MATLAB 기초

-동역학시뮬레이션

전북대학교 전자공학부
이태희

동역학 시스템의 구현

- 미분방정식을 상태방정식으로 변환

$$\ddot{y}(t) - \mu\left(1 - y^2(t)\right)\dot{y}(t) + y(t) = 0$$

Let $y(t) = y_1(t)$ and $\dot{y}(t) = y_2(t)$

$$\begin{cases} \dot{y}_1(t) = y_2(t) \\ \dot{y}_2(t) = \mu\left(1 - {y_1}^2(t)\right)\dot{y}_1(t) - y_1(t) \end{cases}$$

$$\begin{bmatrix} \dot{y}_1(t) \\ \dot{y}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & \mu\left(1 - {y_1}^2(t)\right) \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix}$$

동역학 시스템의 구현

vanderpol.m*

```
1   function dydt=vanderpol(t,y)
2
3 —   mu=5;
4 —   dydt=[y(2); mu*(1-y(1)^2)*y(2)-y(1)];
5
```

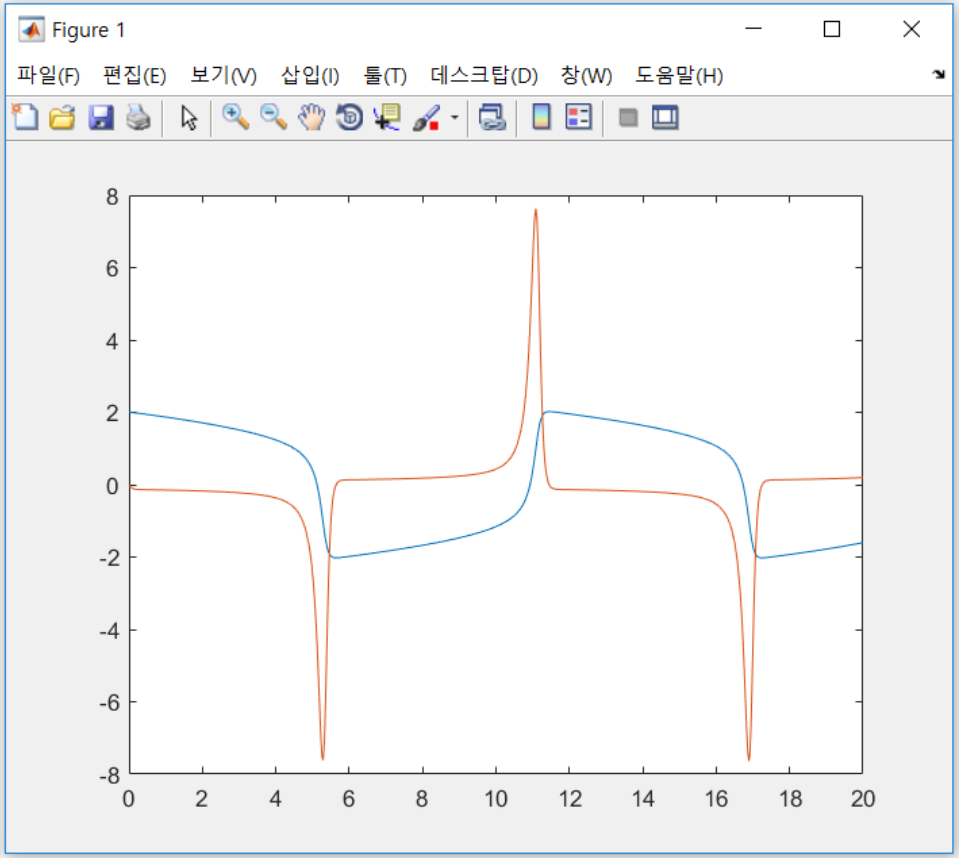
vanderpol.m

```
1   function dydt=vanderpol(t,y)
2
3 —   mu=5;
4 —   dydt=[0 1 ; -1 mu*(1-y(1)^2)]*y;
```

구현된 방정식의 실행

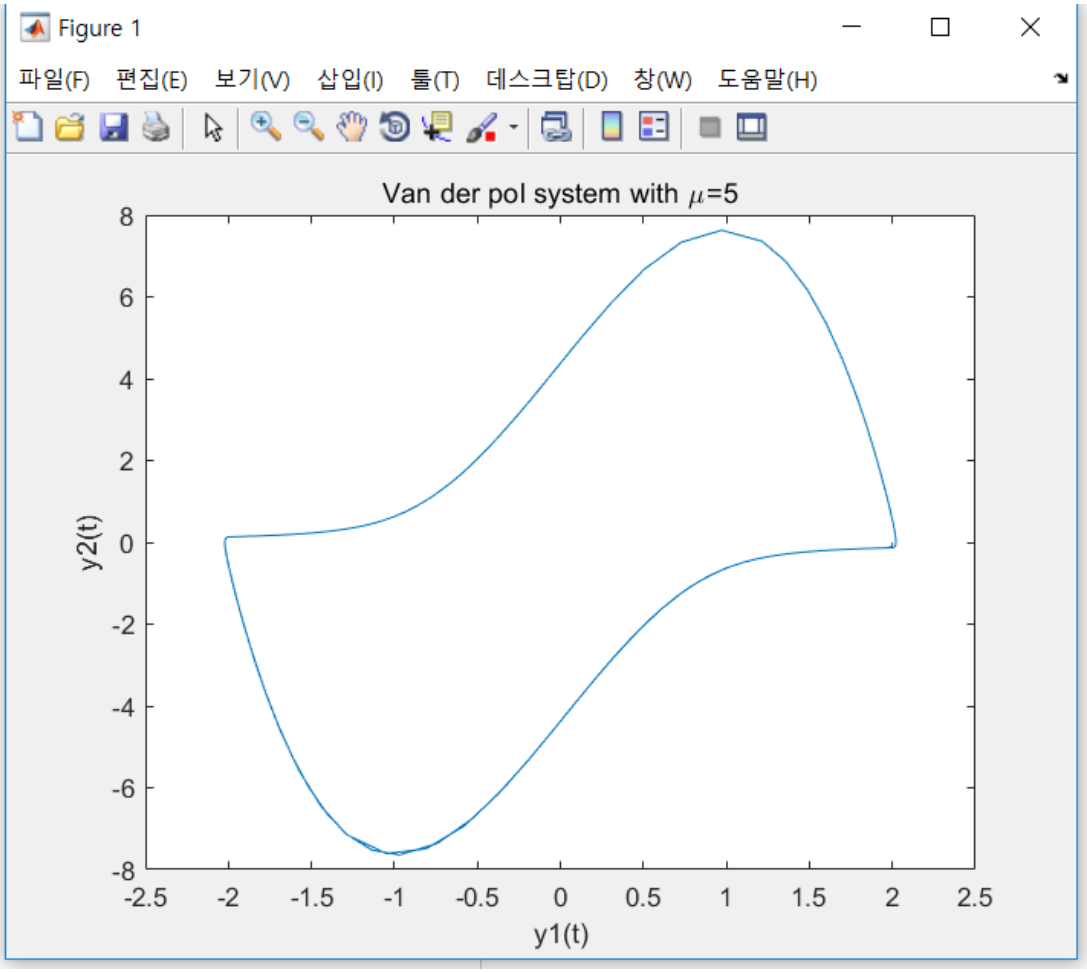
- [t, y]=ode45(@동역학시스템함수명, [시간구간], [초기값])

```
>> [t,y]=ode45(@vanderpol,[0,20],[2,0]);
>> plot(t,y)
>>
```

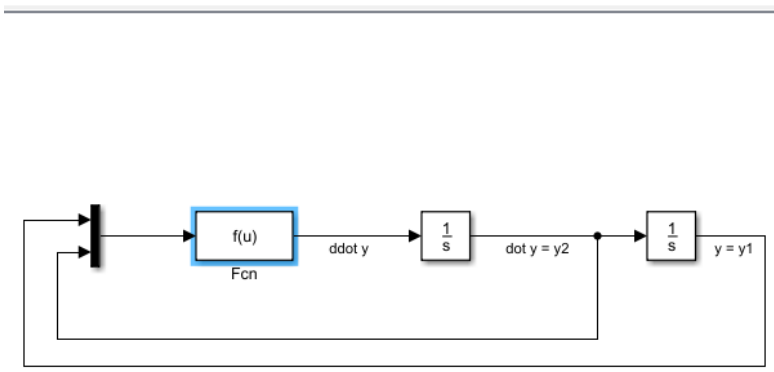


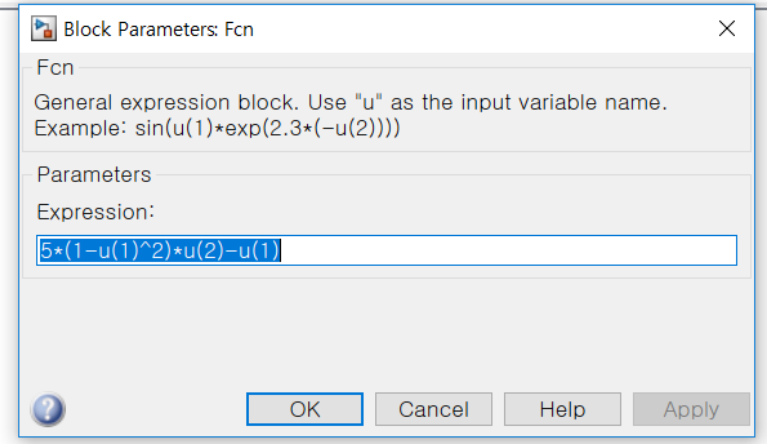
구현된 방정식의 실행

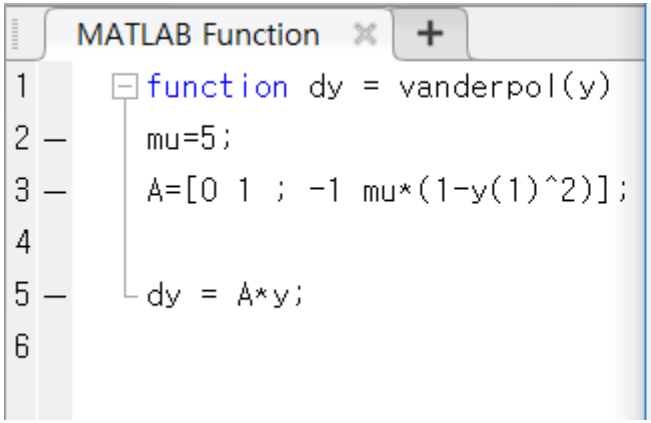
```
myode=@vanderpol; %함수형 변수
tspan=[0, 20]; %시간구간
yinitial=[2, 0]; %초기조건
[t, y]=ode45(myode,tspan,yinitial);
figure;
plot(y(:,1),y(:,2));
xlabel('y1(t)');
ylabel('y2(t)');
title('Van der pol system with ₩mu=5')
```

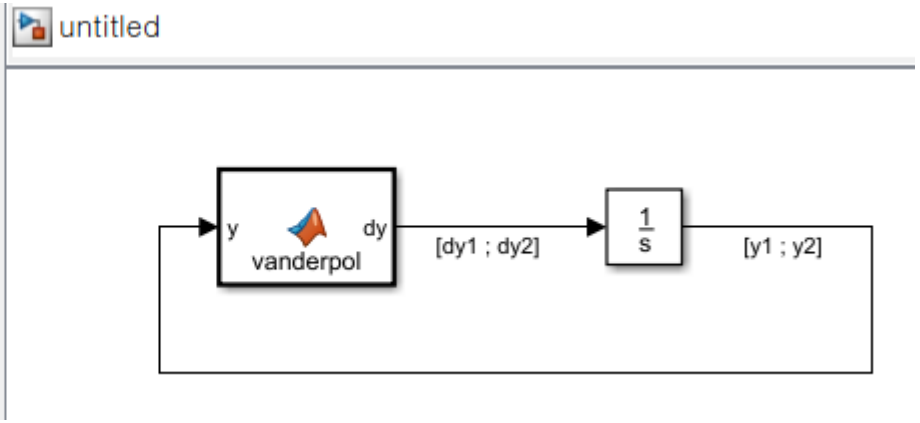


Simulink로의 구현









연습

- 다음 동역학시스템의 0~50초까지의 파형을 그리시오.

$$\dot{x}_1(t) = a(x_2(t) - x_1(t))$$

$$\dot{x}_2(t) = bx_1(t) - x_1(t)x_3(t) - x_2(t)$$

$$\dot{x}_3(t) = x_1(t)x_2(t) - cx_3(t)$$

여기서 a=10, b=28, c=8/3 이다.

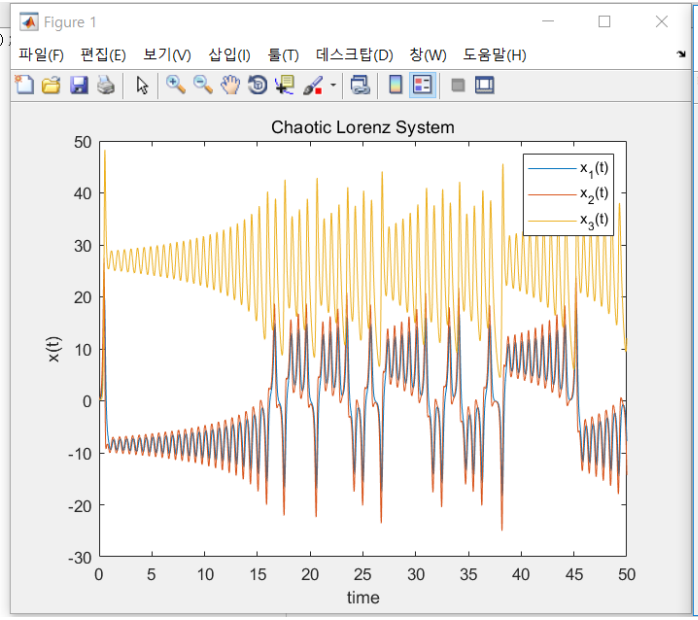
연습

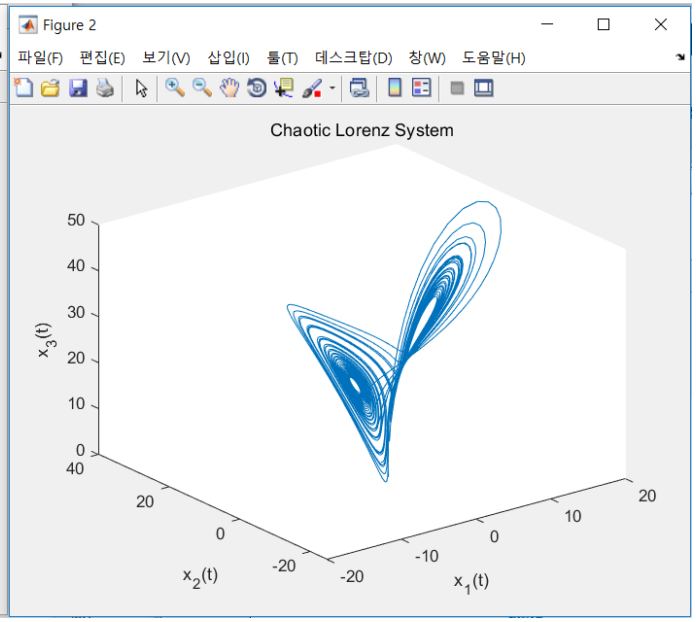
Lorenz.m* | vanderpol.m | U

```
1   function dx=Lorenz(t,x)
2   a=10; b=28; c=8/3;
3   x1=x(1);
4   x2=x(2);
5   x3=x(3);
6
7   dx1=a*(x2-x1);
8   dx2=b*x1-x1*x3-x2;
9   dx3=x1*x2-c*x3;
10  dx=[dx1;dx2;dx3];
```

Lorenz.m | vanderpol.m | Untitled4* | +

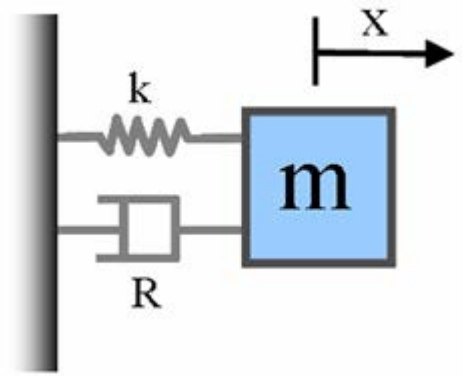
```
1   [t, x]=ode45(@Lorenz,[0, 50],[0.1 0.2 0.3]);
2
3   figure;
4   plot(t,x);
5   xlabel('time');
6   ylabel('x(t)');
7   legend('x_1(t)','x_2(t)','x_3(t)')
8   title('Chaotic Lorenz System')
9   figure;
10  plot3(x(:,1),x(:,2),x(:,3))
11  xlabel('x_1(t)');
12  ylabel('x_2(t)');
13  zlabel('x_3(t)');
14  title('Chaotic Lorenz System')
15
```





연습

- 다음의 spring-damper 시스템에서 m=10 kg, k=10 N/m, R=1 kg/s이고, 변위 x의 초기값이 10일 때 시간에 관한 x의 파형을 그리시오.



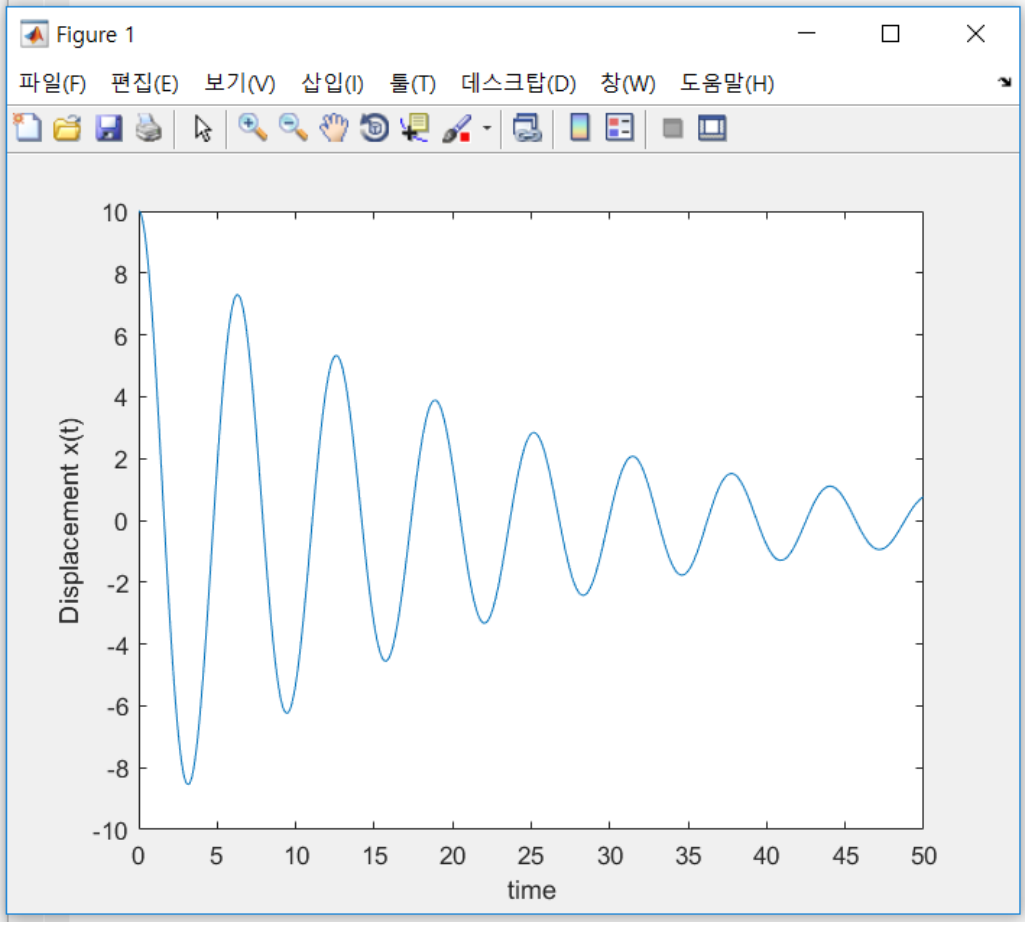
$$m\ddot{x}(t) + R\dot{x}(t) + kx(t) = 0$$

연습

springdamper.m

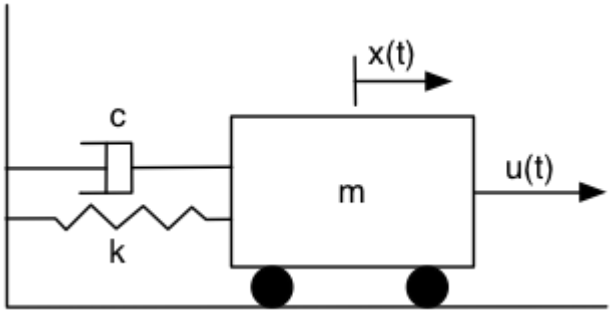
```
1   function dx=springdamper(t,x)
2
3   m=10;
4   k=10;
5   R=1;
6
7   x1=x(1);
8   x2=x(2);
9
10  dx1=x2;
11  dx2=-k/m*x1-R/m*x2;
12  dx=[dx1;dx2];
13
```

```
1   tspan=0:0.01:50;
2   initial=[10 0];
3   [t,x] = ode45(@springdamper,tspan,initial);
4   figure(1);
5   plot(t,x(:,1));
6   xlabel('time');
7   ylabel('Displacement x(t)');
```



외부입력이 존재하는 동역학 시스템

- Spring-Damper with external input $u(t)$



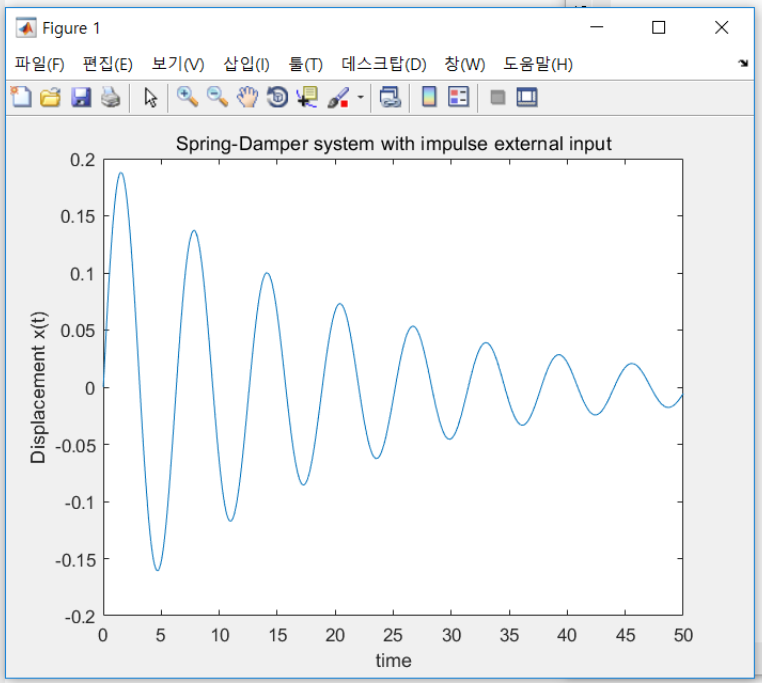
$$m\ddot{x}(t) + R\dot{x}(t) + kx(t) = u(t)$$

외부입력이 존재하는 동역학 시스템

springdamper.m | springdamper_fulfill.m | my_springdamper.m

```
function dx=my_springdamper(t,x,ex)
m=ex{1};
k=ex{2};
R=ex{3};
sf=ex{4};

x1=x(1);
x2=x(2);

dx1=x2;
dx2=-k/m*x1-R/m*x2+sf(t,10);
dx=[dx1;dx2];
```

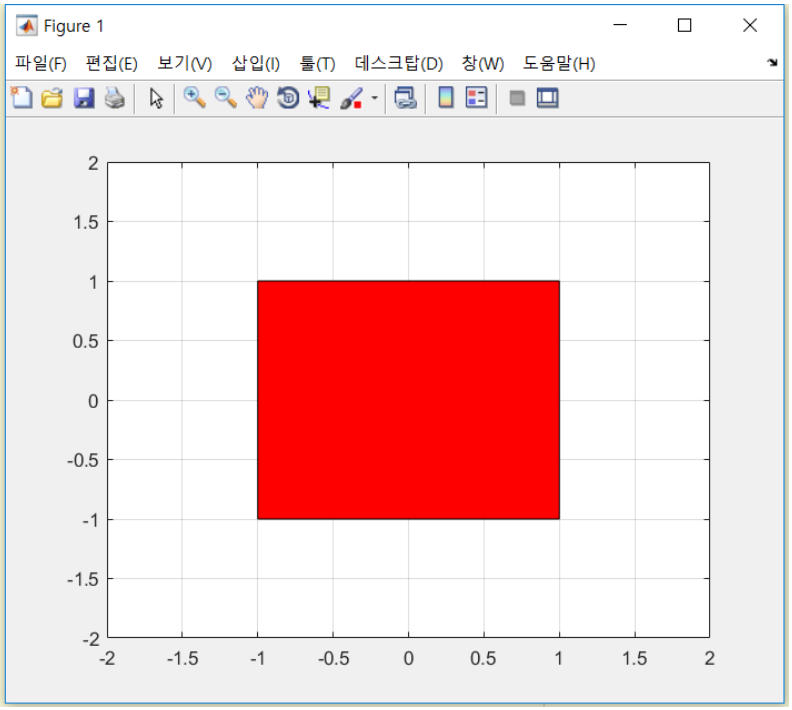
```
function mysd()
st=0.01;
final=50;
tspan=0:st:final;
initial=[0 0];

m=10;
k=10;
R=1;
f=@(t,power) power*heaviside(t)-power*heaviside(t-2*st);
ex={m,k,R,f};
[t,x] = ode45(@my_springdamper,tspan,initial,[],ex);
plot(t,x(:,1));
xlabel('time');
ylabel('Displacement x(t)');
title('Spring-Damper system with impulse external input')
end
```



그림그리기

- fill(x좌표, y좌표, '색상') : x,y를 꼭지점으로 갖는 도형에 주어진 색상으로 채운다.
- properties(클래스명) : 클래스에 대한 속성을 나타내어준다.

```
%% Square
x=[-1 -1 1 1];
y=[-1 1 1 -1];
box=fill(x,y,'r');
grid on;
axis([-2 2 -2 2]);
```

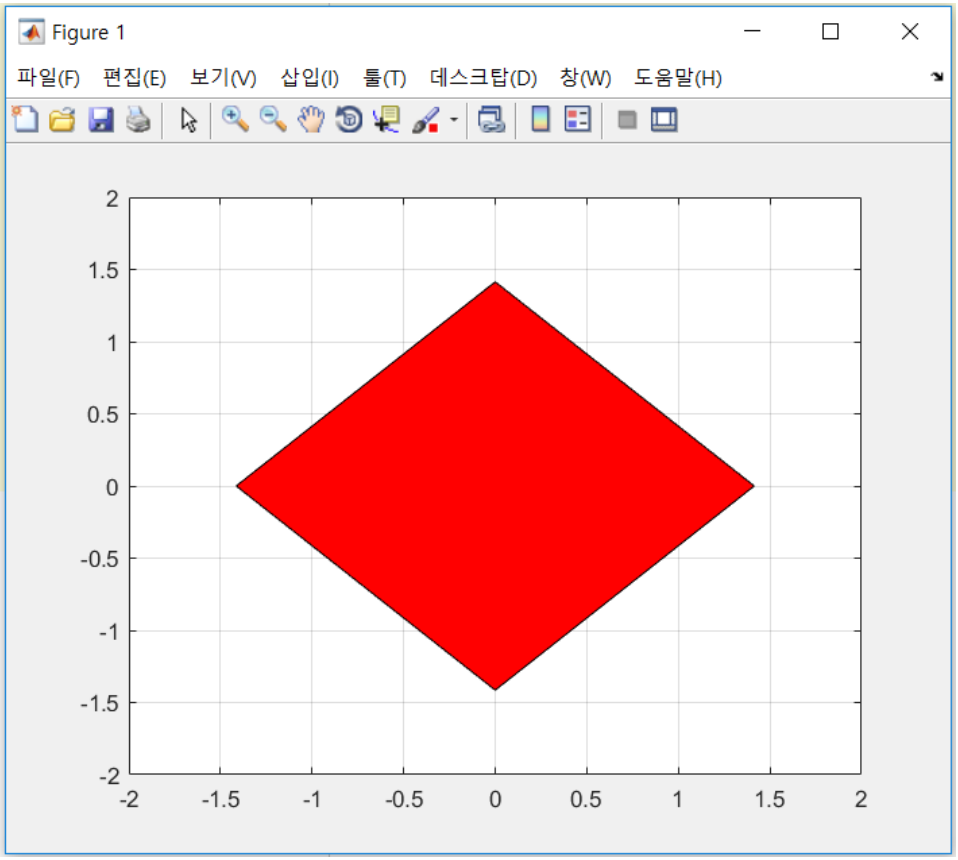



```
>> properties(box)

matlab.graphics.primitive.Patch 클래스에 대한 속성:

    CData
    CDataMapping
    FaceVertexCData
    EdgeColor
    FaceColor
    Faces
    LineStyle
    LineWidth
    VertexNormals
    VertexNormalsMode
    FaceNormals
    FaceNormalsMode
    AmbientStrength
    DiffuseStrength
    SpecularStrength
    SpecularExponent
    SpecularColorReflectance
    Marker
    MarkerEdgeColor
    MarkerFaceColor
    MarkerSize
    Vertices
    Clipping
    AlignVertexCenters
    AlphaDataMapping
    FaceVertexAlphaData
    EdgeAlpha
    FaceAlpha
    XData
    YData
    ZData
    FaceLighting
    EdgeLighting
    BackFaceLighting
    UIContextMenu
    ButtonDownFcn
    BusyAction
    BeingDeleted
    Interruptible
    CreateFcn
    DeleteFcn
    Type
    Tag
    UserData
    Selected
    SelectionHighlight
    HitTest
    PickableParts
    DisplayName
    Annotation
    Children
    Parent
    Visible
    HandleVisibility
```

그림그리기

- line(x값, y값, ‘Color’, ’k’, ‘LineWidth’, 10) : (x,y)로 선을 그린다.
- set(클래스명, 속성, 새로운 데이터,…) :
  저장된 클래스에서 해당하는 속성에 새로운 데이터를 넣어 재실행한다.
- 앞서 그린 사각형을 45도 돌려보자.

```
%% Rotate Square
x=[-1 -1 1 1];
y=[-1 1 1 -1];
box=fill(x,y,'r');
grid on;
axis([-2 2 -2 2]);

t=0.25;
ts=t*pi;
Rotation=[cos(ts), -sin(ts) ; sin(ts) cos(ts)];
next_Position=Rotation*[x ; y];
next_x=next_Position(1,:);
next_y=next_Position(2,:);
set(box,'Xdata',next_x,'Ydata',next_y);
```



애니메이션

- 사각형을 반시계 방향으로 한 바퀴 회전시키는 애니메이션 코드:

```
%% Animation
x=[-1 -1 1 1];
y=[-1 1 1 -1];
box=fill(x,y,'r');
grid on;
hold on;
origin=text(x(1,1),y(1,1),'o');
axis([-2 2 -2 2]);

for t=0:0.01:2*pi % 1바퀴 회전. 즉ㅡ,0도에서 2 pi 까지
    Rotation=[cos(t), -sin(t) ; sin(t) cos(t)]; % 회전행렬
    next_Position=Rotation*[x ; y]; % t도 돌았을때의 꼭지점의 위치
    next_x=next_Position(1,:);
    next_y=next_Position(2,:);
    set(box,'Xdata',next_x,'Ydata',next_y);
    % box 클래스에서 Xdata속성에 해당하는 값에 next_x 값을, Ydata속성에 해당하는 값에 next_y값을 넣어 실행
    set(origin,'Position',[next_x(1,1) ,next_y(1,1)]);
    % origin 클래스에서 Position속성에 해당하는 값에 새로운 값을 넣어 실행
    drawnow; % i가 증가함과 동시에 실시간으로 그림을 그림
    for i=1:5000000 % 위에 for 루프문의 속도를 느리게 하기위해 삽입
        temp=i;
    end
end
```

연습

- Spring-damper 시스템의 동역학 반응을 애니메이션으로 보여라.

```
initial=[1 0];
x0=initial(1);
boxsize=2;
box_x=[x0-boxsize/2 x0-boxsize/2 x0+boxsize/2 x0+boxsize/2];
box_y=[0 boxsize boxsize 0];
box=fill(box_x,box_y,'r');
hold on

horizon_x=[-15 -15 15 15];
horizon_y=[0 -0.1 -0.1 0];
horizon=fill(horizon_x,horizon_y,'black')

vertical_x=[-15 -15 -10 -10];
vertical_y=[4 0 0 4];
vertical=fill(vertical_x,vertical_y,'green')

axis([-15 15 -2 20]);
grid on;

tspan=0:0.01:100;
[t,x] = ode45(@springdamper,tspan,initial);

for i=1:max(size(t))
    next_x=[x(i)-boxsize/2 x(i)-boxsize/2 x(i)+boxsize/2 x(i)+boxsize/2];
    set(box,'Xdata',next_x);
    drawnow;
    for j=1:50000
        temp=j;
    end
end
```