

All of the following information was taken from the tutorials found in MATLAB on-ramp at https://www.mathworks.com/academia/student_center/tutorials/source/simulink/onramp/placeholder.html (The link is dead now but it's really for citation purposes)

Simulink On-Ramp Tutorial

Welcome to the Simulink On-Ramp tutorial. The purpose of this tutorial is to help you get started using Simulink effectively. In this tutorial you will learn the essentials of Simulink and how you can use it to construct and simulate a simple system. We will explore how Simulink works in conjunction with MATLAB and how to work with Simulink block diagrams.

What is Simulink?

If you are not familiar with Simulink let's first answer the following question:

What is Simulink?

Simulink is a graphical design tool that builds on the MATLAB environment. It is a platform for multi-domain simulation and model-based design of dynamic systems.

You can use Simulink to graphically design, simulate and test time varying systems, like a mass spring damper system or a bouncing ball. You can also develop controllers, compensators, and control algorithms and other dynamic systems such as filters and sensors using block diagrams.

Simulink also provides a convenient simulation environment for analysis, refinement, and validation of systems designed.

Engineers and Scientists across the world use Simulink to solve real world applications in many industries including Automotive, Industrial Automation and Machinery, Communications, Aerospace and Defense, Electronics, Semiconductors, Process Industries, and even areas like Medical Devices.

Simulink Tutorial Introduction

On our website you can find many demonstrations that show how Simulink is used in different applications. In this demo you can see how Simulink is used to design an automatic landing system for a space crew return vehicle known as the HL20 lifting body. The purpose of this vehicle is that in case of an emergency in the international space station, crew members of the space station could get inside the vehicle and return safely to earth. The demonstration covers the application of an automotive powertrain control system. Simulink is used to develop a

complete vehicle model of the engine, transmission and drivetrain of a car and implement the control system for the automatic transmission interacting with the rest of the powertrain.

This demonstration explores how Simulink can be used for video motion stabilization. Simulink is used to remove all of the unwanted movement from the video leaving a very fixed image. You can use the links provided in the Appendix to launch any one of these demonstrations and learn more about how Simulink is used.

Simulink Tutorial Examples

In this tutorial we will focus on the fundamentals of the Simulink environment and we will show you how to construct Simulink block diagrams of time varying equations, and run a simulation to observe and analyze the results.

You can also reference the tutorial “Using Simulink to Model Continuous Dynamical Systems” for more details on how to construct models that represent physical systems such as this mass spring damper system.

To learn how to design signal processing systems like this Noise Cancellation Filter, you can watch the tutorial “Using Simulink to Model Discrete Dynamical Systems”.

All of the tutorial example files have been placed in a ZIP file and you can download them from either the Appendix link in the top right of this player or the “Tutorial examples files” link under the Tutorial Appendix section of the Tutorial Launch Pad. Once downloaded, unzip the file and follow the directions of the “readme.txt” file which is one of the unzipped files.

We hope you enjoy the tutorial and find it helpful for your learning. Please forward us any feedback that you have on the tutorial, especially on how it can be improved.

Constructing and Running a Simple Model

In this section you will learn how to start Simulink and find library blocks that allow you to construct your own block diagrams within Simulink. You will also learn how to construct a simple model based on an equation, as well as how to run a simulation.

Starting Simulink

You start Simulink from within MATLAB. To construct a Simulink model you must first launch the Simulink Library Browser from your MATLAB session. To do so, you can select the Simulink icon from the MATLAB Desktop or type `>>simulink<<` at the MATLAB command prompt.

The Simulink Library Browser

Simulink consists of standard block libraries you can use to construct your block diagrams. Each one of the libraries on the Simulink Library Browser contains different kinds of blocks you can use to construct your systems.

The following table describes each of these libraries

- **Commonly Used Blocks**
 - Contains a group of the most commonly used blocks, such as the Constant, Import, Outport, Scope, and Sum blocks. Each of the blocks in this library is also included in other libraries.
- **Continuous**
 - Contains blocks that model dynamic functions, such as the Transfer Function block, and the Integrator block.
- **Discontinuities**
 - Contains blocks with outputs that are discontinuous functions of their inputs, such as the Saturation block.
- **Discrete**
 - Contains blocks that represent discrete time functions, such as the Unit Delay Block.
- **Logic and Bit Operations**
 - Contains blocks that perform logic or bit operations, such as the Logical Operator and Relational Operator blocks.
- **Lookup Tables**
 - Contains blocks that use lookup tables to determine their outputs from their inputs, such as the Cosine and Sine blocks.
- **Math Operations**
 - Contains blocks that perform mathematical and logical functions, such as the Gain, Product, and Sum blocks.
- **Model Verification**
 - Contains blocks that enable you to create self-validating models, such as the Check Input Resolution block.
- **Model-Wide Utilities**
 - Contains blocks that provide information about the model, such as the Model Info block.
- **Ports & Subsystems**
 - Contains blocks that allow you to create subsystems, such as the Inport, Outport, and Subsystems blocks.

- Signal Attributes
 - Contains blocks that modify the attributes of signals such as the Data Type Conversion block.
- Signal Routing
 - Contains blocks that route signals from one point in a block diagram to another, such as the Mux and Switch blocks.
- Sinks
 - Contains blocks that display or export output, such as the Outport and Scope blocks.
- Sources
 - Contains blocks that generate or import system inputs, such as the Constant, Inport, and Sine Wave blocks.
- User-Defined Functions
 - Contains blocks that allow you to define custom functions, such as the MATLAB Function block.
- Additional Math & Discrete
 - Contains two additional libraries for mathematical and discrete function blocks.

Try it, Launching Simulink

Now it's your turn to practice. Open the Simulink Library Browser and locate the Add block.

You can do one of the following:

- Select the Simulink icon from the MATLAB Desktop
- Type `>>simulink<<` at the MATLAB command prompt.

Now try to locate the Add block. Click the appropriate library where the Add block is located.

Hint: The Add block allows you to mathematically add signals of your block diagram.

Creating a new model

We will now show you how to create a new model in Simulink. Since most physical systems can be represented through mathematical equations, we will show you the steps for building a Simulink model based on equations, taking as an example a very simple algebraic equation. The procedure is the same for more complex equations.

The equation we will be modeling is $y = 5 * \sin(t)$

The model described in this chapter takes a sine wave and multiplies it by 5 and displays the results.

Before you can begin building your model, start Simulink. Then, from the Simulink Library Browser Select **File>New>Model** which opens a new empty model window.

Adding blocks to your model

To construct a model, you first copy blocks from the Simulink Library Browser into the model window.

To create the simple model you need three blocks:

1. A Sine Wave block - to generate a Sine Wave input
2. A Gain block - to multiply by 5
3. A Signal Scope block - to visualize the signals

To add blocks to your model:

- Select the Sources Library in the Simulink Library Browser. Select the Sine Wave block, then drag it into the model window.
- Select the Math Operations Library in the Simulink Library Browser. Select the Gain block and then drag it into the model window.
- Select the Sinks library in the Simulink Library Browser. Select the Scope block and drag it into the model window.

Before you connect the blocks in your model, you should arrange them logically to make the signal connections as straightforward as possible. To move a block within the model window, you can either drag the block or select the block and press the arrow keys on the keyboard.

Connecting Blocks

After you add blocks to the model window, you must connect them to represent the signal connections within the model. Notice that each block has angle brackets on one or both sides.

These angle brackets represent the input and output ports.

- The > symbol pointing into a block is an input port
- The > symbol pointing out of a block is an output port

You connect your blocks in your model by drawing lines between output ports and input ports. To draw a line between two blocks position the mouse pointer over the output port on the right side of the Sine Wave block. Note that the pointer changes to a crosshairs (+) shape while over the port.

Drag a line from the output port to the input port of the Gain block. Note that the link is dashed while you hold the mouse button down. Release the mouse button over the input port. Simulink connects the blocks with an arrow that indicates the direction of signal flow.

Drag a line from the output port of the Gain block to the input port on the Scope block. Simulink connects the blocks.

You can automatically connect two blocks by

1. Clicking the source block
2. Press and hold the **ctrl** key
3. Click the destination block.

Defining Block Parameters

All simulink blocks have attributes that you can change. To view and modify the block parameters, double click on the block. The Block Parameters dialog box opens.

You can find a block description at the top of the Block Parameters dialog box and then the attributes that can be modified. Let's change the value of the gain to 5 and click the **OK** button. The value of the gain has now been changed to 5.

Labeling blocks and signals

Block and signal labels can enhance the readability of your model. In Simulink, every block must have a unique name. To rename a block, click the block label and edit. If you type in a duplicate name, a warning message pops up and the label is automatically changed to resolve the conflict.

To label a signal, double click the signal line and enter the label in the active text box.

Q&A Modeling Algebraic Equations

Match each one of the algebraic equations to the correct Simulink block diagram.

Match each one of the algebraic equations to the correct Simulink block diagram

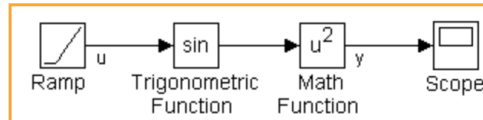
$y = \sin(u^2)$

$y = \sin^2(u)$

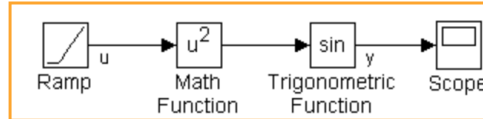
$y = \frac{u}{|u|}$

$y = \frac{|u|}{u}$

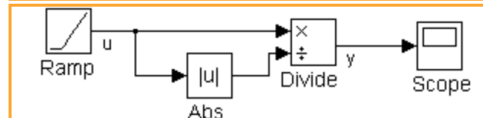
A)



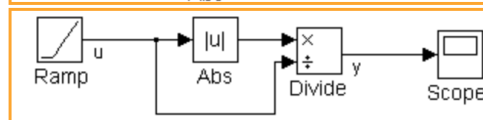
B)



C)



D)



Running a Simulation

After you complete the model block diagram, you can simulate and visualize the results. By default, the simulation runs from a start time of 0 to a stop time of 10. The simulation stop time appears at the top of the model. You can change the stop time from the model screen.

To simulate a model perform one of the following

- Press the **Run** button
- Select **Simulation>Run** from the top of the model window

To visualize the results, double click the Scope block in the model window. The Scope window displays the simulation results. You'll see that the simulation ran from 0 to 10.

Try it, Building a model and running a simulation

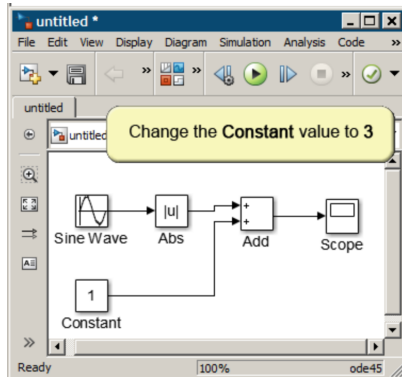
Now it's your turn to try building a model. I'll help you along the way.

We'll construct the following algebraic equation as a Simulink model and run a simulation.

$$y = |\sin(t)| + 3$$

1. Create a new model
2. Add a Sine Wave
3. Add a constant block
4. Add absolute value block (Math)
5. Add Add block (Math)

6. Add a Scope block (Sinks)
7. Connect the blocks appropriately



8. Change the constant to 3
9. Run the simulation
10. Visualize the result

Changing Simulink Parameters

Simulink interacts with MATLAB in several ways. You can, for example, use MATLAB variables to define any parameter in your Simulink block diagram. Instead of using a value of 5 on the gain block we can define the gain parameter with a MATLAB variable. Simply open the gain block by double clicking on it and set the Gain parameter to A.

Defining Simulink Variables with MATLAB Parameters

To use MATLAB variable as Simulink Parameters, you must define the variables in the MATLAB workspace. Let's define a variable called A in the MATLAB workspace and run the simulation.

The variable names that you can use as Simulink Parameters must be identical to the variables defined in the MATLAB workspace. They are case sensitive and they need to follow the naming rules for MATLAB variables.

Using MATLAB variable names as Simulink block parameters produces the same results.

Q&A Using MATLAB Variables

To use MATLAB variables as Simulink block parameters, where do you need to define their values?

- ☐ A) Simulink Library Browser
- ☐ B) MATLAB Current Directory
- ☐ C) MATLAB Workspace
- ☐ D) Simulink Model Editor

Saving the Model

After you complete the model, you can save it for future use. To save the model select **File>Save** in the model window. You can specify the location where you want to save the model. Enter `>>simpleEquation<<` in the File name field and click **Save**.

Simulink saves the model with the filename `>>simpleEquation.slx<<`. You can open the model again by double clicking on the model name within the Current Directory Browser in MATLAB or by typing its name without the SLX extension in the command window.

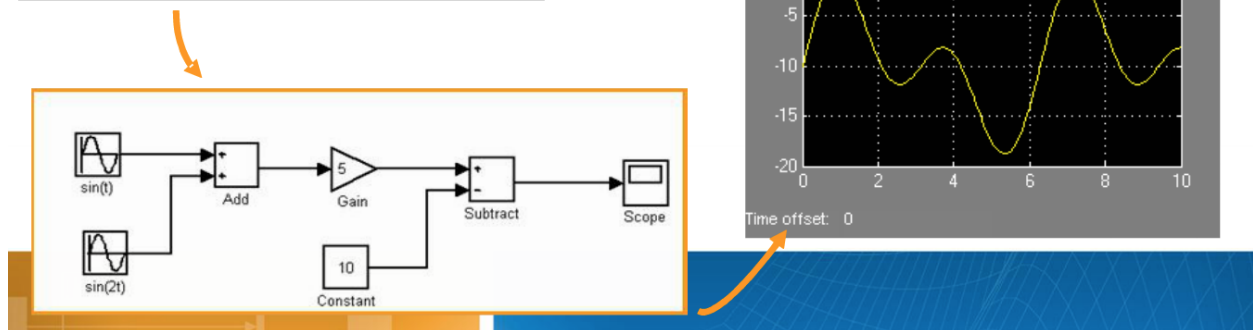
Summary

In this section we have discussed how to launch the Simulink Library Browser, construct a basic Simulink model, and run a simulation.

Simulating a Model

The function we will be modeling is: $y = 5 * (\sin(t) + \sin(2t)) - 10$

$$y = 5 * (\sin(t) + \sin(2t)) - 10$$



Now That we have seen how we can model a system represented by an algebraic equation using Simulink, let's explore how the simulation takes place. In this section we'll see how Simulink propagates signals in time and how changing the parameters of the simulation such as the Stop Time and the Time Step can affect the results of the simulation. We will also discuss how to obtain help when using Simulink.

Simulink propagates signals in time

Time is an inherent component of a Simulink block diagram. Simulink propagates signals in time by calculating a numerical approximation of the outputs of each block in time.

By default, Simulink starts the simulation at time 0 and stops at a time of 10, taking points in between the Start and Stop time to compute the results. This algebraic equation can be represented with the following Simulink diagram. There's only a few more things we need to change in the model.

The frequency of the second Sine Wave needs to be changed to 2 rad/sec and the values of the gain and constant need to match the equation. We can label the blocks appropriately, run the simulation and visualize the results.

The signal cannot be seen with the current axes scale, but we can select the Autoscale icon on the Scope Block to autoscale the axes and display the complete signal. Notice the simulation ran from 0 to 10.

Q&A Simulink Simulations

Now that you understand how Simulink propagates signals, answer the following question.

Time is built into Simulink simulations. True or False?

Changing the Stop Time

If you want the simulation to run for a longer or shorter period of time, you can change the Stop Time from the Model Window. Let's change the Stop Time to 50, run the simulation and visualize the results. Notice that the results look significantly different. Instead of having a curve, we now see a graph with jagged edges.

Why do the results look so different with a longer stop time? You have to be aware that the parameters that you select for running the simulation can significantly affect your results. In this case changing the Stop Time affected the resolution of the simulation results.

Simulation Step Size

Let's explore... Why does changing the stop time affect the results of the simulation? The parameters you set for your simulation can affect the resolution of simulation results.

Simulink calculates the result of the Simulation by stepping the model in time at a rate that is referred to as the Step Size. By default, the model is configured to have an automatic maximum step size which, for simple models, is computed as

$$h = \frac{\text{stop time} - \text{start time}}{50}$$

For the default stop time of 10, the step size is 0.2, so Simulink calculates a value for the output of each block at every 0.2 time units until it reaches the Stop Time of 10.

If the time range is long, the automatic calculation of the step size results in very large steps. With a Stop Time of 50, the Max Step Size is now 1, and Simulink calculates a point at every time step of 1.

If we zoom into the results of the scope block we can see that the calculations are only taken at a time step of 1 and that's why we see the jagged edges. In this case, it might be necessary to force a smaller maximum step size to ensure good resolution in the output.

Changing the Step Size

You can modify all of the parameters that determine how Simulink runs the simulation, including the Maximum Step Size. You can access the Step Size Parameter along with other parameters through the Configuration Parameters Menu. Select **Simulation>Model>Configuration Parameters**.

The model configuration is a set of parameters that specify how the simulation is run, select **Solver**. Under Solver you will find different settings that you can modify including the Maximum Step Size. Instead of auto you can define your own step size for the simulation. Let's force a Max Step Size of 0.1, click on **OK**, and run the simulation again. By forcing a smaller Step Size we have a better resolution even with a Stop Time of 50.

Try it, Changing the Stop Time and Max Step Size

Now it's your turn to practice. First change the stop time to 50 and run the simulation. Open the Configuration Parameters and change the Max Step Size to 0.1.

Solver Options and Getting Help

In the Model **Configuration Parameters** you can modify many factors that affect the simulation result, such as the Start and Stop Time and the Max Step Size.

Notice that there are many parameters that you can modify such as the Solver that is used. You can select a **Fixed Step** or **Variable Step** solver and select a discrete or one of the continuous solvers, which are different methods to calculate a numerical approximation for some of the block's outputs.

If you'd like to understand how each of these options affects the simulation you can right-click any of the menu items of the Configuration Parameters window, like the Max Step Size, select the "What's this?" and you will see an explanation of the specific menu item you have selected. In this case you will see how the Maximum Step Size is calculated.

For a more general overview of the options of the Configuration Parameters Menu you can also use the Help button. For example, if you want to understand the options of the **Solver Pane** in the Configuration Parameters window, click on the Help button, which will provide a brief explanation of the Solver options. If you need more information you can launch the Simulink Documentation by scrolling down and using the link for the Solver Pane.

On the documentation, you'll first see the Solver Menu just as it appears in the Configuration Parameters. As you scroll down you'll see a list of the options of the menu with hyperlinks to the explanations for each one of these items. If you click on Solver you'll see a detailed explanation of the solvers, differences between them and how they affect the simulation.

Block Help

You can also obtain information about each one of the blocks of the Simulink Library Browser through the Simulink documentation. If you want to know more about a specific block, you can open it from the Simulink Library Browser, or from a model that contains the block, and click on

the Help in the Block Parameters dialog box. This launches the documentation for the specific block.

You'll first see the Library in which the block is located and a brief description of the block. You'll also find the parameters dialog box for the block along with an explanation of these parameters.

Simulink Documentation

For General Information on how to use Simulink, you can launch the Documentation from the Simulink Library Browser. Select the Help Menu and click on the option Simulink Help. You can now explore the Simulink documentation by topics.

You can also explore many examples by selecting the Examples tab. There You can browse through the various Simulink application examples. For each example, you can see the model and the results of the simulation in the documentation and open the actual example Simulink model.

Summary

In this section you have learned how Simulink propagates signals in time and how to change parameters of the simulation. You have also learned different ways of obtaining help through the Simulink documentation.

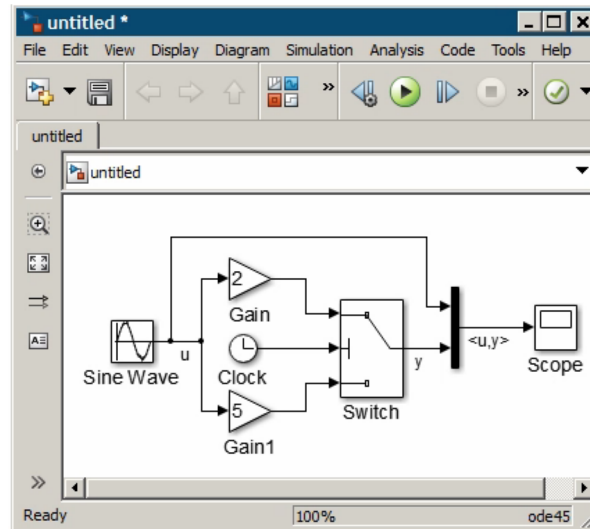
Working with MATLAB

Simulink is a modeling environment that runs on top of MATLAB's computational engine. You can benefit by using both environments together as they are both powerful and have unique characteristics. In this section we'll show you how Simulink can interact with MATLAB.

We'll first discuss how you can send inputs from MATLAB into a Simulink model, and then we'll talk about how to send outputs from a Simulink model to MATLAB.

Constructing a Logical Equation

$$y = \begin{cases} 2*u(t) & \text{if } t \geq 5 \\ 5*u(t) & \text{if } t < 5 \end{cases}$$



In this model we have constructed a logical equation in Simulink by using a switch block. The switch block can be used to enter an “if” condition into a diagram. It has 3 input ports and one output port.

The output is equal to the first input if the criteria determined by the second input is met, and the output is equal to the third input if the criteria determined by the second input is not met. In this example the criteria that the Switch Block will be using is the time of the simulation, a clock has been connected to the second input port of the switch to feed the time as a control signal. The passing criteria has been entered as \geq than a threshold value. The threshold value has been set to 5 which translates to the following.

- If the time is greater than or equal to 5, the output y is equal to the first signal on the switch block (two times the input u)
- If the criteria is not met, the output y is equal to the third signal on the switch block (five times the input u)
- A Mux block is then used to combine both the input u and the output y as one signal so we can see it on the same scope.

If we run the simulation we can see the input sine wave and the output which is a sign wave with a different amplitude depending on the time.

Using Inputs From MATLAB

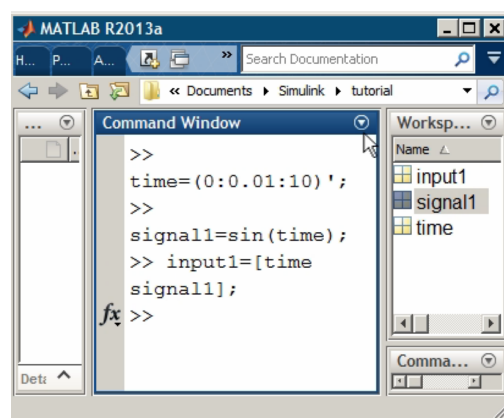
Simulink allows you to import data from the MATLAB workspace as inputs to the model. This allows you to send any type of signal into Simulink and import actual physical data into your model.

There are many blocks you can use to send inputs from MATLAB, let's delete the Sine Wave block being used as an input to the model and replace it with a From Workspace block from the sources library. If we open the Parameters Dialog box for the block we can enter the MATLAB variable name that the block will be reading data from. We can enter any valid MATLAB variable name, for example `>>input1<<`, which we will create next. The block shows the variable that Simulink will be reading data from.

The From Workspace block reads data from the MATLAB workspace. The block's Data parameter specifies the MATLAB variable that will be loaded into your Simulink model. In this case, `>>input1<<`. Since Simulink is a time based simulation, the signal sent from the MATLAB workspace must be defined in time. The variable `>>input1<<` must have at least two columns of information. The first column needs to have the time data and the second column needs to have the associated signal values.

Let's define a time base for the input signal and create a column vector `>>time<<` in the MATLAB workspace. Make sure it is defined as a column vector.

We can now create the input signal. Let's create the same Sine Wave signal we had originally and create variable `>>signal1<<`. Make sure again that you have defined a column vector. Combine both signals in one variable `>>input1<<`.



Notice that `>>input1<<` has two columns, the first column is the time data and the second column is the associated signal values.

`input1 =`

time	signal1
0	0
0.1	-1
0.2	1
0.3	2
0.4	-3
0.5	4
.	.
.	.
.	.

Running a Simulation with Inputs from MATLAB

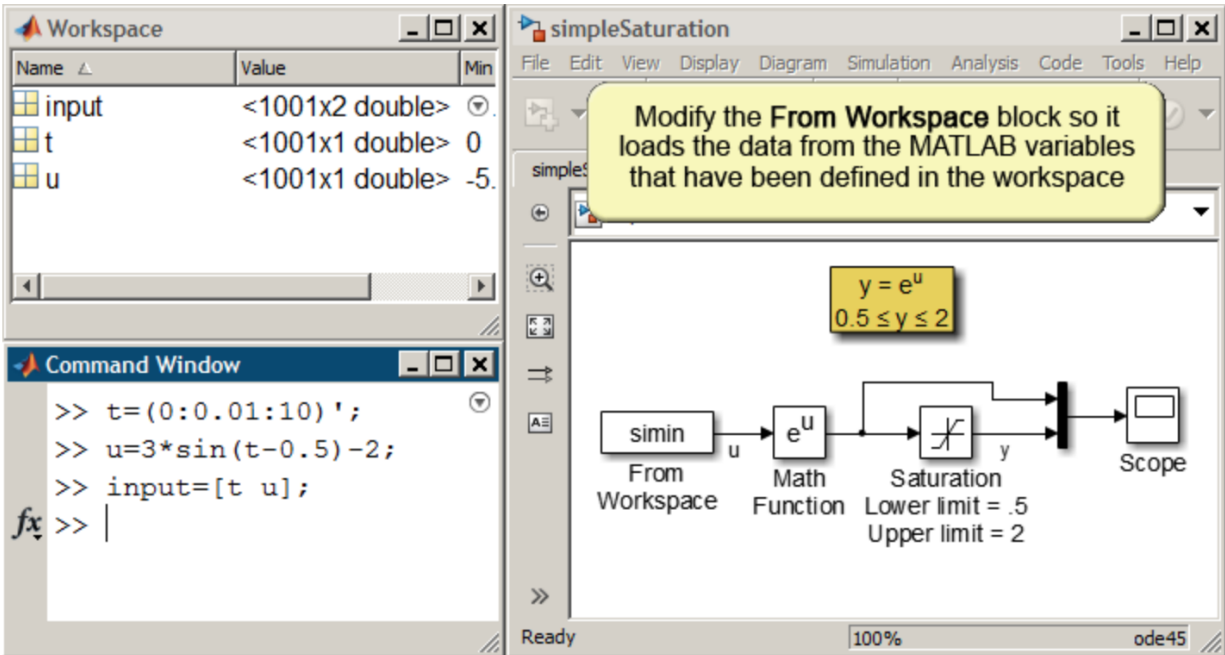
Now that the variable `>>input1<<` has been defined in the workspace with two columns of data, we are ready to run the simulation. The results are the same as when we had a Sine Wave block as an input to the model, but now we are driving system simulation with the input data defined in the MATLAB Workspace.

Initializing Inputs

Now that we have a **From Workspace** block in the model, we can quickly change the signal being sent as an input to the model. We can easily define a new signal in the MATLAB workspace and run the simulation. Defining inputs in MATLAB not only allows you to send any type of signal into Simulink, but it also allows you to quickly initialize or change the data used by Simulink.

Try it, Complete the following Model

Now it's your turn to practice. The model already contains a From Workspace block. Let me define some variable `sin` in the workspace. Now, change the From Workspace block so that it loads the data from the MATLAB variables that have been defined in the workspace and run the simulation.



1. Modify the From Workspace block so it loads the correct data from MATLAB variables
2. Run the Simulation
3. Visualize the results

Sending Outputs to MATLAB

Once you have completed a model, you may want to export your simulation results to MATLAB for further data analysis or visualization. The Outport block sends the signal to the MATLAB workspace as a variable.

Let's add an Outport block from the Sinks library into the model, we can branch off a signal by right-clicking and holding on the signal line and dragging the branch to the other block.

Open the Configuration Parameters Dialog Box, select the Data Import/Export pane on the left and change the output variable name to `>>Output<<`.

Run the simulation and now we have a variable called `>>Output<<` in the MATLAB workspace. The variable contains two columns of information because there are two signals passing through the Mux block that connects to the Outport block. The first column of values refers to the signal `>>u<<` and the second column refers to the second signal `>>y<<`.

Simulink automatically records the simulation time step of the simulation into a variable called `>>tout<<`. You can now create a plot of the output in MATLAB.

Q&A Working with MATLAB

Now that you know how to use the From Workspace and Outport blocks, answer the following question.

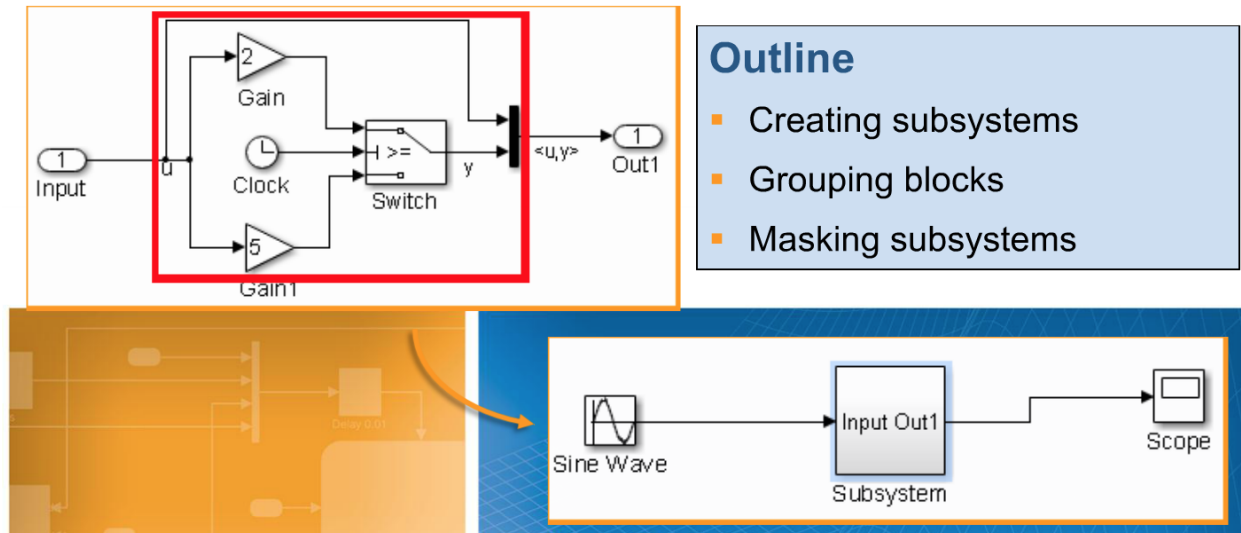
What are the advantages of importing inputs from MATLAB and exporting outputs to the MATLAB Workspace? (Select all that apply)

- ☐ A) Simulation results can be visualized with a wide variety of MATLAB plotting functions
- ☐ B) You can import actual physical data into your model
- ☐ C) Simulation results can be analyzed further in MATLAB
- ☐ D) You can drag blocks from the Simulink Library Browser into the MATLAB Workspace

Summary

In this section, you have learned how to interact with the MATLAB workspace. You now know how to send inputs from the MATLAB workspace into a Simulink model, and how to send signals from the model to the MATLAB workspace.

Creating Subsystems



Using subsystems helps reduce the number of blocks displayed in your model window. You can create a subsystem in two ways.

- By grouping existing blocks together
- Using the Subsystem block.

If your model already contains the blocks you want to convert to a subsystem, you can create the subsystem by grouping those blocks.

1. Drag a box around the blocks and connecting lines that you want to include in the subsystem.
2. When you release the mouse button, the blocks and all connecting lines are selected.
3. Select the **Diagram** menu and choose **Subsystem & Model Reference>Create Subsystem from Selection**.

Simulink replaces the selected blocks with a Subsystem block. You can double click on the subsystem to open its contents. You can also change any of the blocks inside the subsystem.

Subsystems Inputs and Outputs

The input and output ports inside the subsystem act as gateways to send signals across the borders of the subsystem. If you change the block names of the input or output ports inside the subsystem it automatically changes the names of the ports in the subsystem block at the top level of the model as well.

There are a number of ways to navigate back to the top level of the model. Click the “Show Model Browser” button at the bottom left corner of the model. The Model Browser enables the ability to navigate between the different levels within a model.

Other navigation options include the up-arrow icon and the model icons displayed in the Explorer Bar. Let's click on the SimpleSubsystem icon in the Explorer Bar to go back to the top level of the model. Notice the subsystem's input port label changed from In1 to Input. Let's navigate back into the subsystem via the Model Browser and hide the browser for maximum display.

You can manually add or delete input ports and output ports to have more inputs or outputs in the subsystem. You can find these blocks in the ports and subsystems library. Let's delete the Mux block and send signal `>>u<<` as an output of the subsystem by adding an additional output port. We can make a copy of the existing output port that is already inside the subsystem by right-clicking and dragging. Now there are two outputs of the subsystem. We can connect the second output to another Scope block.

Using the Subsystem Block

You can also create a subsystem by adding a Subsystem block to your model from the Ports & Subsystems library. Open the subsystem block by double clicking it. By default the subsystem has one input port and one output port by which you can add any blocks to the subsystem window the same as you were adding to a model. Use Inport blocks to represent inputs from outside the subsystem and Outport blocks to represent outputs from the subsystem. Let's complete the model and run the simulation.

Q&A Creating a Subsystem

Let's review how you can create a subsystem.

What are possible ways to create a subsystem?
(Select all that apply)

- ☐ A) Select blocks in a model and group them into a subsystem
- ☐ B) Select blocks in a model and type `subsystem` in the Command Window
- ☐ C) Drag a Subsystem block into a model, and add blocks to the subsystem window
- ☐ D) Drag variables from the MATLAB workspace as a group into a Simulink model

Masked Subsystem

A mask is a custom user interface for a subsystem that can be treated as a separate block with its own icon and parameter dialog box.

By masking a subsystem, you are encapsulating a block diagram to have its own parameter dialog with its own block description, parameter prompts, and help text. You can make an independent custom block that you can reuse as a unique block like those defined in Simulink.

Creating a Masked Subsystem

To create a mask, select the subsystem and click the **Diagram** menu, select **Mask>Create Mask**, or simply right-click on the subsystem and select the option **Mask>Create Mask**.

The mask icon replaces a subsystem's standard icon. In other words, it appears in a block diagram in place of the standard icon for a subsystem block. Simulink uses MATLAB code that you can supply to draw the custom icon. You can use a subset of MATLAB drawing commands in the **Icon Tab**, such as `IMAGE`, `DISP`, and `PLOT`.

As an icon, let's try `>>image(imread('clock.jpg'))<<`. A custom icon is displayed.

You can also define a set of parameters so the user can enter the values in the block dialog box. In this version of the logical subsystem the gain values have been specified as MATLAB workspace variables `a` and `b`. You can prompt the user to enter the values of `a` and `b`.

Now if you open the subsystem you can only see its mask. The block parameters dialog box is prompting the user to enter the values of `a` and `b`. To edit the mask, select the subsystem and click on the **Diagram** menu, select **Mask>Edit Mask**.

The **Documentation** pane enables you to define or modify the type, description, and help text for a masked block. The mask type is a block classification. The block description is informative text that appears in the block's dialog box under the mask type. If you are designing a system for others to use, this is a good place to describe the block's purpose or function.

You can customize every feature of the mask dialog box, including which parameters appear on the dialog box, the order in which they appear, the parameter prompts, the controls used to edit the parameters and the parameter callbacks.

For more information about Masked Subsystems, you can look up the Simulink documentation for masks or click on the link provided in the Appendix.

Once a mask is created, we cannot see the contents of the subsystems simply by double clicking on the block. To look under the mask, click on the down arrow icon located in the bottom left corner of the subsystem block.

Summary

In this section you have learned how to use subsystems to reduce the number of blocks displayed in your model window. You have also learned how to create your own custom blocks by masking a subsystem.