Control System Design for Automated Driving

Lecture 03

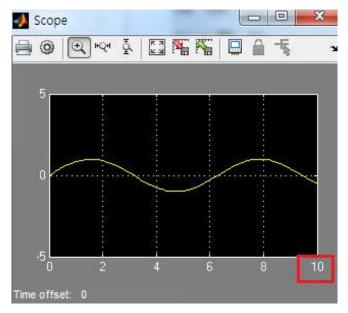




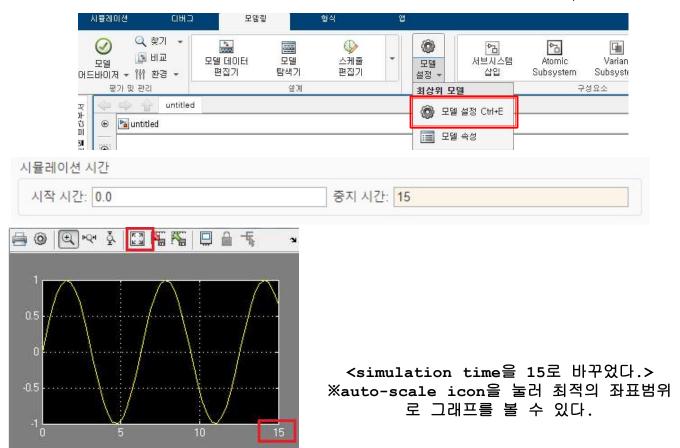


- Simulation Time
 - Simulink의 기본적인 simulation time은 10이다.





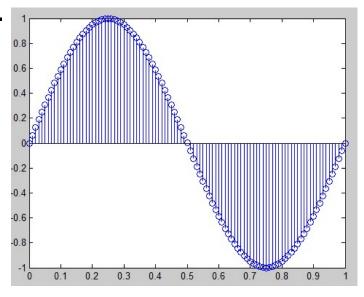
- Simulation Time 바꾸기
 - Simulation 메뉴의 "모델링/모델설정"에서 start time과 stop time을 결정할 수 있다.(stop time을 "inf"라고 줄 경우 사용자 가 멈추고 싶을 때까지 simulation을 진행한다.)



- time step과 step size
 - Time step
 - Model을 구성하는 states와 outputs가 계산되는 연속적인 시간을 말한다.
 - Step size
 - Time step간의 크기를 말한다.

※예를 들어

matlab 에서 >>Time = 0:0.01:1 이라고 할 때 0.01, 0.02, 0.03 ... 1 을 time step이라고 하며 각 time step간의 간격인 0.01을 step size라고 한다.



<각 점들은 time step이며 점들간의 간격이 step size 이디

- ▶ 시뮬레이션 결과를 얻기위한 Solver 지정 방법
 - Simulation model에 적합한 solver를 선택해야 더 효율적 이고 정확한 결과를 얻을 수 있다.
 - Solver는 메뉴의 "모델링/모델설정" 에서 선택할 수 있다.

솔버 선택	택			
유형:	가변 스텝	-	솔버: 자동(솔버 자동 선택)	·
	가변 스텝			
▶ 솔버 시	고정 스텝			

◦ 솔버 세부정보를 클릭하면 아래와 같은 정보를 얻을 수 있다.

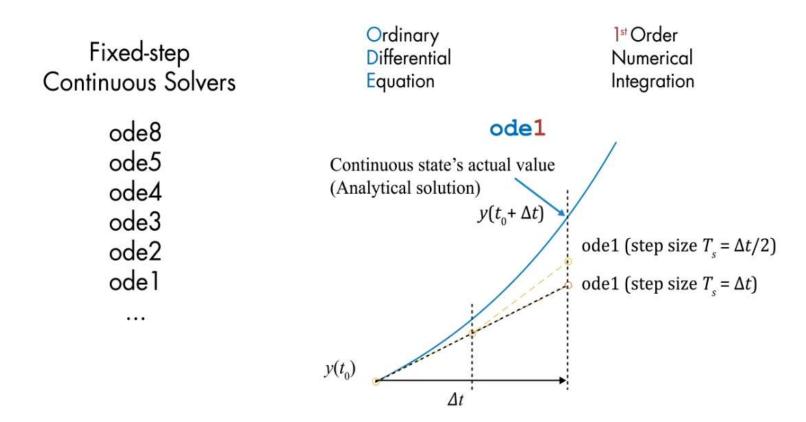
최대 스텝 크기: auto	상대	내 허용오차:	1e-3	
최소 스텝 크기: auto	절대	내 허용오차:	auto	
초기 스텝 크기: auto	1	절대 허용오차 제	ト동 스케일링	
형태 보존:		모두 사용 안 함		
연속 최소 스텝 수:	1	1		
점교차 옵션				
영점교차 제어: 로컬 설정 시	용	▼ 알고리즘:	비적응형	
시간 허용오차: 10*128*eps		신호 임계	Zt. auto	

- Solver Type
 - 고정스텝 솔버(Fixed-step Solver)
 Simulation의 시작부터 끝까지 고정된 step size를 가지고 model states를 풀어간다.
 - 가변스텝 솔버 (Variable-step solver)
 Simulation하는 동안 step size를 바꾸어가며 model의 states를 풀어간다.

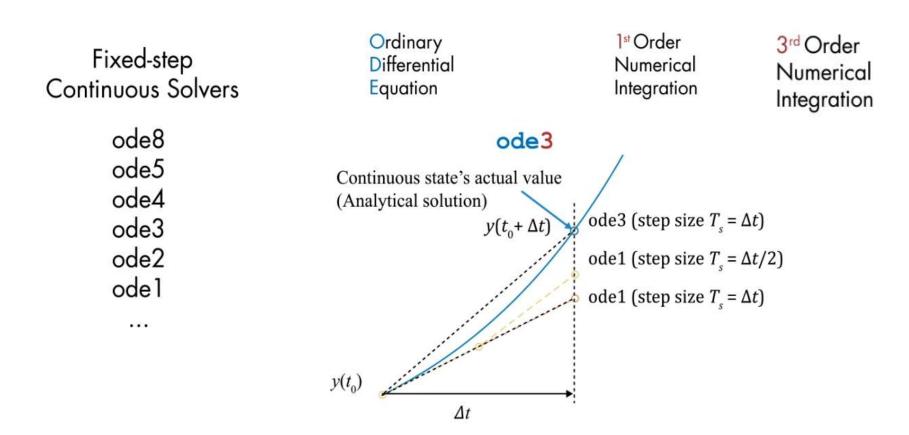
States가 급격히 바뀌는 경우 step size를 줄여 정밀도를 높이고, States의 변화가 완만하면 step size를 늘려 불필요한 계산 량을 줄여준다.

	Continuous solver (Model contains continuous states)	Discrete solver (No continuous states)
Fixed - step	ode8, ode5,ode4, ode3, ode2, ode1	FixedStepDiscrete
Variable - step	ode 45, ode 23, ode 113, ode 15s	VariableStepDiscrete

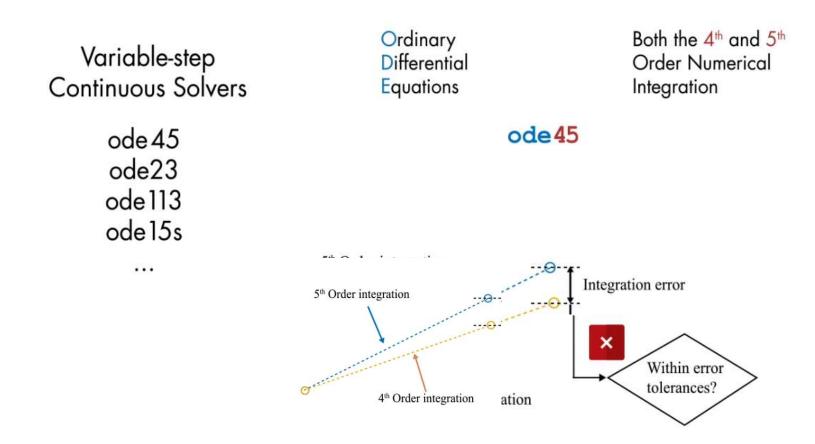
Fixed Step Solver



Fixed Step Solver



Variable Step Solver

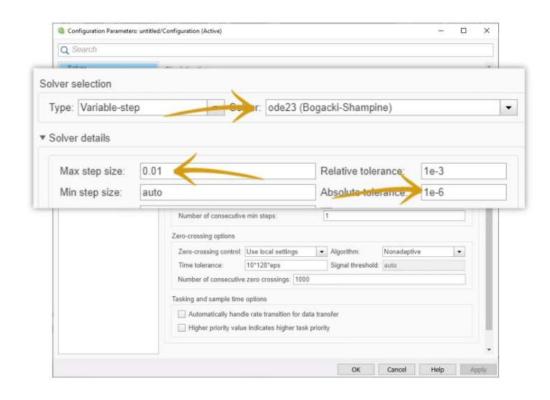


Variable Step Solver

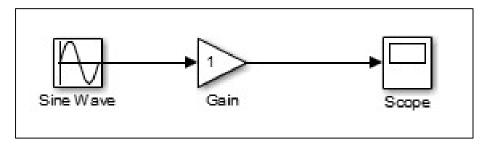
Variable-step Continuous Solvers

> ode 45 ode 23 ode 113 ode 15s

> > • • •



- Fixed Step Solvers VS Variable Step Solvers
 - Use the example below.



- Use the fixed step solver with step size 1 and 0.1.
 - Compare the two different sine wave output.
- Use the variable step solver.
 - Check the workspace variable "tout" to find the step size.

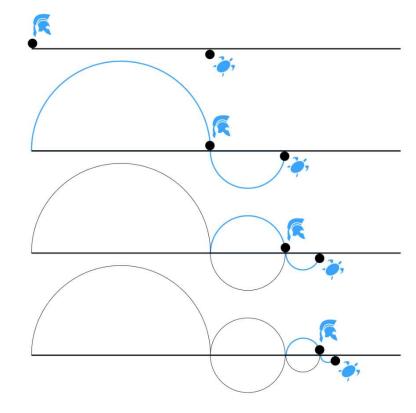
- Solver: field 옆의 popup menu를 클릭하면 구체적으로 적용할 solvers를 선택할 수 있다.
 - discrete(no continuous states)
 순수한 discrete-time models를 해결하기 위한 solver이다.
 만일 continuous와 discrete states를 모두 포함한 models를 풀기 위해서는 discrete를 선택해서는 안 된다.
 - others
 - Continuous states를 표현하는 ordinary differential equations(ODEs)에 대한 수치해를 구해줄 수 있다.
 - 여러 가지 종류의 솔버들 중에서 선택할 수 있다. (e.g. ODE45)
- Max step size
 - Solver Type이 Variable-step인 경우에 선택할 수 있는데, variable step solver가 취할 수 있는 가장 큰 step size 를 의미한다.
 - 일반적으로 Simulink가 자동으로 잡아주는 "auto"로 사용하는 것이 바람직하다.

Zero-Crossing detection

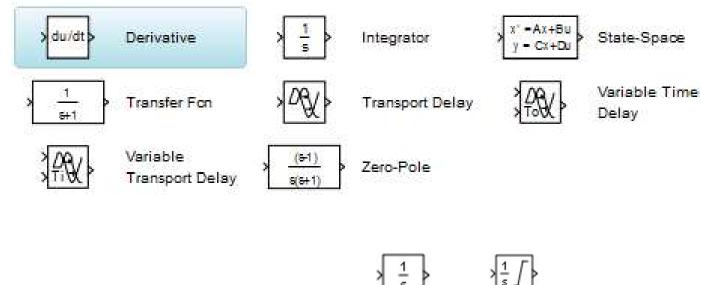
- When a variable step solver is used, simulation can take many small steps in the vicinity of the a discontinuity.
- This can lead to excessive simulation times.
- Zero-Crossing Detection can cause simulations to halt be efore the intended completion time.
- Example : "Bouncing Ball Model"
 - On the command window, type "demo", then search "Bouncing Ball"
 - Try the simulation time upto 25 sec.
 - Simulation stops before 21 sec because of zero-crossing detection.
 - This occurrence of infinite number of events is called "Zeno Phenomenon".
 - To fix the problem, change the "zero-crossing options/ Algorithms: "from "Nonadaptive" to "Adaptive"
 - Or Change the "zero-crossing options" to "Disable All" option.

Zeno's Paradox

- 아킬레우스와 거북이
 - <u>아킬레우스</u>가 거북이보다 10배 빨리 이동할 수 있다고 가정하고, 아킬레우스보다 거북이를 100m 앞에서 출발 시킨다.
 - 아킬레우스가 100m를 달려가 면 거북이는 10m를 가고, 거 북이를 따라잡기 위해 아킬레 우스가 10m를 가면 그동안 거 북이는 1m를 나아간다.
 - 아킬레우스가 거북이를 따라 잡기 위해 달리는 동안 거북이 역시 움직이므로 아킬레우스 는 영원히 거북이를 따라잡을 수 없다.



Continuous Time Library



- Integrator Blocks
 - Outputs the value of the integral of its input signals with respect to time.
 - Initial conditions are required. (zero by default)
 - User can create output saturation limits.

- Signals Routing Library
 - ∘ Mux 🔥
 - · Combine several input signals into vector
 - ∘ Demux 🕴
 - Extracts the components of an input signal and outputs the components as separate signals

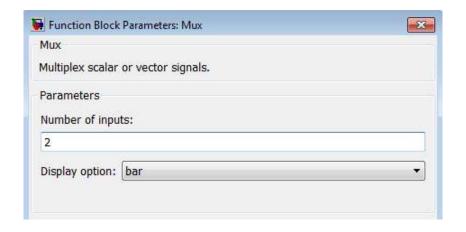
Bus Assignment

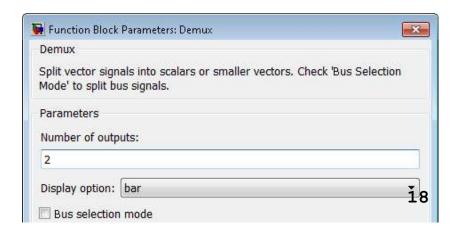
Data Store

Memory

Demux

Manual Switch





Bus Creator

Data Store

Environment

Controller

Goto Tag

Visibility

Selector

Read

Bus Selector

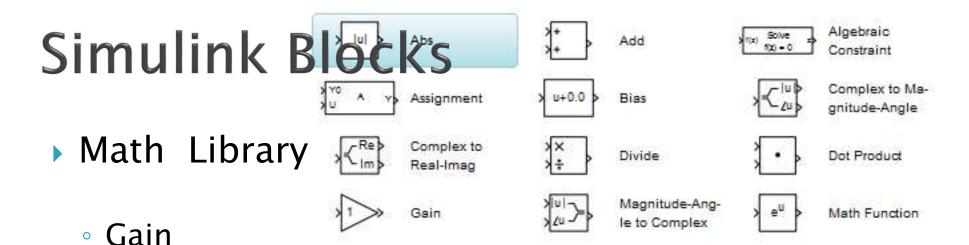
From

Switch

Data Store Write

Index Vector

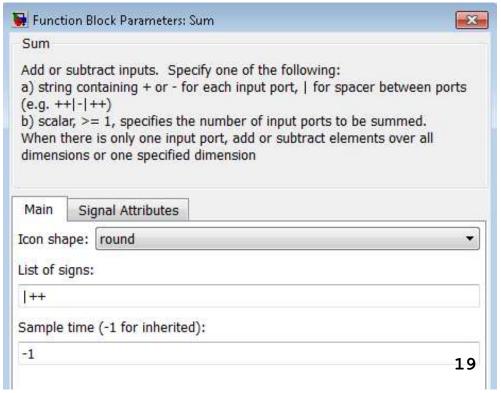
Multiport Switch

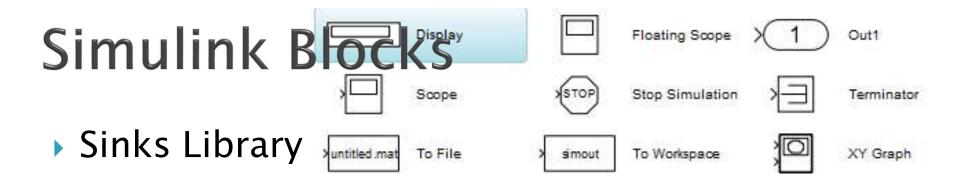


- Multiply input by a constant value
- Sum

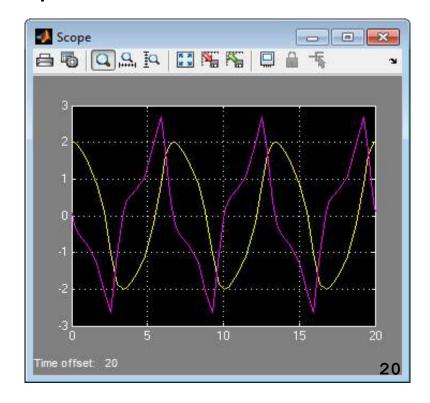


Add or subtract inputs.

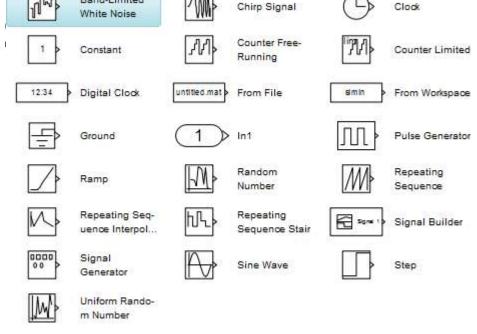




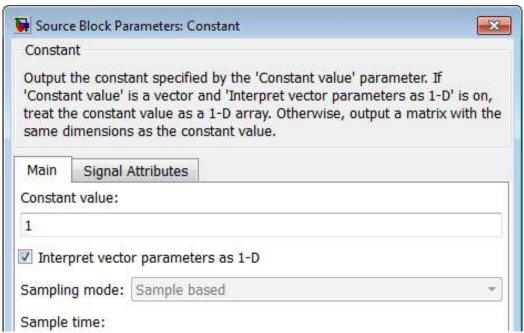
- Scope
 - Displays input signals with respect to simulation time.



- Sources Library
 - Constant
 - Generate constant value signals.



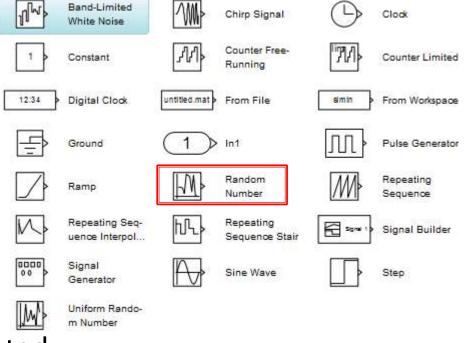
Band-Limited

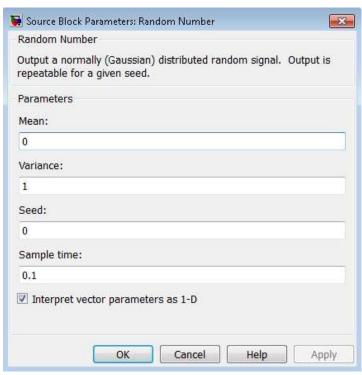


- Sources Library
 - Random Number

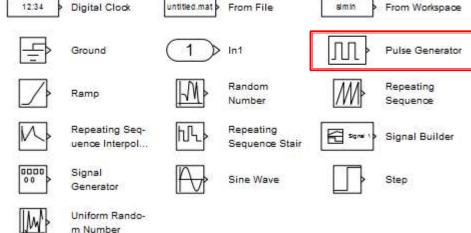


- Generate normally distributed random numbers.
- To generate uniformly distributed random numbers, use the Uniform Random Number block.





- Sources Library
 - Pulse generator



Chirp Signal

Counter Free-

Running

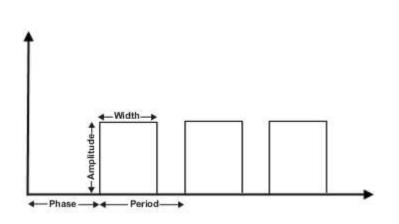
Counter Limited

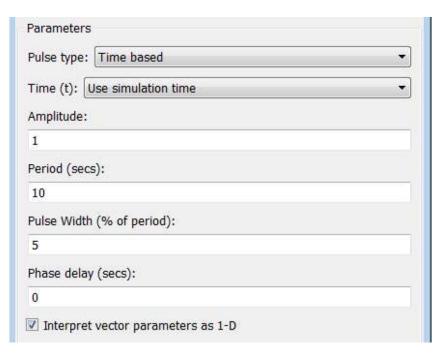
Band-Limited

White Noise

Constant

Generate square wave pulses at regular intervals.



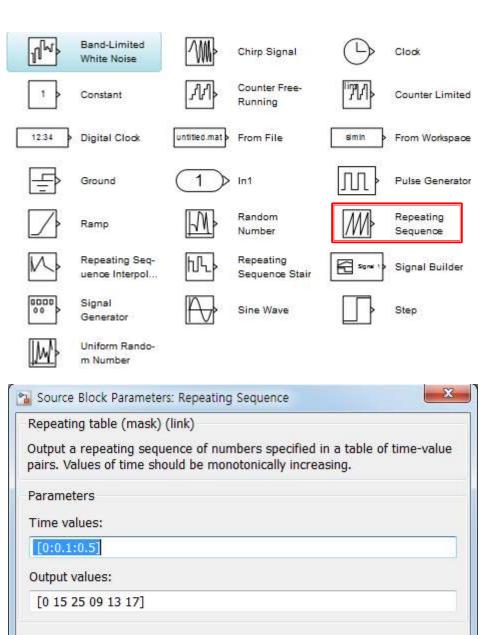


Sources Library



- Repeating Sequence
 - Generate arbitrarily shaped periodic signal.
 - Create a model as below.





OK

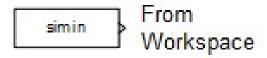
Cancel

Help

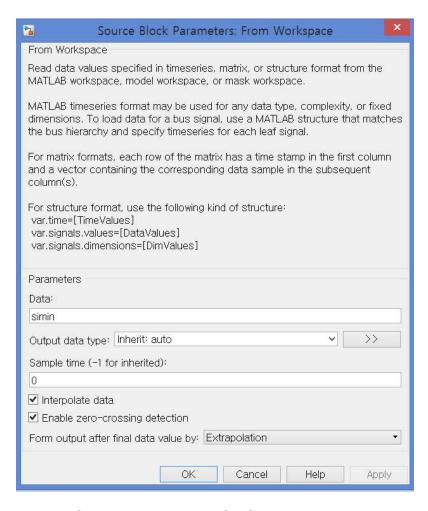
Apply

From Work Space Block

- Sources Library
 - From Workspace



- Reads data from workspace and outputs signal
- Input data format
 - Matrix format: first column contains the time, and the second column contains the values
 - Structure format :
 - var.time=[TimeValues]
 - var.signals.values=[DataValues]
 - var.signals.dimensions=[DimValues]



From Work Space Block

- Example (Matrix format)
 - In matlab command window type the following.

```
>> t = 0:0.1:10;

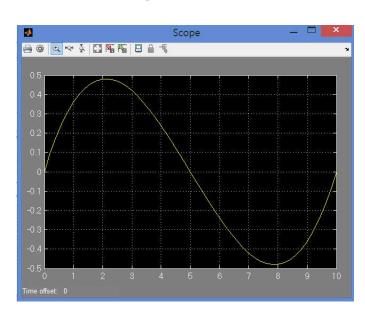
>> y = 0.01*t.*(t-5).*(t-10);

>> data(:,1) = t';

>> data(:,2) = y';
```

Then create the following block diagrams





From Work Space Block

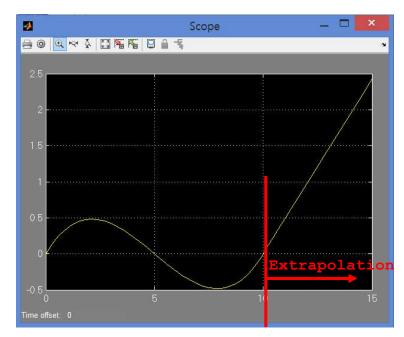
- Example (Structure Format)
 - In matlab command window type the following.

```
>> t = 0:0.1:10;
>> y = 0.01*t.*(t-5).*(t-10);
>> data.time = t';
>> data.signals.values = y';
```

This will generate the same output with previous

simulink model

- Change the simulation time to 15 sec and run.
- After the pre-defined signal runs out, the signal is "extrapolated".

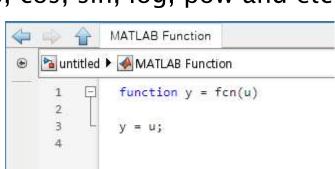


- User-Defined Functions
 - MATLAB Function





- u → The input to the block. If u is a vector, u(i) represents the ith element of the vector.
- Numerical constants
- Arithmetic (+ * / ^), Relational(==!=> < >= <=),
 Logical (&& ||!) operators.
- Mathematical functions: abs, cos, sin, log, pow and etc.
- Workspace variables.



<FunctionName>

C Caller

(1) initialize

C Function

Interpreted

MATLAB Fon

Function

System MATLAB System

Initialize Function Interpreted MATLAB

Function Caller

matlabfile >

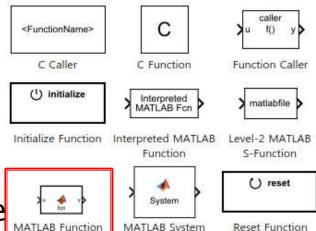
Level-2 MATLAB

S-Function

() reset

Reset Function

- User-Defined Functions
 - MATLAB Function block Example



2. 다음과 같이 함수 선언문을 편집하여 함수 입력과 출력을 정의합니다.

```
function [mean, stdev] = stats(vals)
```

이 선언문은 stats라는 함수를 3개의 변수로 정의합니다. 선언문은 입력 인수 vals 및 두 개의 출력 인수 mean과 stdev를 정의합니다.

3. 함수 선언문 이후 하나의 새 줄에 다음 코드를 추가합니다.

```
% Calculates a statistical mean and a standard
% deviation for the values in vals.

len = length(vals);
mean = avg(vals,len);
stdev = sqrt(sum(((vals-avg(vals,len)).^2))/len);
plot(vals,"-+");

function mean = avg(array,size)
mean = sum(array)/size;
```

- 4. 블록을 종료합니다. 블록은 포트 이름을 업데이트합니다. 함수 출력 mean과 stdev는 블록 출력 포트 mean과 stdev에 대응하고, 함수 입력 vals는 블록 입력 포트 vals에 대응합니다.
- 5. 다음과 같이 MATLAB Function 블록에 대한 연결을 완료합니다.

