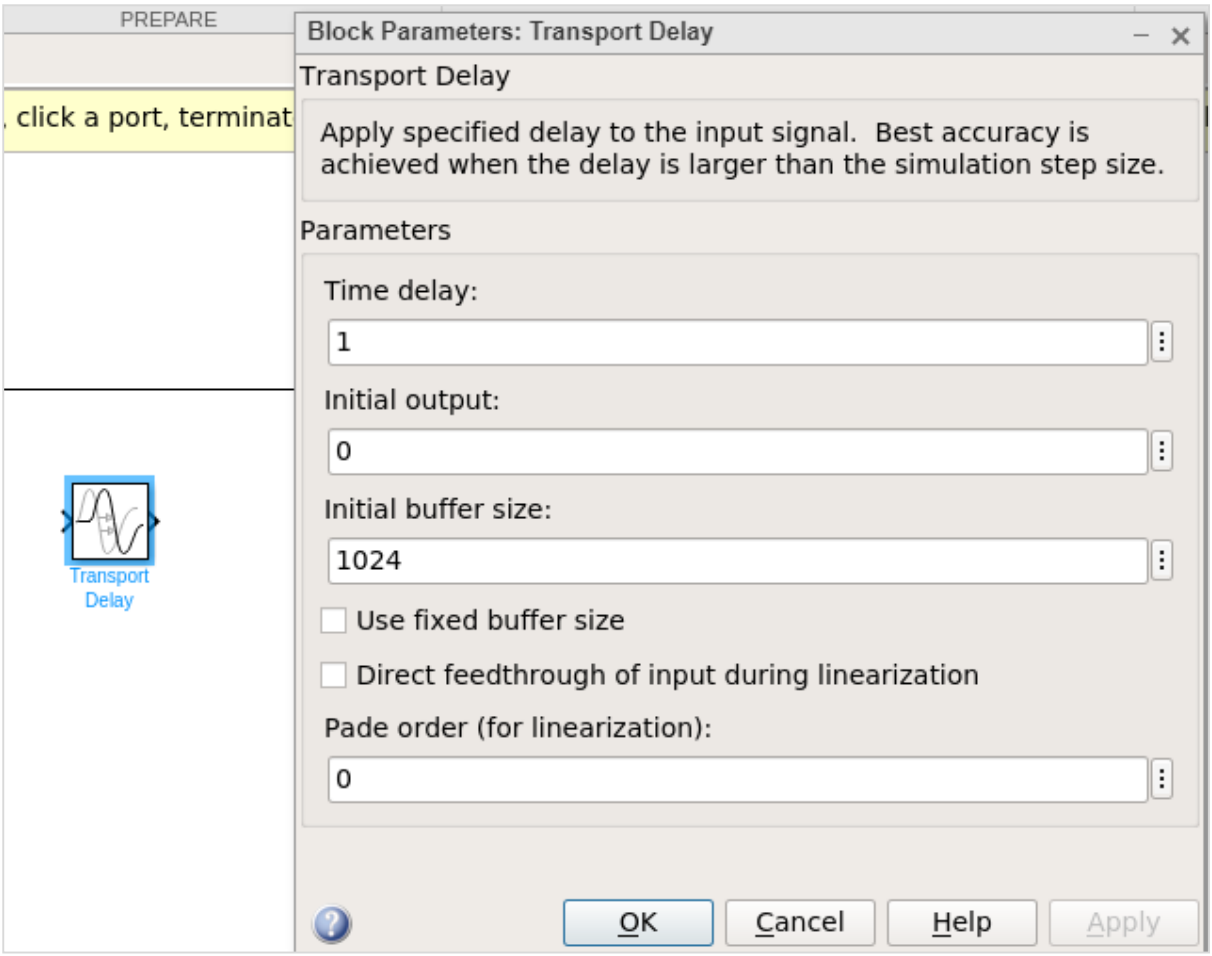
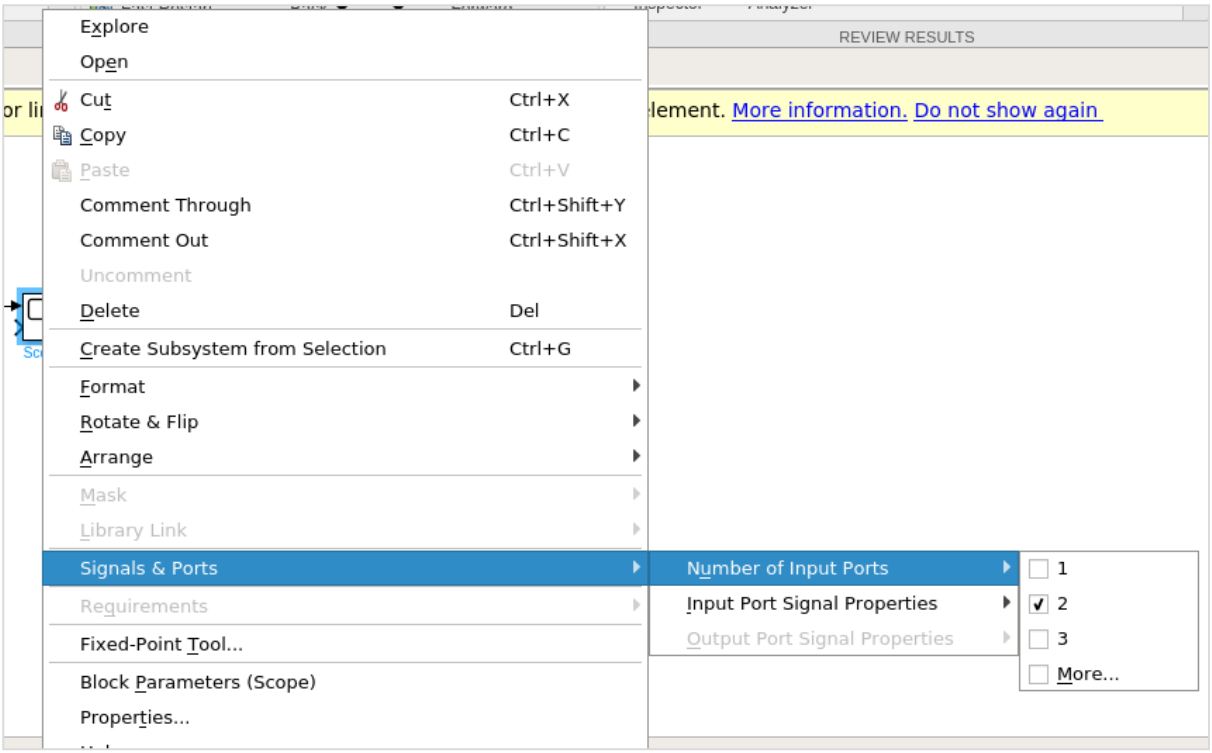
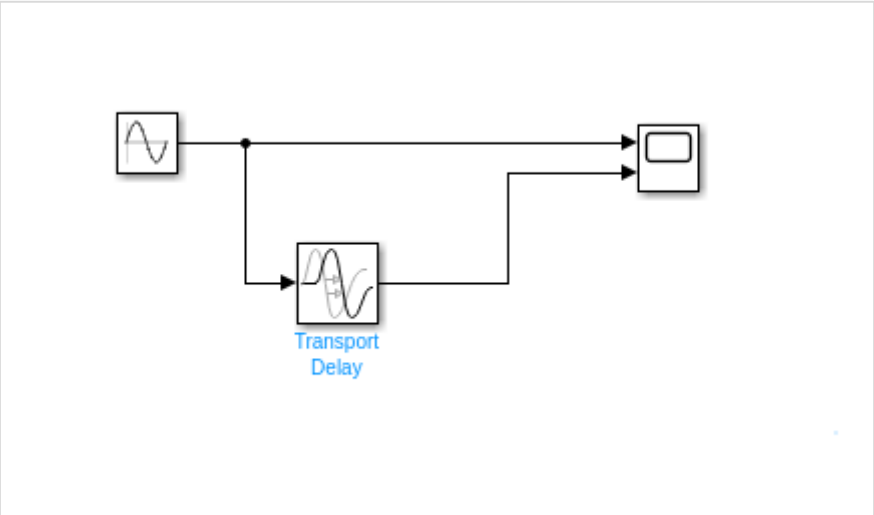


Let us change the time delay from 1 to 3. Make the changes and click on OK button.

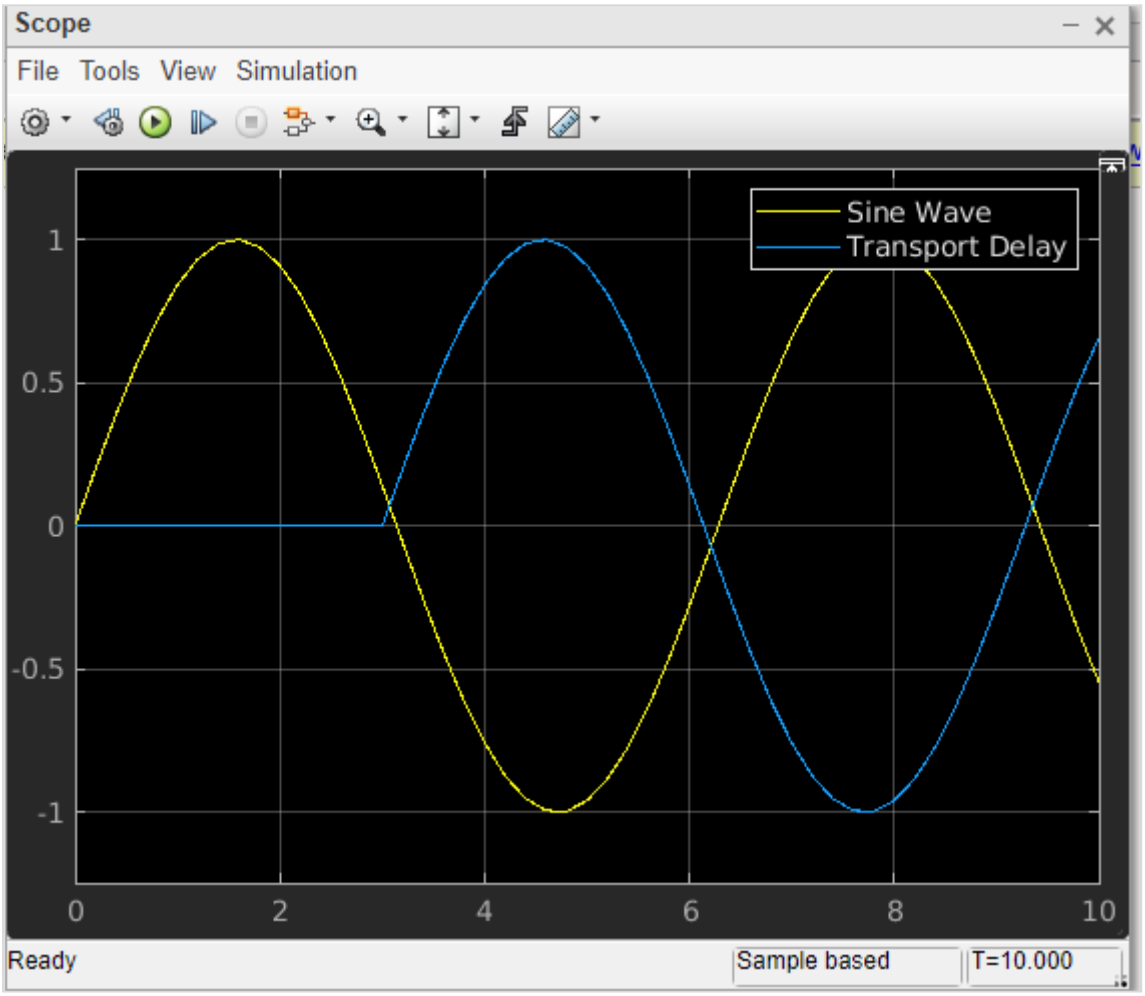
Now add one more input port to scope block. Right click on scope block and select the signals and ports. Select 2 for number of input ports as shown below:



Now connect the transport delay to sine wave and to scope as shown below:



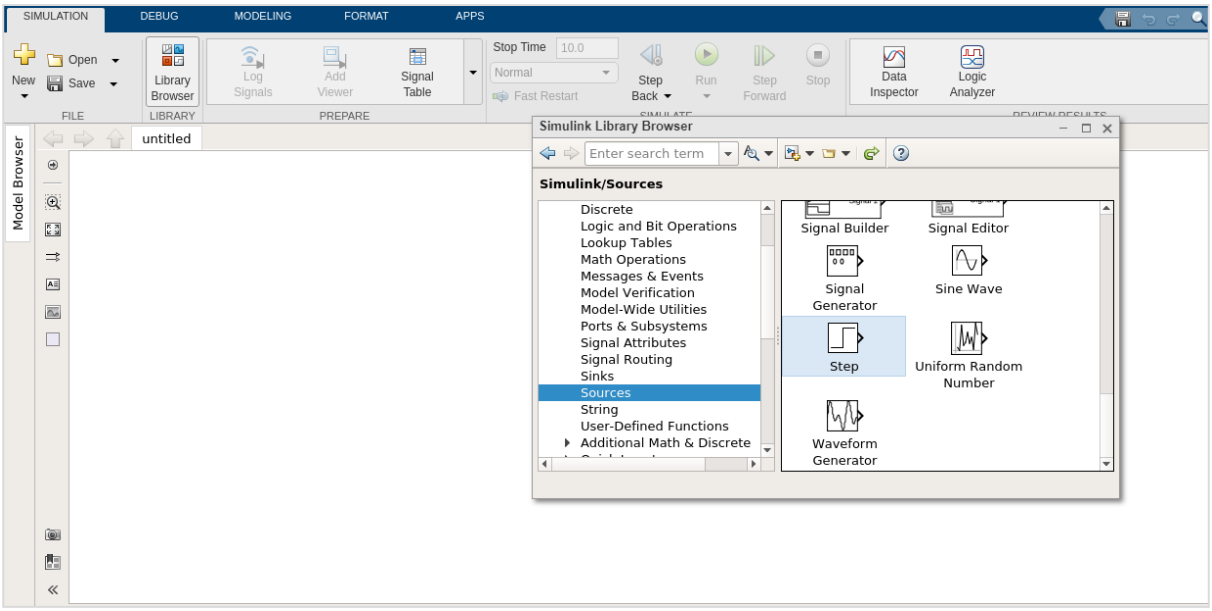
Now run the simulation to see a delay of 3 seconds to the sign wave. Right click scope block and select block parameters to see the display.



9. MATLAB Simulink — Mathematical Library

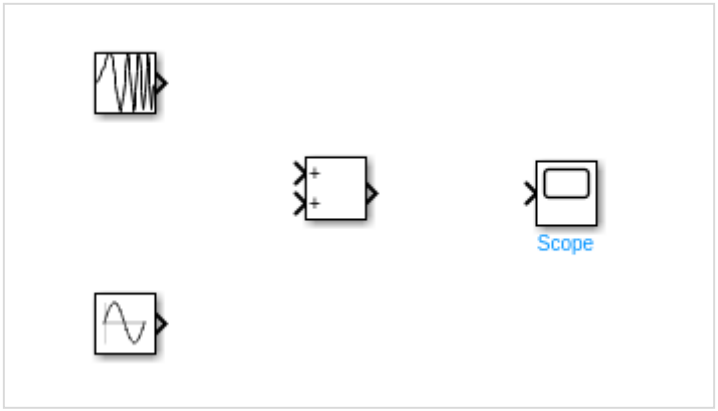
In this chapter, we will learn how to sum the two given signals and get the output.

Select the blank model and open Simulink library browser as shown below:

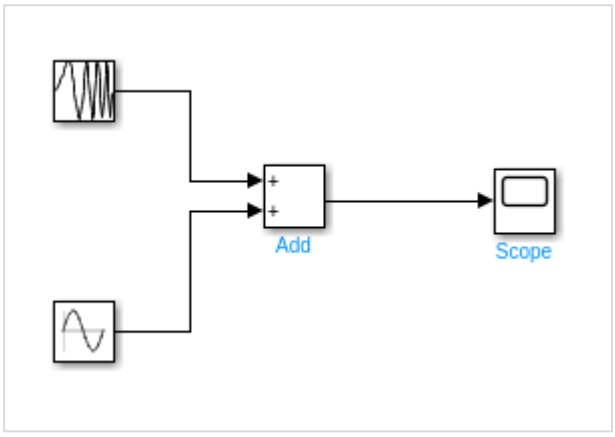


We are going to combine chirp signal and sine wave blocks by using add block from Math operation and see the final display.

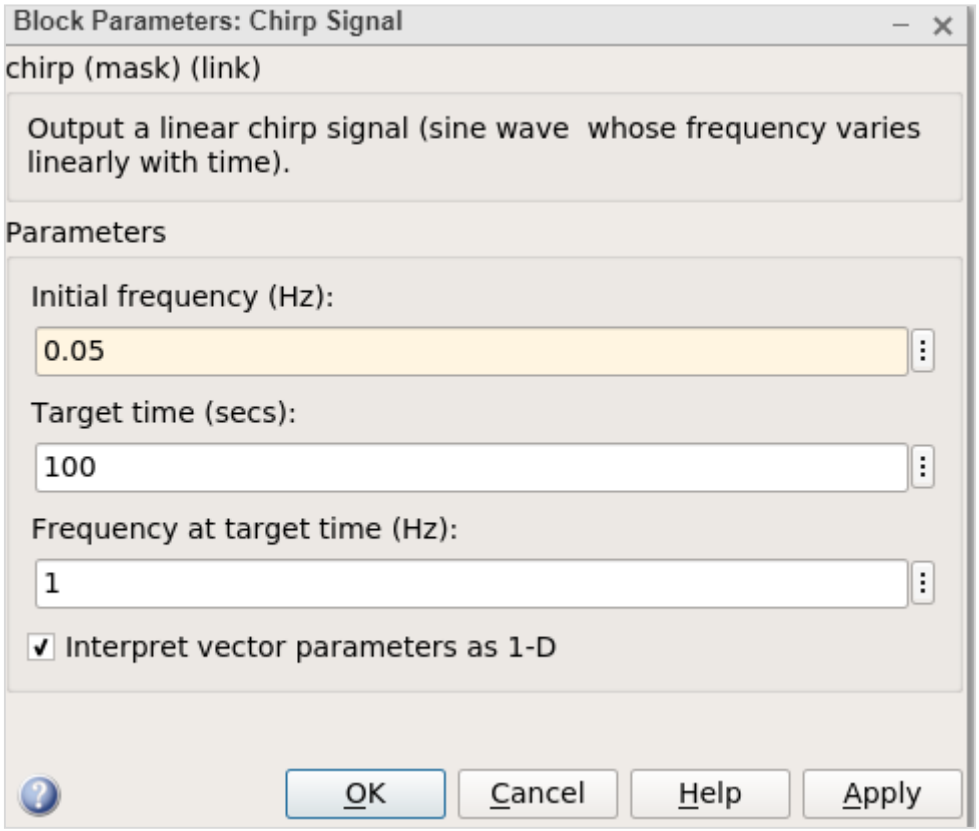
Let us pick the block we want. Select chirp signal and sine wave from sources library, add block from math operations, scope block from sinks library.



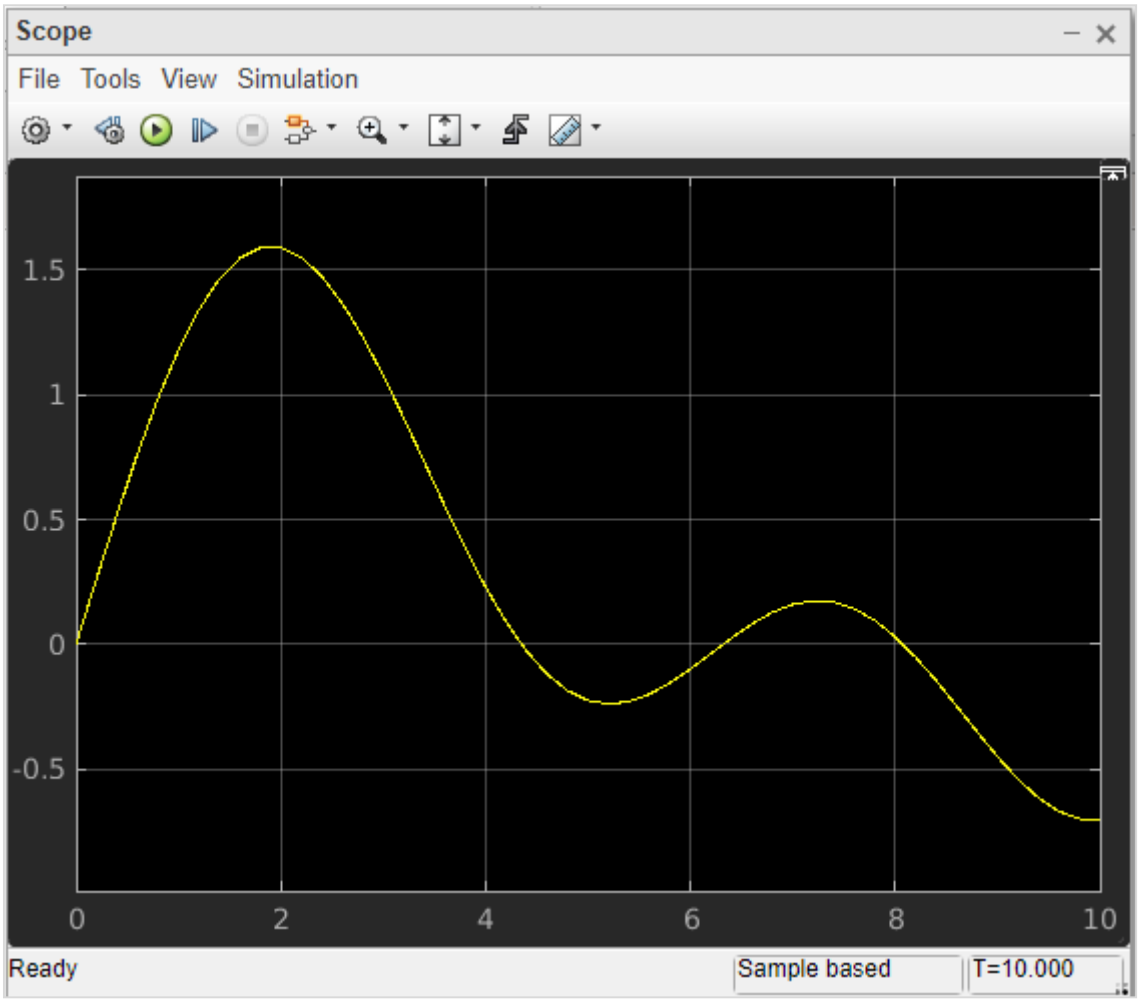
Join the lines to each block.



Double click chirp signal and change the initial frequency from 0.1 to 0.05 and click on Ok button.



The other blocks are kept as the default values. Now, click on run to see the output in scope as shown below.

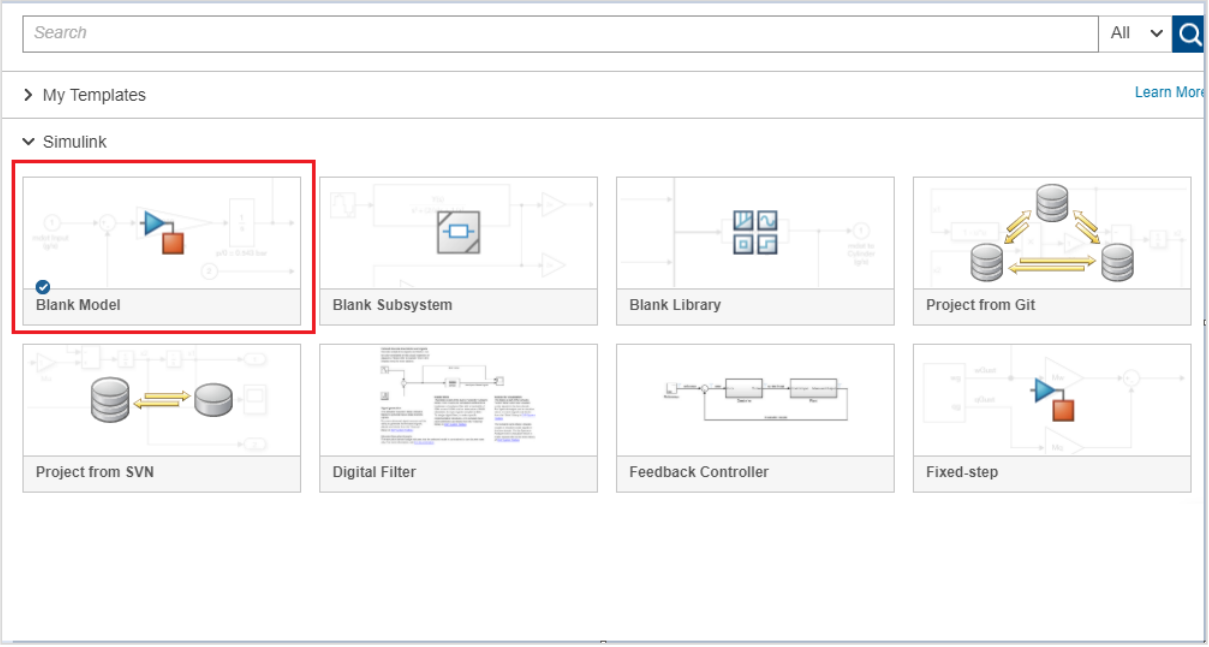


10. MATLAB Simulink — Build Model and Apply If-Else Logic

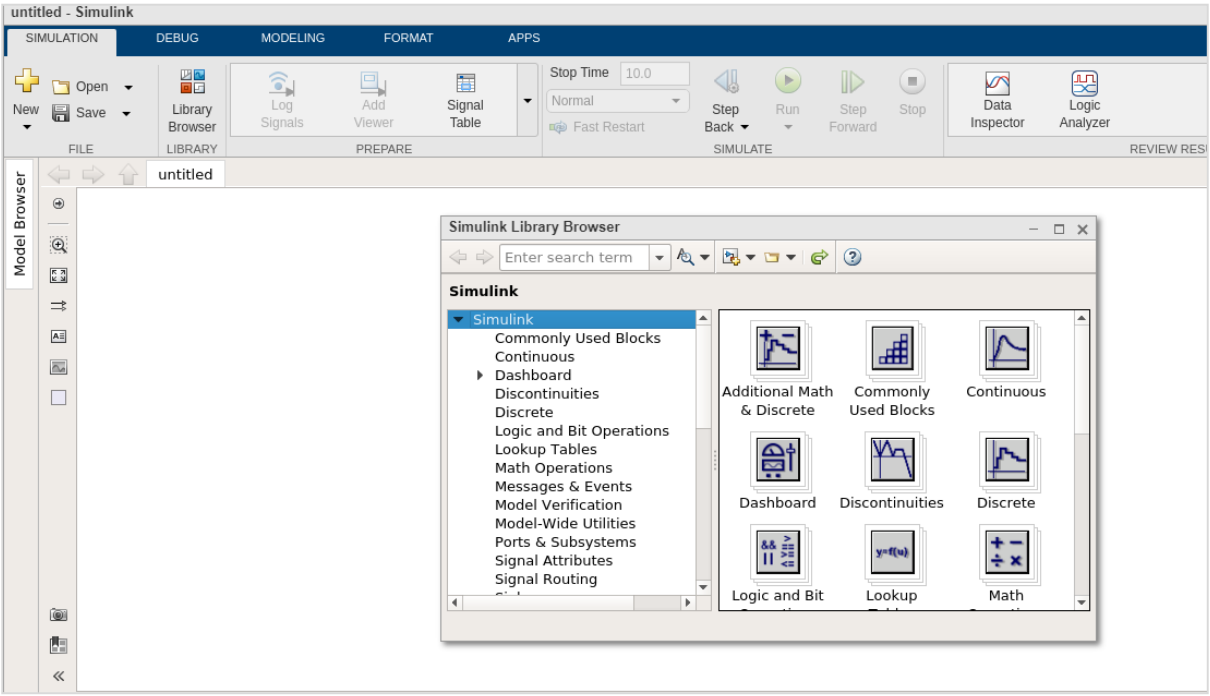
In this chapter, we will create a model and apply if-else logic to it.

Let us first collect blocks to create our model.

Now, open MATLAB Simulink (blank model) and the Simulink library browser as shown below:



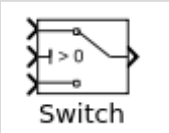
Click on the Blank Model and open Simulink library browser as shown below:



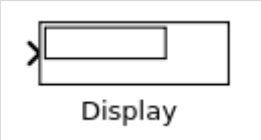
The blocks we require to build the model with if-else logic is as follows:

Constant block from Commonly used blocks

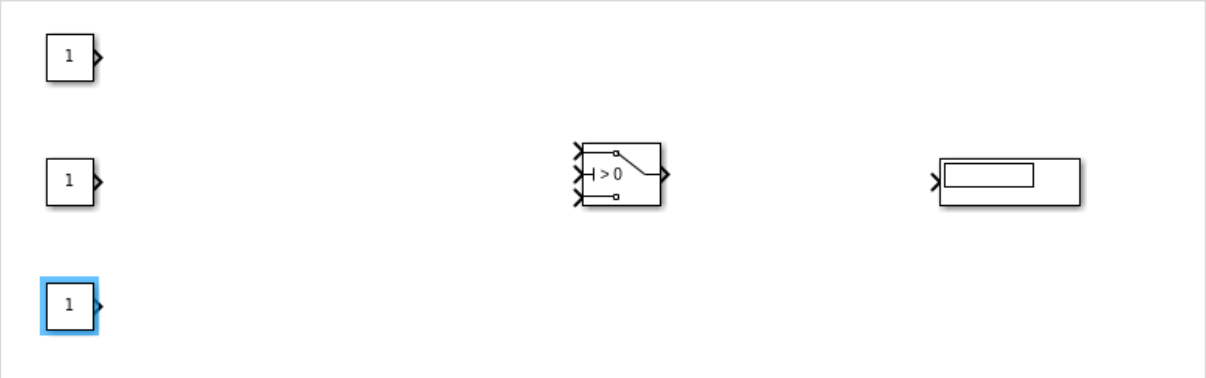
Switch block from Signal Routing



Display block from Sinks

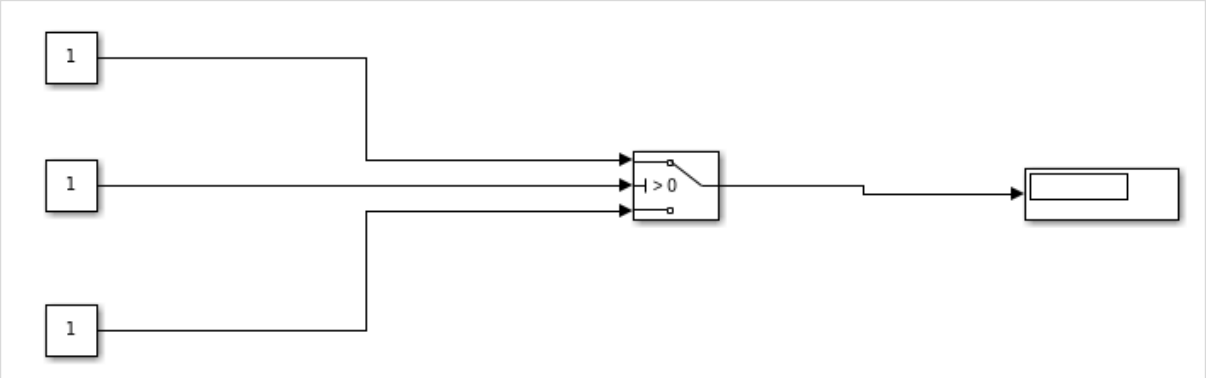


Let us now get all the blocks together to create a model as shown below:



Let us now connect the lines with each block. So you can see that the constant block has one output and the switch has three inputs and one output. We are going to connect them to the display block.

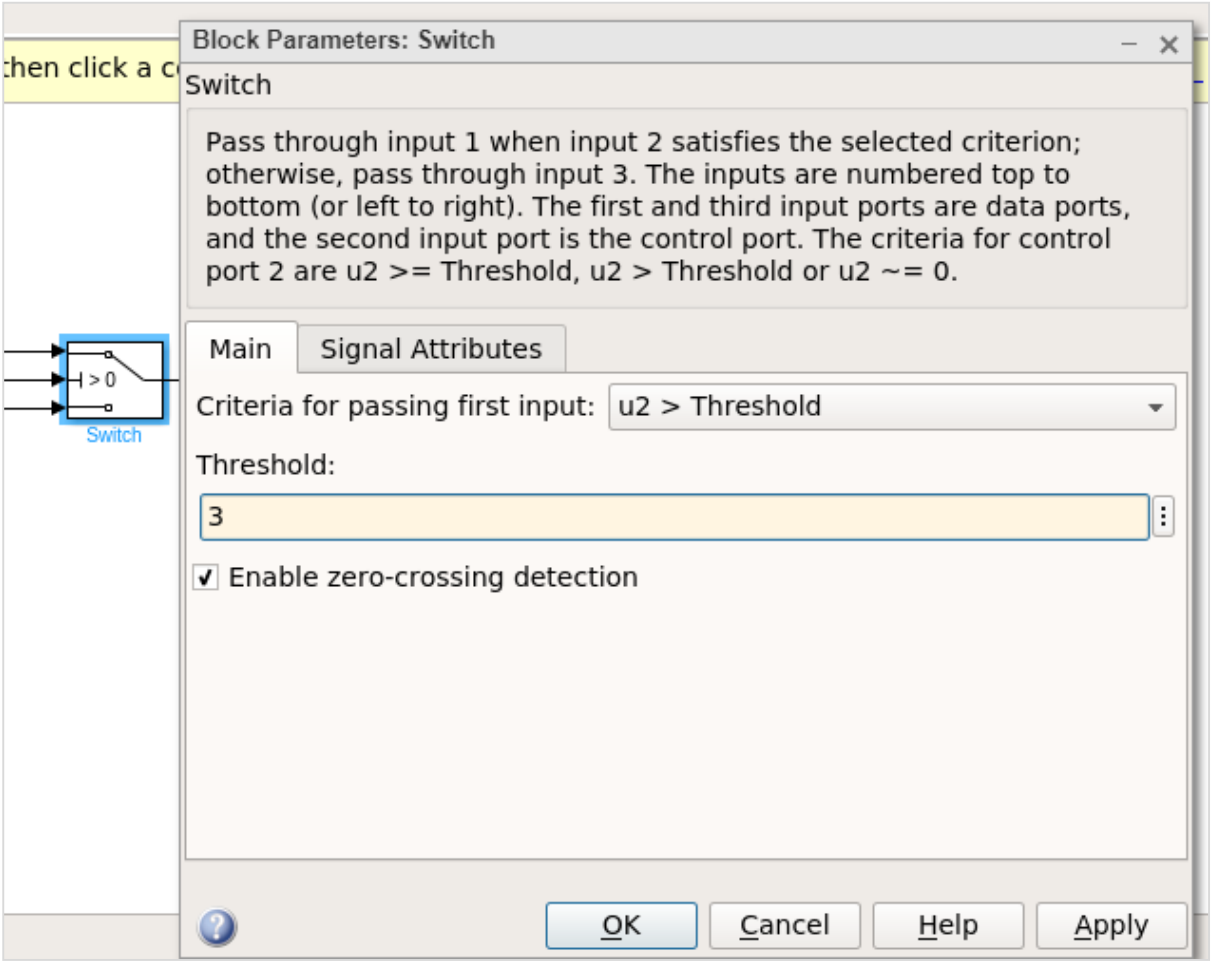
After connecting the lines, the model is as shown below:



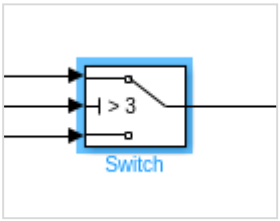
Now, double click the switch block and add a threshold.

The threshold value will be compared with the block in the center. Based on the constant value of the middle block, the first block value will be displayed or the last constant block value will be displayed.

Let us add a threshold value to the switch as shown below:

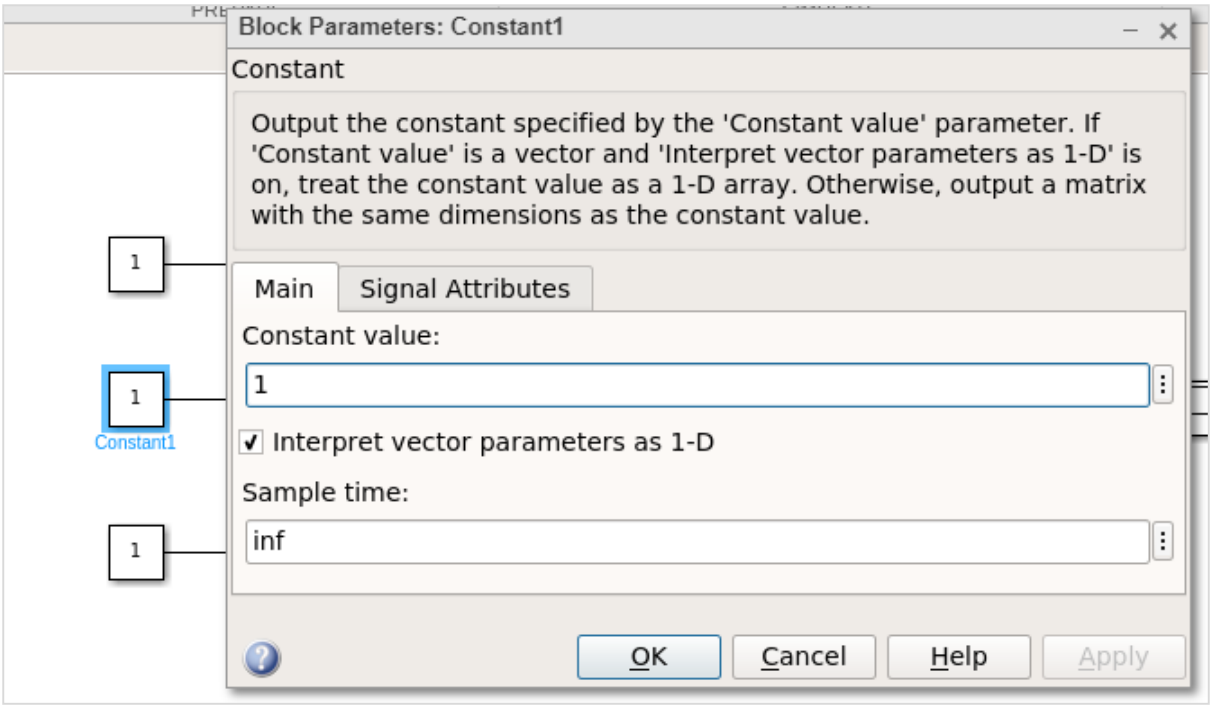


The threshold value given is 3. Click on OK to update the threshold. Now the threshold value is seen inside the switch block as shown below:

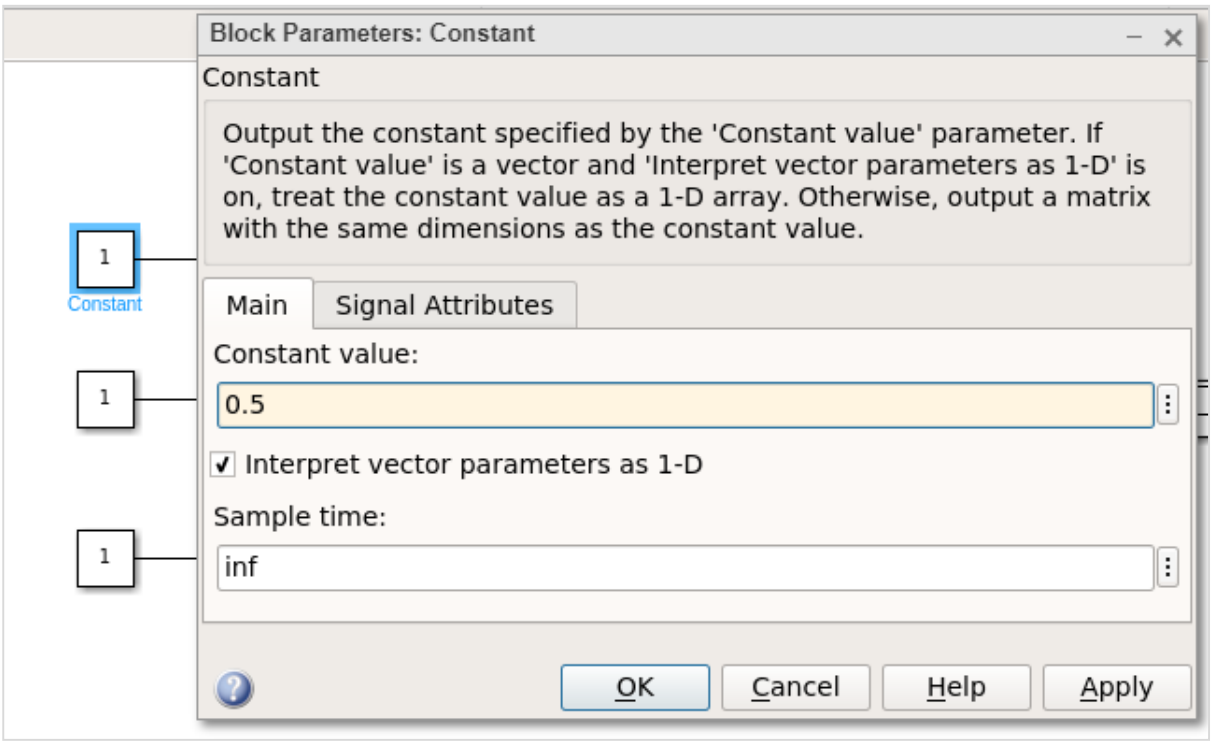


The middle constant block will be compared with the switch threshold and accordingly the display will be decided.

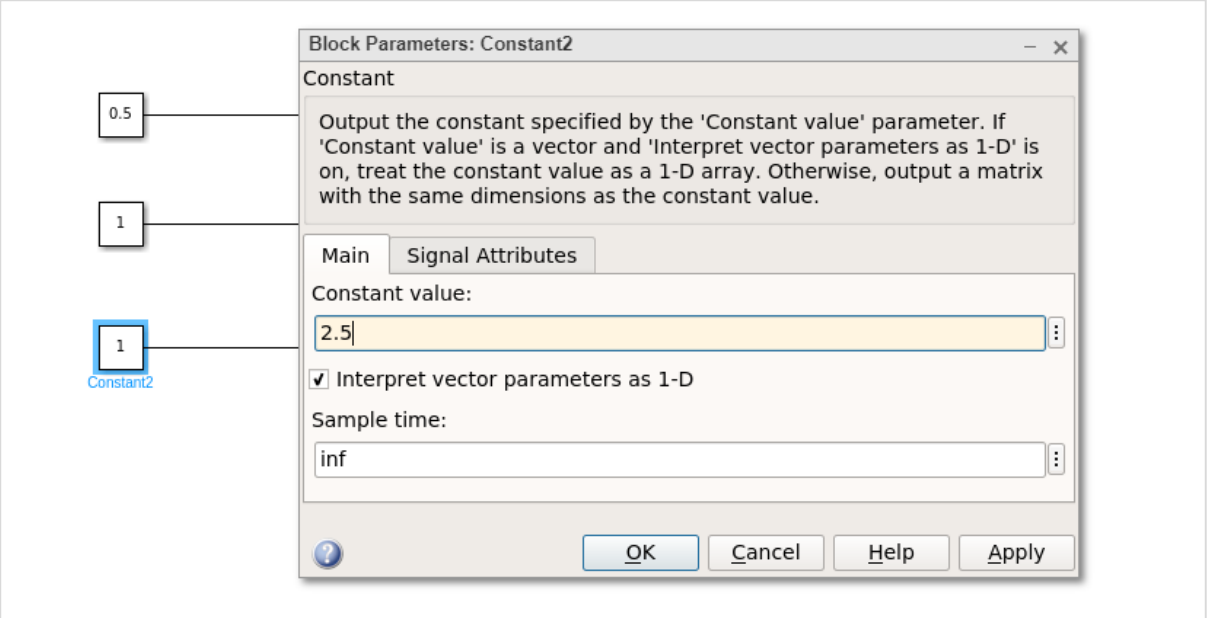
Let us now update the middle constant block with some value as shown below:



The value of the constant block is 1. Let us now change the first constant block and give it a value as 0.5 as shown below:

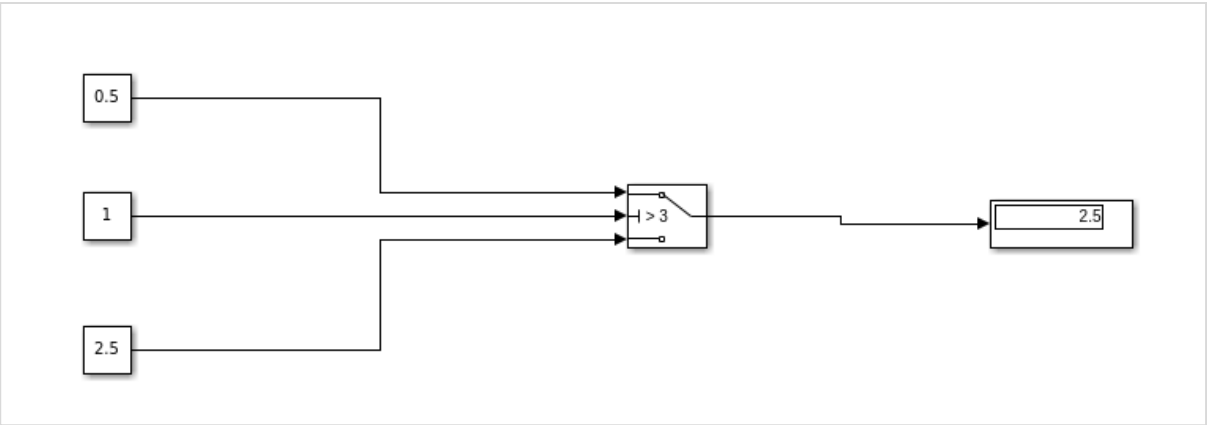


Let us now change the last constant with value as 2.5 as shown below:

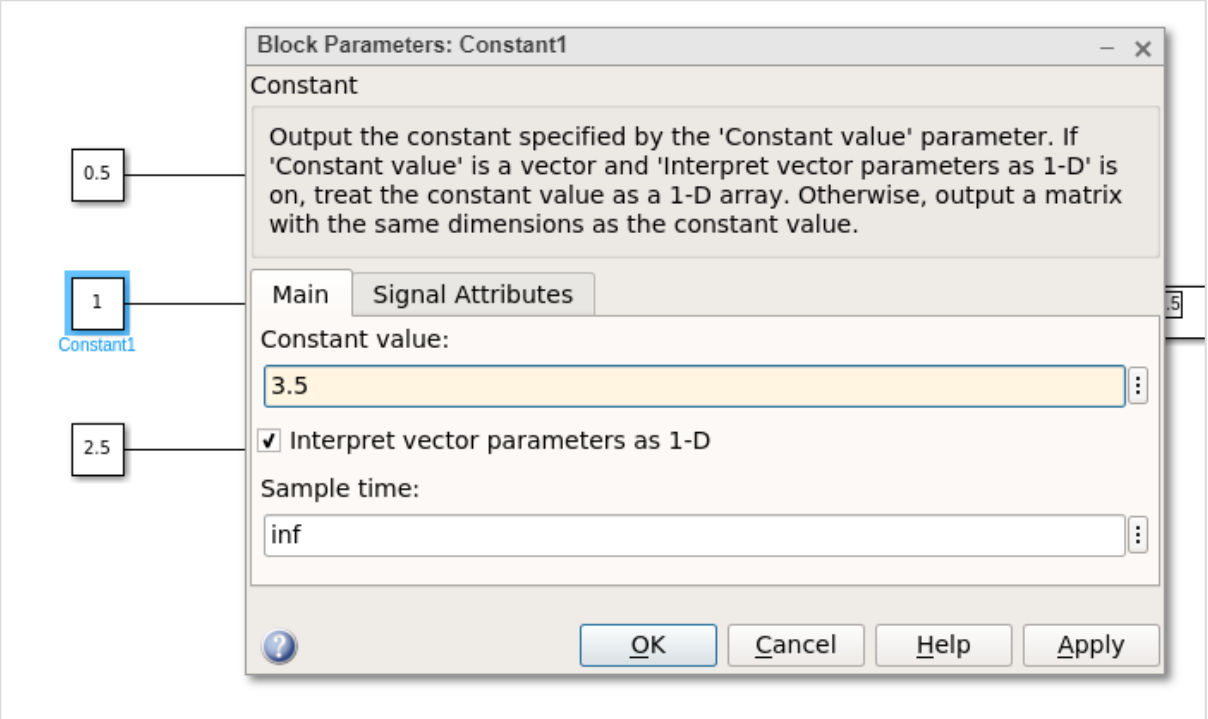


Hence, the first constant value is 0.5, the middle constant value is 1 and the last one is 2.5. The middle constant value 1 will be compared to switch threshold value i.e. 3 as (1 >3). It will print the value as 2.5 the last constant value.

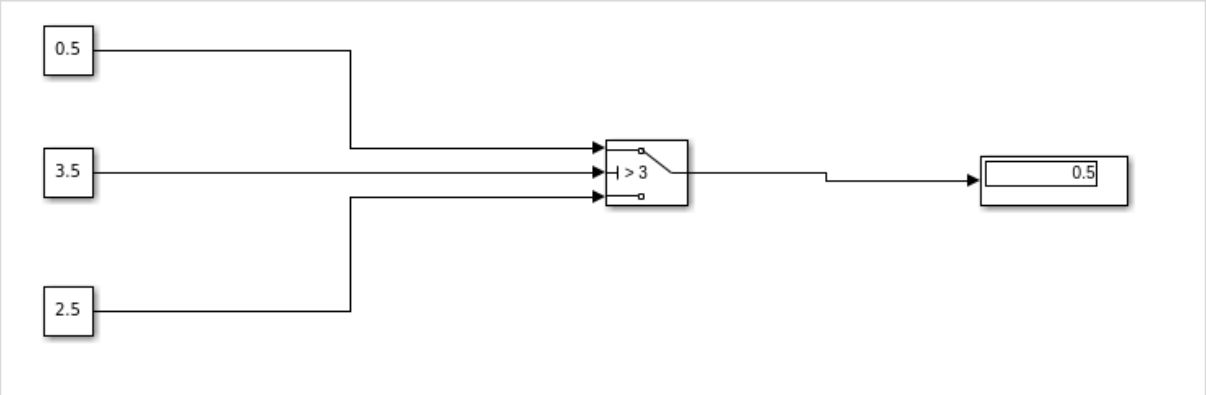
Click on the run button to get the output in the display block as shown below:



Let us now change the middle constant to a higher value than the threshold of switch and see the output:



The value is changed from 1 to 3.5. Click on OK and run the model to see the output in the display:

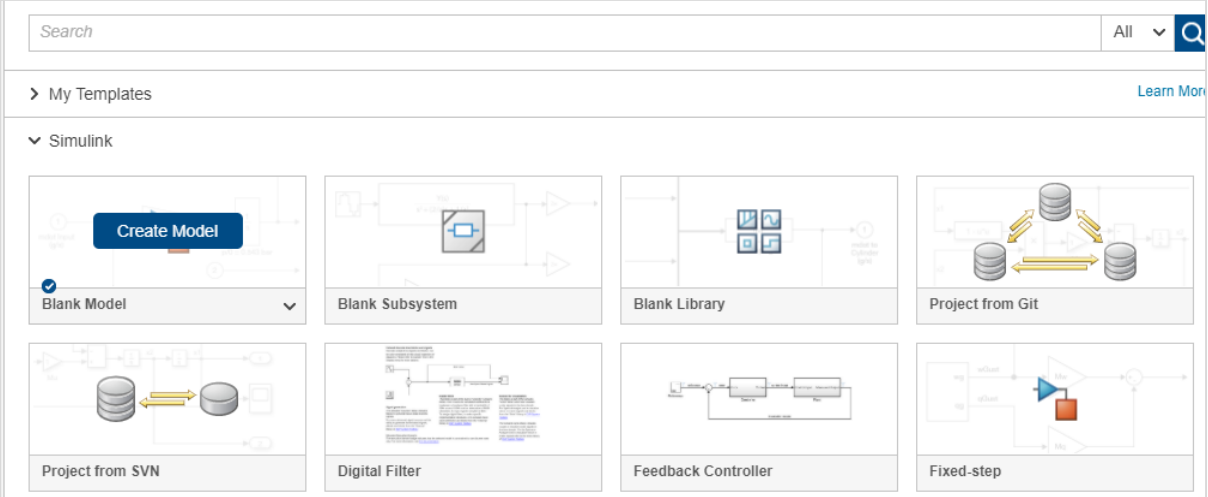


Now, since the value of the middle constant is greater, the value from the first constant is printed in display. If it is less, then the value from the last constant will be printed.

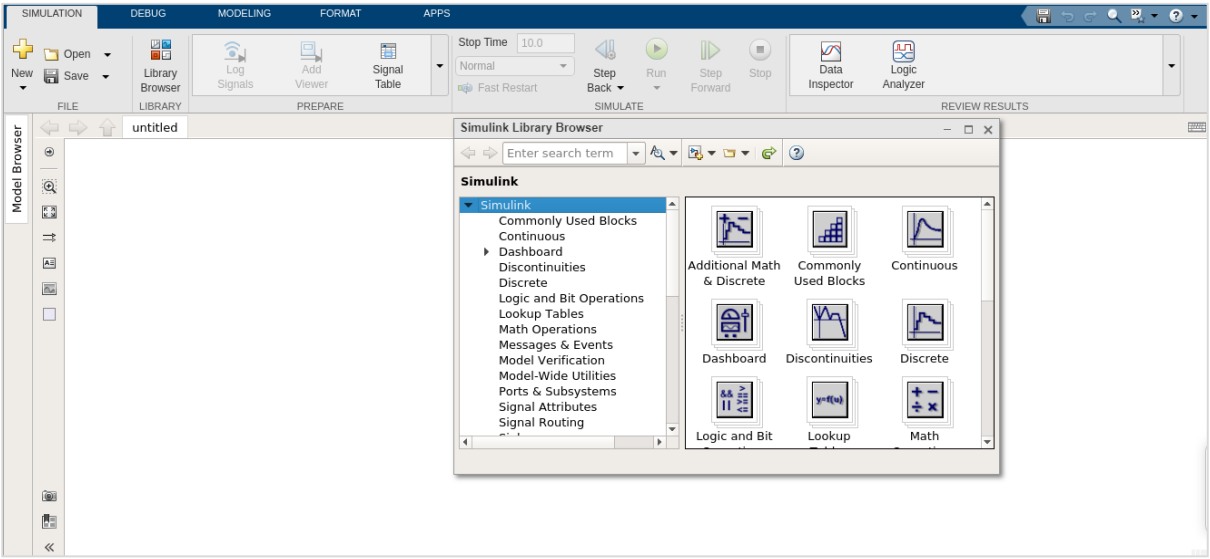
11. MATLAB Simulink — Logic Gates Model

In this chapter, let us understand how to build a model that demonstrates the logic gates. For example, gates like OR, AND, XOR etc.

Open the Simulink and open a blank model as shown below:

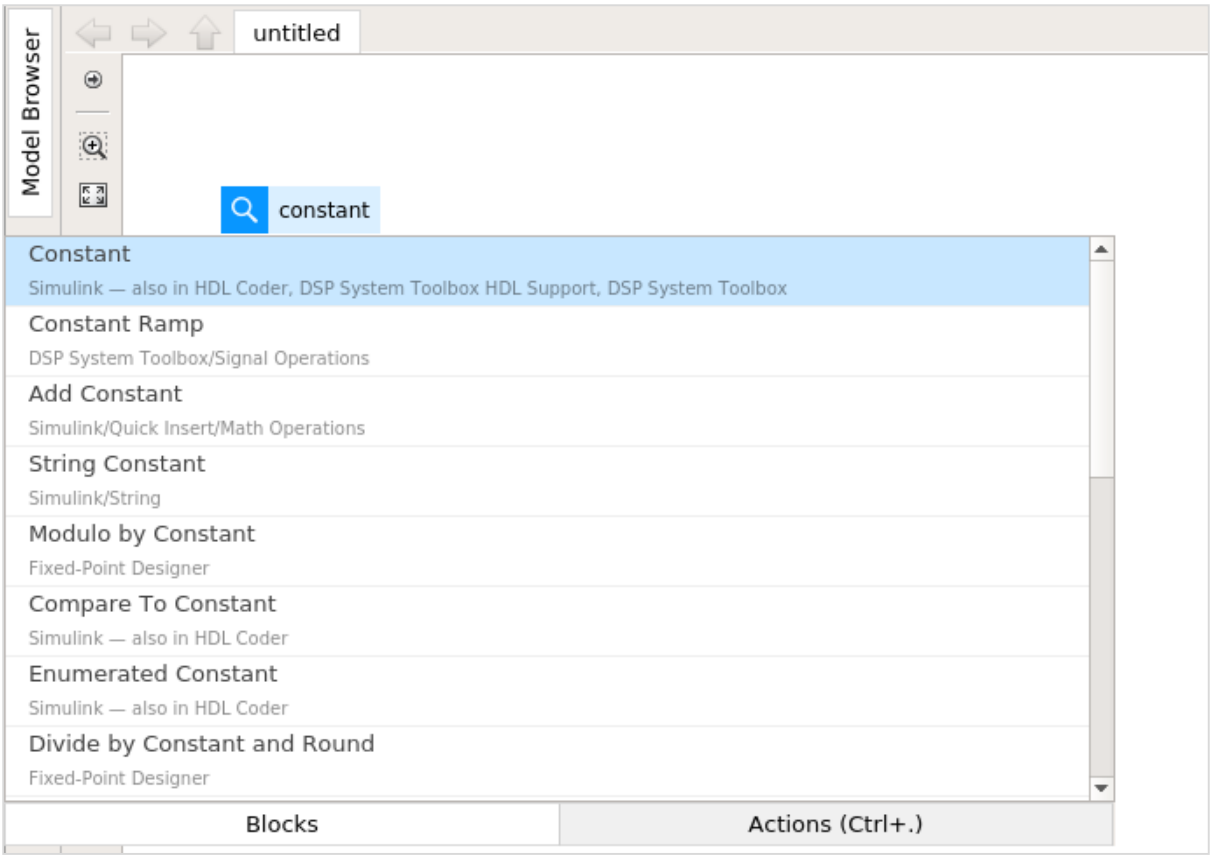


Click on blank model and select the Simulink library as shown below:

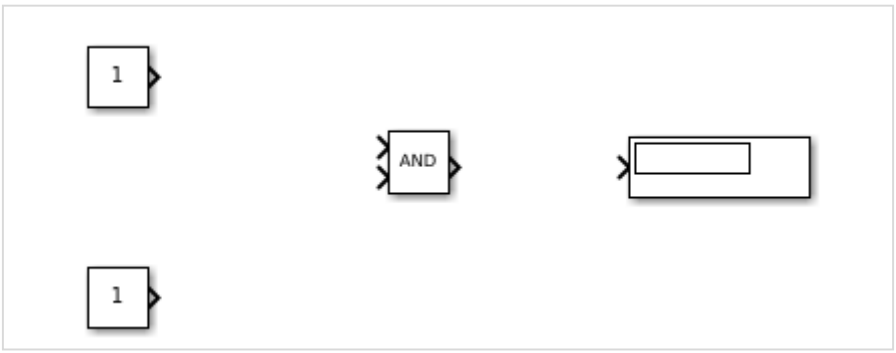


Let us select the block that we want to build a OR gate. We need two constant blocks to act as inputs, a logic operator block and a display block.

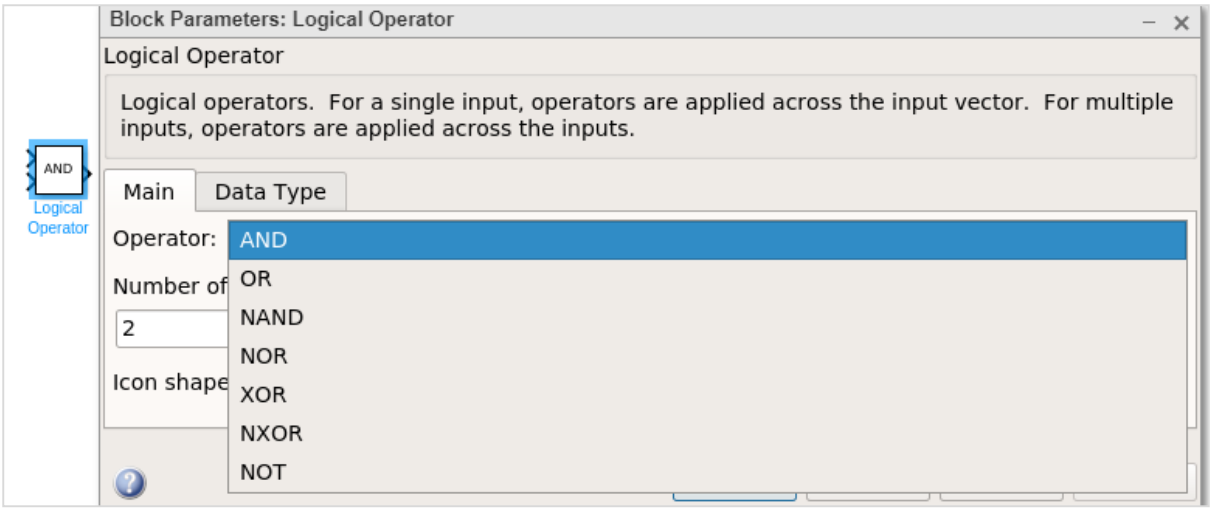
The constant and logic operator block will get from commonly used blocks library. Select the blocks and drag in your model or just type the name of the block in your model and select the block as shown below:



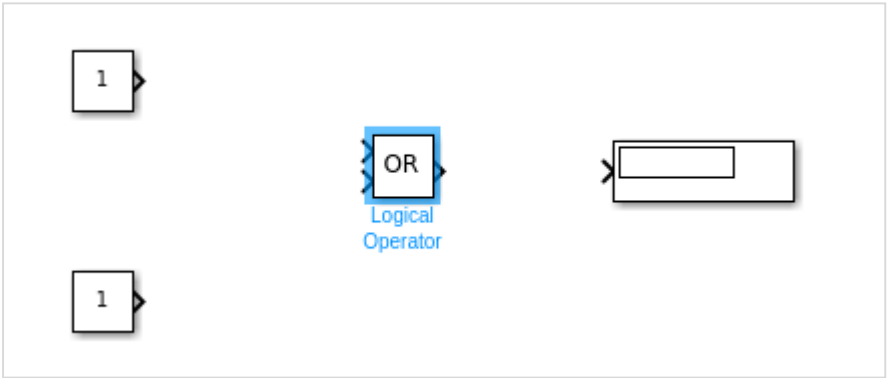
Select the constant block, we need two constant block, a logical operator and a constant. The blocks will look as shown below:



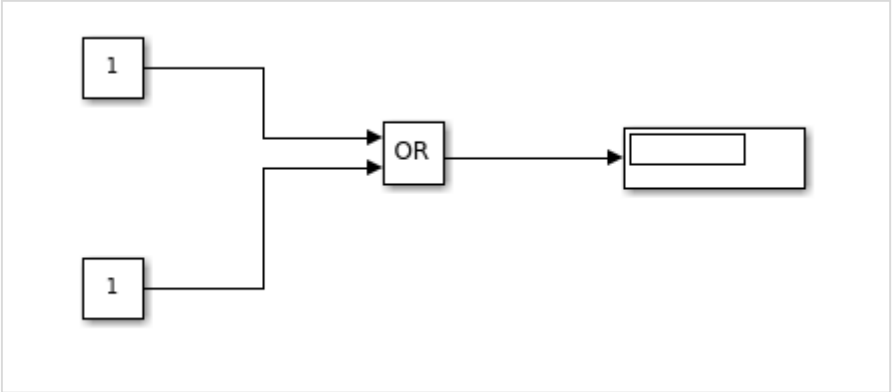
Right click on the logic operator block and it will display the block parameters as shown below:



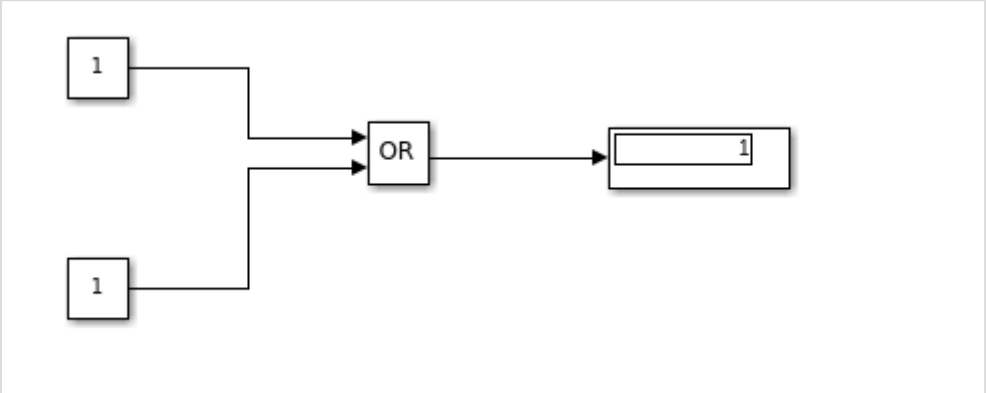
With logical operator you can use AND, OR, NAND, NOR, XOR, NXOR and NOT gates. Right now we are going to select the OR gate.



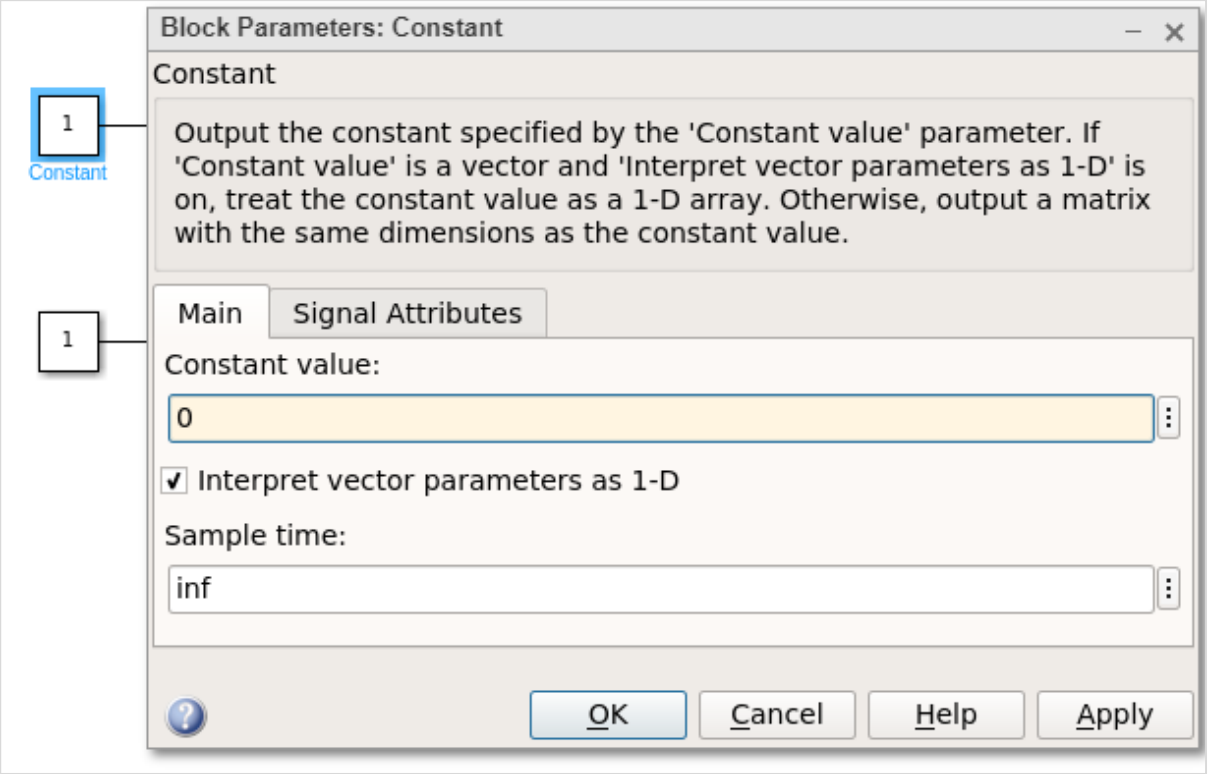
Now connect the lines and the model will be as shown below:



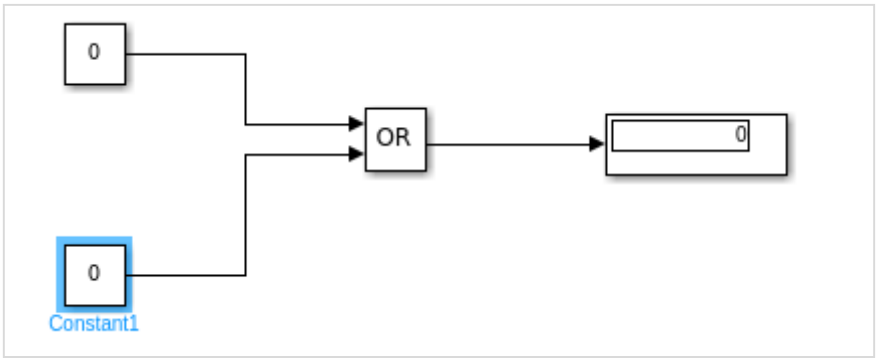
For an OR gate if the inputs are 1,1 the output will be 1. If the inputs are 0,0 the output will be 0. Right now, the constant has values 1,1. Let us run the model to see the output as shown below:



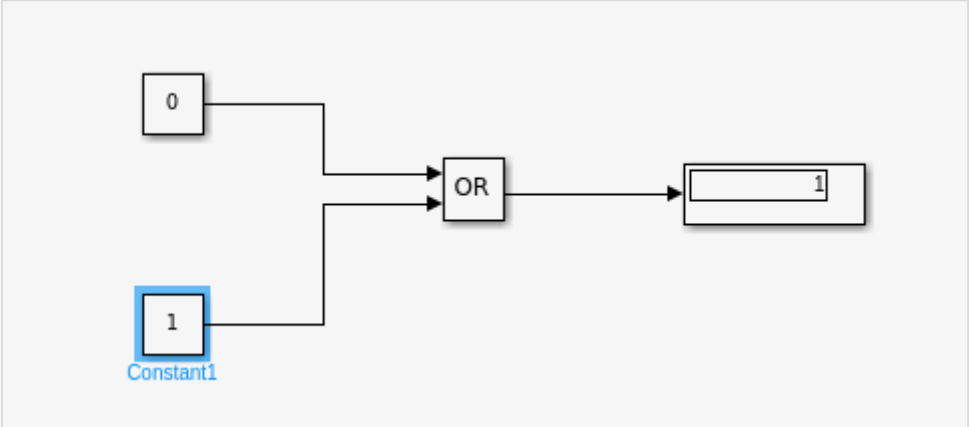
We can see in the display block the output shown is 1. Let us now change the constant value to 0. Right click on constant block and change the value as shown below:



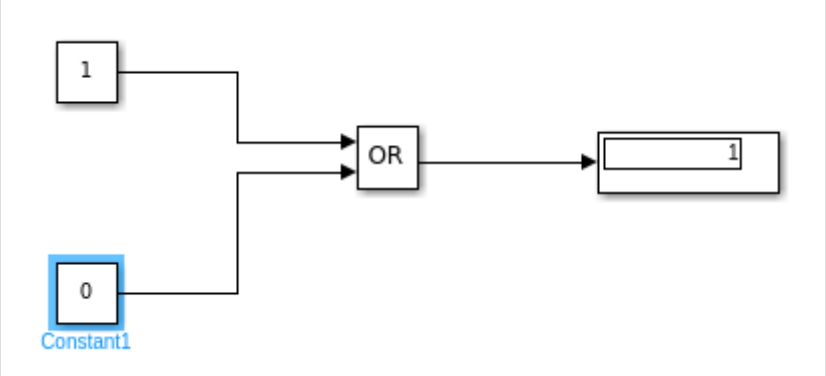
After changing the values of constant to 0, the output will become 0 when you run the model. The output is as shown below:



Let us now change the constant values to 0,1 and see the output:



With values as 1,0, the display will be as follows:

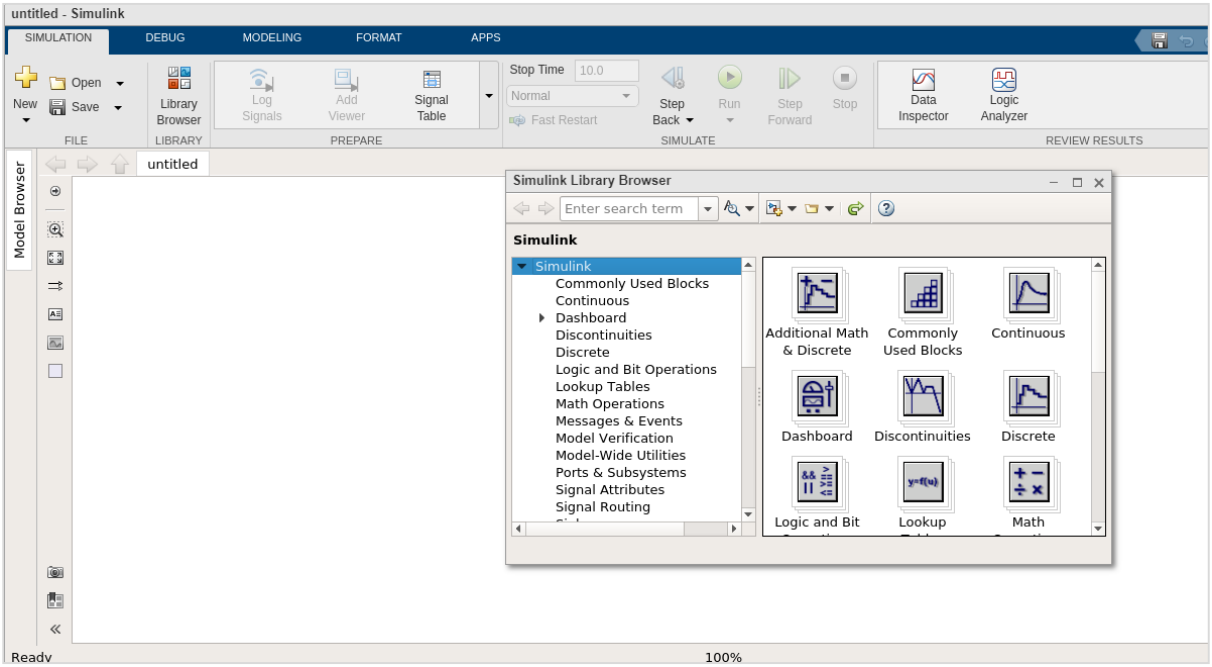


Similarly, you can design the AND and other gates.

12. MATLAB Simulink — Sine wave

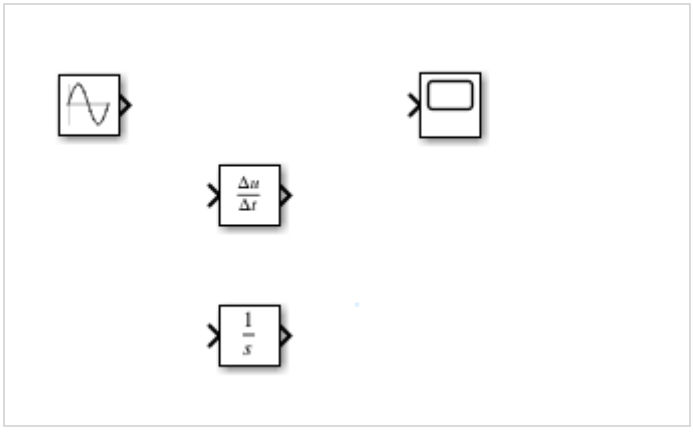
In this chapter we will integrate and differentiate sine wave by using the derivative and integrator blocks.

Open blank model and Simulink library as shown below:



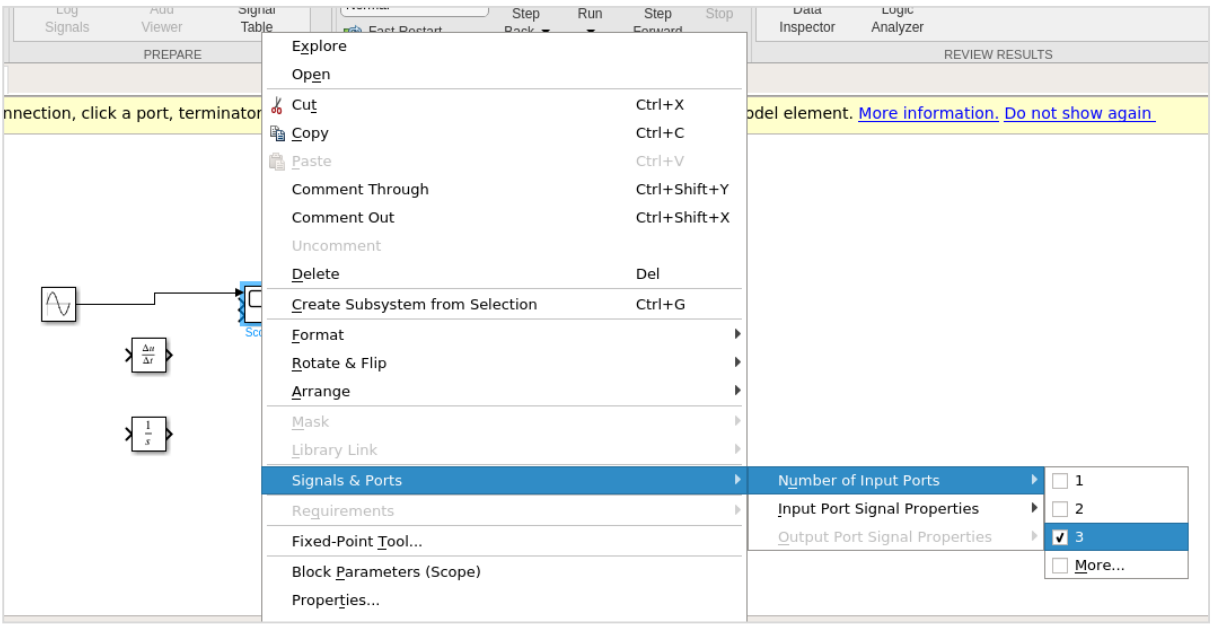
Let us pick the sine wave from sources library and scope block from sinks library.

We would like to add the derivative and integrator block from continuous library as shown below:

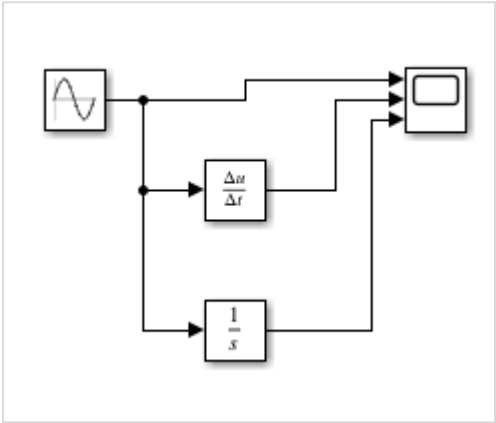


We would need 3 input ports for scope block as the sine wave, derivative and integrator block will be connected to it.

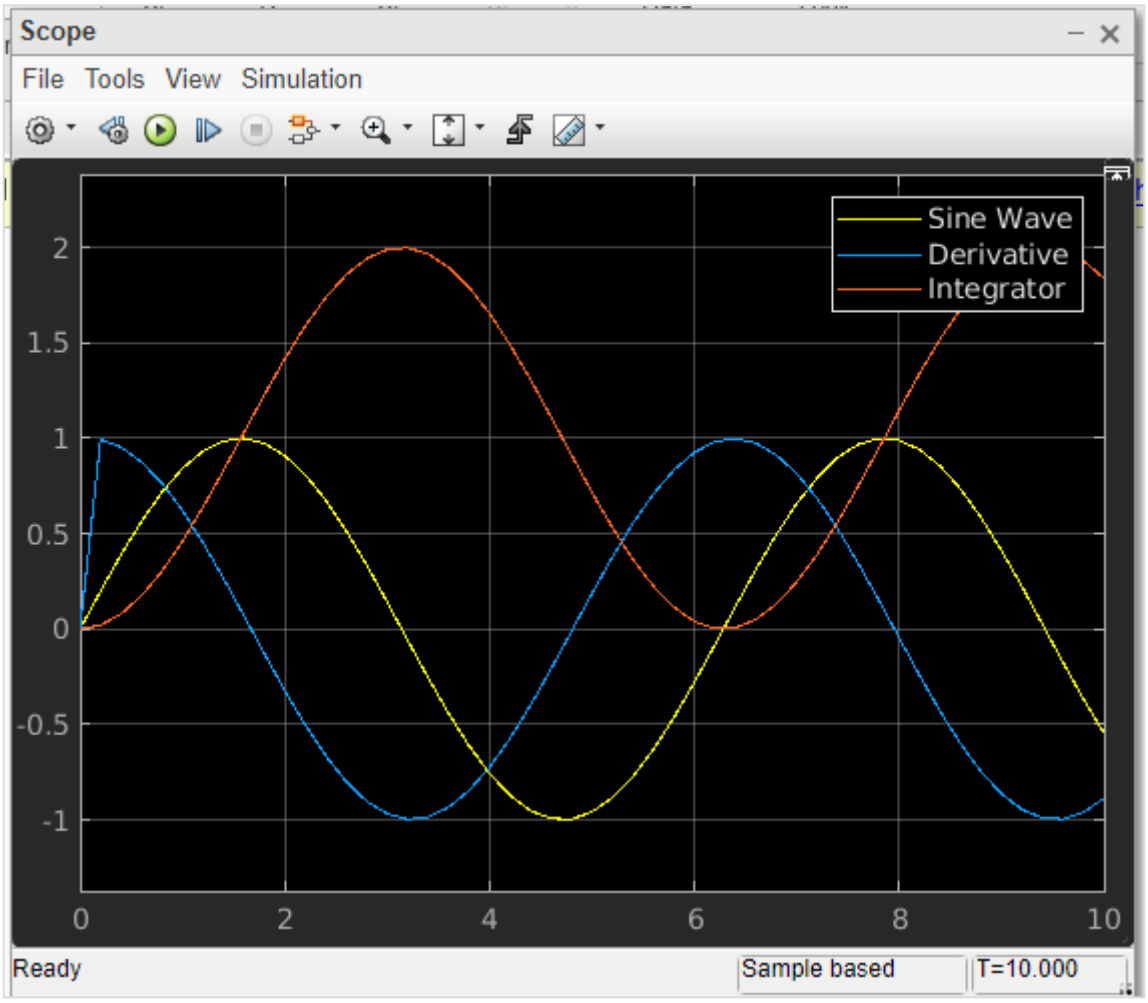
Right click on the scope block and change the inputs from 1 to 3 as shown below:



Connect the lines as shown below:



Now, run the model to see the display.

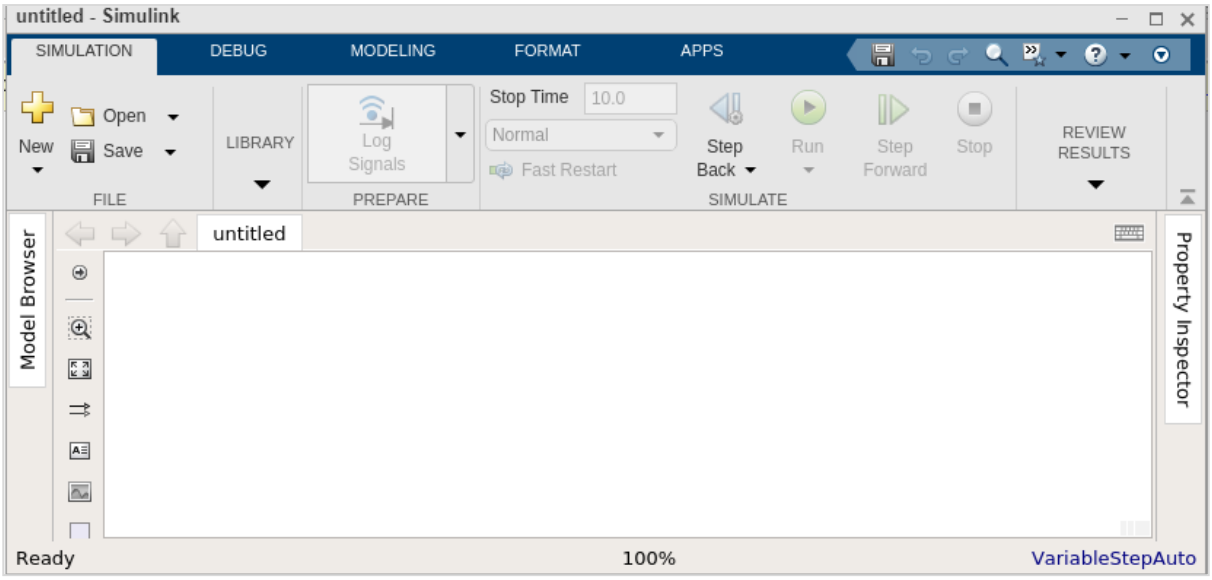


So, we have three signals sine wave, derivative and integrator.

13. MATLAB Simulink — MATLAB Function

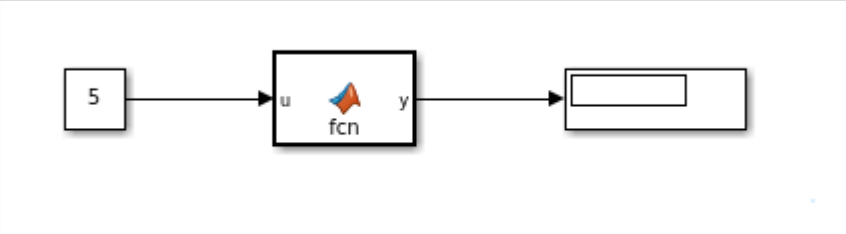
In this chapter we will understand how to make use of MATLAB function in Simulink.

Open Simulink and click on blank model.

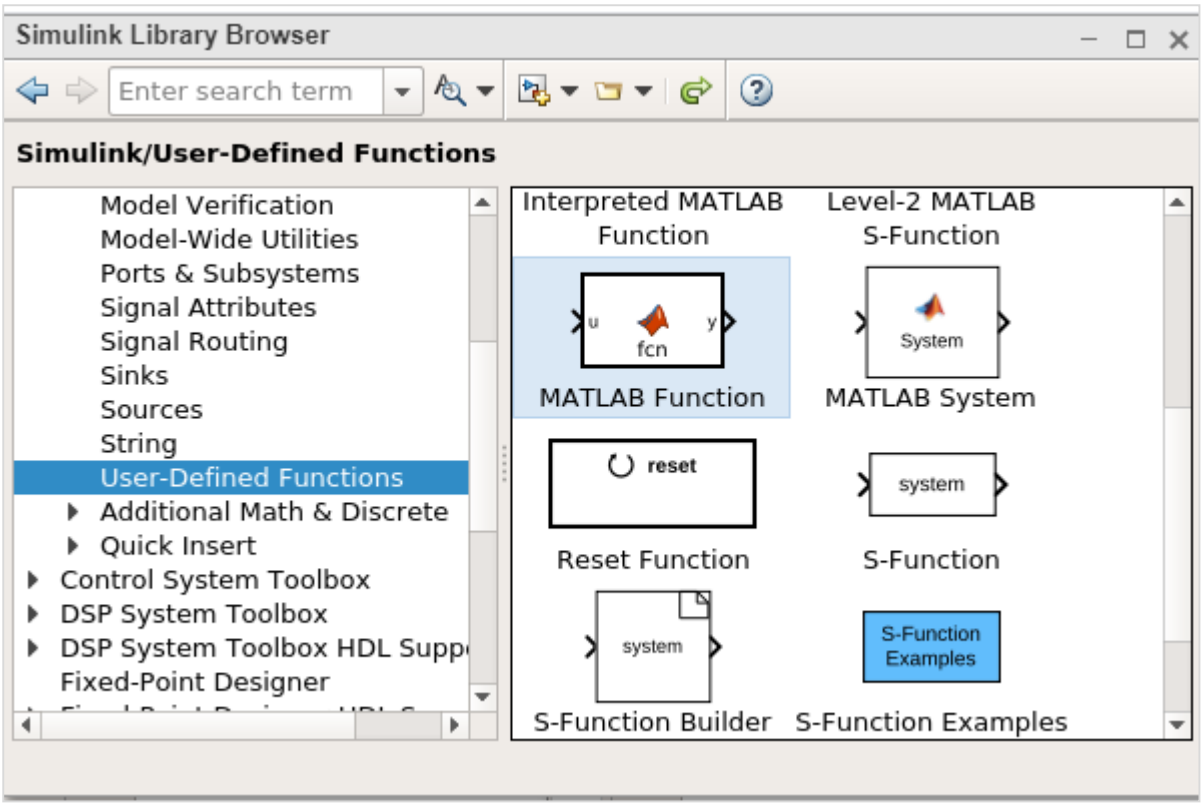


We need a constant block, a MATLAB function block and a display block for the output.

Here is the model created:

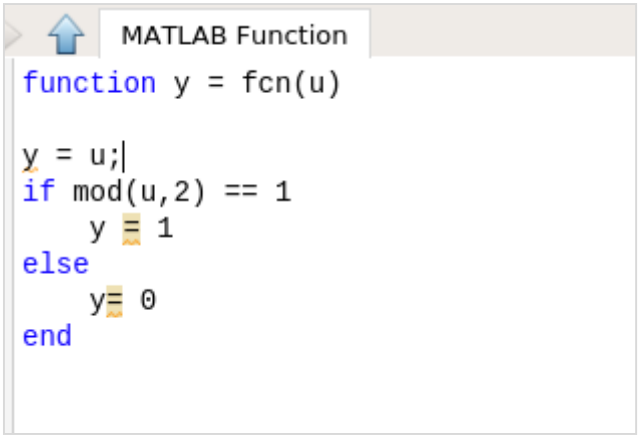


The MATLAB function is available inside the user defined functions library.



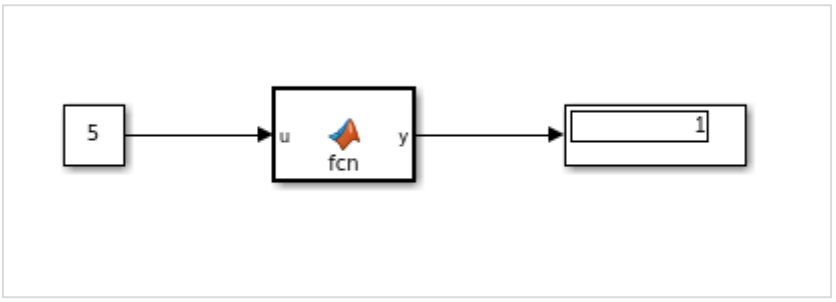
Now, click on MATLAB function and drag it inside your model canvas. Double click on the MATLAB function block and write a function of your choice.

We will try to display 1 if the number given is odd and 0 if it is even. You can also save the function to be used later with another one.



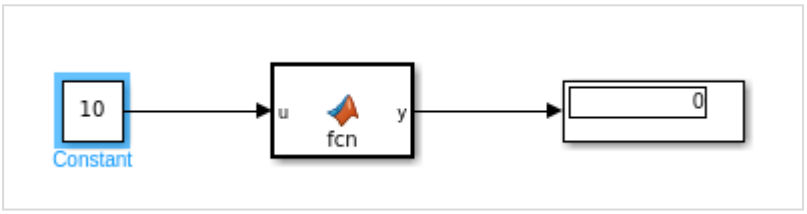
Once the function is done click on the upward arrow close to the MATLAB function and it will take you back to the model you created earlier.

Now, let us test the model by using the constant value as 5.



We can see the display as 1, which indicates that the value 5 is an odd number.

Let us now change the constant value to 10 as shown below and run the simulation: The output is 0 indicating it is even number.

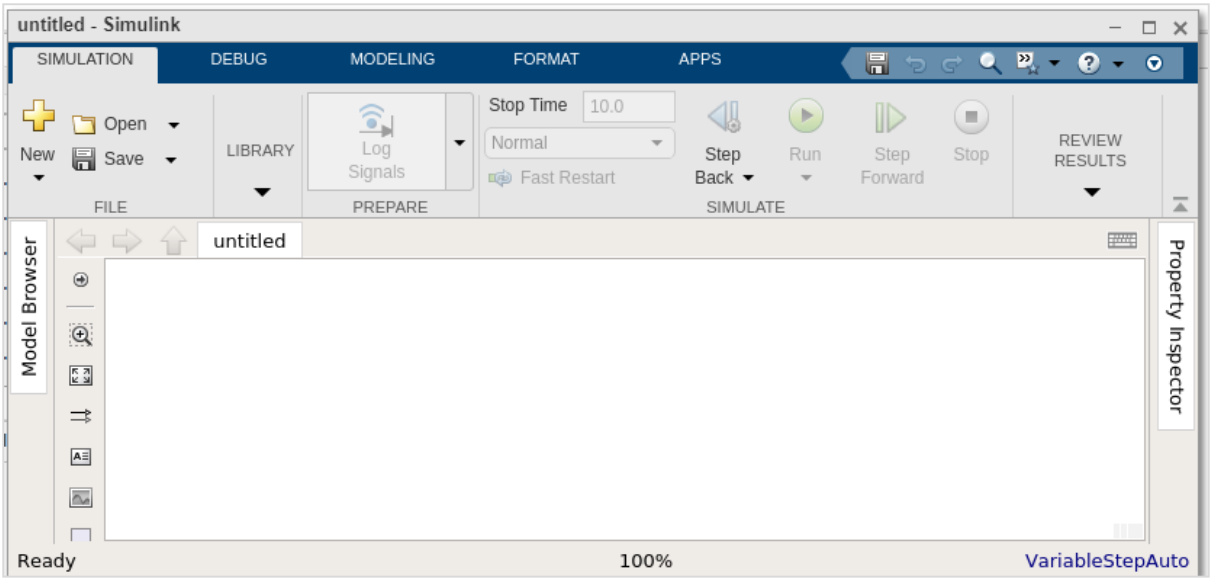


14. MATLAB Simulink — Create Subsystem

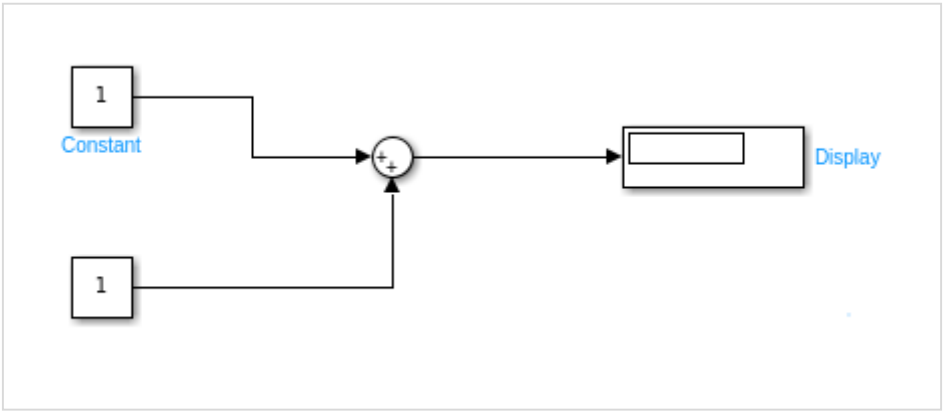
Subsystems are useful when your model gets big and complex. You can change a part of the model into a subsystem that helps to keep the flow very clean and understandable.

In this chapter, let us learn how to create a simple subsystem in Simulink.

First, create a blank model, as shown below:

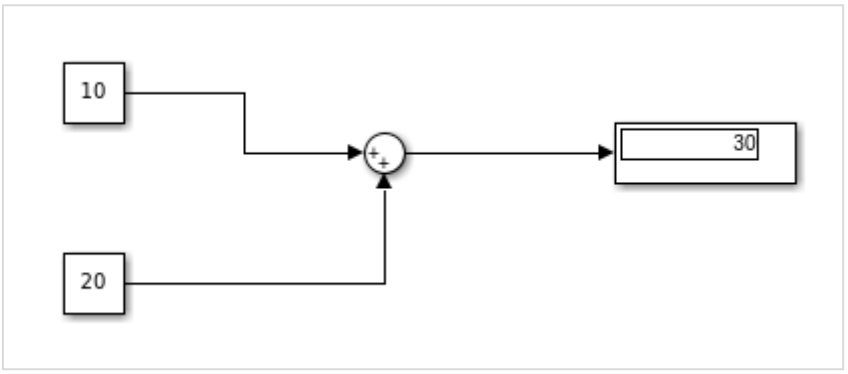


Now, we will create a simple model that adds two numbers and later converts a part of the model into a subsystem.

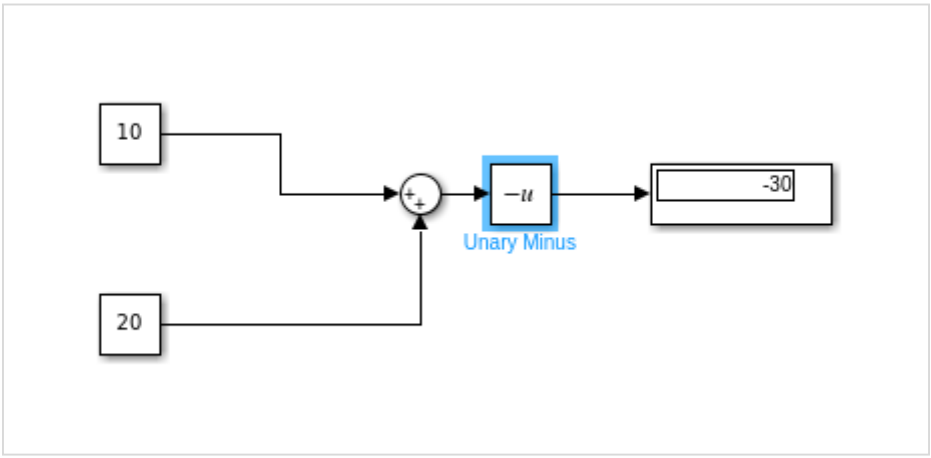


We have created a simple model that has two inputs. These inputs will add up and show the result inside the display.

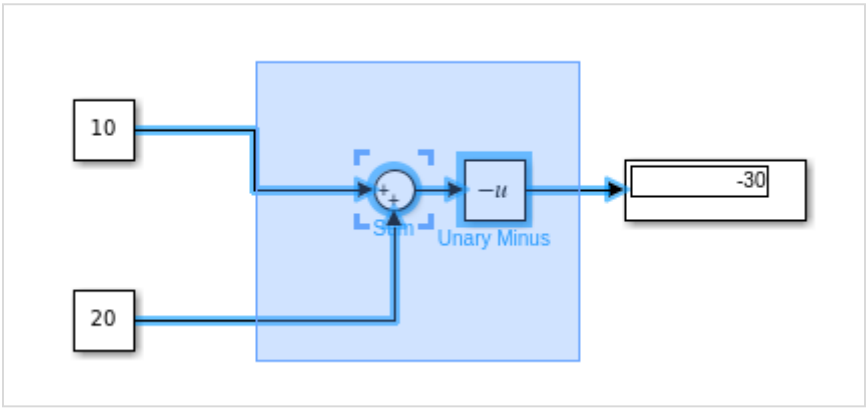
We will change the value of the constant as 10 and 20. The result 10+20 = 30 should be seen inside the display block.

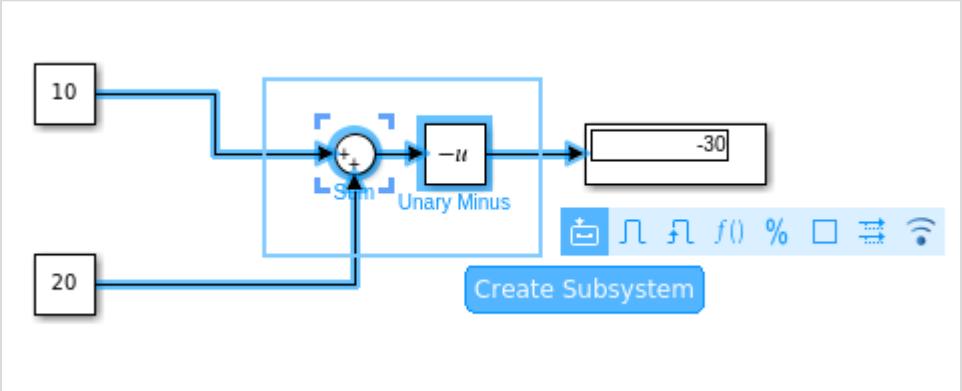


Let us add one more block named Unary minus that will change the output from 30 to -30 as shown below:

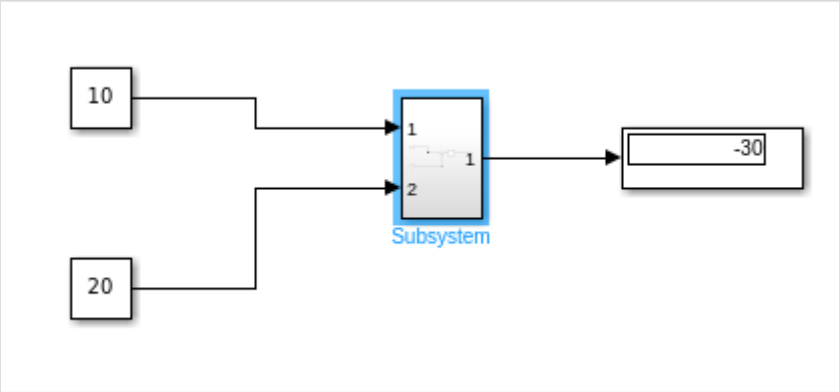


Now, let us select the portion i.e. the sum and the Unary minus block to create a subsystem as shown below:

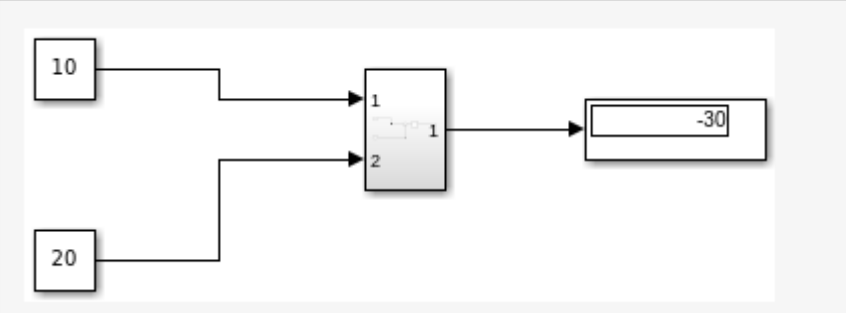




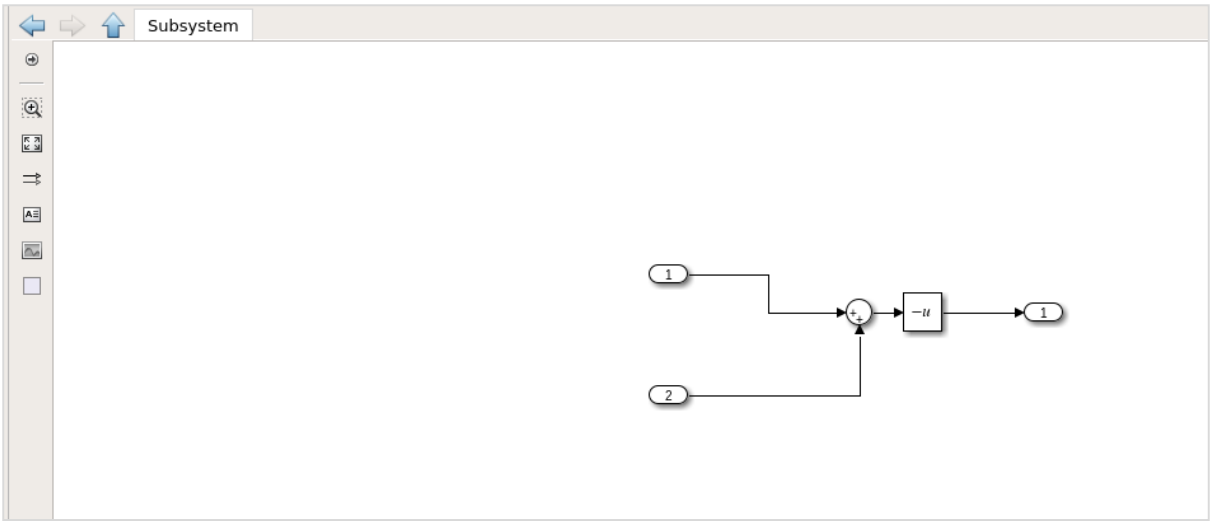
Click on Create Subsystem. Once done, the sum block and the Unary minus block will be converted into a subsystem as shown below:



Now when you run the simulation, it will display the same result as earlier:



Double click on the subsystem to see the original block as shown below:

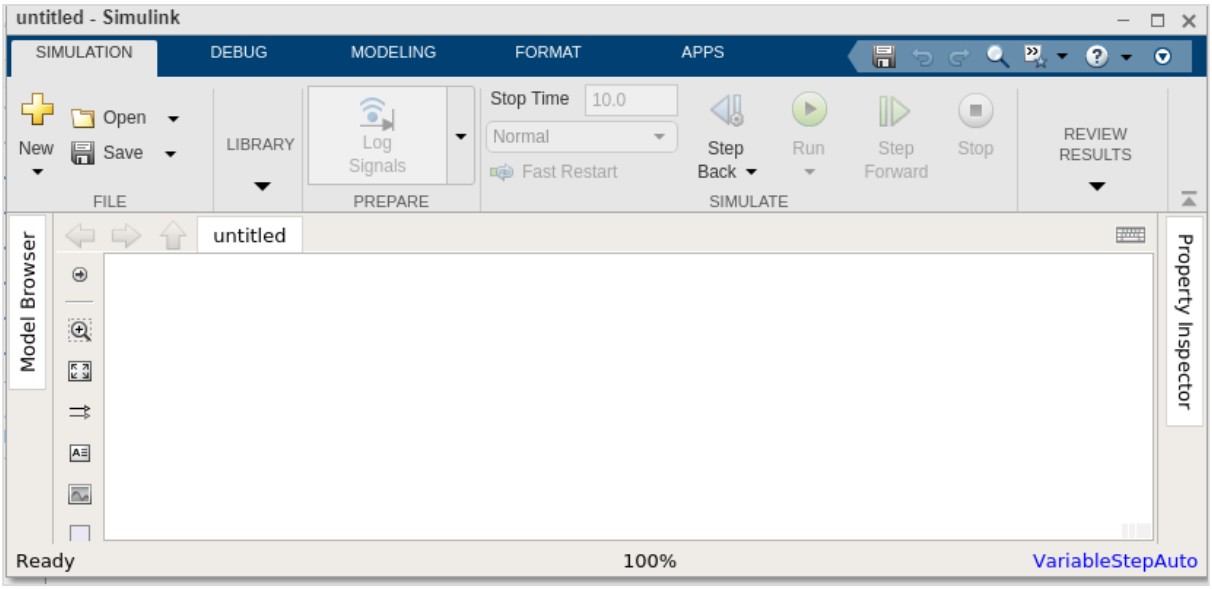


Click on the upward arrow close to the subsystem to get back to the model.

15. MATLAB Simulink — for loop

In this chapter, let us understand the working of for-iterator block.

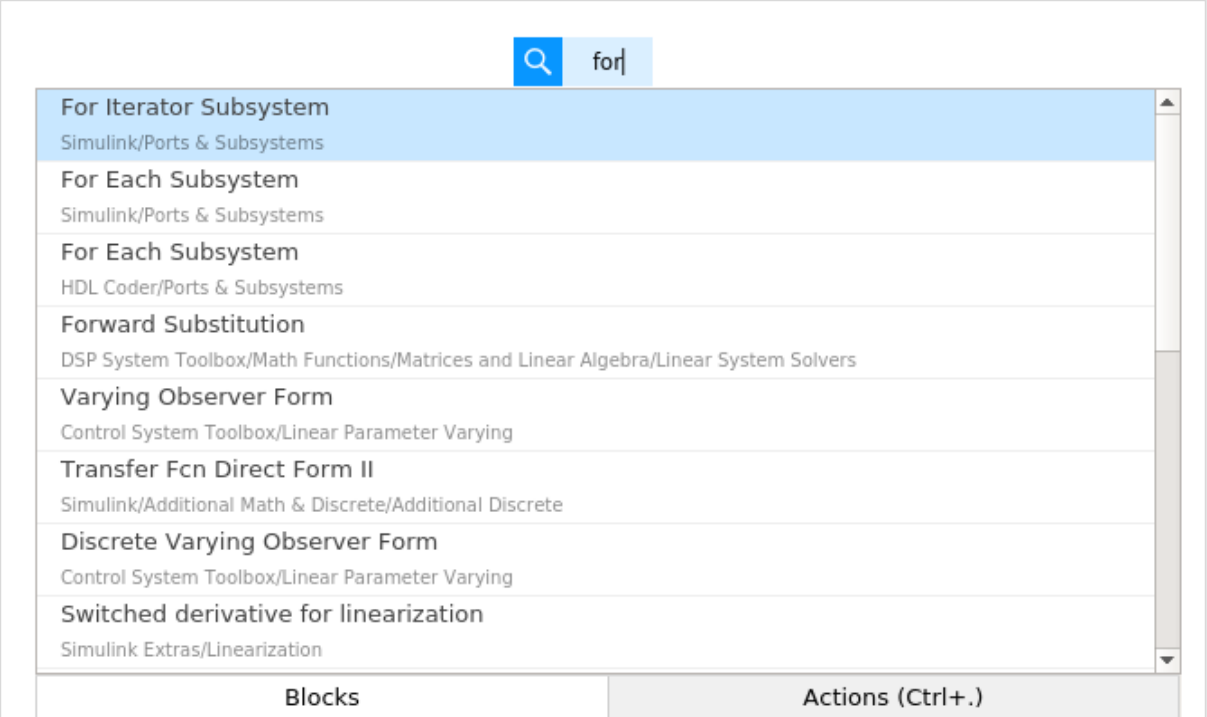
First, create a blank model as shown below:



In this model, we are going to make use of **for iterator** that will give us the sum of 1..N.

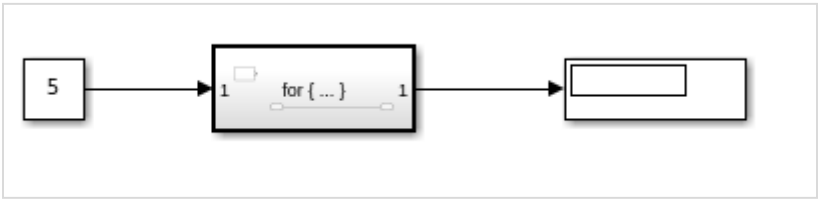
You can use the value of n as per your choice. This value will take the constant block and update it with value 5 as shown below:

Let us add the for-iterator block as shown below:

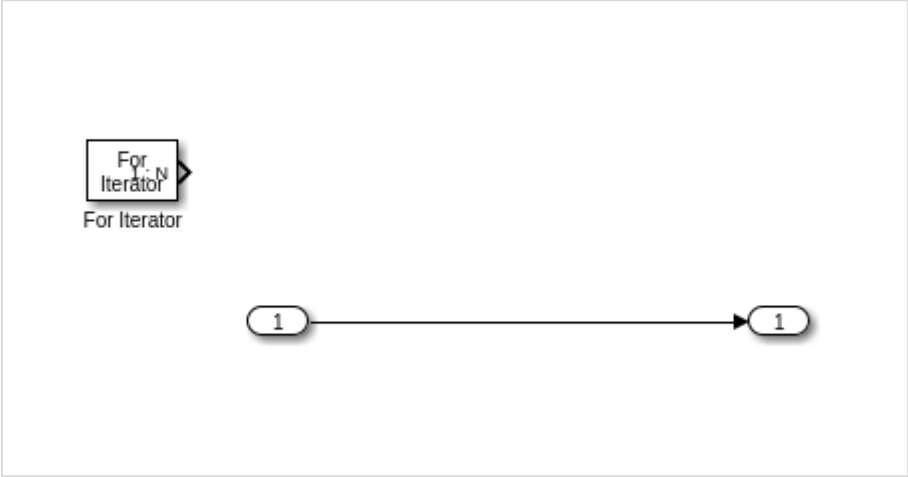


Select the for Iterator subsystem block and add to your model. Next, we need display block as shown below:

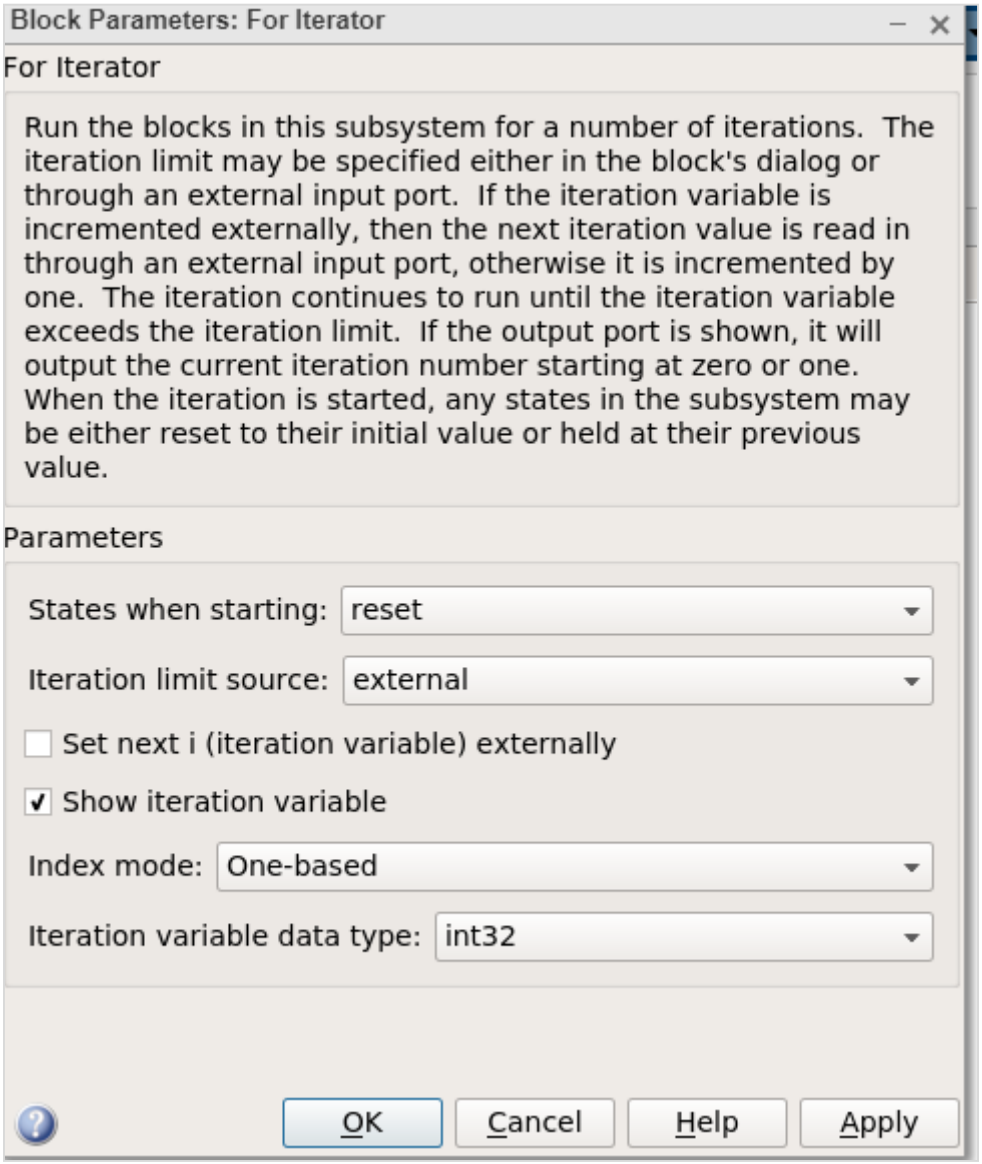
Connect the blocks as shown below:



The for iterator block is a subsystem. Select the block and click enter. It will take you to new model area, where the for block has to be defined.

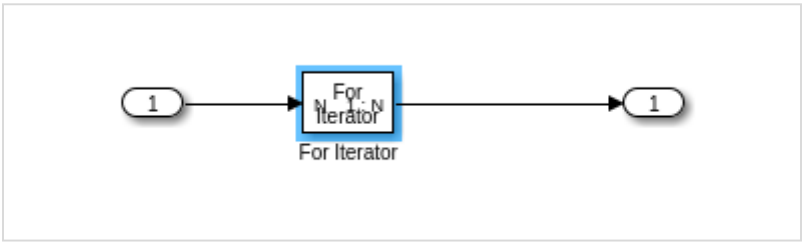


Right click on the for iterator and select the block parameters, as shown below:

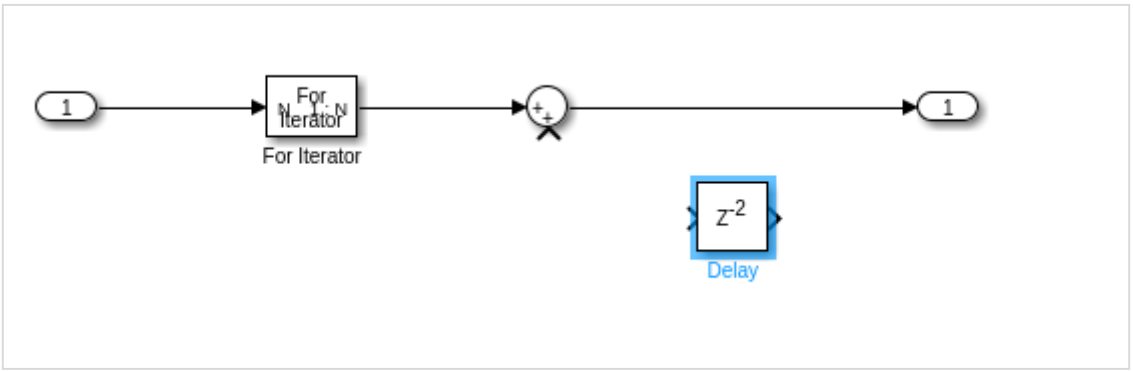


Change the States when starting as reset and Iteration limit source as external. Click on Ok to update the changes.

Now, you will get an input block to your for loop, as given below:

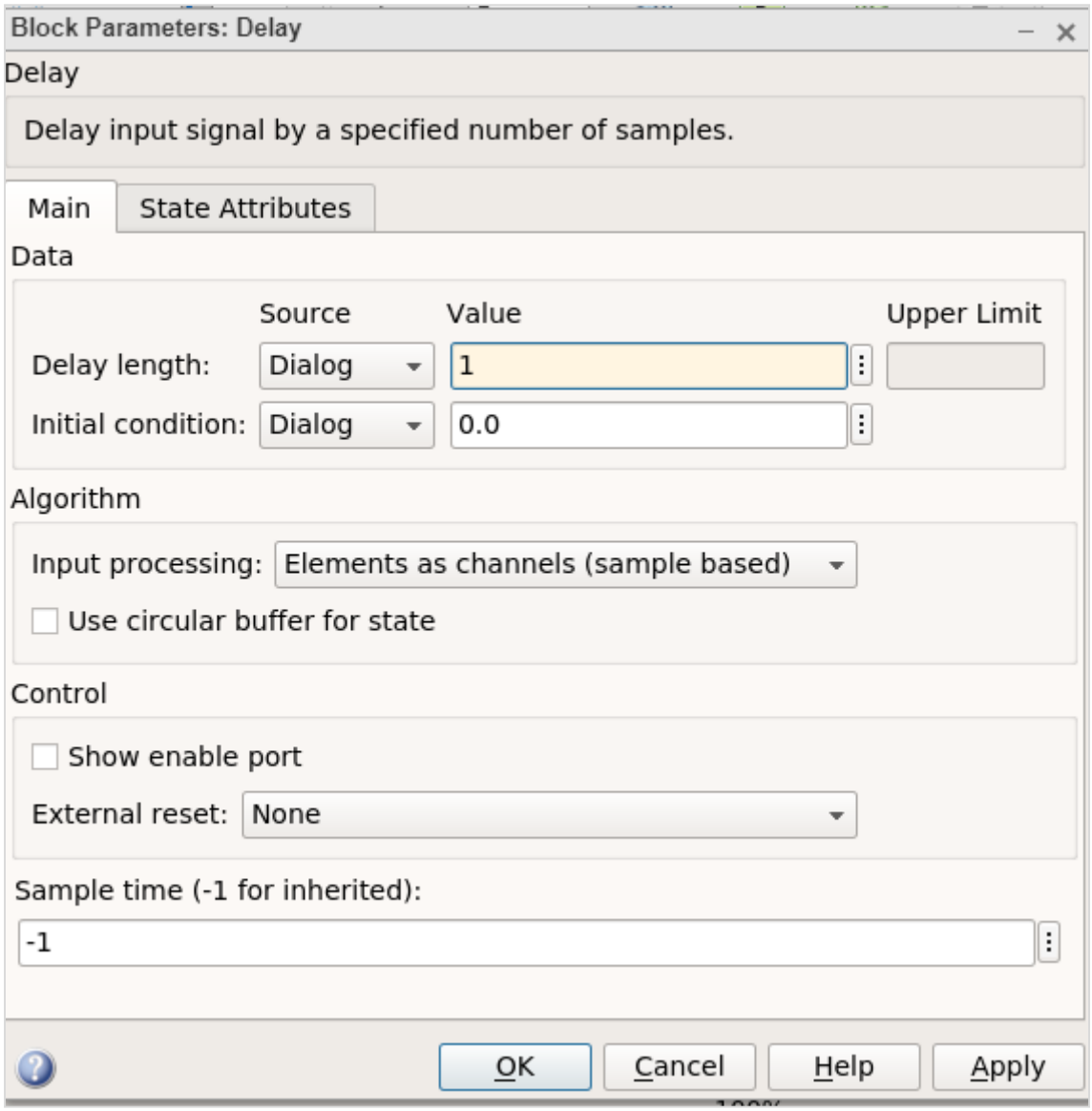


We need a sum block and a delay block as shown below:

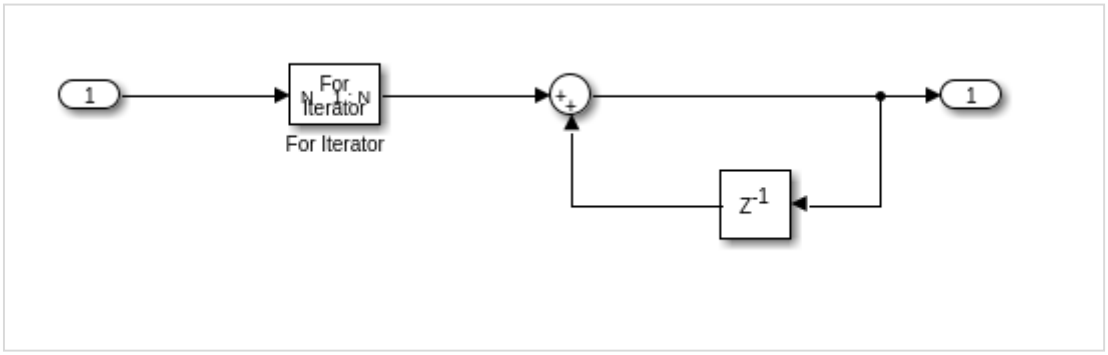


The delay block has to be flipped so that it can be added to the output. We need to give the output back to the sum block so that it can be added with the current iteration.

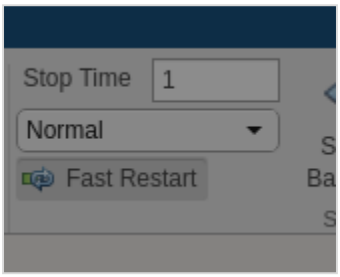
Right click on delay block and change the delay length from 2 to 1 as shown below. Click on OK to update the changes.



The final for-loop subsystem block will look as follows:

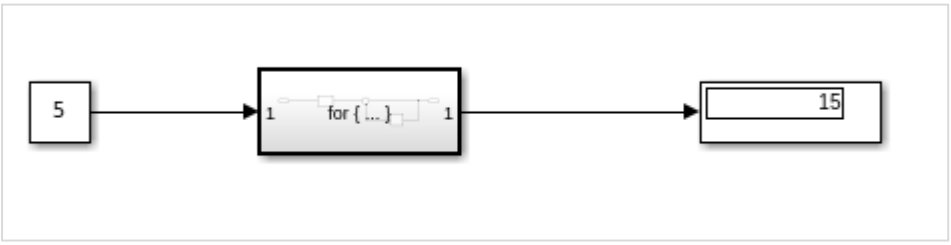


Now before you run the simulation, change the stop time to 1. We do this because we want the simulation to run only once.



Click on Run now to see the result in display block as shown below:

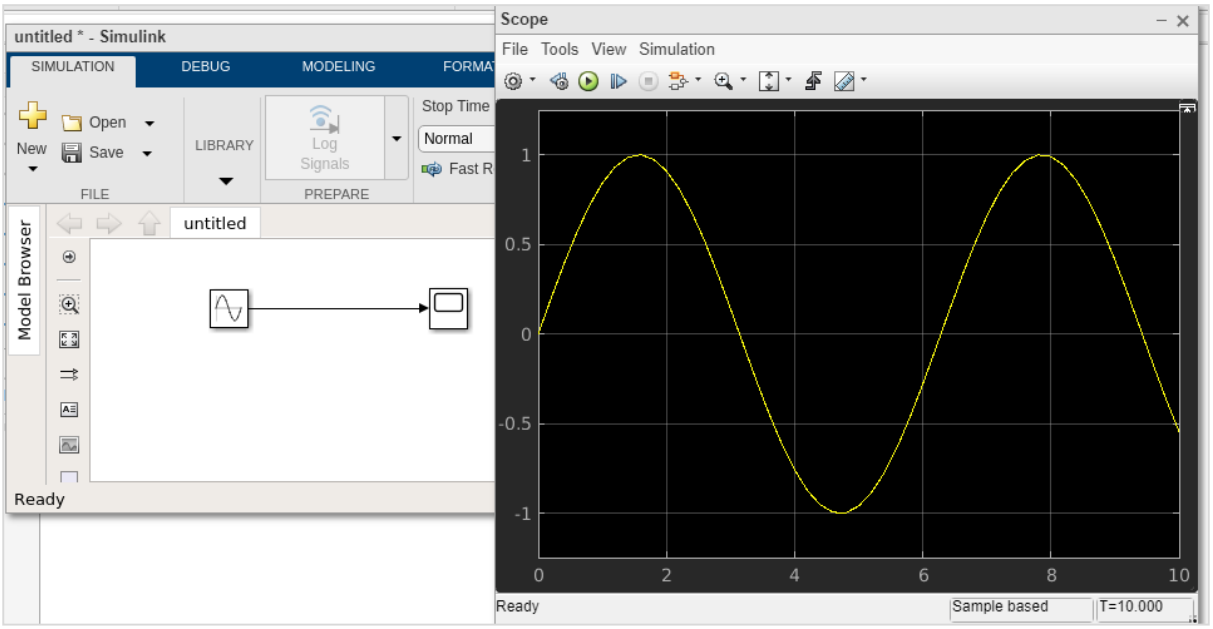
The input value is 5, so the for-loop will go from 1 to 5. Hence, the values 1+2+3+4+5 = 15 is shown in the display.



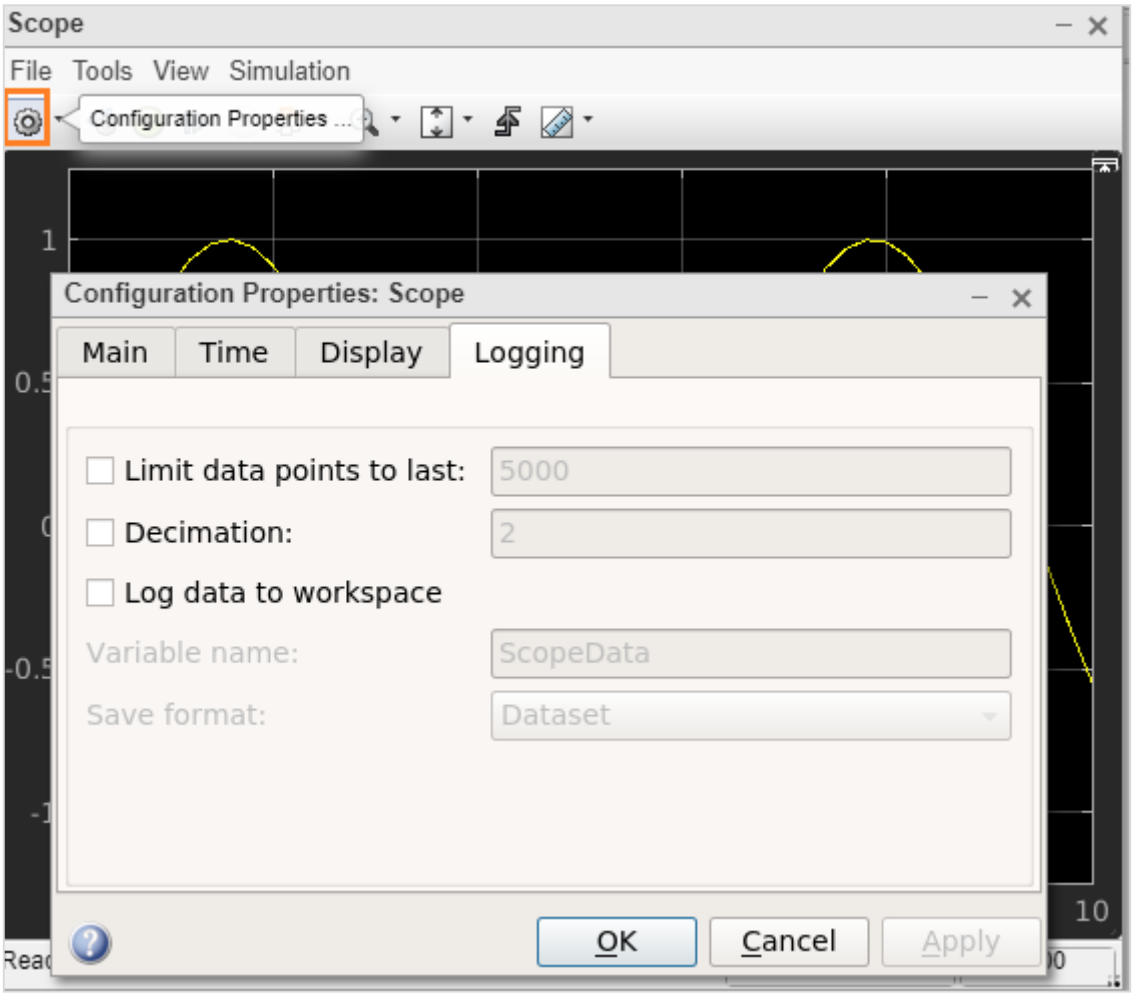
16. MATLAB Simulink — Export Data

In this chapter, we will learn how to use Simulink output results inside MATLAB. Let us try a simple model of sine wave as shown below:

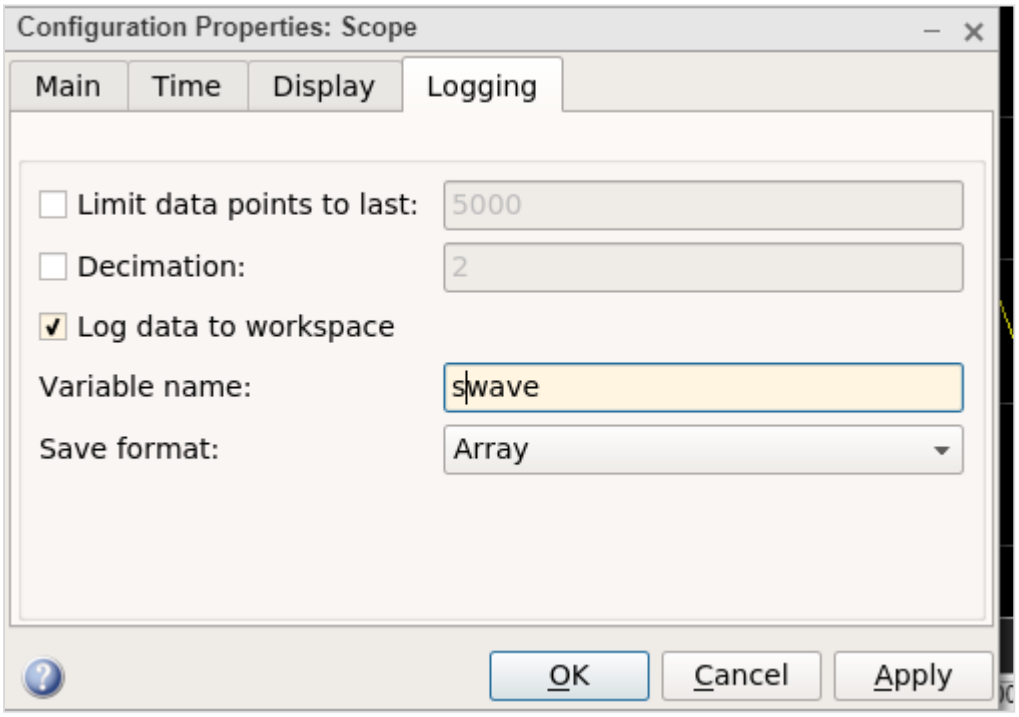
When we run the model, the scope displays the sine wave as shown below:



Now to get the data of the sine wave, go to the configuration properties and select logging tab.

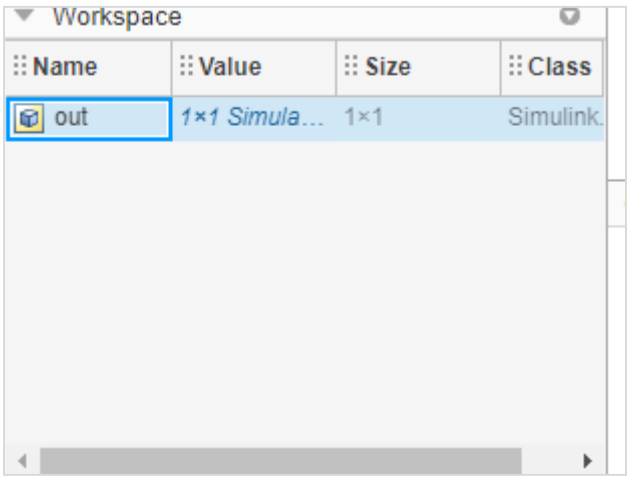


Select the log data to workspace checkbox, as shown below:

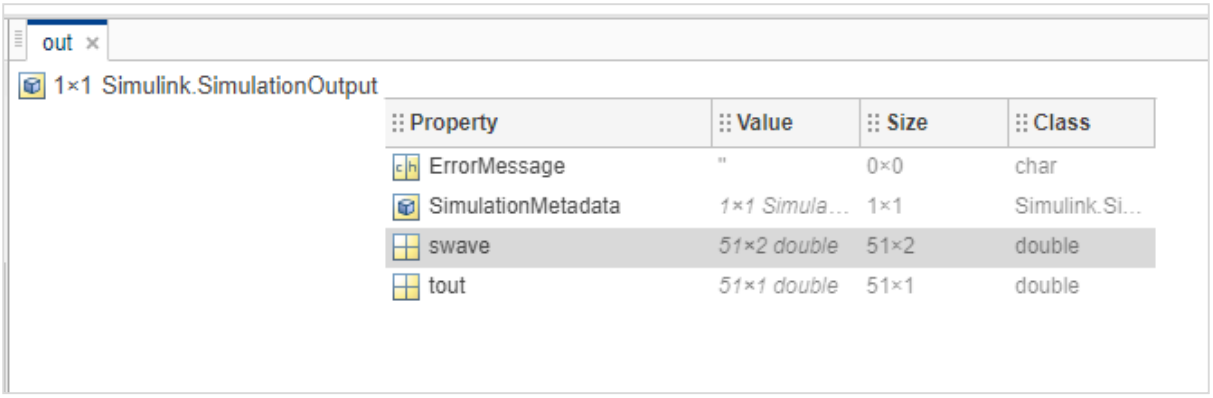


Set the variable name of your choice. Here, we have given the name as swave and the save format as Array. Click on OK button and run the simulation again.

You should see the output in the workspace.



Double click and it will show you the details of the swave variable which we saved earlier.



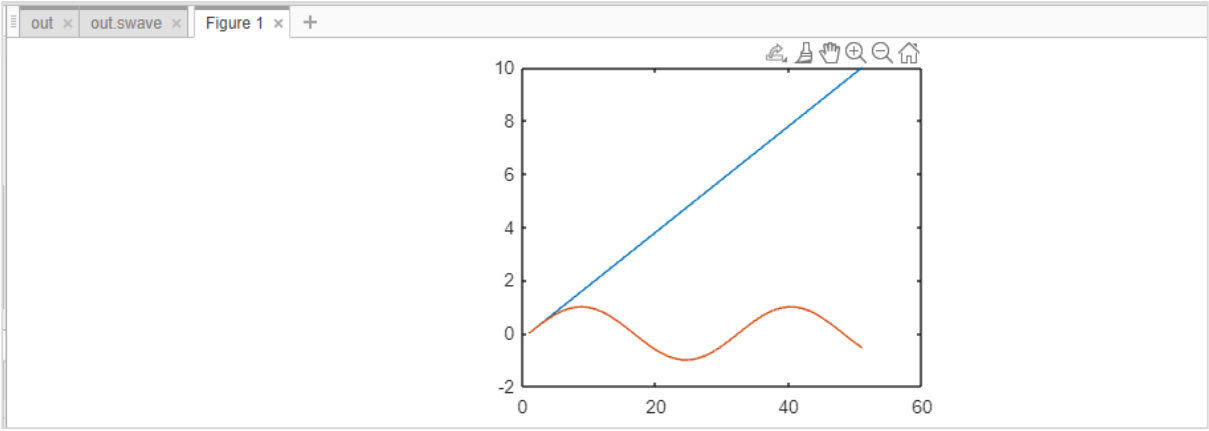
Inside command prompt type out **.swave** and it will give you the output as shown below:

```
>> out.swave

ans =

         0         0
    0.2000    0.1987
    0.4000    0.3894
    0.6000    0.5646
    0.8000    0.7174
    1.0000    0.8415
    1.2000    0.9320
    1.4000    0.9854
    1.6000    0.9996
    1.8000    0.9738
    2.0000    0.9093
    2.2000    0.8085
    2.4000    0.6755
    2.6000    0.5155
    2.8000    0.3350
    3.0000    0.1411
```

You can plot the sine wave using plot command as shown below:

```
Command Window
>> plot(out.swave)
>>
```

The graph is shown as follows:



17. MATLAB Simulink — MATLAB script

In this chapter, we will use MATLAB script to create a model. We do have a direct and easy method to create a model by just picking the blocks we need. But, writing the code to create a model can sometimes help you to automate a task as your projects get complex.

So, let us learn how to create models by using the Application Programming Interfaces (APIs) as discussed below.

We will create a very simple sine wave model. For that we need sine wave and scope blocks.

Inside the MATLAB command window, we can use API to create the model and blocks. To create a new model, the API is as follows:

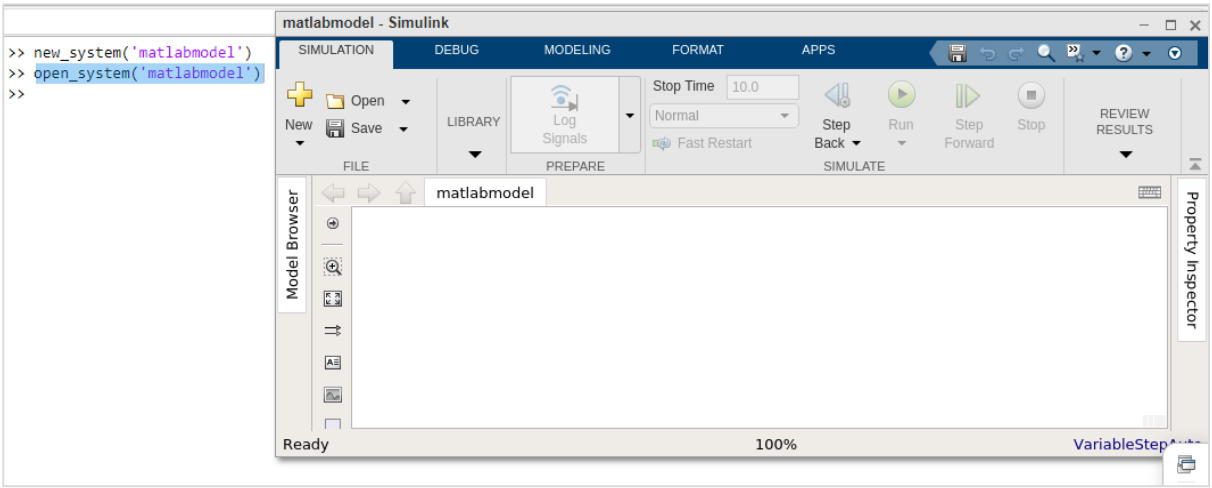
```
new_system('matlabmodel')
```

Here, matlabmodel is the name of the model. You can open the model by using open_system() with the name of the model as parameter to the function .

The command is as shown below:

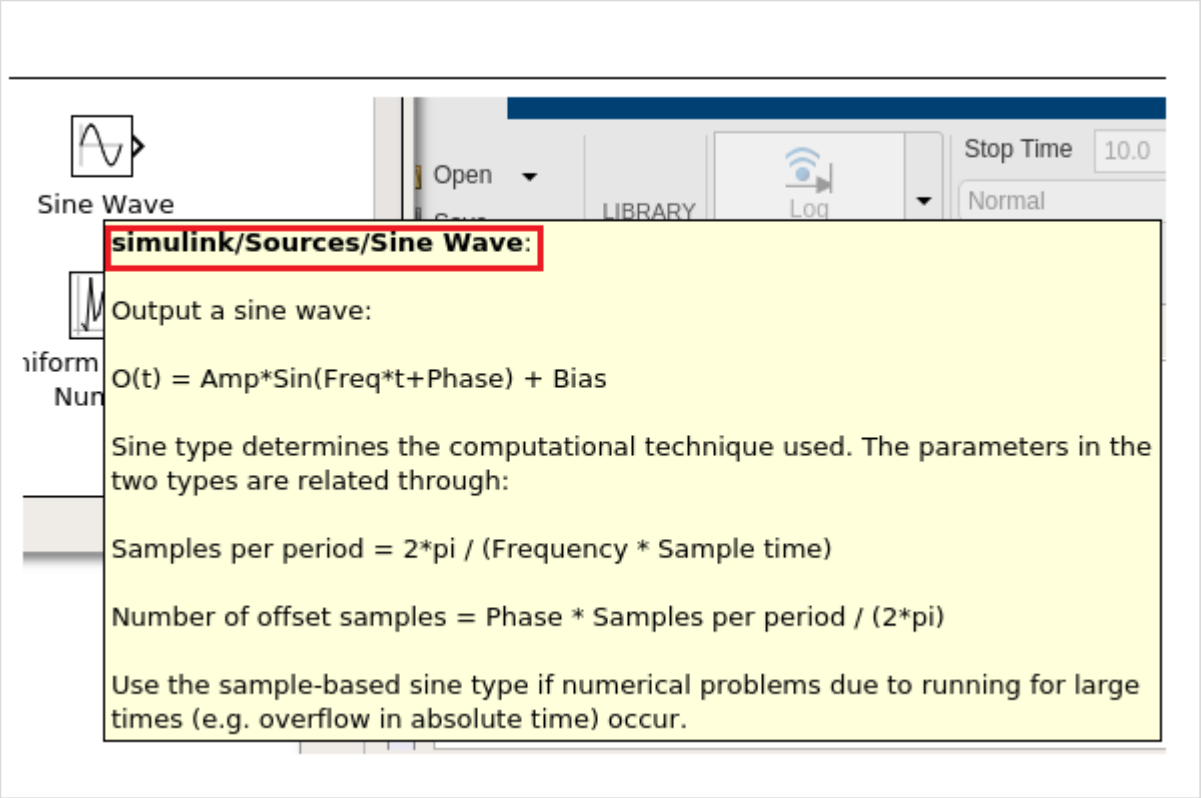
```
open_system('matlabmodel')
```

When you click enter, the model is opened as shown below:



Now, let us add sine wave block to it. The command is to add_block(source, dest).

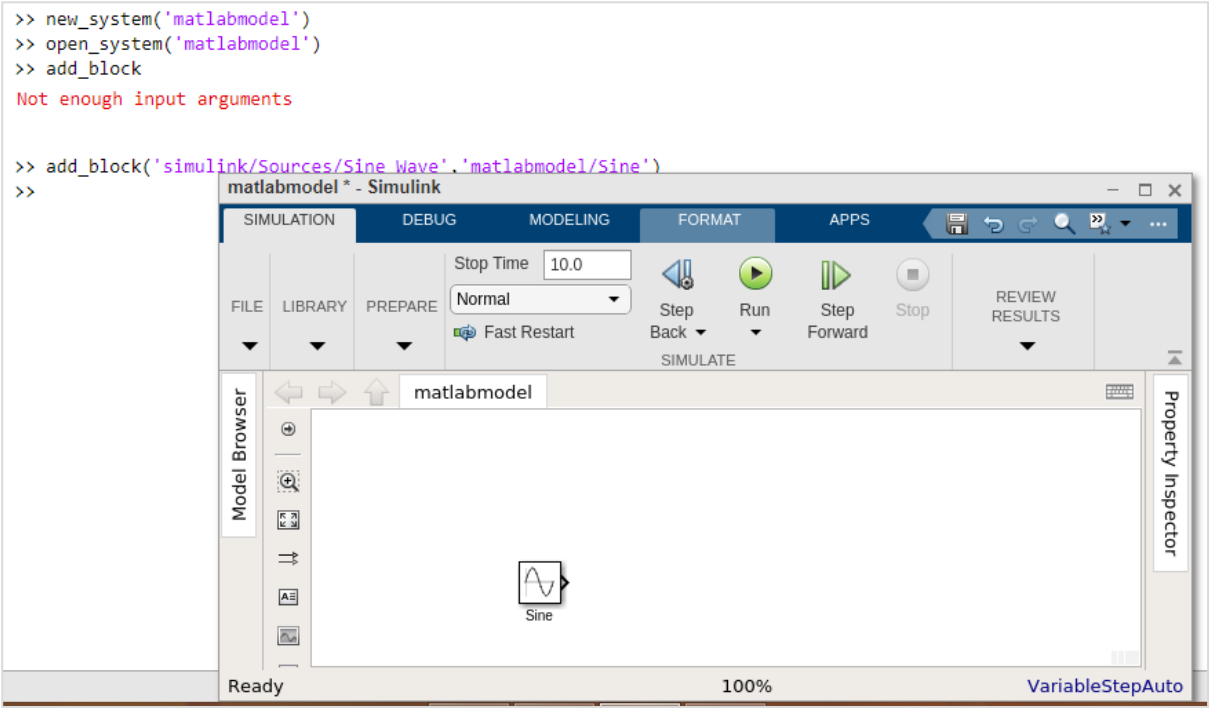
You will get the source of the block from Simulink library browser.



The highlighted code is the source of sine wave. Let us add that in the add_block as shown below:

```
add_block('simulink/Sources/Sine Wave','matlabmodel/Sine')
```

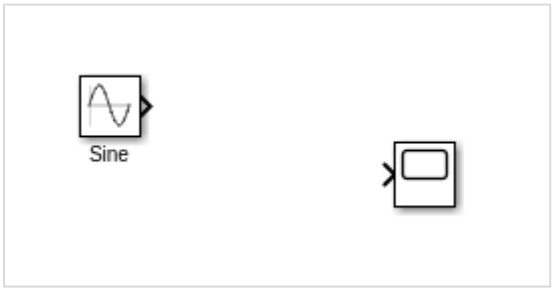
The following screen will appear on your computer:



Let us now add the scope block, as mentioned below:

```
add_block('simulink/Sinks/Scope','matlabmodel/Scope', 'Position' , [200 315 135 50])
```

The model shows the scope block as shown below:



You can make use of position inside the add_block to position the block properly.

For example,

```
add_block('simulink/Sinks/Scope', 'matlabmodel/Scope', 'Position' , [200 315 135 50]
```

Now, let us connect the line between the sine wave and scope by using the command as shown below:

```
add_line('matlabmodel', 'Sine/1', 'Scope/1');
```

For add_line, you have to pass the name of your model, followed by the block name and the input port of the blocks.

So now, we want to connect the first output port of sine wave with the first input port of scope.

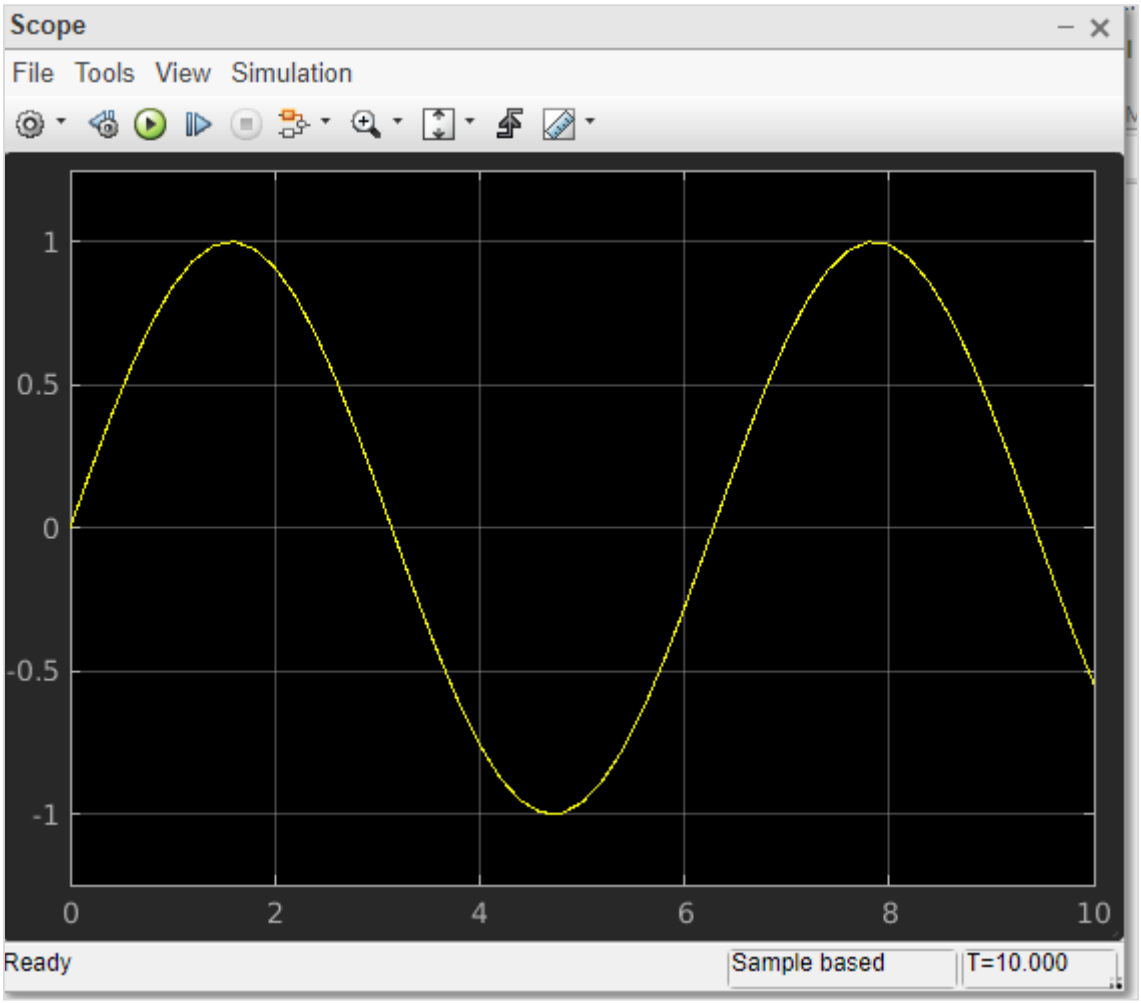
Let us now run the simulation by using the command below:

```
result = sim('matlabmodel');
```

Now to view the simulation result, run one more command as shown below:

```
open_system('matlabmodel/Scope');
```

You will get the output inside the scope as follows:



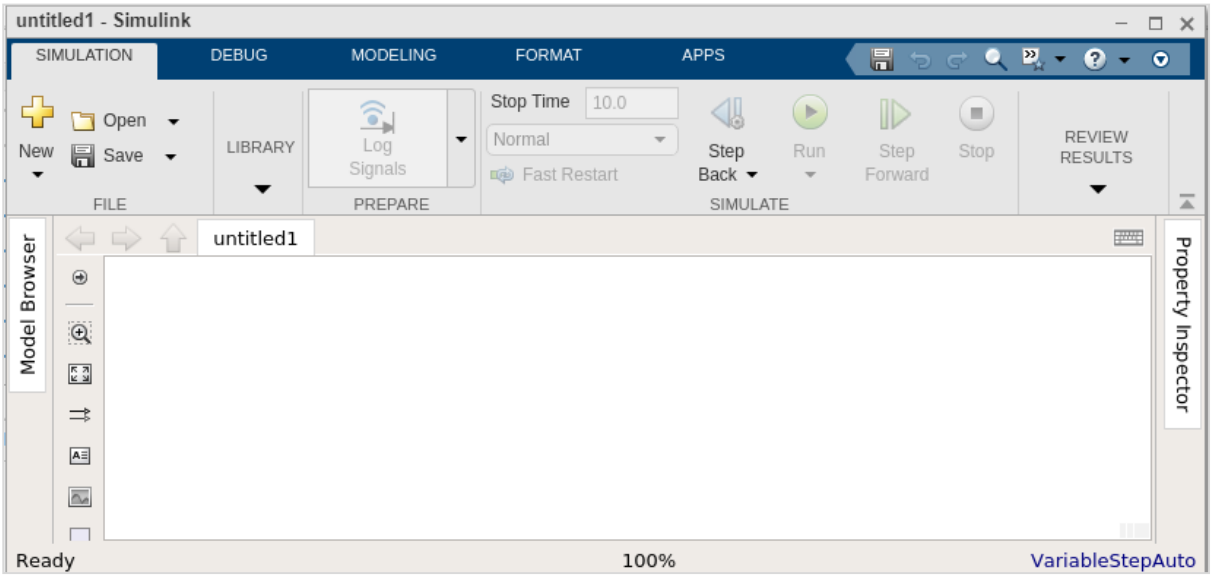
18. MATLAB Simulink — Solving Mathematical Equation

In this chapter, we will solve a simple mathematical equation by using Simulink.

The equation is given below:

```
y(t) = 2Sin(t) + 5Sin(2t) - 10
```

Let us create a model for the above equation. Open a blank model as shown below:

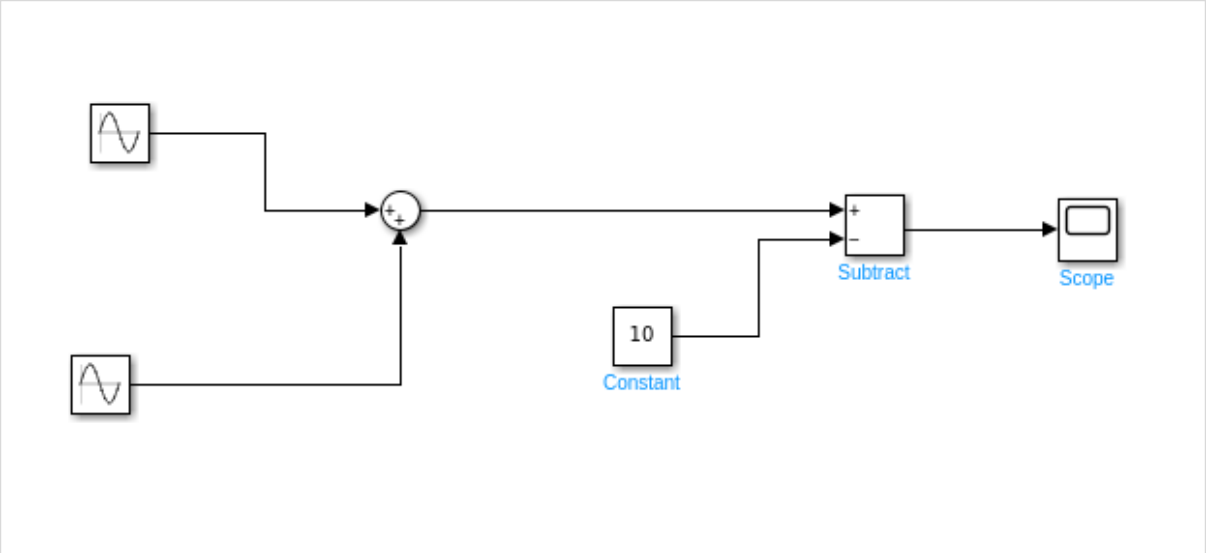


Following are the steps to solve the equation:

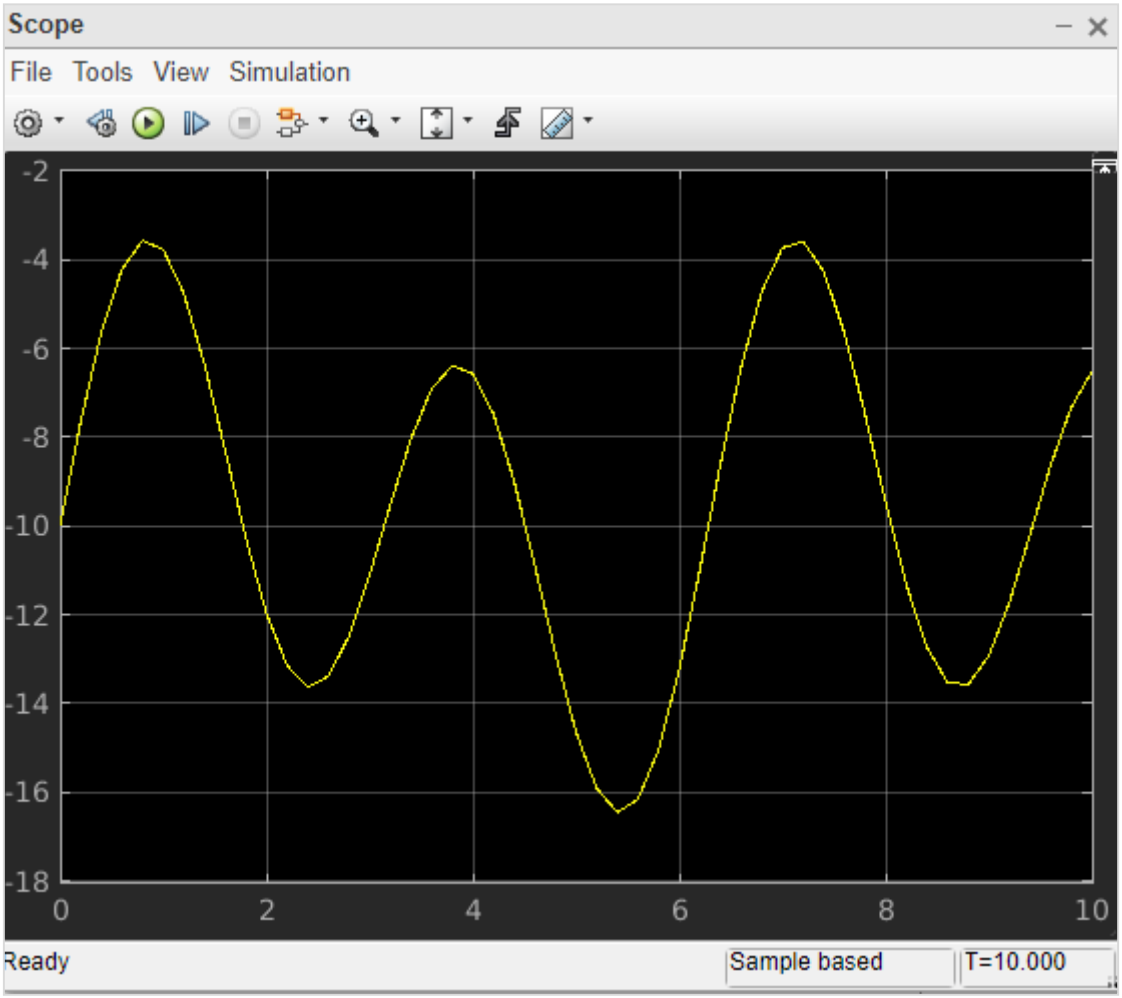
- Get a Sine wave block. Right click and select block parameters. Select sign type as time based. Change the amplitude to 2 and frequency to 1. That will be 2Sin(t).
- Get another sine wave block. Now, set the amplitude to 5 and frequency to 2 to display 5Sin(2t). Select sign type as time based.
- Now get an add block and add both sine waves.
- Get a constant. Right click and select block parameters. Change the value from 1 to 10.
- Get a subtract block where one input will come from step 3 and another from constant i.e. step 4.
- Get a scope block and connect the input from step 6 to it.

This is how the final Simulink model looks like for the equation:

```
y(t) = 2Sin(t) + 5Sin(2t) - 10
```



Click on the run button to compile. Right Click on scope block to see the signal plotted.



19. MATLAB Simulink — First Order Differential Equation

Here, we will learn how to solve the first order differential equation in Simulink.

The first order differential equation that we are trying to solve using Simulink is as follows:

```
dy/dt = 4sin2t - 10y
```

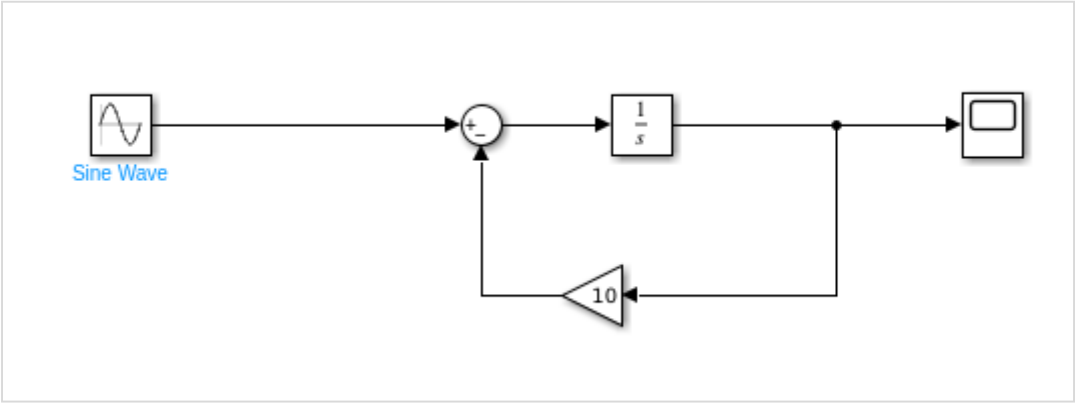
The equation can be solved by integrating dy/dt to the following:

$$y(t) = \int (4\sin 2t - 10y(t))\, dt$$

Following are the steps to build a model for above equation.

- Pick up sine wave from sources library and change the amplitude to 4 and frequency to 2. This will give us 4sin2t.
- The integrator block will be used to show dy/dt that will give output y(t).
- The gain block will represent 10y.
- The input of step 1 and 3 will be given to step2.
- We need scope block to see the output y(t). The step 4 will be connected to the scope block.

Let us see the above steps in model as shown below:



Run the block to see the following output: