طبقات اول، سوم و پنجم در ازه پنج طبقه تحت اثر زلزله El-Centro.



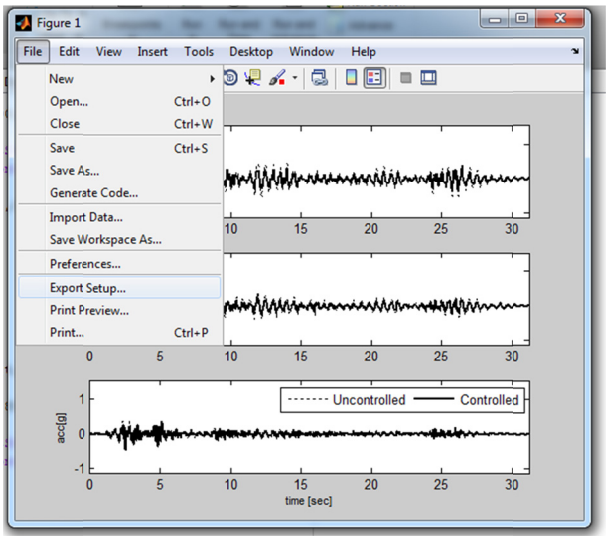
شکل۳. تاریخچه زمانی پاسخ سرعت در طبقات اول، سوم و پنجم در ازه پنج طبقه تحت اثر زلزله El-Centro.



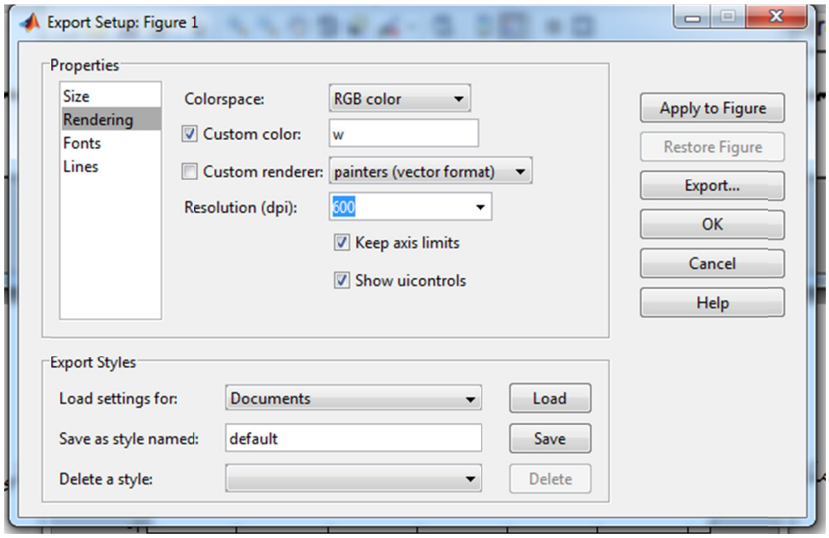
شکل۴. تاریخچه زمانی پاسخ شتاب در طبقات اول، سوم و پنجم در ازه پنج طبقه تحت اثر زلزله El-Centro.

برای ذخیره نمودن شکل‌های خروجی Matlab با کیفیت بالا از روش زیر عمل می‌کنیم:

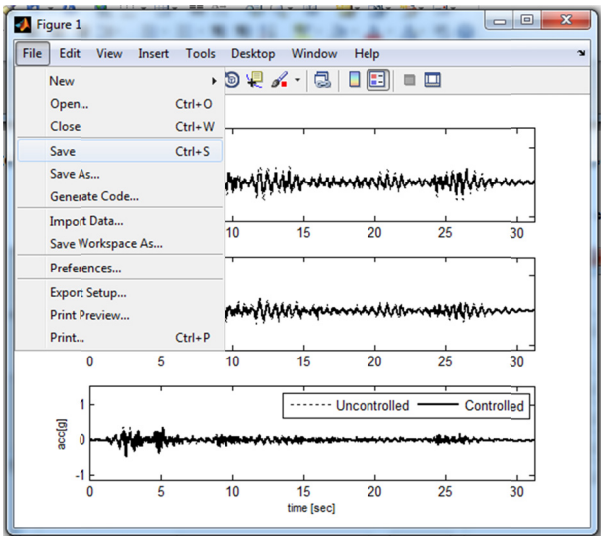
از منوی File گزینه Export Setup انتخاب شود.



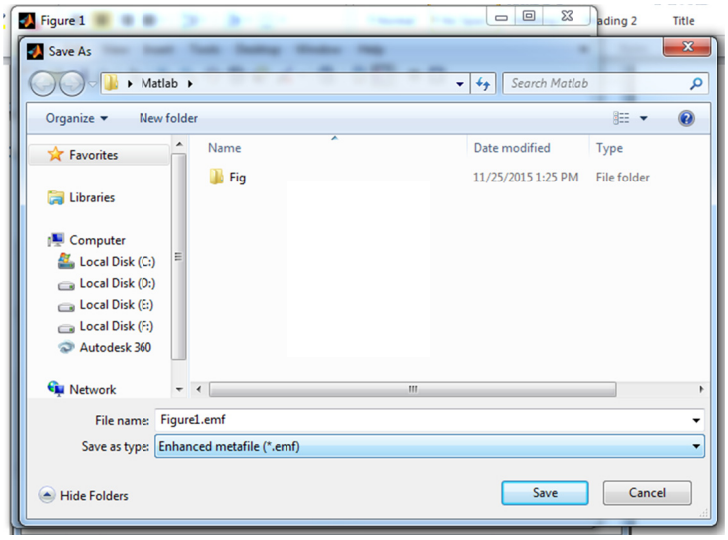
سپس Rendering انتخاب شده و Resolution (dpi) را بر روی 600 تنظیم می‌کنیم. سپس Apply to Figure را کلیک کرده و با زدن دکمه OK از پنجره خارج می‌شویم.



از منوی File گزینه Save انتخاب شود.



در قسمت Save as type فرمت Enhanced metafile (*.emf) انتخاب گردد و دکمه Save را کلیک نمایید.



ب- حوضه گسسته زمانی

معادله (۸) در فضای گسسته زمانی به صورت زیر است

$$\begin{aligned} \mathbf{q}(i+1) &= \mathbf{A}\mathbf{q}(i) + \mathbf{B}\ddot{\mathbf{x}}_{\mathbf{g}}(i) \\ \mathbf{y}(i) &= \mathbf{C}\mathbf{q}(i) + \mathbf{D}\ddot{\mathbf{x}}_{\mathbf{g}}(i) \end{aligned} \qquad (۱۲)$$

اندیس i شماره گام زمانی را نشان می‌دهد. ماتریس‌های $\mathbf{A}$ و $\mathbf{B}$ به ترتیب ماتریس سیستم و ماتریس مکان ورودی در حوضه گسسته زمانی می‌باشند و از رابطه زیر به دست می‌آیند:

$$\mathbf{A} = e^{\mathbf{A}_c \cdot \Delta t} = \sum_{i=0}^{\infty} \left(\frac{\mathbf{A}_c^{\,i} \cdot \Delta t^{i}}{i!}\right)$$
$$\mathbf{B} = \int_0^{\Delta t} e^{\mathbf{A}_c \cdot \Delta \tau} d\tau \, \mathbf{B}_c = \sum_{i=0}^{\infty} \left(\frac{\mathbf{A}_c^{\,i} \cdot \Delta t^{i+1}}{(i+1)i!}\right) \mathbf{B}_c \qquad (۱۳)$$

در رابطه بالا Δt گام زمانی است و برابر با گام زمانی شتاب نگاشت زلزله می‌باشد.

معادله حرکت به فرم معادله (۱۲) را می‌توان با استفاده از برنامه 'DynamicResponseAgUcGd.m' در محیط MATLAB حل کرد. برای این منظور همان مثال قبلی در نظر گرفته می‌شود. مراحل کار تا تعریف سیستم در فضای حالت در حوضه پیوسته زمانی مشابه برنامه 'DynamicResponseAgUcG.m' است. در ادامه با استفاده از دستور c2d سیستم از حوضه پیوسته زمانی sysc تبدیل به سیستم در حوضه گسسته زمانی sysd می‌شود (رابطه ۱۳).

```
sysd=c2d(sysc,st);
```

به طوری که

```
A=sysd.a;
B=sysd.b;
```

ابعاد بردار حالت و خروجی به صورت زیر تعریف می‌شود

```
q(1:2*n,1:length(xg))=0;
y(1:length(mv),1:length(xg))=0;
```

با استفاده از دستور زیر معادلات حرکت در فضای حالت و در حوضه زمانی پیوسته (رابطه ۱۲) برای هر گام زمانی محاسبه شده و پاسخ‌های سیستم محاسبه می‌گردد.

```
for i=1:length(xg)
    if i<length(xg)
        q(:,i+1)=A*q(:,i)+B*xg(i);
    end
    y(:,i)=C*q(:,i)+D*xg(i);
end

y=y';
```

در اینجا نیز مشابه با حالت قبل، برای مقایسه سازه کنترل شده با سازه کنترل نشده باید پاسخ سازه را در حالتی که نیروی کنترل صفر است (حالت بدون کنترل) y0 نیز تعیین کرد.

```
Gk0=zeros(length(rv),n);
Gc0=zeros(length(rv),n);
```

```
[Ac0,Bc0,C0,D0]=StateSpaceAgUcG(m,k,c,Gk0,Gc0,rv,mv,out);
sysc0=ss(Ac0,Bc0,C0,D0);
sysd0=c2d(sysc0,st);
A0=sysd0.a;
B0=sysd0.b;


q0(1:2*n,1:length(xg))=0;
y0(1:length(mv),1:length(xg))=0;
for i=1:length(xg)
    if i<length(xg)
        q0(:,i+1)=A0*q0(:,i)+B0*xg(i);
    end
    y0(:,i)=C*q0(:,i)+D*xg(i);
end

y0=y0';
```

به طور مشابه با توجه به نوع خروجی می توان پاسخ‌های سیستم را نیز رسم نمود.