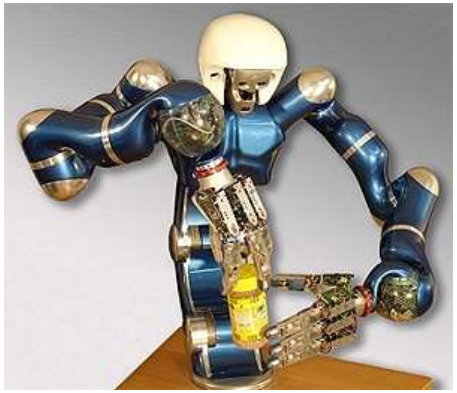
Control System Design for Automated Driving

Lecture 04



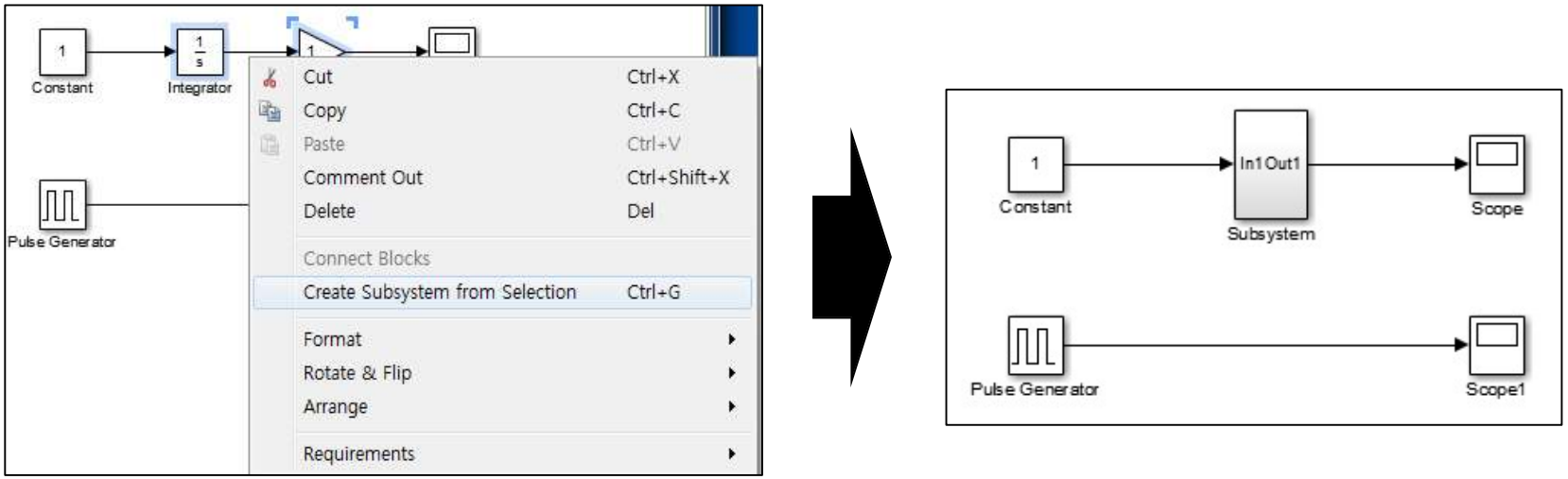
Conditionally Executed Subsystems

Enabled Subsystems

- Conditionally Executed Subsystem
  - 입력 signal의 값에 따라서 실행이 결정되는 subsystem
- Types of Conditionally Executed Subsystems
  - Enabled subsystem
  - Triggered subsystem
  - Triggered and Enabled subsystem
  - Control Flow subsystem
- Enabled Subsystem
  - Control signal이 positive인 경우, 각각의 simulation step에서 실행한다.
  - 즉, control signal이 negative에서 positive로 바뀌는 순간 실행을 시작하여 positive에서 negative로 바뀌는 순간 실행을 멈춘다.

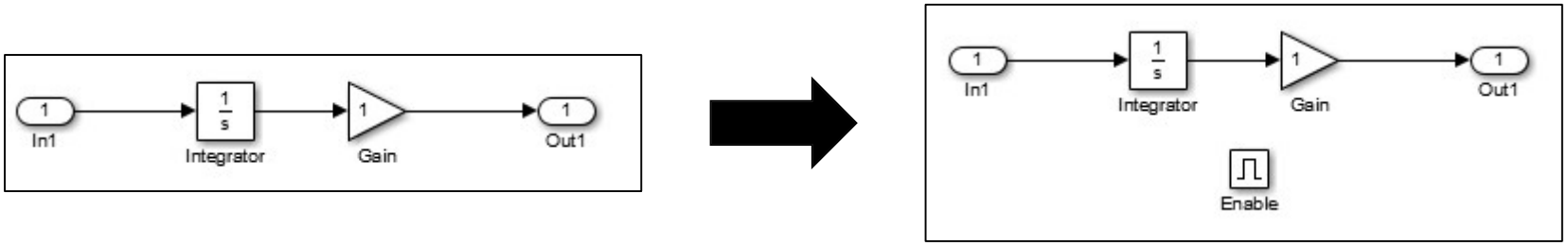
Enabled Subsystems

- Enabled Subsystem model 제작



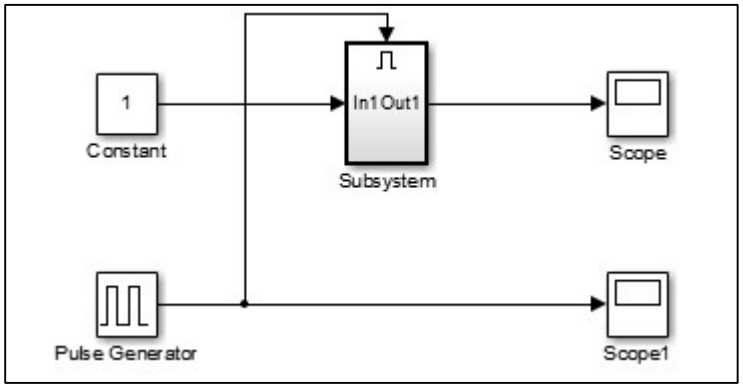
<위와 같이 **integrator block**과 **gain block**을 하나의 **subsystem** 으로 만든다.>

Enabled Subsystems



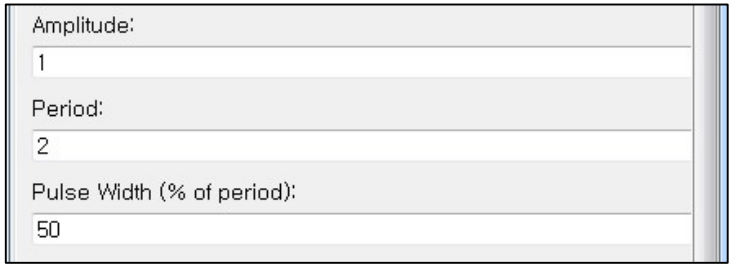
<Port & Subsystems Library에 있는 enable block를 subsystem에 넣는다.>
※subsystem을 더블 클릭하면 위와 같이 subsystem의 구조를 볼 수 있다.

<subsystem의 모양에 enable 표시가 생긴 것을 확인할 수 있다.>

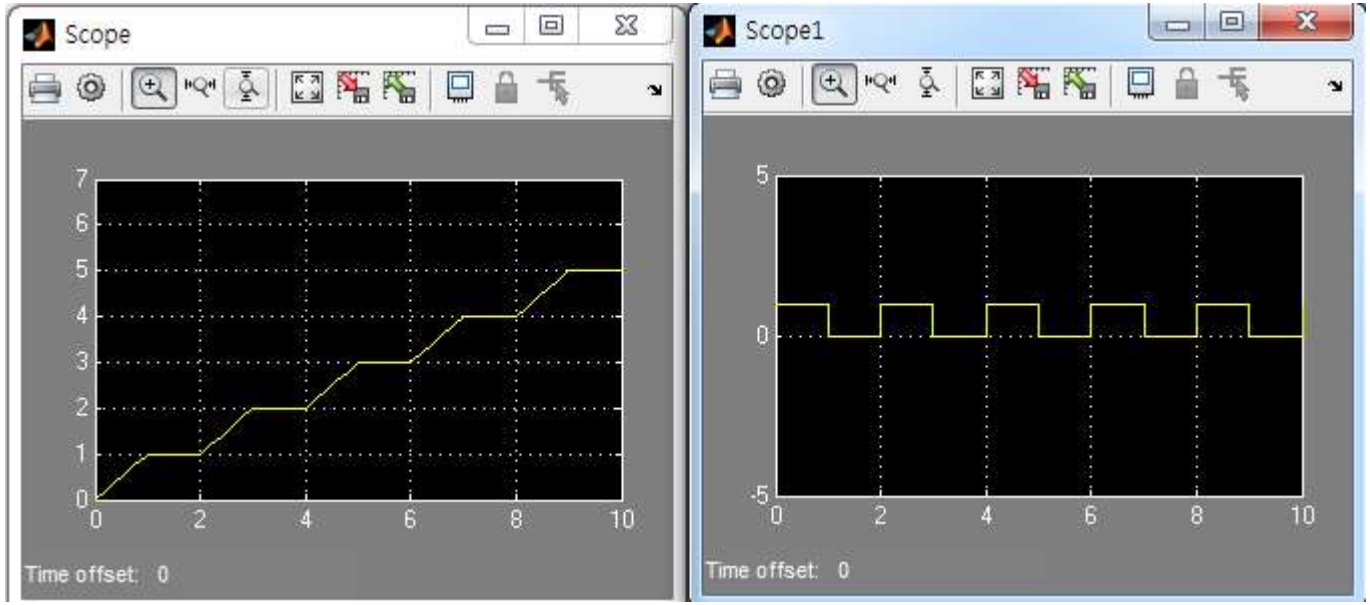


<완성된 enabled subsystem 을 포함한 model>

Enabled Subsystems



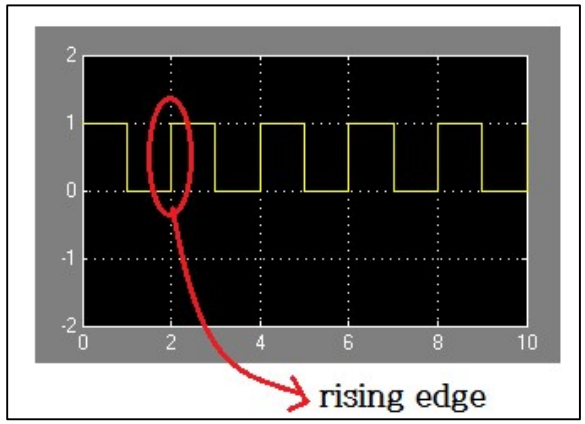
<주기가 2이고 진폭이 1인 signal생성을 위하여 Pulse Generator의 parameter 값을 다음과 같이 입력한다.>



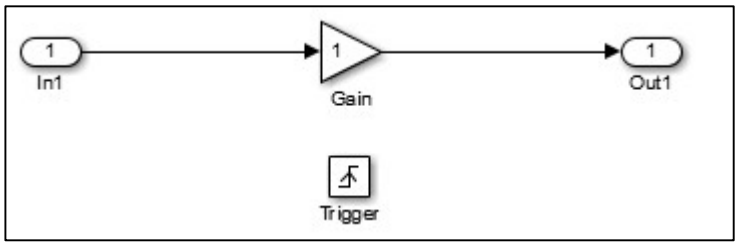
<model을 simulation한 결과>
<control signal의 값이 0보다 클 때만 동작하는 것을 볼 수 있다.>

Triggered Subsystems

- Triggered Subsystem
  - 임의의 trigger event가 발생할 때 마다 한번 실행된다.
  - Trigger 이란 signal의 rising edge 또는 falling edge를 뜻한다.

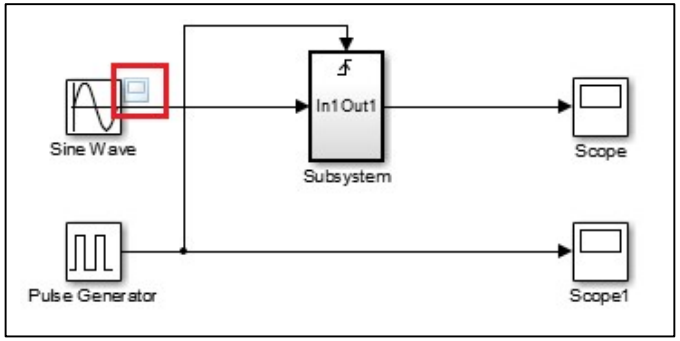


  - Triggered Subsystem model 만들기

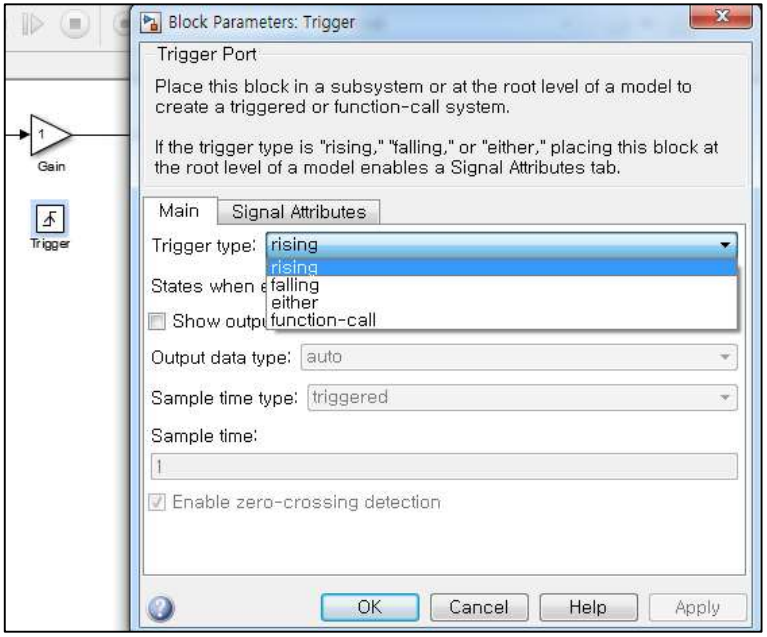


<Gain block으로만 이루어진 subsystem에 Trigger block를 추가한다.>

Triggered Subsystems

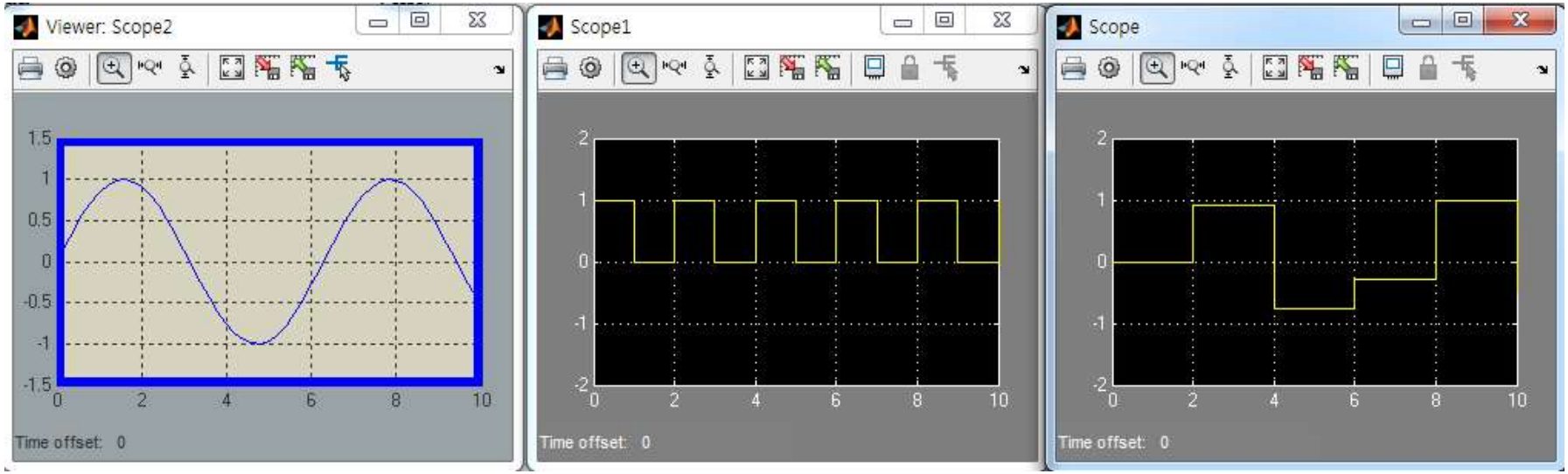


<위와 같이 **Triggered Subsystem**을 포함한 **model**이 완성되었다.>



<Trigger block의 block parameters 에서 trigger type를 결정 할 수 있다>

Triggered Subsystems

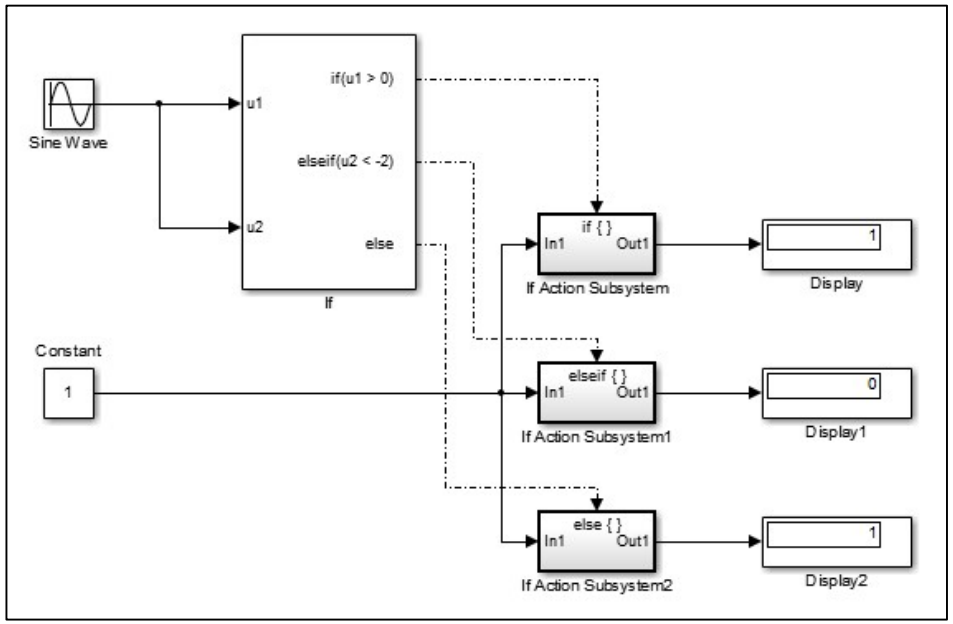


<scope1에서 rising edge일 때 scope2의 sine 그래프의 값이 scope에 표시되었다.>

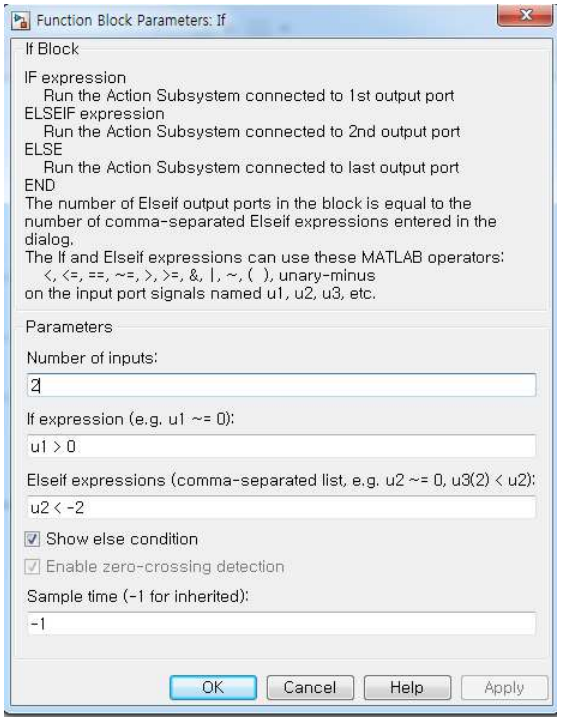
- Triggered Subsystem은 주기적으로 Update 되는 Embedded System 을 모델링하는데 주로 사용된다.

Control Flow Subsystems

- Control Flow Subsystem
  - Programming 언어의 control flow 문들과 유사한 control logic 을 Simulink에서도 제공한다.
  - Simulink에는 for, if-else, switch, while, do-while과 같은 control flow문에 해당하는 block가 존재하며 그 사용법 또한 programming 언어와 비슷하다.



<if block의 사용 예>



<if block의 parameter>

Simulink Modeling Examples

Linear System Examples

- A Linear Second Order Examples

$$\dot{x}_1 = ax_1 + bx_2 \quad (1)$$
$$\dot{x}_2 = cx_1 + dx_2 \quad (2)$$

- Notice that the system can be transformed into a scalar second order equation.

$$b\dot{x}_2 = bcx_1 + d(\dot{x}_1 - ax_1)$$

- Substitute above equation into the time derivative of (1)

$$\ddot{x}_1 = (a+d)\dot{x}_1 + (cb - ad)x_1$$

Linear System Examples

- The above equation can be simplified by the second order equation such as

$$\ddot{x} + \alpha\dot{x} + \beta x = 0$$

- The solution of above equations is

$$x(t) = k_1 e^{\lambda_1 t} + k_2 e^{\lambda_2 t} \text{ for } \lambda_1 \neq \lambda_2$$

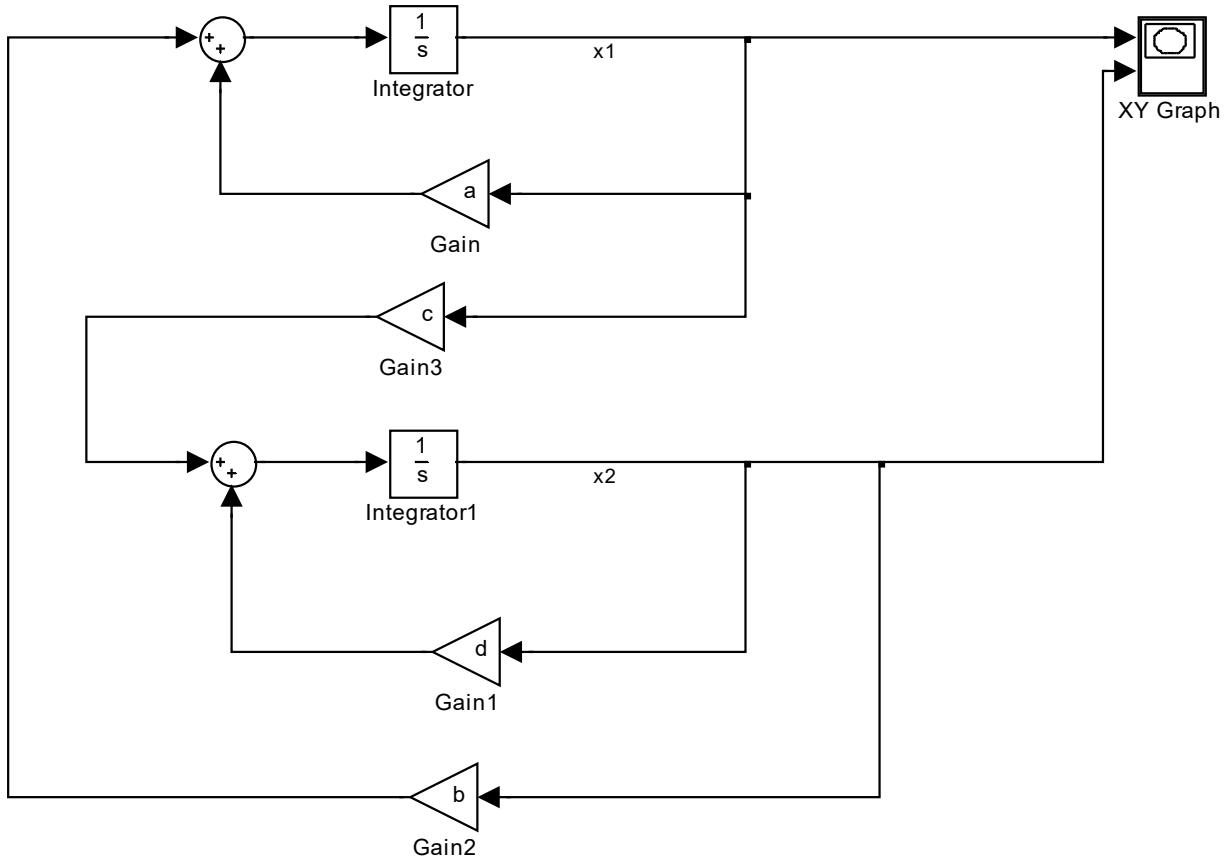
$$x(t) = k_1 e^{\lambda_1 t} + k_2 t e^{\lambda_2 t} \text{ for } \lambda_1 = \lambda_2$$

where the values of λ_1 and λ_2 are obtained by solving the characteristic polynomial defined by

$$\lambda^2 + \alpha\lambda + \beta = 0$$

Linear System Examples

- Simulink Diagram



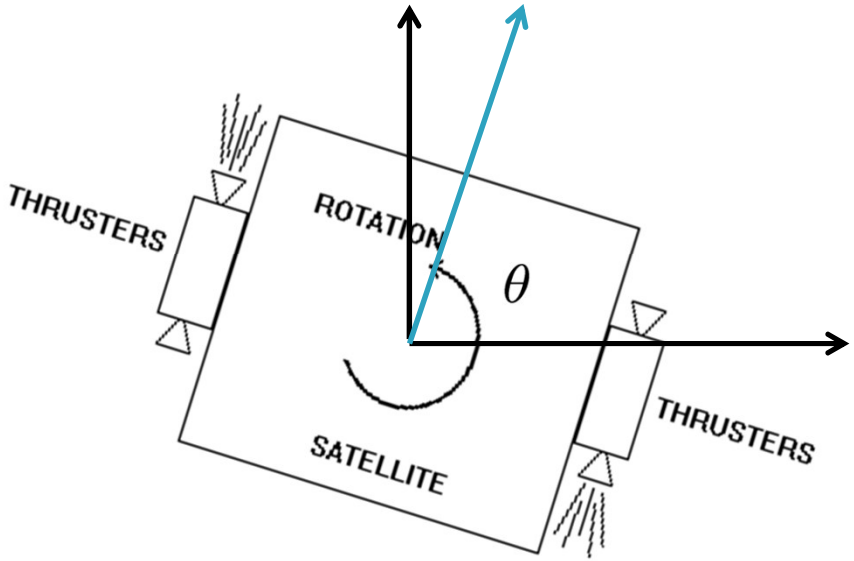
Nonlinear System Examples

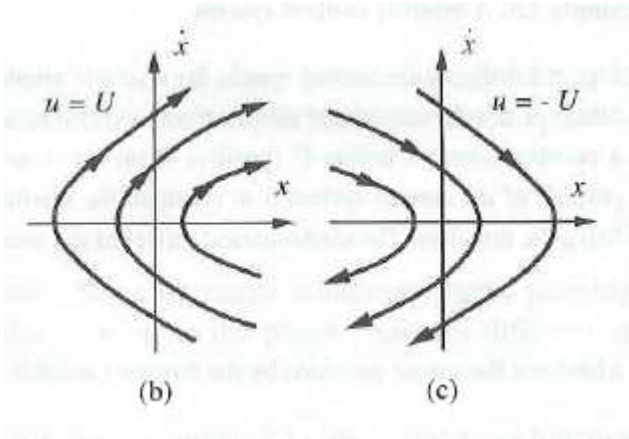
- A Satellite Control System

$$\ddot{\theta} = u$$

$$u(t) = \begin{cases} -U \text{ if } \theta > 0 \\ U \text{ if } \theta < 0 \end{cases}$$

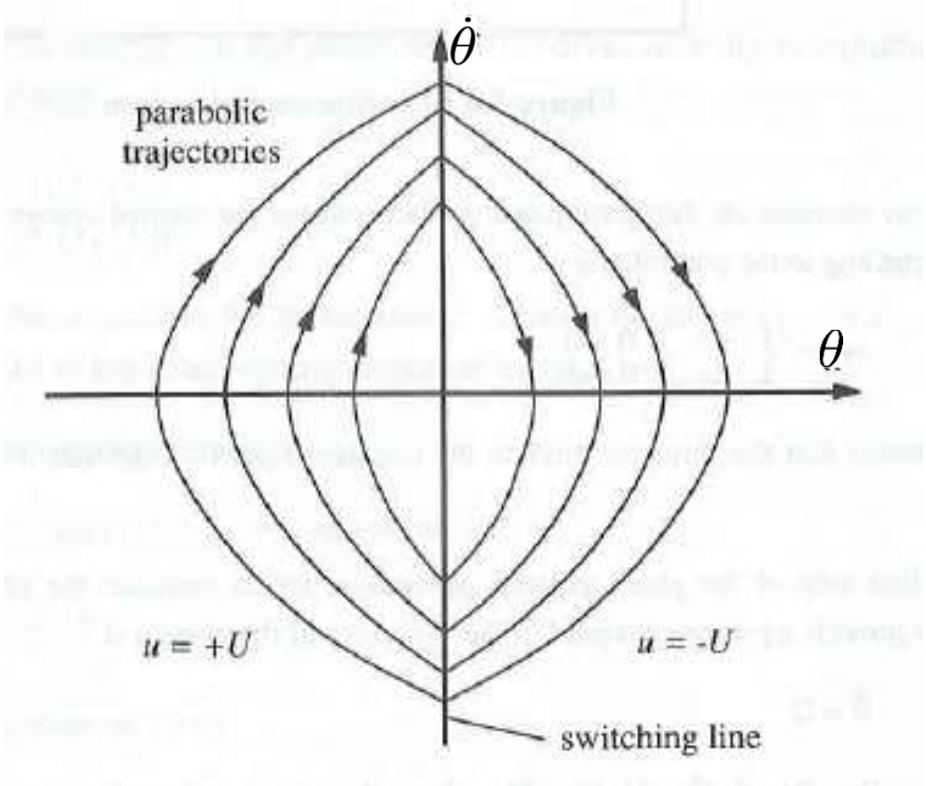
$\dot{\theta}$





Nonlinear System Examples

- Model satellite dynamics and control with Simulink.



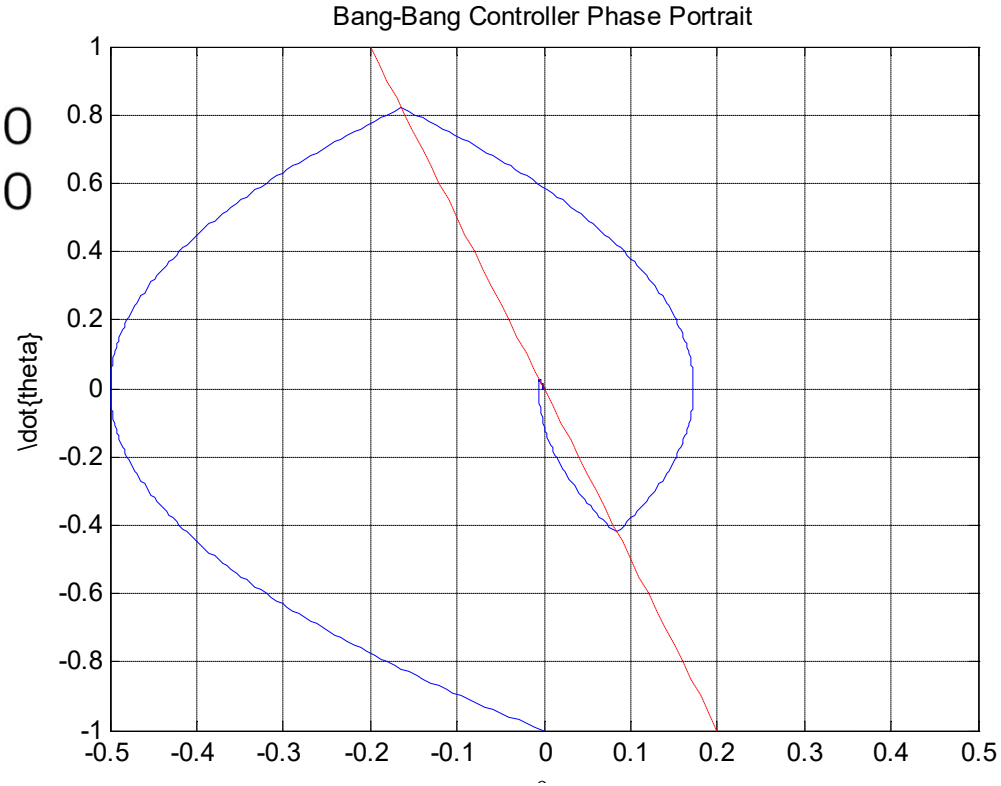
Nonlinear System Examples

- How can we make the system to converge using bang-bang controller?

- Solution

$$u(t) = \begin{cases} U & \text{if } (-\dot{\theta} - a\theta) > 0 \\ -U & \text{if } (-\dot{\theta} - a\theta) < 0 \end{cases}$$

$$(a > 0)$$



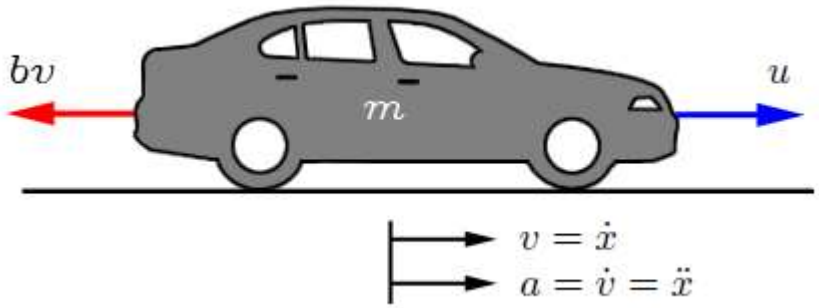
Simulink Modeling Resources

- Control Tutorials for Matlab & Simulink
  - https://ctms.engin.umich.edu
  - 미시건 대학교에서 제작한 Matlab 기반 제어시스템 예제를 소개한 웹사이트
  - 간단한 자동차 모델을 비롯하여 다양한 시스템에 대한 모델과 제어기 설계를 다루고 있음
  - 시뮬링크를 활용한 모델링 예제 제공

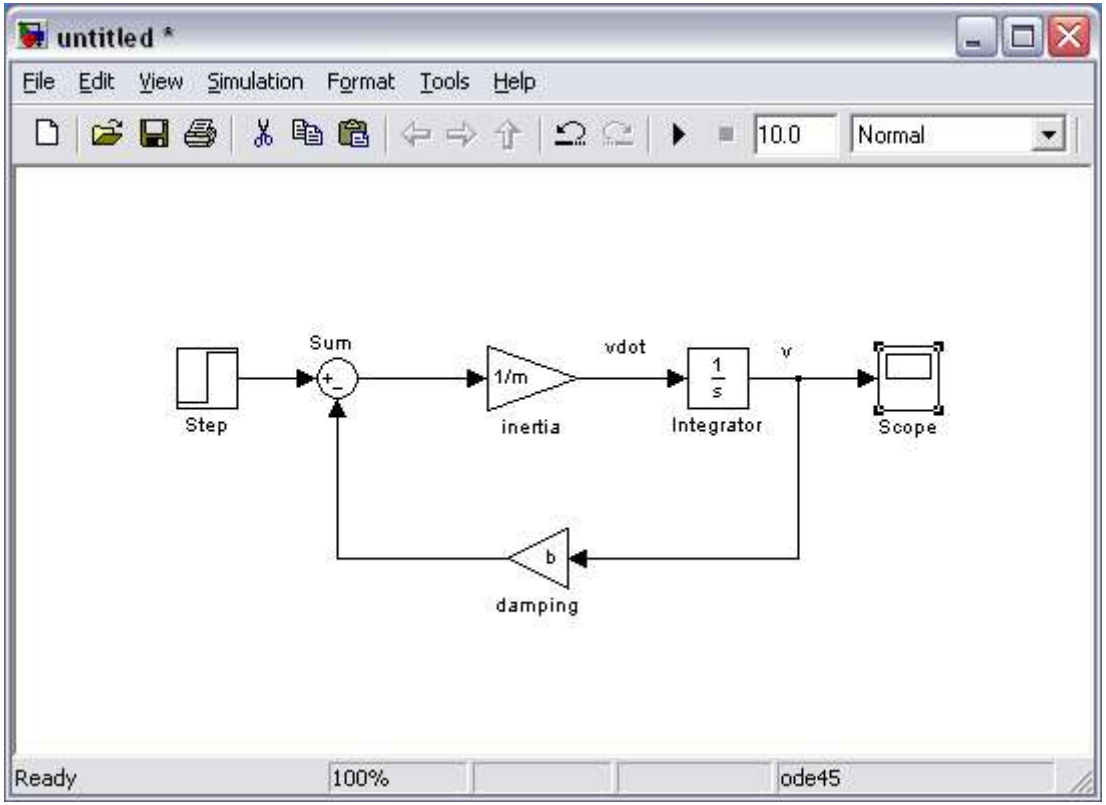
Cruise Control Example

- Physical setup

$$m\dot{v} = u - bv$$
$$\dot{x} = v$$




```
m = 1000 kg
b = 50 N.sec/m
u = 500 N
```

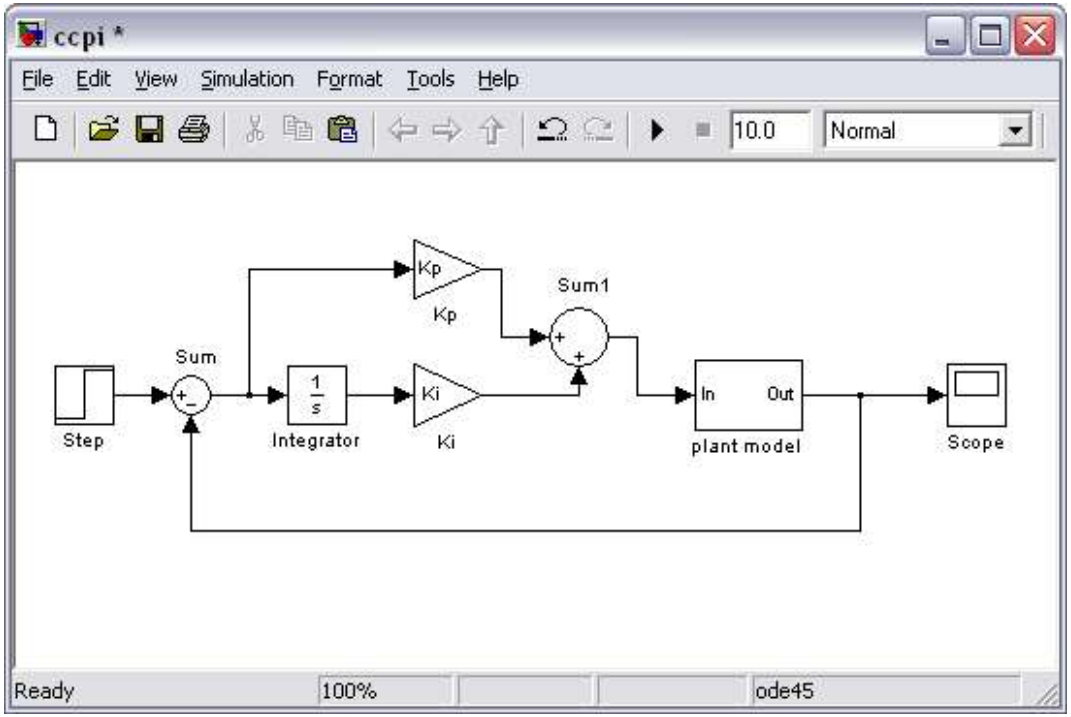


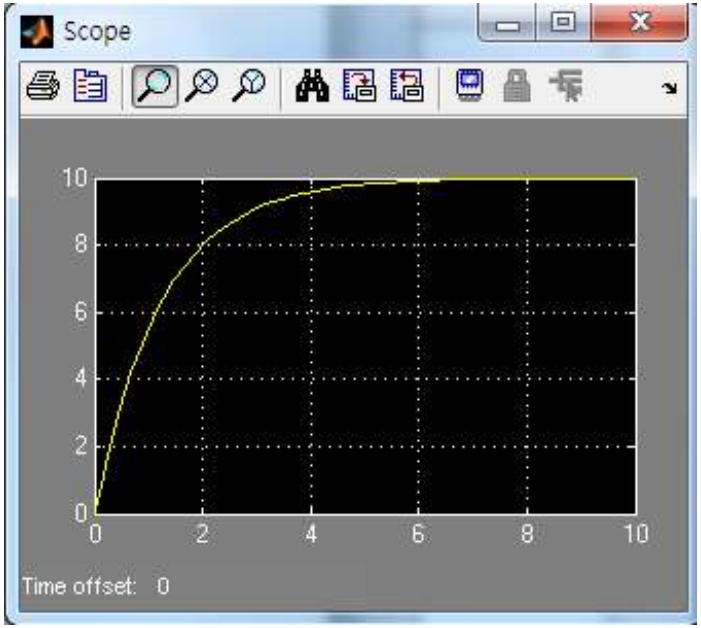
Cruise Control Example

- PI Controller

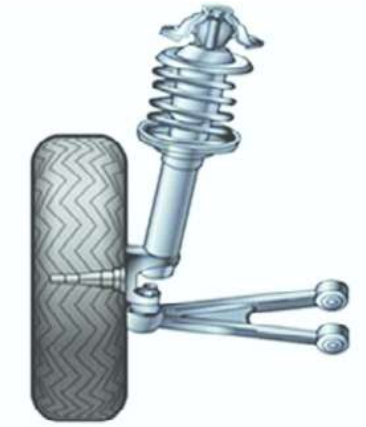
```
m = 1000;
b = 50;

Kp = 800;
Ki = 40;
```





Suspension System



- System Parameters

$$M_1\ddot{X}_1 = -b_1(\dot{X}_1 - \dot{X}_2) - K_1(X_1 - X_2) + U$$

$$M_2\ddot{X}_2 = -b_1(\dot{X}_1 - \dot{X}_2) + K_1(X_1 - X_2) + b_2(\dot{W} - \dot{X}_2) + K_2(W - X_2) - U$$

(m1) body mass 2500 kg

(m2) suspension mass 320 kg

(k1) spring constant of suspension system 80,000 N/m

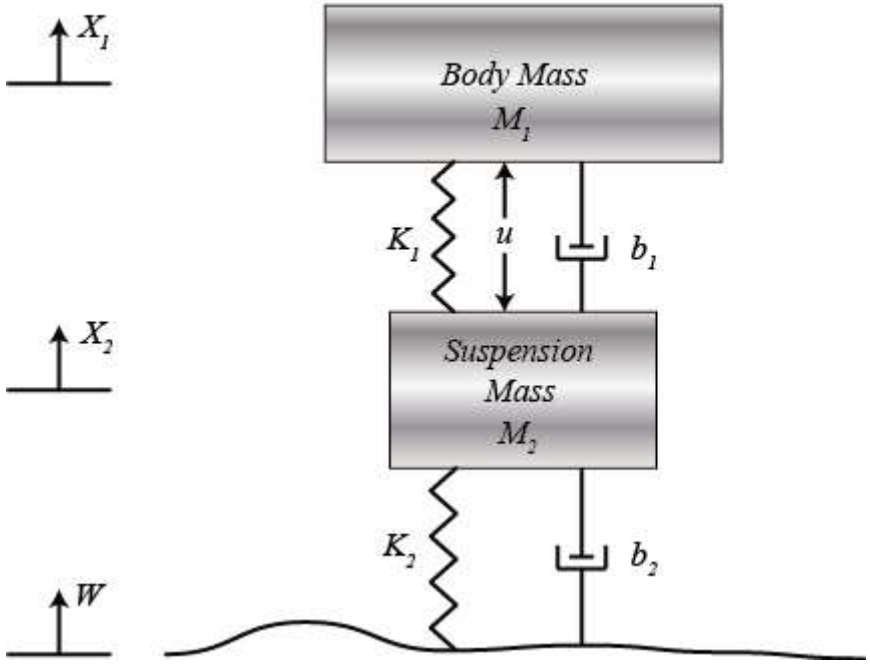
(k2) spring constant of wheel and tire 500,000 N/m

(b1) damping constant of suspension system 350 N.s/m

(b2) damping constant of wheel and tire 15,020 N.s/m

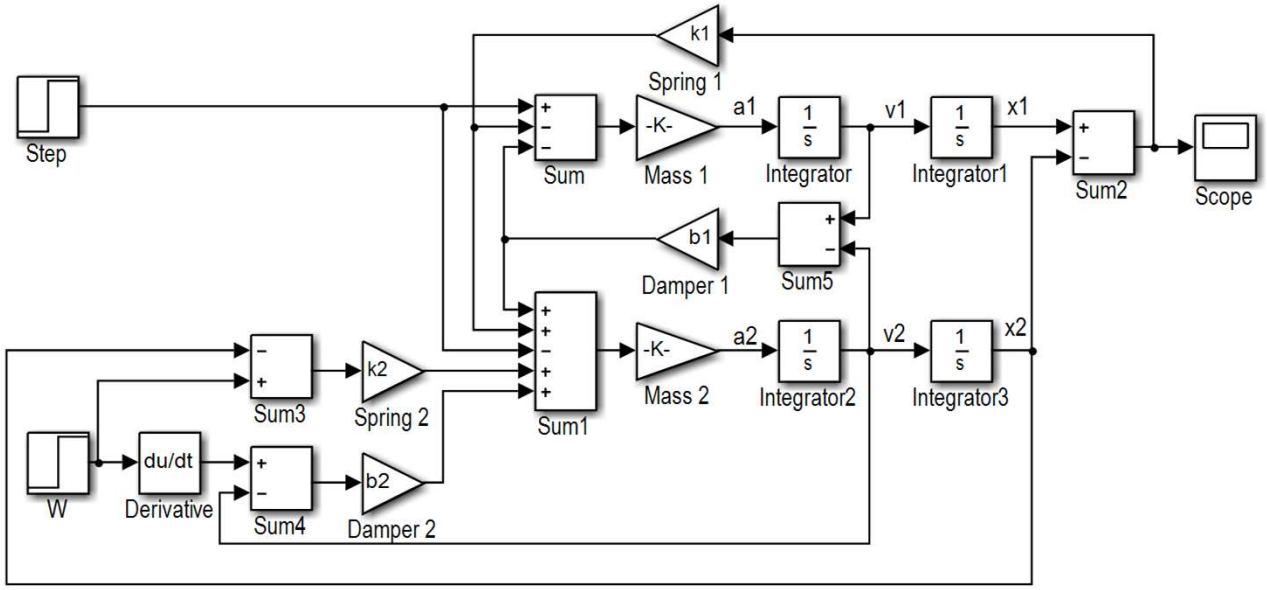
(u) control force = force from the controller we are going to design

Model of Bus Suspension System (1/4 Bus)



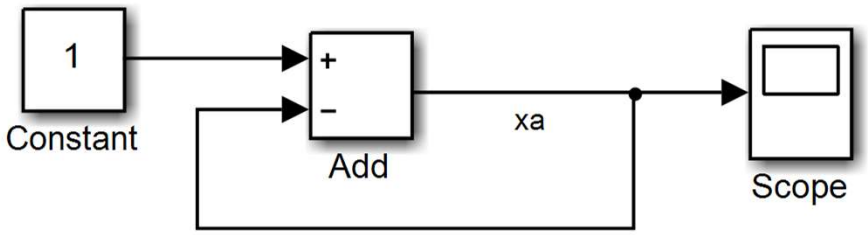
Suspension System

- Simulink Diagram



Algebraic Loop Error (or Warning)

- Algebraic Loop Error
  - Feedback Loop 를 만들 때 발생할 수 있는 모델링 오류
  - Direct Feedthrough : 출력이 결정되는데 “현재”의 입력이 영향을 미치는 블록을 의미함. (e.g. Add, Gain, Sum 등의 Math Operation 과 Signal Routing 라이브러리)
  - Non-direct feedthrough block : Integrator, Memory, Delay, Transfer function 등의 Continuous, Discrete 라이브러리에 있는 블록
  - Direct Feedthrough 블록만으로 구성된 출력신호가 입력에 다시 연결되었을 때 Algebraic Loop Error (or Warning) 가 발생한다.
- Example 1



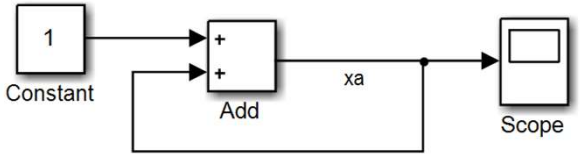
!! 수학적으로 **xa = 1 – xa** 이므로 **Xa = 1/2** 가 솔루션이지만 시뮬링크에서는 **Algebraic Loop warning** 을 발생함

Algebraic Loop Error (or Warning)

- **Effect of Algebraic Loops in a Model**
  - You cannot generate code for the model.
  - The Simulink algebraic loop solver might not be able to solve the algebraic loop.
  - While Simulink is trying to solve the algebraic loop, the simulation can execute slowly.
  - For most models, the algebraic loop solver is computationally expensive for the first time step. Simulink solves subsequent time steps rapidly because a good starting point for x_a is available from the previous time step.

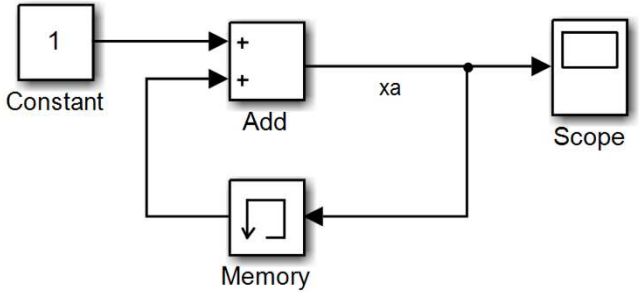
Algebraic Loop Error (or Warning)

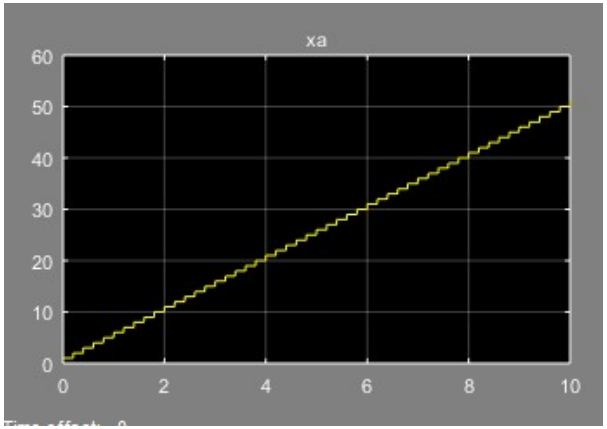
- Example 2



`!! xa = xa + 1` 이므로 프로그램상 간단한 누적기 이지만 마찬가지로 **`Algebraic Loop Error`** 를 발생함

  - 해결책으로 Memory 또는 Delay 블록을 사용 ➔ 누적 가능





  - 그러나, Memory, Delay 블록은 Discrete Time 시스템 블록이므로 Triggered System으로 모델링 하는 것이 바람직함