

حل معادله حرکت در فضای حالت

به کمک نرم افزار 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 می آید:

$$\mathbf{m}\ddot{\mathbf{x}} + \mathbf{c}\dot{\mathbf{x}} + \mathbf{k}\mathbf{x} = -\mathbf{m}\ell\ddot{\mathbf{x}}_g + \mathbf{\Gamma}\mathbf{B}_u\mathbf{u}_c \quad (1)$$

که در آن \mathbf{m} ، \mathbf{c} و \mathbf{k} به ترتیب ماتریس های جرم، میرایی و سختی با ابعاد $n \times n$ می باشند. پاسخ های 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 $\ddot{\mathbf{x}}_g$ با ابعاد 1×1 قرار می گیرد. بردار ℓ با ابعاد $n \times 1$ بردار تاثیر شتاب زمین در جرم های سازه است. همچنین نیروی کنترل \mathbf{u}_c با ابعاد $r \times 1$ نیز به سازه اعمال می شود. ماتریس \mathbf{B}_u با ابعاد $n \times r$ محل درجات آزادی که نیروهای کنترل در آنها وارد می گردد را نشان می دهد. ماتریس $\mathbf{\Gamma}$ با ابعاد $n \times n$ اندرکنش نیروهای کنترل در طبقات است. رابطه (۱) را می توان به صورت زیر نوشت:

$$\begin{aligned} \dot{\mathbf{x}} &= \dot{\mathbf{x}} \\ \ddot{\mathbf{x}} &= -\mathbf{m}^{-1}\mathbf{k}\mathbf{x} - \mathbf{m}^{-1}\mathbf{c}\dot{\mathbf{x}} - \ell\ddot{\mathbf{x}}_g + \mathbf{m}^{-1}\mathbf{\Gamma}\mathbf{B}_u\mathbf{u}_c \end{aligned} \quad \mathbf{\Gamma} = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \cdots & 0 \\ 0 & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -1 \end{bmatrix} \quad (2)$$

فرم ماتریسی رابطه (۲) به صورت زیر در می آید:

$$\begin{Bmatrix} \dot{\mathbf{x}} \\ \ddot{\mathbf{x}} \end{Bmatrix}_{2n \times 1} = \begin{bmatrix} \mathbf{O}_{n \times n} & \mathbf{I}_{n \times n} \\ -\mathbf{m}^{-1}\mathbf{k} & -\mathbf{m}^{-1}\mathbf{c} \end{bmatrix}_{2n \times 2n} \begin{Bmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \end{Bmatrix}_{2n \times 1} + \begin{bmatrix} \mathbf{O}_{n \times 1} & \mathbf{O}_{n \times r} \\ -\ell & \mathbf{m}^{-1}\mathbf{\Gamma}\mathbf{B}_u \end{bmatrix}_{2n \times (r+1)} \begin{Bmatrix} \ddot{\mathbf{x}}_g \\ \mathbf{u}_c \end{Bmatrix}_{(r+1) \times 1} \quad (3)$$

در رابطه بالا ماتریس‌های \mathbf{O} و \mathbf{I} به ترتیب ماتریس‌های صفر و واحد می‌باشند. با تعریف پارامترهای زیر

$$\mathbf{A}_c = \begin{bmatrix} \mathbf{O}_{n \times n} & \mathbf{I}_{n \times n} \\ -\mathbf{m}^{-1}\mathbf{k} & -\mathbf{m}^{-1}\mathbf{c} \end{bmatrix}_{2n \times 2n} \quad \mathbf{B}_c = \begin{bmatrix} \mathbf{O}_{n \times 1} & \mathbf{O}_{n \times r} \\ -\ell & \mathbf{m}^{-1}\Gamma\mathbf{B}_u \end{bmatrix}_{2n \times (r+1)} \quad \mathbf{q} = \begin{Bmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \end{Bmatrix}_{2n \times 1} \quad \mathbf{u} = \begin{Bmatrix} \ddot{\mathbf{x}}_g \\ \mathbf{u}_c \end{Bmatrix}_{(r+1) \times 1} \quad (4)$$

می‌توان معادلات حرکت را به صورت زیر نوشت:

$$\begin{aligned} \dot{\mathbf{q}} &= \mathbf{A}_c \mathbf{q} + \mathbf{B}_c \mathbf{u} \\ \mathbf{y} &= \mathbf{C} \mathbf{q} + \mathbf{D} \mathbf{u} \end{aligned} \quad (5)$$

معادله (۵) را معادله حرکت سیستم در فضای حالت^۱ و در حوضه زمانی پیوسته^۲ می‌نماید. 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 ۵) 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 $2n \times (r+1)$ ، $m \times 2n$ و $m \times (r+1)$ 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
$\mathbf{C} = \begin{bmatrix} -\mathbf{C}_s \mathbf{m}^{-1} \mathbf{k} & -\mathbf{C}_s \mathbf{m}^{-1} \mathbf{c} \end{bmatrix}$ $\mathbf{D} = \begin{bmatrix} -\mathbf{C}_s \ell & \mathbf{C}_s \mathbf{m}^{-1} \Gamma \mathbf{B}_u \end{bmatrix}$	$\mathbf{C} = [\mathbf{O}_{m \times n} \quad \mathbf{C}_s]$ $\mathbf{D} = \mathbf{O}_{m \times (r+1)}$	$\mathbf{C} = [\mathbf{C}_s \quad \mathbf{O}_{m \times n}]$ $\mathbf{D} = \mathbf{O}_{m \times (r+1)}$

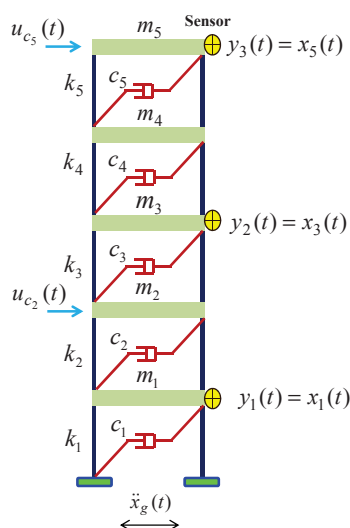
(۶)

ماتریس \mathbf{C}_s با ابعاد $m \times n$ محل نصب حسگرها را مشخص می‌کند. معادله حرکت به فرم معادله (۵) را می‌توان با استفاده از برنامه 'DynamicResponseAgUc.m' در محیط MATLAB حل کرد. برای این منظور مثال زیر در نظر گرفته می‌شود.

¹ State Space

² Continues Time Domain

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. نیروی کنترل در طبقات برابر با شتاب زلزله ضرب در جرم آن طبقه است.



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:

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Xg=xlsread('ElCentro.xls','Accelerograph','B3:B1562');
out='acc';
m=diag([12 12 12 11 10])*1e3;
k0=[22e3 20e3 17.8e3 16e3 14.3e3]*1e3;
xi=0.05;
st0=0.02;
st=0.01;
mv=[1 3 5];
rv=[2 5];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

که در آن X_g شتاب زلزله El-Centro می‌باشد که معمولاً بر حسب $g=9.807 \text{ m/s}^2$ مقیاس شده است و توسط دستور `xlsread` از فایل Excel تحت عنوان 'ElCentro.xls' خوانده می‌شود. Out نوع خروجی سیستم را نشان می‌دهد که می‌توان از 'dis'، 'vel' و 'acc' به ترتیب برای نمایش خروجی‌های جابجایی، سرعت و شتاب سیستم استفاده کرد. m ماتریس جرم طبقات است که با استفاده از دستور `diag` به صورت قطری تشکیل می‌گردد. k_0 بردار سختی طبقات است. ξ ضریب میرایی سیستم را مشخص می‌کند. st_0 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 نهایتاً در شتاب $g=9.807 \text{ m/s}^2$ 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:

$$\begin{cases} K_{(i,i)} = k_i + k_{i+1} & i \neq n_m \\ K_{(i,i)} = k_i & i = n_m \\ K_{(i,j)} = -k_{\text{Max}(i,j)} & |i-j|=1 \\ K_{(i,j)} = 0 & |i-j| > 1 \end{cases} \quad (7)$$

که در آن $K_{(i,j)}$ نشان دهنده (i,j) آمین درایه ماتریس سختی می‌باشد. همچنین k_i سختی طبقه i ام است. تابع DampingMatrix نیز ماتریس میرایی را براساس رابطه زیر تشکیل می‌دهد:

$$\begin{aligned} \mathbf{M} &= \Phi^T \mathbf{m} \Phi \\ \Phi^T \mathbf{c} \Phi &= 2\xi \omega \mathbf{M} \Rightarrow \mathbf{c} = 2\xi (\Phi^T)^{-1} \omega \mathbf{M} \Phi^{-1} \end{aligned} \quad (8)$$

که در رابطه بالا، Φ و ω به ترتیب ماتریس اشکال مودی و ماتریس فرکانس سیستم می‌باشند. پارامتر ξ ضریب میرایی سیستم است. نیروهای کنترل نیز به صورت زیر تعریف می‌شود.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
uc(1:length(rv),1:length(xg))=0;
for i=1:length(rv)
    uc(i,:)=m(rv(i),rv(i))*xg;
end
```

%%%

تابع StateSpaceAgUc ماتریس‌های A_c ، B_c ، C و D را با توجه به نوع خروجی out تشکیل می‌دهد.

```
[Ac,Bc,C,D]=StateSpaceAgUc(m,k,c,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 ماتریس u تشکیل و معادلات حرکت سیستم در فضای حالت محاسبه می‌گردد:

```
u(1,:)=xg;  
u(2:length(rv)+1,:)=uc;  
[y]=lsim(sysc,u,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.

```
u(2:length(rv)+1,:)=0;  
[y0]=lsim(sysc,u,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
```

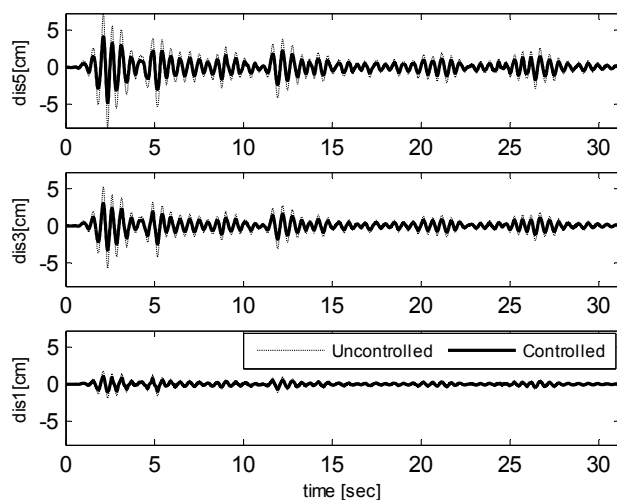
پاسخ سازه در حالتی که خروجی سرعت باشد با استفاده از دستور زیر رسم می‌گردد:

```
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
```

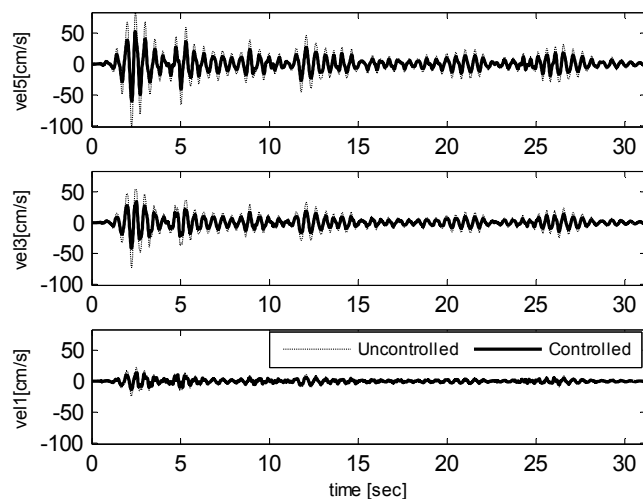
پاسخ سازه در حالتی که خروجی شتاب باشد با استفاده از دستور زیر رسم می‌گردد:

```
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
```

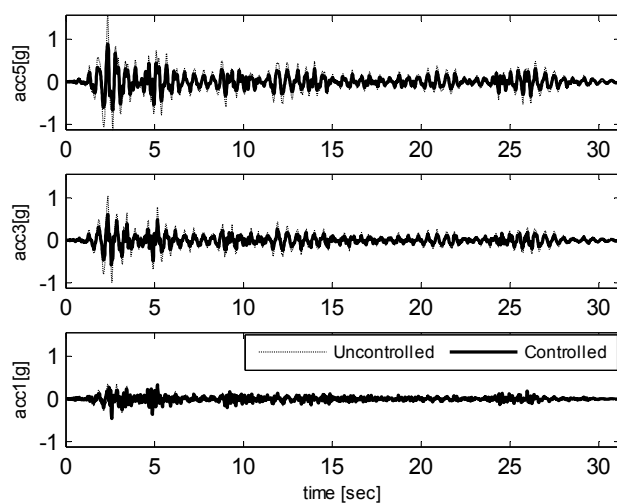
پاسخ‌های سازه به صورت زیر می باشد:



شکل ۲. تاریخچه زمانی پاسخ جابجایی در طبقات اول، سوم و پنجم در ازه پنج طبقه تحت اثر زلزله El-Centro.

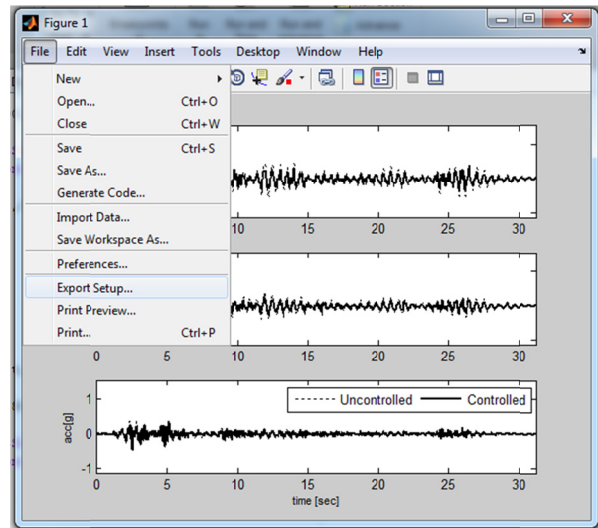


شکل ۳. تاریخچه زمانی پاسخ سرعت در طبقات اول، سوم و پنجم در ازه پنج طبقه تحت اثر زلزله El-Centro.

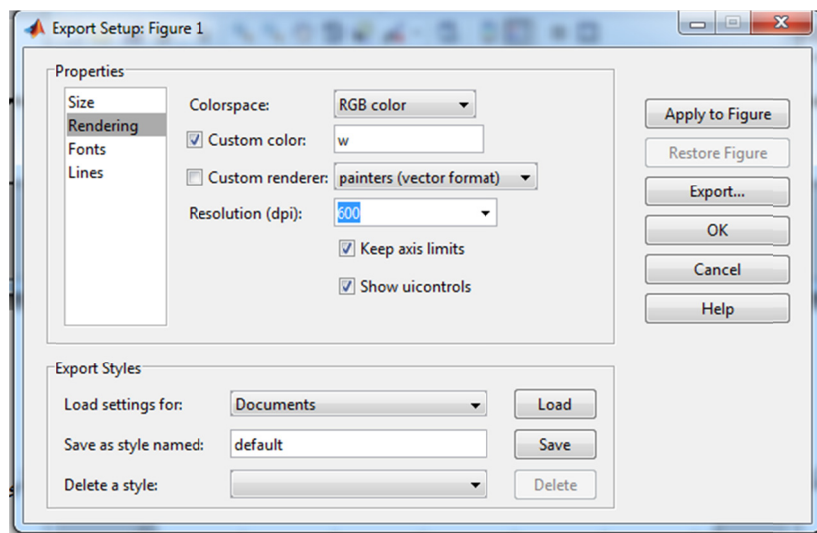


شکل ۴. تاریخچه زمانی پاسخ شتاب در طبقات اول، سوم و پنجم در ازه پنج طبقه تحت اثر زلزله El-Centro.

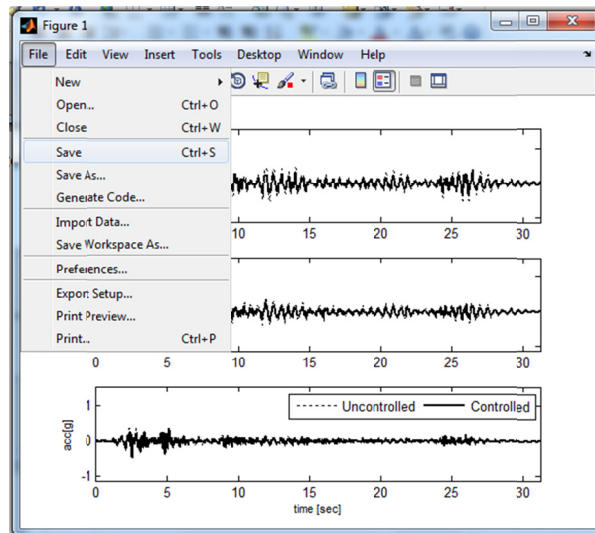
برای ذخیره نمودن شکل‌های خروجی Matlab با کیفیت بالا از روش زیر عمل می‌کنیم:
از منوی File گزینه Export Setup انتخاب شود.



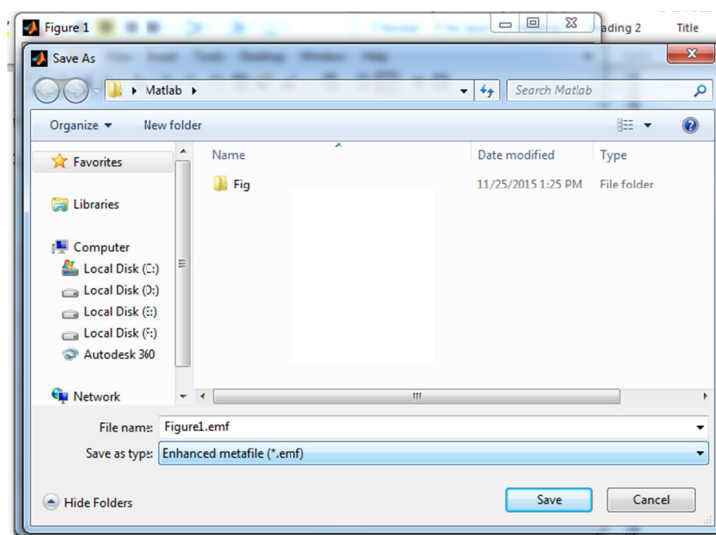
سپس Rendering انتخاب شده و Resolution (dpi) را بر روی 600 تنظیم می‌کنیم. سپس Apply to Figure را کلیک کرده و با زدن دکمه OK از پنجره خارج می‌شویم.



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 (۵) در فضای گسسته زمانی به صورت زیر است

$$\begin{aligned} \mathbf{q}(i+1) &= \mathbf{A}\mathbf{q}(i) + \mathbf{B}\mathbf{u}(i) \\ \mathbf{y}(i) &= \mathbf{C}\mathbf{q}(i) + \mathbf{D}\mathbf{u}(i) \end{aligned} \quad (9)$$

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

$$\begin{aligned} A &= e^{A_c \cdot \Delta t} = \sum_{i=0}^{\infty} \left(\frac{A_c^i \cdot \Delta t^i}{i!} \right) \\ B &= \int_0^{\Delta t} e^{A_c \cdot \Delta \tau} d\tau B_c = \sum_{i=0}^{\infty} \left(\frac{A_c^i \cdot \Delta t^{i+1}}{(i+1)i!} \right) B_c \end{aligned} \quad (10)$$

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

معادله حرکت به فرم معادله (۹) را می‌توان با استفاده از برنامه 'DynamicResponseAgUcd.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 'DynamicResponseAgUcd.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 ۱۰).

```
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*u(:,i);
    end
    y(:,i)=C*q(:,i)+D*u(:,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) y_0 نیز تعیین کرد.

```
u(2:length(rv)+1,:) = 0;
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) = A*q0(:, i) + B*u(:, i);
    end
    y0(:, i) = C*q0(:, i) + D*u(:, i);
end

y0 = y0';
```

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