

EXPERIMENT-SIX

SIMULINK

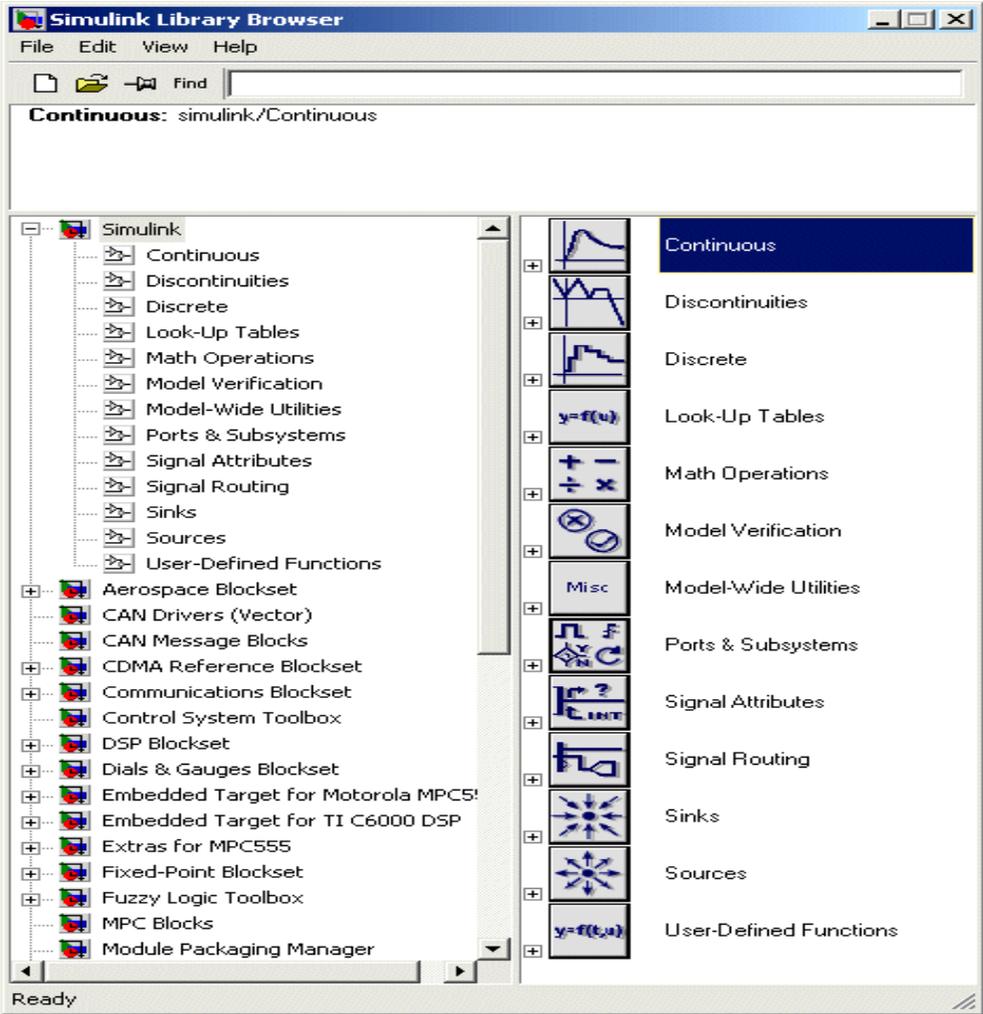
Simulink is a software package for modeling, simulating, and analyzing dynamic systems. It supports linear and nonlinear systems, modeled in continuous time, sampled time, or a hybrid of the two. Systems can also be multirate, i.e., have different parts that are sampled or updated at different rates. For modeling, Simulink provides a graphical user interface (GUI) for building models as block diagrams, using click-and-drag mouse operations. With this interface, you can draw the models just as you would with pencil and paper (or as most textbooks depict them). This is a far cry from previous simulation packages that require you to formulate differential equations and difference equations in a language or program. Simulink includes a comprehensive block library of sinks, sources, linear and nonlinear components, and connectors. You can also customize and create your own blocks.

1- Starting Simulink

To start Simulink, you must first start MATLAB. Consult your MATLAB documentation for more information. You can then start Simulink in two ways:

- Click the Simulink icon  on the MATLAB toolbar.
- Enter the `simulink` command at the MATLAB prompt.

EXPERIMENT-SIX: SIMULINK



The Library Browser displays a tree-structured view of the Simulink block libraries installed on your system. You can build models by copying blocks from the Library Browser into a model window. Some of the important library blocks and its purposes are illustrated in the following subsections. Also you can apply the other blocks depending on your requirements.

EXPERIMENT-SIX: SIMULINK

1.1- Sinks library:

The *Sinks* library contains blocks that display or write block output.

Block Name	Purpose
Display	Show the value of the input.
Outport	Create an output port for a subsystem or an external output.
Scope, Floating Scope	Display signals generated during a simulation.
Stop Simulation	Stop the simulation when the input is nonzero.
Terminator	Terminate an unconnected output port.
To File	Write data to a file.
To Workspace	Write data to a variable in the workspace.
XY Graph	Display an X-Y plot of signals using a MATLAB figure window.

1.2- Sources library:

The *Sources* library contains blocks that generate signals.

Block Name	Purpose
Band-Limited White Noise	Introduce white noise into a continuous system.
Chirp Signal	Generate a sine wave with increasing frequency.
Clock	Display and provide the simulation time.
Constant	Generate a constant value.
Digital Clock	Generate simulation time at the specified sampling interval.
From File	Read data from a file.
From Workspace	Read data from a variable defined in the workspace.
Ground	Ground an unconnected input port.
Inport	Create an input port for a subsystem or an external input.
Pulse Generator	Generate pulses at regular intervals.
Ramp	Generate a constantly increasing or decreasing signal.
Random Number	Generate normally distributed random numbers.
Repeating Sequence	Generate a repeatable arbitrary signal.
Signal Builder	Generate an arbitrary piecewise linear signal.
Signal Generator	Generate various waveforms.
Sine Wave	Generate a sine wave.
Step	Generate a step function.
Uniform Random Number	Generate uniformly distributed random numbers.

EXPERIMENT-SIX: SIMULINK

1.3- Continuous library:

The Continuous library contains blocks that model linear functions.

Block Name	Purpose
Derivative	Output the time derivative of the input.
Integrator	Integrate a signal.
State-Space	Implement a linear state-space system.
Transfer Fcn	Implement a linear transfer function.
Transport Delay	Delay the input by a given amount of time.
Variable Transport Delay	Delay the input by a variable amount of time.
Zero-Pole	Implement a transfer function specified in terms of poles and zeros.

1.4- Discrete library:

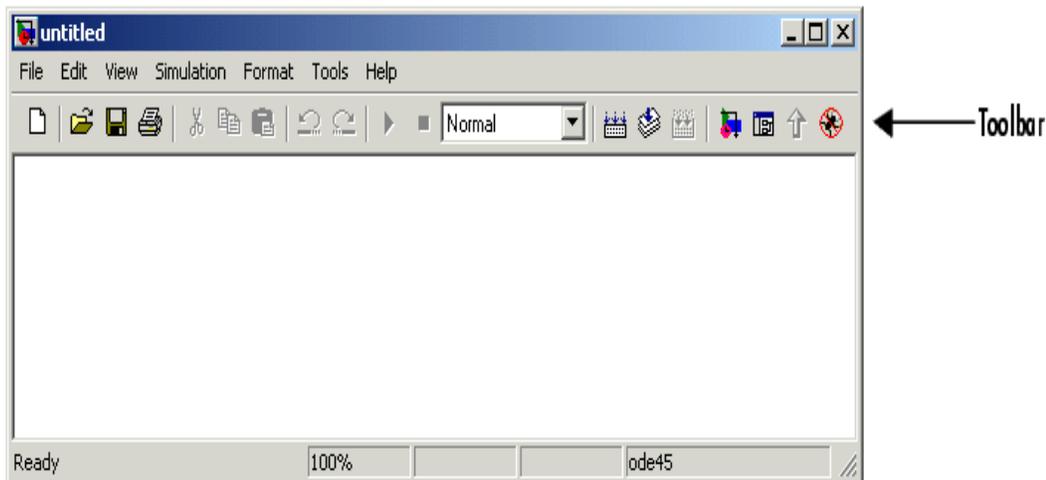
The *Discrete* library contains blocks that represent discrete-time functions.

Block Name	Purpose
Discrete Filter	Implement IIR and FIR filters.
Discrete State-Space	Implement a discrete state-space system.
Discrete Transfer Fcn	Implement a discrete transfer function.
Discrete Zero-Pole	Implement a discrete transfer function specified in terms of poles and zeros.
Discrete-Time Integrator	Perform discrete-time integration of a signal.
First-Order Hold	Implement a first-order sample-and-hold.
Memory	Output the block input from the previous time step.
Unit Delay	Delay a signal one sample period.
Zero-Order Hold	Implement zero-order hold of one sample period.

2 Creating a New Model

To create a new model, click the **New** button on the Library Browser's toolbar (Windows only) or choose **New** from the library window's **File** menu and select **Model**. You can move the window as you do other windows. The figure below shows the new model that you open it.

EXPERIMENT-SIX: SIMULINK



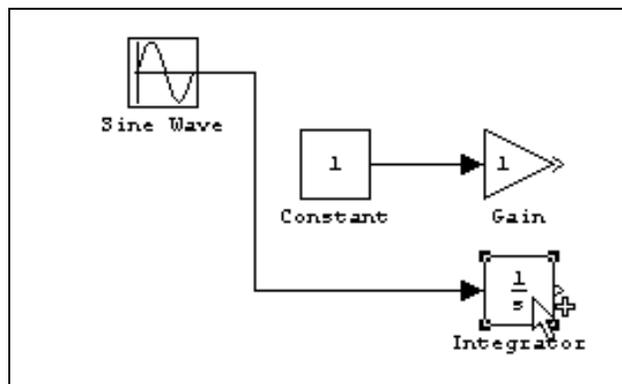
3- Connecting Blocks

Simulink block diagrams use lines to represent pathways for signals among blocks in a model. Simulink can connect blocks for you or you can connect the blocks yourself by drawing lines from their output ports to their input ports.

3.1- Connecting Two Blocks:

To auto connect two blocks:

1. Select the source block.
2. Hold down **Ctrl** and left-click the destination block.



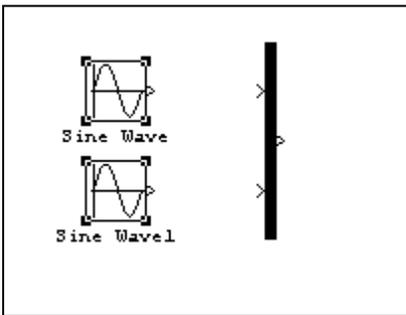
EXPERIMENT-SIX: SIMULINK

3.2- Connecting Groups of Blocks:

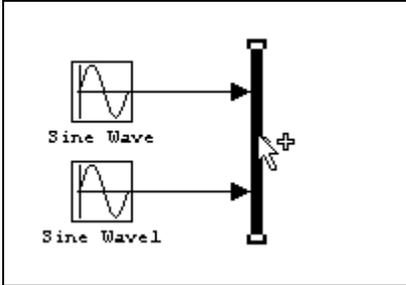
Simulink can connect a group of source blocks to a destination block or a source block to a group of destination blocks.

To connect a group of source blocks to a destination block:

1. Select the source blocks.



2. Hold down **Ctrl** and left-click the destination block.

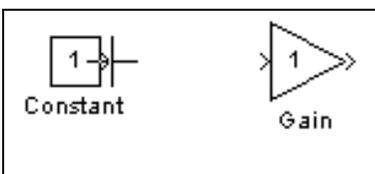


In the same way we can connect a source block to a group of destination blocks.

4- Drawing a Line Between Blocks

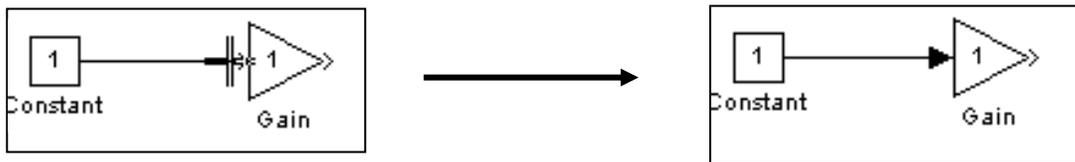
To connect the output port of one block to the input port of another block:

1. Position the cursor over the first block's output port. It is not necessary to position the cursor precisely on the port. The cursor shape changes to crosshairs.



EXPERIMENT-SIX: SIMULINK

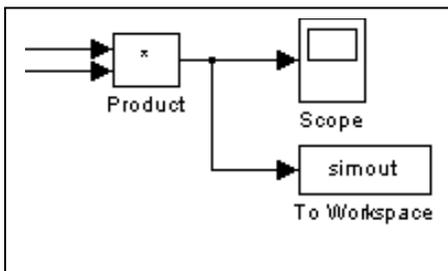
3. Press and hold down the mouse button.
4. Drag the pointer to the second block's input port. The cursor shape changes to double crosshairs. Release the mouse button. Simulink replaces the port symbols by a connecting line with an arrow showing the direction of the signal flow.



Simulink draws connecting lines using horizontal and vertical line segments. To draw a diagonal line, hold down the **Shift** key while drawing the line.

We can also draw a branch lines. A *branch line* is a line that starts from an existing line and carries its signal to the input port of a block. Both the existing line and the branch line carry the same signal. Using branch lines enables you to cause one signal to be carried to more than one block.

In this example, the output of the Product block goes to both the Scope block and the To Workspace block.



To add a branch line, follow these steps:

1. Position the pointer on the line where you want the branch line to start.
2. While holding down the **Ctrl** key, press and hold down the left mouse button.
3. Drag the pointer to the input port of the target block, then release the mouse button and the **Ctrl** key.

You can also use the right mouse button instead of holding down the left mouse button and the **Ctrl** key.

EXPERIMENT-SIX: SIMULINK

To draw a line segment, you draw a line that ends in an unoccupied area of the diagram. An arrow appears on the unconnected end of the line. To add another line segment, position the cursor over the end of the segment and draw another segment. Simulink draws the segments as horizontal and vertical lines. To draw diagonal line segments, hold down the **Shift** key while you draw the lines.

5- Selecting Objects

Many model building actions, such as copying a block or deleting a line, require that you first select one or more blocks and lines (objects).

5.1- Selecting One Object:

To select an object, click it. Small black square "handles" appear at the corners of a selected block and near the end points of a selected line. For example, the figure below shows a selected Sine Wave block and a selected line.

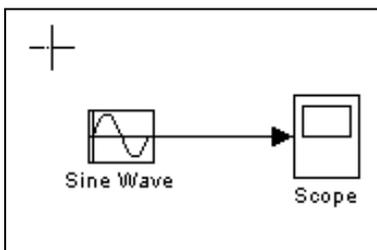


When you select an object by clicking it, any other selected objects are deselected.

5.2- Selecting Multiple Objects Using a Bounding Box:

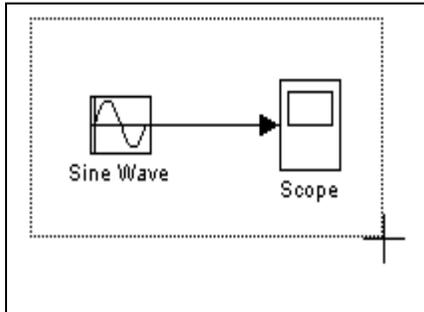
An easy way to select more than one object in the same area of the window is to draw a bounding box around the objects:

1. Define the starting corner of a bounding box by positioning the pointer at one corner of the box, then pressing and holding down the mouse button. Notice the shape of the cursor.

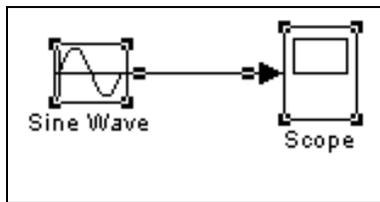


EXPERIMENT-SIX: SIMULINK

2. Drag the pointer to the opposite corner of the box. A dotted rectangle encloses the selected blocks and lines.



3. Release the mouse button. All blocks and lines at least partially enclosed by the bounding box are selected.



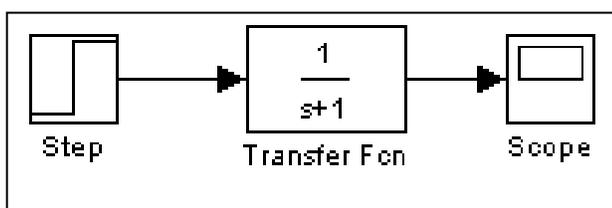
6- Viewing Output Trajectories

Output trajectories from Simulink can be plotted using one of three methods:

- Feed a signal into either a Scope or an XY Graph block.
- Write output to return variables and use MATLAB plotting commands.
- Write output to the workspace using To Workspace blocks and plot the results using MATLAB plotting commands.

6.1- Using the Scope Block:

You can use display output trajectories on a Scope block during a simulation. This simple model shows an example of the use of the Scope block.

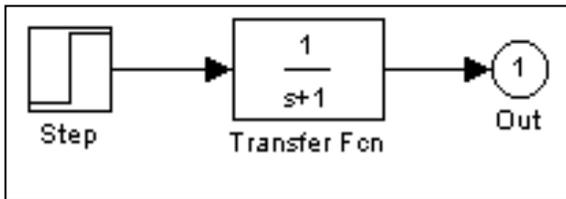


EXPERIMENT-SIX: SIMULINK

The display on the Scope shows the output trajectory. The Scope block enables you to zoom in on an area of interest or save the data to the workspace. The XY Graph block enables you to plot one signal against another.

6.2- Using Return Variables:

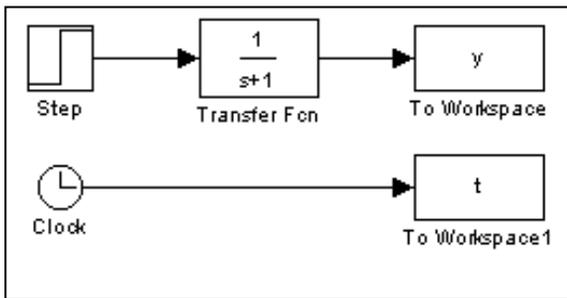
By returning time and output histories, you can use MATLAB plotting commands to display and annotate the output trajectories.



The block labeled Out is an Outport block from the Signals & Systems library.

6.3- Using the To Workspace Block:

The To Workspace block can be used to return output trajectories to the MATLAB workspace. The model below illustrates this use.



The variables *y* and *t* appear in the workspace when the simulation is complete. You store the time vector by feeding a Clock block into a To Workspace block. You can also acquire the time vector by entering a variable name for the time on the **Workspace I/O** pane of the **Simulation Parameters** dialog box, for menu-driven simulations. The To Workspace block can accept an array input, with each input element's trajectory stored in the resulting workspace variable.