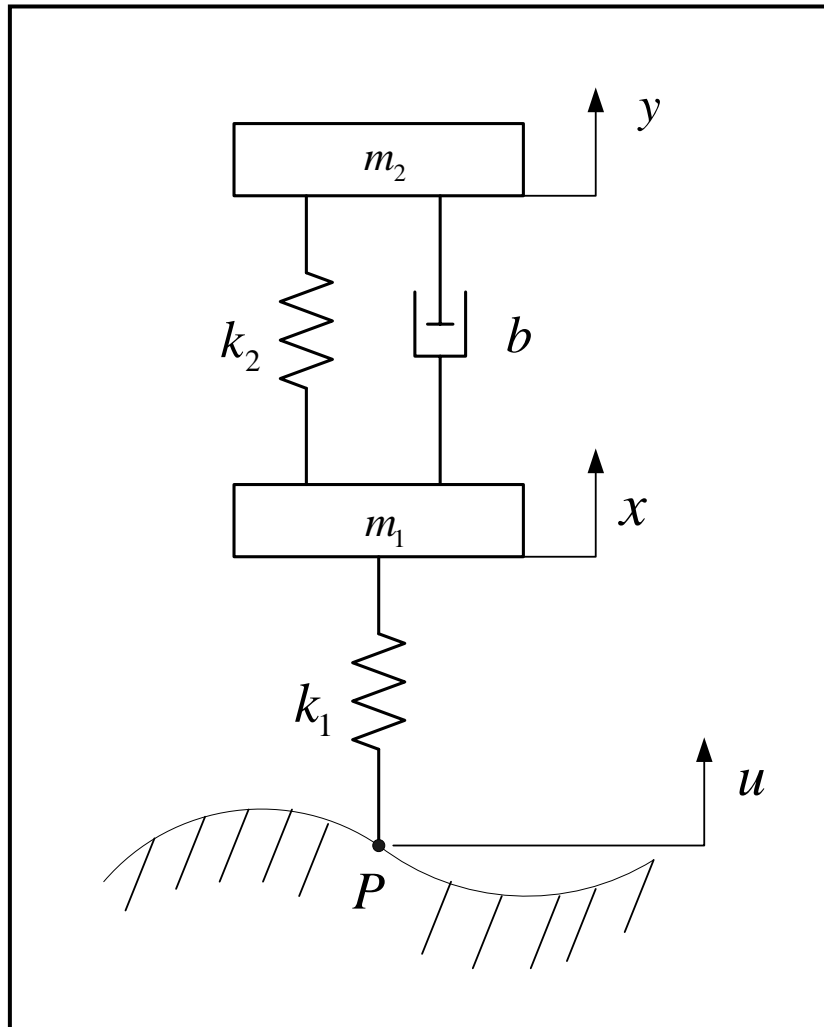


MATLAB 실습 2

Suspension

Vehicle Dynamics &
Control Laboratory

1. Vehicle Suspension Problem



▪ Design Considerations

1. Ride Quality

→ *Sprung mass acceleration* : \ddot{y}

2. Rattle space

→ *Suspension Deflection* : $y - x$

3. Tire Force Vibration

→ *Tire Deflection* : $x - u$

▪ Suspension Design Parameters

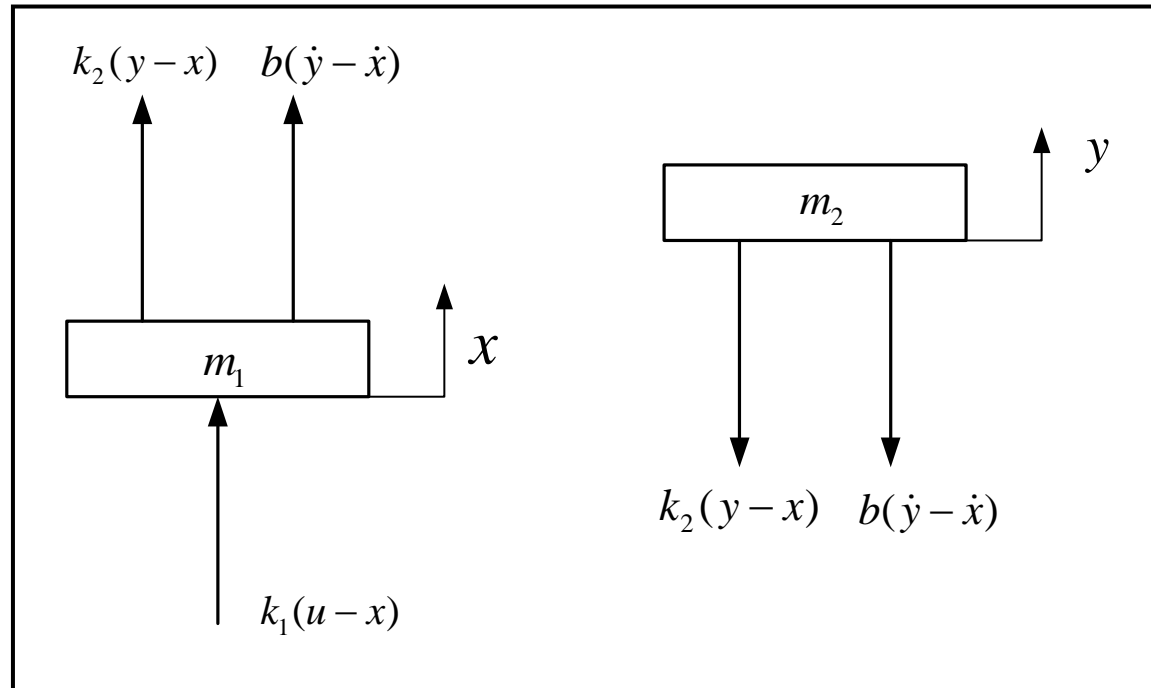
→ *Spring Stiffness* : k_2

→ *Damping Ratio* : b

→ *Tire Stiffness* : k_1

1. Vehicle Suspension Problem : Dynamic Equations

▪ Free Body Diagram



▪ Dynamic Equations

$$m_1 \ddot{x} = k_2(y - x) + b(\dot{y} - \dot{x}) + k_1(u - x)$$

$$m_2 \ddot{y} = -k_2(y - x) - b(\dot{y} - \dot{x})$$

1. Vehicle Suspension Problem : Transfer Functions

▪ Laplace Transform

$$[m_1 s^2 + bs + (k_1 + k_2)]X(s) = (bs + k_2)Y(s) + k_1 U(s)$$

$$[m_2 s^2 + bs + k_2]Y(s) = (bs + k_2)X(s)$$

▪ Displacement of Mass

$$\frac{Y(s)}{U(s)} = \frac{k_1 (bs + k_2)}{m_1 m_2 s^4 + (m_1 + m_2)bs^3 + [(k_2 m_1 + (k_1 + k_2)m_2)s^2 + k_1 bs + k_1 k_2]}$$

$$\frac{X(s)}{U(s)} = \frac{k_1 (m_2 s^2 + bs + k_2)}{m_1 m_2 s^4 + (m_1 + m_2)bs^3 + [(k_2 m_1 + (k_1 + k_2)m_2)s^2 + k_1 bs + k_1 k_2]}$$

▪ Design Considerations

$$G_1(s) = \frac{s^2 Y(s)}{U(s)} = \frac{s^2 k_1 (bs + k_2)}{m_1 m_2 s^4 + (m_1 + m_2)bs^3 + [(k_2 m_1 + (k_1 + k_2)m_2)s^2 + k_1 bs + k_1 k_2]} \rightarrow \text{Sprung mass acceleration : } \ddot{y}$$

$$G_2(s) = \frac{Y(s) - X(s)}{U(s)} = \frac{-k_1 m_2 s^2}{m_1 m_2 s^4 + (m_1 + m_2)bs^3 + [(k_2 m_1 + (k_1 + k_2)m_2)s^2 + k_1 bs + k_1 k_2]} \rightarrow \text{Suspension Deflection : } y - x$$

$$G_3(s) = \frac{X(s) - U(s)}{U(s)} = \frac{-m_1 m_2 s^4 - (m_1 + m_2)bs^3 - k_2 (m_1 + m_2)s^2}{m_1 m_2 s^4 + (m_1 + m_2)bs^3 + [(k_2 m_1 + (k_1 + k_2)m_2)s^2 + k_1 bs + k_1 k_2]} \rightarrow \text{Tire Deflection : } x - u$$

1. Vehicle Suspension Problem : State Equation(1)

- General Form of State Equation

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

- The State variables ($x = z_u, \quad y = z_s$)

$$x_1 = z_s - z_u \quad : \text{Suspension Deflection}$$

$$x_2 = \dot{z}_s \quad : \text{absolute velocity of sprung mass}$$

$$x_3 = z_u - u \quad : \text{Tire Deflection}$$

$$x_4 = \dot{z}_u \quad : \text{absolute velocity of unsprung mass}$$

- Dynamic Equations

$$m_1 \ddot{z}_u = k_2(z_s - z_u) + b(\dot{z}_s - \dot{z}_u) + k_1(u - z_u)$$

$$m_2 \ddot{z}_s = -k_2(z_s - z_u) - b(\dot{z}_s - \dot{z}_u)$$

- 1st order State equations

$$\dot{x}_1 = \dot{z}_s - \dot{z}_u = x_2 - x_4$$

$$\dot{x}_2 = -\frac{k_2}{m_2}(z_s - z_u) - \frac{b}{m_2}(\dot{z}_s - \dot{z}_u) = -\frac{k_2}{m_2}x_1 - \frac{b}{m_2}x_2 + \frac{b}{m_2}x_4$$

$$\dot{x}_3 = \dot{z}_u - \dot{u} = x_4 - \dot{u}$$

$$\dot{x}_4 = \frac{k_2}{m_1}(z_s - z_u) + \frac{b}{m_1}(\dot{z}_s - \dot{z}_u) - \frac{k_1}{m_1}(z_u - u) = \frac{k_2}{m_1}x_1 + \frac{b}{m_1}x_2 - \frac{k_1}{m_1}x_3 - \frac{b}{m_1}x_4$$

1. Vehicle Suspension Problem : State Equation(2)

- Matrix Form of State equations (system matrix)

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & -1 \\ -\frac{k_2}{m_2} & -\frac{b}{m_2} & 0 & \frac{b}{m_2} \\ 0 & 0 & 0 & 1 \\ \frac{k_2}{m_1} & \frac{b}{m_1} & -\frac{k_1}{m_1} & -\frac{b}{m_1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -1 \\ 0 \end{bmatrix} \dot{u}$$

- Matrix Form of State equations (output matrix)

$$y_1 = \ddot{x}_2 = -\frac{k_2}{m_2} x_1 - \frac{b}{m_2} x_2 + \frac{b}{m_2} x_4 \quad : \text{Sprung mass acceleration}$$

$$y_2 = z_s - z_u = x_1 \quad : \text{Suspension Deflection}$$

$$y_3 = z_u - u = x_3 \quad : \text{Tire Deflection}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} -\frac{k_2}{m_2} & -\frac{b}{m_2} & 0 & \frac{b}{m_2} \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

2. MATLAB Simulation : using Transfer function

▪ Suspension Parameters

```
m1=55;           % unsprung mass
m2=400;          % sprung mass
b=1000;          % damping ratio
k1=180000;       % stiffness of Tire
k2=18000;        % stiffness of spring
```

▪ Displacement of Mass (Transfer function)

```
% Transfer Function of sprung mass displacement
num_s=[k1*b k1*k2];
den=[m1*m2 (m1+m2)*b [k2*(m1+m2)+k1*m2] k1*b k1*k2];

% Transfer Function of sprung mass displacement
num_u=[k1*m2 k1*b k1*k2];
```

▪ Design Considerations (Transfer function)

```
% Transfer Function of sprung mass acceleration
num_1=[k1*b k1*k2 0 0];

% Transfer Function of suspension deflection
num_2=[-k1*m2 0 0];

% Transfer Function of tire deflection
num_3=[-m1*m2 -(m1+m2)*b -k2*(m1+m2)];

prints(sys(num_1,den) % print system transfer function
```

2. MATLAB Simulation

▪ Making Input functions

```
t=0:0.01:20; % 시간을 정의

%sine 함수
u1=0.1*sin(0.2*t);

% sine 함수를 이용한 자갈길
u2=0.02*sin(4*t)+0.02*abs(sin(4*t)); % abs() : 절대값 함수

% 과속방지턱
u3=0.05*sin(2*pi/20*(t-5))+abs(0.05*sin(2*pi/20*(t-5)));
```

▪ Simulation & Plot result

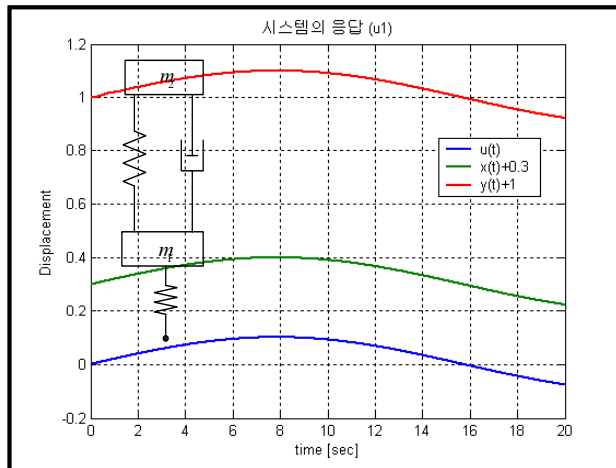
```
% Linear simulation
y=lsim(num_s,den,u1,t);
x=lsim(num_u,den,u1,t);

plot(t,u1,t,x+0.3,t,y+1);
grid on;

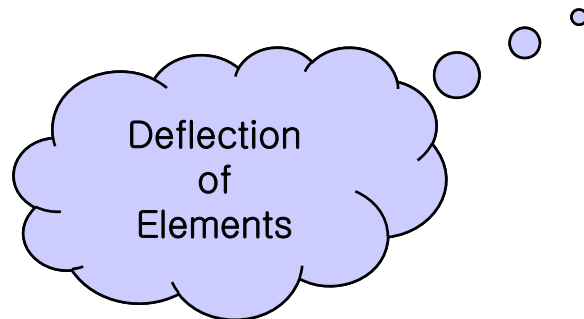
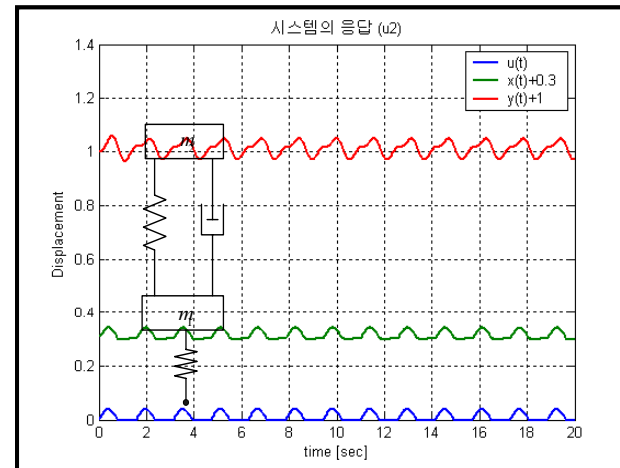
title('시스템의 응답');
xlabel('time [sec]');
ylabel('Displacement');
legend('u(t)', 'x(t)+0.3', 'y(t)+1');
```


2. MATLAB Simulation : simulation result

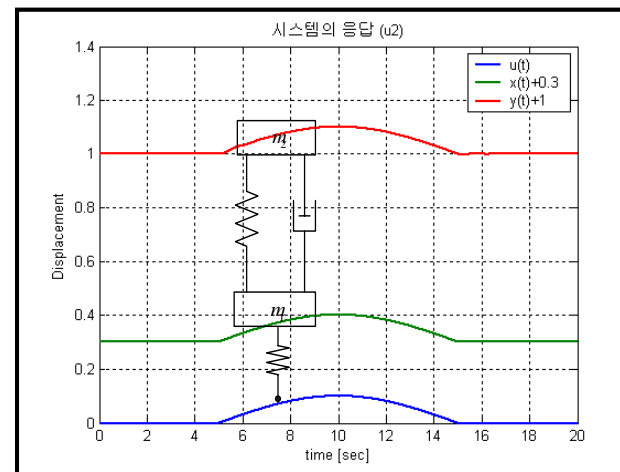
Simulation result(1)



Simulation result(2) : 자갈길

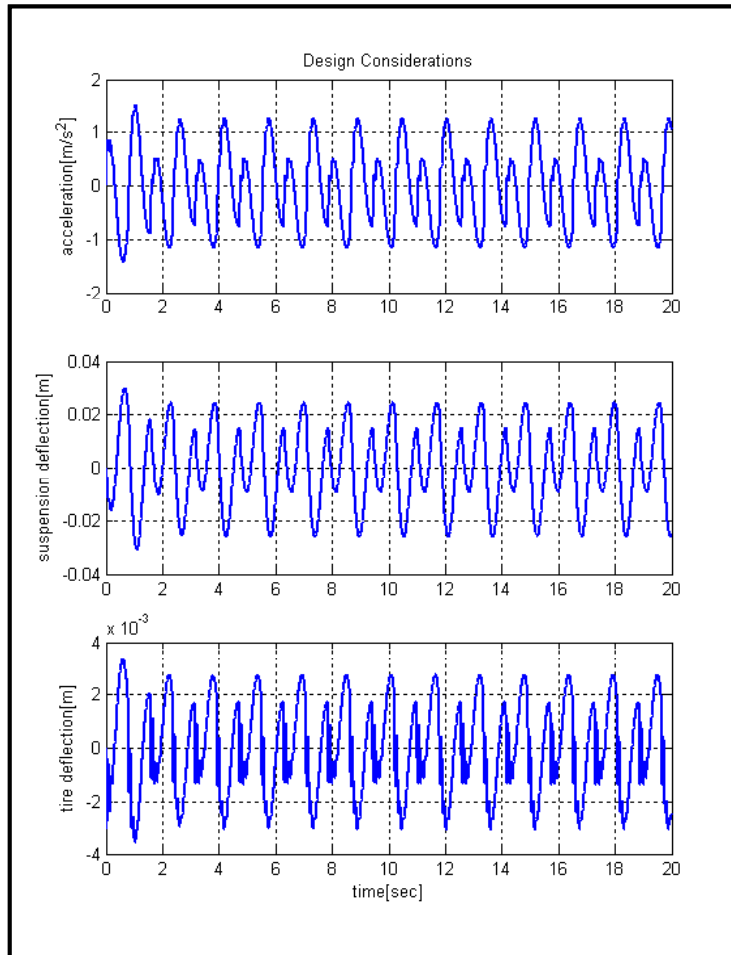


Simulation result(3) : 과속방지턱

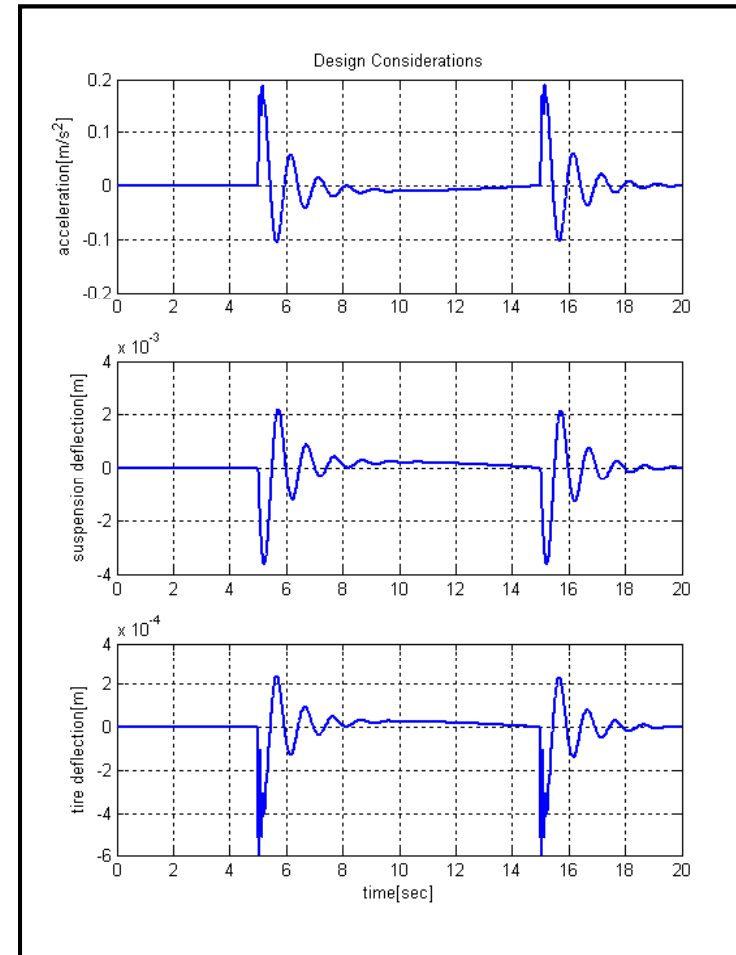


2. MATLAB Simulation : simulation result

■ Simulation result : Design Considerations



자갈길



과속방지턱

2. MATLAB Simulation : Using State equation

▪ Suspension Parameters

```
m1=55;           % unsprung mass
m2=400;          % sprung mass
b=1000;          % damping ratio
k1=180000;       % stiffness of Tire
k2=18000;        % stiffness of spring
```

▪ State Equation

```
% Define State eqations
A=[ 0  1  0  -1;
   -k2/m2 -b/m2  0  b/m2;
    0  0  0  1
   k2/m1  b/m1 -k1/m1 -b/m1];

B=[0; 0; -1; 0];

C=[-k2/m2 -b/m2  0  b/m2;
   1  0  0  0;
   0  0  1  0];
D=[0; 0; 0];
```

▪ Making Input functions

```
t=0:0.01:20; % 시간을 정의

% sine 함수
u1=0.1*sin(0.2*t);

% sine 함수를 이용한 자갈길
u2=0.02*sin(4*t)+0.02*abs(sin(4*t)); % abs() : 절대값 함수

% 과속방지턱
u3=0.05*sin(2*pi/20*(t-5))+abs(0.05*sin(2*pi/20*(t-5)));

% 입력의 미분함수 구하기 : State equation의 입력
u3_d=diff(u3)./0.01;
u3_d=[u3_d u3_d(length(u3_d))];
```

2. MATLAB Simulation : Using State equation

- Simulation & plot result

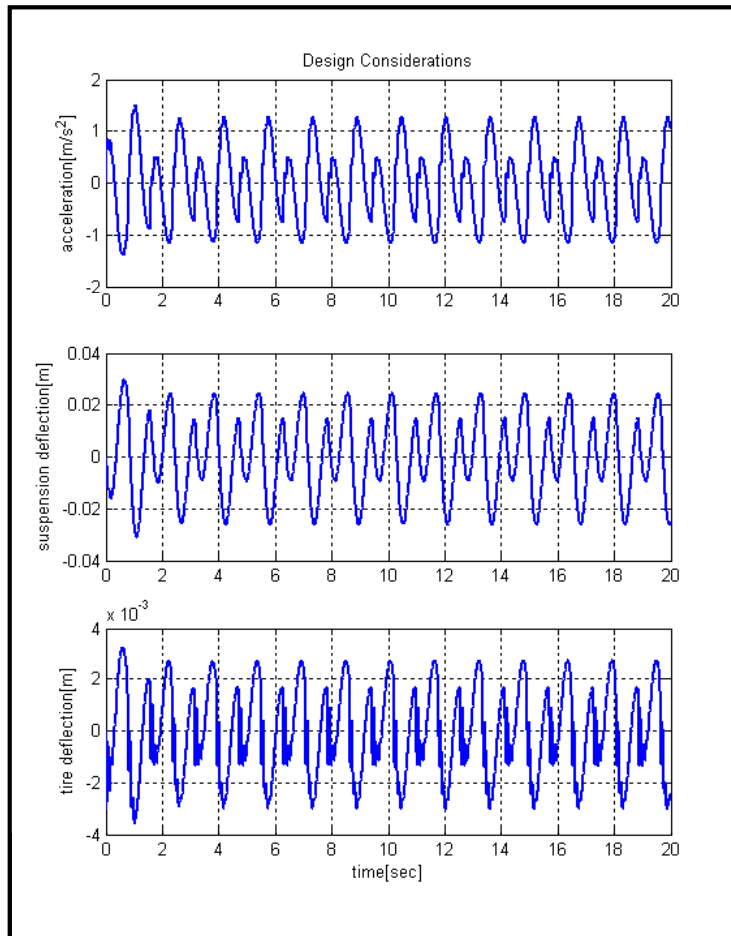
```
% Linear simulation
y=lsim(A,B,C,D,u3_d,t); % n-row, 3-column

figure
subplot(311)
plot(t,y(:,1),'linewidth',2); grid on;
ylabel('acceleration[m/s^2]'); title('Design Considerations');
subplot(312)
plot(t,y(:,2),'linewidth',2); grid on;
ylabel('suspension deflection[m]');
subplot(313)
plot(t,y(:,3),'linewidth',2); grid on;
ylabel('tire deflection[m]'); xlabel('time[sec]');
```

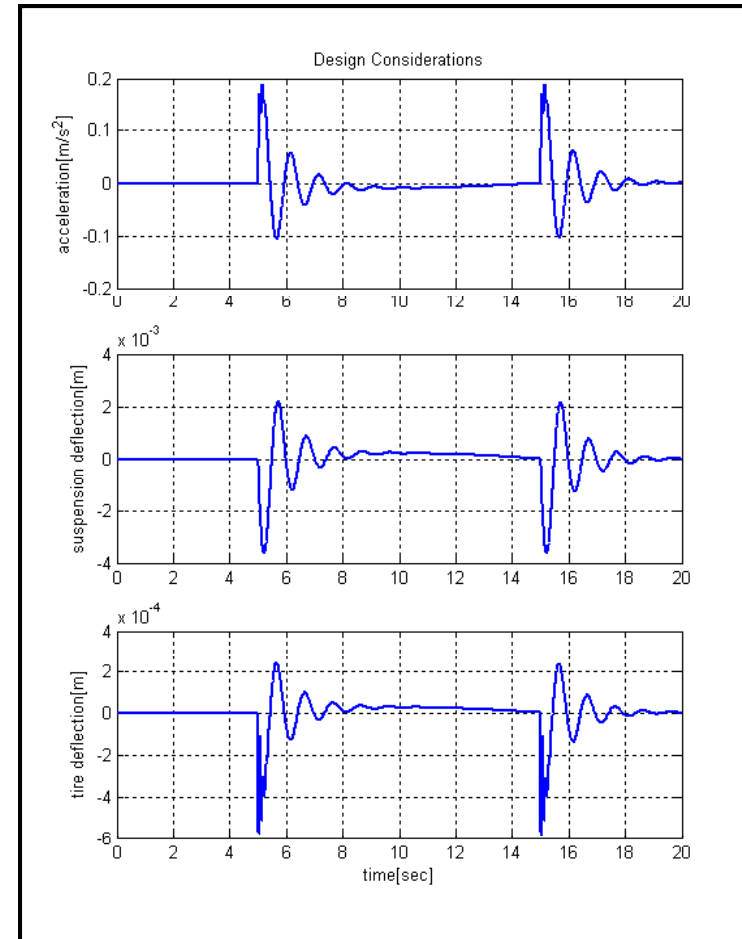
y가 n행 3열의
데이터

2. MATLAB Simulation : simulation result with state equation

▪ Simulation result : Design Considerations



자갈길



과속방지턱

3. For문을 이용한 Parametric Study

▪ Parameters & Input function

```
m1=55;           % unsprung mass
m2=400;          % sprung mass
b=1000;          % damping ratio
k1=180000;       % stiffness of Tire

t=0:0.01:20;     % 시간을 정의

% sine 함수를 이용한 자갈길
u1=0.02*sin(4*t)+0.02*abs(sin(4*t)); % abs() : 절대값 함수
```

▪ Parametric study

```
k2=[12000 18000 24000 30000]; % 변화시킬 parameter를 배열로 정의

for i=1:4         % index의 설정, 시작:간격:끝
    num1=[k1*b k1*k2(i)]; % 전달함수를 정의
    den1=[m1*m2 (m1+m2)*b [m1*k2(i)+(k1+k2(i))*m2] k1*b k1*k2(i)];
    num2=[k1*m2 k1*b k1*k2(i)];
    den2=[m1*m2 (m1+m2)*b [m1*k2(i)+(k1+k2(i))*m2] k1*b k1*k2(i)];

    y(:,i)=lsim(num1,den1,u1,t); % index에 부여하여 matrix로 저장
    x(:,i)=lsim(num2,den2,u1,t);
end

figure
plot(t,y);        % matrix의 출력
grid on;
title('차체의 응답(y(t))');
xlabel('time [sec]');
ylabel('Displacement'); legend('k_2=12000','k_2=18000','k_2=24000','k_2=30000');
```

3. For문을 이용한 Parametric Study

- Displacement of Sprung mass of K2 (자갈길)

