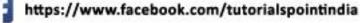
Matlab-Simulink



www.tutorialspoint.com





https://twitter.com/tutorialspoint

About the Tutorial

MATLAB (Matrix Laboratory) is a programming language developed by a computer software company MathWorks. Simulink is a simulation and model-based design environment for dynamic and embedded systems, which are integrated with MATLAB. Simulink is also developed by MathWorks. This tutorial is designed to give students fluency in MATLAB Simulink. Problem-based examples have also been given in simple and easy way to make your learning fast and effective.

Audience

This tutorial has been prepared for the beginners to help them understand basic to advanced functionality of MATLAB Simulink. After completing this tutorial you will find yourself at a moderate level of expertise in using Simulink from where you can take yourself to next levels.

Prerequisites

We assume you have a little knowledge of any computer programming and understand concepts like variables, constants, expression, statements, etc. If you have done programming in any other high-level programming language like C, C++ or Java, then it will be very much beneficial and learning MATLAB Simulink will be like a fun for you.

Copyright & Disclaimer

© Copyright 2021 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at <u>contact@tutorialspoint.com</u>



Table of Contents

	About the Tutoriali
	Audiencei
	Prerequisitesi
	Copyright & Disclaimeri
	Table of Contents ii
1.	MATLAB Simulink – Introduction1
2.	MATLAB Simulink — Environment Setup6
3.	MATLAB Simulink — Starting Simulink8
	Libraries in Simulink11
4.	MATLAB Simulink — Blocks
5.	MATLAB Simulink — Lines
6.	MATLAB Simulink — Build & Simulate Model
7.	MATLAB Simulink — Signals Processing
8.	MATLAB Simulink — Adding Delay to Signals45
9.	MATLAB Simulink — Mathematical Library50
10.	MATLAB Simulink — Build Model and Apply If-Else Logic53
11.	MATLAB Simulink — Logic Gates Model59
12.	MATLAB Simulink — Sine wave64
13.	MATLAB Simulink — MATLAB Function67
14.	MATLAB Simulink — Create Subsystem70
15.	MATLAB Simulink — for loop
16.	MATLAB Simulink — Export Data80
17.	MATLAB Simulink — MATLAB script84
18.	MATLAB Simulink — Solving Mathematical Equation88
19.	MATLAB Simulink — First Order Differential Equation90



1. MATLAB Simulink – Introduction

Simulink is a simulation and model-based design environment for dynamic and embedded systems, which are integrated with MATLAB. Simulink was developed by a computer software company MathWorks.

It is a data flow graphical programming language tool for modelling, simulating and analysing multi-domain dynamic systems. It is basically a graphical block diagramming tool with a customisable set of block libraries.

Furthermore, it allows you to incorporate MATLAB algorithms into models as well as export the simulation results into MATLAB for further analysis.

Simulink supports the following:

- System-level design.
- Simulation.
- Automatic code generation.
- Testing and verification of embedded systems.

To get started with Simulink, type simulink in the command window as shown below:

Script Live Script Commands Commands Commands	Simulink La
	SIMULINK
Name A >>	
Published (my site) ····	
🖆 firstmfile.m	
🖺 matrix.m 🚥	
MaxNumber.m ····	
🖄 mymax.m 🚥	
WORKSPACE O	
Name Value Size Class	

It will open the Simulink page as shown below:



Simulink Start Page				- ×
SIMULINK®	New Examples			
Open Recent	Search			All VQ
Learn	My Templates Simulink			Leall wore
Ep Stateflow Onramp	Blank Model	Blank Subsystem	Blank Library	Digital Filter
	Feedback Controller	Fixed-step	HDL Code Generation	Signal Processing
	Show more	Thou-step	HDE COUC GENERATION	Signal Processing
-	DSP System Toolbox			
Starting Simulink				

You can also make use of Simulink icon present in MATLAB to get started with Simulink:

	но	DME	I	PLOTS	APPS									
Ne	ew .		New	🛍 Upload 🖗 Download	🐔 Go to File	Import Data	Clear Workspace 🕶	Favorites	Clear Commands	Ŧ	Simulink	Layout	aranci	Add-Ons
_				FILE		١	/ARIABLE		CODE		SIMULIN	start Simi	IRONMENT	
\Leftrightarrow		1	ر 🗖	MATLAB Dri	ive ≯									

When you start Simulink, you are navigated to the start page as shown below:

Simulink Start Page				- ×
SIMULINK®	New Examples			
Open	Search			All 🗸 🔾
Recent	> My Templates			Learn More
Learn 📴 Simulink Onramp	✓ Simulink			
Stateflow Onramp				
	Blank Model	Blank Subsystem	Blank Library	Digital Filter
	Feedback Controller	Fixed-step	HDL Code Generation	Signal Processing
	Show more			
	> DSP System Toolbox			

Here you can create your own model, and also make use of the existing templates.

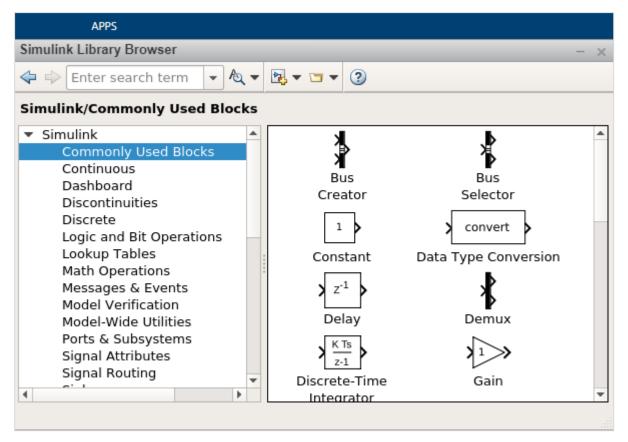


Click on the Blank Model and you will get a Simulink library browser that can be used to create your own model.

untit	led - Sin	nulink									- × .
SI	ULATION		DEBUG	MODELING	FORMAT	APPS			50	≌, • ? •	v
New	Cope Cope File	en ↓ re ▼		Log Signals PREPARE	Stop Time 10.0 Normal	Step Back - SIMULA	Run TE	Step Forward	Stop	REVIEW RESULTS	M
er			untitled								Pr
Model Browser	• • • • • • • • • • • • • • • • • • •										Property Inspector
	AE										
Read	dy				100)%			Va	ariableStepAut	:o

The screen for Blank model is as follows:

Click on Library and it will display you the Simulink library as shown below:



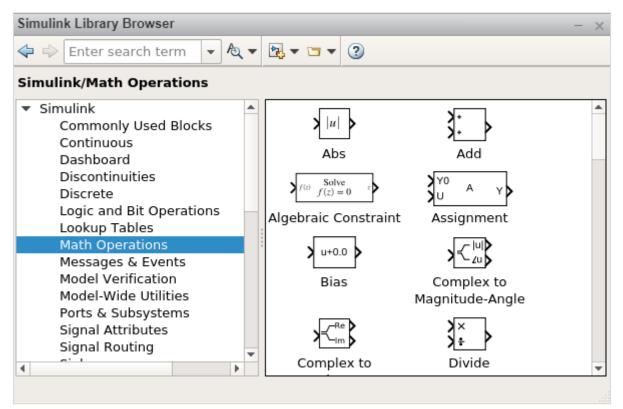
The Simulink library browsers is a collection of many libraries. It offers Commonly Used Blocks, Continuous, Dashboard, Logic and Bit Operation, Math Operations etc.



Besides that, you will get other library list like Control system toolbox, DSP system toolbox etc.

Simulink Library Browser				- ×
< 💠 Enter search term 📼 🍳 🕶	🔁 🕶 💌 🥝			
Simulink/Logic and Bit Operations	;			
String User-Defined Functions Additional Math & Discrete Quick Insert Control System Toolbox DSP System Toolbox DSP System Toolbox HDL Supp Fixed-Point Designer GPU Coder Support Package for HDL Coder Instrument Control Toolbox SimEvents Simulink 3D Animation Simulink Coder Simulink Design Verifier	Clear bit 0 Bit Clear Combinatorial Logic U < 0 & NOT U/z < 0 Detect Fall Negative	Set bit 0 Bit Set Compare To Constant U <= 0 & NOT U/z <= 0 Detect Fall Nonpositive	Bitwise AND 0xD9 Bitwise Operator Compare To Zero U>= 0 & NOT U/z>= 0 Detect Rise Nonnegative	•

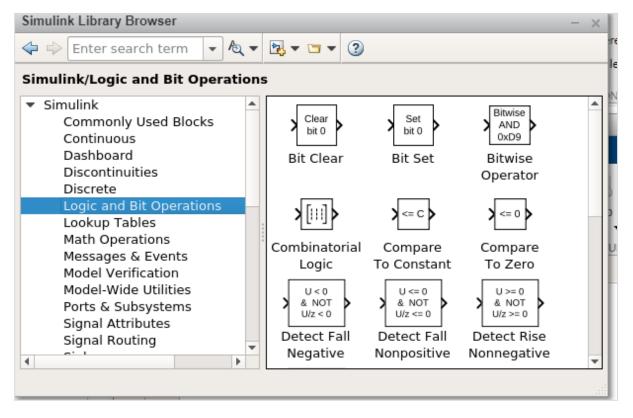
Here is an example of Math operations library list:





It has Abs, Add, Algebraic Constraint, Assignment etc. that you can make use in your model.

Given below is an example of Logic and Bit Operations:

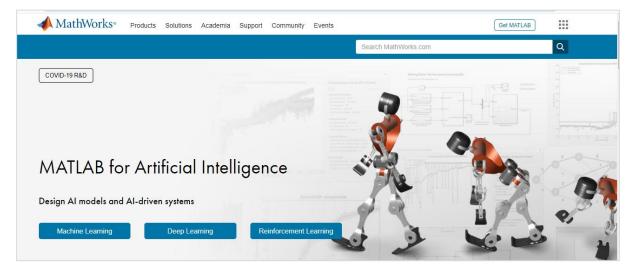




2. MATLAB Simulink — Environment Setup

MATLAB simulink is a MATLAB product and to work with it, we need to download MATLAB. The official website of matlab is <u>https://www.mathworks.com/</u>

The following page will appear on your screen:



To download MATLAB go to https://in.mathworks.com/downloads/ as shown below:

\leftarrow \rightarrow X $($ \triangleq in.mathworks.com/downloads/	☆ 🕒 🕄
✓ MathWorks [∞]	Get MATLAB
Downloads	
FAQ + Download & Install Troubleshooting	📞 Contact support
Get Latest Release	
R2021a » Learn More	Get Trial Software Try MATLAB, Simulink, and Other Products Create a trial Download an existing trial
Click here to download any MathWorks release	

MATLAB is not free to download and you need to pay for the licensed copy. Later on you can download it.



С	in.mathworks.com/downloads/message/error_page?error=no_active_licenses	☆ ⓒ
	✓ MathWorks [∞]	Get MATLAB
	Downloads	
	FAQ - Download & Install Troubleshooting	Contact support
	To download products and updates, link a license to your MathWorks Account. Visit the license center to link to a license. Alternatively, you can get a trial. Link to a License Get a Trial	

A free trial version is available for which you have to create a login for your respective account. Once an account is created, they allow you to download MATLAB and also an online version for a trial of 30 days' license.

G	et Trial Software
	MATLAD Circulials and Other
-	/ MATLAB, Simulink, and Other oducts
Pro	

Once you are done with the creating a login from their website, download MATLAB and install on your system. Then, start MATLAB or you can also make use of their online version.

Simulink comes in-built with MATLAB. Once you install MATLAB, you will get Simulink as shown below:

но	ME	F	PLOTS		APPS													umentation 🔍
New Script	New Live Script	New V		Dow:		Go to File Find Files	Import Data	Workspace	 New Varia Open Varia Open Varia Clear Worl VARIABLE 	able 🔻	Favorites	Clear Commands ▼ CODE	Simulink SIMULINK	Layout	 Preferences Parallel ENVIRONMENT 	Add-Ons	Community Feedback	
\Leftrightarrow	la 🔊	ر 🖨	➤ MATL	AB Drive	>													
	ent Folder				0	>>												
▼ Work	-				0	1												
:: Name	:: Va	lue	:: Si	ze	:: Class]												



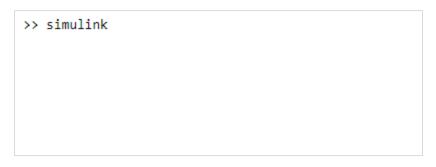
3. MATLAB Simulink — Starting Simulink

In this chapter, we will understand about using Simulink to build models.

Here is a MATLAB display:

нс	ME	P	LOTS		APPS											Search Doc	umentation Q
New Script	New Live Script	New		街 Dow		🖷 Go to F 🗔 Find Fil	ort	New Vari Open Va Zopen Va Zopen Va Zopen Va	riable 🔻	Favorite	s Clear Commands v CODE	Simulink	 Preferences Parallel ENVIRONMENT 	Add-Ons •	? Help ▼	Community Feedback Learn MATLAB RESOURCES	
<p td="" ↔<=""><td>a 3</td><td>67</td><td></td><td></td><td>• •</td><td></td><td></td><td></td><td></td><td></td><td>0000</td><td>Sindentic</td><td>Erenconner</td><td></td><td></td><td>Resources</td><td></td></p>	a 3	67			• •						0000	Sindentic	Erenconner			Resources	
▼ Curr	ent Folder					• >>											
Name	•																
ی 🗖 ا	ession					^											
D P	ublished (my site															
渣 a	ddmodel.s	slx															
🚹 fi	rstmfile.m																
칠 fo	orloop.slx																
칠 if	lese.slx																
1 🚹 п	natrix.m																
🖄 N	1axNumbe	er.m				-											
• Worl	space					0											
Name	:: Va	alue	:: Si	ze	:: Cla	iss											
4						•											

You can start Simulink by using simulink command in the MATLAB command window as shown below:



Click on enter to open the Simulink startup page as shown below:



Simulink Start Page				- ×
SIMULINK°	New Examples			
🛅 Open	Search			All 🗸 🔾
Recent	> My Templates			Learn More
 MATLAB Drive/testfor.slx MATLAB Drive/signalprocessin 	✓ Simulink			
MATLAB Drive/iflese.slx				
MATLAB Drive/addmodel.slx	$0 \longrightarrow 0$			
Learn	2		- Call -	
Simulink Onramp	Blank Model	Blank Subsystem	Blank Library	Project from Git
🕞 Stateflow Onramp				
	Project from SVN	Digital Filter	Feedback Controller	Fixed-step
	Show more			
	> DSP System Toolbox			

You can also open Simulink from MATLAB interface directly by clicking on Simulink icon as shown below:

HOME		PLOT	TS		APPS							Searc	h Documentation	Q
New Ne Script Live S	w Ne	w Op		10 - E - 10 - E		Save Workspace VARIABLE	₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽	Favorites	Clear Commands ▼ CODE	Simulink SIMULINK	·• "	Add-Ons	Alignment Alignme	
💠 🔶 🔁 🤅	2	/ > 1	MATL	AB Driv	e ≯									
Current F Name Sessie Publis addmo firstmf forloop iflese: Workspace	on hed (my s odel.slx ile.m o.slx slx	site)		*	simuli	nk								
:: Name	:: Value		:: Siz	ze										
4				•										

When you click on the Simulink icon, it will take you to a Simulink startup page, as shown below:



Simulink Start Page				- ×
SIMULINK®	New Examples			
🛅 Open	Search			All 🗸 🔾
Recent	> My Templates			Learn More
MATLAB Drive/testfor.slx	✓ Simulink			
 /MATLAB Drive/signalprocessin /MATLAB Drive/iflese.slx 				
MATLAB Drive/addmodel.six				
Learn	2		Egiteder Ight	
Simulink Onramp	Blank Model	Blank Subsystem	Blank Library	Project from Git
🕞 Stateflow Onramp				
	Project from SVN	Digital Filter	Feedback Controller	Fixed-step
	Show more			
	> DSP System Toolbox			

The startup page has a blank model, subsystem, library to start the model from scratch. There are also some built-in templates that can help the users to start with.

To create a model, the user can click on blank model and it will display a page as shown below:

untit	led - S	Simuli	ink							- 🗆 🗙
SIN	IULAT	ION		DEBUG	MODELING	FORMAT		APPS		
New		Open Save LE		Library Browser LIBRARY	Log Signals	Add Viewer PREPARE	Signal Table	-	Stop Time 10.0 Normal • Step Run Back • • SIMULATE	Step Forward
e				untitled						
Model Browser										

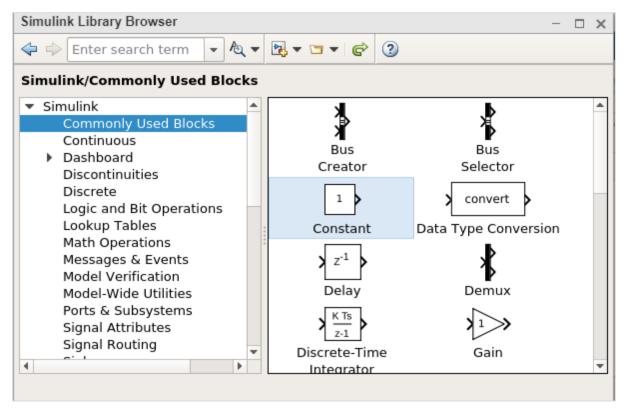
Click on Save to save your model. The blocks to build your model are available inside the Simulink library browser.

Click on library browser as shown below:



untit	tled - Simulink								-	□ × □
SI	MULATION	DEBUG	MODELING	FORMAT	APPS		1 5	c 🔍	₽, • ? •	•
New	☐ Open ↓ ∏ Save ↓ FILE	LIBRARY	Log Signals	Stop Time 10.0 Normal	▼ Step Back ▼ SIMULA	Run TE	Step Forward	Stop	REVIEW RESULTS	I
Model Browser		Library Browser								Property Inspector
Rea	dy			10	0%				VariableSter	Auto

The library browser has a list of all types of libraries with different blocks as shown below:



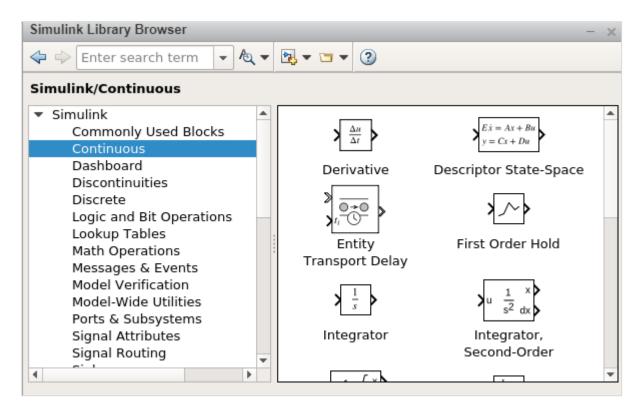
Libraries in Simulink

Let us understand some of the commonly used libraries in Simulink.

Continuous

A continuous blocks library gives you blocks related to derivatives and integrations. The list of blocks are as follows:





Dashboard

With Dashboard, you will get controls and indicator blocks that help to interact with simulations. The following screen will appear on your computer:

Simulink Library Browser	- ×
🗢 🗇 Enter search term 📼 🔩 🕶	№ ▼ 🖼 ▼ (2)
Simulink/Dashboard	
▼ Simulink	
Commonly Used Blocks	fx >>
Continuous Dashboard	
Discontinuities	Callback Button
Discrete	
Logic and Bit Operations	⊻ —
Lookup Tables	Check Box
Math Operations	
Messages & Events Model Verification	
Model-Wide Utilities	
Ports & Subsystems	Circular Gauge
Signal Attributes	
Signal Routing	
4	·

Discontinuities

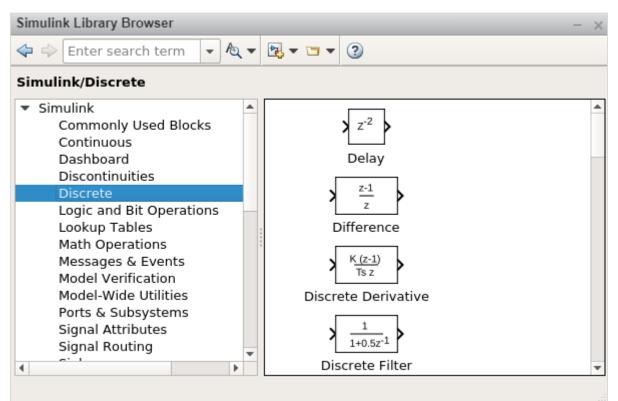


Simulink Library Browser < i>Enter search term - 🙇 - 🔁 - 🗂 -(?) Simulink/Discontinuities Simulink * . Commonly Used Blocks Continuous Backlash Dashboard Discontinuities Discrete Logic and Bit Operations Coulomb & Lookup Tables Viscous Friction Math Operations Messages & Events Model Verification Dead Zone Model-Wide Utilities Ports & Subsystems Signal Attributes Signal Routing Dead Zone ۹. . Dunami

Here, you will get a list of discontinuous functions blocks as displayed below:

Discrete

Here, you will get time relation function blocks as shown below:



Logic and Bit Operations

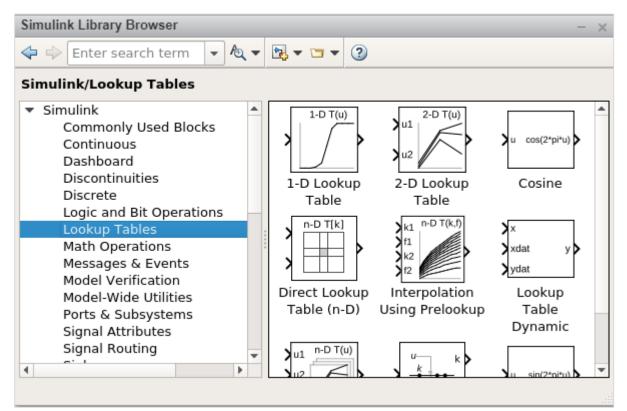


Simulink Library Browser - Az - 🔁 - 🗂 -Enter search term 2 Simulink/Logic and Bit Operations Simulink 4 Bitwise Clear Set Commonly Used Blocks AND bit 0 bit 0 0xD9 Continuous Dashboard Bit Clear Bitwise Bit Set Discontinuities Operator Discrete Logic and Bit Operations 1111 <= C <= 0 Lookup Tables Math Operations Combinatorial Compare Compare Messages & Events Logic To Constant To Zero Model Verification Model-Wide Utilities U < 0 U <= 0 U >= 0 & NOT & NOT & NOT Ports & Subsystems U/z < 0 U/z <= 0 U/z >= 0 Signal Attributes Detect Fall Detect Fall Detect Rise Signal Routing Negative Nonpositive Nonnegative ۰. Þ

In this category, you will get all logical and relational type blocks as displayed below:

Lookup Tables

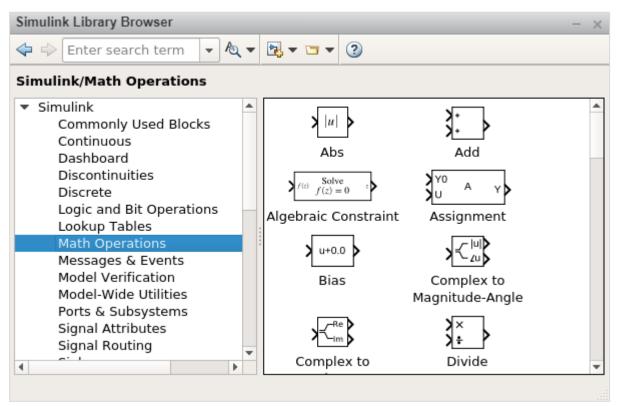
You will all the sine, cosine function blocks as shown below:



Math Operations

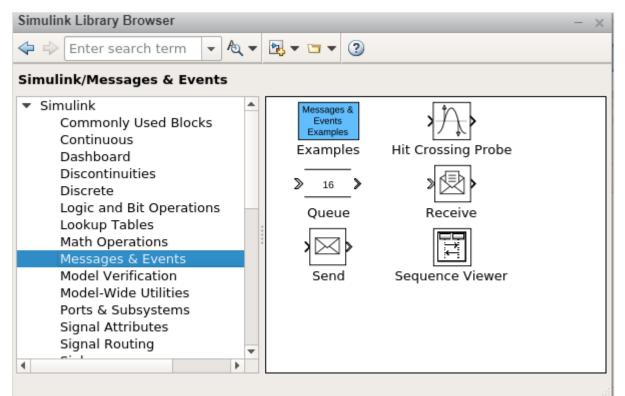


All mathematical operations like Add, Absolute, divide, subtract are available. The list is as follows:



Messages and Events

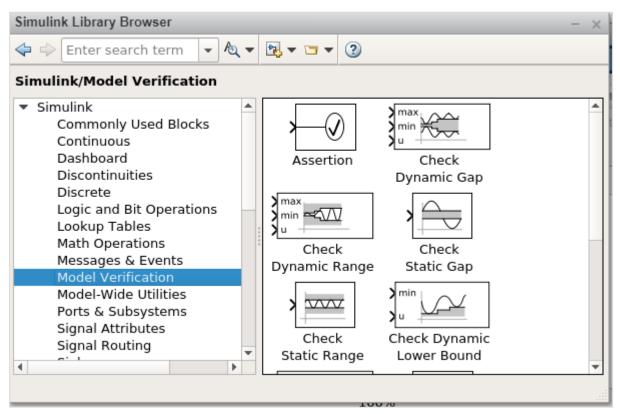
This block has all the message/communication related functions as shown below:



Model Verification



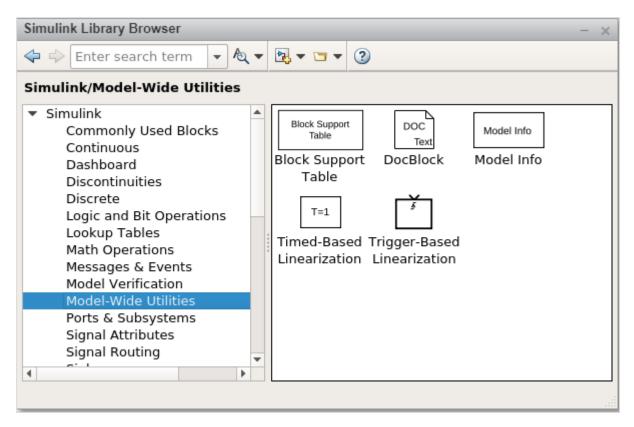
The blocks present here helps to self-verify models, such as Check Input Resolution. The following screen will appear on your computer:



Model-Wide Utilities

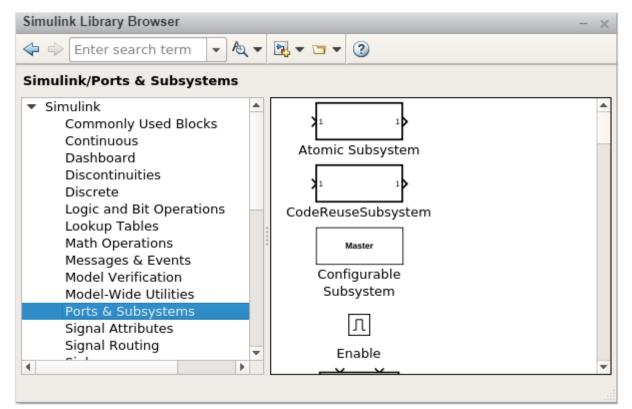
This gives you blocks like Model info, Block Support Table etc. The following screen will appear on your computer:





Ports and Subsystems

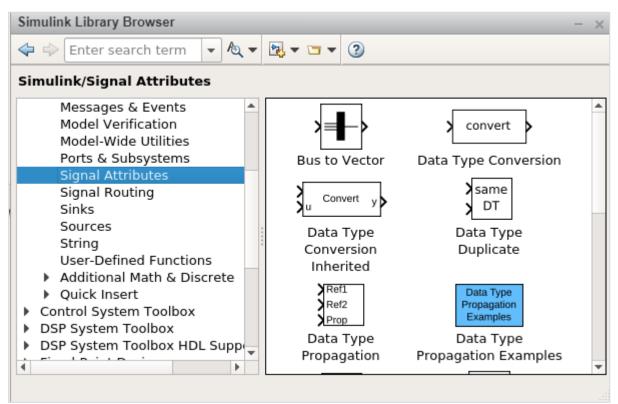
You will get blocks like a subsystem, switch case, enable etc. The list is displayed below:



Signal Attributes



Modify the signal attribute blocks such as Data Type Conversion. The following screen will appear on your computer:



Signal Routing

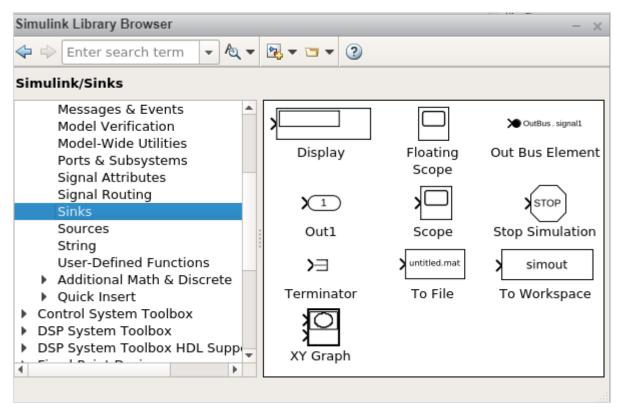
The blocks in this category is used to route signal blocks such as bus creator, switch etc. The following screen will appear on your computer:

Simulink Library Browser	- ×
< 🗣 Enter search term 💌 🔩 🕶	№, ?
Simulink/Signal Routing	
Messages & Events Model Verification	InBus.signal1 → OutBus.signal1
Model-Wide Utilities Ports & Subsystems	Bus Element In Bus Element Out
Signal Attributes	Bus
Signal Routing	:= signal1
Sinks	Bus Bus
Sources	Assignment Creator
String	
User-Defined Functions	
Additional Math & Discrete	
Quick Insert	Bus Data Store
Control System Toolbox	Selector Memory
DSP System Toolbox	(global)
DSP System Toolbox HDL Supp	
4	Data Store Data Store



Sinks

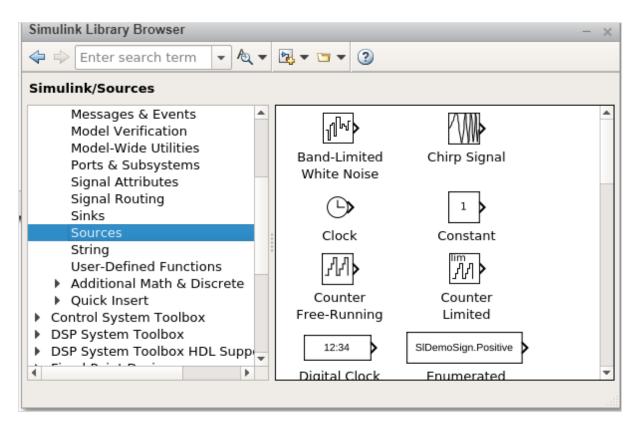
The blocks in this category help to display or export signal data blocks such as Scope and To Workspace. The following screen will appear on your computer:



Sources

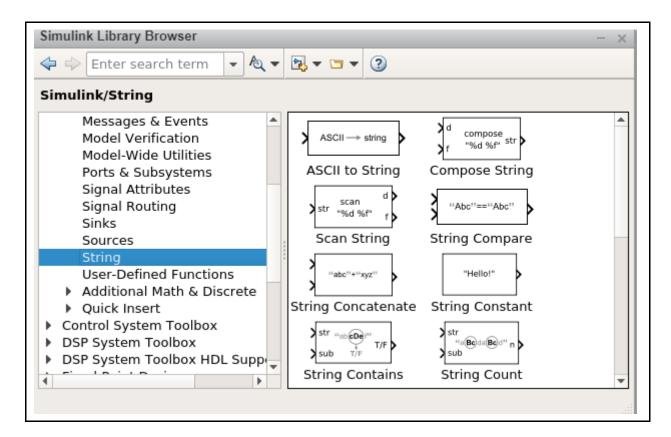
It helps to generate or import data blocks. For example, sine wave. The following screen will appear on your computer:





String

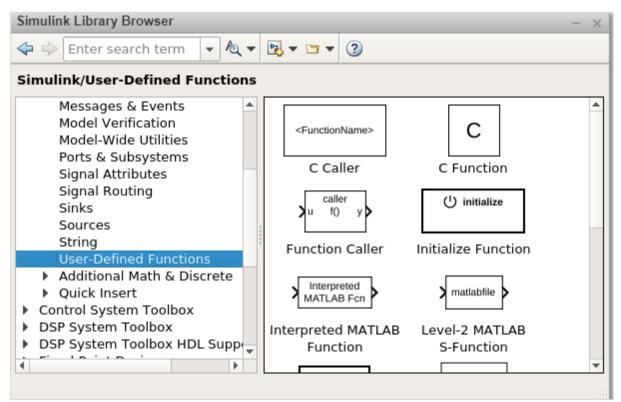
This category has string related blocks as shown below:



User Defined functions



Custom function blocks such as MATLAB Function, MATLAB System, Simulink Function, and Initialize Function. The following screen will appear on your computer:





In this chapter, we will learn about one of the basic elements in Simulink. These are termed as blocks.

Blocks in Simulink helps to create models. You can make use of a Simulink library browser that has different types of blocks for creating a model.

untit	tled - S	imulink								_	□ ×
SI	MULATIO	DN	DEBUG	MODELING	FORMAT	APPS			c Q	≌, - ? -	
New		Open ← Save ←	LIBRARY	Log Signals PREPARE	Stop Time 10.0 Normal	Step Back -	Run	Step Forward	Stop	REVIEW RESULTS	I.
er		-	untitled								P
Model Browser											Property Inspector
Rea	dy				10	00%				VariableSte	pAuto

First, open a blank model. The display will be as shown below:

You can save your model by clicking on the Save button. Hence, your changes will be saved successfully. Now, open the library browser to get the blocks into your model canvas.

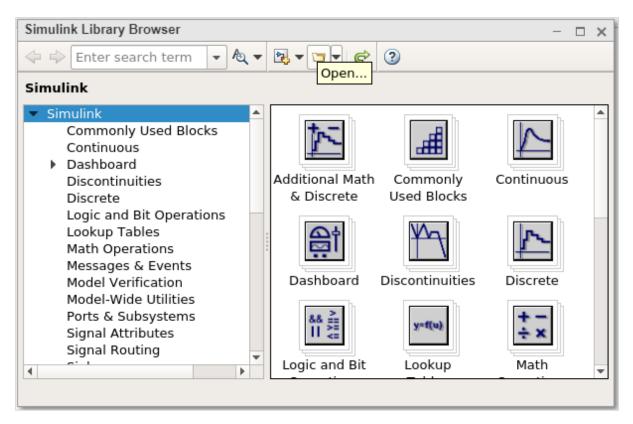
The two ways to select the blocks are as follows:

- Using Simulink browser library.
- Searching for block inside model canvas.

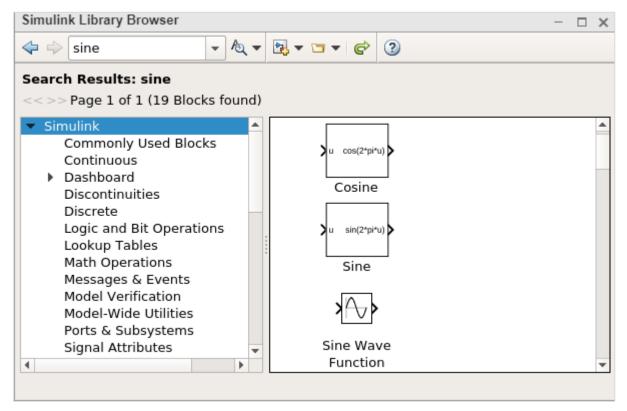
Simulink browser library

Open the Simulink library browser as shown below:





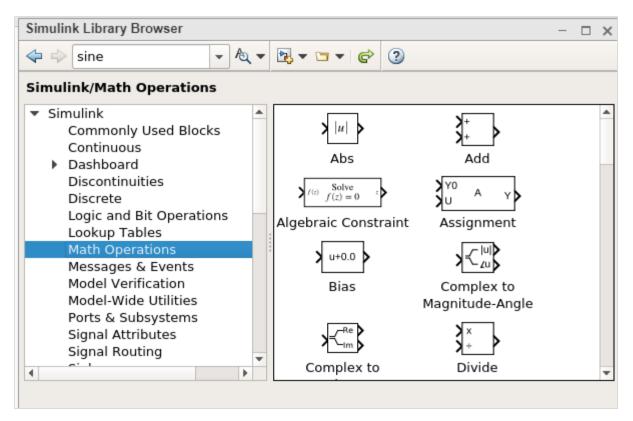
If you looking for a specific block and don't know which library, you can search for it inside the search block which is available as shown below:



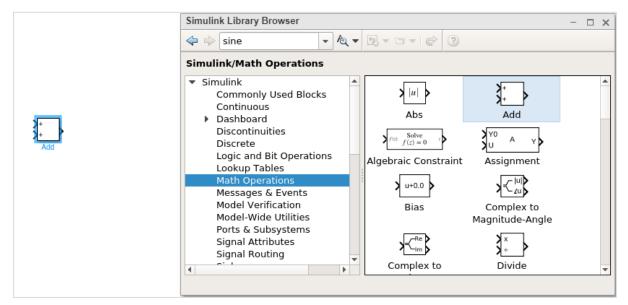
Here, we got all the blocks related to Sine. You can also go inside the library and pick your block.

Add block, Product block etc. The display will be as shown below:





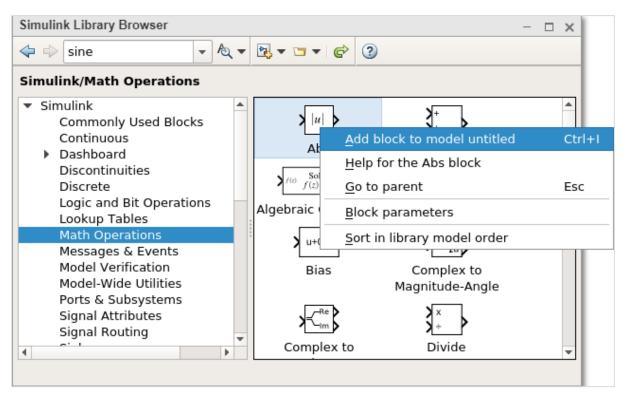
To bring the block inside your model, you can select the block and drag it inside your model, as shown below:



Another way is to right click the block and add to your current model.

An example for the same is given below:





We have not saved our model. Hence, it comes as untitled. Now, you can add to the model untitled. The block will be seen inside your mode.

Searching for block inside model canvas

Another way to add blocks to your model is to click inside the model and type the name of the block. It will search in the library browser and list all the model as per what you have typed.

An example for the same is given below:

<u>ସ</u>	sine
Sine	4
Simulink/Lookup Tables	
Sine Wave	
Simulink/Sources	
Sine Wave	
DSP System Toolbox HDL Support/Sources	
Sine Wave	
DSP System Toolbox/Sources	
Sine Lookup	
Simulink/Quick Insert/Lookup Tables	
Sine Wave Function	
Simulink	
Sine HDL Optimized	
HDL Coder/Lookup Tables	
Cosine <sine and="" cosine=""></sine>	
Simulink/Lookup Tables	
Blocks	Actions (Ctrl+.)



We have typed Sine and it displays all the blocks related to sine.



5. MATLAB Simulink — Lines

In the previous chapter, we learnt about the different types of blocks which are available with Simulink library. In this chapter, we are going to understand about lines.

Lines are used to connect the blocks with an arrow. Each block will have its own input and output connector. The communication between the blocks will take place with the help of lines.

Let us understand the same with an example. Select a blank model from Simulink page as shown below:

Simulink Start Page			
SIMULINK®	New	Examples	
Copen Recent Learn Simulink Onramp Stateflow Onramp	Search My Templates Simulink Blank Model	1 pQ = 0.540 bar 2 →	Blank Subsystem
	Feedback Controller	To any marked a	Fixed-step
	Show more		

It will open a blank model workspace as shown below:



unt	tled - S	Simulink												- ×
S	MULATI	ON	DEBUG	MODELING	FORMAT	APP	s					be	- 🚬 🕶 🕐 📼	•
Nev		Open ↓ Save ↓	Library Browser	Log Signals	Add Viewer	Signal Table	Stop Time Normal	•	Step Back 🗸	Nun	Step Forward	Stop	Data Inspector	•
	FIL		LIBRARY		PREPARE				SIMULA	TE			REVIEW RESUL	ts 🔳
Model Browser			untitled											Property Inspector
Rea	« dy						100%					,	VariableStepAu	ito

Click on Simulink Library browser to drag some blocks in the model workspace.

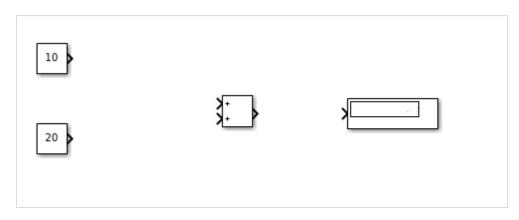
	IULAT	Simulink	DEBUG	MODELING	FORMAT	APF	s				- ²⁰ - 0	-
Jin	IULAI		DEBOG	MODELING	FORMAT						े 🗟 🖌 🕈 🕄 🗸	~
ew	_	Open ▼ Save ▼	Library Browser	Log Signals	Add Viewer	Signal Table	Stop Time 10.0 Normal • Fast Restart	Step Back 🗸	Run	Step Sto Forward		•
	F	ILE	LIBRARY		PREPARE			SIMULAT	ΓE		REVIEW RESULT	TS
	$\langle \neg$		untitled									
	۲											
	Đ,											
	K 7 K 2											
	⇒											
	AE											
	\sim											
	01											
	BE .											

Consider that in the model, we want to add two given numbers. So let us pick the Add block, the display block and the constant block.

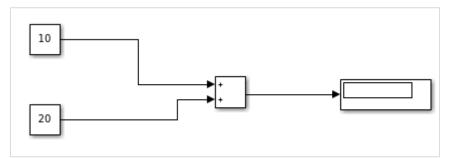
The constant block has one output connector, the Add block has two input connectors and the display block has one input connector respectively. You can drag the link from one output to another input as shown below.

Here, we have two constants with values 10 and 20. They are connected to the add block with lines. The add block is connected to display with a line.

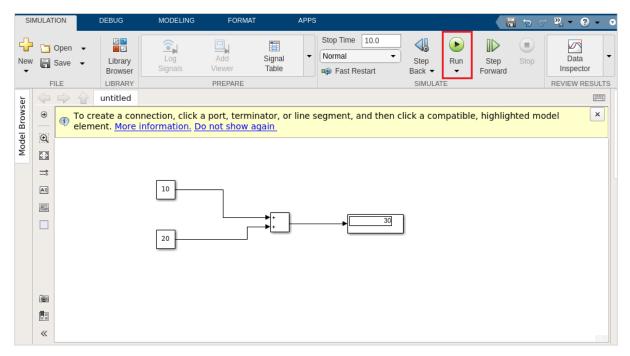




When the lines are connected, the display is as follows:



Now click on Run to see the result in the display block. It will add 10 + 20 to give the result as 30 in the display block.





6. MATLAB Simulink — Build & Simulate Model

We have seen the Simulink library browser and the blocks available in the library list. In this chapter, we are going to use the blocks to build a simple sine wave model.

Simulink Start Page				- ×
SIMULINK®	New Examples			
C Open	Search			All 🗸 🔾
Recent	> My Templates			Learn More
Learn P Simulink Onramp Stateflow Onramp	Simulink Create Model Blank Model V	Blank Subsystem	Blank Library	Digital Filter
	Feedback Controller	Fixed-step	HDL Code Generation	Signal Processing

Open Simulink and click on blank model as shown below:

The blank model will open a blank popup window as shown below:

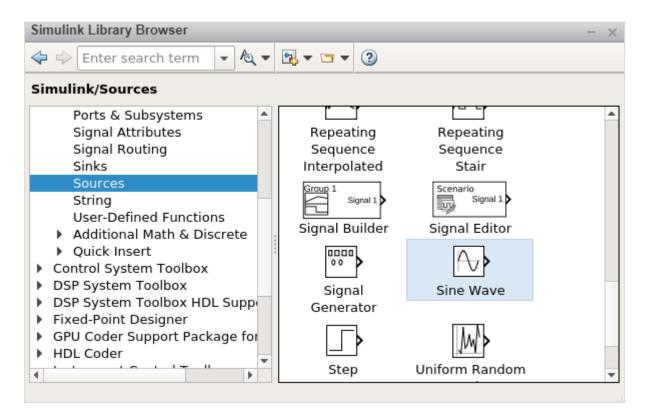
untit	led - Simulink									- × .
SIMULATION		DEBUG	MODELING	FORMAT	APPS	APPS		5 ¢	₽, - ? -	
New	☐ Open ↓ ☐ Save ↓ FILE	LIBRARY	Log Signals PREPARE	Stop Time 10.0 Normal Imp Fast Restart	Step Back V	Run TE	Step Forward	Stop	REVIEW RESULTS	<u>ا</u> ا
er	$\langle \Rightarrow \Rightarrow \uparrow$	untitled								P
Model Browser										Property Inspector
Read	ly			100%)			Va	ariableStepAut	to 🔡

Now, open the Simulink Library browser so that we can select the blocks.



untit		Simulink						- ×
SI	MULAT	ION	DEBUG	MODELING	FORMAT	APPS	6	Simulink Library Browser – ×
New		Open 👻 Save 👻 LE		Log Signals	Stop Time 10.0 Normal	Step Back V	Step Forward	← ← Enter search term
ser			untitled					Commonly Used Blocks Continuous Dashboard
Brow	۲							Discontinuities Additional Math Commonly Continuous Discrete & Discrete Used Blocks
Model	•							Logic and Bit Operations Lookup Tables Math Operations
	⇒							Math Operations Messages & Events Model Verification Dashboard Discontinuities Discrete
	A							Model-Wide Utilities Ports & Subsystems Signal Attributes
								Signal Routing
Rea	dy				100	%	 	C CUCC and Bit LOOKUP Main +

The following screen will appear on your computer:

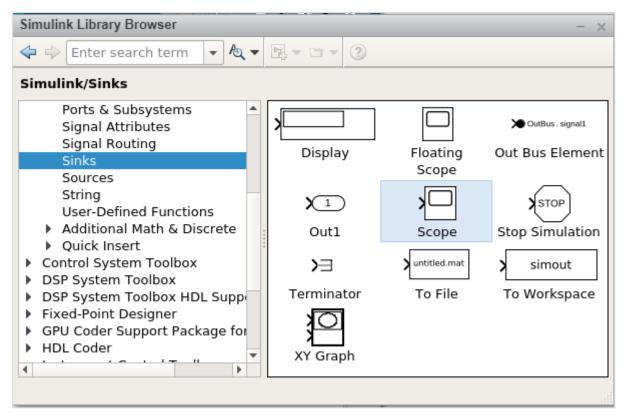


To move the sine wave to your model, select the block and drag it inside your model workspace. We want to display the sine wave and here, we have taken four sign wave in our model workspace as shown below:



untitled * - Simulink – ×												- ×	
S	SIMULATION		DEBUG	MODELING	FORMAT	APPS				1 5 C	₽, -	? •	•
Nev			Library Library LibrAry	Log Signals	Add Viewer	Stop Time 10.0 Normal • Fast Restart	Step Run S Back - Fo		Stop	Data Inspector REVIEW RESULTS			
er	$\langle -$	⇒ ☆	untitled										Pr
Model Browser													Property Inspector
Rea	dy					100%				Va	riable	StepAut	to _{st}

Now we want to display the output of the signal, so let us use a Scope block from sinks library as shown below:



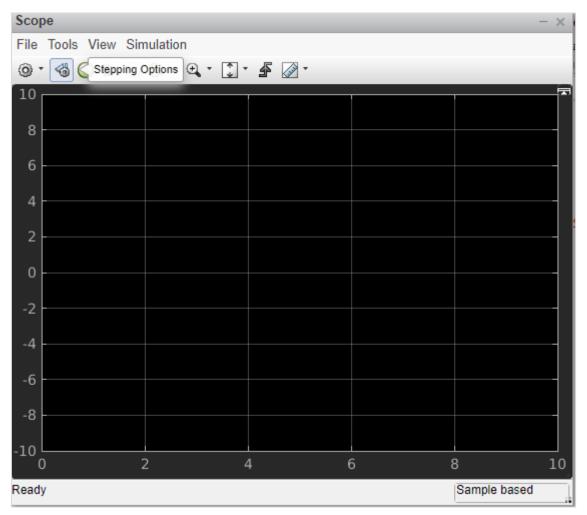
Now select and drag the Scope block inside your model workspace.





The sine wave has one output and the scope block has one input. We have four sine waves displayed. We have to change the parameters of scope block to take four inputs.

Right click on scope block and click on Block Parameters and it will display the screen as shown below:





Go to settings icon and change the input parameter from 1 to 4 as shown below:

Configur	Configuration Properties: Scope - ×						
Main	Time	Display	Logging				
🗌 Ope	en at sim	ulation star	rt				
🗌 Disp	play the f	full path					
Number of input ports: 4 Layout							
Sample time:			L				
Input processing: Elements as channels (sample based					e based) 👻		
Maximize axes:			ff			-	
Axes scaling:			lanual		•	Configure	
0			<u> </u>	<u>o</u> k	<u>C</u> ancel	<u>A</u> pply	'

Click on Apply to save the changes.

Let us now connect the sine wave to the scope block with arrows as shown below:



We would like to change the frequency of each sine wave to a different one, so that we get a signal graph of different frequencies.

So right click on sine wave and open the Sine wave block parameters as shown below:



Block Parameters: Sine Wave - ×					
Sine Wave					
Output a sine wave:					
O(t) = Amp*Sin(Freq*t+Phase) + Bias					
Sine type determines the computational technique used. The parameters in the two types are related through:					
Samples per period = 2*pi / (Frequency * Sample time)					
Number of offset samples = Phase * Samples per period / (2*pi)					
Use the sample-based sine type if numerical problems due to running for large times (e.g. overflow in absolute time) occur.					
Parameters					
Sine type: Time based 🔹					
Time (t): Use simulation time					
Amplitude:					
1					
Bias:					
0					
Frequency (rad/sec):					
3					
Phase (rad):					
0					

We are going to keep the amplitude as 1 for all sine waves and the frequency of the first sine wave is 1, second one is 3, third one is 6 and the last one is 10.

Click on the Run button as shown below to see the sine wave.



- ×		0						- Simulink	titled *	unt
• ? • •	🖥 🕤 🗟 🖳		BLOCK	APPS	FORMAT	MODELING	DEBUG	ION	SIMULAT	ę
	Data Inspector REVIEW RESULTS	Step Forward	Step Back V SIMULATE	Stop Time 10.0 Normal	Add Viewer	Log Signals	Library Browser	Open 👻 Save 👻	W	Ne'
р							untitled			er
Property Inspector						↓ Ay ine Wave	r s			Model Browser
						ine Wave	ſ		E 4 63	

Open the scope block parameters to see the sine wave as shown below:

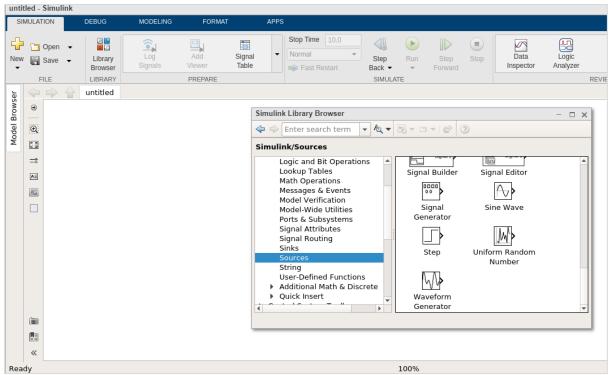




7. MATLAB Simulink — Signals Processing

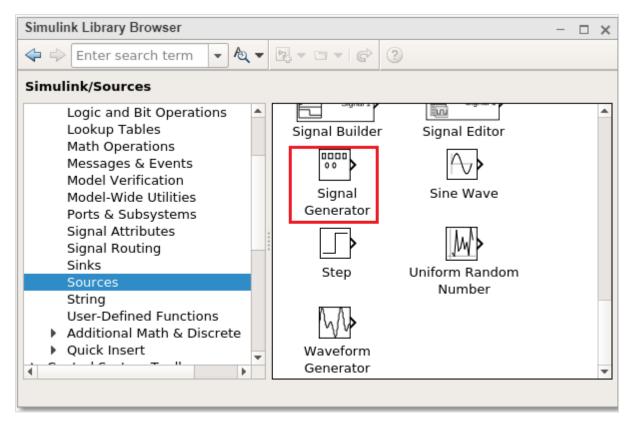
In this chapter, we will understand the signals generation in Simulink.

To start with, select a blank model from Simulink page and open Simulink browser library as shown below:



In sources library, you will get a signal generator symbol. It will help us to create different types of signals.



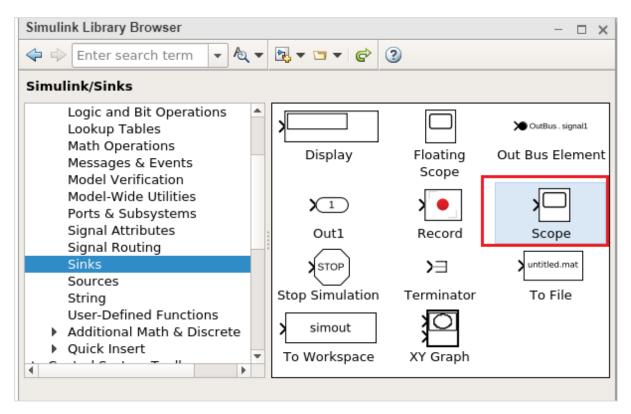


Select the Signal Generator and drag it to get inside the blank model as shown below:



To see the output of the signal generator, we need one more block called scope from sinks library as shown below:





Select the block and drag it to get inside the model.



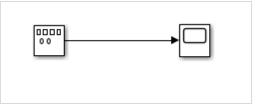
Double click on signal generator or right click and select block parameters and it will display a screen as shown below:



Block Parameters: Signal Generator - ×
Signal Generator
Output various wave forms: Y(t) = Amp*Waveform(Freq, t)
Parameters
Wave form: sine
Time (t): Us square sawtooth random
Frequency:
1
Units: rad/sec Interpret vector parameters as 1-D
OK Cancel Help Apply

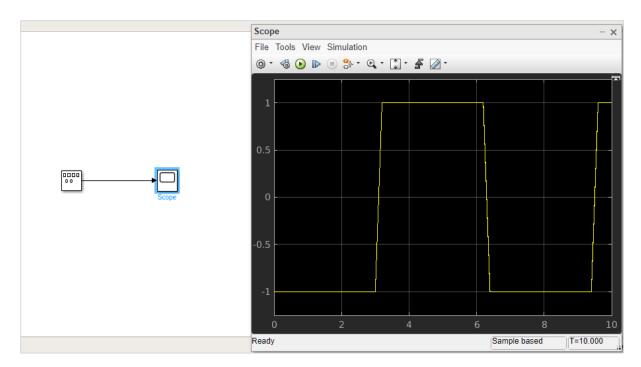
The signal generator can show waveforms like sine, square, sawtooth, random. We will select the square waveform. Let the amplitude and frequency be as 1. Click on OK to update the changes made.

Now, connect the lines between signal generator and scope block as shown below:



Now click on Run button to see the square waveform as shown below:



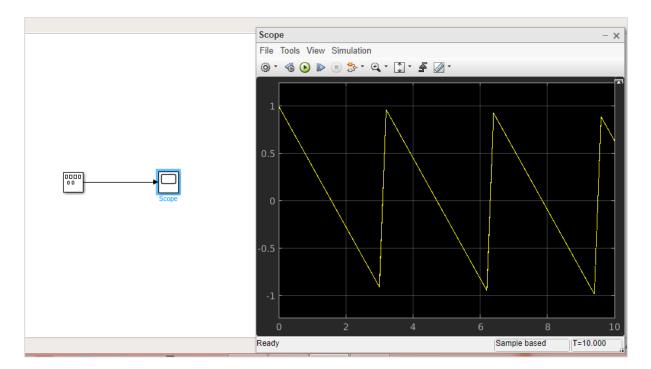


Let us now try the sawtooth wave form. Right click signal generator or double click and change the waveform to sawtooth.

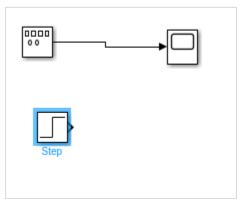
	Block Parameters: Signal Generator	- ×
	Signal Generator	
	Output various wave forms: Y(t) = Amp*Waveform(Freq, t)	
	Parameters	
	Wave form: sawtooth	•
	Time (t): Use simulation time	-
Signal	Amplitude:	
Generator	1	:
	Frequency:	
	2	
	Units: rad/sec	-
	Interpret vector parameters as 1-D	
	OK <u>C</u> ancel <u>H</u> elp <u>A</u>	pply

Let us change the frequency to 2. Click on OK to update the changes. Now run the model to see the changes as shown below:





Let us now add some more signals to the above model. We will take the step signal from the sources library as shown below:

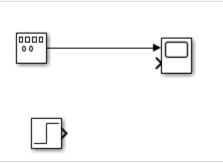


We just have one input for the scope block. Let us increase it to 2 inputs. Right click and open the block parameters.



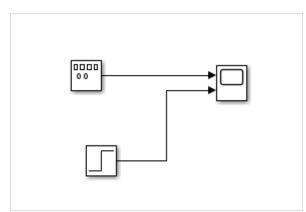
Configuration Properties: Scope – 🗙							×
Main	Time	Display	Logging				
Open at simulation start							
Display the full path							
Number of input ports: 2					Layout	וו	
Sample time:			1				
Input processing:			Elements as	channels	(sample	e based) 👻	
Maximize axes:			Off			•	
Axes scaling:			Manual		•	Hide	
Sc	ale axes	limits at s	top				
Y-axi	s						
Data range (%): 80				Align:	Cente	er 👻	
Autoscale X-axis limits							
Da	Data range (%): 100 Align: Center 🗸						
0				<u>o</u> ĸ	<u>C</u> ancel	Apply	

Click on OK button to update the changes. Now, the scope block has 2 inputs as shown below:

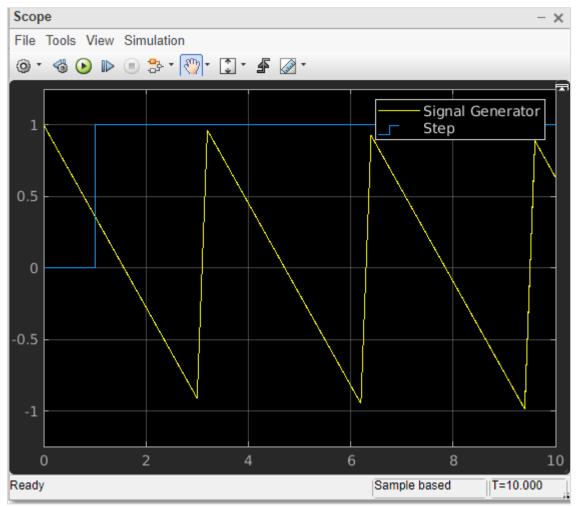


Connect the step input arrow with the scope arrow.





Now click on Run button to run the model.



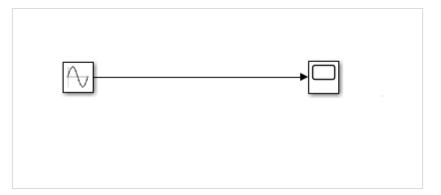
You can add some more signals and test the same.



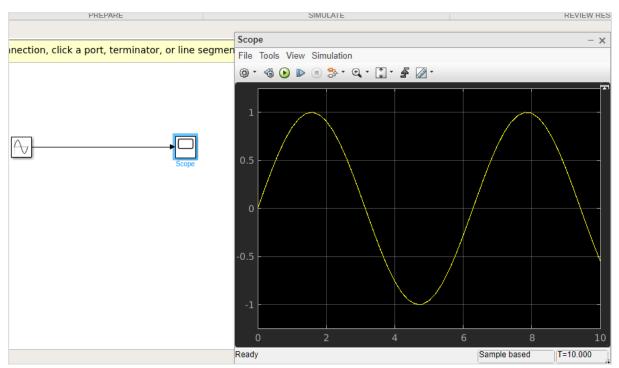
8. MATLAB Simulink — Adding Delay to Signals

We have learnt in the previous chapter about the different signal simulations. In this chapter, we will learn how to add delay to the signals.

Let us take a blank model and add sine wave and scope block to it as shown below:

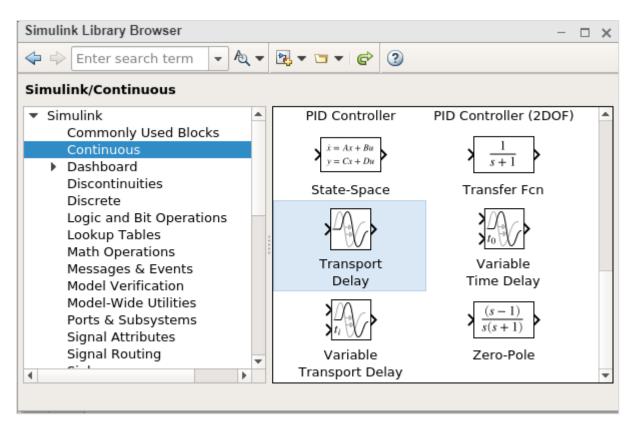


Let us now run the model to see the simulation in scope block. The sine wave is as shown below:



Let us now add delay for the sine wave. We will make use of transport delay block from continuous library as shown below:





Select the block and drag it in your model canvas. Now that we have the Transport delay in our model, right click on it and open block parameters as shown below:



PREPARE	Block Parameters: Transport Delay	×					
	Transport Delay						
, <mark>click a port, terminat</mark>	Apply specified delay to the input signal. Best accuracy is achieved when the delay is larger than the simulation step size						
	Parameters						
	Time delay:						
	1	:					
	Initial output:						
	0	:					
DAV	Initial buffer size:						
Transport	1024	:					
Delay	Use fixed buffer size						
	Direct feedthrough of input during linearization						
	Pade order (for linearization):						
	0	:					
	OK <u>Cancel H</u> elp <u>Apply</u>	/					

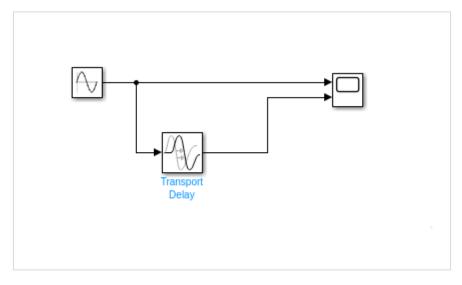
Let us change the time delay from 1 to 3. Make the changes and click on OK button.

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:



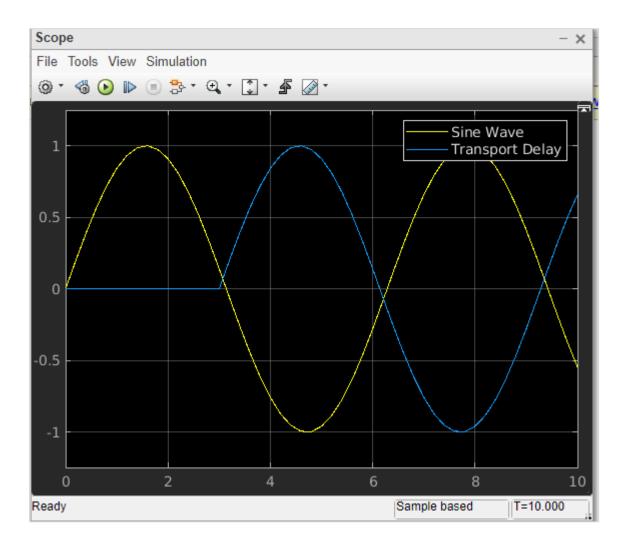
	-	Explore			
	-	_ ,		REVIEW RESULTS	
		Op <u>e</u> n			
or li	i 4	Cu <u>t</u>	Ctrl+X	lement. More information. Do not sh	ow again
_	1	<u>і С</u> ору	Ctrl+C		
	Ê	Paste	Ctrl+V		
		Comment Through	Ctrl+Shift+Y		
		Comment Out	Ctrl+Shift+X		
		Uncomment			
+0		<u>D</u> elete	Del		
		Create Subsystem from Selection	Ctrl+G		
		<u>F</u> ormat	•		
		<u>R</u> otate & Flip	•		
		<u>A</u> rrange	•		
		Mask	•		
		Library Link	Þ		
		Signals & Ports		N <u>u</u> mber of Input Ports	1
		Reguirements	Þ	Input Port Signal Properties	v v 2
		Fixed-Point <u>T</u> ool		Output Port Signal Properties	3
		Block <u>P</u> arameters (Scope)			<u>M</u> ore
		Proper <u>t</u> ies			

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



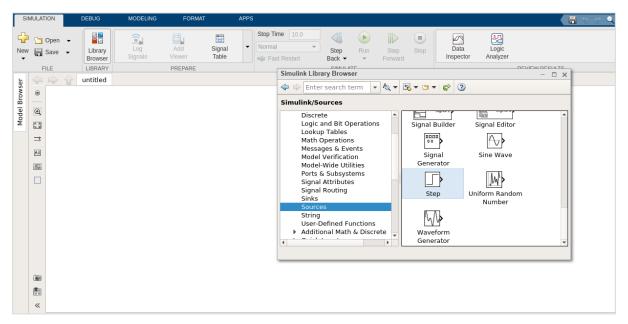
Now run the simulation to see a delay of 3 seconds to the sign wave. Right click scope block and select block parameters to see the display.





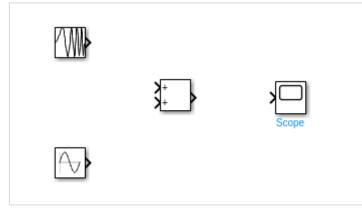


In this chapter, we will learn how to sum the two given signals and get the output. Select the blank model and open Simulink library browser as shown below:



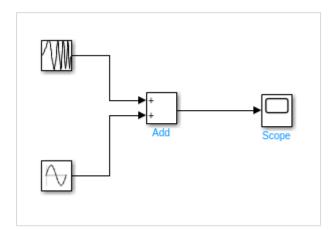
We are going to combine chirp signal and sine wave blocks by using add block from Math operation and see the final display.

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.



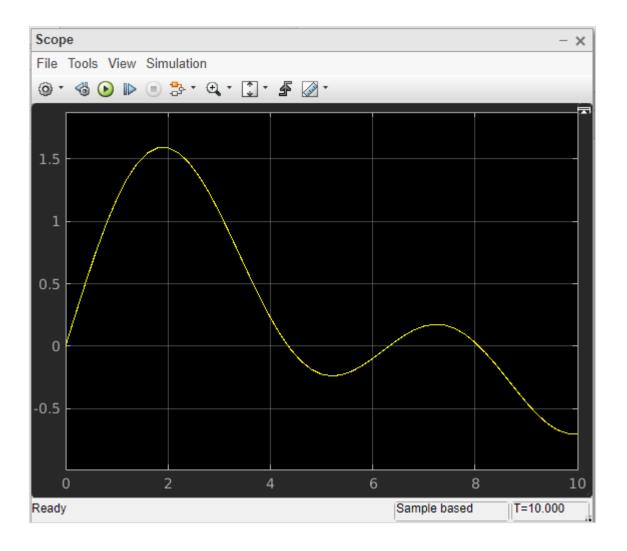


Double click chirp signal and change the initial frequency from 0.1 to 0.05 and click on Ok button.

Block Parameters: Chirp Signal –	×
chirp (mask) (link)	
Output a linear chirp signal (sine wave whose frequency varies linearly with time).	
Parameters	
Initial frequency (Hz):	
0.05	
Target time (secs):	
100	
Frequency at target time (Hz):	
1	
 Interpret vector parameters as 1-D 	
OK <u>Cancel H</u> elp <u>Apply</u>	

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:

Search			All 🗸 🔾
> My Templates			Learn Mor
✓ Simulink			
Blank Model	Blank Subsystem	Blank Library	Project from Git
Project from \$VN	Digital Filter	Feedback Controller	Fixed-step

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

itled - Simu	link				
IMULATION	DEBUG	MODELING	FORMAT	APPS	
V 🔒 Save	 Library Browser 	Log Signals	Add Viewer	Signal Table Stop Time 10.0 Normal Table Step Back - Forward Stop Stop Data Logic Inspector Analyzer	REVIEW RES
			T TALE PARA	OMOUNE	HETEN HED
 ● ●				Simulink Library Browser - ×	
				 Simulink Commonly Used Blocks Continuous Dashboard Discontinuities Discrete Logic and Bit Operations Lookup Tables Math Operations Messages & Events Model Verification Model Verification Model Verification Model Verification Model Vultities Ports & Subsystems Signal Attributes Signal Routing 	
	HMULATION → Open → → ← → ← → → ← → ← → ← → → ← → ← → ← → ← → ← → ← → ← → ← → ← → ←	Open Save FILE Ubrary Browser Library Browser Library Browser Italian Italian Italian Italian Italian	IMULATION DEBUG MODELING Open Image: Standard Standar	IMULATION DEBUG MODELING FORMAT Open Image: Comparison of the second	INNULATION DEBUG MODELING FORMAT APPS Image Step Run Step Step Data Logic Image Step Run Step Step Data Logic Image PREPARE PREPARE Simulink Simulink Simulink Simulink Image Image Simulink Simulink Image Image Image Image Simulink Simulink Simulink Image Image Image Image Simulink Simulink Image Image Image Image Image Image Simulink Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image Image



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

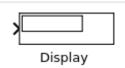
Constant block from Commonly used blocks



Switch block from Signal Routing



Display block from Sinks

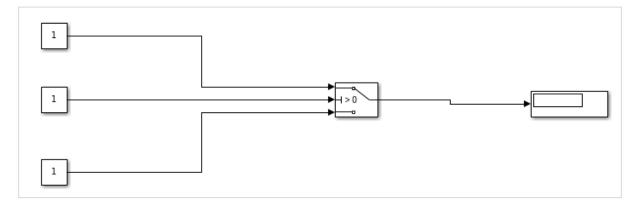


Let us now get all the blocks together to create a model as shown below:

1	
1	
1	

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.

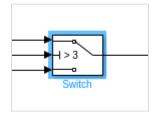


The threshold value will be compared with the block in the center. Based on the constant value of the middle block, the first block value will be displayed or the last constant block value will be displayed.

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

	Block Parameters: Switch –	×				
then click a c	Switch					
	Pass through input 1 when input 2 satisfies the selected criterion; otherwise, pass through input 3. The inputs are numbered top to bottom (or left to right). The first and third input ports are data ports, and the second input port is the control port. The criteria for control port 2 are u2 >= Threshold, u2 > Threshold or u2 \sim = 0.					
-	Main Signal Attributes					
	Criteria for passing first input: u2 > Threshold	-				
Switch	Threshold:					
	3					
	Enable zero-crossing detection					
	OK Cancel Help Apply					

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:



PR	Block Parameters: Constant1 - ×
	Constant
	Output the constant specified by the 'Constant value' parameter. If 'Constant value' is a vector and 'Interpret vector parameters as 1-D' is on, treat the constant value as a 1-D array. Otherwise, output a matrix with the same dimensions as the constant value.
1	Main Signal Attributes
	Constant value:
1	1
Constant1	✓ Interpret vector parameters as 1-D
	Sample time:
1	inf
	OK Cancel Help Apply

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:

	Block Parameters: Constant – 🗙
	Constant
1	Output the constant specified by the 'Constant value' parameter. If 'Constant value' is a vector and 'Interpret vector parameters as 1-D' is on, treat the constant value as a 1-D array. Otherwise, output a matrix with the same dimensions as the constant value.
Constant	Main Signal Attributes
_	Constant value:
1	0.5
	 Interpret vector parameters as 1-D
	Sample time:
1	inf
	OK Cancel Help Apply

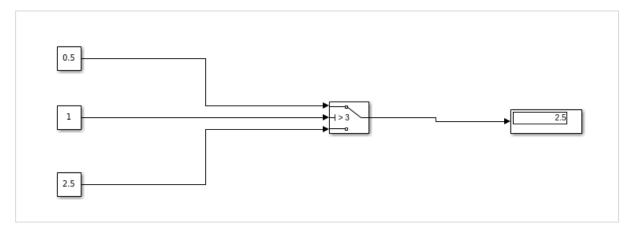
Let us now change the last constant with value as 2.5 as shown below:



	Block Parameters: Constant2 – 🗙
_	Constant
.5	Output the constant specified by the 'Constant value' parameter. If 'Constant value' is a vector and 'Interpret vector parameters as 1-D' is on, treat the constant value as a 1-D array. Otherwise, output a matrix with the same dimensions as the constant value.
1	Main Signal Attributes
_	
ı	2.5
stant2	✓ Interpret vector parameters as 1-D
	Sample time:
	inf :
	OK Cancel Help Apply

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:

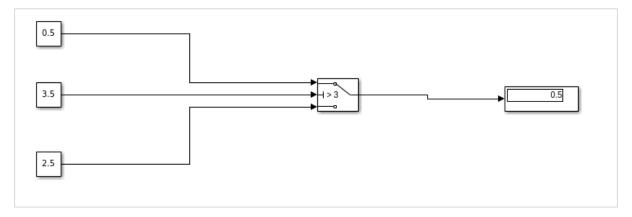


Let us now change the middle constant to a higher value than the threshold of switch and see the output:



	Block Parameters: Constant1 – 🗙
	Constant
0.5	Output the constant specified by the 'Constant value' parameter. If 'Constant value' is a vector and 'Interpret vector parameters as 1-D' is on, treat the constant value as a 1-D array. Otherwise, output a matrix with the same dimensions as the constant value.
1 Constant1	Main Signal Attributes Constant value:
	3.5
2.5	 ✓ Interpret vector parameters as 1-D Sample time:
	inf
	OK Cancel Help Apply

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.



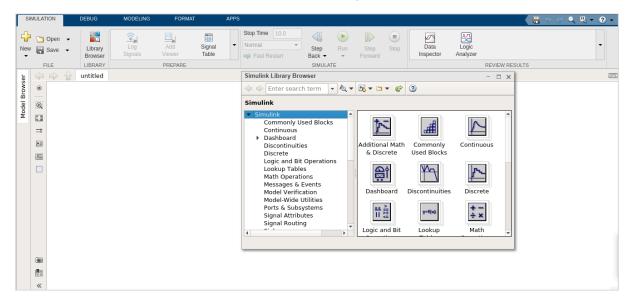
11. MATLAB Simulink — Logic Gates Model

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

Search All 🗸 a Learn Mor > My Templates Simulink アン ----Create Model ✓ Blank Model Blank Subsystem Blank Library Project from Git ~ -(m) ---Digital Filter Fixed-step Project from SVN Feedback Controller

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

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



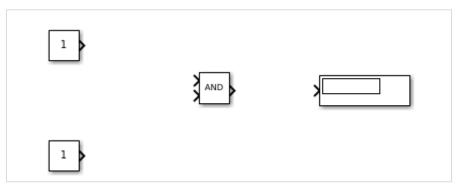
Let us select the block that we want to build a OR gate. We need two constant blocks to act as inputs, a logic operator block and a display block.

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:



Model Browser	K3 €	untitled					
Co	nstan	t	<u> </u>				
Sim	ulink –	also in HDL Coder, DSP System Toolbox HDL Su	pport, DSP System Toolbox				
		t Ramp					
	-	n Toolbox/Signal Operations					
	d Con						
Simulink/Quick Insert/Math Operations							
	_	onstant					
	ulink/S	-					
		by Constant					
Fixed-Point Designer							
	-	e To Constant					
Simulink — also in HDL Coder							
Enumerated Constant							
Simulink — also in HDL Coder							
Divide by Constant and Round							
Fixed-Point Designer							
		Blocks	Actions (Ctrl+.)				

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

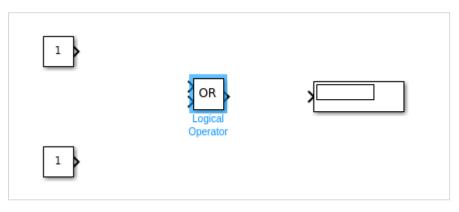


Right click on the logic operator block and it will display the block parameters as shown below:

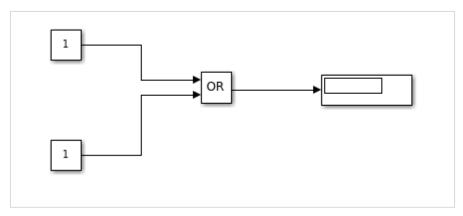


	Block Param	neters: Logical Operator	- ×
	Logical Ope	erator	
		perators. For a single input, operators are applied across the input vector. For multiperators are applied across the inputs.	ple
	Main D	Data Type	
Operator	Operator:	AND	
	Number of	OR	
	2	NAND	
		NOR	
	Icon shape	XOR	
		NXOR	H
	0	NOT	

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.

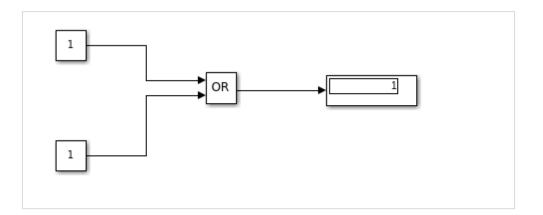


Now connect the lines and the model will be as shown below:



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:

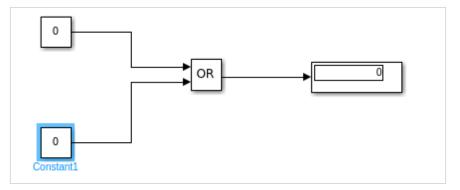




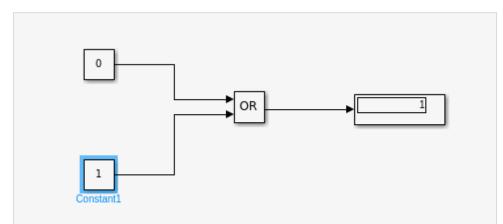
We can see in the display block the output shown is 1. Let us now change the constant value to 0. Right click on constant block and change the value as shown below:

	Block Parameters: Constant – 🗙					
	Constant					
L Constant	Output the constant specified by the 'Constant value' parameter. If 'Constant value' is a vector and 'Interpret vector parameters as 1-D' is on, treat the constant value as a 1-D array. Otherwise, output a matrix with the same dimensions as the constant value.					
	Main Signal Attributes					
1	Constant value:					
	0					
	✓ Interpret vector parameters as 1-D					
	Sample time:					
	inf					
	OK Cancel Help Apply					

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

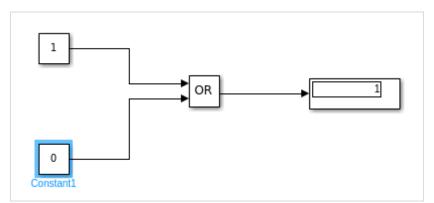






Let us now change the constant values to 0,1 and see the output:

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



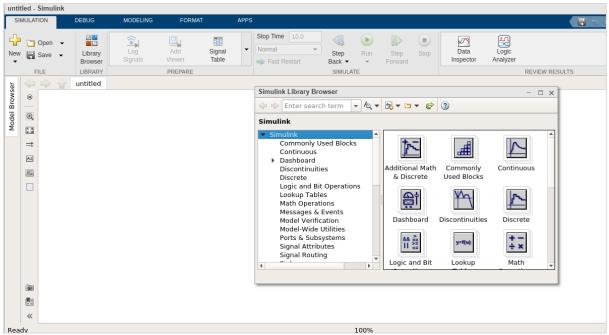
Similarly, you can design the AND and other gates.



12. MATLAB Simulink — Sine wave

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

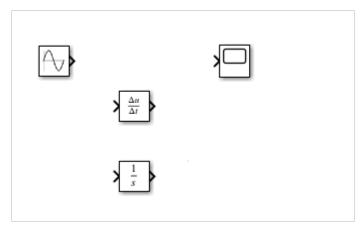
Open blank model and Simulink library as shown below:



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:



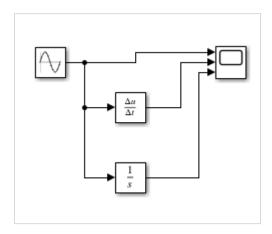


We would need 3 input ports for scope block as the sine wave, derivative and integrator block will be connected to it.

Signals		ble Fast Postart Pack	Step Stop	Inspector Analyzer
	PREPARE	Explore		REVIEW RESULTS
		Op <u>e</u> n		
nnection, click	a port, terminato			odel element. <u>More information.</u> Do not show again
		■ <u>C</u> opy	Ctrl+C	
		Paste	Ctrl+V	
		Comment Through	Ctrl+Shift+Y	
		Comment Out	Ctrl+Shift+X	
		Uncomment		
		Delete	Del	
\wedge		Create Subsystem from Selection	Ctrl+G	
	S.	<u> </u>	•	
		<u>R</u> otate & Flip	•	
		Arrange	•	
		Mask	Þ	
		Library Link	►	
		Signals & Ports	•	Number of Input Ports
		Reguirements	Þ	Input Port Signal Properties
		Fixed-Point <u>T</u> ool		Output Port Signal Properties 3
		Block Parameters (Scope)		<u>M</u> ore
		Proper <u>t</u> ies		

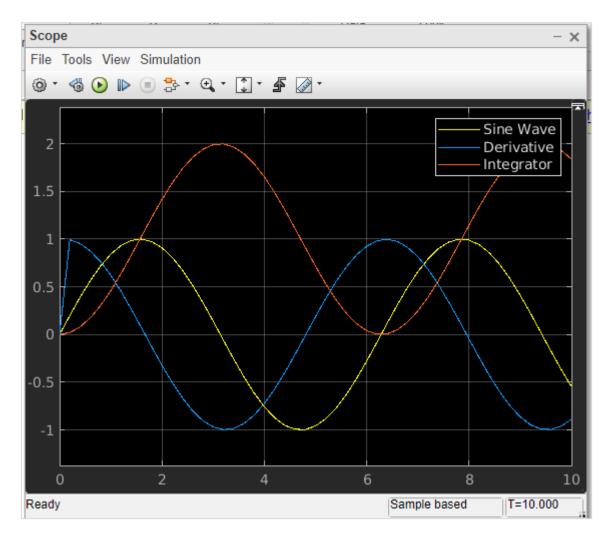
Right click on the scope block and change the inputs from 1 to 3 as shown below:

Connect the lines as shown below:



Now, run the model to see the display.





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

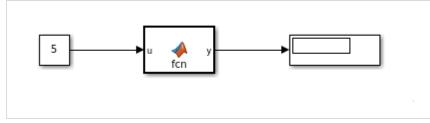


13. MATLAB Simulink — MATLAB Function

In this chapter we will understand how to make use of MATLAB function in Simulink. Open Simulink and click on blank model.

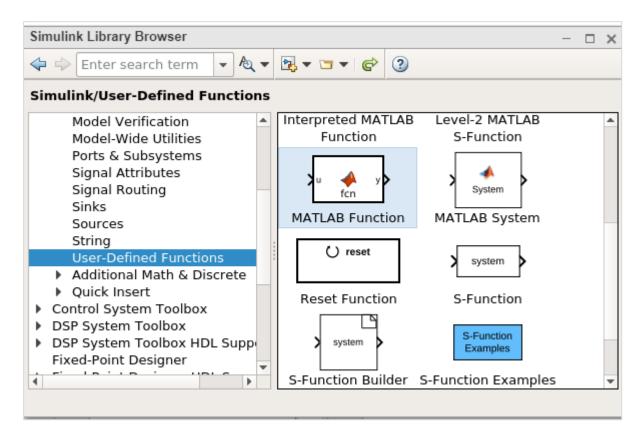
untit	tled - Simuli	ink								-	□ ×
SI	MULATION		DEBUG	MODELING	FORMAT	APPS		6	c 🔍	🖳 🔹 🕐 👻	
New	🎦 Open Save FILE			Log Signals PREPARE	Stop Time 10.0 Normal	Step Back -	Run	Step Forward	Stop	REVIEW RESULTS	I.4
er	$\langle \Rightarrow \Rightarrow$		untitled								Pr
Model Browser											Property Inspector
Read	dy				1	00%				VariableSte	Auto

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.





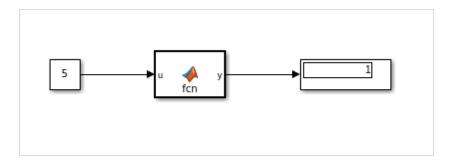
Now, click on MATLAB function and drag it inside your model canvas. Double click on the MATLAB function block and write a function of your choice.

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

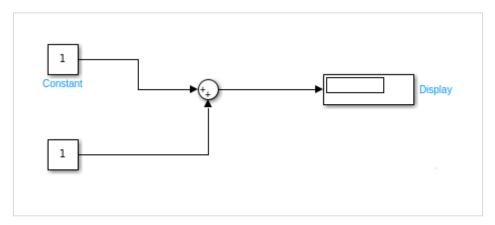
Subsystems are useful when your model gets big and complex. You can change a part of the model into a subsystem that helps to keep the flow very clean and understandable.

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

First, create a blank model, as shown below:

untit	tled - Simulink								-	□ ×
SI	MULATION	DEBUG	MODELING	FORMAT	APPS			c Q	≌, - ? -	•
New T	<u> </u>	LIBRARY	Log Signals	Stop Time 10.0 Normal	Step Back -	Run	Step Forward	Stop	REVIEW RESULTS	►
ser	\langle	untitled								Pro
Model Browser	● ● € * *									Property Inspector
Read	dy			10	0%				VariableStep	Auto

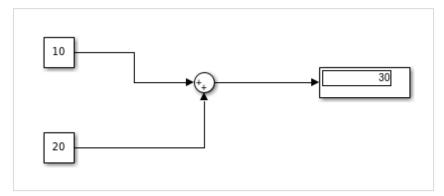
Now, we will create a simple model that adds two numbers and later converts a part of the model into a subsystem.



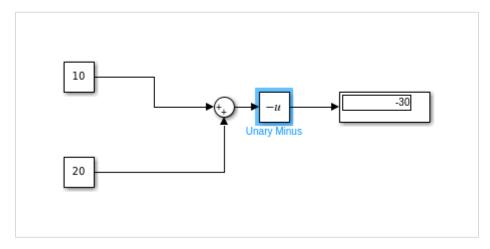
We have created a simple model that has two inputs. These inputs will add up and show the result inside the display.

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

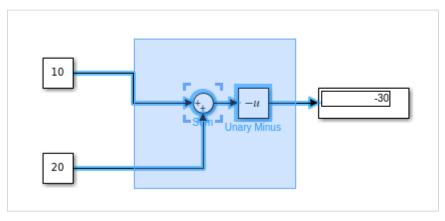




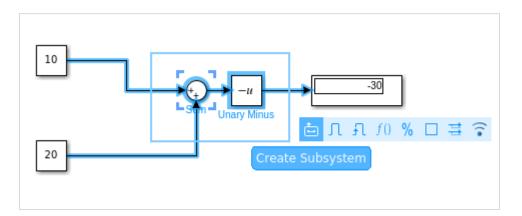
Let us add one more block named Unary minus that will change the output from 30 to -30 as shown below:



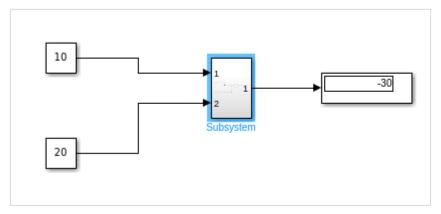
Now, let us select the portion i.e. the sum and the Unary minus block to create a subsystem as shown below:



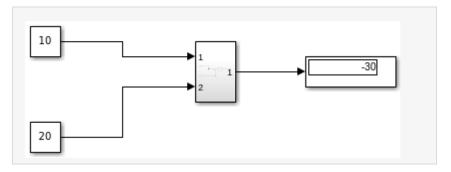




Click on Create Subsystem. Once done, the sum block and the Unary minus block will be converted into a subsystem as shown below:



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:



2	

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



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

First, create a blank model as shown below:

untit	tled - Simul	ink								-	□ ×
SI	MULATION		DEBUG	MODELING	FORMAT	APPS			0 Q	₽ ₂ - ? -	۲
New				Log Signals PREPARE	Stop Time 10.0 Normal	▼ Step Back ▼ SIMULA	Run TE	Step Forward	Stop	REVIEW RESULTS	
er	$\langle \Rightarrow \Rightarrow \rangle$		untitled								P
Model Browser	•										Property Inspector
Rea	dy				10	00%				VariableSte	

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:

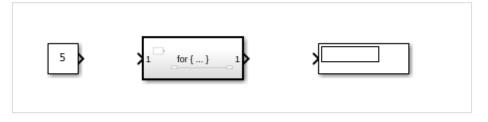


Let us add the for-iterator block as shown below:

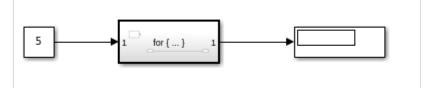


	for
For Iterator Subsystem	
Simulink/Ports & Subsystems	
For Each Subsystem	
Simulink/Ports & Subsystems	
For Each Subsystem	
HDL Coder/Ports & Subsystems	
Forward Substitution	
DSP System Toolbox/Math Functions/Matrices and Linear Al	gebra/Linear System Solvers
Varying Observer Form	
Control System Toolbox/Linear Parameter Varying	
Transfer Fcn Direct Form II	
Simulink/Additional Math & Discrete/Additional Discrete	
Discrete Varying Observer Form	
Control System Toolbox/Linear Parameter Varying	
Switched derivative for linearization	
Simulink Extras/Linearization	

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.



For Iteration For Iterator			
	1		→ 1

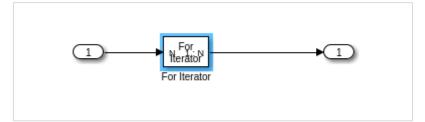
Right click on the for iterator and select the block parameters, as shown below:

Block Parameters: For Iterator – 🗙
For Iterator
Run the blocks in this subsystem for a number of iterations. The iteration limit may be specified either in the block's dialog or through an external input port. If the iteration variable is incremented externally, then the next iteration value is read in through an external input port, otherwise it is incremented by one. The iteration continues to run until the iteration variable exceeds the iteration limit. If the output port is shown, it will output the current iteration number starting at zero or one. When the iteration is started, any states in the subsystem may be either reset to their initial value or held at their previous value.
Parameters
States when starting: reset
Iteration limit source: external
Set next i (iteration variable) externally
✔ Show iteration variable
Index mode: One-based
Iteration variable data type: int32 🔹
OK Cancel Help Apply

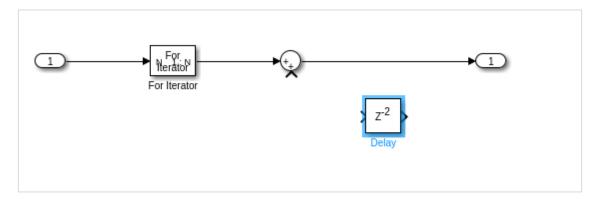
Change the States when starting as reset and Iteration limit source as external. Click on Ok to update the changes.



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:



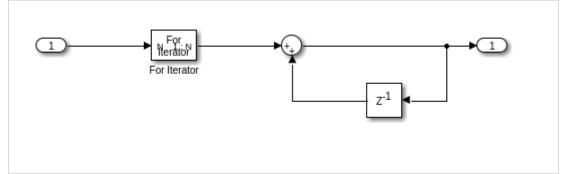
The delay block has to be flipped so that it can be added to the output. We need to give the output back to the sum block so that it can be added with the current iteration.

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



Block Parameters: Delay		- ×
Delay		
Delay input signal by	a specified number of samples.	
Main State Attribut	tes	
Data		
Sou	rce Value	Upper Limit
Delay length: Dia	log 🔻 1	
Initial condition: Dia	log 👻 0.0	
Algorithm		
Input processing: Ele	ements as channels (sample based) 📼	
Use circular buffer	for state	
Control		
Show enable port		
External reset: None	•	
Sample time (-1 for inl	nerited):	
-1		E
0	<u>O</u> K <u>C</u> ancel <u>H</u> elp	Apply

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



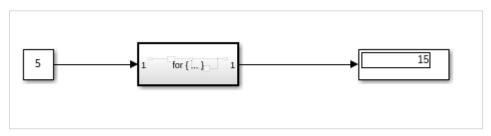
Now before you run the simulation, change the stop time to 1. We do this because we want the simulation to run only once.



Stop Time 1	<
Normal -	s
🟟 Fast Restart	Ba
	S

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.



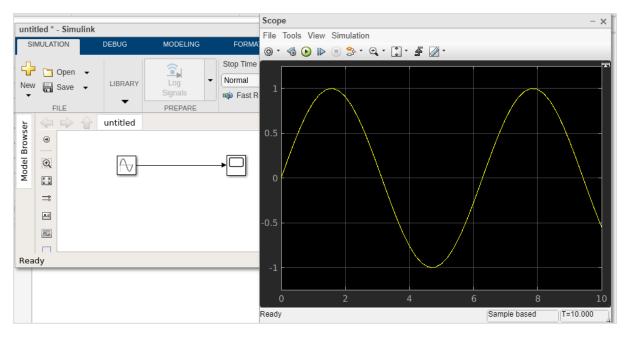


16. MATLAB Simulink — Export Data

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.



Sco	pe				- ×
File	Tools View Simulation				
0	Configuration Properties	🗲 🕢 -			
1					
	Configuration Properties: Scope			- ×	
0.5	Main Time Display L	ogging			
0.5					
	Limit data points to last:	5000			
C	Decimation:	2			
	Log data to workspace				\backslash
-0.5	Variable name:	ScopeData			$ \rightarrow $
	Save format:	Dataset		-	
-1					
		ОК	Cancel	Apply	10
Read	<u> </u>				0

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

Configur	ation Prop	erties: Scop	е			_	×
Main	Time	Display	Logging				
🗌 Lim	it data p	oints to last	5000				
🗌 Dec	imation:		2				
✓ Log	data to	workspace					
Variab	le name:	:	swave				
Save f	ormat:		Array			•	
				к	Cancel	<u>A</u> pply	



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.

 Workspace 	 Workspace 					
:: Name	:: Value	:: Size	:: Class			
🗑 out	1×1 Simula	1×1	Simulink.			
			-			
4			E F			

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

1×1 Simulink.SimulationOut	put			
	:: Property	:: Value	:: Size	:: Class
	ErrorMessage		0×0	char
	🗑 SimulationMetadata	1×1 Simula	1×1	Simulink.Si
		51×2 double	51×2	double
	🕂 tout	51×1 double	51×1	double

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

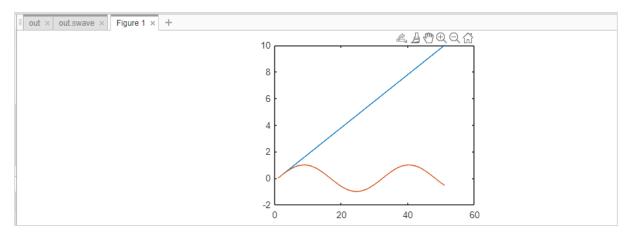


>> out.swav	e
ans =	
0	0
0.2000	0.1987
0.4000	0.3894
0.6000	0.5646
0.8000	0.7174
1.0000	0.8415
1.2000	0.9320
1.4000	0.9854
1.6000	0.9996
1.8000	0.9738
2.0000	0.9093
2.2000	0.8085
2.4000	0.6755
2.6000	0.5155
2.8000	0.3350
3.0000	0.1411

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



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:

matl	abmod	el - Simuli	ink					- [
SIMULATION		DEBUG	MODELING	FORMAT	APPS	📙 ରେଟ 🔍	₽ ₄ - ? -	•	
	F S	ave 👻		Log Signals PREPARE	 ✓ Stop Time 10.0 ✓ Normal Imp Fast Restart 	-		REVIEW RESULTS	4
er	4		matlabmode	el					Pr
Model Brows									Property Inspector
Rea					10	0%		VariableStep	
	Model Browser	SIMULATIC New III S FIL Woodel Browser III III IIII IIIIIIIIIIIIIIIIIIIIIIII	SIMULATION New Save • FILE Open • Price Open • FILE Open • • • • • • • • • • • • • • •	Open New Save FILE matlabmode Image: Save Image: Save FILE matlabmode	SIMULATION DEBUG MODELING	SIMULATION DEBUG MODELING FORMAT COPEN COPEN LIBRARY FILE FILE FILE matlabmodel FILE FILE FILE FILE FILE FILE FILE FIL	SIMULATION DEBUG MODELING FORMAT APPS Image: Constraint of the state of the s	SIMULATION DEBUG MODELING FORMAT APPS Image: Comparison of the state of the	SIMULATION DEBUG MODELING FORMAT APPS Image: Comparison of the state of the

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.

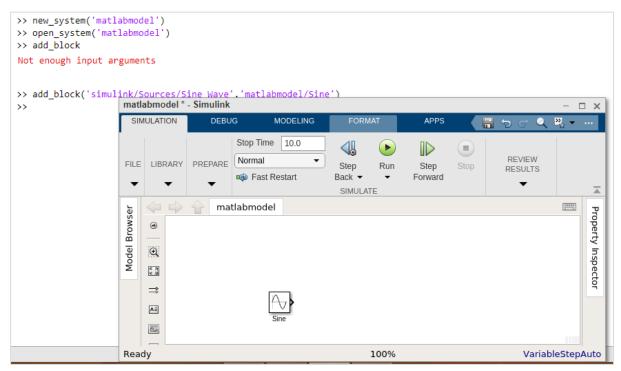


Sine	✓ J Open → Stop Time 10.0 Wave LIBRARY Log ✓ Normal								
	simulink/Sources/Sine Wave:								
Ņ	Output a sine wave:								
າiform Nun	$1000 = Amp^*Sin(Freq^*(\pm Phase) \pm Blas)$								
	Sine type determines the computational technique used. The parameters in the two types are related through: Samples per period = 2*pi / (Frequency * Sample time)								
	Number of offset samples = Phase * Samples per period / (2*pi)								
	Use the sample-based sine type if numerical problems due to running for large times (e.g. overflow in absolute time) occur.								

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:

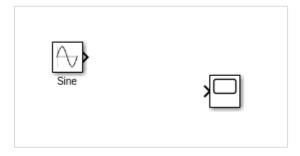




Let us now add the scope block, as mentioned below:

```
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:

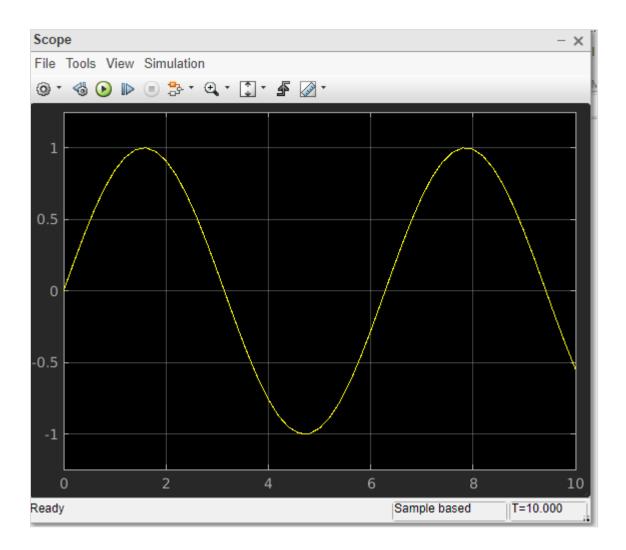
```
result = sim('matlabmodel');
```

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:

untitled1 - Simulink – 🗆 🗙											
SIMULATION			DEBUG	MODELING	FORMAT	APPS		ি 🔒 🕤	c 🔍	≥, - ? -	۲
New	Dpen			Log Signals PREPARE	Stop Time 10.0 Normal	▼ Step Back ▼ SIMULAT	Run TE	Step Forward	Stop	REVIEW RESULTS	► ►
er	$\langle \Rightarrow \Rightarrow$		untitled1								P
Model Browser											Property Inspector
Ready					10	0%				VariableSte	pAuto

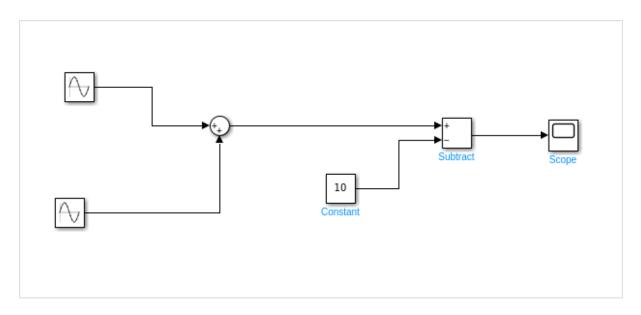
Following are the steps to solve the equation:

- 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.

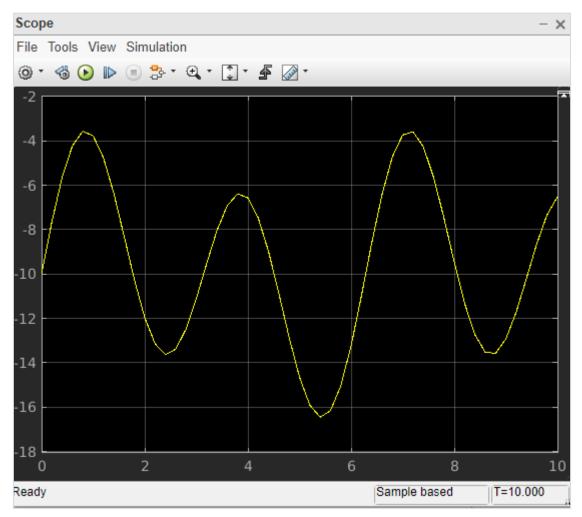
This is how the final Simulink model looks like for the equation:

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

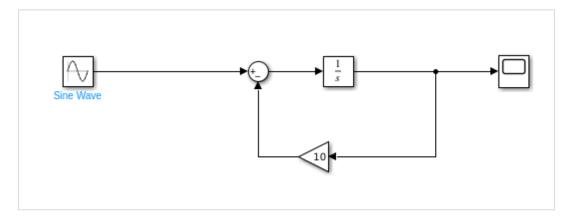
The equation can be solved by integrating dy/dt to the following:

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

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:



