MATLAB 실습 3

-시뮬링크 활용 방법-

박사과정 서종상

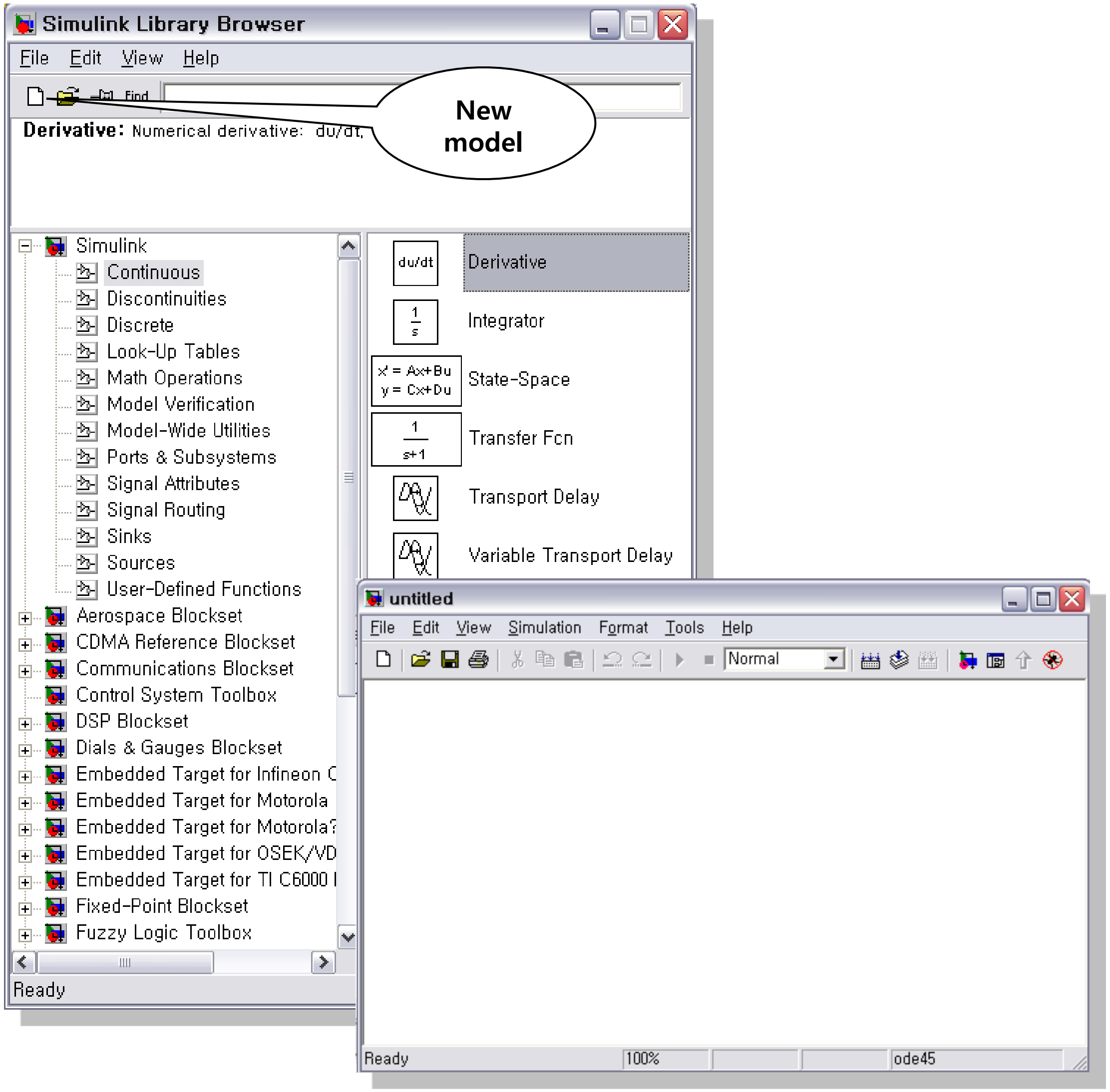
azuresky@snu.ac.kr

Tel:02-880-1942

301-113

Vehicle Dynamics & Control Laboratory

MATLAB/Simulink의 구성



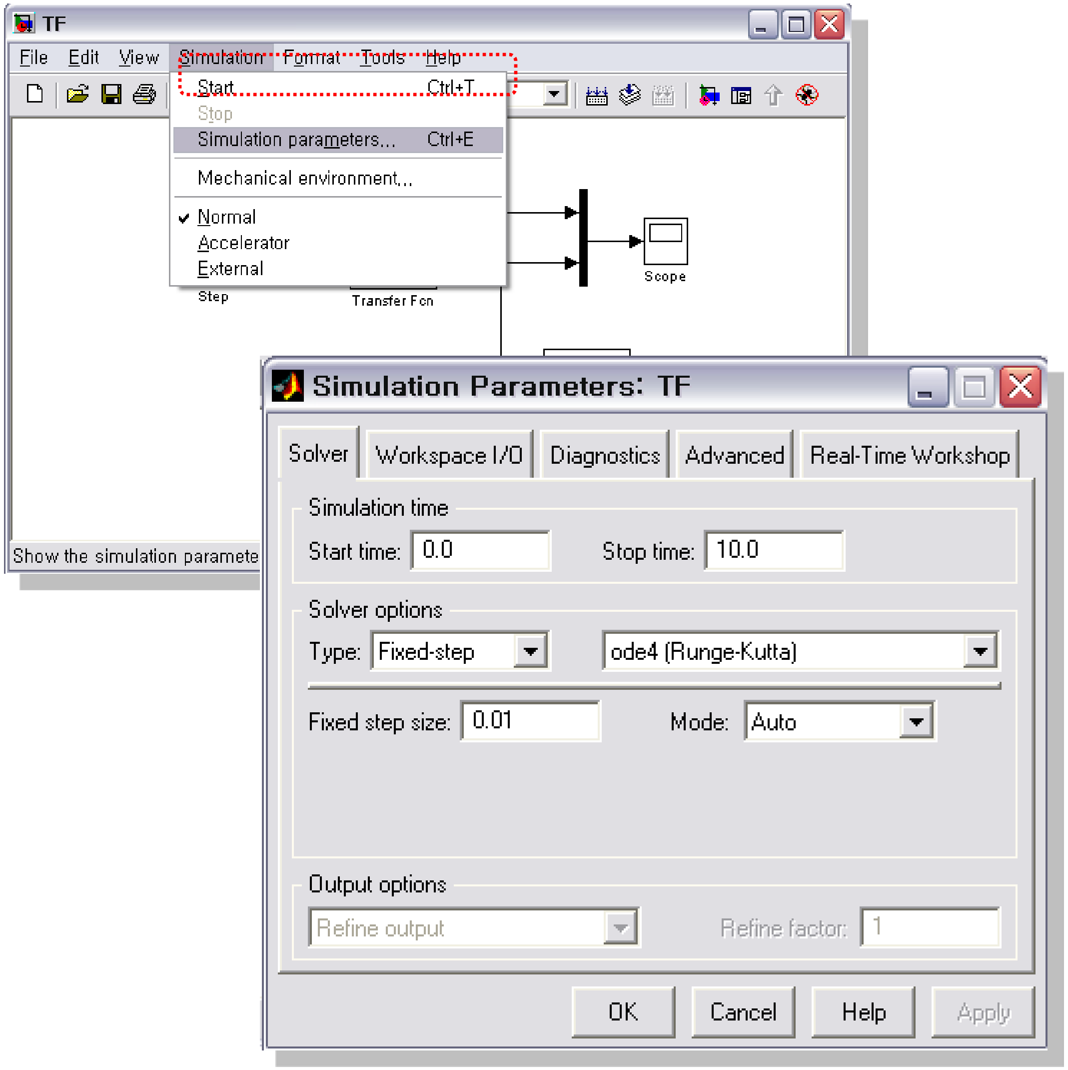
- Simulink Library Browser의 실행

> - MATLAB command창
>   >> simulink
> - Toolbar에서 simulink 버튼

> - Simulink Library Browser
>   - New model을 생성
>   - 주로 Simulink의 Block을 이용
>   - model 창에 필요한 Block으로 model을 구성

- 선형, 비선형 모델을 Block을 이용하여 시각적으로 표현하고 시뮬레이션 할 수 있다.

MATLAB/Simulink Solver Option



- Simulink Simulation Parameter의 설정

– Simulation time : Start/Stop time을 설정

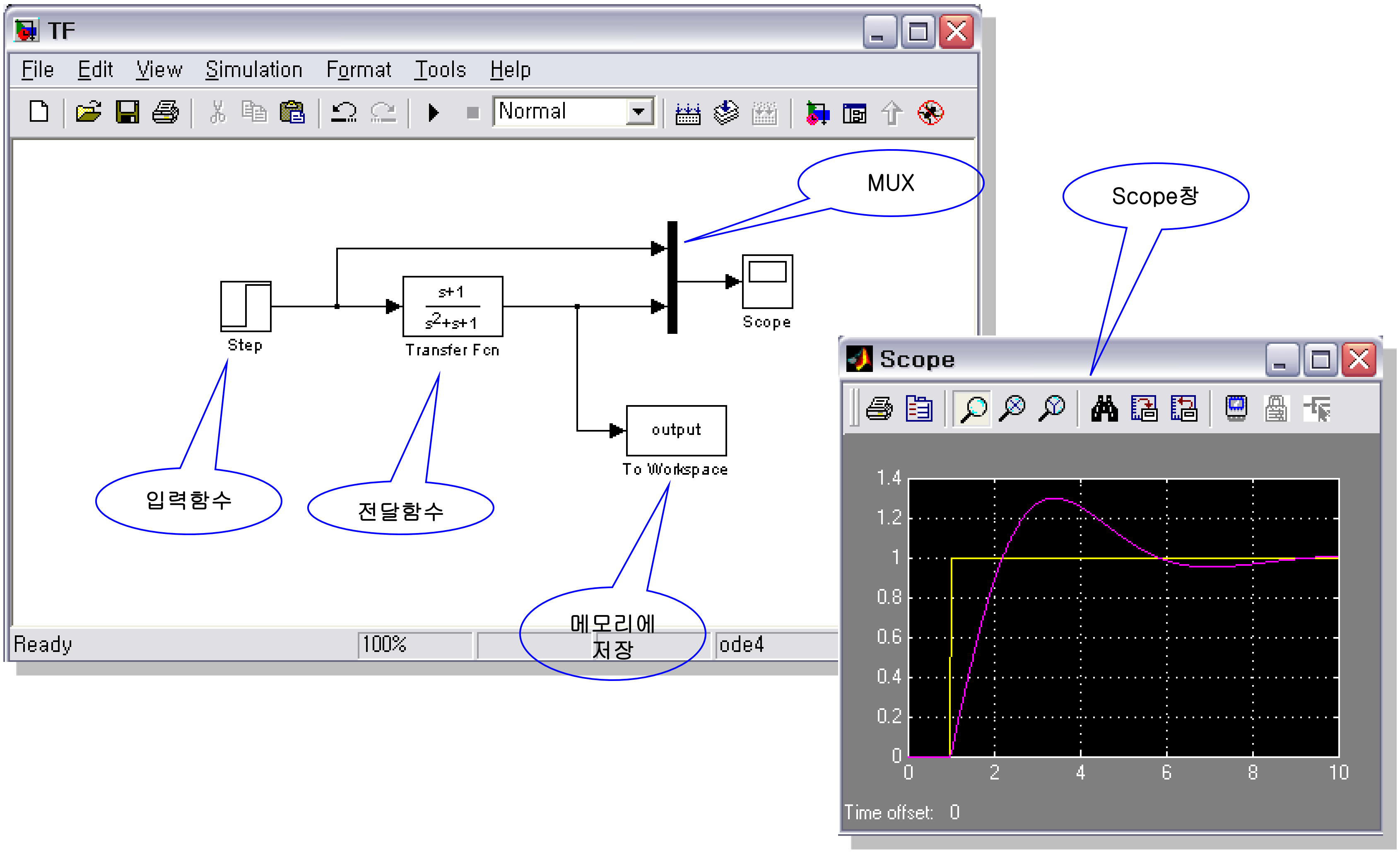
– Solver options : Type/method 설정

– step size 설정

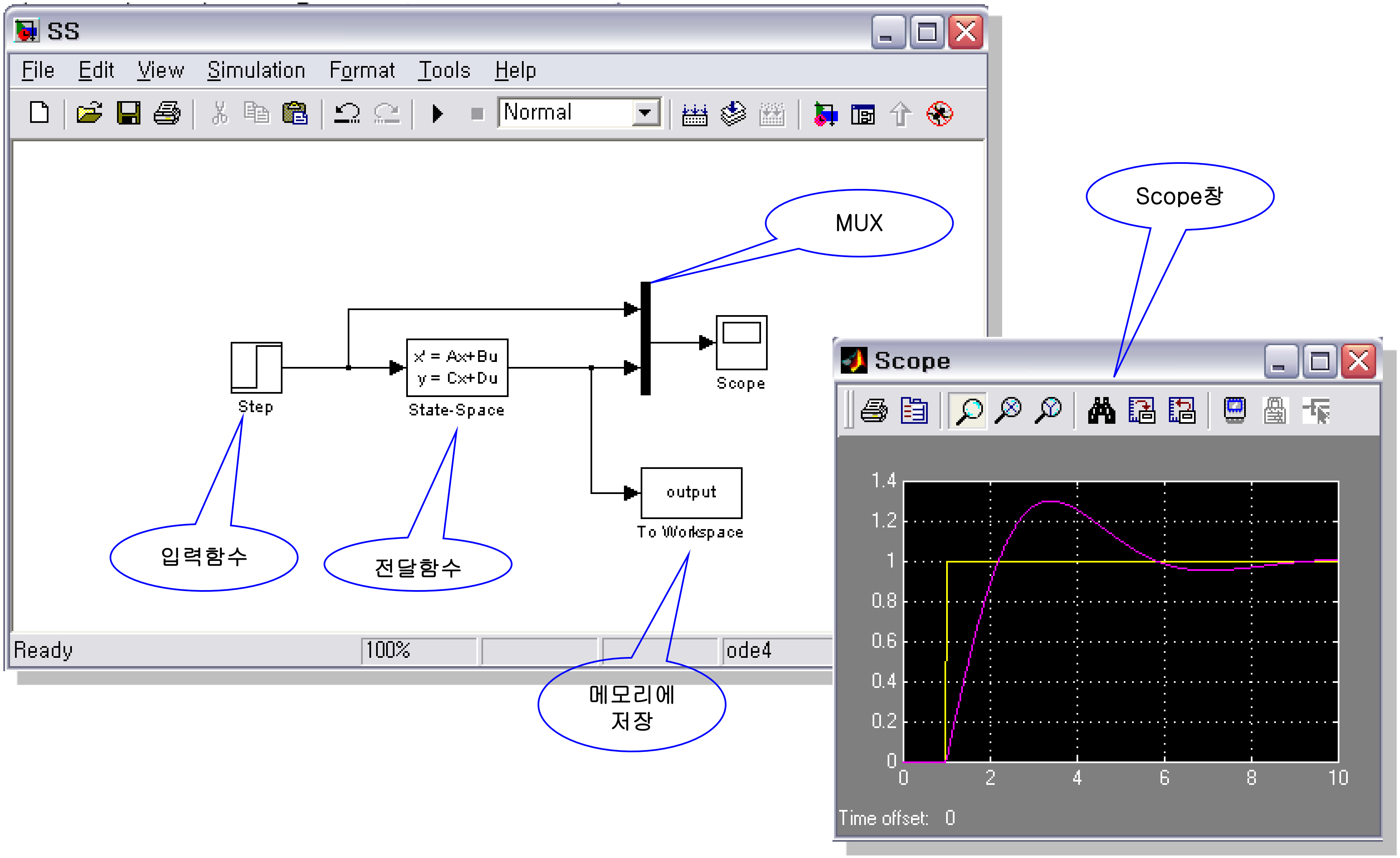
- 왼쪽의 그림과 같이 설정

– 경우에 따라 알맞은 옵션을 선택

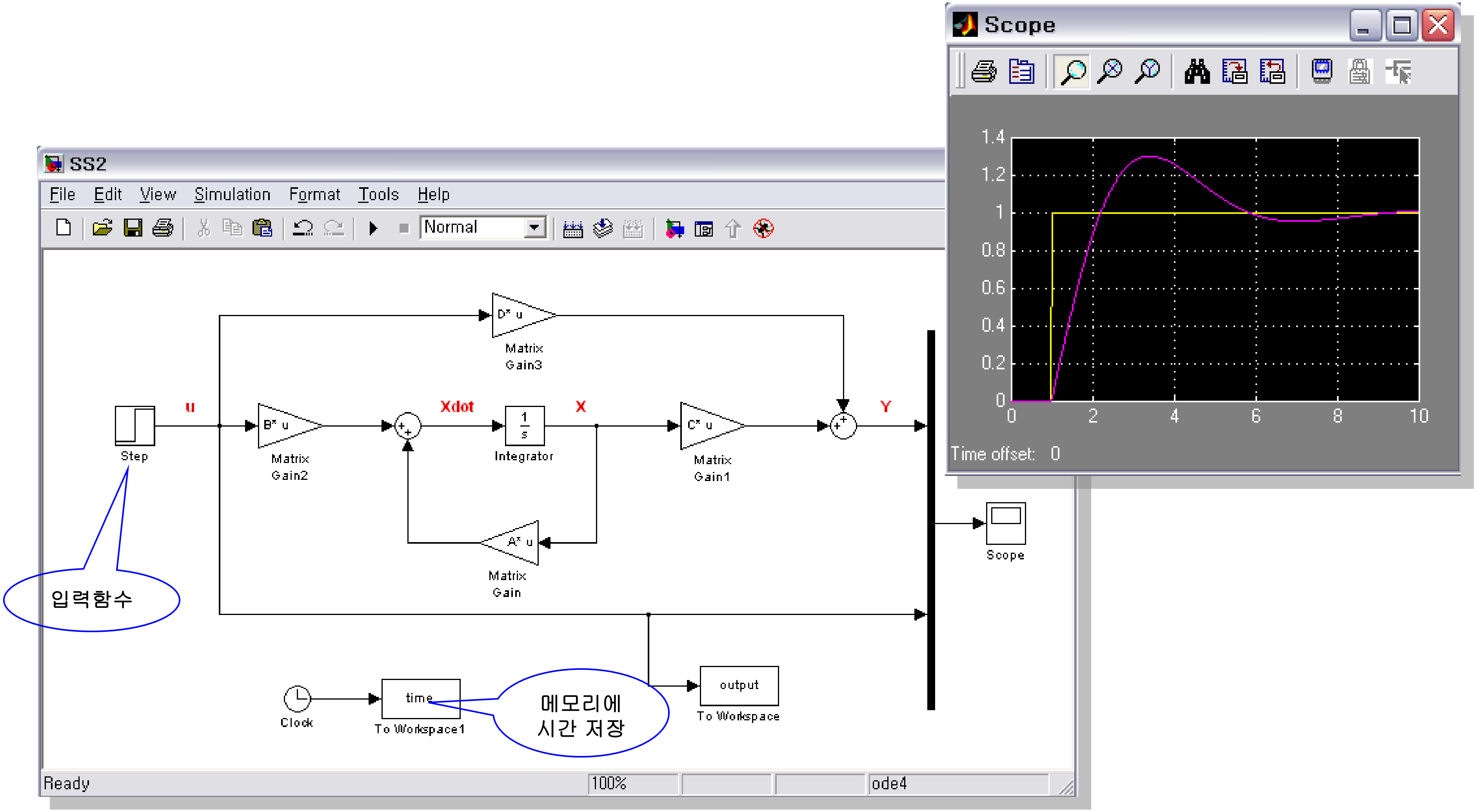
MATLAB/Simulink의 구성 – Transfer Function



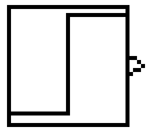
MATLAB/Simulink의 구성 – State Equation [1/2]



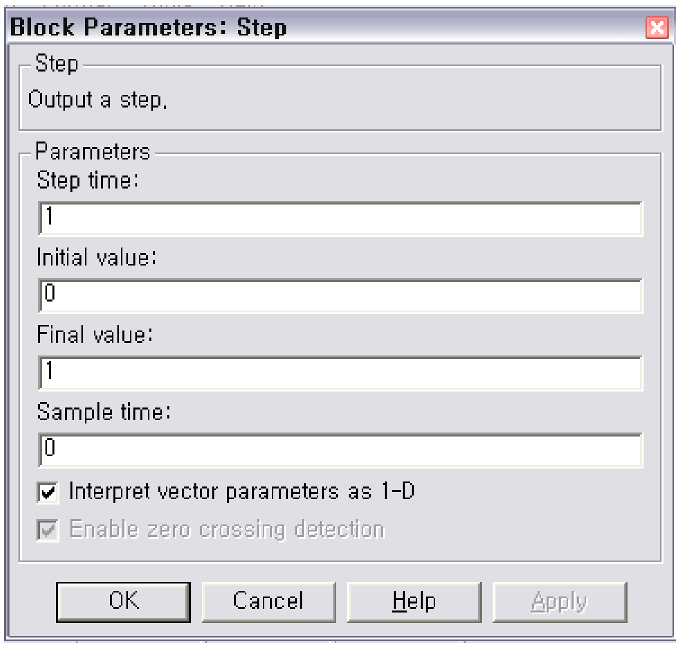
MATLAB/Simulink의 구성 – State Equation [2/2]



MATLAB/Simulink Blocks - Step



Step



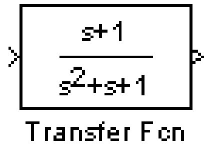
– Step time : Step이 발생할 시간

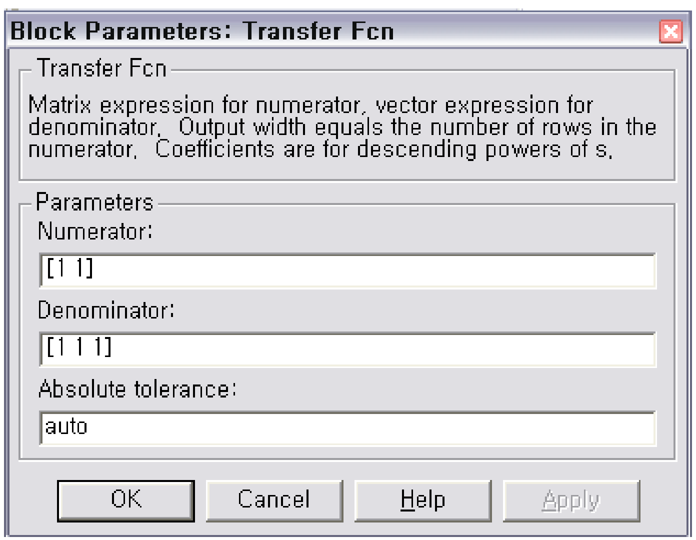
– Initial value : Step이 발생하기 전의 값

– Step value : Step이 발생한 후의 값

▪ Source 디렉토리에서 다양한 입력함수를 사용할 수 있다.

MATLAB/Simulink Blocks – Transfer Function

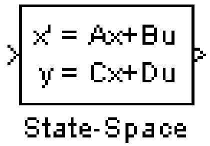


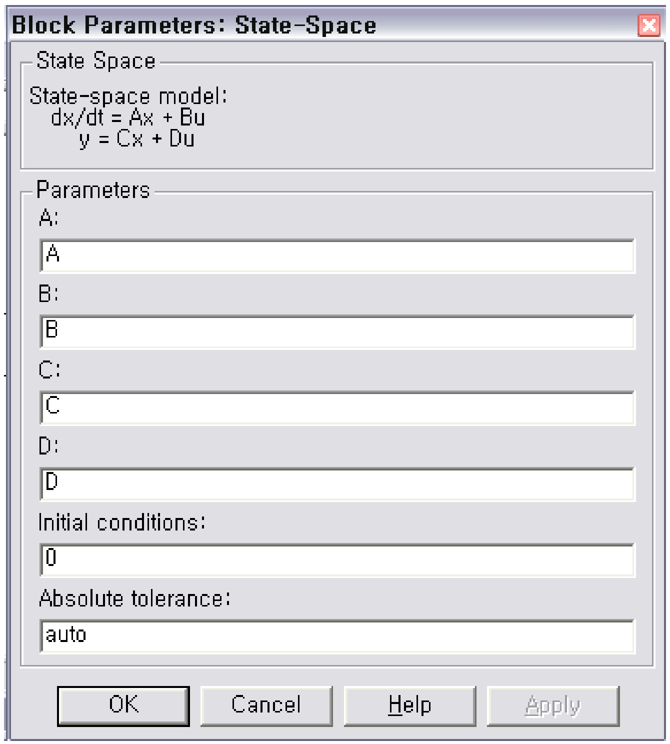


- Numerator : 전달함수의 분자 입력
  (m-file에서 num으로 저장한 후 입력을 하여 사용가능)

- Denominator : 전달함수의 분모 입력
  (m-file에서 num으로 저장한 후 입력을 하여 사용가능)

MATLAB/Simulink Blocks – State Space

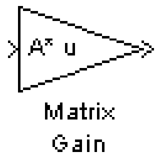


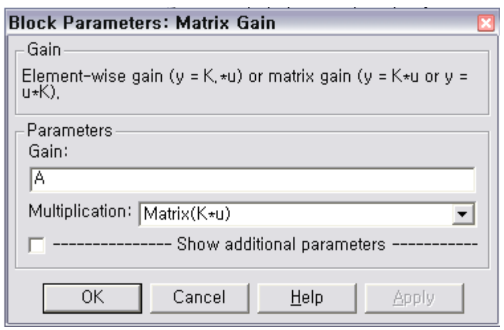


- A : System matrix를 정의
- B : Input matrix를 정의
- C : Output matrix를 정의
- D : Direct transmission matrix를 정의

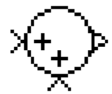
(행렬을 직접 입력 가능)

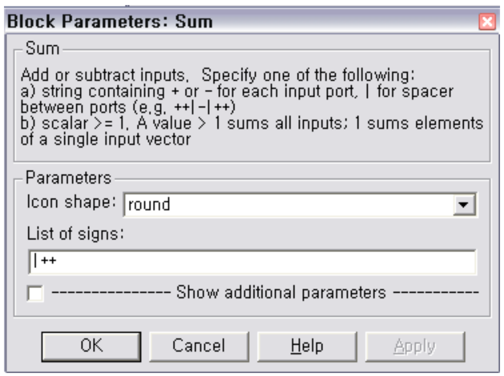
MATLAB/Simulink Blocks – Variable Blocks [1/2]



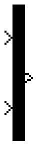


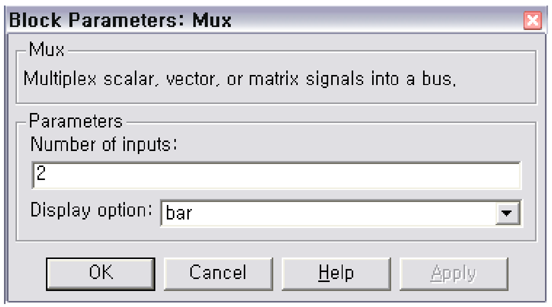
- 행렬 곱을 수행하는 Block
- 행렬을 직접 입력 가능





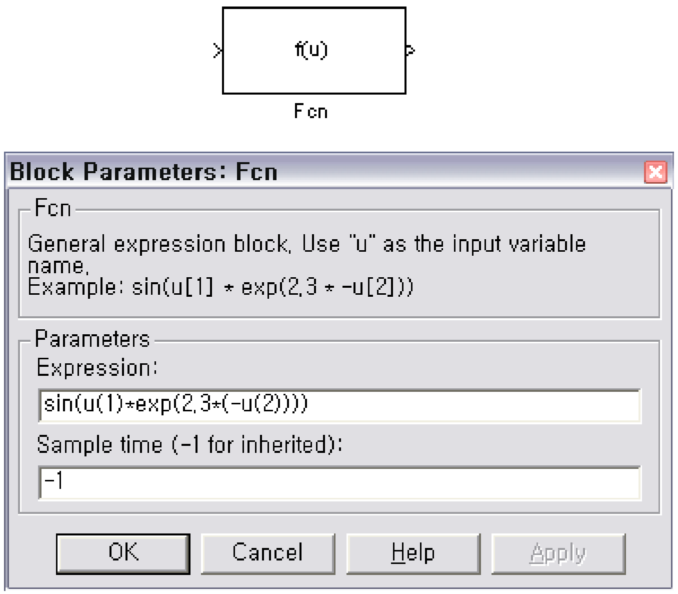
- 합 연산을 하는 Block
- 합, 차 연산
- 입력 port를 정할 수 있다.
- Icon의 형태를 결정할 수 있다.



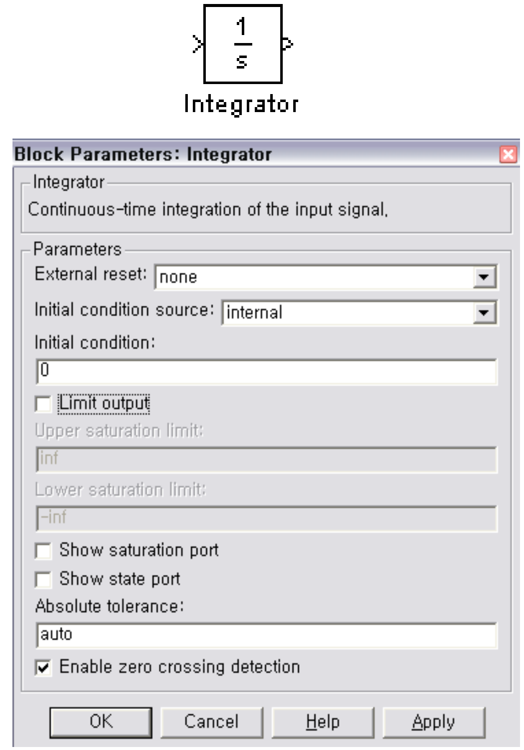


- 2개 이상의 데이터를 배열
- Input을 증가시킬 수 있다.
- 연산할 때 (1 X n)의 의미를 가진다.

MATLAB/Simulink Blocks – Variable Blocks [1/2]

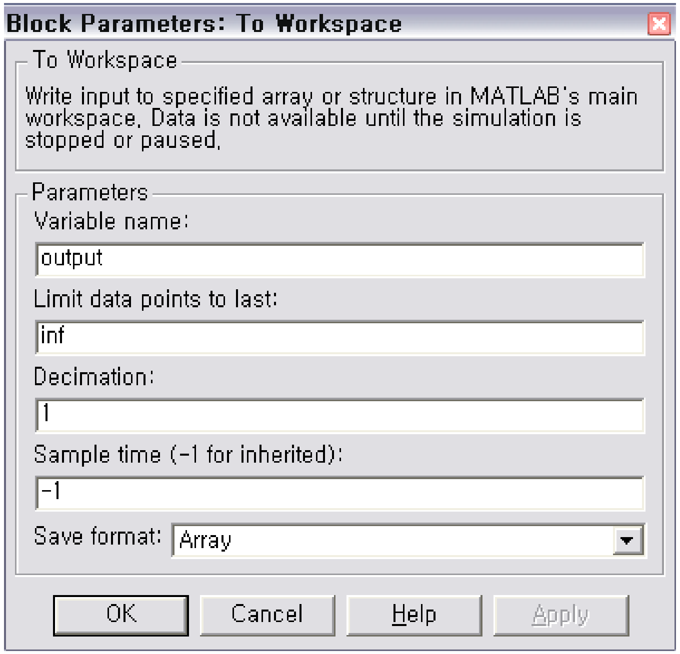


- 사용자 정의 함수
- User Defined Functions 디렉토리에서 선택
- 입력을 변수 u로 놓고 MATLAB 함수로 연산
- 입력이 vector일 경우 u(1), u(2)… 의 형태로 사용
- 차량모델의 다양한 수식을 구현할 때 간편하게 사용



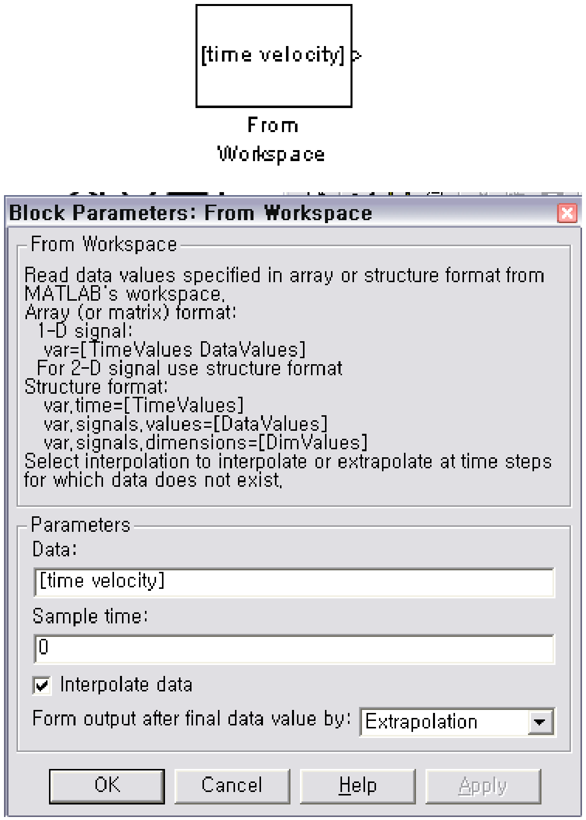
- 적분을 수행
- 미분방정식을 구현할 때 사용
- Initial condition : 초기 값을 선정
  (예 : 초기속도, 초기엔진속도, 초기구동축 토크…)

MATLAB/Simulink Blocks – To Workspace



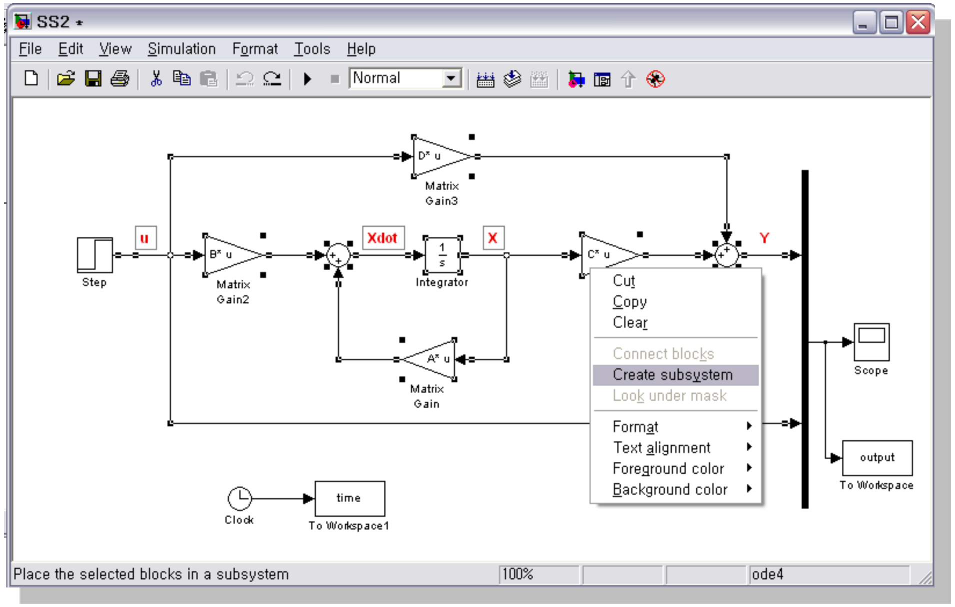
- Workspace에 저장하는 Block (메모에 데이터 저장)
- Variable name : 변수의 이름을 설정
- Save format : Array type으로 설정해야 함

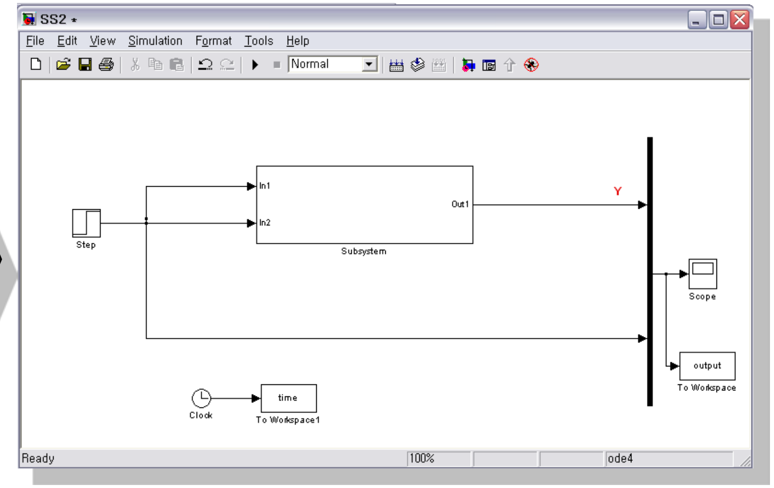
MATLAB/Simulink Blocks – From Workspace



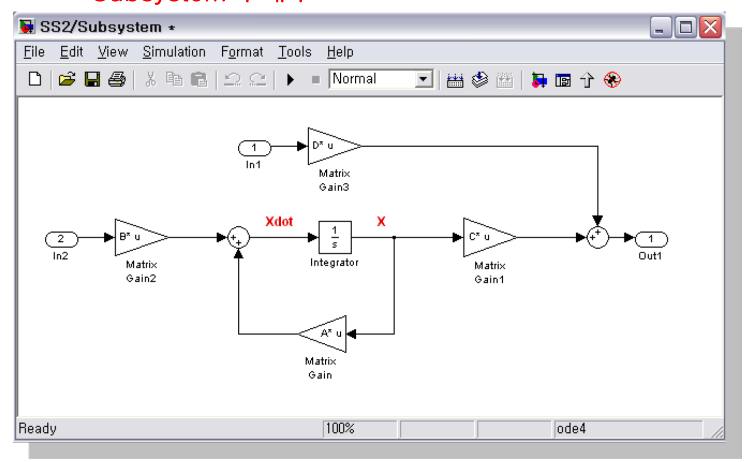
- From Workspace Block (source 디렉토리)
- 메모리(Work Space)에 있는 데이터를 시간에 대하여 생성
- [시간데이터 속도데이터]의 2열의 형태로 정의
- 각 사이의 값은 interpolation
- 주어진 속도 Profile을 입력신호로 쓸 때 사용
- 속도 Profile은 실험데이터로 제공할 것임

MATLAB/Simulink Tips - Subsystem





▪ Subsystem의 내부

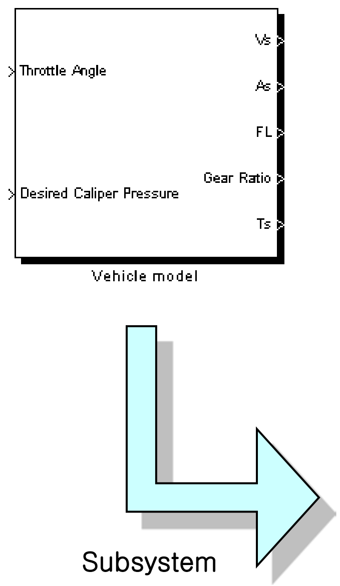


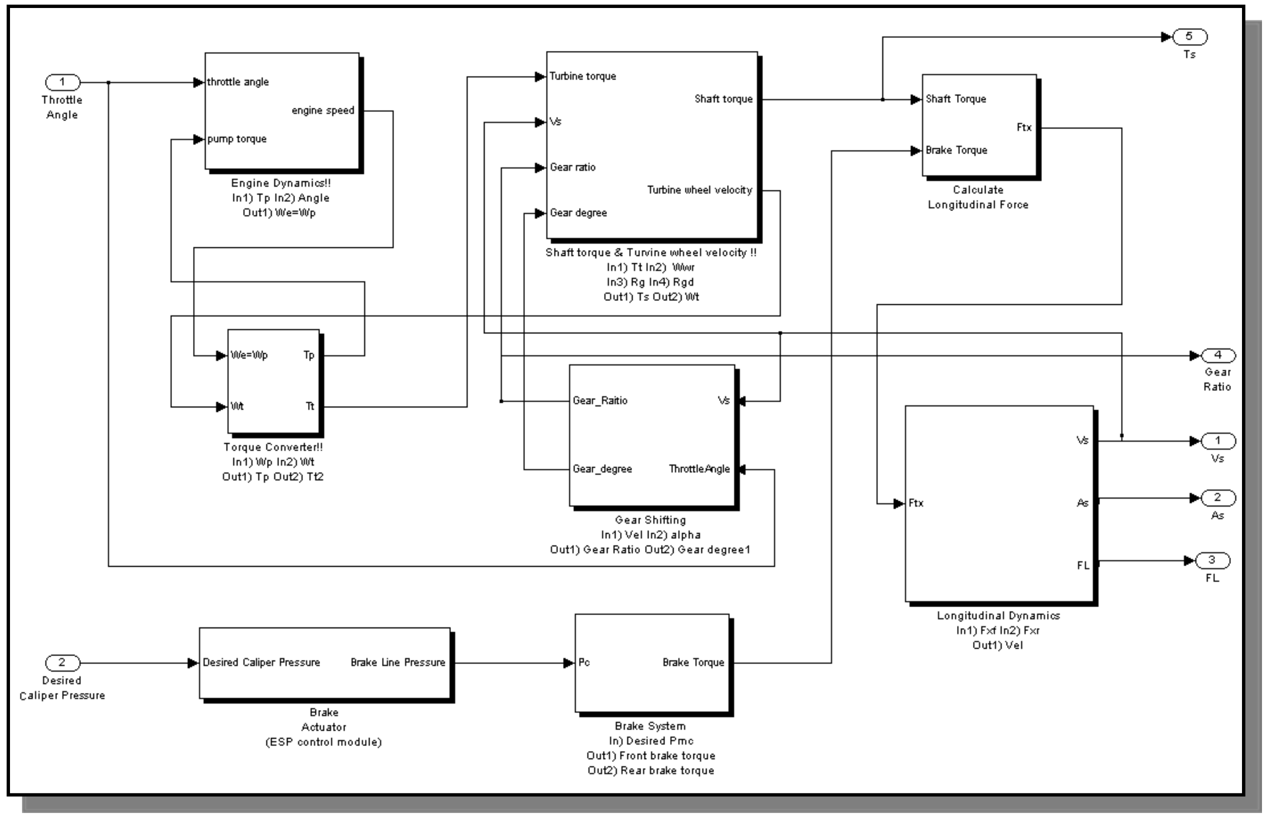
- sub system을 구성할 부분을 Drag하여 선택
- 마우스 오른쪽 클릭하여, Create subsystem 선택
- 오른쪽 그림을 보면 하나의 subsystem이 생성
- subsystem을 더블클릭 하면 내부 구조 확인가능
- input, output port의 이름을 변경할 수 있다.
- subsystem의 이름을 목적에 맞게 바꾸어 모듈화한다.

▪ Library의 port & subsystems 디렉토리에서 반대로 Subsystem을 만들고 내부를 구성할 수 있다.

Example of MATLAB/Simulink

Power Train Model



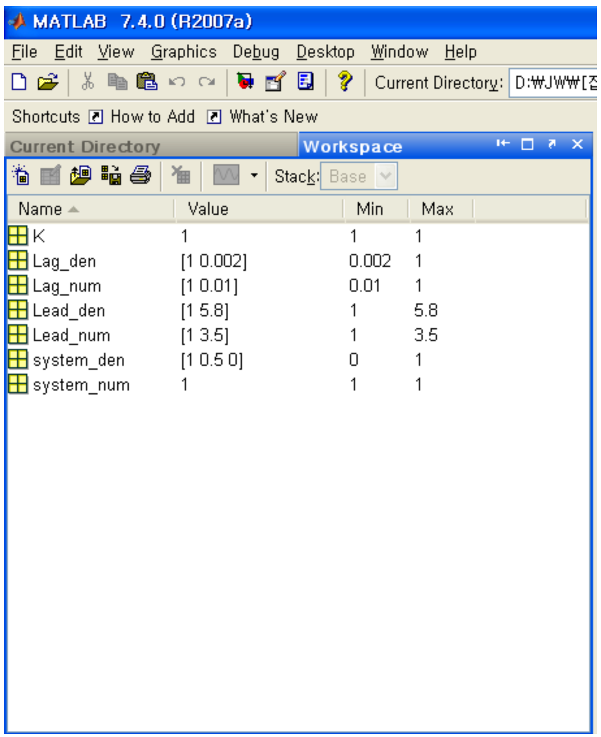


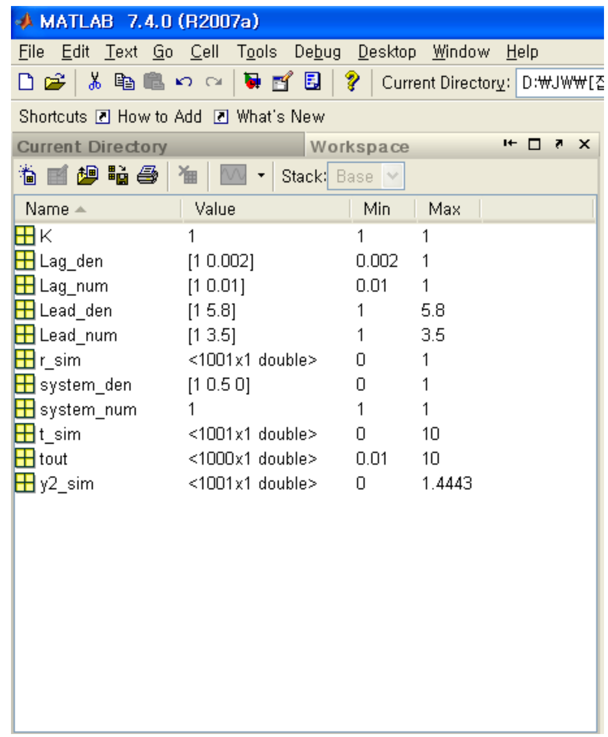
MATLAB/Simulink in m-file

M-file 에서 미리 정의해둔 simulink 실행.

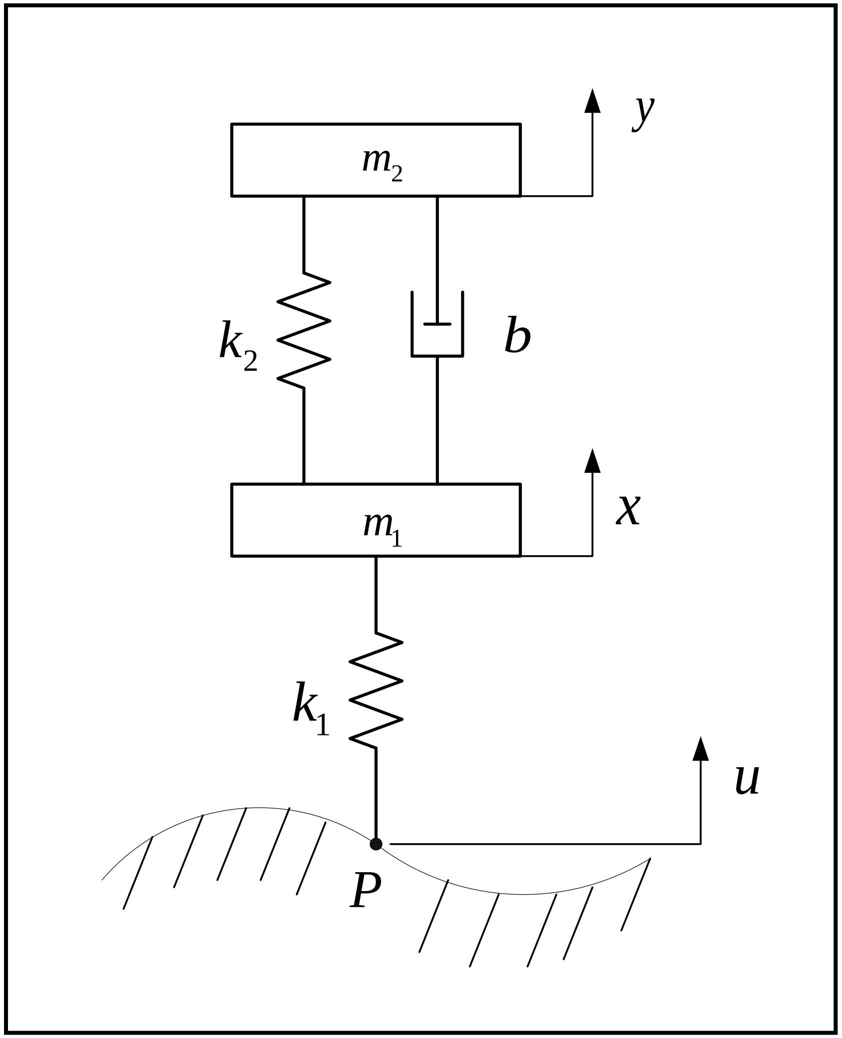
```
sim('class2.mdl');
```

Simulink 상 save to workspace block 을 통해 원하는 값을 workspace 에 저장할 수 있다.





Suspension Example



- Design Considerations

1. Ride Quality
$\rightarrow$ *Sprung mass acceleration* : $\ddot{y}$

2. Rattle space
$\rightarrow$ *Suspension Deflection* : $y - x$

3. Tire Force Vibration
$\rightarrow$ *Tire Deflection* : $x - u$

- Suspension Design Parameters

$\rightarrow$ *Spring Stiffness* : k_2

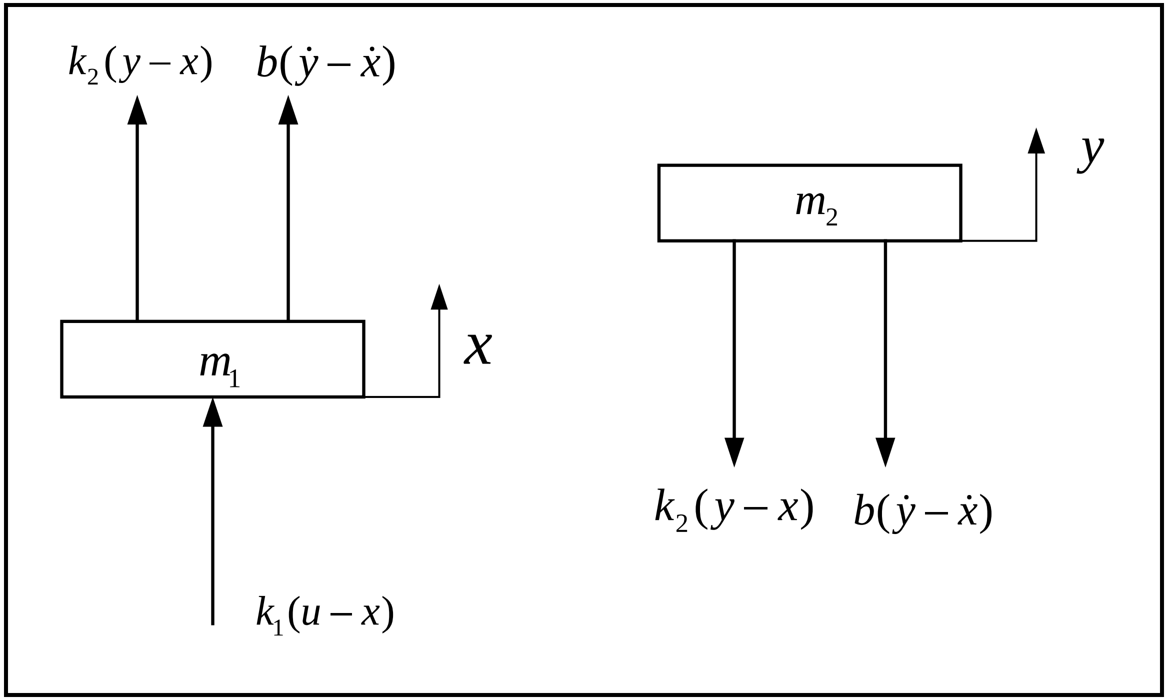
$\rightarrow$ *Damping Ratio* : b

$\rightarrow$ *Tire Stiffness* : k_1

.

Dynamic Equations

- Free Body Diagram



- Dynamic Equations

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

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

.

Laplace Transform

- 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

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

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

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

.

State Equation

- General Form of State Equation

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

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

$x_1 = z_s - z_u$: *Suspension Deflection*

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

$x_3 = z_u - u$: *Tire Deflection*

$x_4 = \dot{z}_u$: *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

$$\dot{x}_1 = \dot{z}_s - \dot{z}_u = x_2 - x_4$$
$$\dot{x}_2 = -\frac{k_2}{m_2}(z_s - z_u) - \frac{b}{m_2}(\dot{z}_s - \dot{z}_u) = -\frac{k_2}{m_2}x_1 - \frac{b}{m_2}x_2 + \frac{b}{m_2}x_4$$
$$\dot{x}_3 = \dot{z}_u - \dot{u} = x_4 - \dot{u}$$
$$\dot{x}_4 = \frac{k_2}{m_1}(z_s - z_u) + \frac{b}{m_1}(\dot{z}_s - \dot{z}_u) - \frac{k_1}{m_1}(z_u - u) = \frac{k_2}{m_1}x_1 + \frac{b}{m_1}x_2 - \frac{k_1}{m_1}x_3 - \frac{b}{m_1}x_4$$

System Matrix

- 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)

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

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

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

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

.

MATLAB Simulation using Laplace Transform

- Suspension Parameters

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

- 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)];

printsys(num_1,den)    % print system transfer function
```

.

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 sping
```

- State Equation

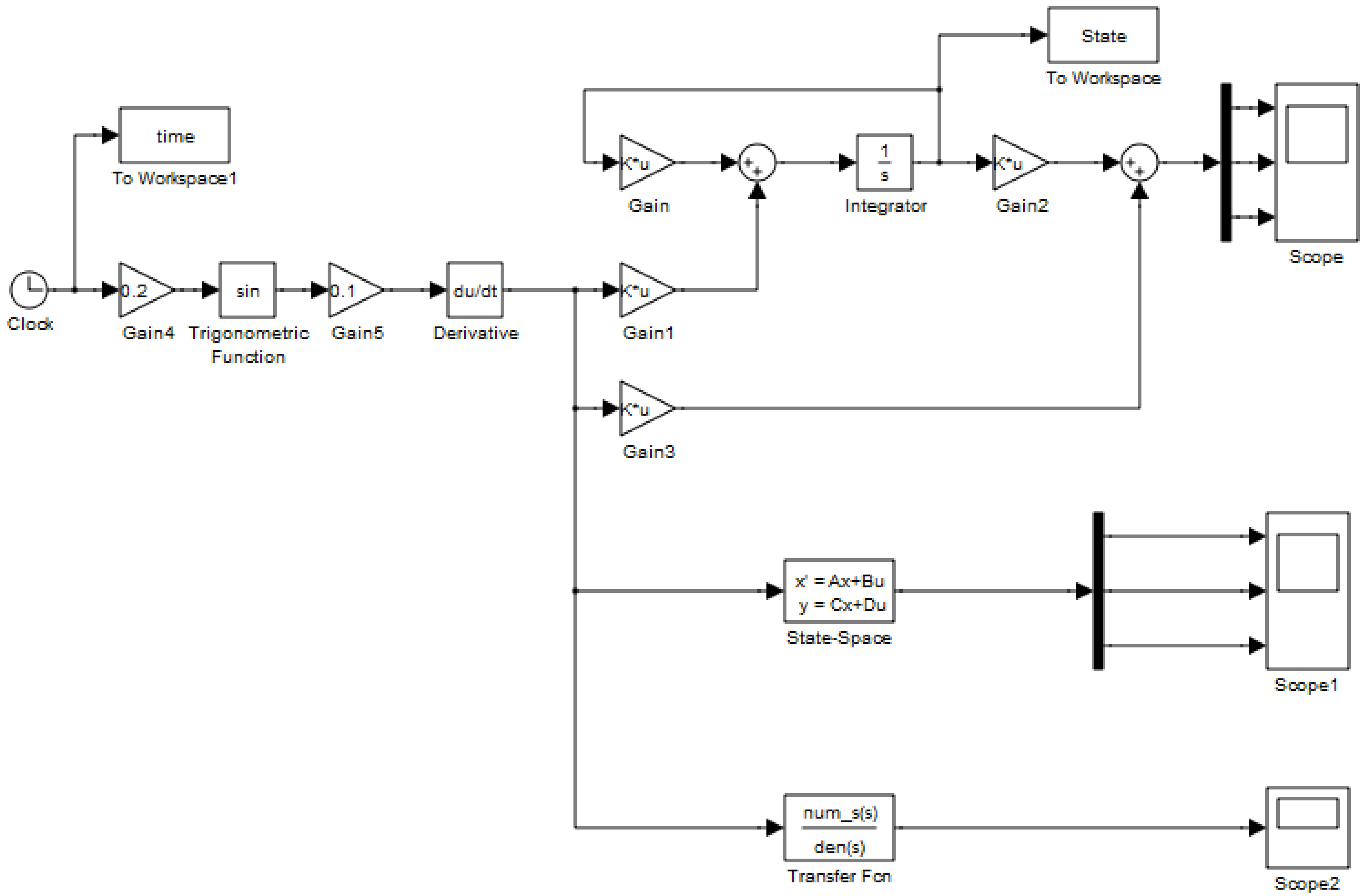
```
% Define State eqations
A=[   0      1       0        -1;
   -k2/m2 -b/m2     0       b/m2;
     0      0       0         1
   k2/m1   b/m1  -k1/m1  -b/m1];

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)));
```

.

Simulink Model



Simulation Results