Introduction to Simulink for Dynamic System Modeling and Simulation

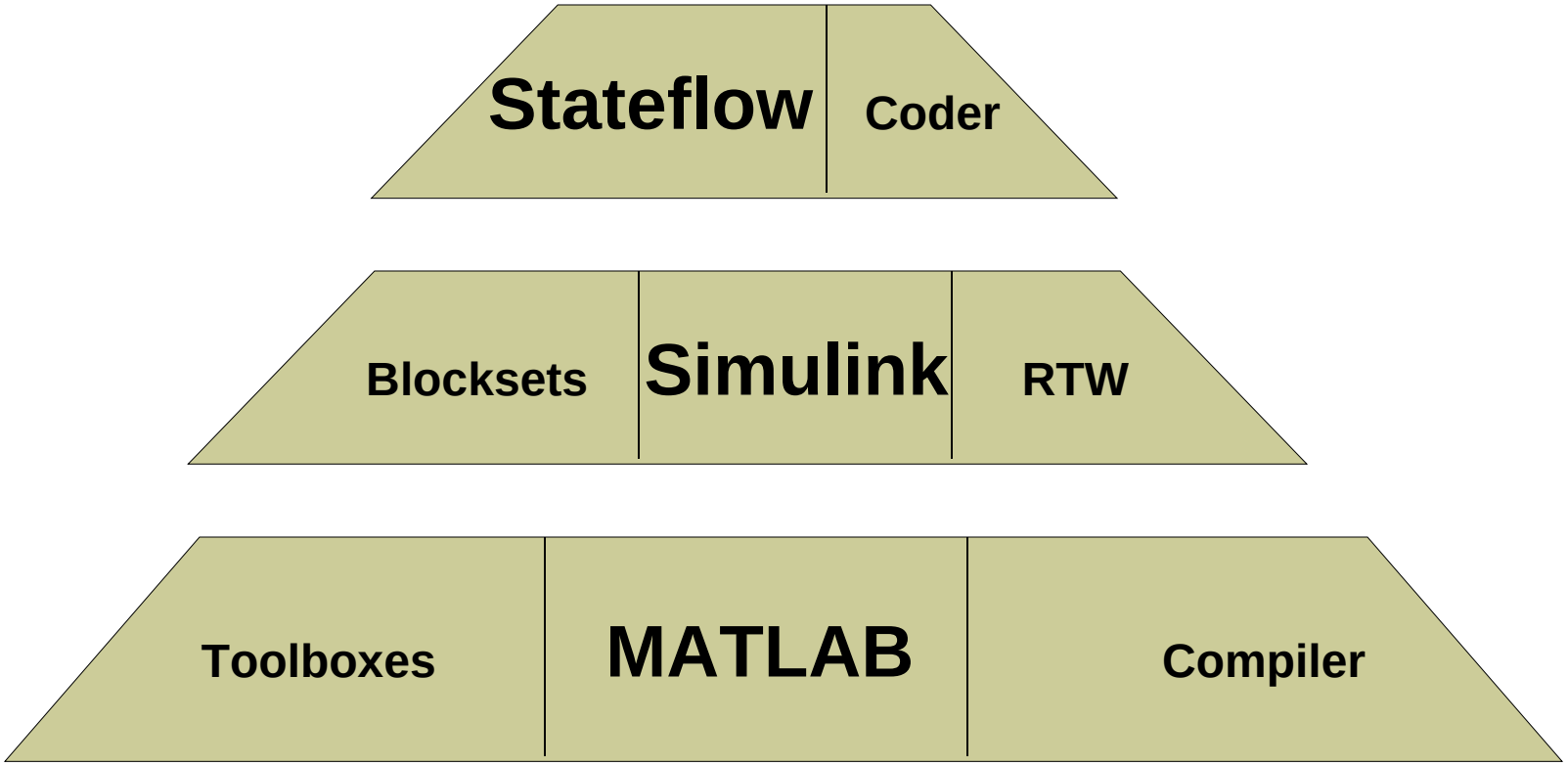
Atilla Dogan

XX4378 and XX5378 – Introduction to UVS

September 13, 2017

Fall 2017

The MATLAB Product Family



What is Simulink

- Simulink is a tool that allows you to model and simulate a wide variety of dynamic systems
- Among other things, this means
  - Block diagram editing
  - Nonlinear Simulation
  - Continuous and discrete simulation
  - Hybrid simulation
  - Asynchronous (non-uniform sampling)
  - Integration with MATLAB, Extensions, Blocksets & Toolboxes

What can you model with Simulink

- Just about anything you can model mathematically
  - Aircraft and spacecraft dynamics and systems
  - Automotive systems
  - Ship systems
  - Communications and satellite systems
  - Monetary systems
  - Biological systems

What we are going to learn

- Building and running models
- Viewing Data
- Masking
- GUI interfaces
- And more

Building a model

- There are 4 steps
  - Collecting and connecting blocks
  - Simulating the system
  - Analyzing the results
  - Refining the model

  - Each of these steps will be done several times

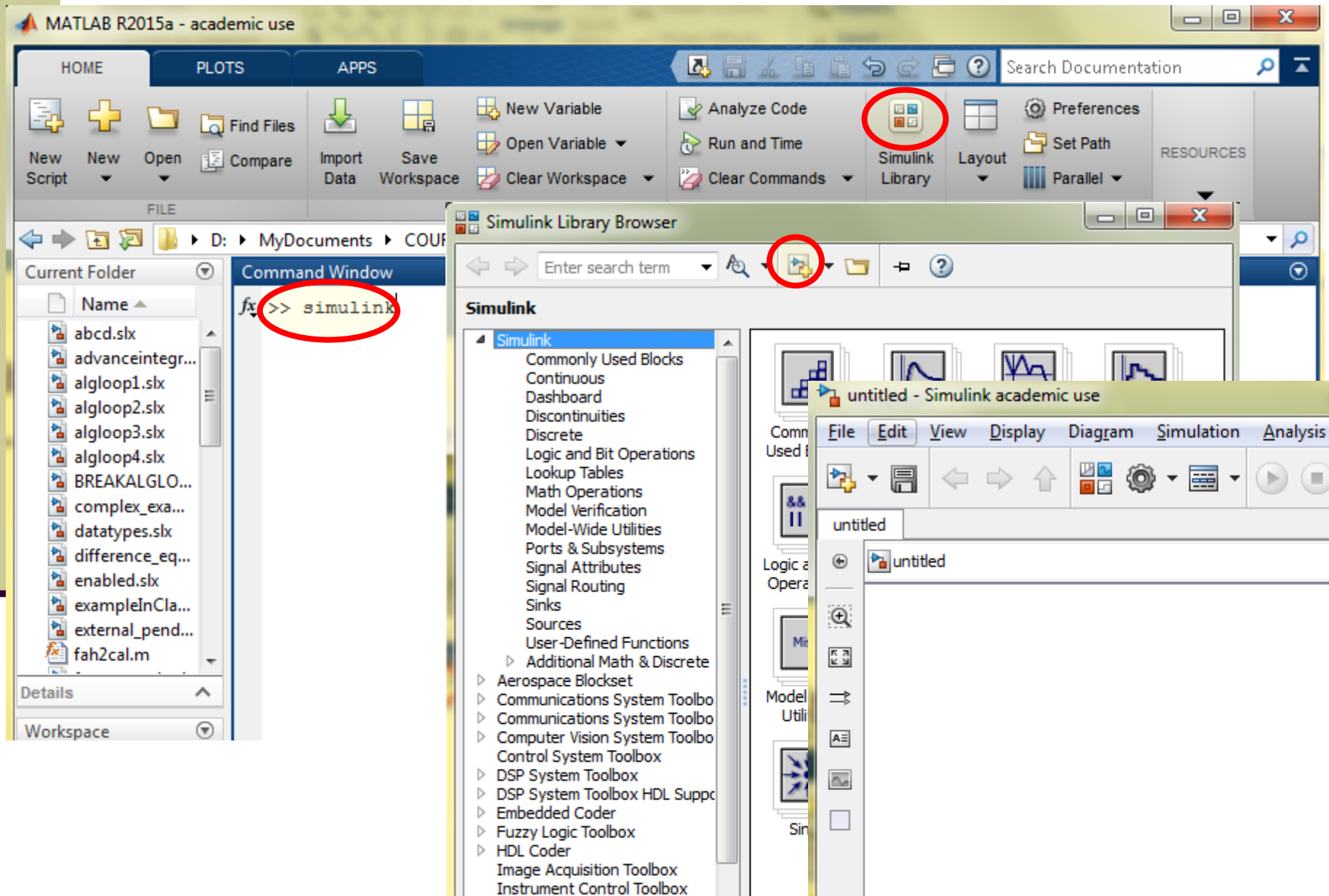
Creating a model

- Simulink basics
- Starting Simulink
- The Simulink block library
- Editing models
- Setting block dialog parameters
- Creating subsystems
- Saving (SLX File format)
- Annotations and signal labels
- Printing
- Getting Help
- Tips for building models

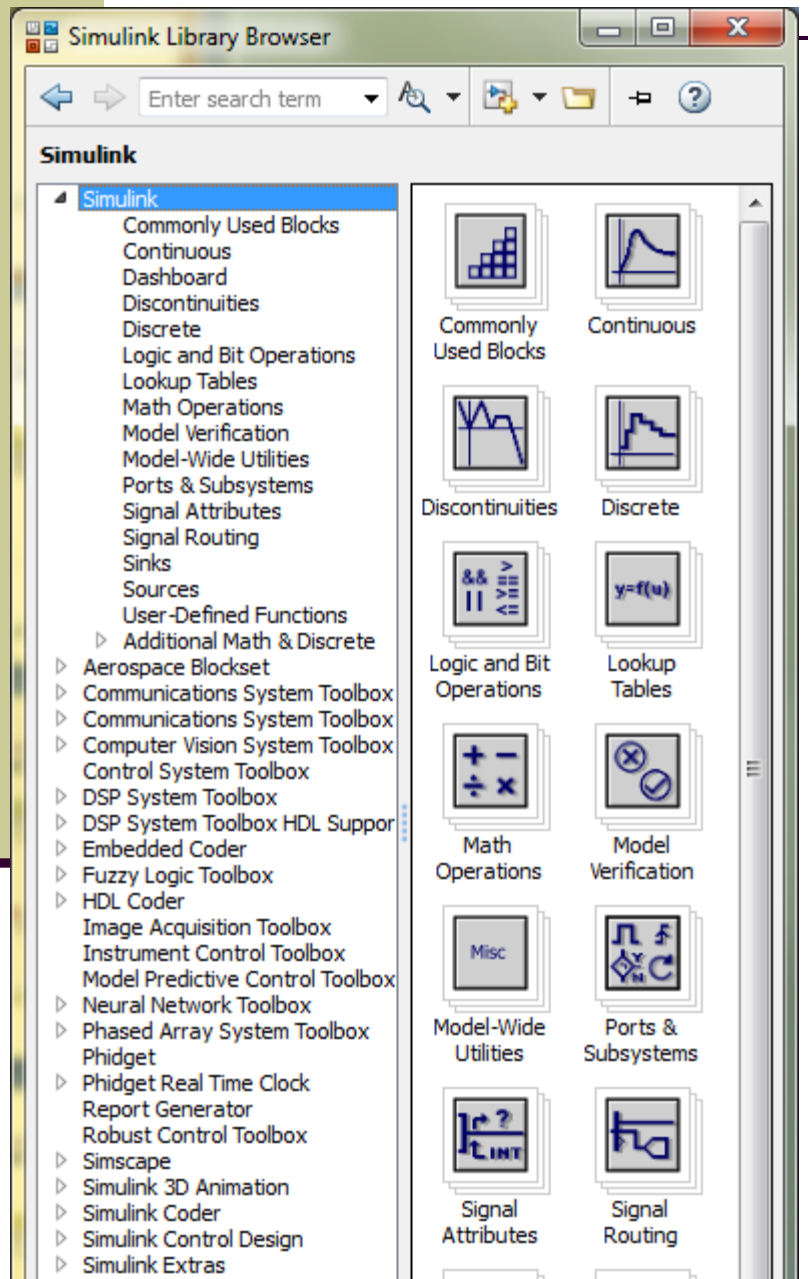
Simulink basics

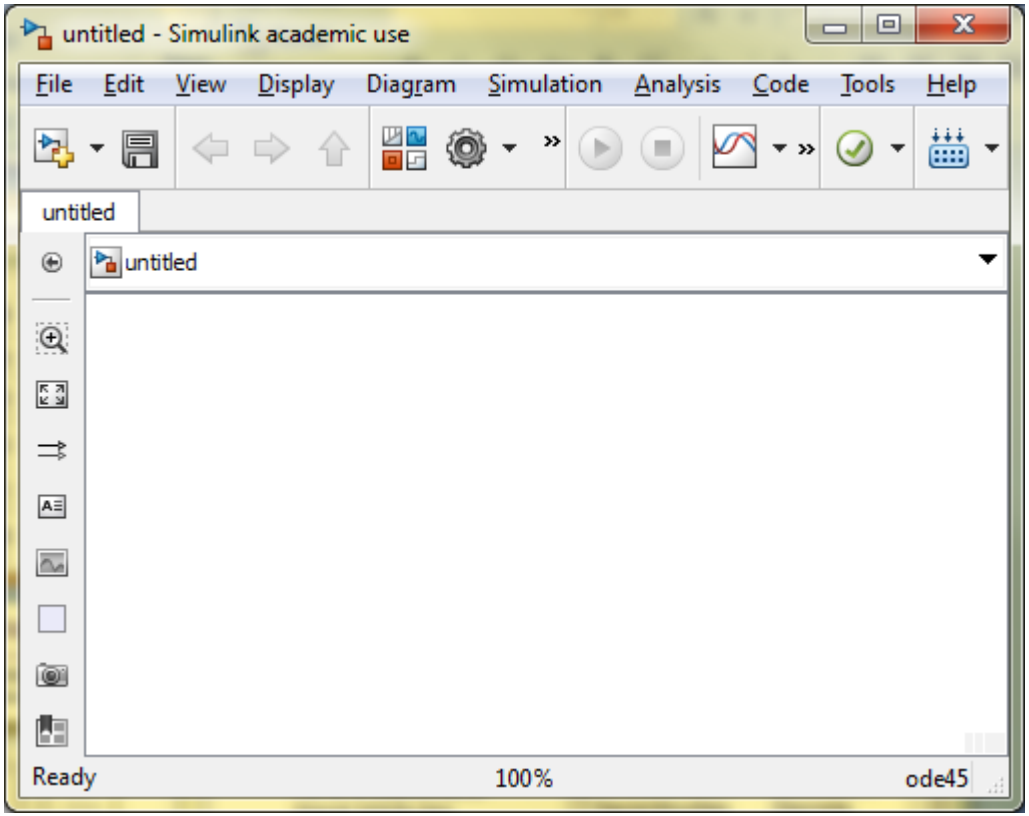
- It has its own special windows for block diagrams
- It works by setting up a dialogue between the system and the solver
- It shares MATLAB’s workspace

Starting Simulink



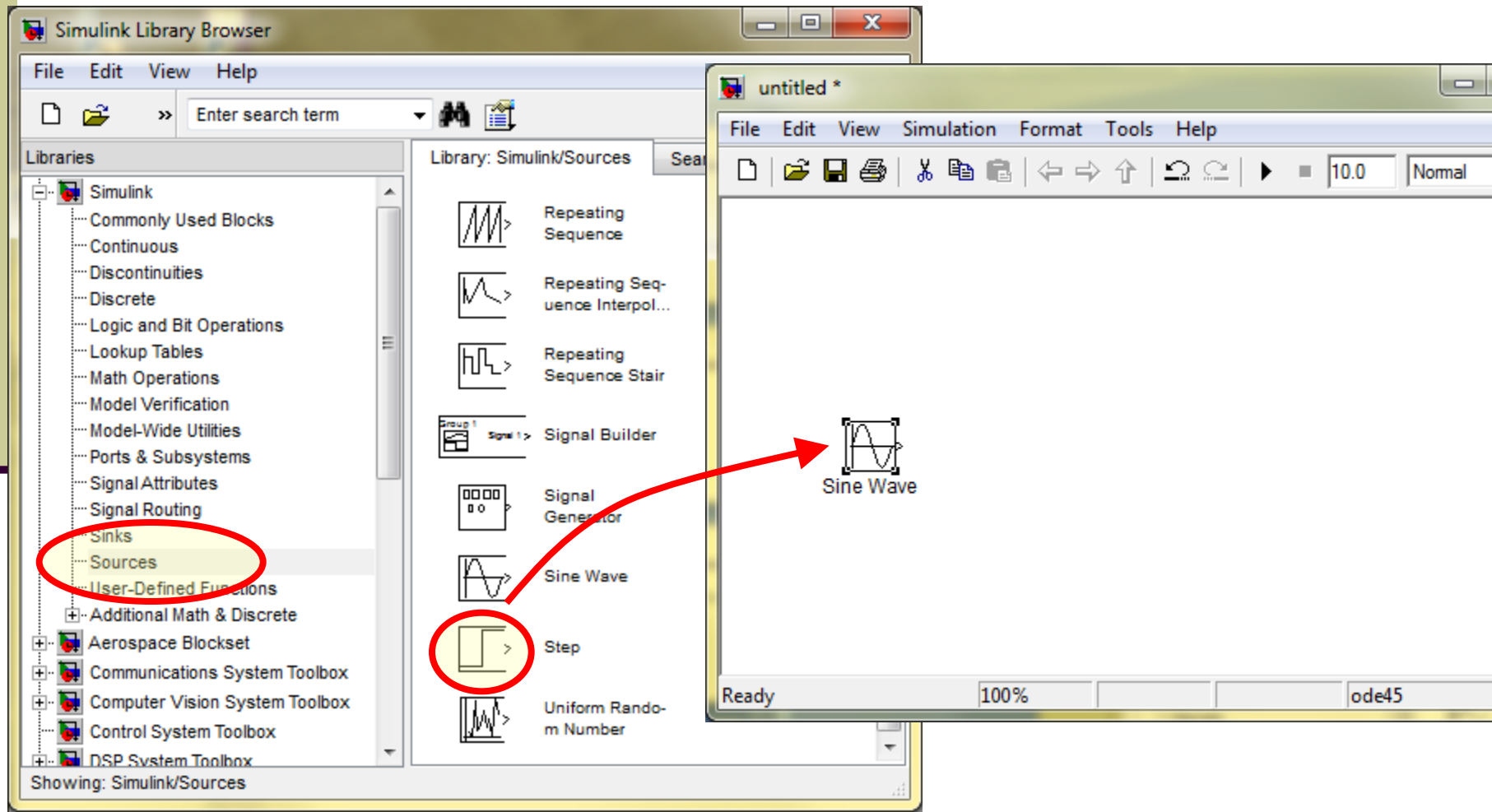
The Simulink block library





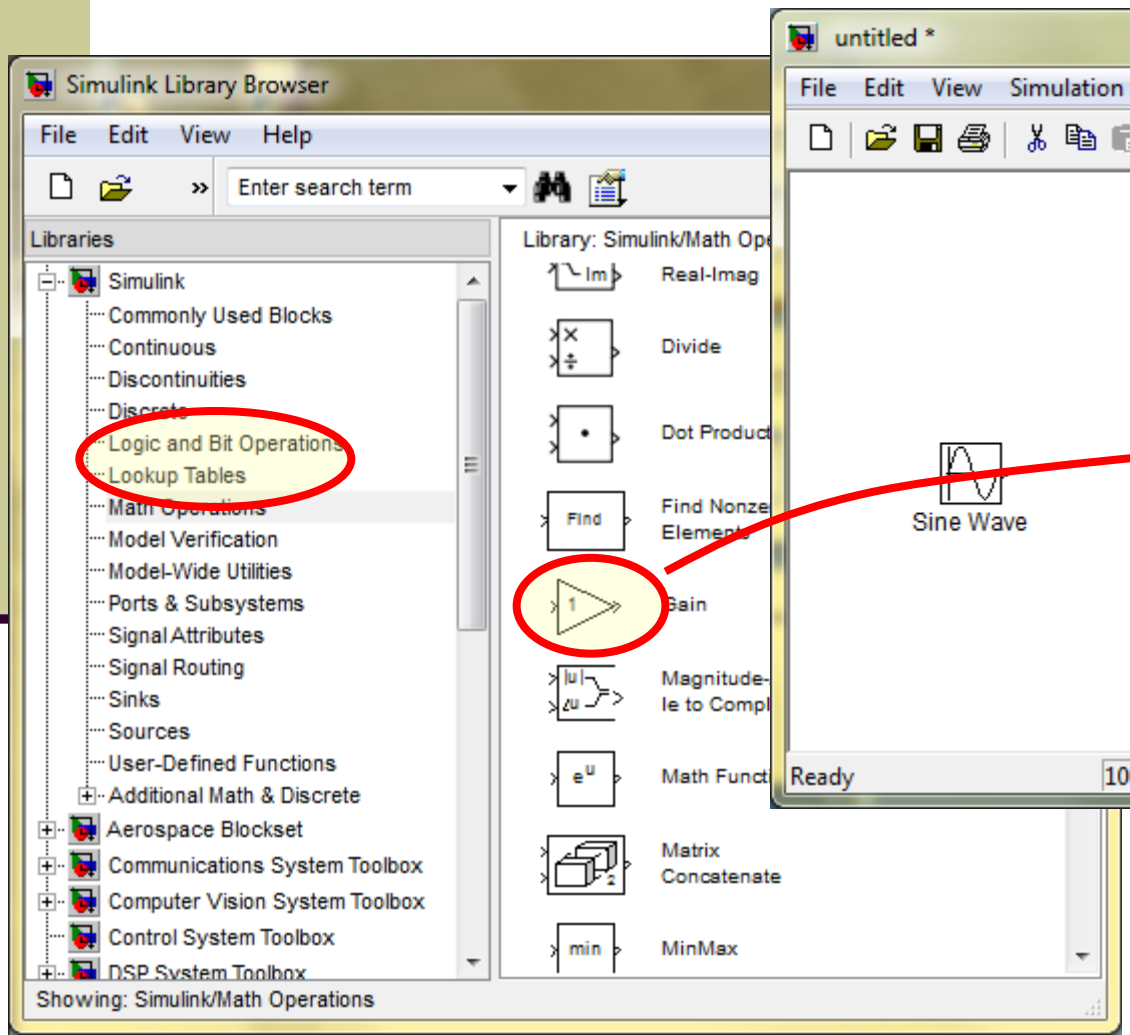
Collecting blocks

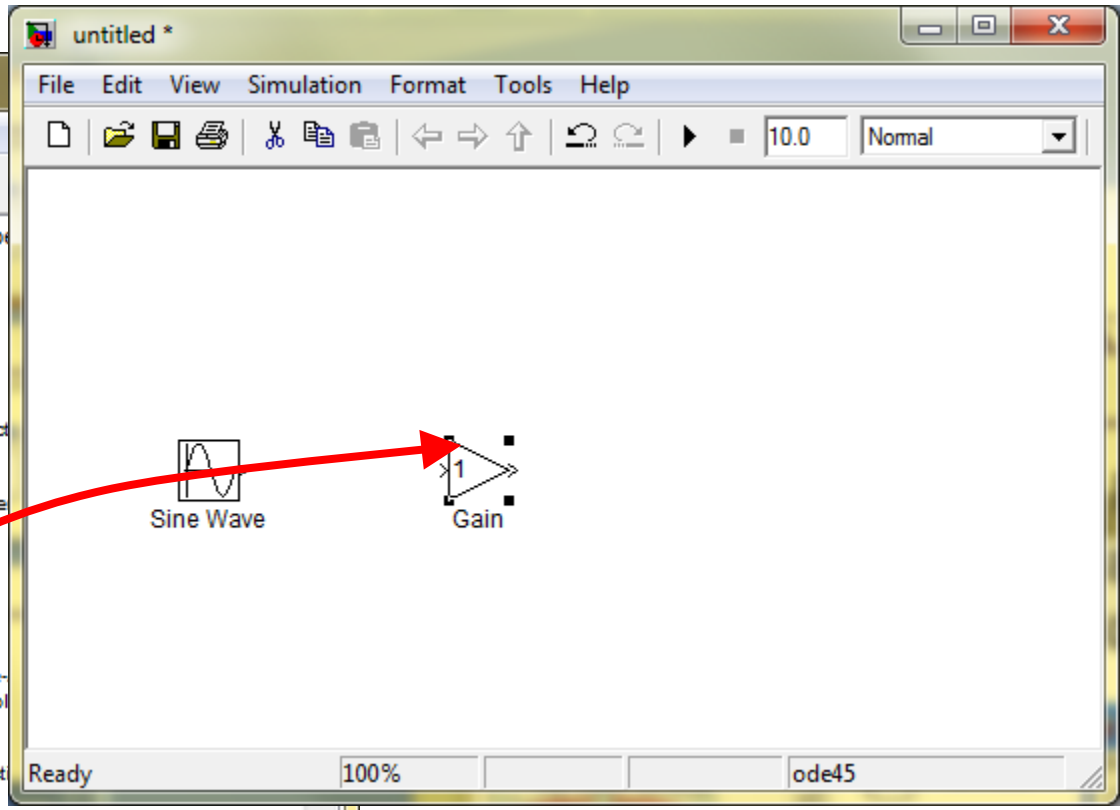
Pull the required blocks into a new Simulink block diagram window



Collecting blocks

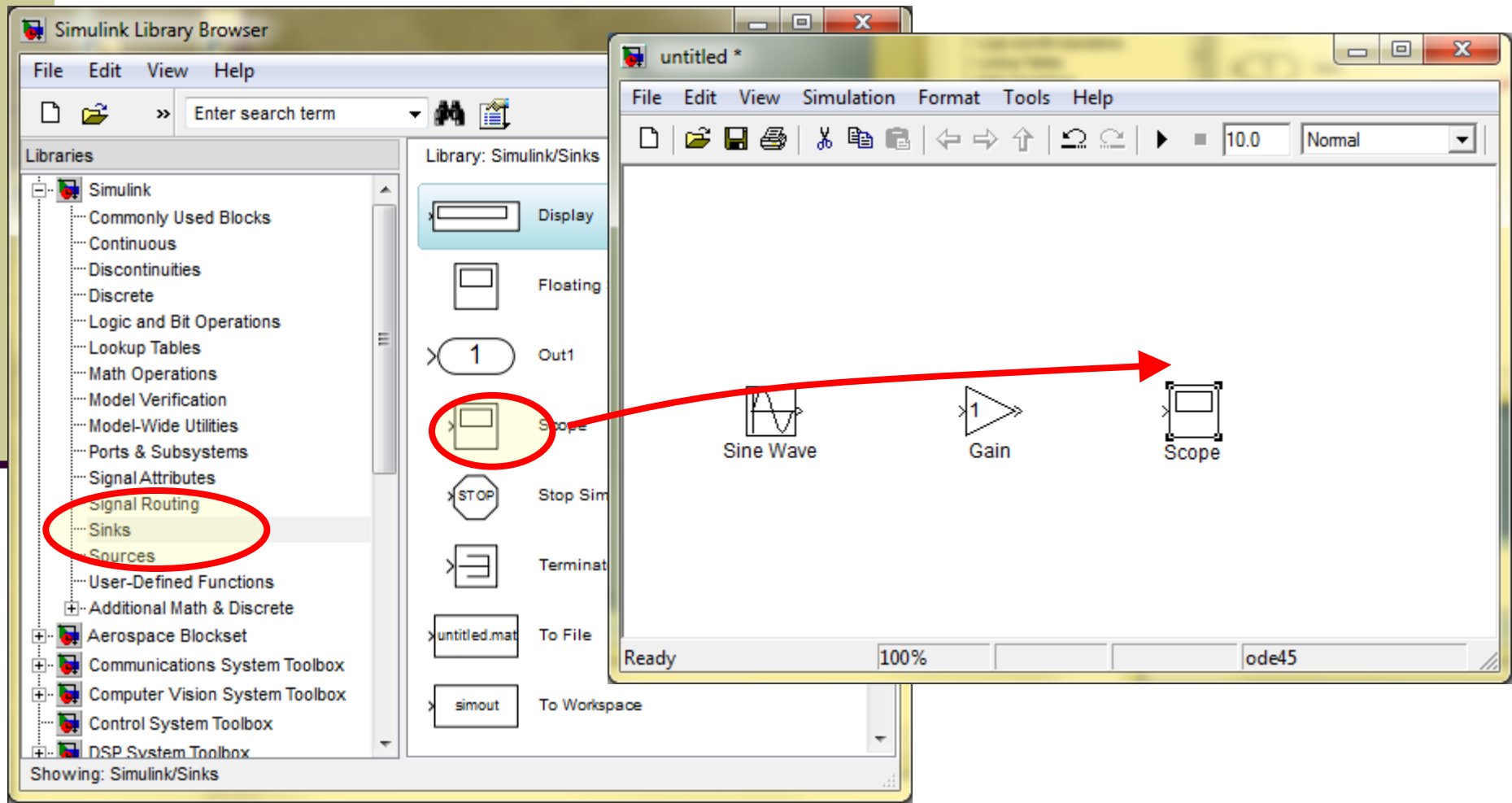
Pull the required blocks into a new Simulink block diagram window





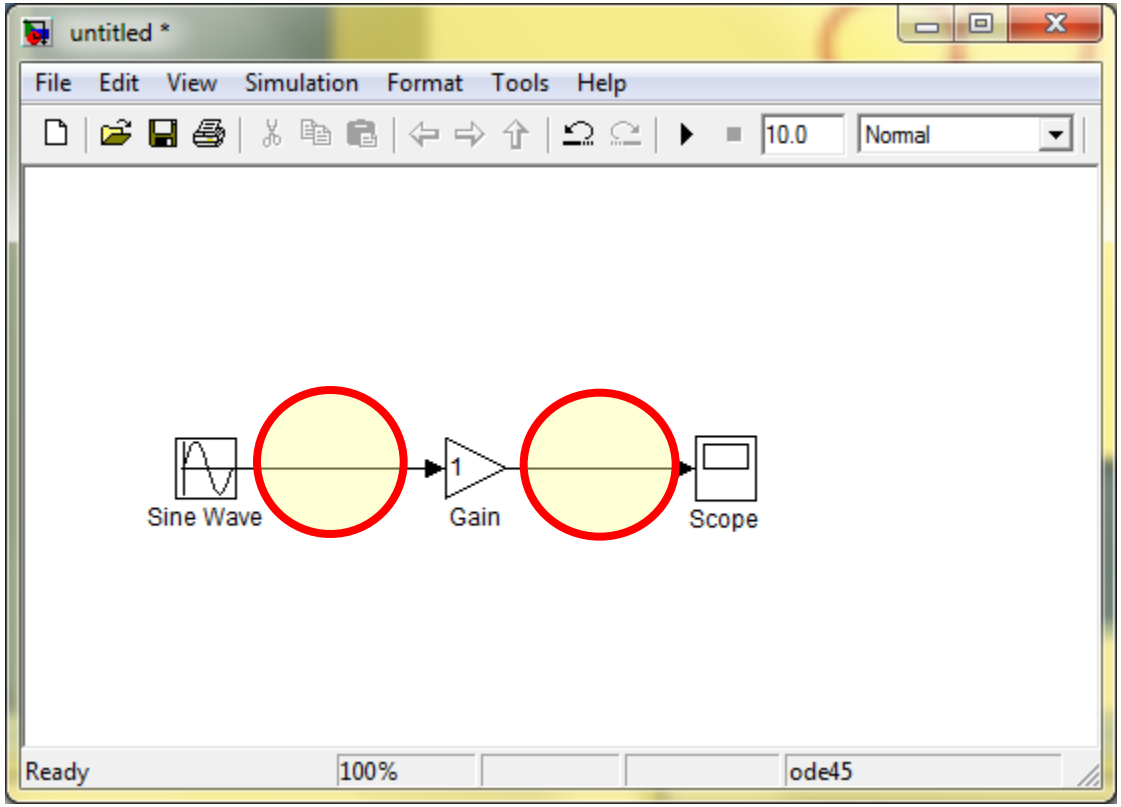
Collecting blocks

Pull the required blocks into a new Simulink block diagram window



Connecting blocks

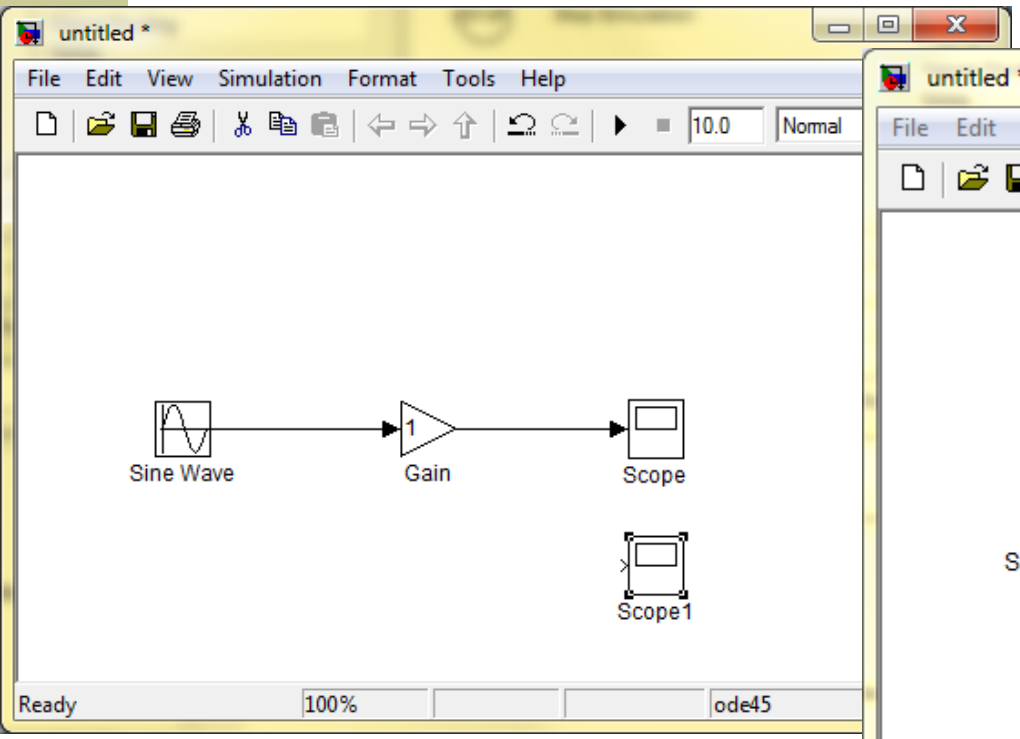
Left Mouse drag from port to add line

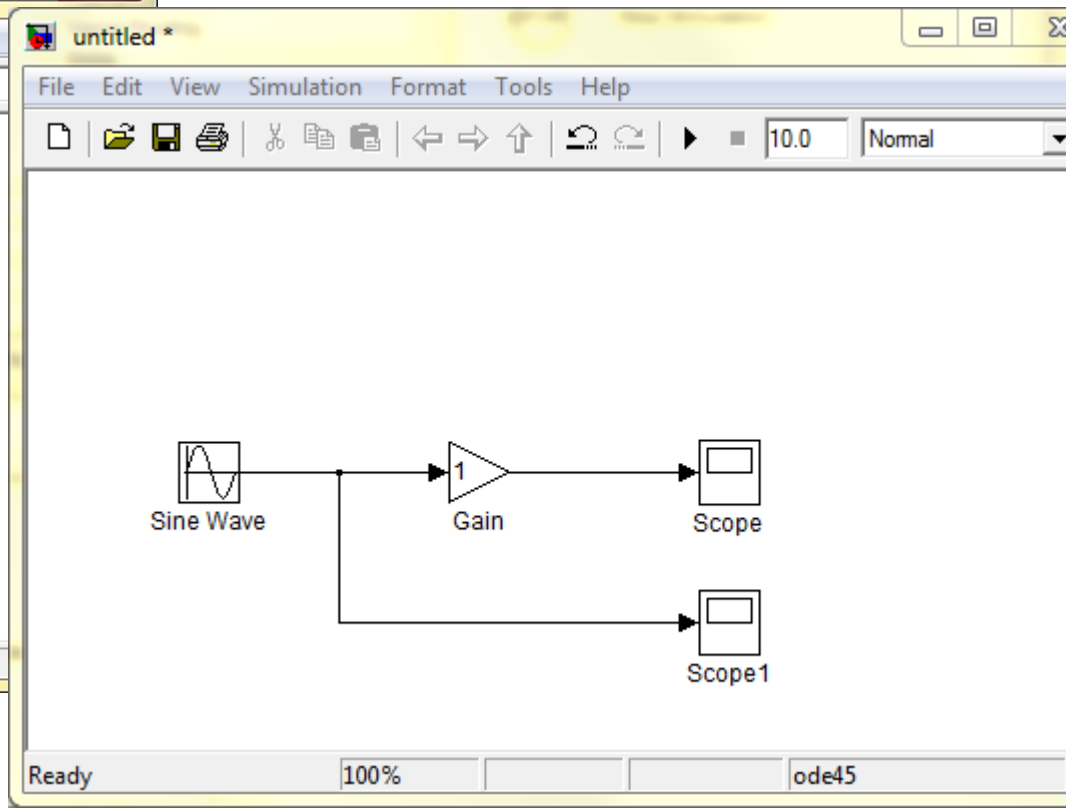


Copying blocks & branching signal

Right Mouse drag
to copy

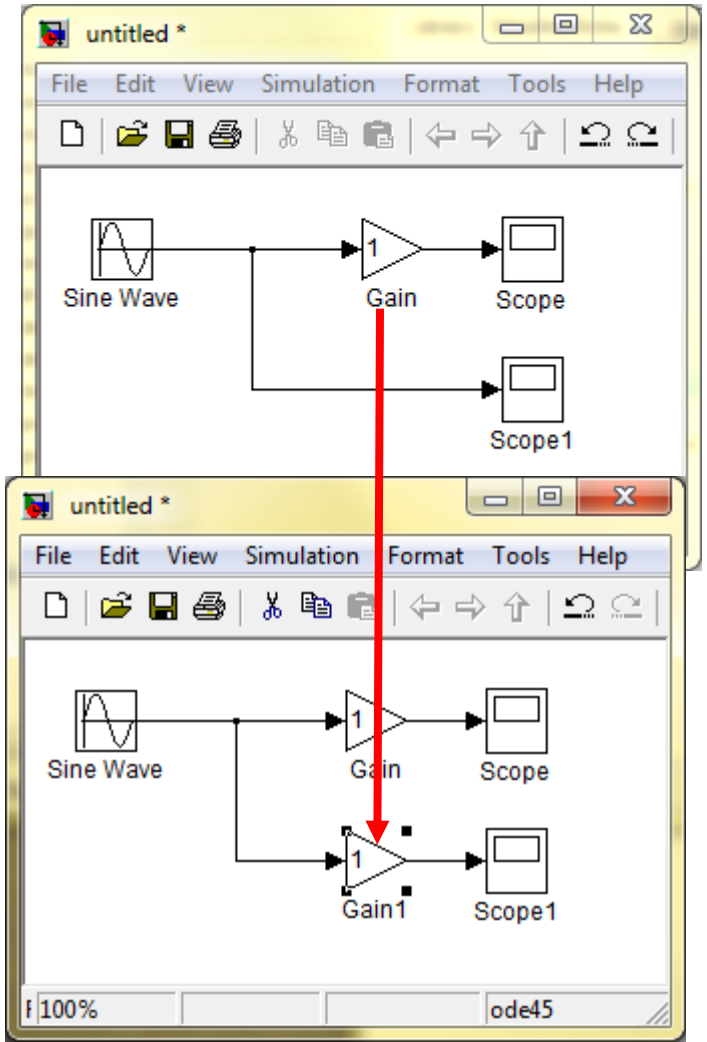
Right Mouse drag
to add branch line



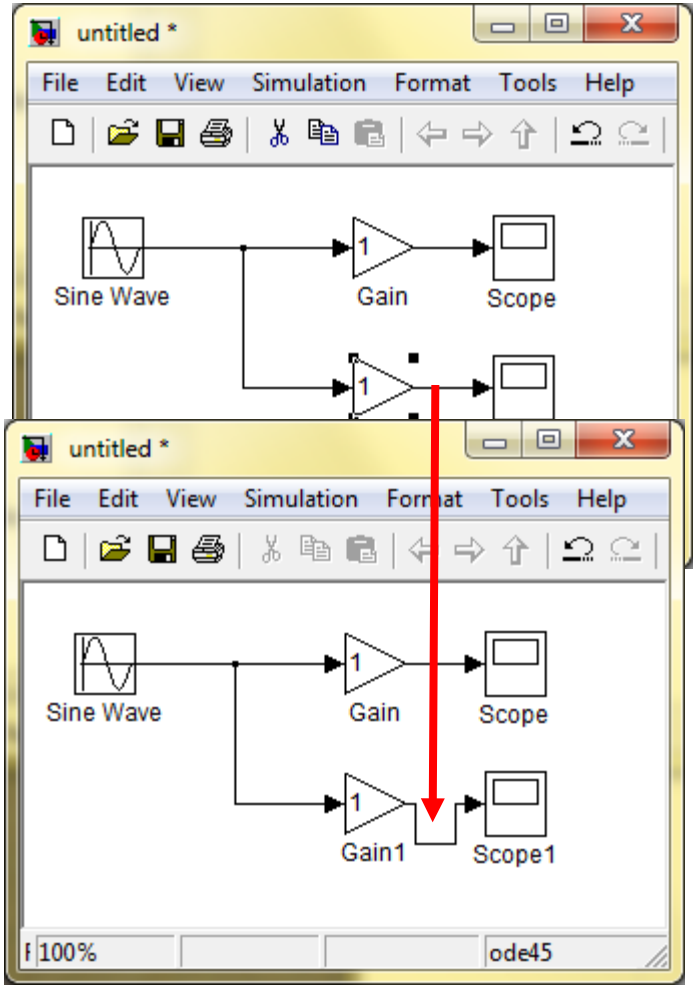


Inserting Blocks, rerouting lines

Place block over line to insert

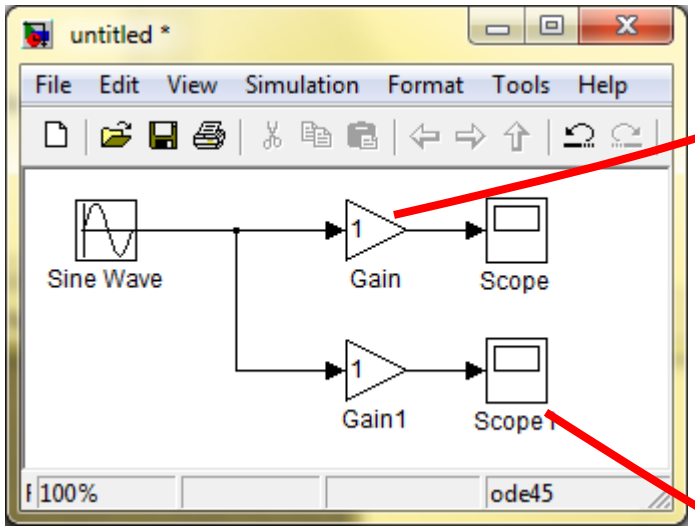


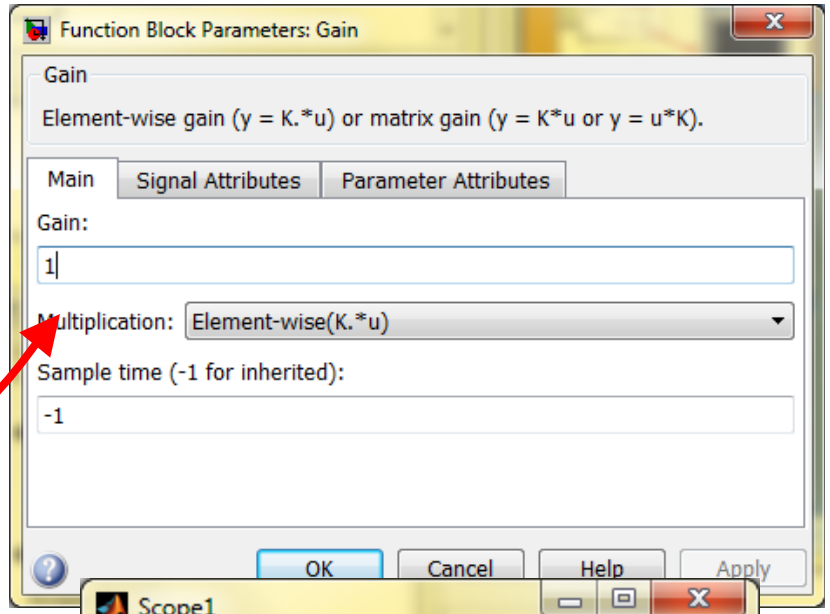
Left mouse drag between breakpoints

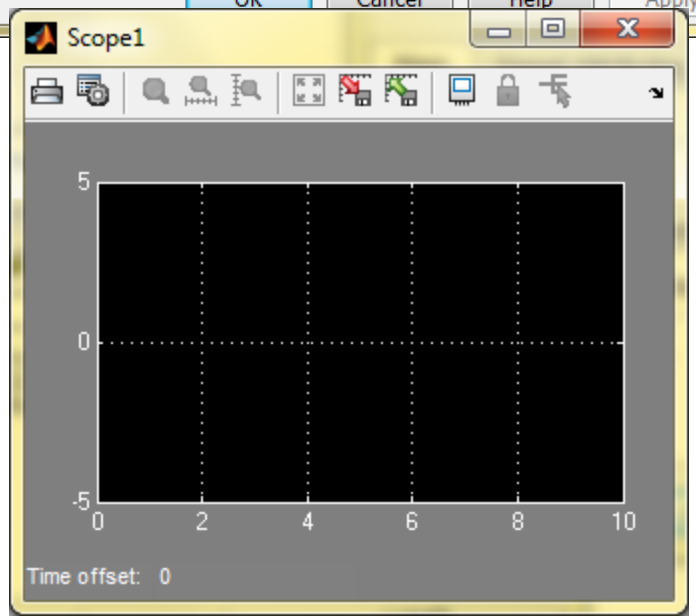


Setting block parameters

Double or right click on block to access parameter window



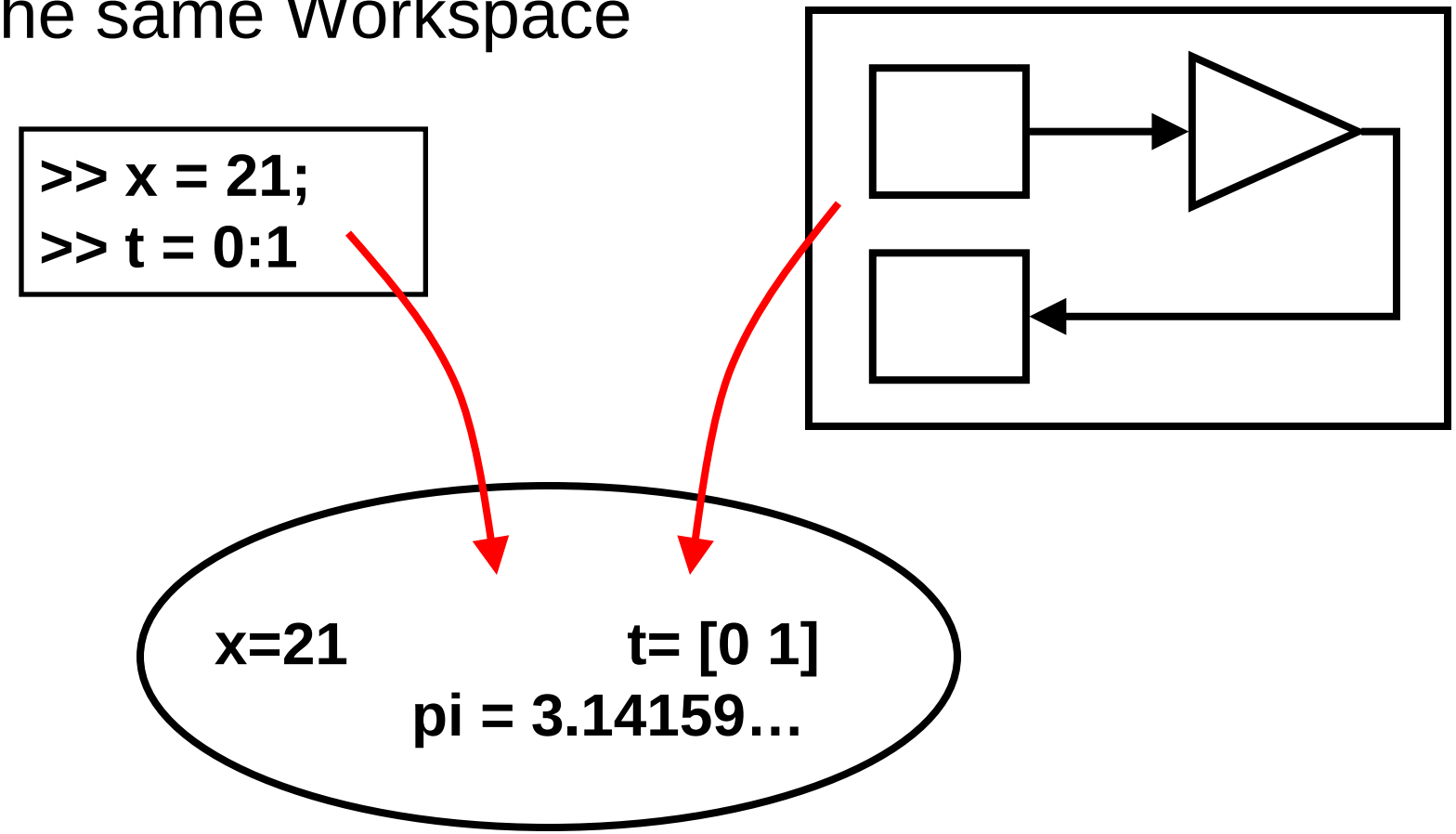




Double or right click on Scope block to open

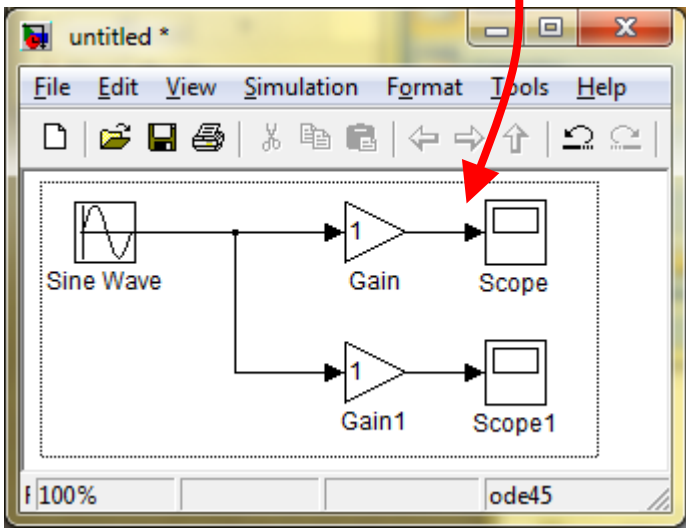
Defining variables

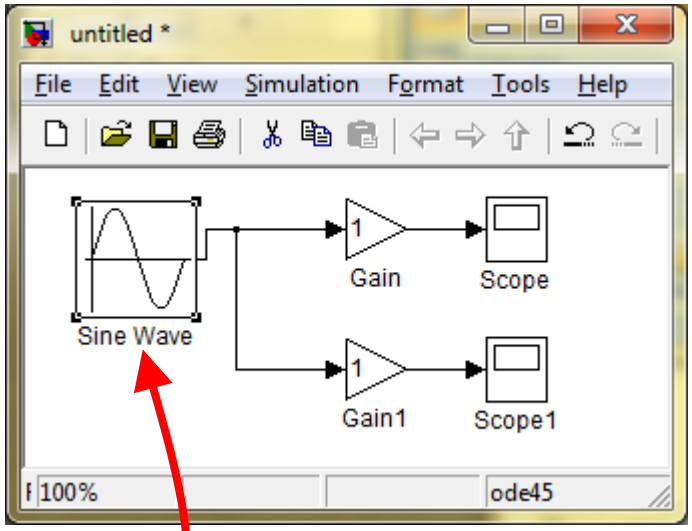
Both MATLAB and Simulink Windows “see” into the same Workspace



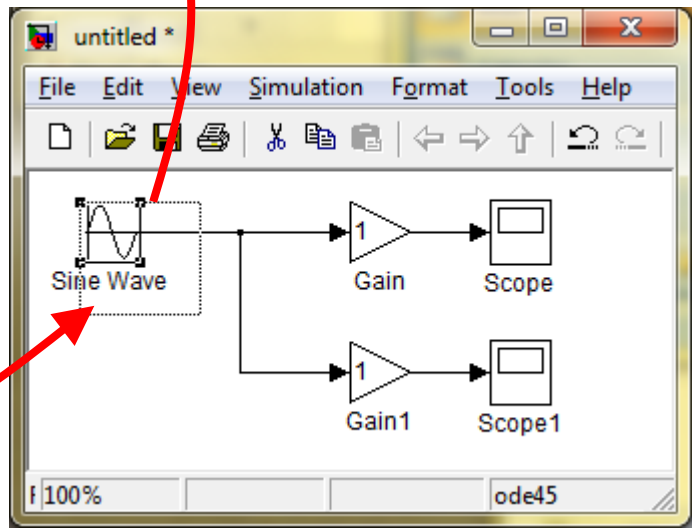
Setting block characteristics

Select blocks by drawing a box around them



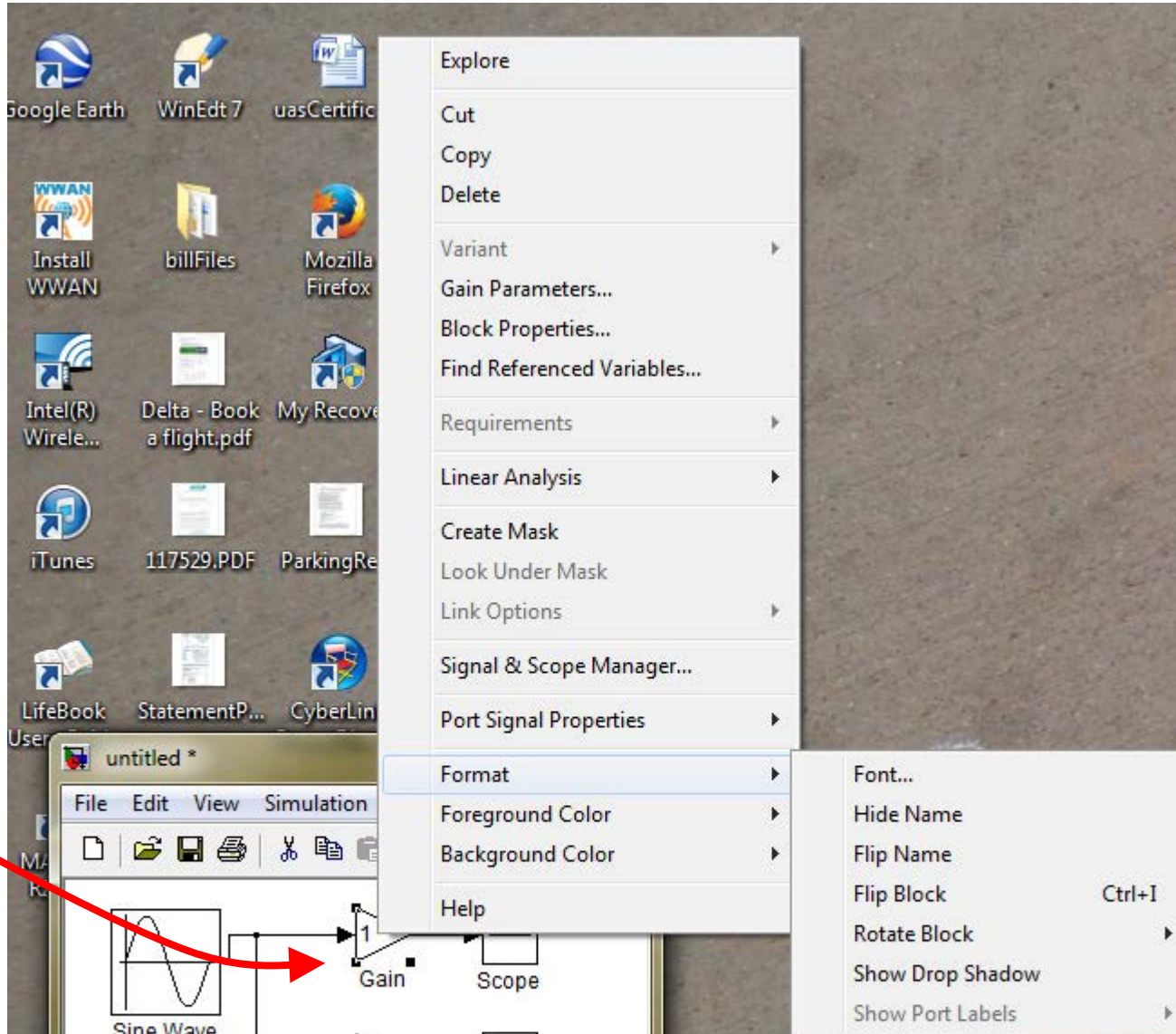


Drag corner of selected block to resize



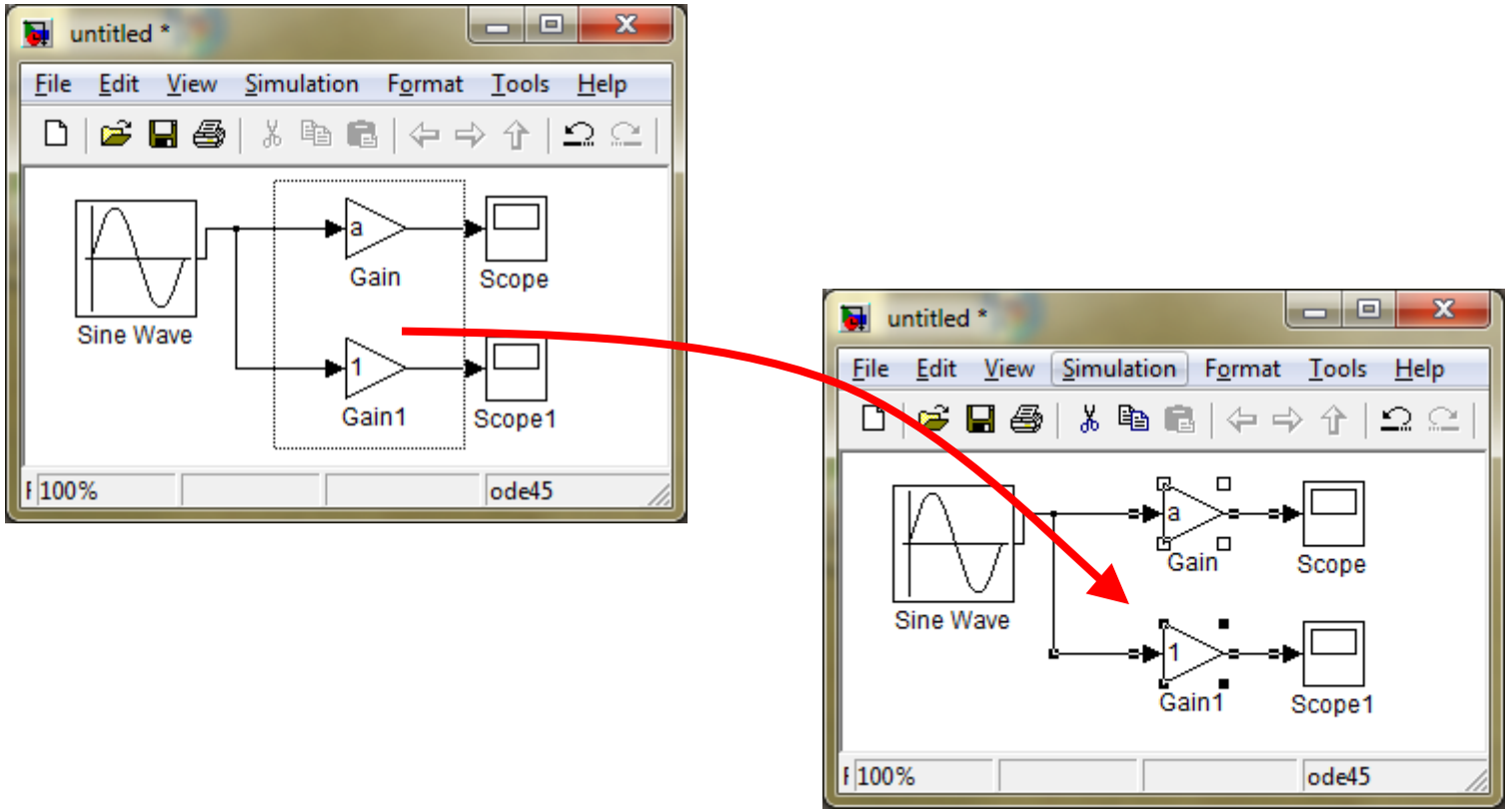
Setting block characteristics

Right click on block to show formatting menu



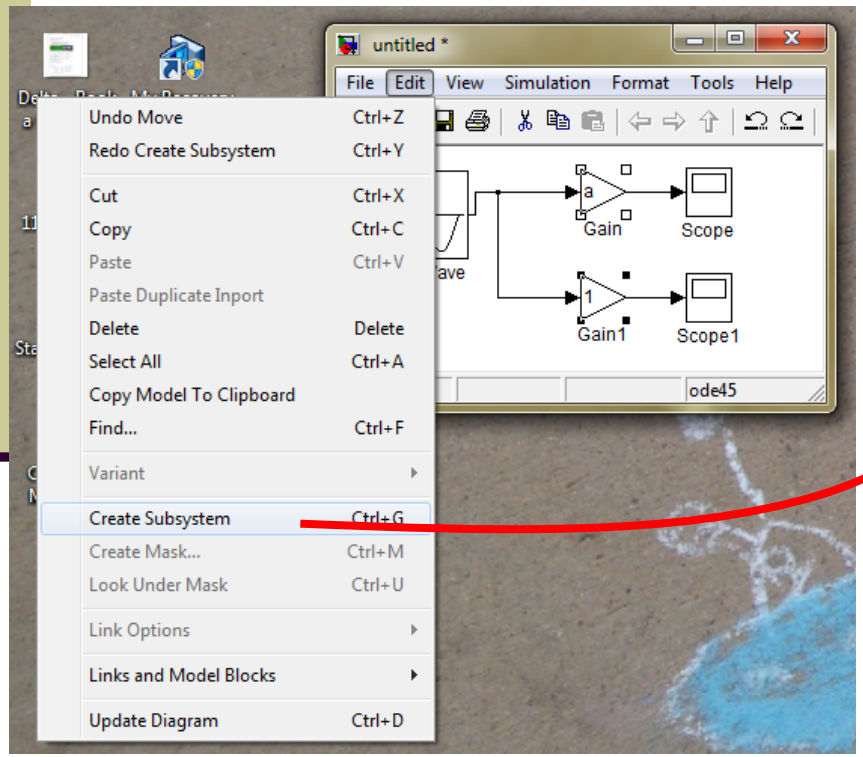
Creating Subsystems

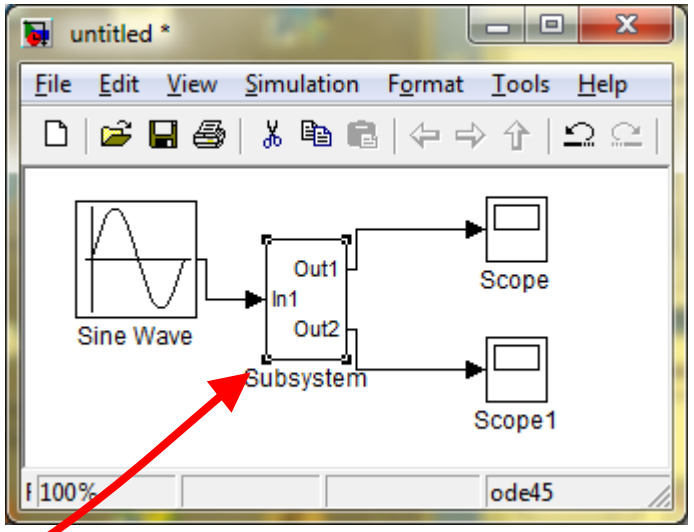
Draw a box around
blocks to select area



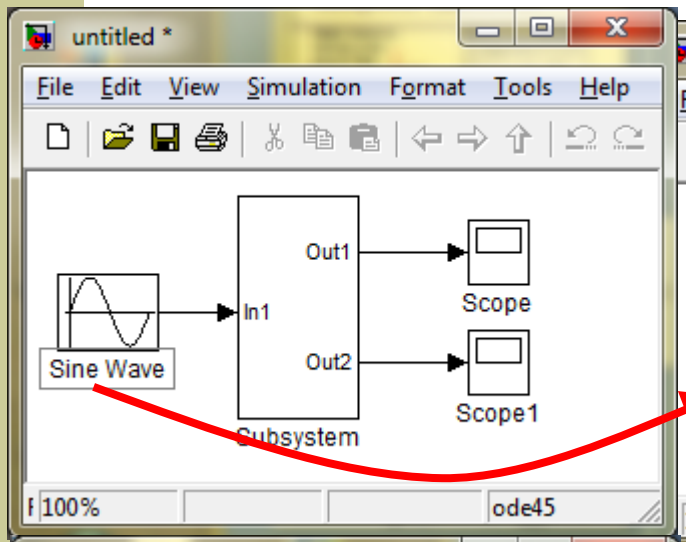
Creating Subsystems

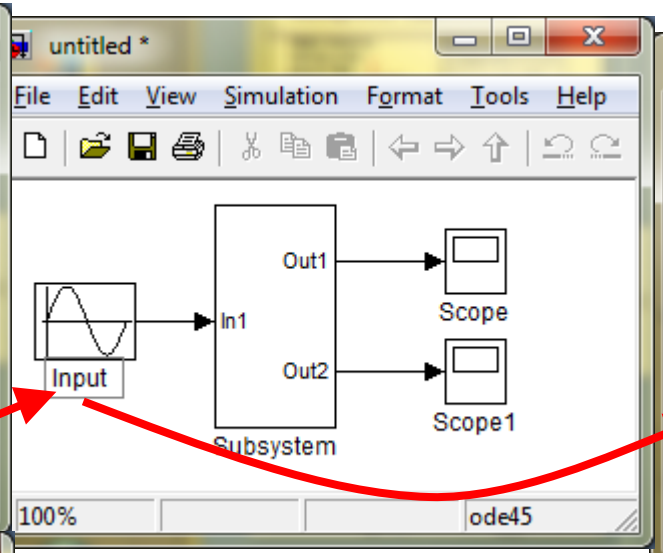
Select “Create Subsystem” under the “Edit” menu

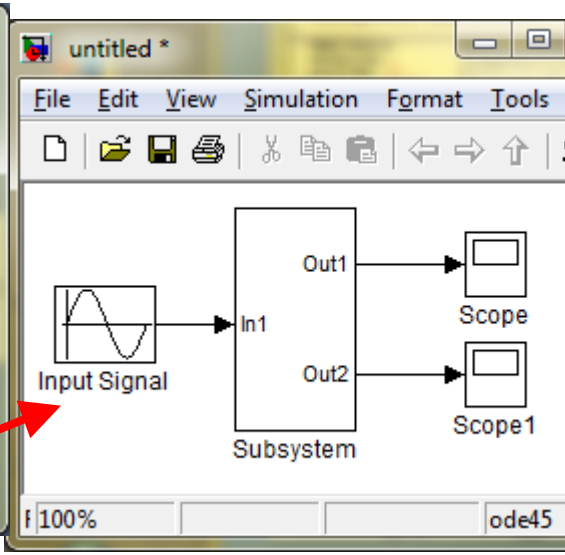




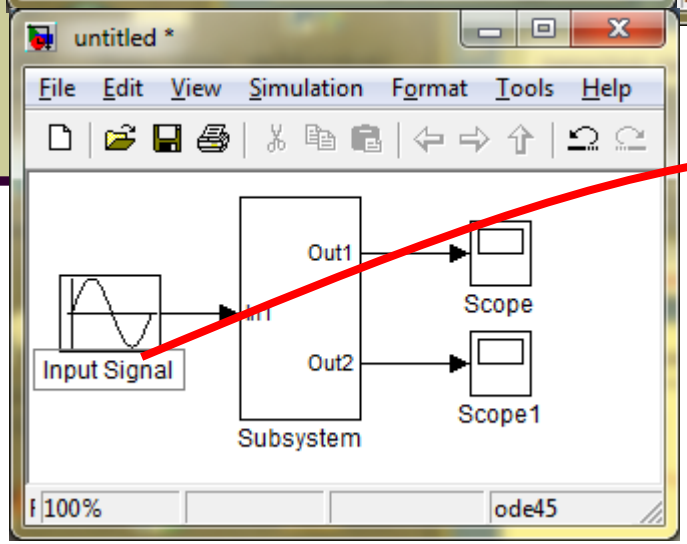
Changing block names



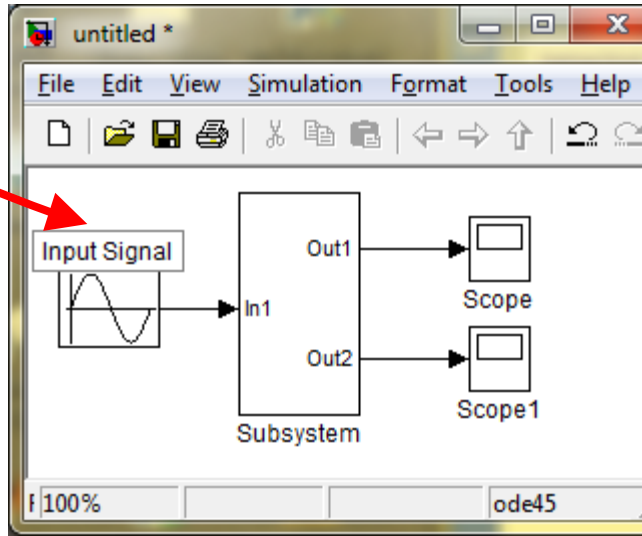




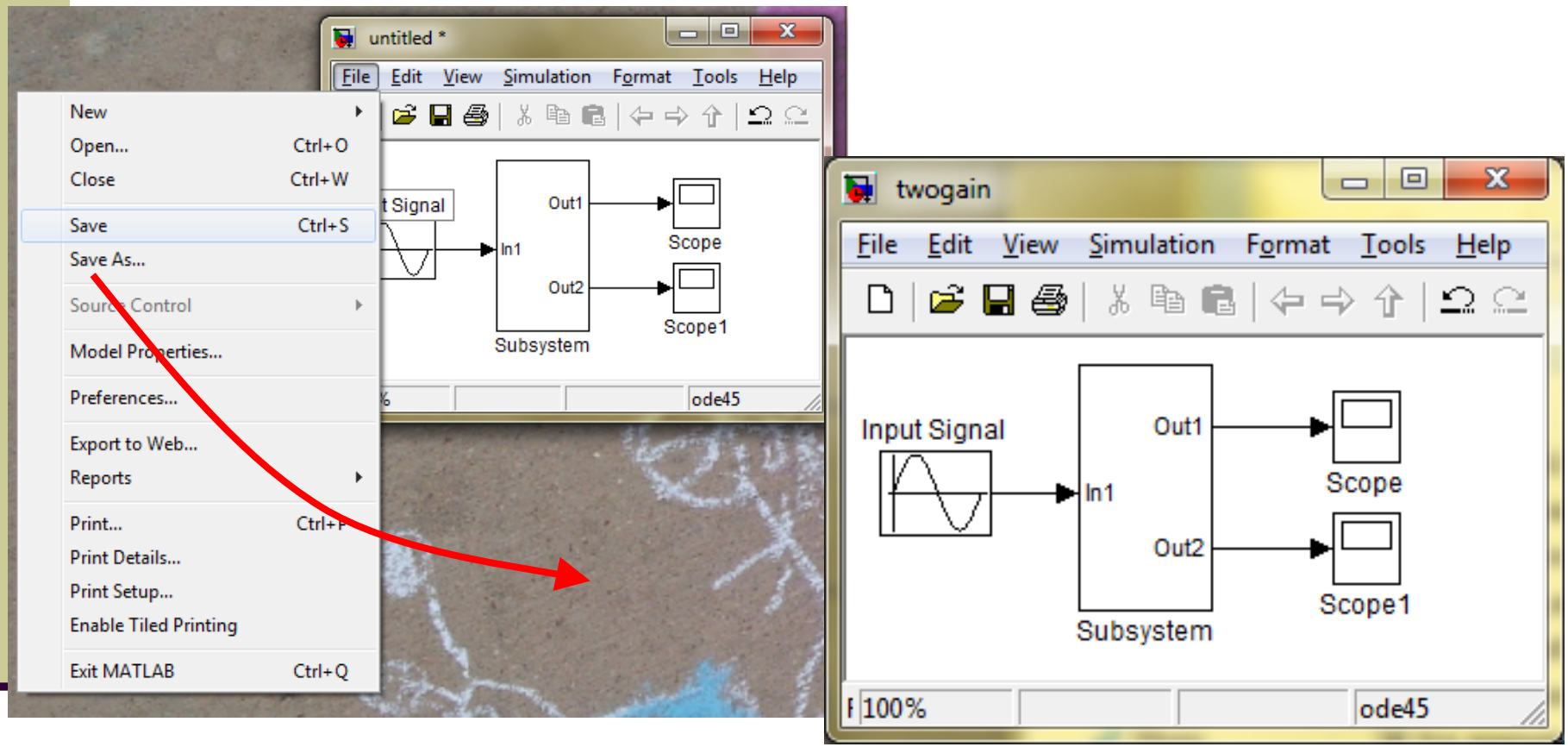
Enter the new name



Drag the name to the opposite side of the block



Saving



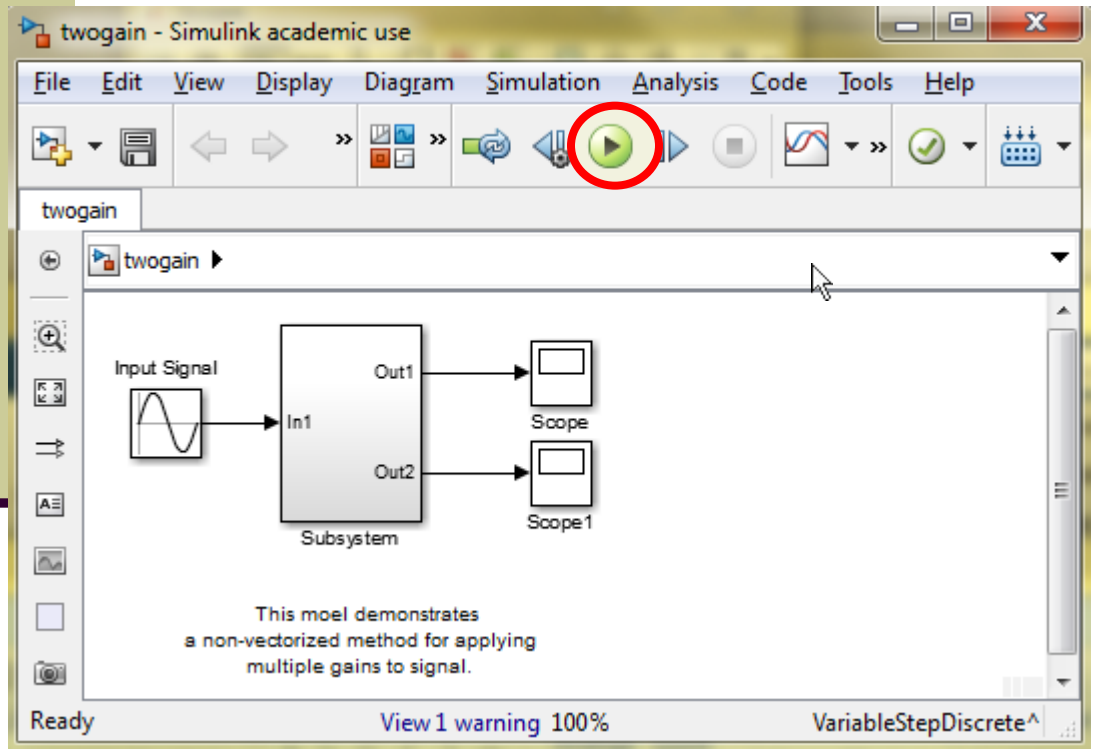
>> twogain

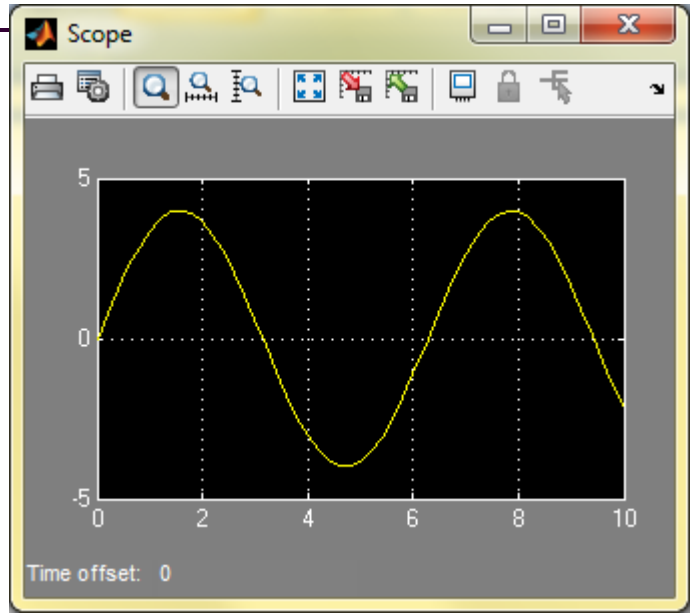
Simulating

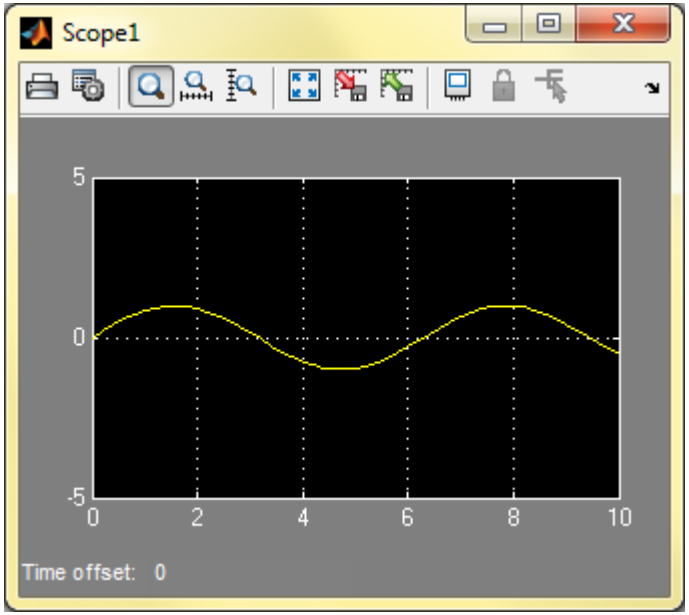
Click on the

“Run” button to

Run simulation

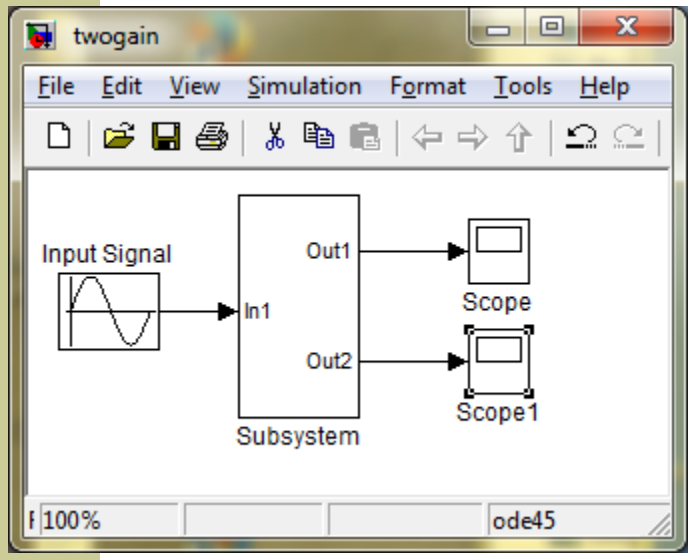


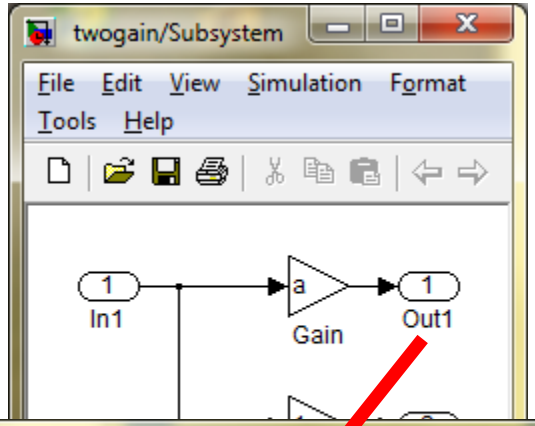




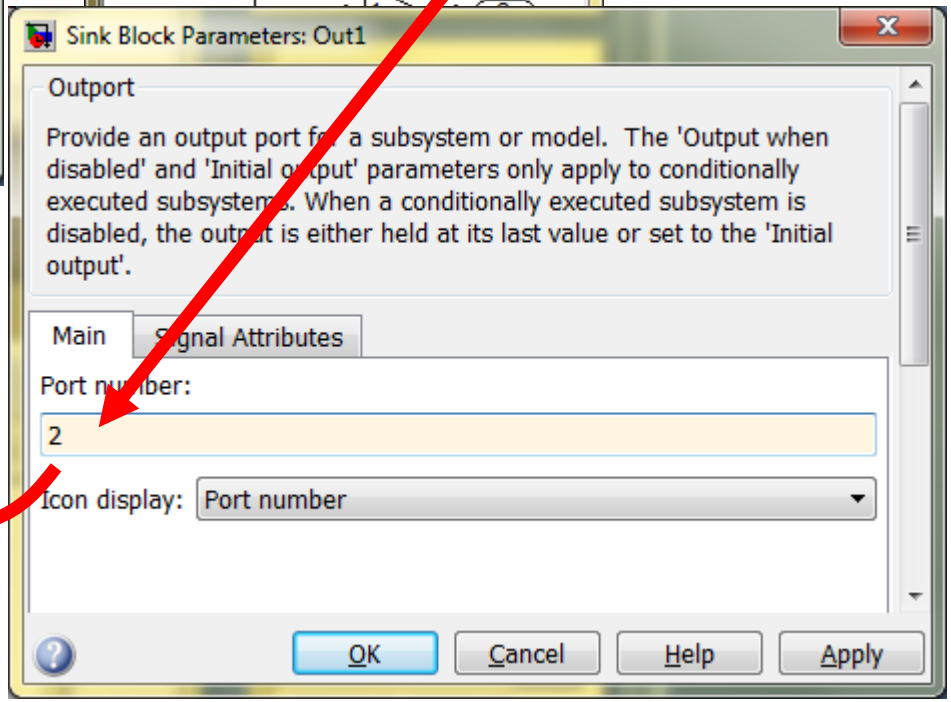

```
>> a=4 ;
```

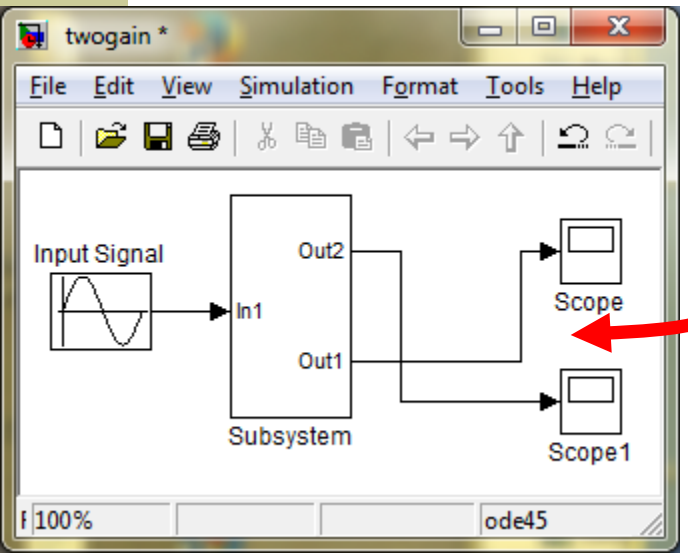
Inports and Outports





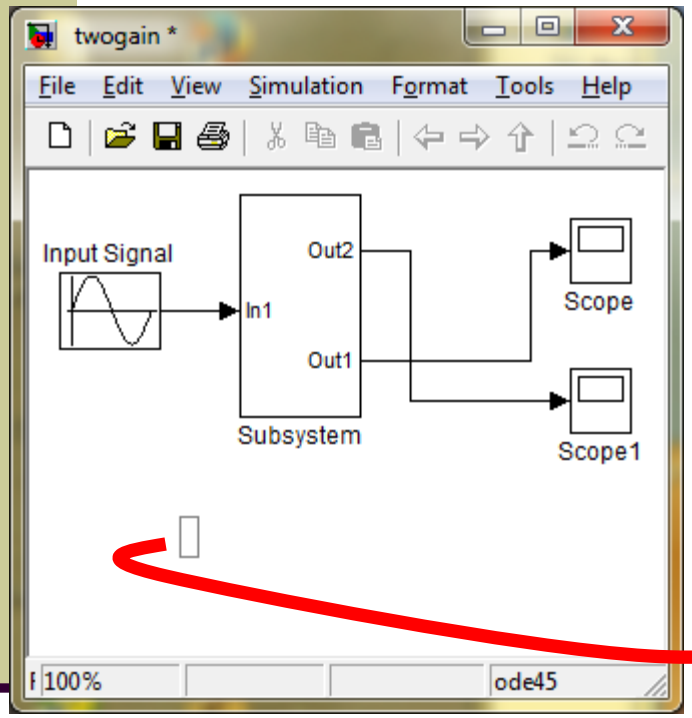
Inports and outports remain monotonically Increasing when renumbered

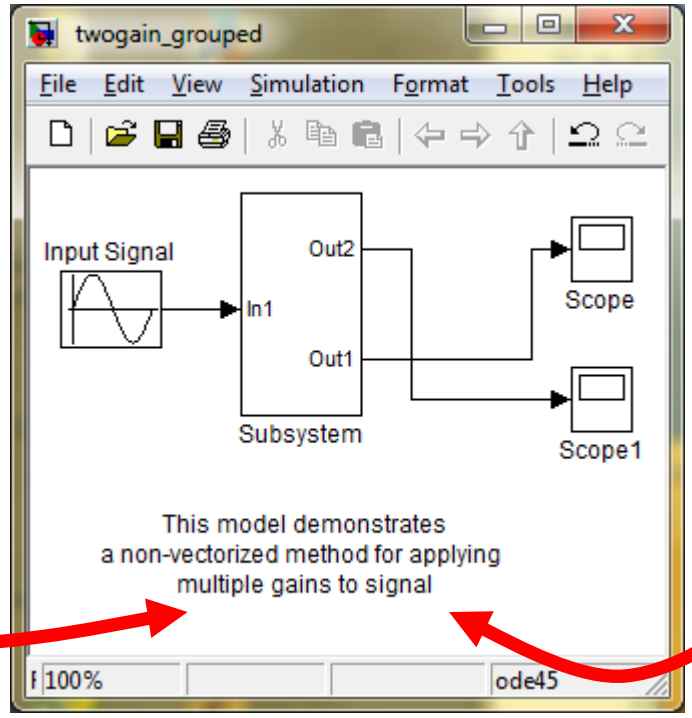




Connectivity remains constant

Annotations





Resize font as desired by right-clicking on text

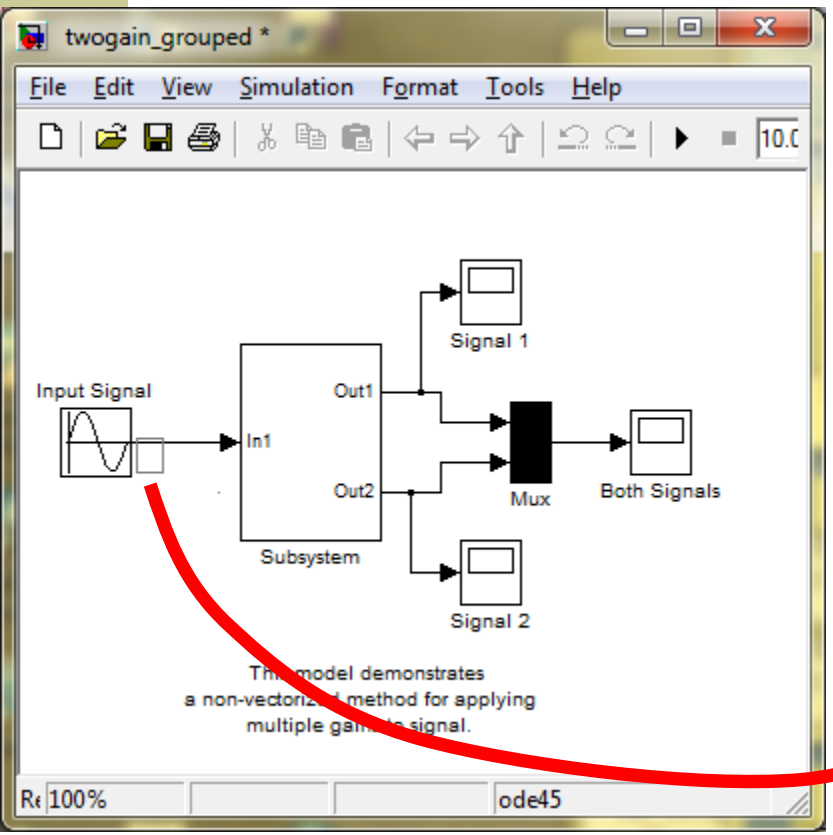
Double click on Background

>>twogain_grouped

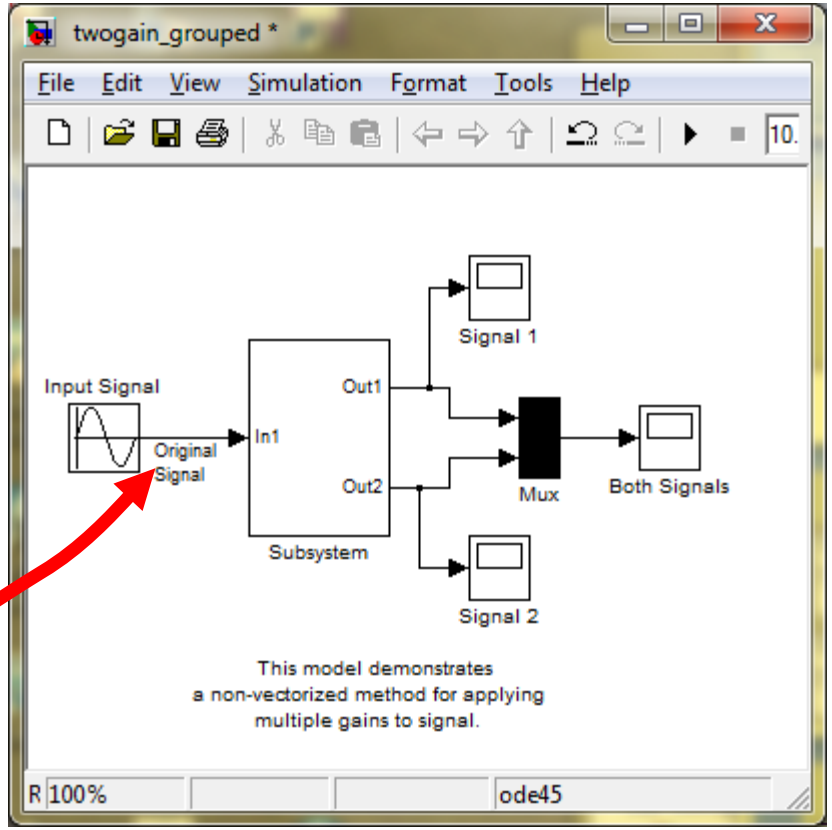
Type in the text and save as 'twogain_grouped'

Signal labels

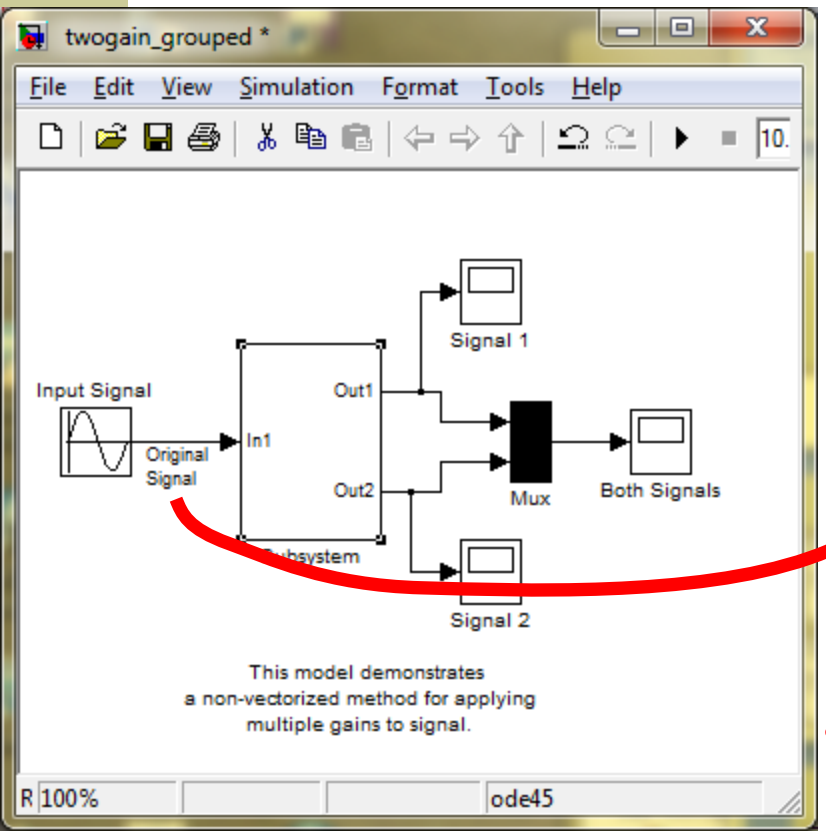
Add a **Mux** block and a third scope to twogain_grouped

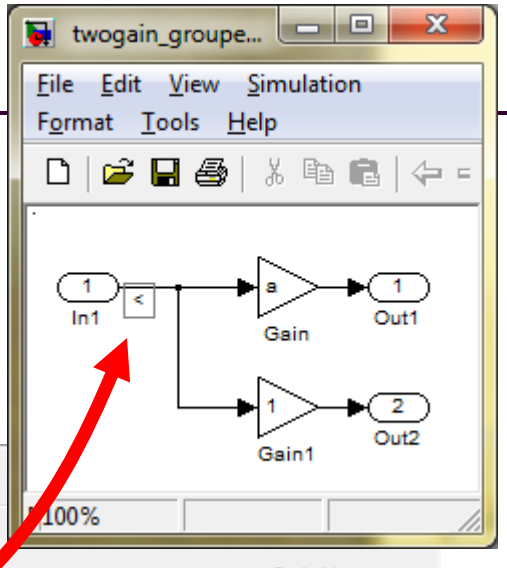


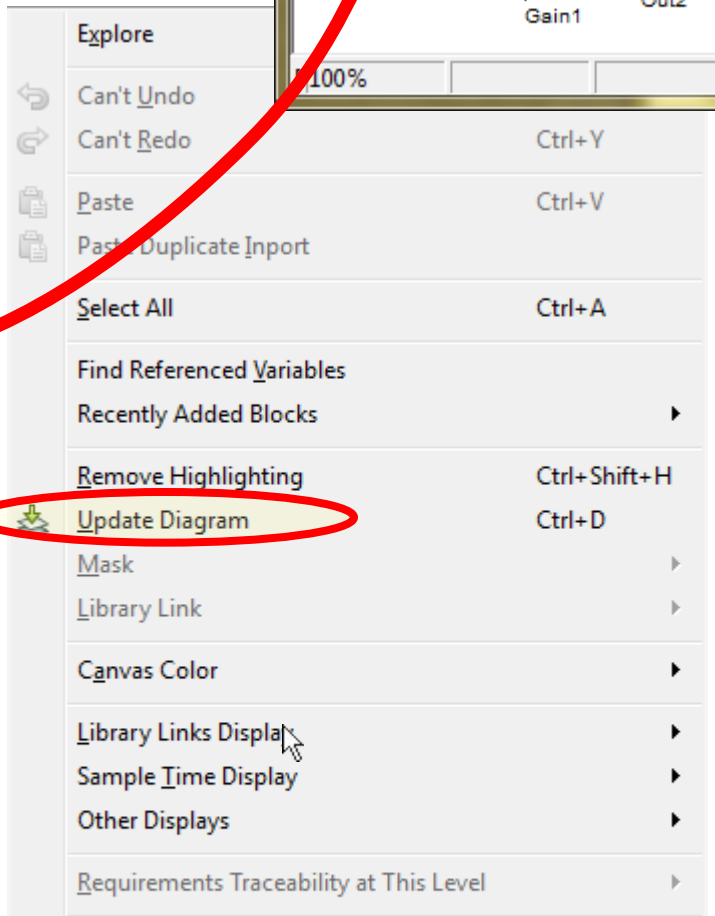
Signal labels works just like annotation, but are attached to signals



Signal labels

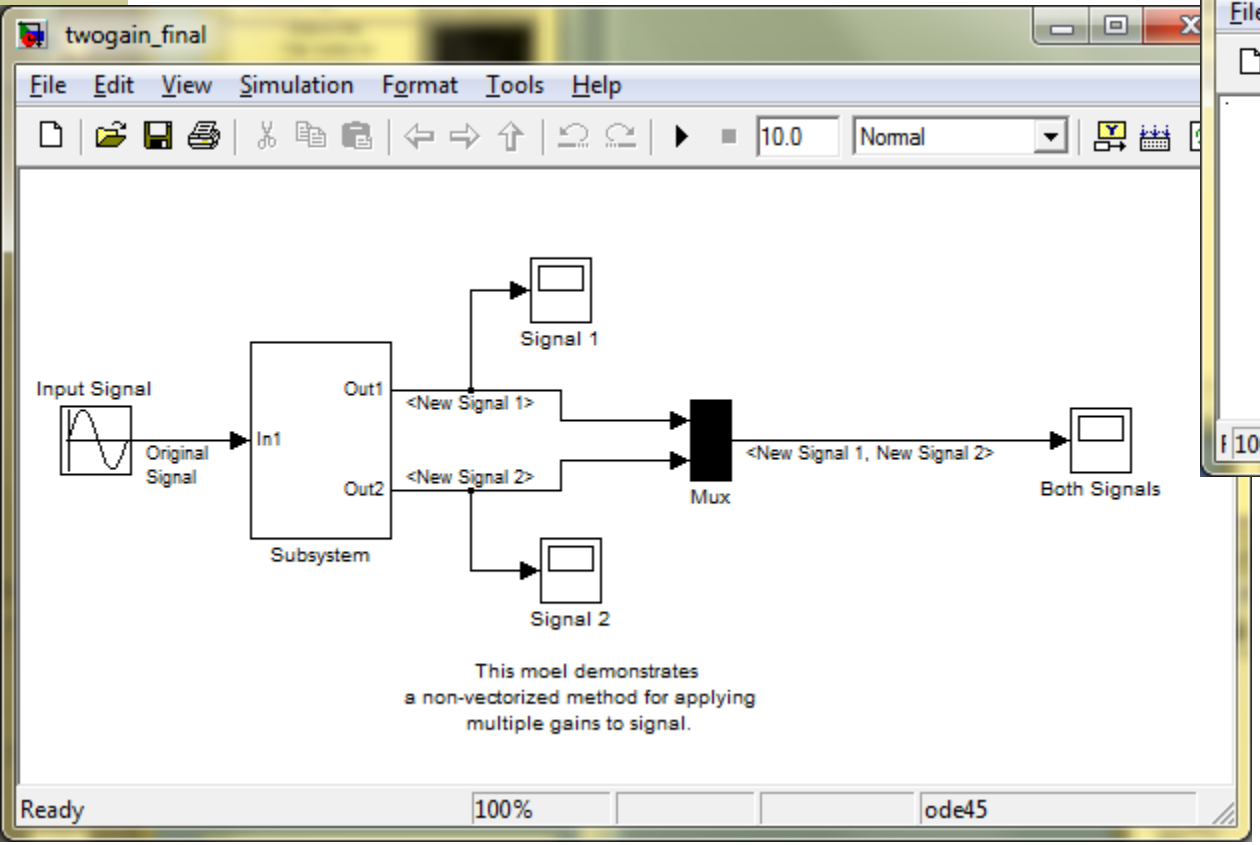


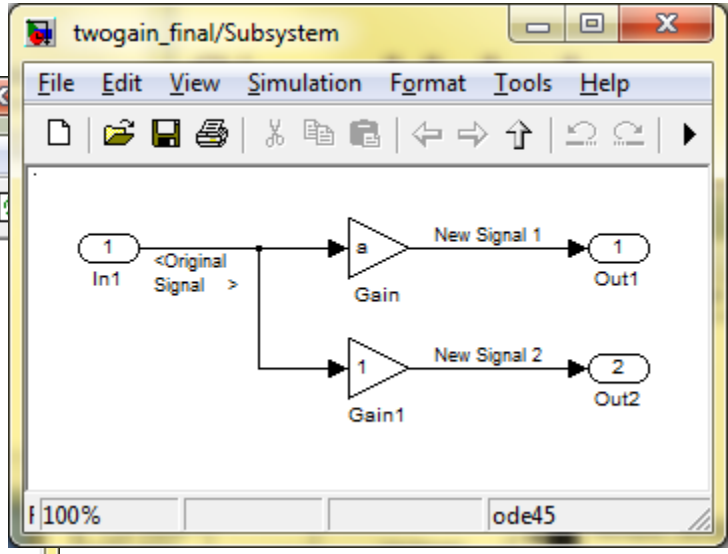




Signal labels can pass through "virtual block"

Signal labels

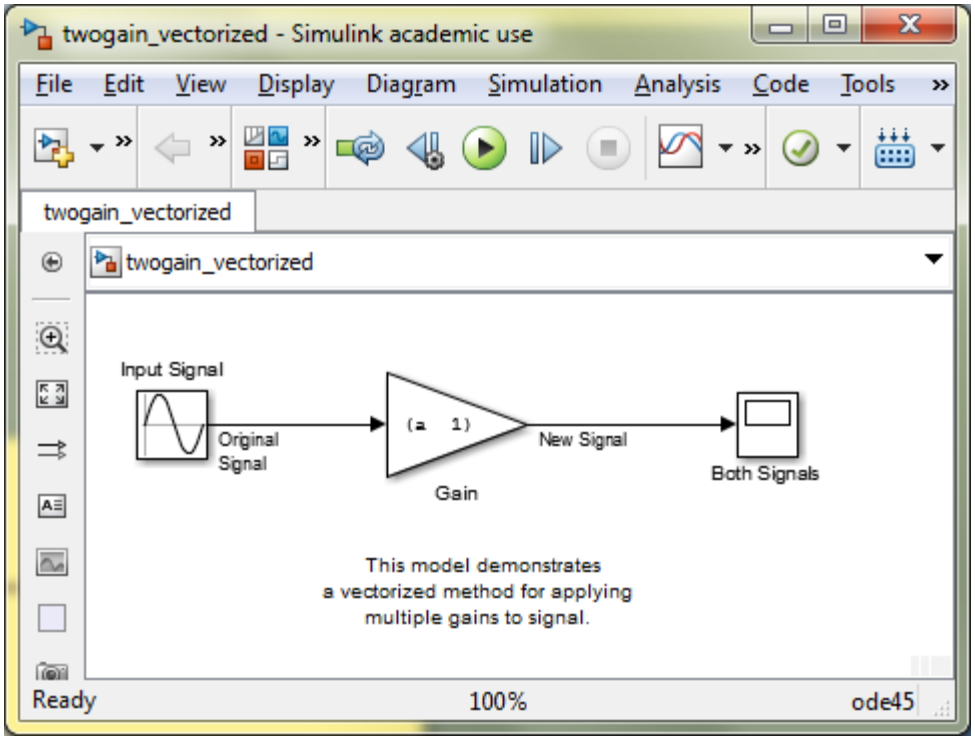




Label properties can be set for each signal.

```
>>twogain_final
```

Vectorized Signals

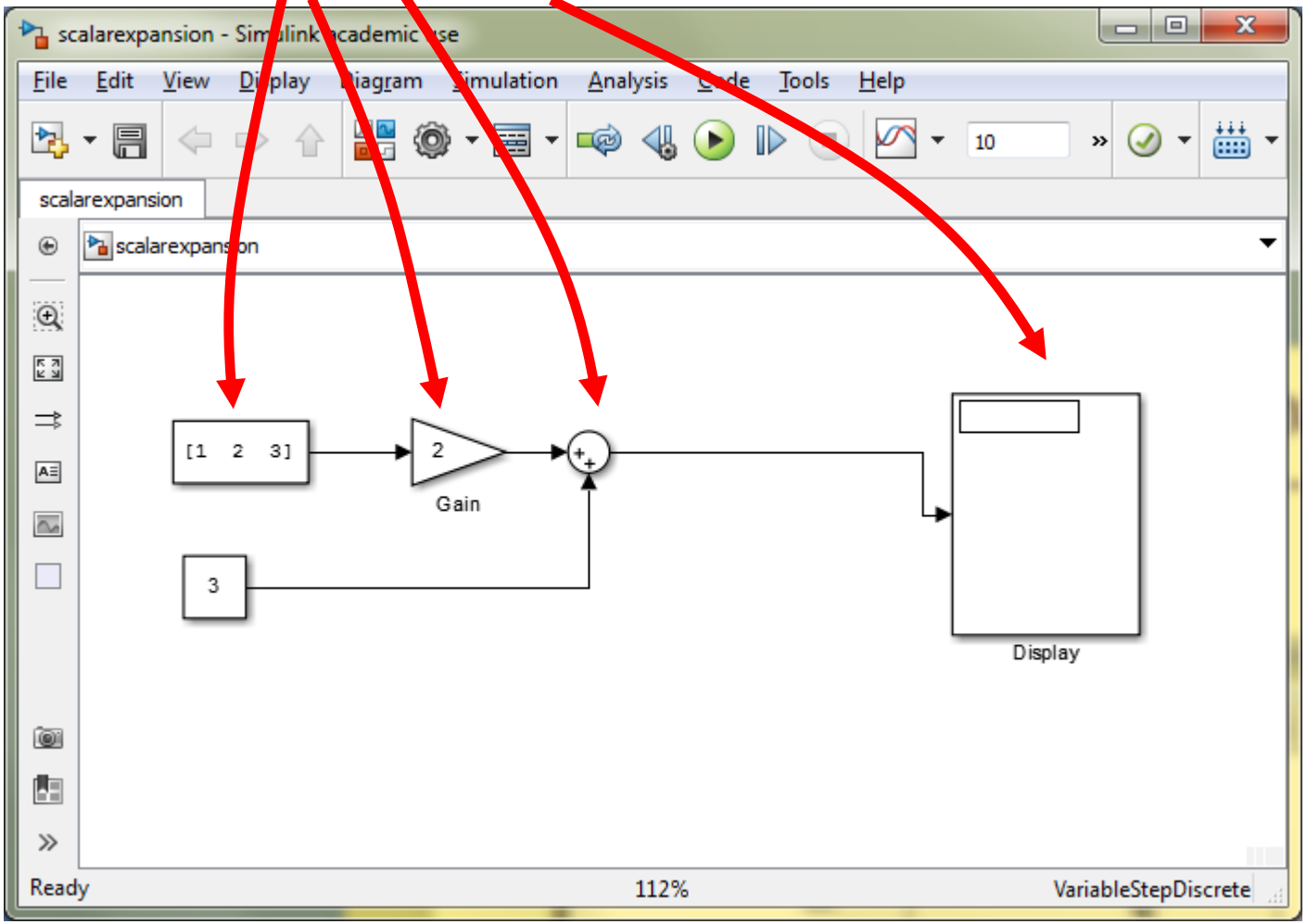


Signals can be vectorized.

```
>> twogain_vectorized
```

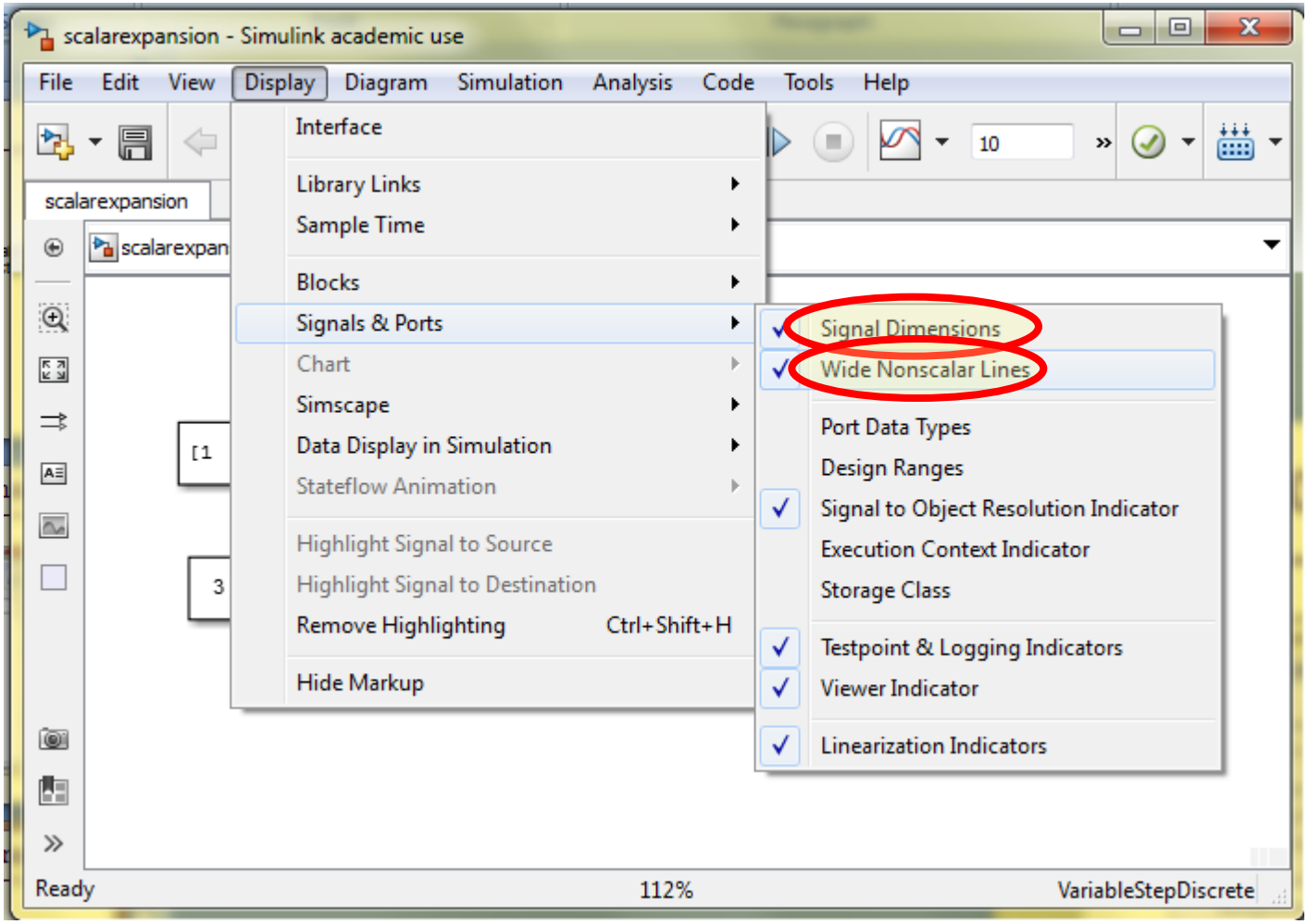
Vector I/O and Scalar expansion

Vectorized Inputs and Outputs



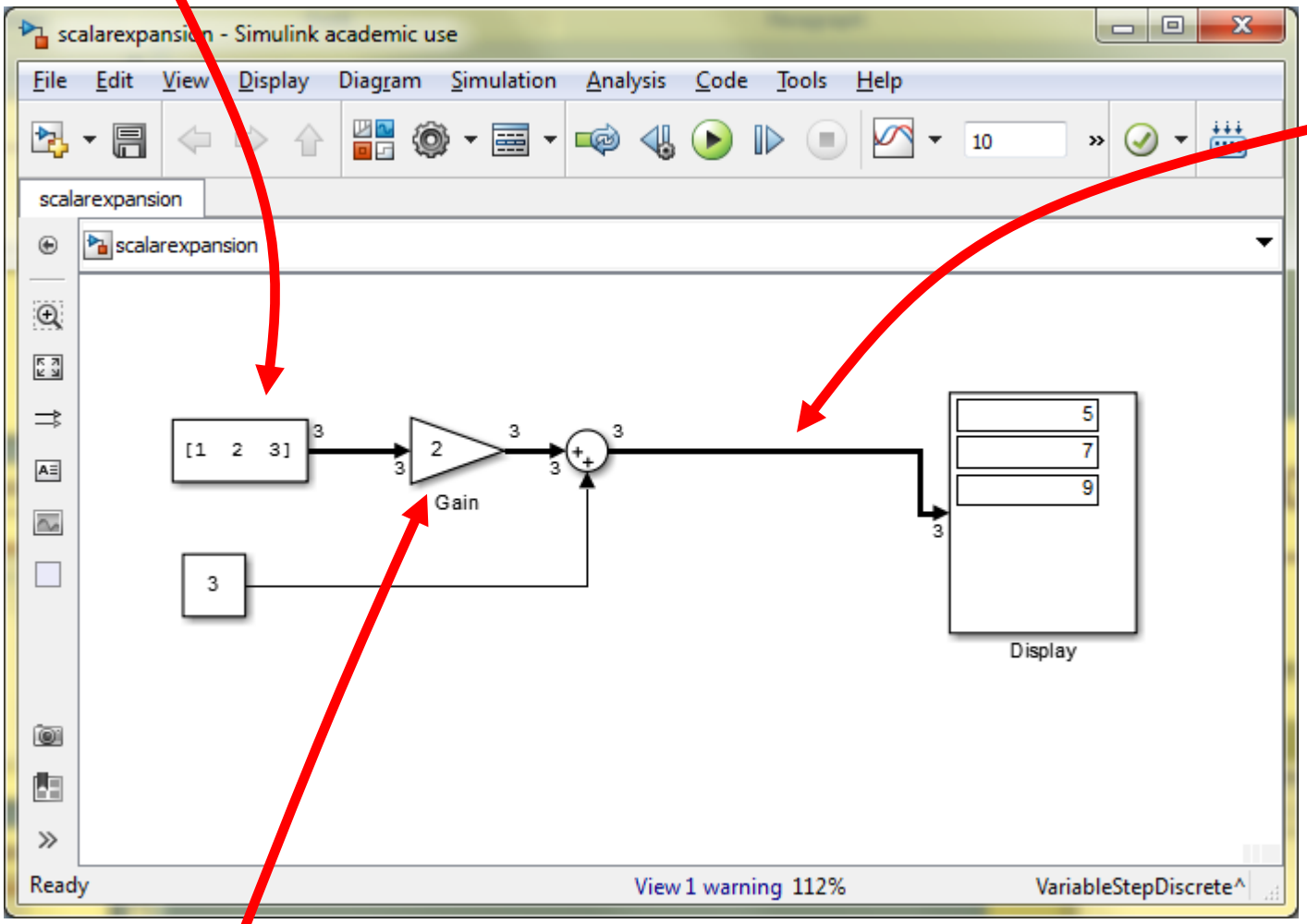

```
>> scalarexpansion
```

Vector I/O and Scalar Expansion



Vector I/O and Scalar Expansion

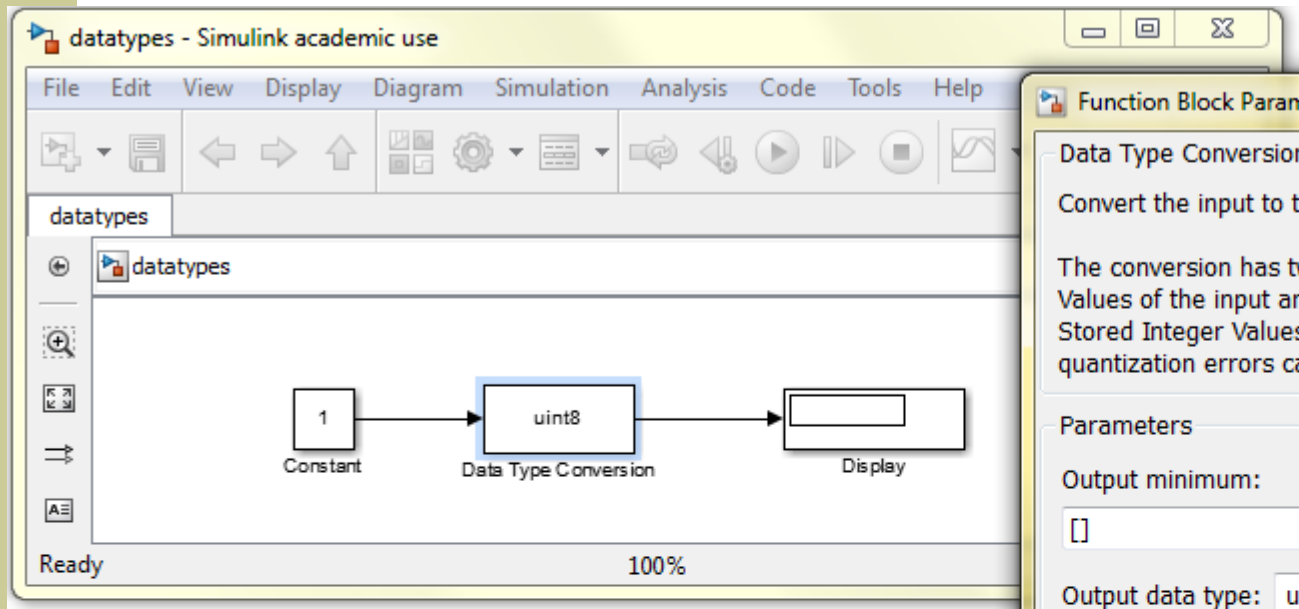
Scalar expansion of parameter

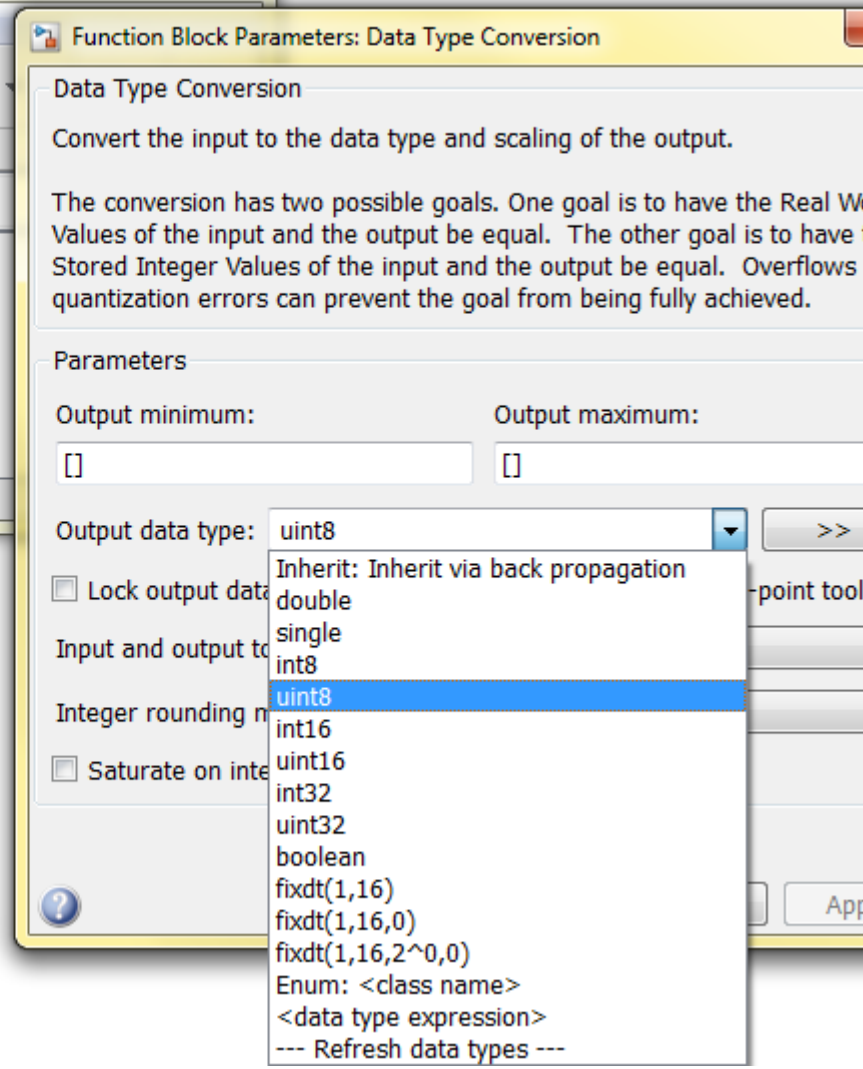


Wide Vector Lines and Line widths

Scalar expansion of inputs

Data Types

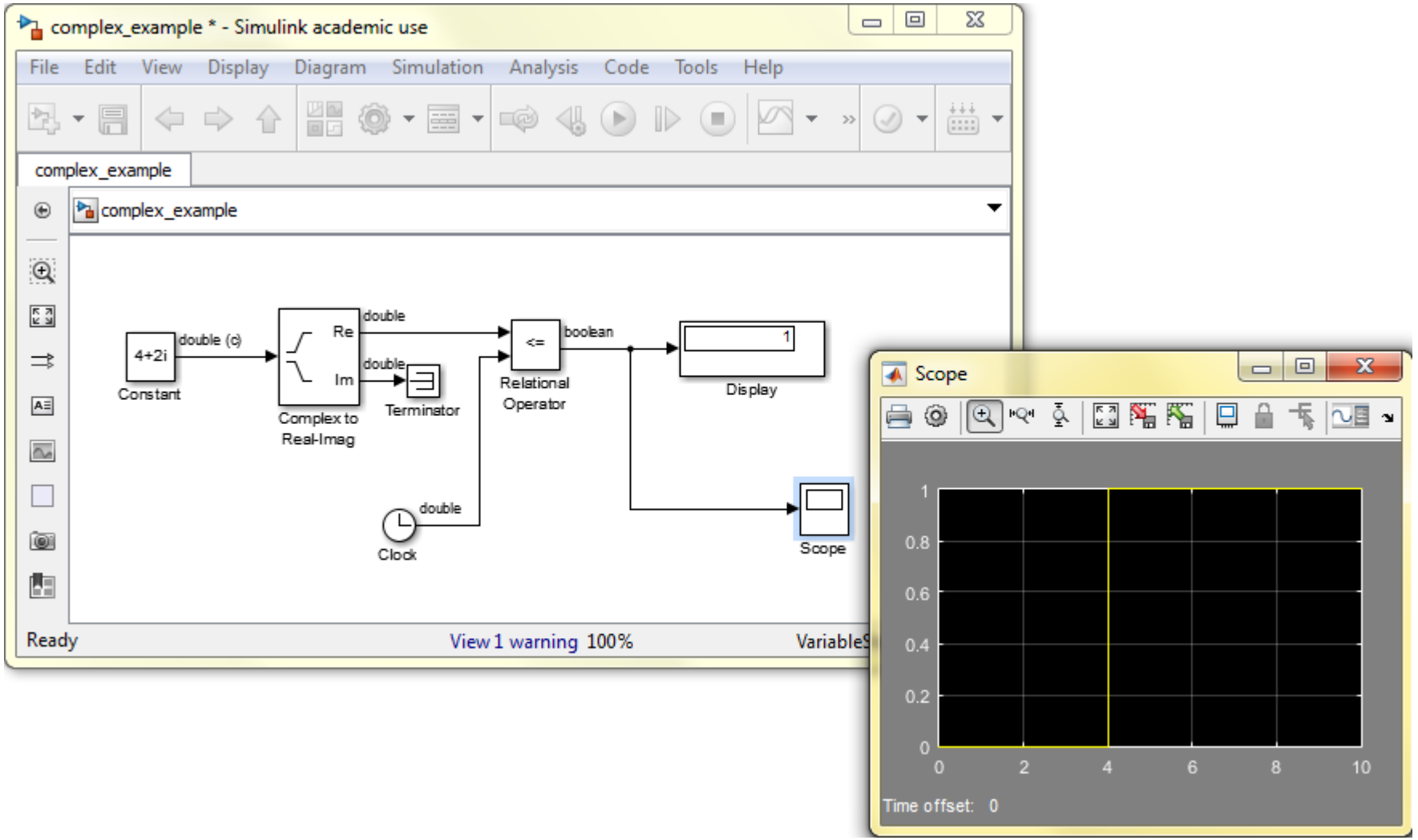




- Conversion Blocks
- Multiple Data types supported for most signal and block parameters
- Complex/Real conversions
- Automatic typecasting of parameters to signal type

```
>> datatypes
```

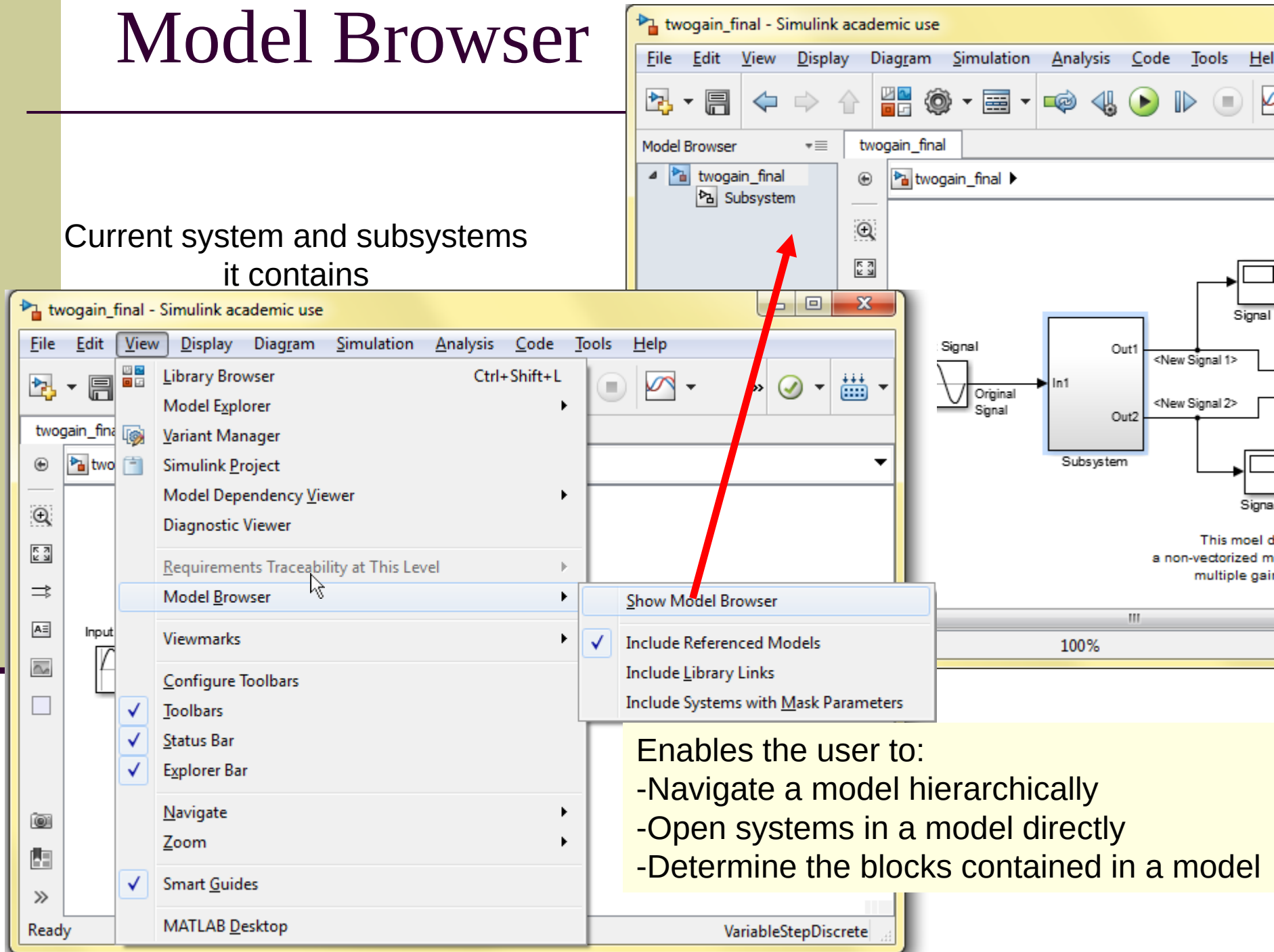
Data Types




```
>> complex_example
```

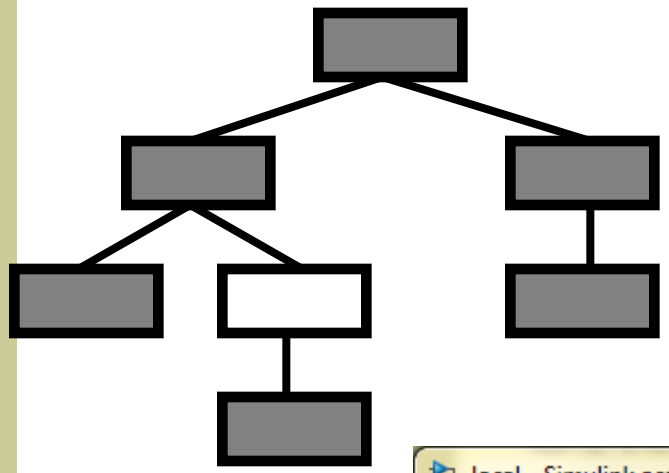
Model Browser

Current system and subsystems it contains

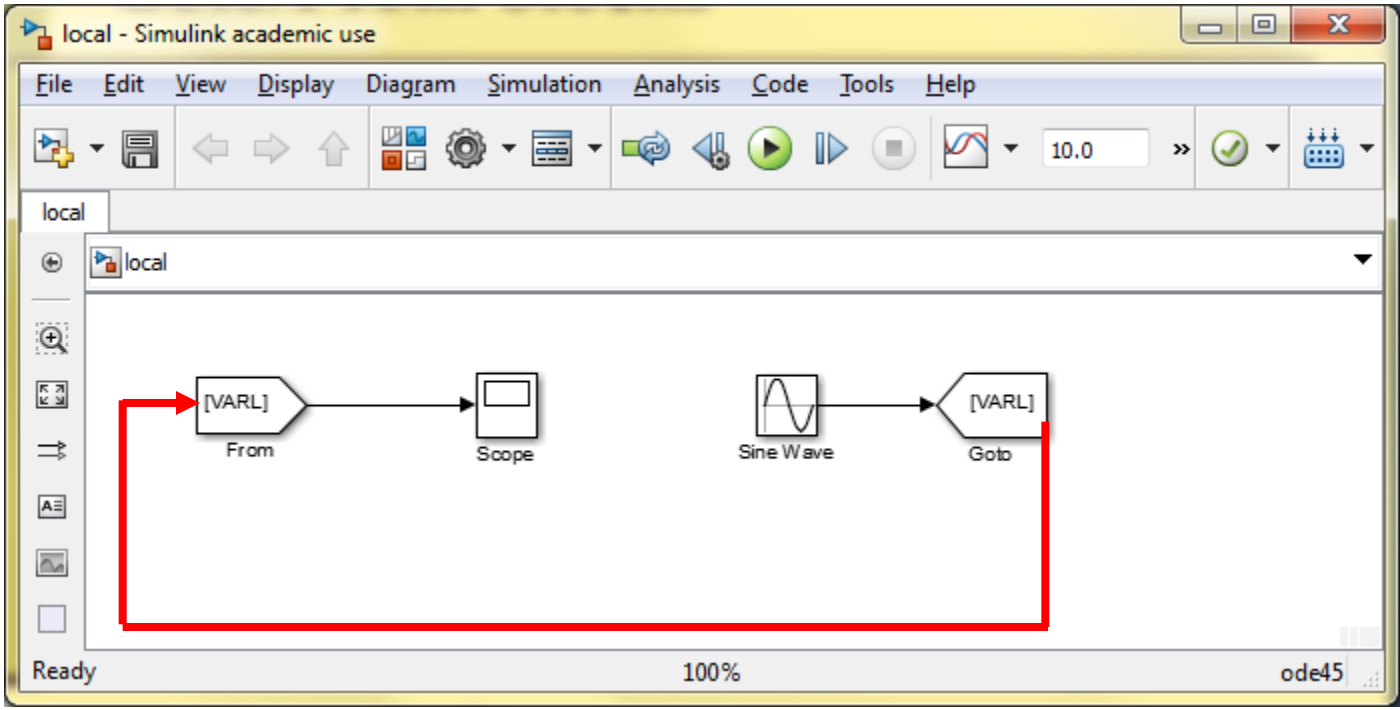


Enables the user to:
-Navigate a model hierarchically
-Open systems in a model directly
-Determine the blocks contained in a model

Goto/From blocks

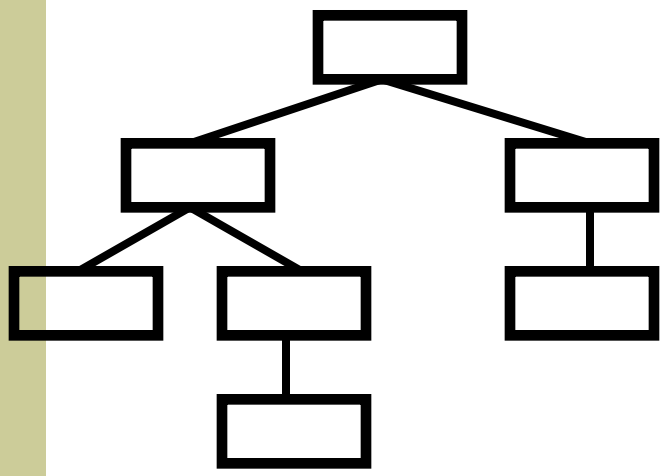


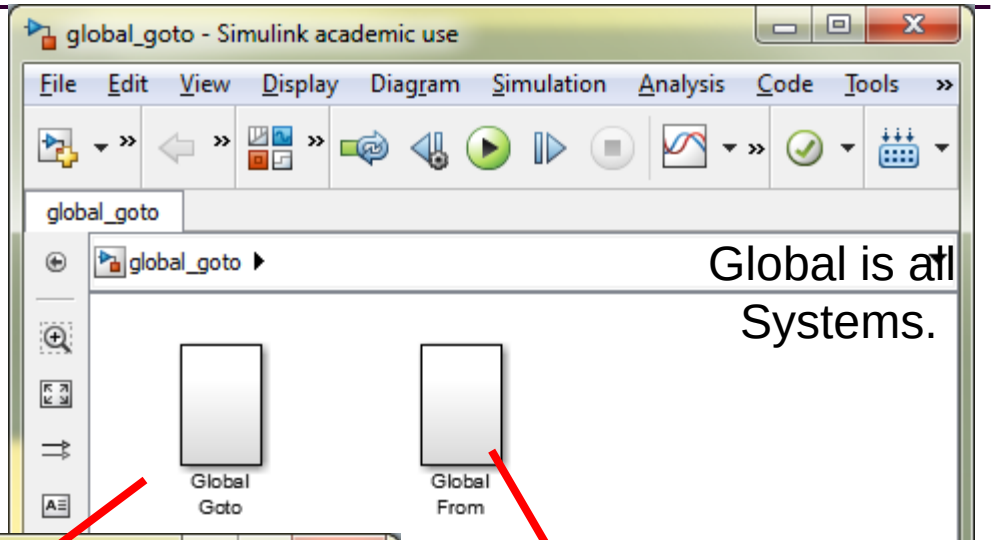
Local [ ] is only on the current window

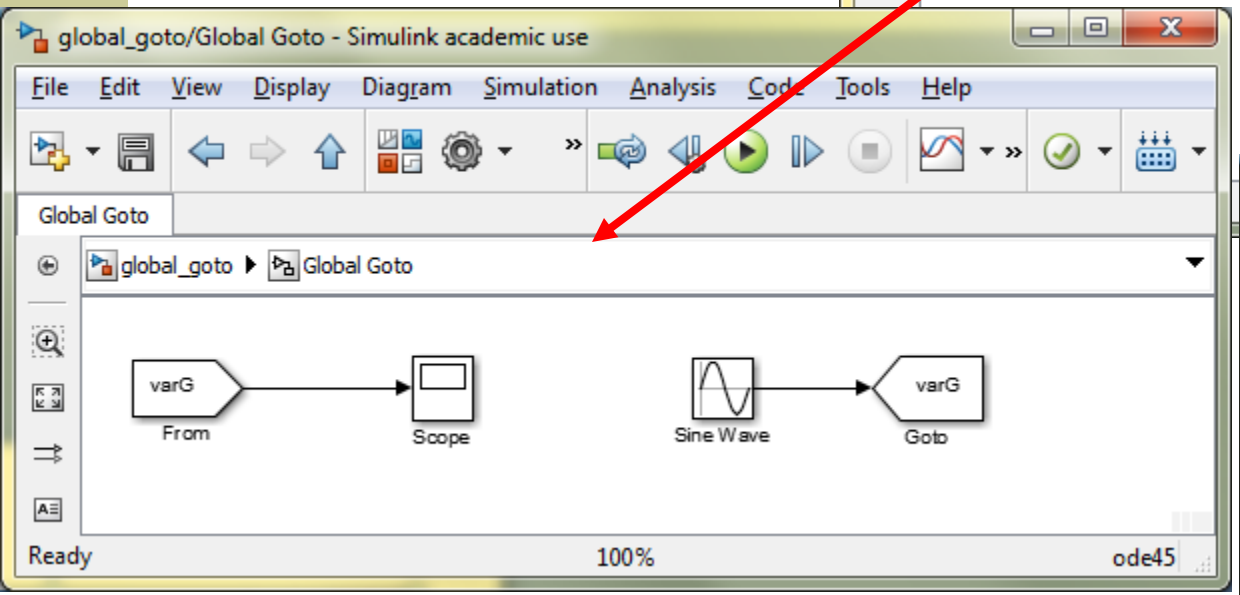


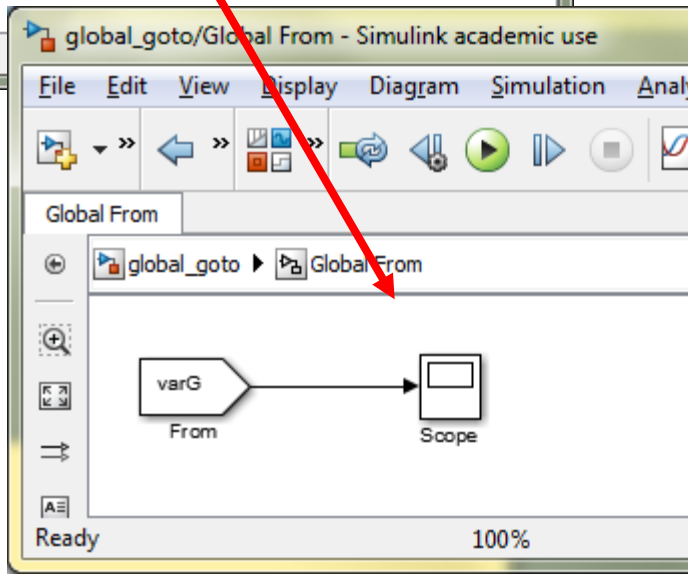

```
>> local
```

Goto / From blocks





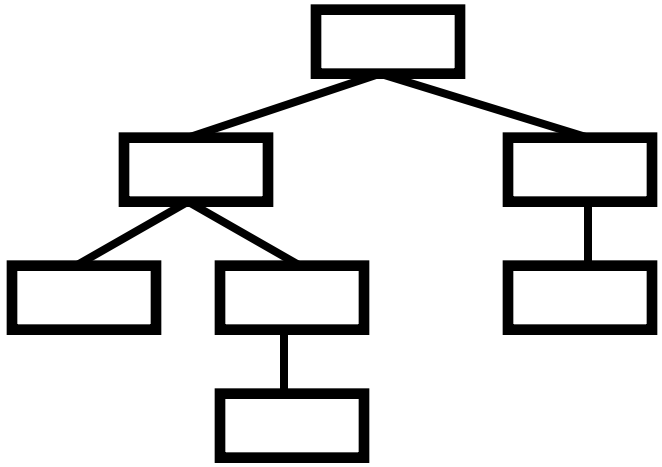


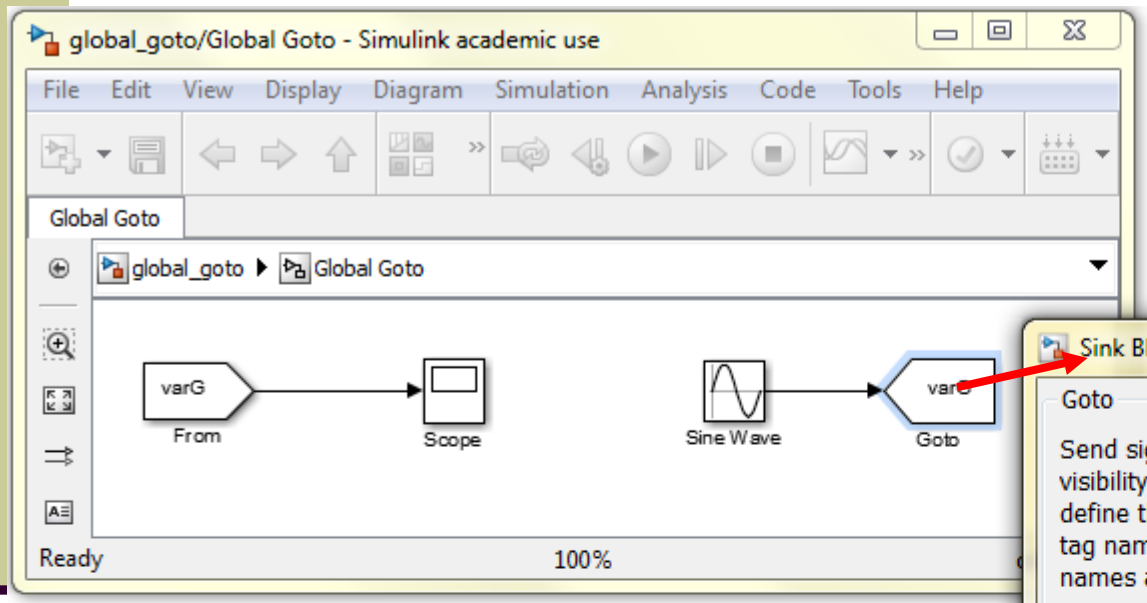


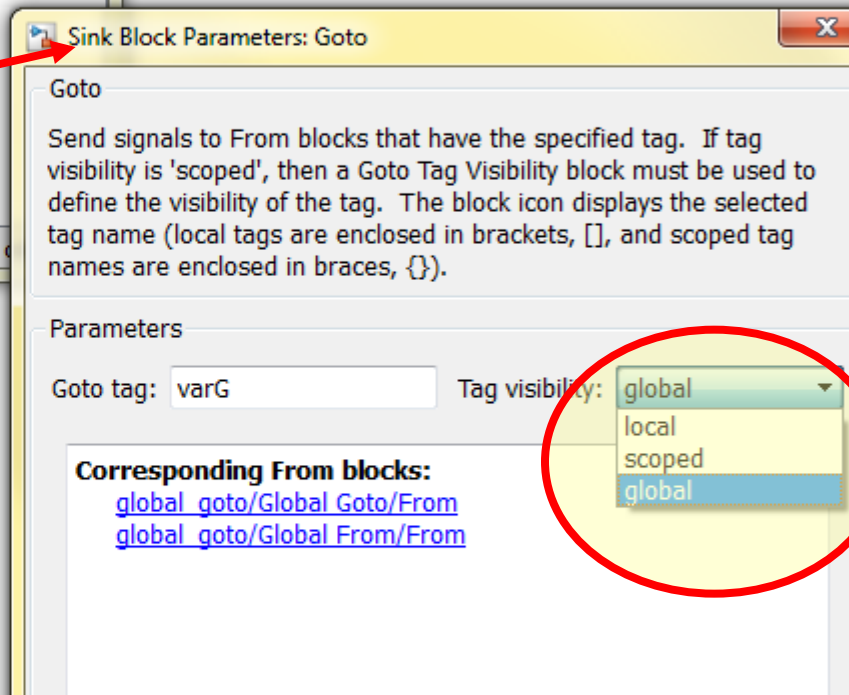

```
>> global_goto
```

Goto / From blocks

Global is all Systems.



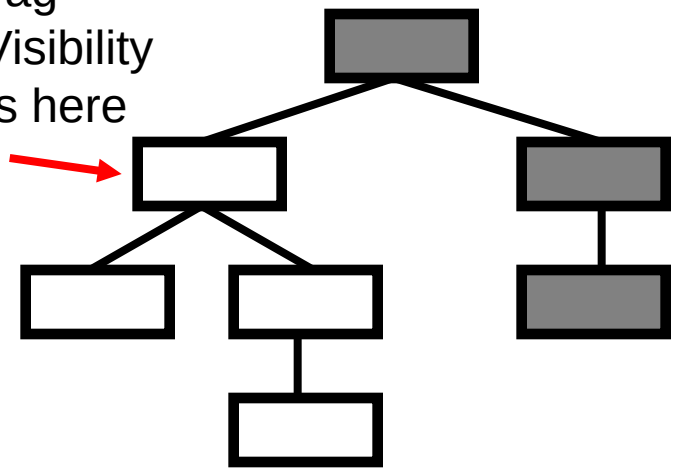




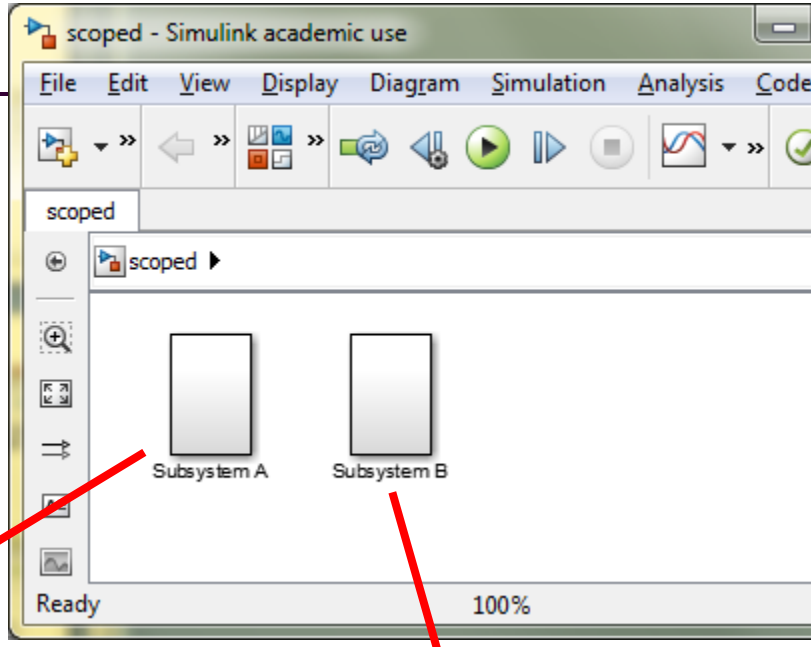

```
>> global_goto
```

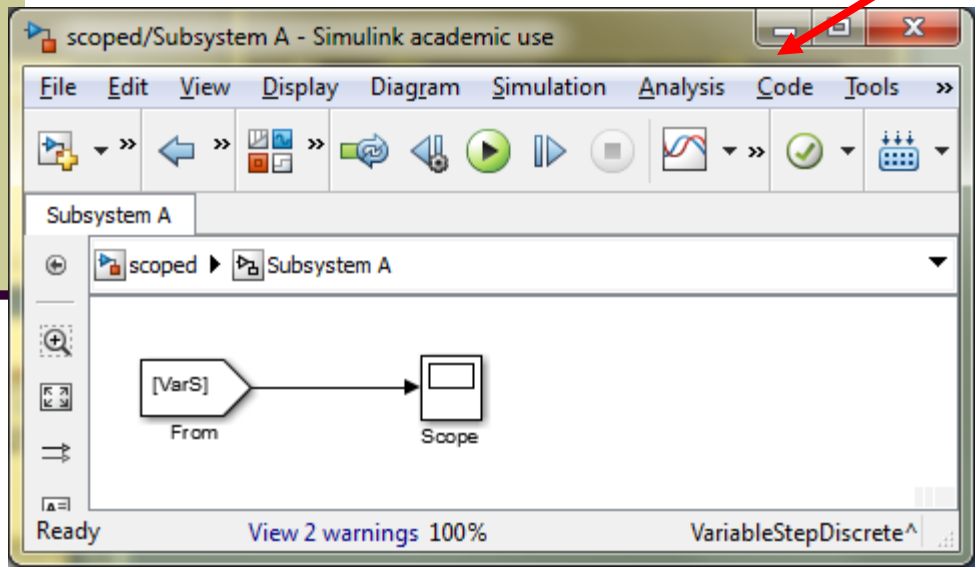
Goto / From blocks

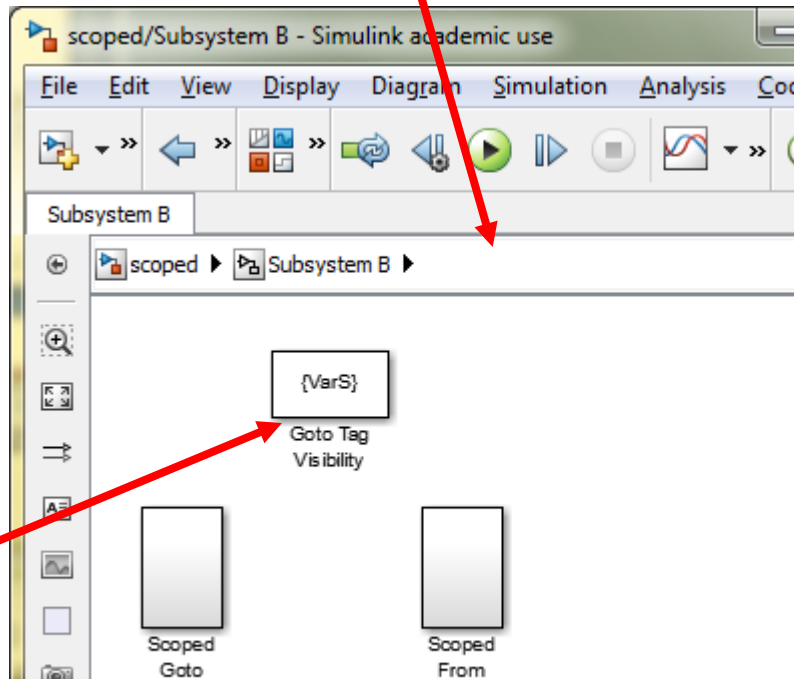
Tag
Visibility
Is here



>> scoped

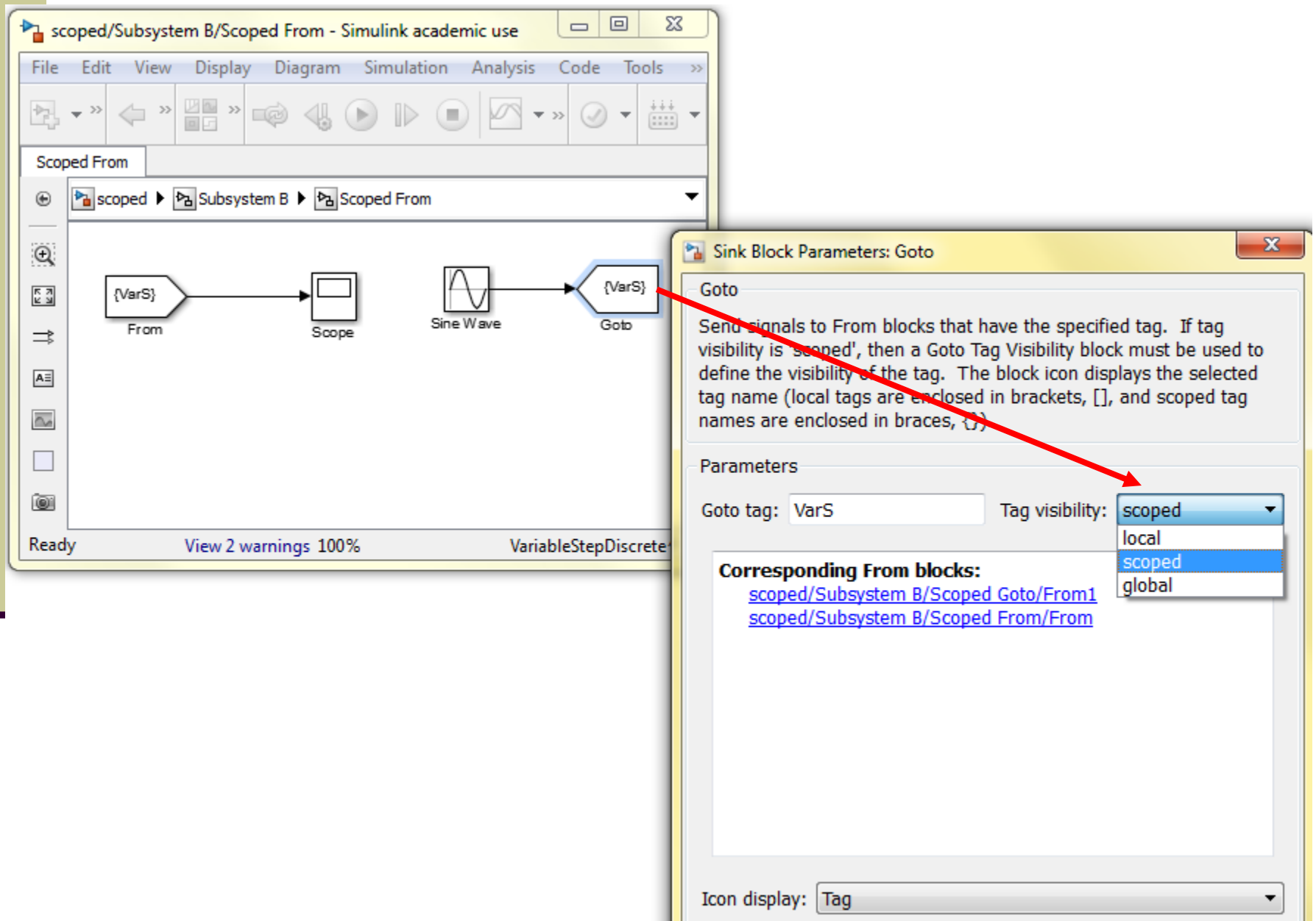




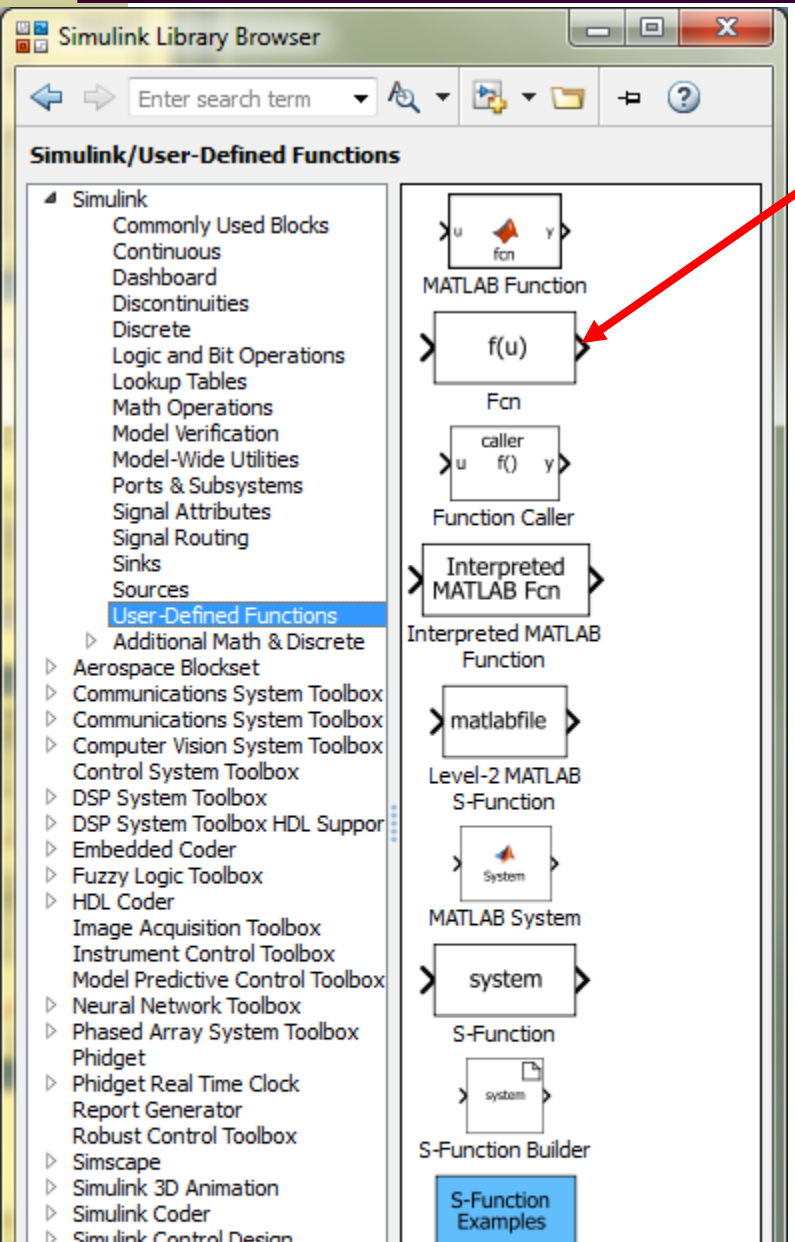


Scoped { } is all systems
under the Tag Visibility block.

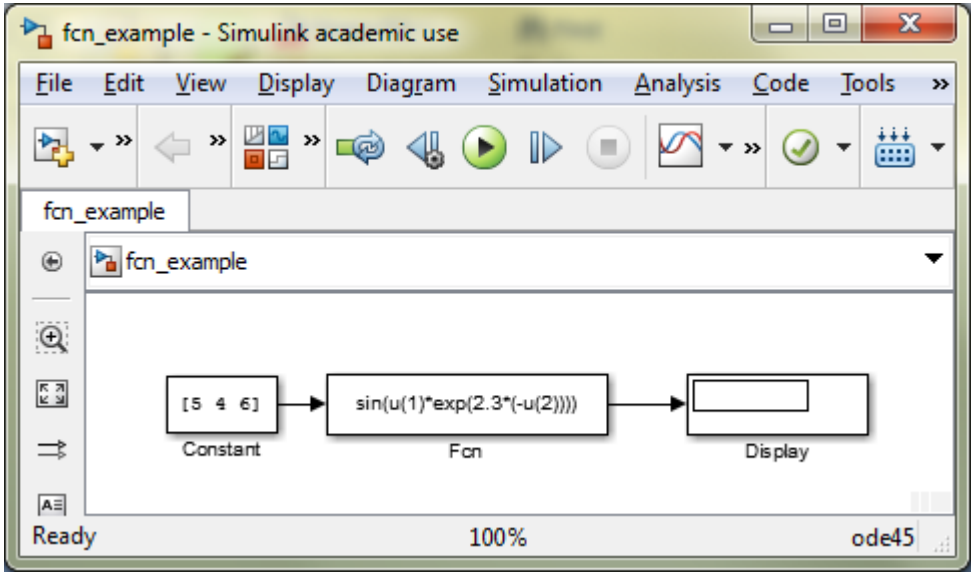
Goto/From blocks



User Defined Functions



Allows general expression in terms of u.



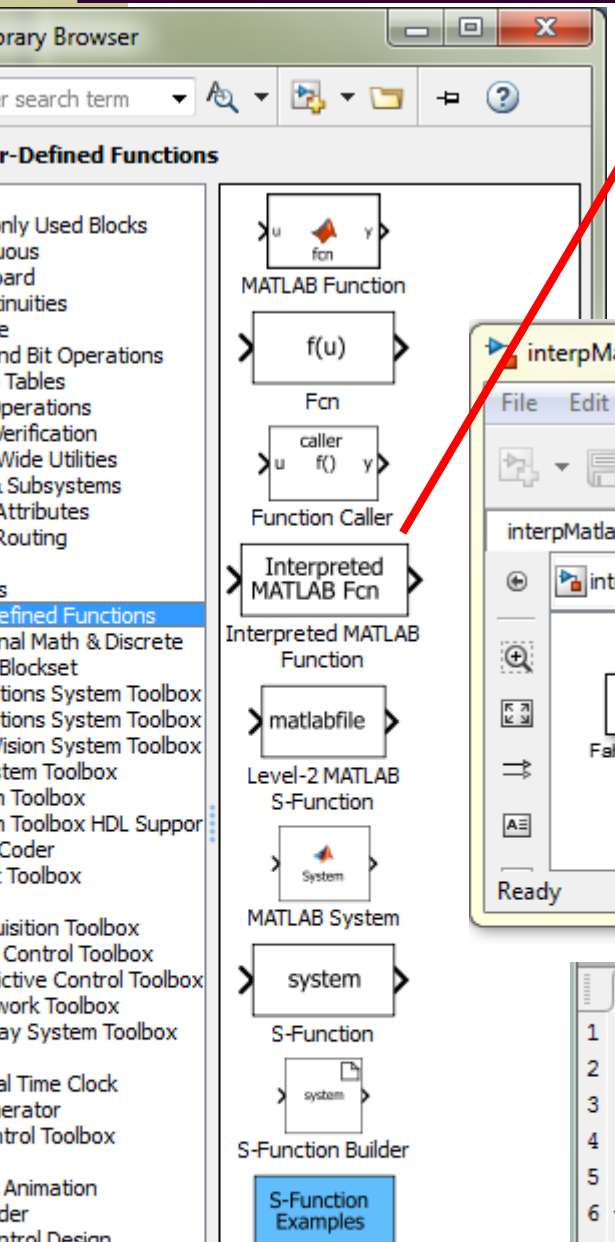
If signal are vectors, index into u.
sin(u(1)*exp(2.3*(-u(2))))

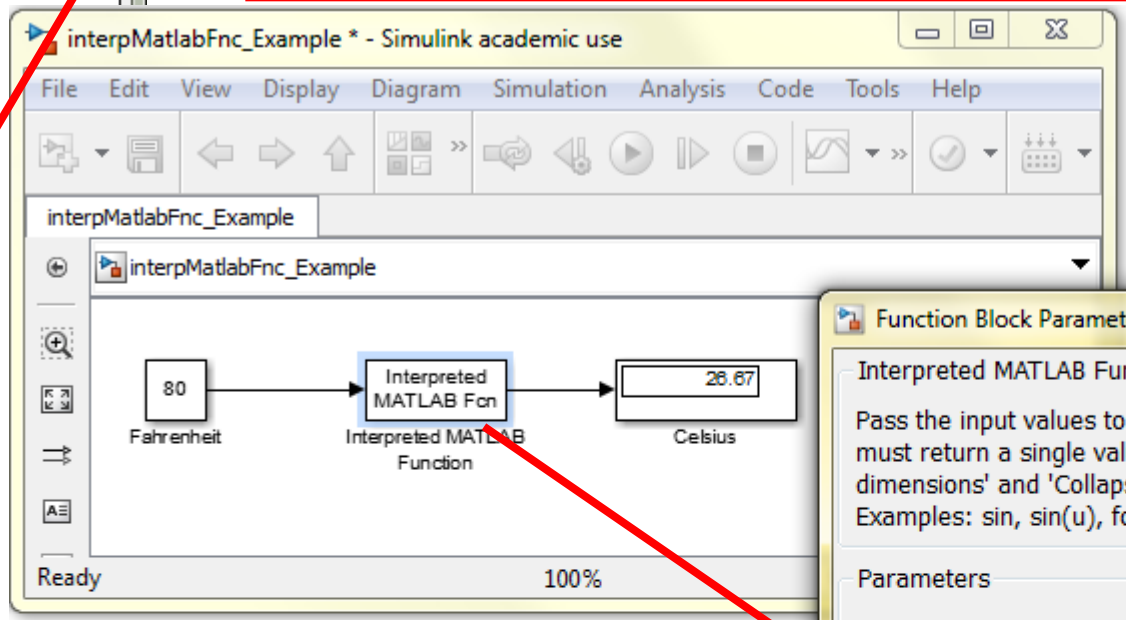
```
>> fcn_example
```

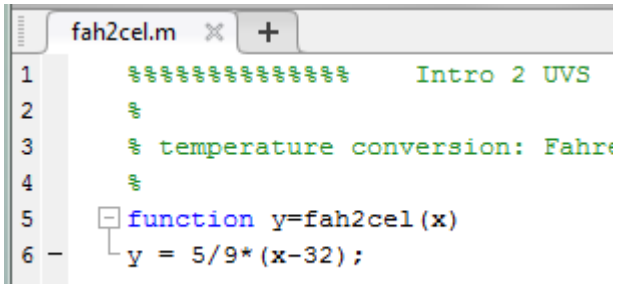
User Defined Functions

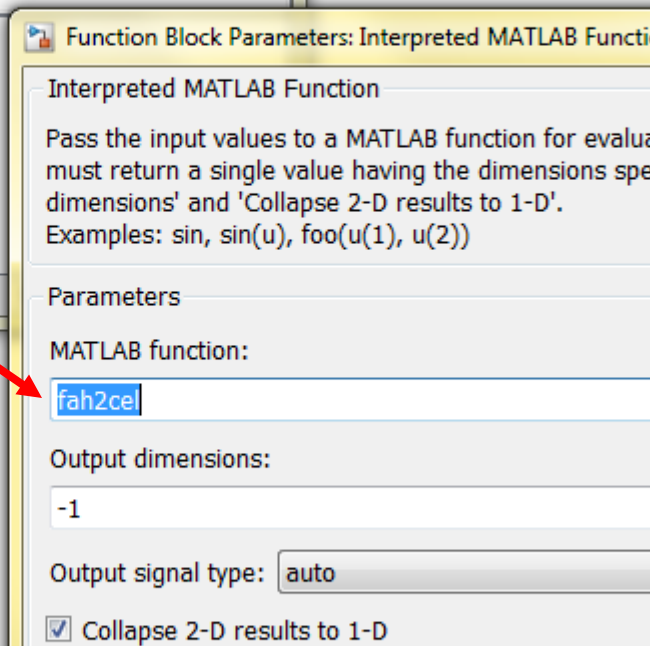
Passed the input to
a MATLAB function for evaluation

Note: “Interpreted MATLAB Function” Block is slower than the Fcn Block because it calls the MATLAB parser during each integration step. Consider using Fnc block or “MATLAB Function” blocks

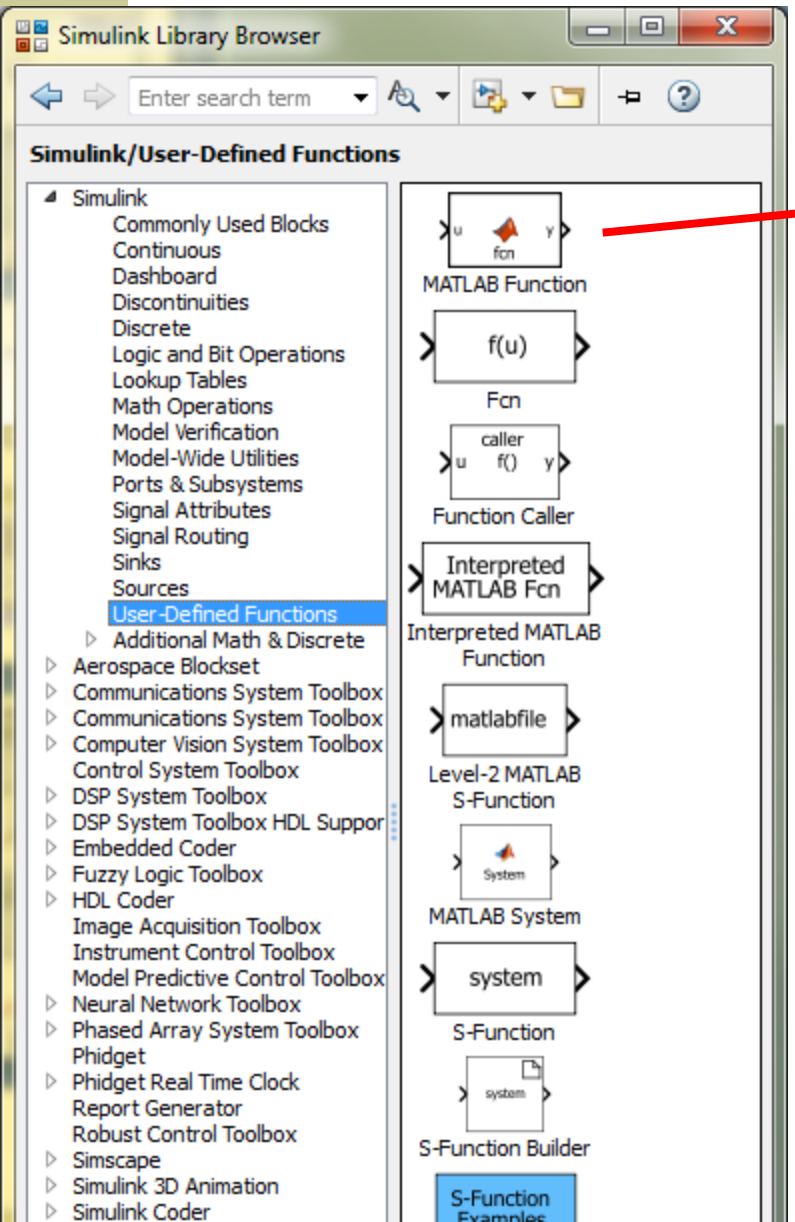








“MATLAB Function” Blocks



You can write a MATLAB function for use in a Simulink model.

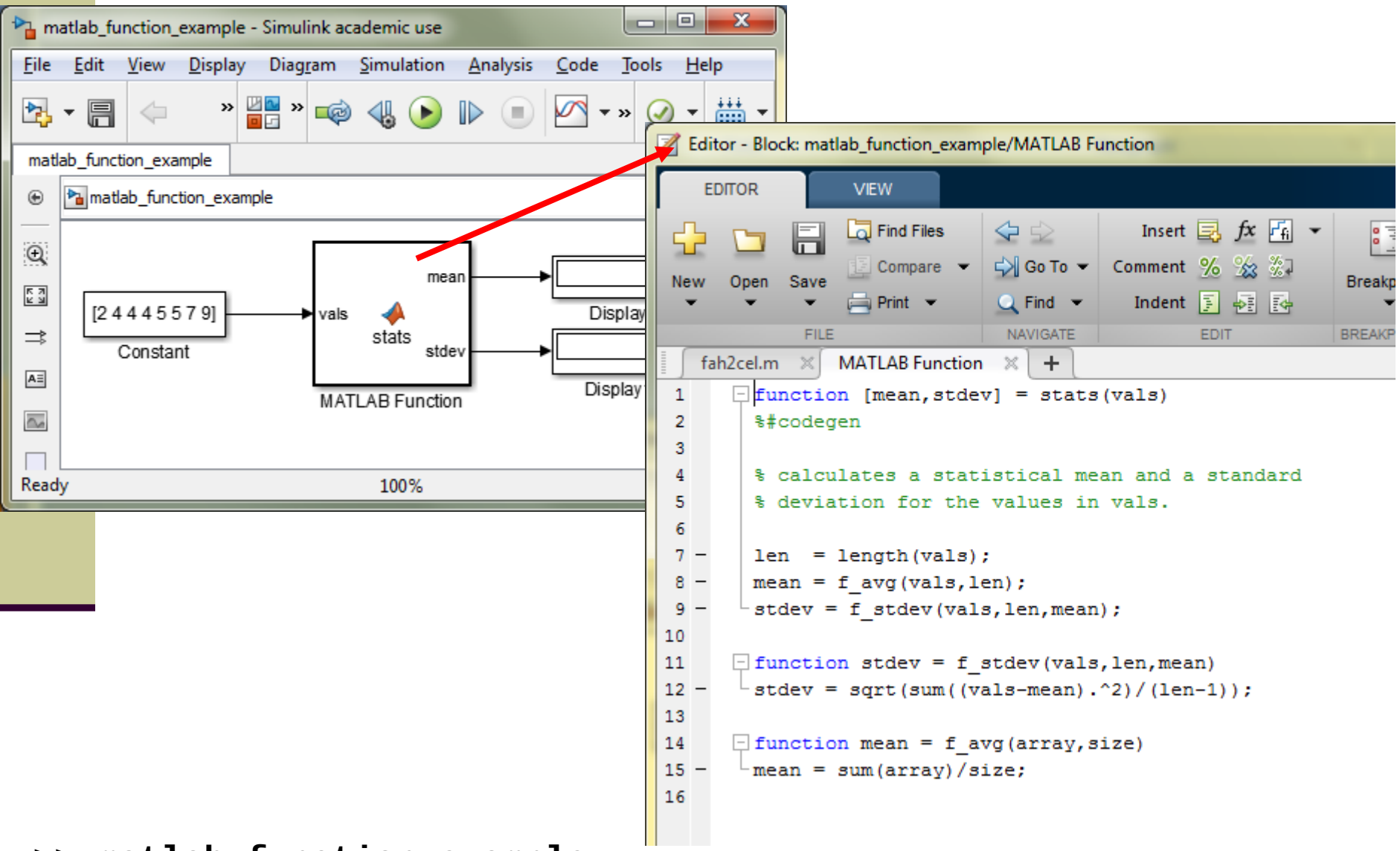
The MATLAB function you create executes for simulation and generates code for a Simulink Coder target.

The MATLAB Function block can generate efficient embeddable code.

The MATLAB Function block supports only a subset of the functions available in MATLAB

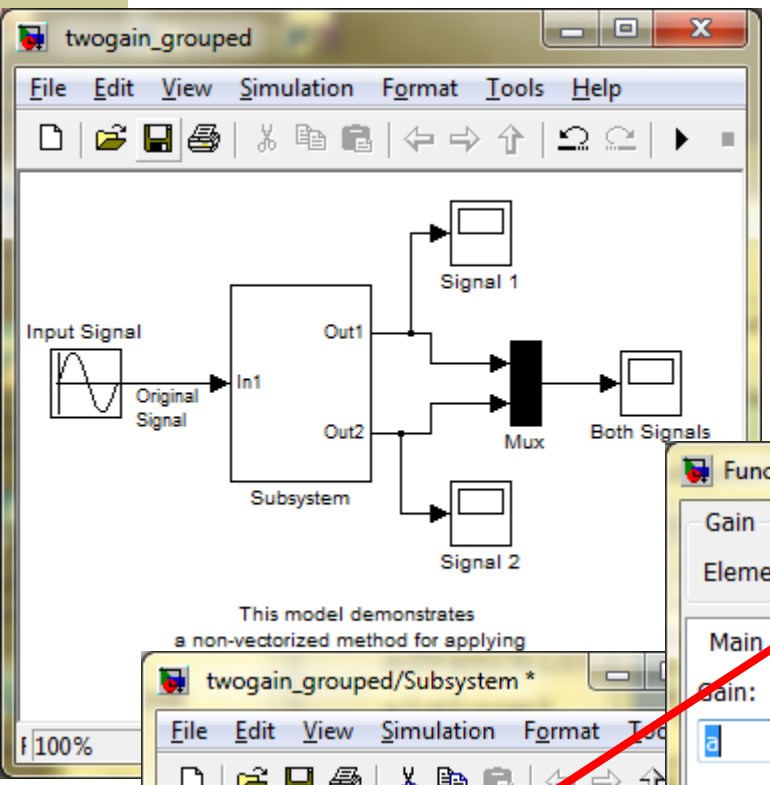
```
>> matlab_function_example
```

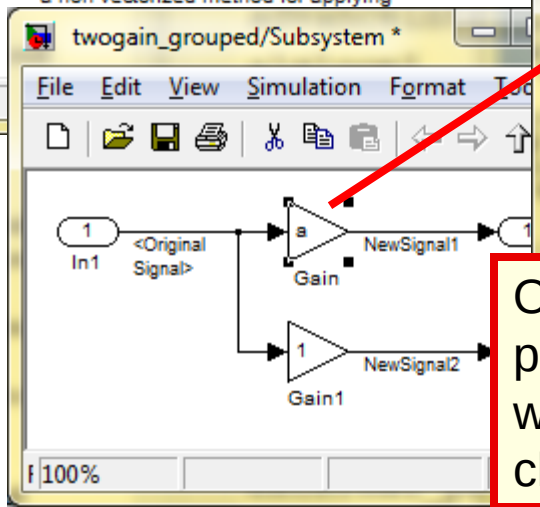
“MATLAB Function” Blocks

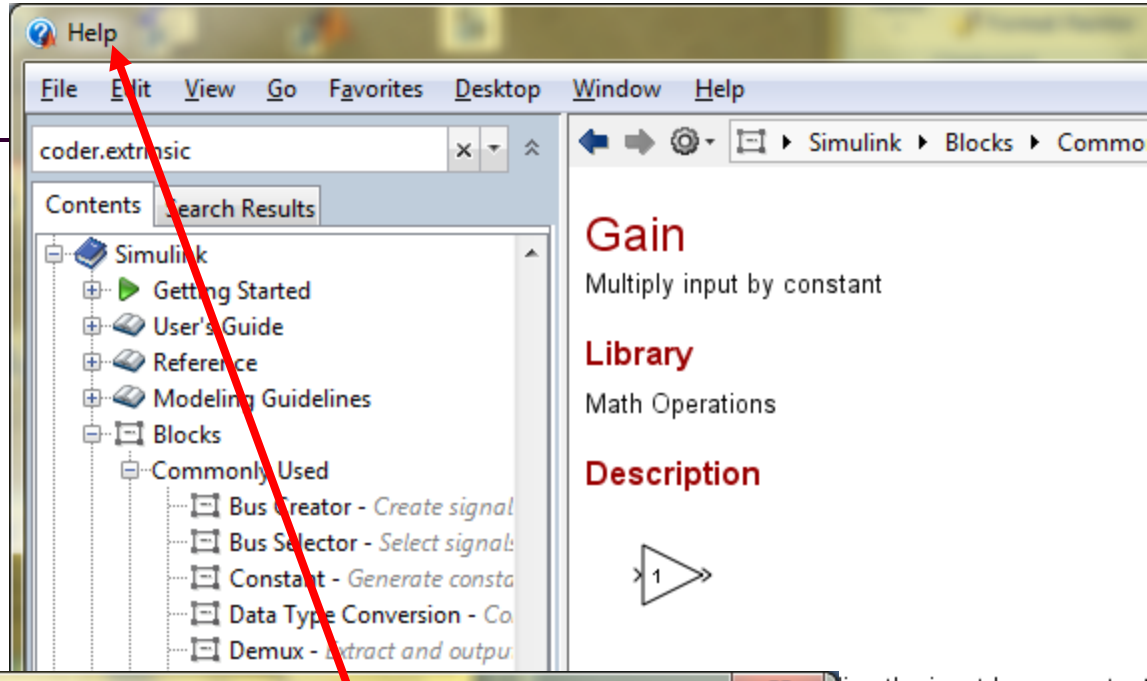


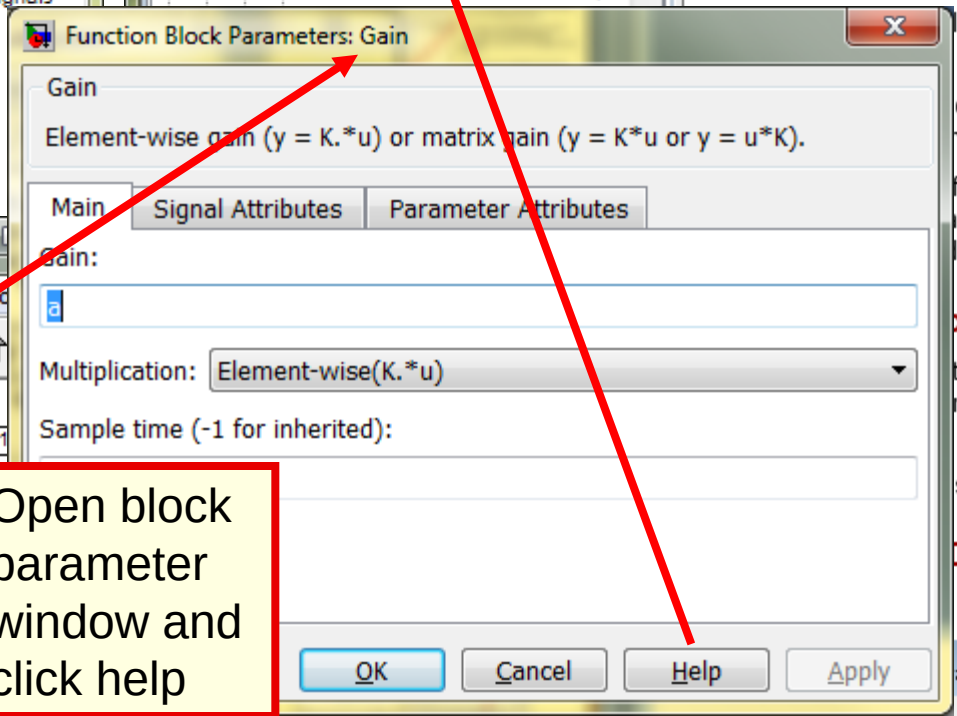

```
>> matlab_function_example
```

Getting help



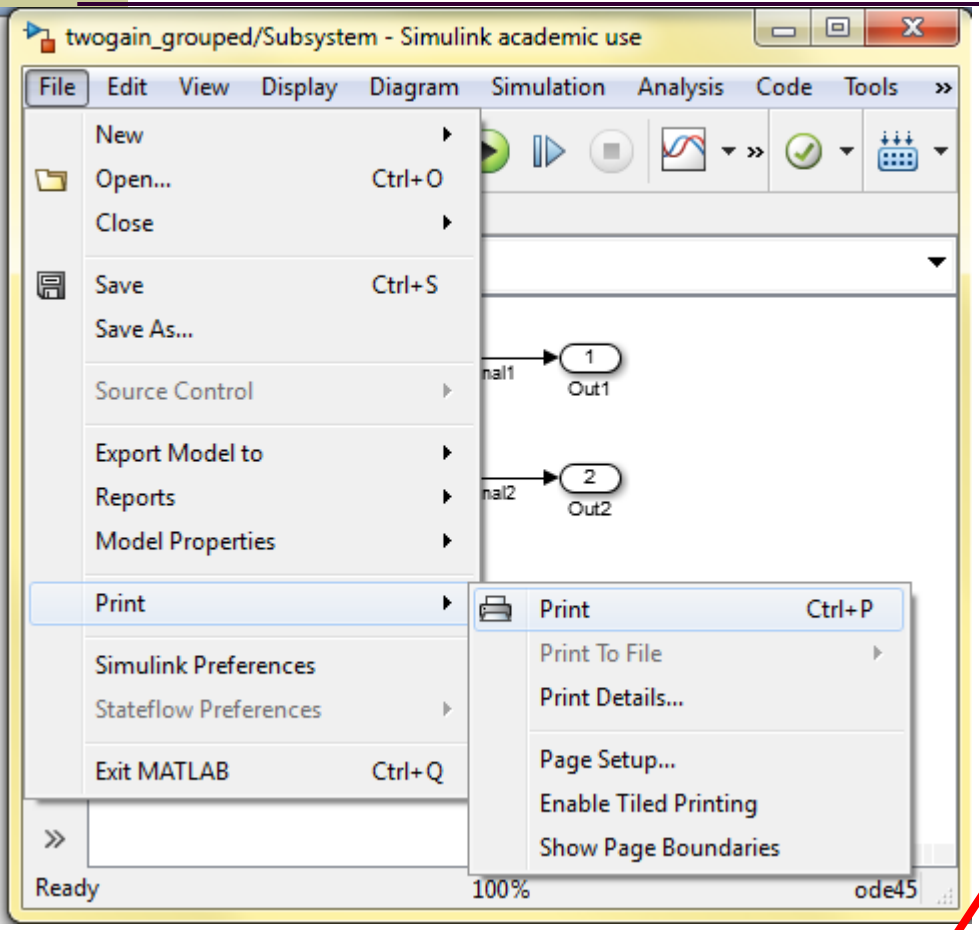


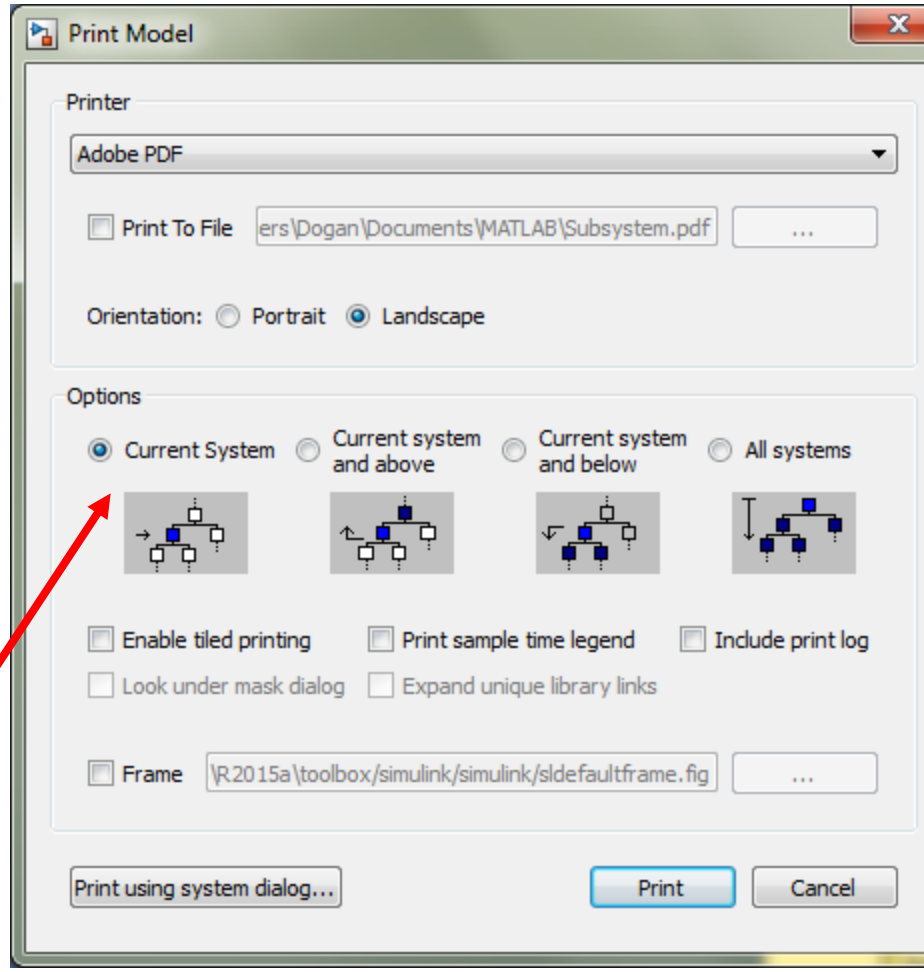




Open block parameter window and click help

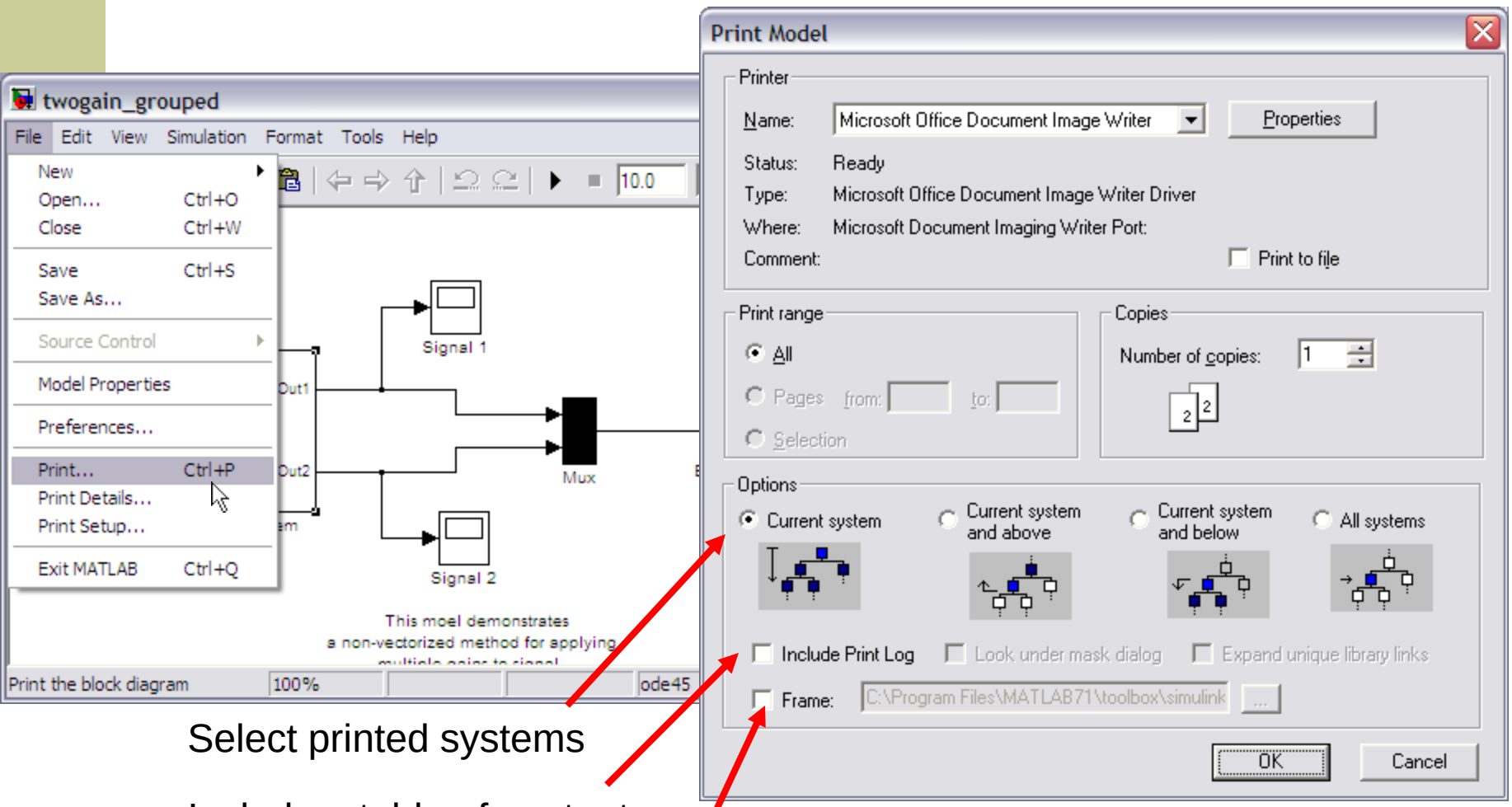
Print Dialog Box





Select printed systems

Print Dialog Box



Select printed systems

Include a table of contents

Include a frame

Tips for building a model

"You should be able to know what is going on in a diagram by looking at the printout"

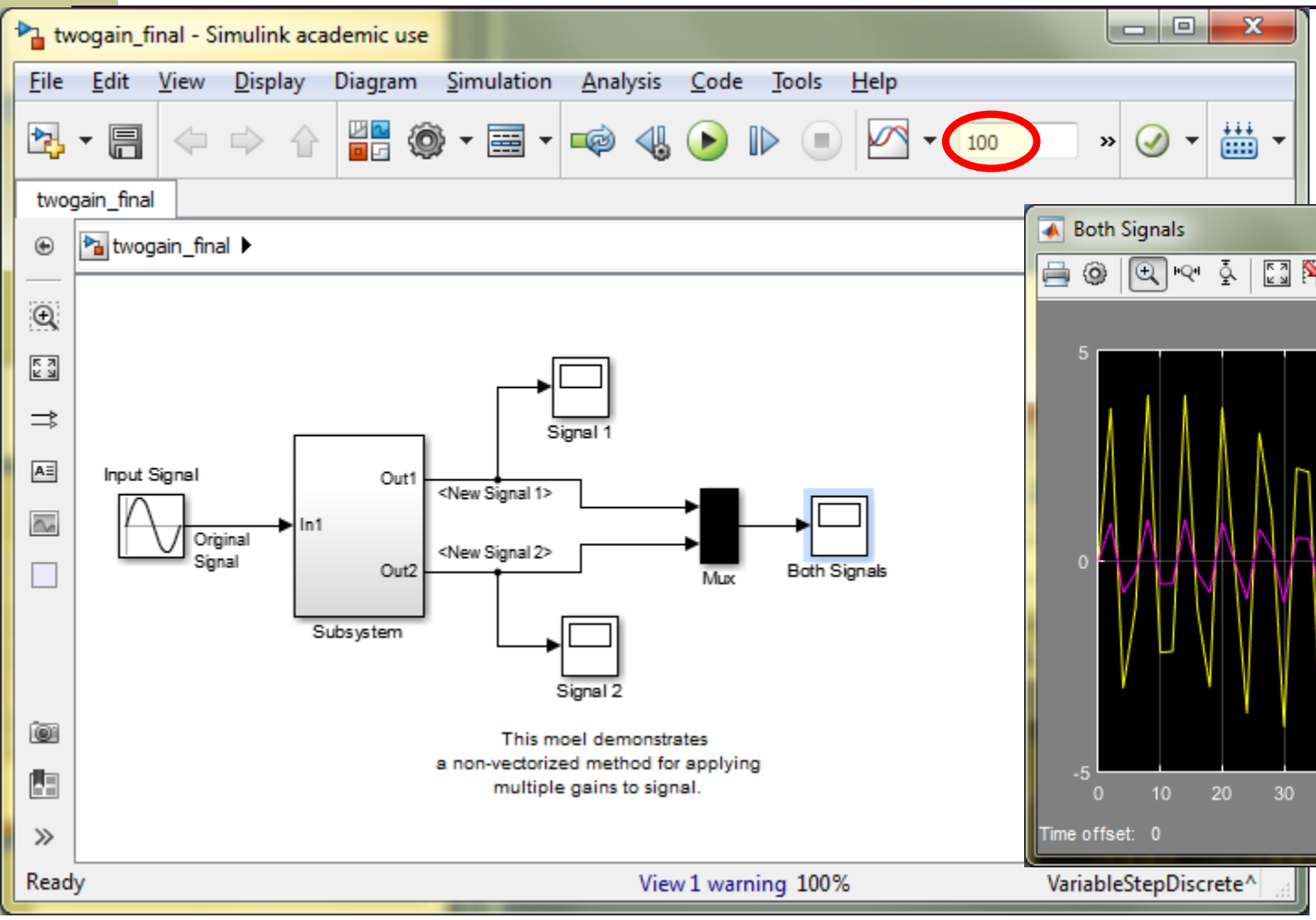
- Minimize line crossings
- Remove titles when not needed
- Show dialog parameters in title
- Use Goto/From Block wisely
- Size blocks appropriately
- Use subsystems appropriately

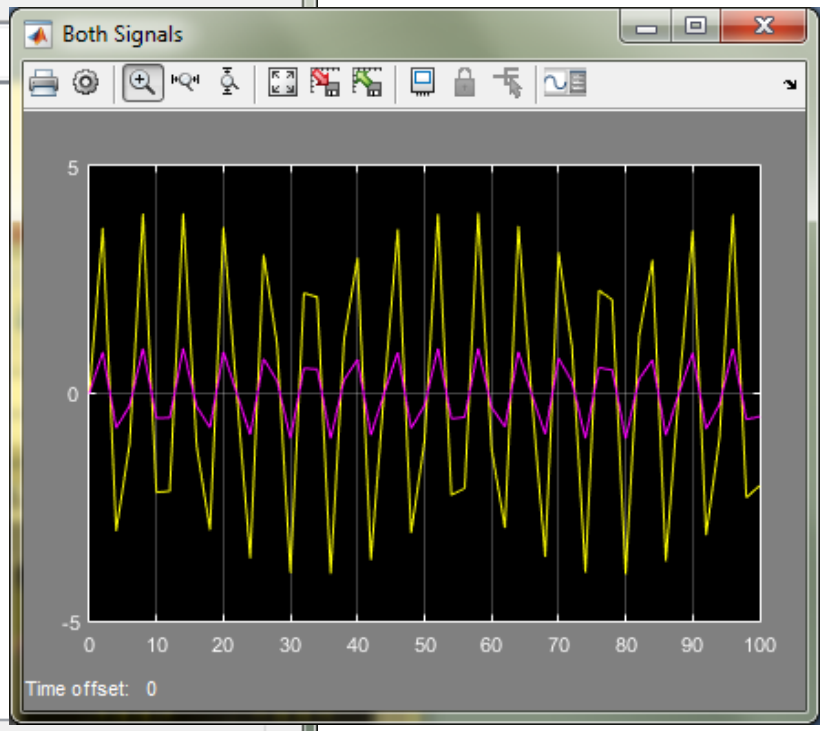
Creating a model summary

You should be able to do the following

- Open a new model window
- Navigate Simulink block libraries
- Annotate and label blocks and signals
- Edit models
- Change block parameters and characteristics
- Create and manage subsystems
- Save and print your models
- Get help on blocks
- Build a sensible model

Simulating



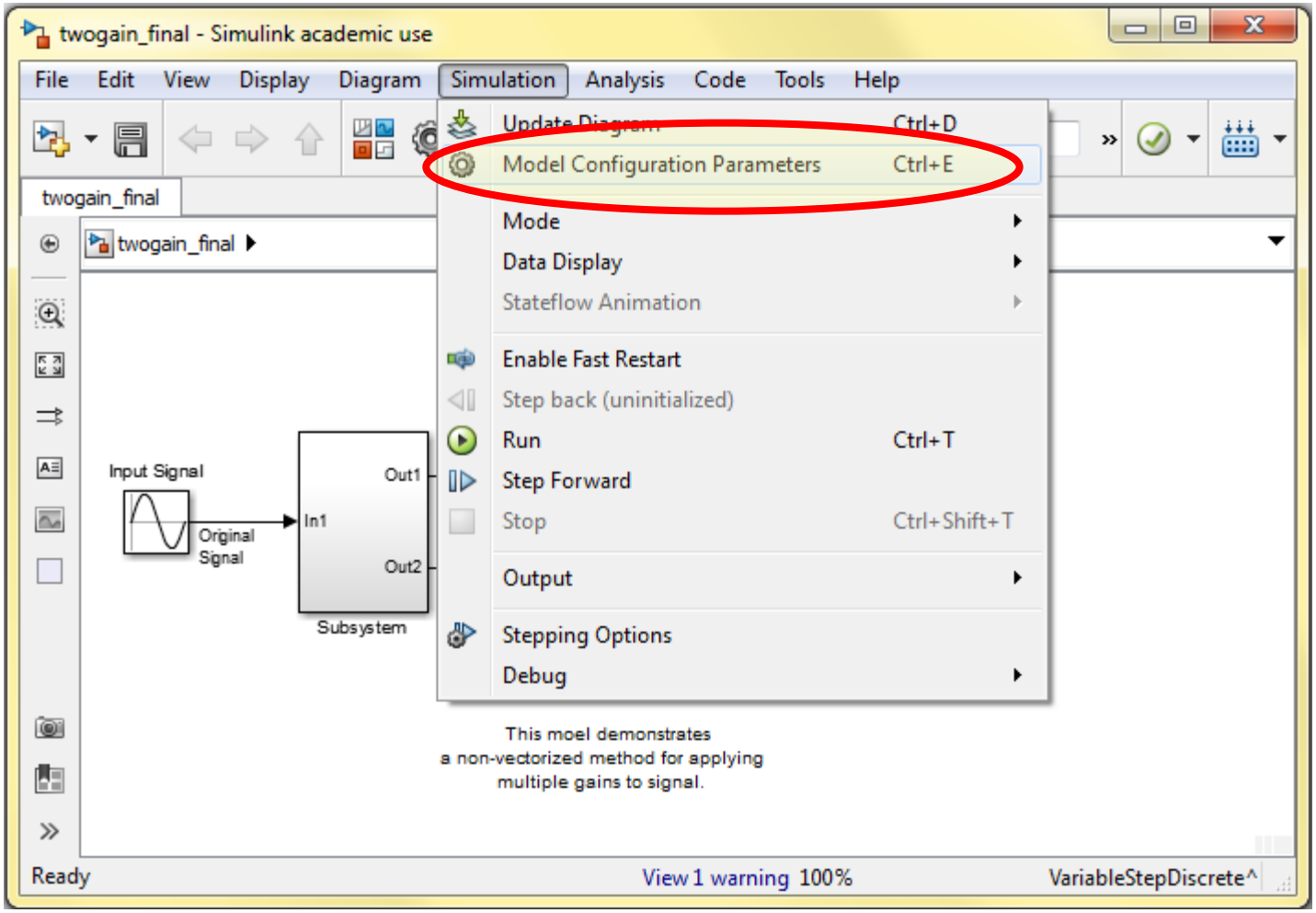


Stop time:
100 seconds

```
>> twogain_final
```

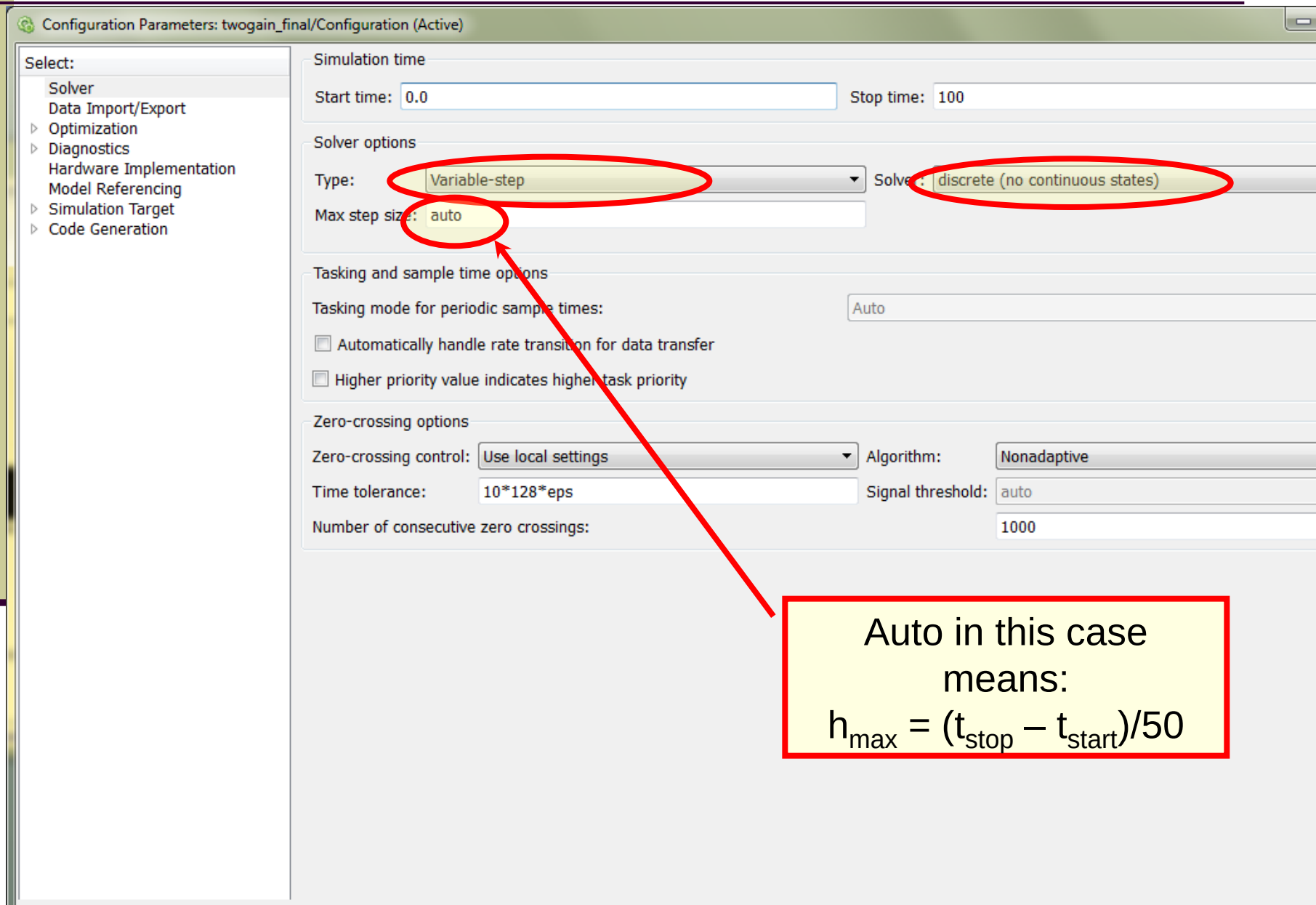
Final simulation: jagged!

Setting simulation parameters

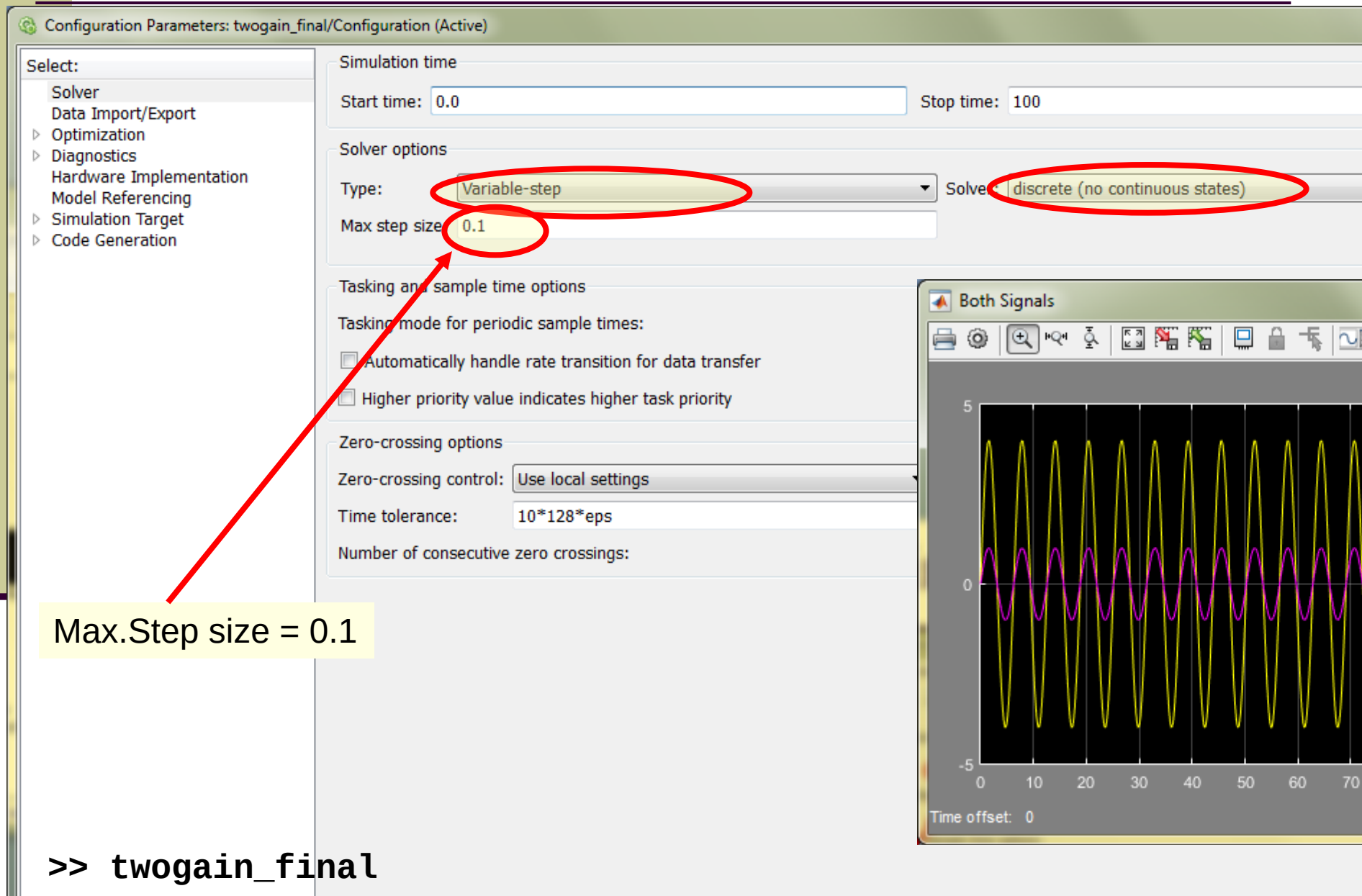



```
>> twogain_final
```

Setting simulation parameters



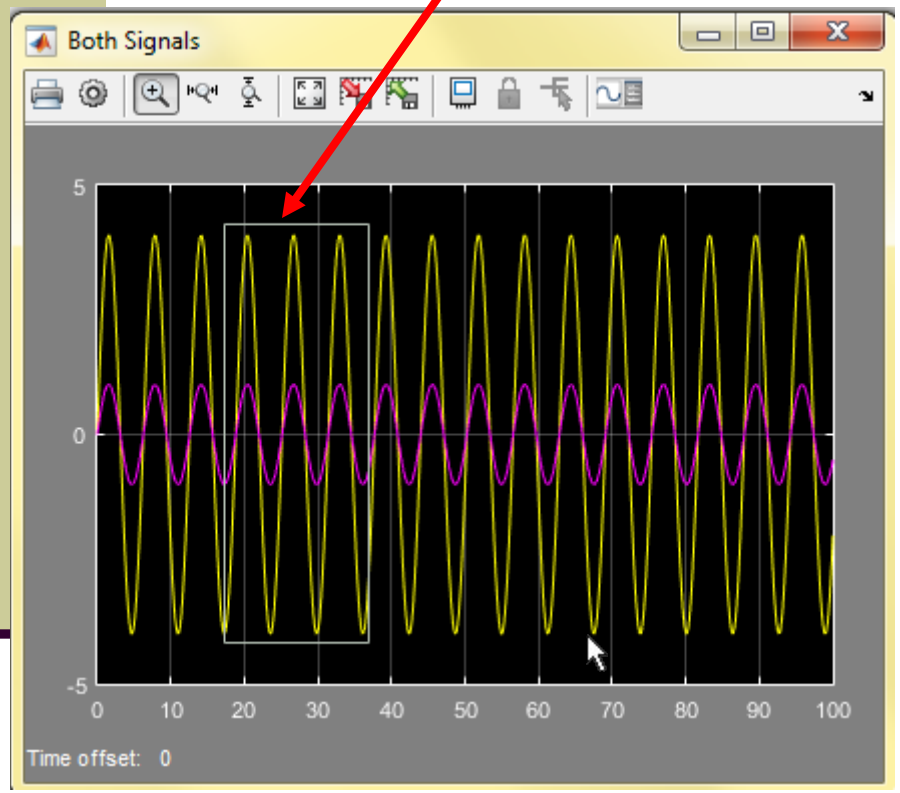
Setting simulation parameters

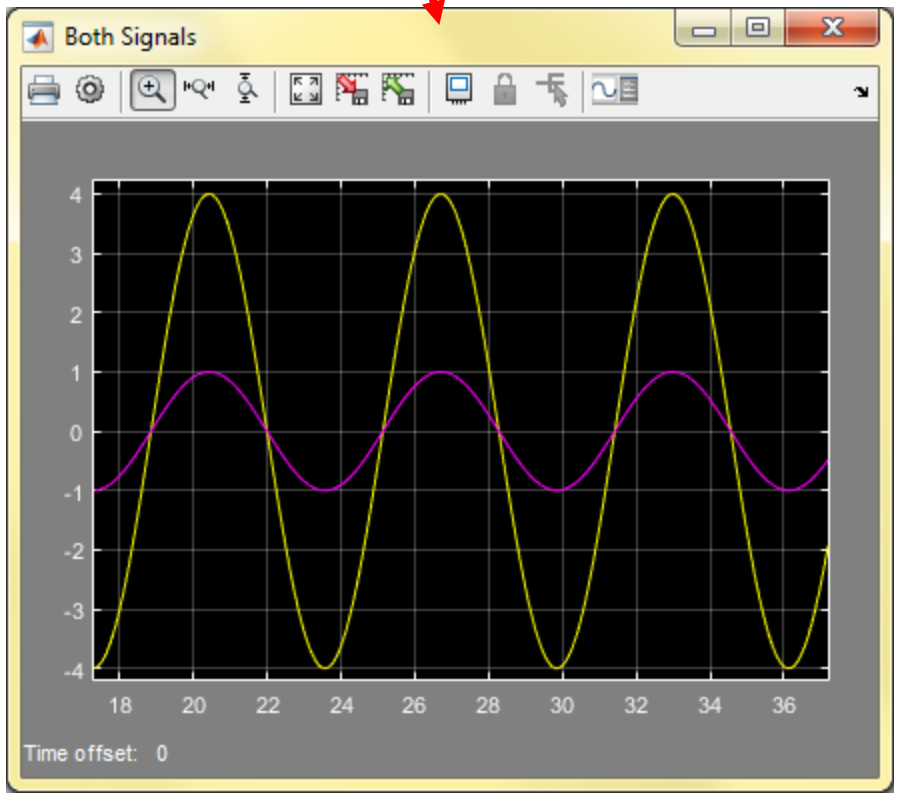


Simulating

Second Simulation: Much Better…
You can even zoom in to see.

If you're happy with these setting you can save or print them

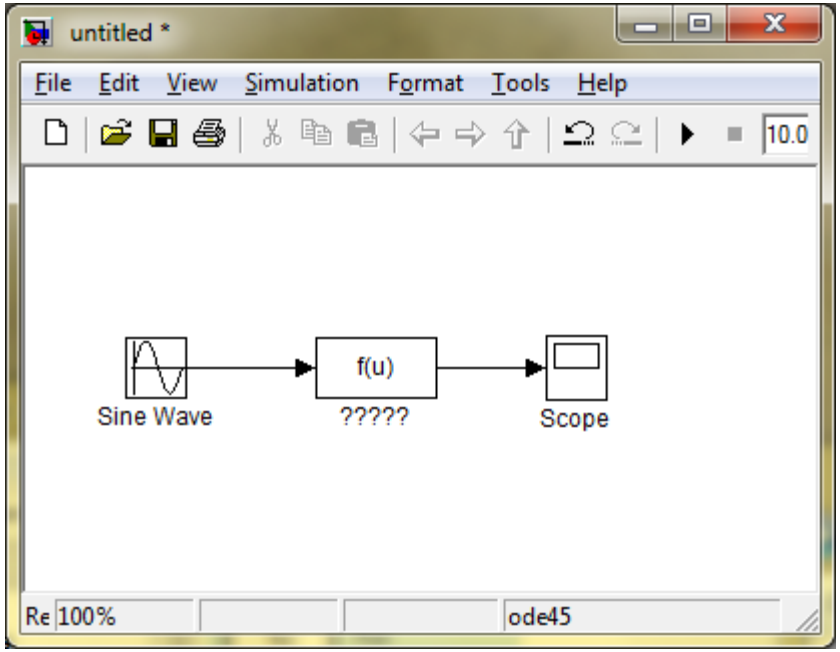





```
>> twogain_final
```

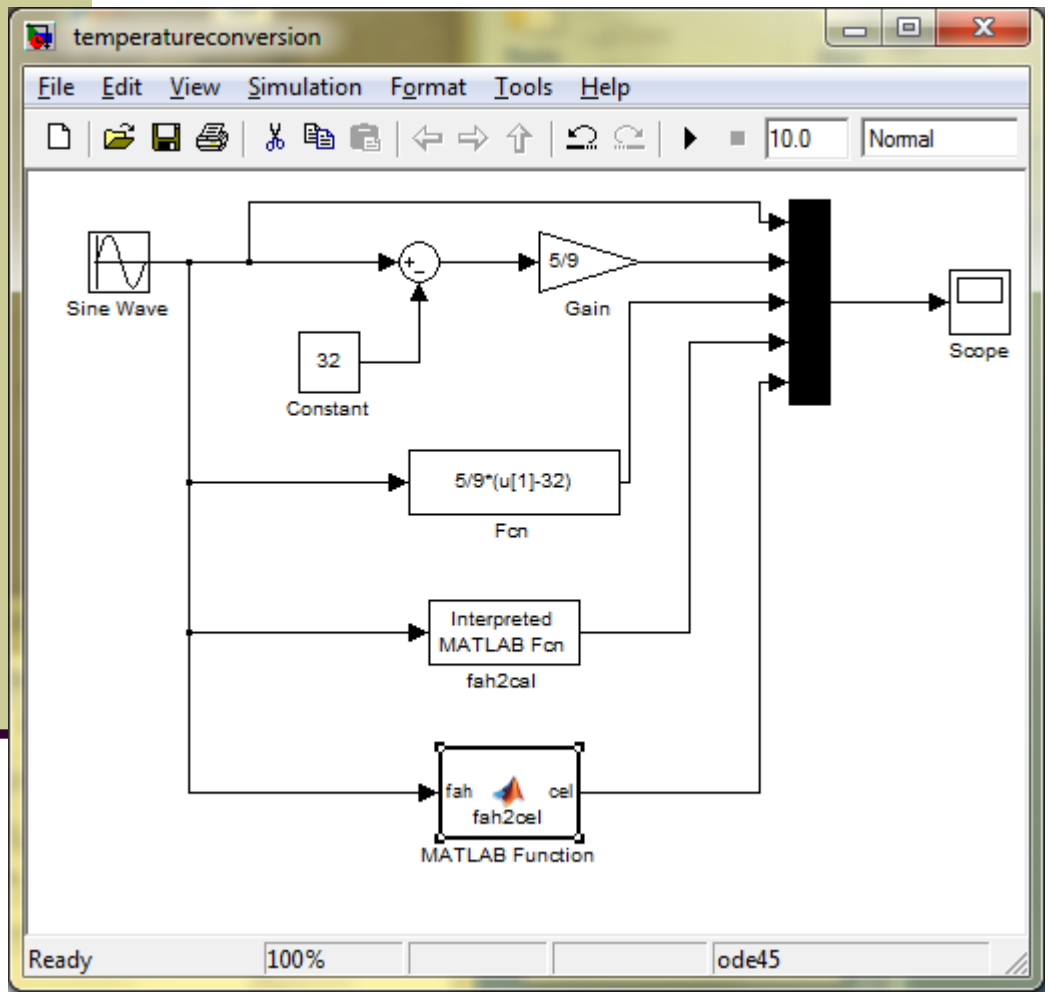
Example: Temperature Conversion

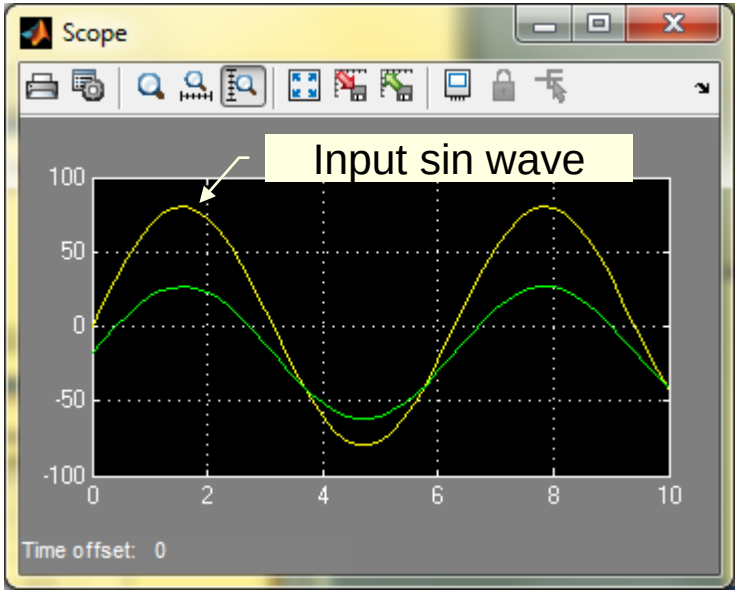
C = (5/9)*(F-32)



1) Determine what blocks are needed to convert the temperature in Simulink?
2) Using a sin wave as the input (set amplitude to 80 degF.), compare F vs C on the scope

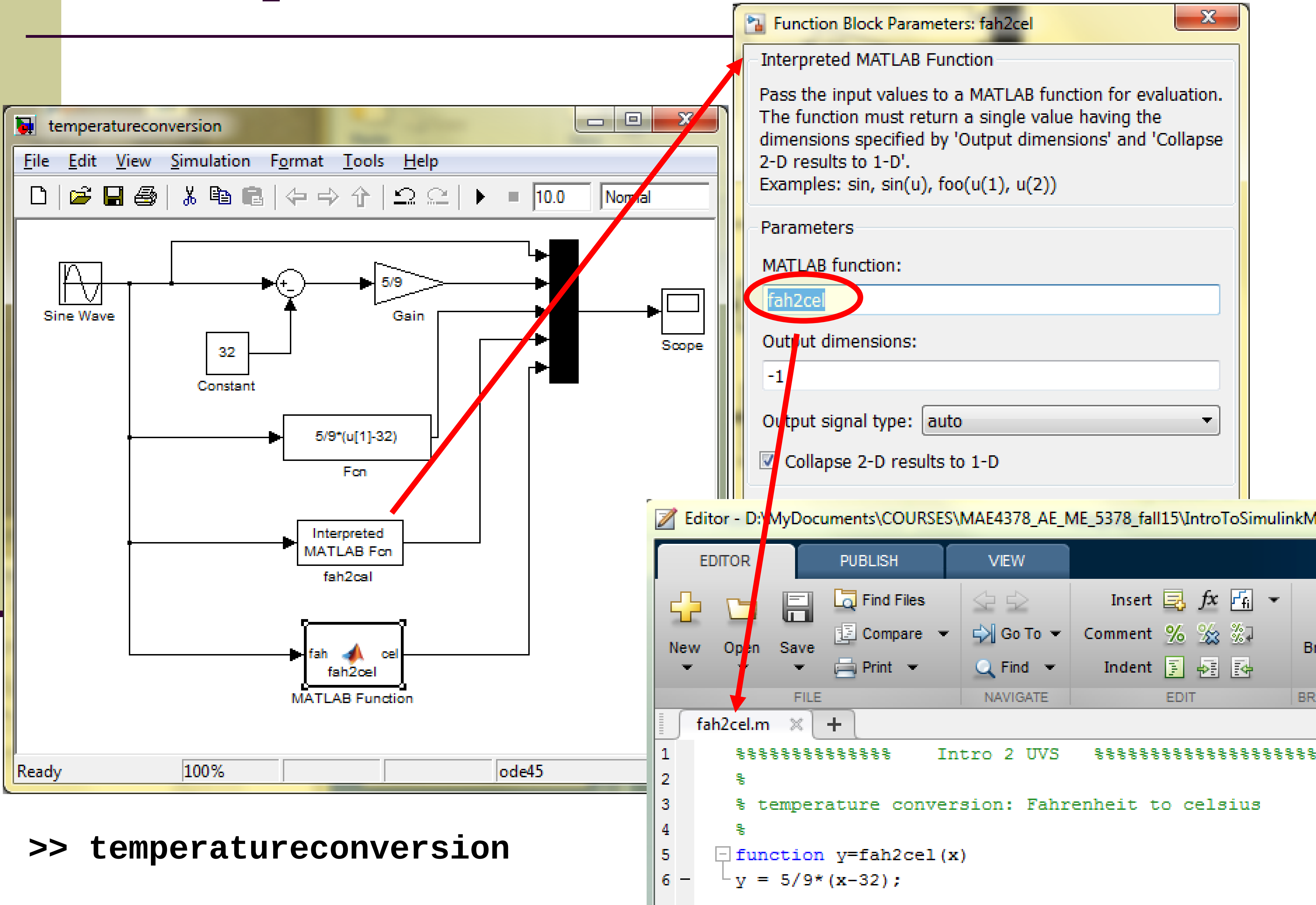
Temperature Conversion





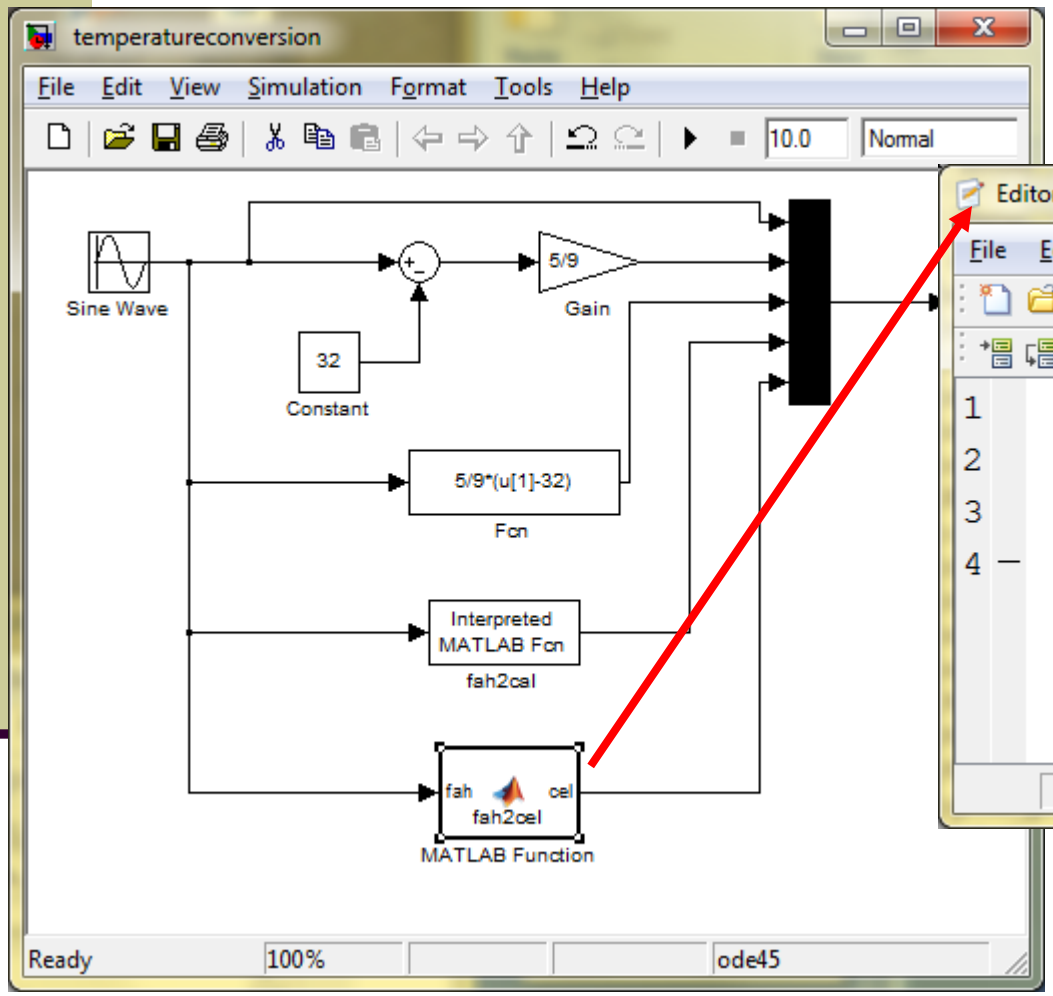

```
>> temperatureconversion
```

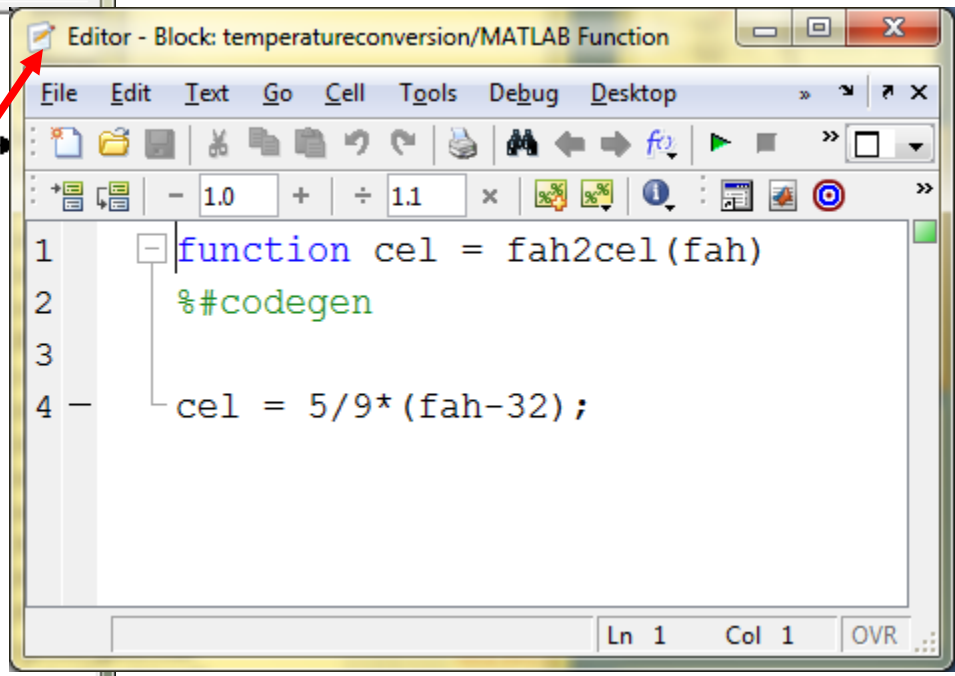
Temperature Conversion




```
>> temperatureconversion
```

Temperature Conversion






```
>> temperatureconversion
```

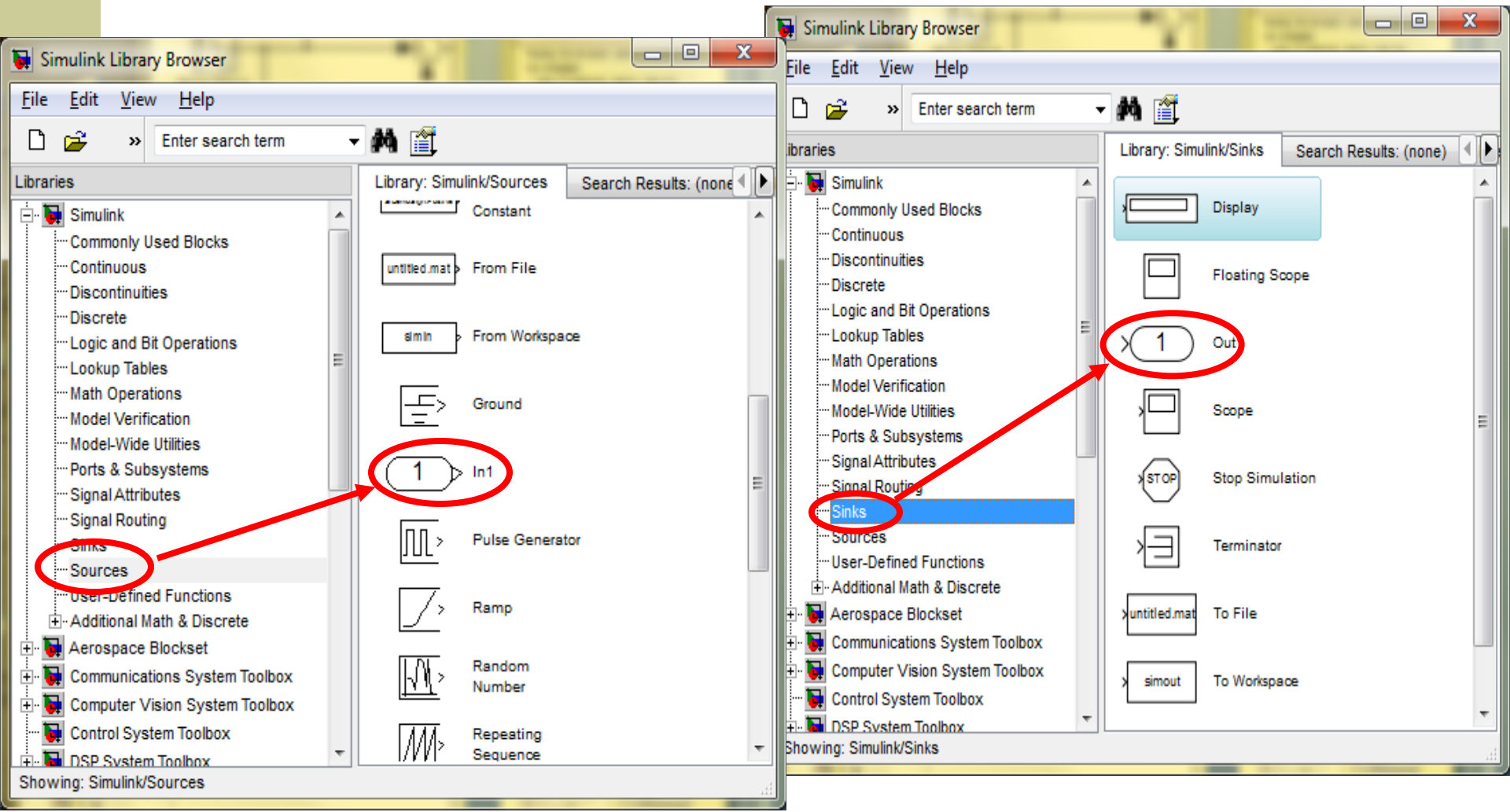
Getting data out of model

Pausing the simulation sends all stored data to the workspace

- Useful for analyzing results while the simulation is “running”
- Useful for storing information during different parts of the simulation

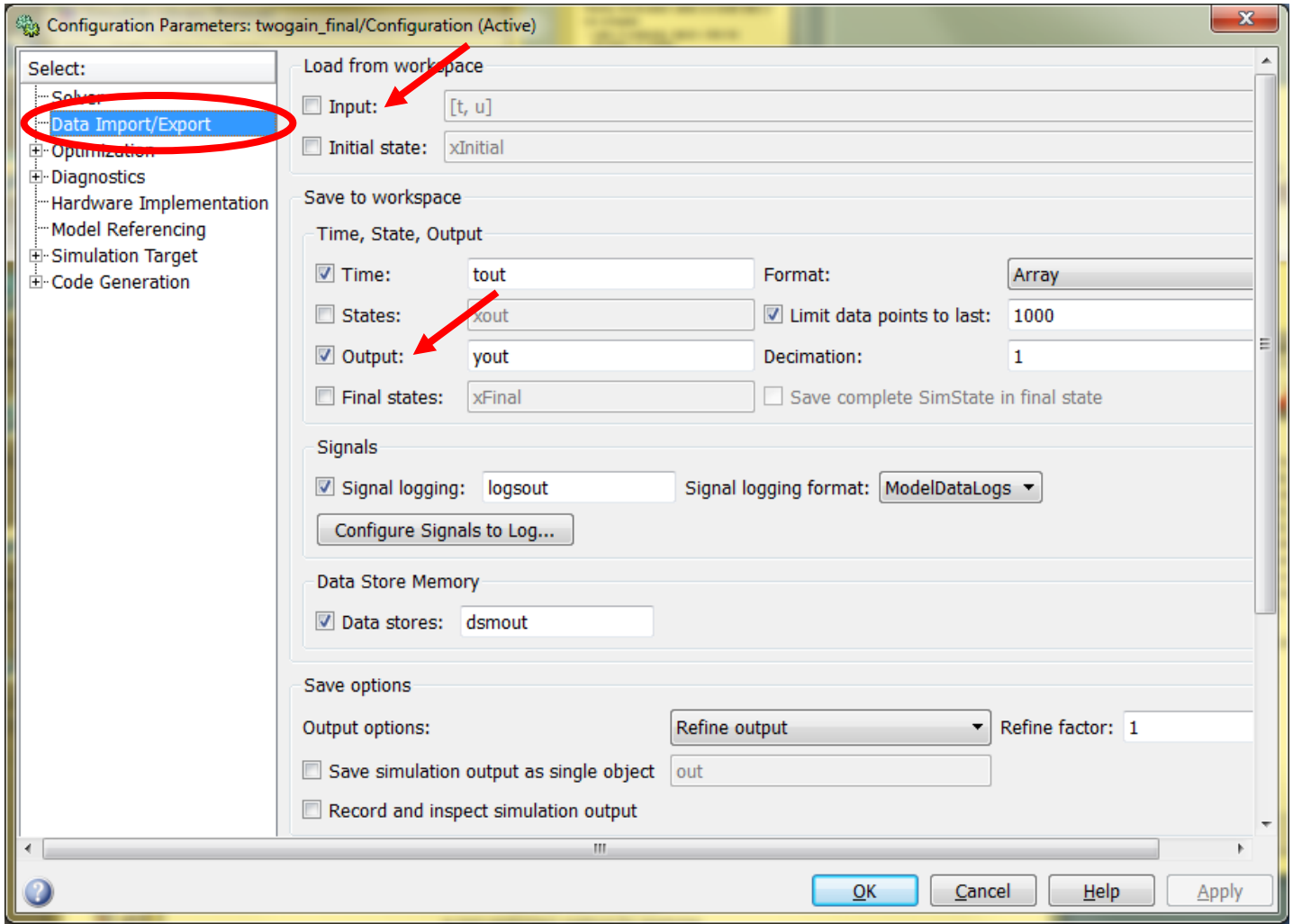
Getting data into and out of models

Applies to Inports and Outports on top level only

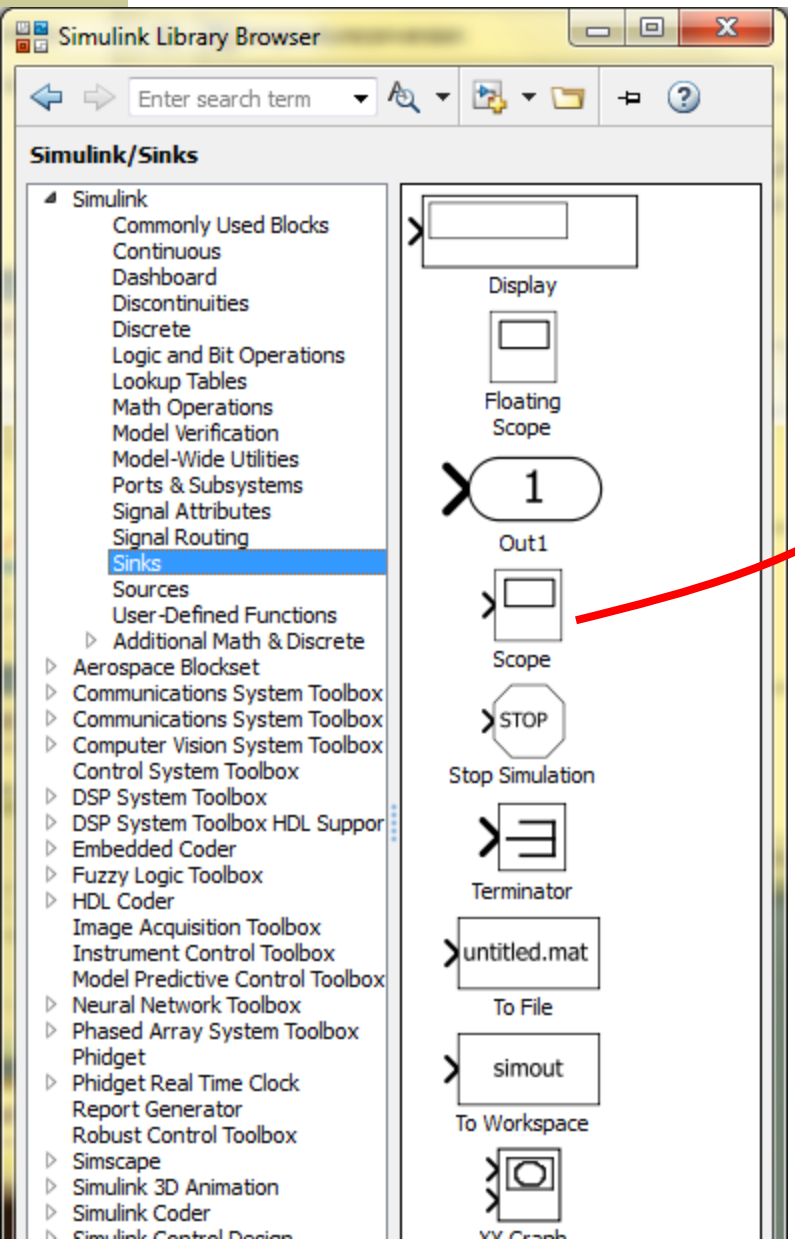


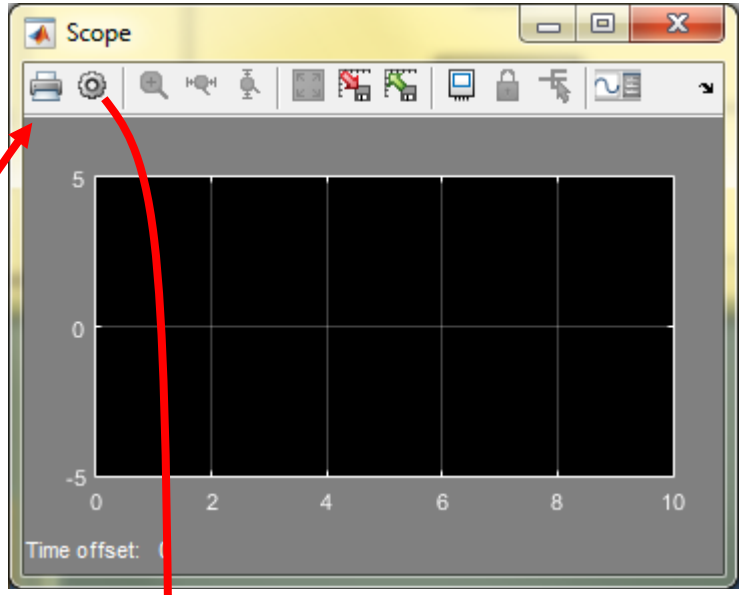
Getting data into and out of models

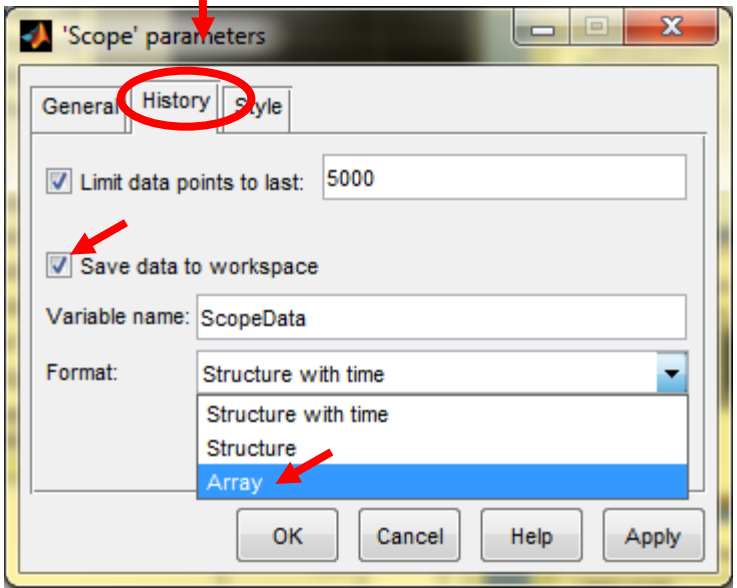
Applies to Inports and Outports on top level only



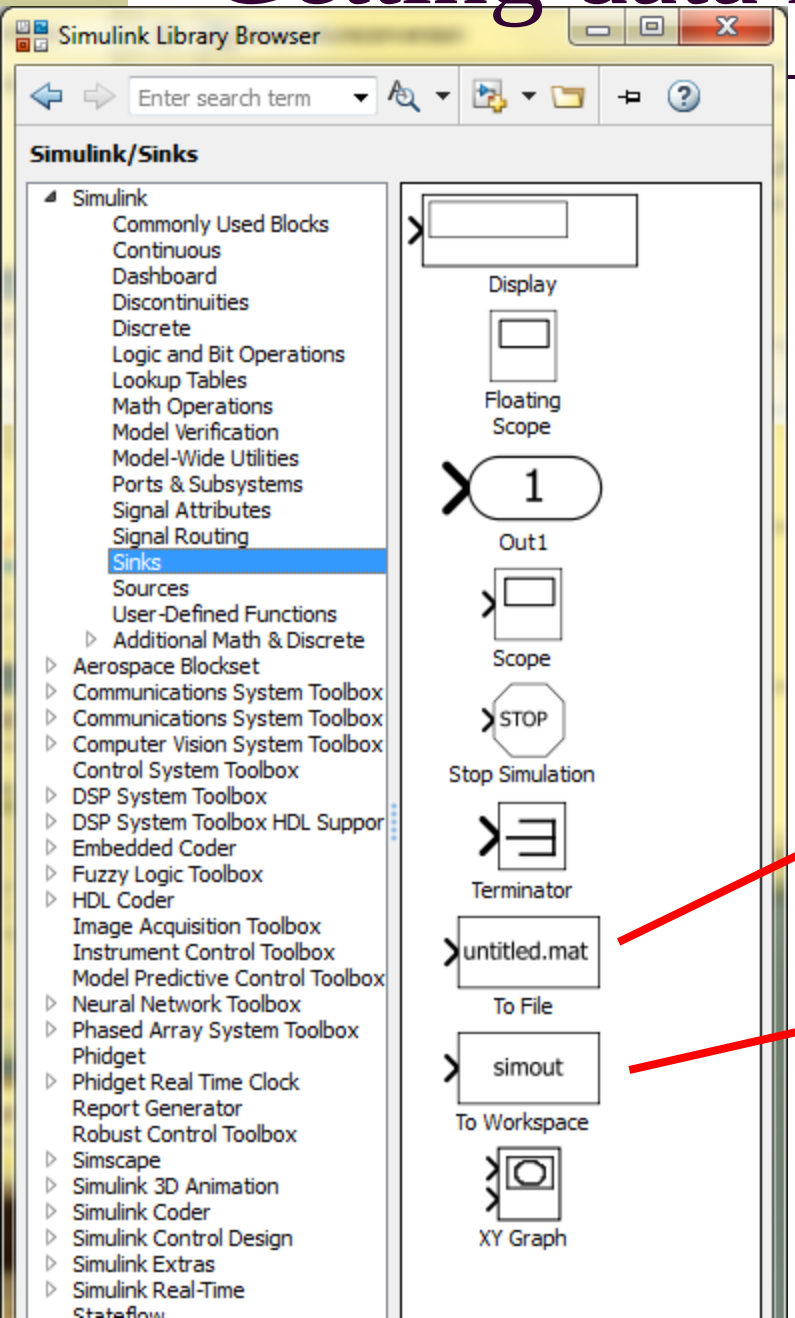
Getting data into and out of models







Getting data into and out of models



to write signal history to a file

to write signal history to Workspace