Experiment 3

Getting Start with Simulink

<u> Objectives :</u>

By the end of this experiment, the student should be able to:

- 1. Build and simulate simple system model using Simulink
- 2. Use Simulink test and measurement tools.

1. Introduction

Simulink is a program for simulating signals and dynamic systems. Simulink has two phases of use: model definition and model analysis. It provides an interactive graphical environment and a customizable set of block libraries that let you design, simulate, implement, and test a variety of time-varying systems, including communications, controls, signal processing, video processing, and image processing.

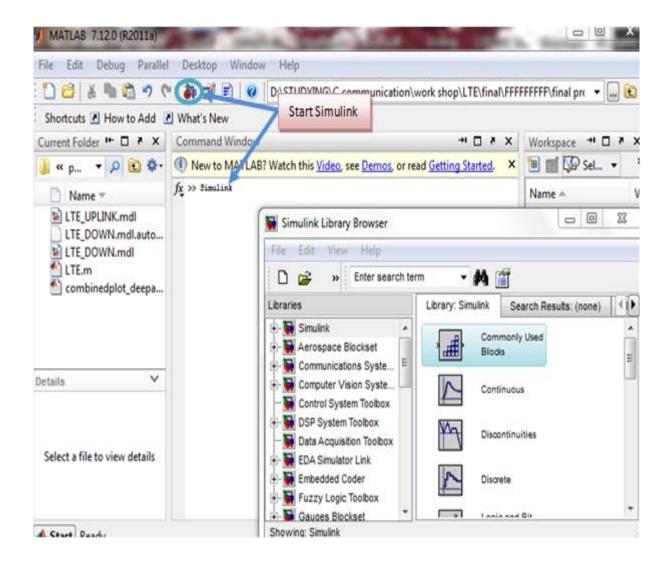
Simulink provides toolboxes for designing, simulating, and analyzing communications systems. The Simulink enables source coding, channel coding, interleaving, analog and digital modulation, equalization, synchronization, and channel modeling.

A typical session starts by either defining a new model or by recalling a previously defined model, and then proceeds to analyze that model. In order to facilitate the model definition, Simulink has a large library of blocks. Models are created by combining proper blocks from the library and edited in the model window principally using mouse-driven operation (**Drag and Drop**). An important part of mastering Simulink is to become familiar with manipulations of various model components in these windows.

After you create (or define) a model, you can analyze it either by choosing options from the Simulink menus in the model window or by entering commands in the Matlab command window. The progress of an ongoing simulation can be viewed while it is running, and the final results can be made available in the Matlab workspace when the simulation is complete.

1.1 Starting Simulink:

- 1. Start up Matlab.
- 2. Start up Simulink by clicking the Simulink icon ^{III} or by typing >>Simulink at the Matlab command window.
- 3. You should see the Simulink Block Library window as shown in figure 1.



1.3 The Simulink Library

The Simulink Library Browser is the library where you find all the blocks you may use in Simulink. Simulink software includes an extensive library of functions commonly used in modeling a system. These include:

🙀 Simulink Library Browser		
File Edit View Help		
🗅 🚅 🔹 Enter search te	rm 👻 🚧 🎬	
Libraries	Library: Simulink Search Results: (none) Most Frequently Used Blocks	
Simulink Aerospace Blockset	Commonly Used Continuous Discontinuitie	
⊕ Aerospace Blockset ⊕ Communications Syste	Blocks Continuous Discontinuitie	,
+- W Computer Vision Syste	Logic and Bit	
Control System Toolbox	Discrete II a Coperations Logic and Bit Logic and Bit Logic and Bit Lookup Tables	i .
⊕ DSP System Toolbox	Andel Model-Wide	
🙀 Data Acquisition Toolbox	+ - Math Model Model-Wide + × Operations Ø Verification Misc Model-Wide	
🗄 🕞 EDA Simulator Link		
🗄 🔂 Embedded Coder	Signal Attributes Signal Routin	9
🕀 🙀 Fuzzy Logic Toolbox	Ya Obosystems	
🕀 🙀 Gauges Blockset	Sinks Sources y-ttue User-Defined	
🙀 Image Acquisition Toolb	Functions	
뒑 Instrument Control Tool	+ - Additional Math	
Model Predictive Contr	& Discrete	
🕀 💀 Neural Network Toolbox		
Real-Time Windows Ta		
Report Generator		
Robust Control Toolbox		
E SimEvents		
E SimPowerSystems		
🗄 💀 SimRF		
🕂 🙀 Simscape		
🕀 🔂 Simulink 3D Animation		
🕀 🤯 Simulink Coder		
+- W Simulink Extras		
Simulink Extras		
Stateflow		
System Identification T		
+ Vehicle Network Toolbox		
+ V xPC Target		
Showing: Simulink		

1.3.1 Common Block Libraries:

In this section we will see the most common used block libraries in communication system models.

1. Commonly Used Block

🙀 Simulink Library Browser						
File Edit View Help						
🗅 🚅 » Enter search term 👻 🛤 🎬						
Libraries	Library: Simulink/Commonly Used Blocks	Search Results: (none) Mos	at Frequently Used Blocks			
⊡- 🙀 Simulink						
Commonly Used Blocks	Bus Creator x	Bus Selector 1	Constant			
···· Discontinuities ···· Discrete	Convert Data Type Conversion	Demux	Discrete-Time Integrator			
… Logic and Bit Operations … Lookup Tables … Math Operations	A Gain	Ground	In1			
···· Model Verification	> 1/s Integrator AND	Logical > Operator >	Mux			
···· Ports & Subsystems ···· Signal Attributes	X 1 Out1 X	Product	Relational Operator			
Signal Routing Sinks Sources	Saturation	Scope xim out	Subsystem			
User-Defined Functions	→++→ Sum	Switch	Terminator			
Aerospace Blockset	> 1 z Unit Delay	Vector Concatenate				
Computer Vision Syste Control System Toolbox						
Control System Toolbox DSP System Toolbox						
Data Acquisition Toolbox						
EDA Simulator Link						
Embedded Coder						
Showing: Simulink/Commonly Used Blo	ocks					

2. Continuous :

🙀 Simulink Library Browser		
File Edit View Help		
🗅 😅 \Rightarrow Enter search ter	n 🗸 🚧 📺	
Libraries	Library: Simulink/Continuous Search Results: (none) Mo	st Frequently Used Blocks
Simulink	du/dt Derivative	1 Integrator Limited
··· Discontinuities ··· Discrete	yu 1 x Integrator, u 1 / x Integrator, Second-Order u 2 √ σx Nd-Order Limit	PID(5) PID Controller
… Logic and Bit Operations … Lookup Tables … Math Operations	Ant PID Controller (2DOF) X ⁺ − Ax+Bu y − Cx+Du State-Space	> 1 s+1 > Transfer Fcn
Model Verification	Transport Delay	Variable Tite Transport Delay
···· Ports & Subsystems ···· Signal Attributes	> (8-1) 8(8+1)	
···· Signal Routing		
Sources		
···· User-Defined Functions		
🖭 🔂 Aerospace Blockset		
E Communications Syste		
🕂 🖳 Computer Vision Syste		
🙀 Control System Toolbox		
🖶 🔂 DSP System Toolbox		
🙀 Data Acquisition Toolbox		
EDA Simulator Link		
🖅 🔂 Embedded Coder 🚽		
Showing: Simulink/Continuous		ii.

3. Math Operation:

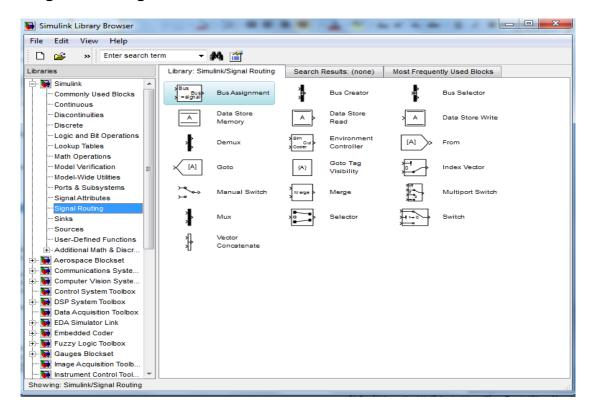
Simulink Library Browser				
File Edit View Help				
Enter search ter	rm 🔻 🚧 📺			
Libraries	Library: Simulink/Math Operations	Search Results: (none)	Most Frequently Us	sed Blocks
Simulink	> u Abs	Add	x ^{r(x)} Solve x Algebr Constra	
··· Discontinuities ··· Discrete	>U Assignment >	u+0.0 Bias		ex to Ma- e-Angle
Cogic and Bit Operations Cookup Tables Math Operations	Real-Imag	X Divide	> Dot Pro	oduct
···· Model Verification	Find Nonzero Elements	Gain		tude-Ang- omplex
····Ports & Subsystems ····Signal Attributes	e ^u Math Function	Matrix Concatenate	> min > MinMa	x
Signal Routing Sinks Sources	R MinMax Runni- ng Resettable	Permute P(2,1) Dimensions	P(u) O(P) = 5	mial E
User-Defined Functions	X Product	Product of Elements	Real-Ir Im	
Aerospace Blockset Sommunications Syste Sommunications Syste	>1//u Reciprocal Sqrt >	U(:)> Reshape	floor Round Function	
Control System Toolbox	> Sign	>±√ ^u > Signed Sqrt	>t Sine W Function	
Data Acquisition Toolbox EDA Simulator Link	> 1 > Slider Gain	> <mark>√u</mark> > Sqrt	Squeeze Squee	ze
Embedded Coder Fuzzy Logic Toolbox Gauges Blockset	>+	X++ Sum	Sum o Eleme	
Image Acquisition Toolb	sin Function	-u Vnary Minus	Vector	tenate 👻
Showing: Simulink/Math Operations				

4. Ports and Subsystem

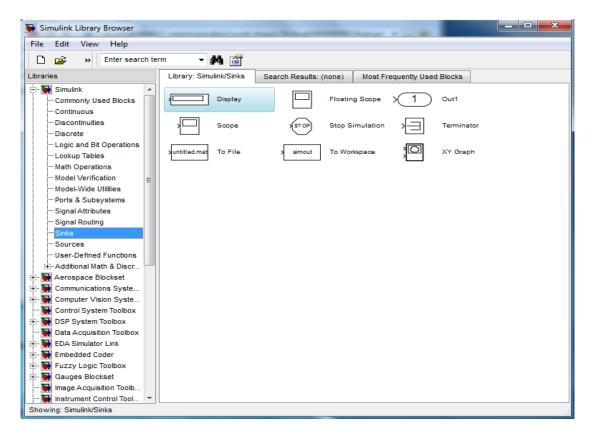
File Edit View Help	
🗅 😅 🔹 Enter search ter	m 👻 🚧 🎬
ibraries	Library: Simulink/Ports & Subsystems Search Results: (none) Most Frequently Used Blocks
Simulink	Atomic Subsystem
··· Discontinuities	☐ Enable
Logic and Bit Operations Lookup Tables Math Operations	For Each For Iterator Subsystem Function-Call Subsystem
	f0 Function-Call Function-Call Generator Split Subsystem
Ports & Subsystems Signal Attributes	r mr×5) as If √m Court If Action 1 In1
····Signal Routing ····Sinks ····Sources	Model Model Variants Out1
	Subsystem Subsystem Subsystem Examples Subsystem Examples
Aerospace Blockset Gommunications Syste Gomputer Vision Syste	Switch Case Ad- ion Subsystem
- 🙀 Control System Toolbox	Variant Verient While Iterator Subsystem Verie () our Subsystem
Data Acquisition Toolbox Data Acquisition Toolbox DDA Simulator Link DDA Simulator Coder	
Erroeaded Coder Erroe Fuzzy Logic Toolbox Erroe Gauges Blockset	
Image Acquisition Toolb	

A.ALASHQAR & A.KHALIFEH

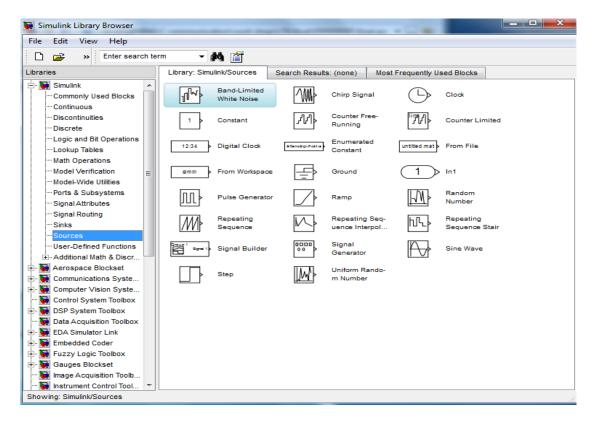
5. Signal Routing



6. Sinks

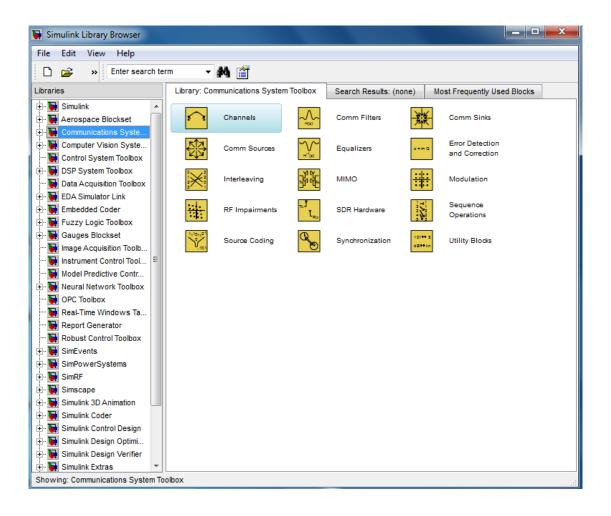


7. Sources



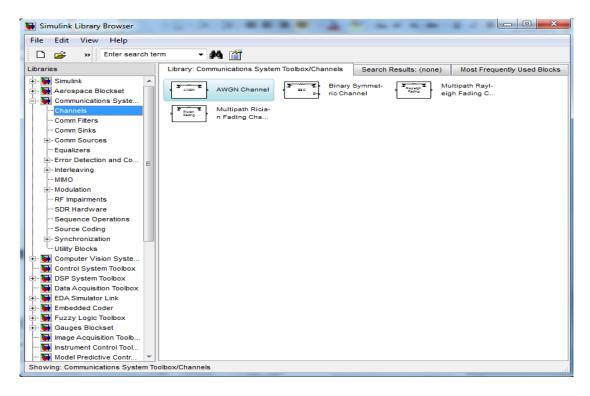
1.4 Communications System Toolbox

Communications System Toolbox provides algorithms for designing, simulating, and analyzing communications systems. The system toolbox enables source coding, channel coding, interleaving, modulation, equalization, synchronization, and channel modeling. You can also analyze bit error rates, generate eye and constellation diagrams, and visualize channel characteristics. Using adaptive algorithms, you can model dynamic communications systems that use OFDM, OFDMA, and MIMO techniques. Algorithms support fixed-point data arithmetic and C or HDL code generation.



1.4.1 Common Block Libraries:

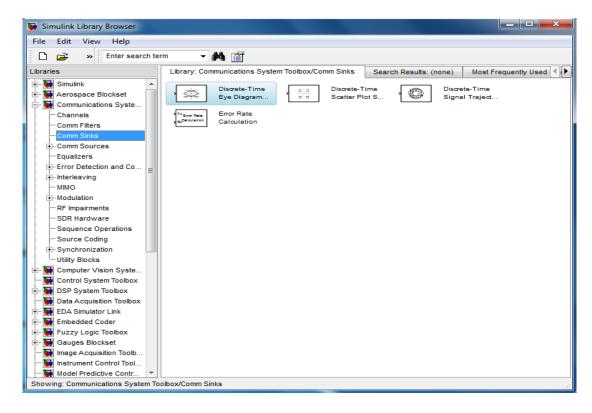
1. Channel :



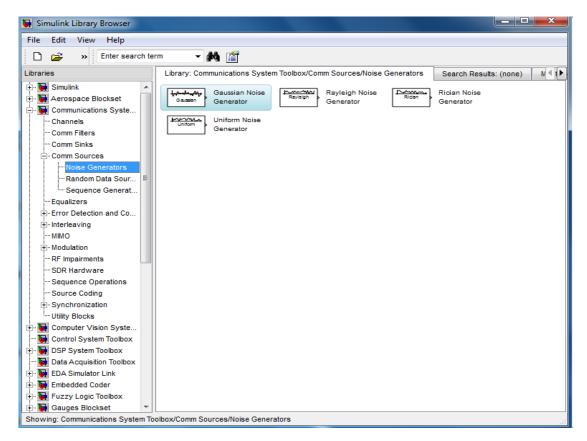
1. Common filters:

Simulink Library Browser	
File Edit View Help	
🗋 🚅 🔹 Enter search te	m 🗸 🗛 🎬
Libraries	Library: Communications System Toolbox/Comm Filters Search Results: (none) Most Frequently Used
Libraries	Library: Communications System Toolbox/Comm Filters Search Results: (none) Most Frequently Used
🙀 Model Predictive Contr 🔻	
Showing: Communications System To	olbox/Comm Filters

2. Common Sink:

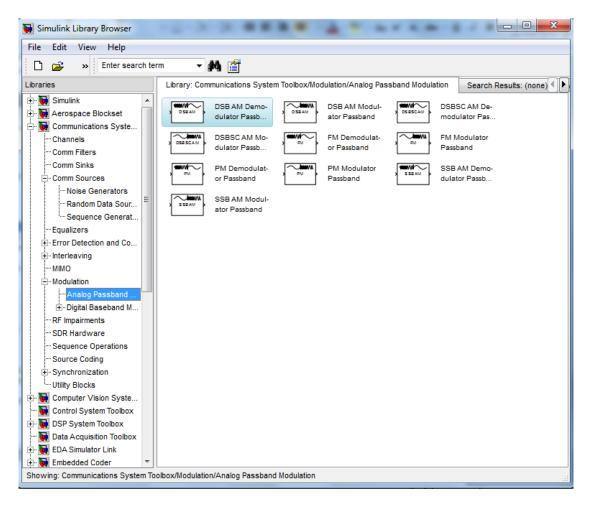


3. Common Source:



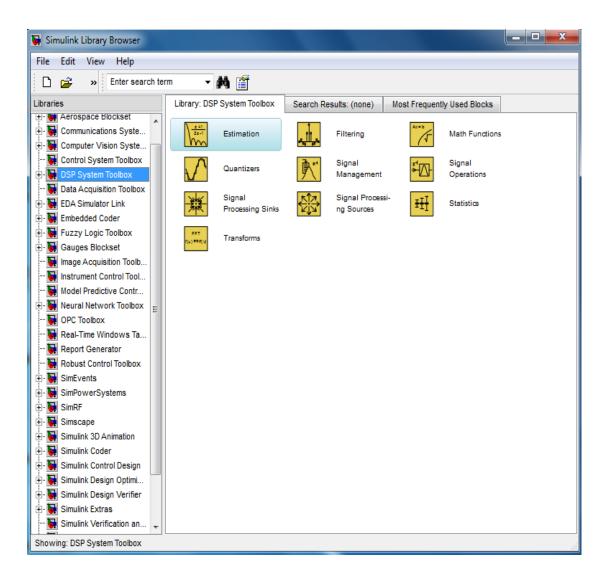
Getting Start with Simulink

4. Modulation:



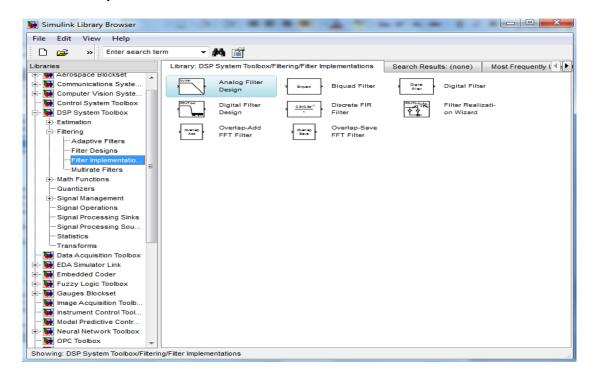
1.5 DSP System Toolbox

DSP System Toolbox provides algorithms for designing and simulating signal processing systems. he system toolbox includes design methods for filters, FFTs, multirate processing, and DSP techniques for processing streaming data and creating real-time prototypes. You can design adaptive and multirate filters, implement filters using computationally efficient architectures, and simulate floating-point digital filters. Tools for signal I/O from files and devices, signal generation, spectral analysis, and interactive visualization enable you to analyze system behavior and performance. For rapid prototyping and embedded system design, the system toolbox supports fixed-point arithmetic and C or HDL code generation.



Common Block Libraries:

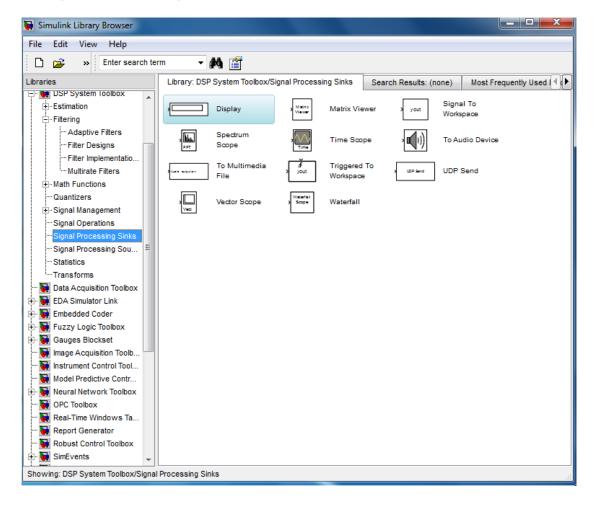
1. Filter Implementation



2. Quantizers

🙀 Simulink Library Browser						
File Edit View Help						
Enter search term						
Libraries	Library: DSP System Toolbox/Quantizers Search Results: (none) Most Frequently Used Blocks					
Libraries DSP System Toolbox 	Library: DSP System Toolbox/Quantizers Search Results: (none) Most Frequently Used Blocks Soalar Quantizer Comparing Soalar Quantizer Soalar Quantizer Comparing Soalar Quantizer Comparing Soal					
Showing: DSP System Toolbox/Quan	lizers					
Showing, DSF System Toolbox/Quan						

3. Signal Processing Sinks



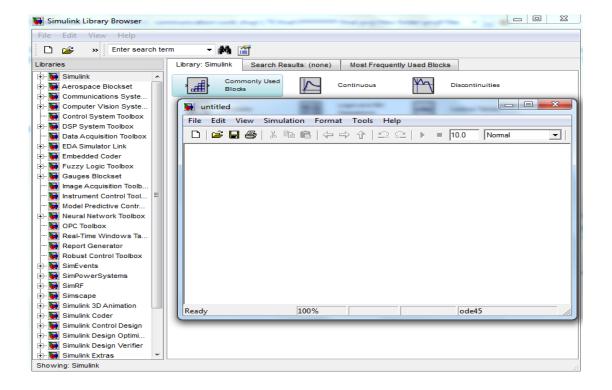
1.6 Building a Simple Model

The Basic Steps

This section describes the basic steps in building a model. It explains how to:

- 4 Open a new model window
- 🖊 Open block libraries
- ∔ Move blocks into a model window
- + Connect the blocks
- 📥 Set block parameters
- 🖊 Set simulation parameters
- Run the model
- 1. Opening a New Model Window

The first step in building a model is to open a new model window. To do so, by clicking the new form icon D or select New from the **File** menu, and then select **Model**. This opens an empty model window, as shown in the following figure.



2. Opening Block Libraries

The next step is to select the blocks for the model. These blocks are contained in libraries. To view the libraries for the products you have installed, type Simulink at the MATLAB prompt (or, on Microsoft Windows, click the Simulink button on the MATLAB toolbar).

3. Simulink Library Browser

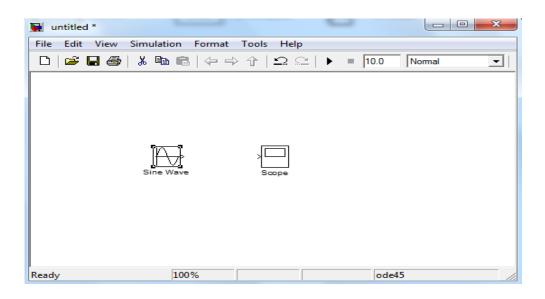
The left pane displays the installed products, each of which has its own library of blocks. To open a library, click the plus sign (+) next to the name of the blockset in the left pane. This displays the contents of the library in the right-hand pane. You can find the blocks you need to build models of communication systems in the libraries of the Communications System Toolbox, the DSP Toolbox Blockset, and Simulink.

A.ALASHQAR & A.KHALIFEH

4. Moving Blocks into the Model Window

The next step in building the model is to move blocks from the Simulink Library Browser into the model window. To do so,

- 1. Click the + sign next to **Simulink** in the left pane of the Library Browser. This displays a list of the Signal Processing Blockset libraries.
- 2. Click Sources in the left pane. This displays a list of the Sources library blocks in the right pane.
- 3. Click the Sine Wave block and Drag and Drop it into the model window.
- **4.** Click **Sinks** in the left pane of the Library Browser. This displays a list of the Sinks library blocks in the right pane.
- 5. Click the Scope Block and drag and Drop it into the model window.



5. Connecting Blocks

The small arrowhead pointing outward from the right side of the Sine Wave block is an output port for the data the block generates. The arrowhead pointing inward on the Scope block is an input port.

To connect the two blocks, click the output port of the SineWave block and drag the mouse toward the input port of the Vector Scope block, as shown in the following figure.



When the pointer is on the input port of the Vector Scope block, release the mouse button. You should see a solid arrow appear, as in the following figure.



6. Setting Block Parameters

To set parameters for the Sine Wave block, double-click the block to open its dialog, as shown in the following figure. Change the following parameters by clicking in the field next to the parameter, deleting the default setting, and entering the new setting in its place:

Parameters
Sine type: Time based
Time (t): Use simulation time
Amplitude:
5
Bias:
0
Frequency (rad/sec):
2*pi*1000
Phase (rad):
pi/2
Sample time:
1e-6
✓ Interpret vector parameters as 1-D
OK Cancel Help

7. Setting Simulation Parameters

Besides individual block parameters, the model also has overall simulation parameters. To view the current settings,

1. Select the Simulation menu at the top of the model window.

2. Select **Configuration parameters** to open the **Configuration Parameters** dialog box, as shown in the following figure.

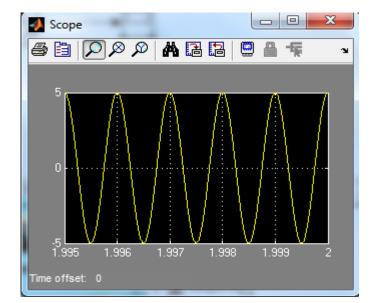
Configuration Parameters: unt	a transformer and	Active)		X
Select:	Simulation time			
Solver Data Import/Export	Start time: 0.0		Stop time: 2	
Optimization Diagnostics Hardware Implementat Model Referencing Simulation Target Code Generation HDL Code Generation	-	cutive min steps:	Solver: Relative tolerance: Absolute tolerance: Shape preservation: 1 Auto	ode45 (Dormand-Prince) 1e-3 auto Disable all
	Higher priority	value indicates higher task priority		
	Time tolerance:	tions trol: Use local settings 10*128*eps cutive zero crossings:	Algorithm: Signal threshold	Nonadaptive
•		m		,
2			ОК	Cancel Help Apply

The **Stop time** determines the time at which the simulation ends. Setting **Stop time** to inf causes the simulation to run indefinitely, until you stop it by selecting **Stop** from the **Simulation** menu.

The **Stop time** is not the actual time it takes to run a simulation. The actual run-time for a simulation depends on factors such as the model's complexity and your computer's clock speed. The settings in the **Configuration Parameters** dialog box affect only the parameters of the current model.

8. Running the Model

Run the model by clicking start simulation icon **b** or selecting **Start** from the **Simulation** menu. When you do so, a scope window appears, displaying a sine wave as shown in the following figure.



1.7 Continuous Signals

The Simulink libraries also contain blocks that generate continuous signals. This means that they update the signal at variable time intervals, whose length is determined by the numerical solver the simulation uses. For example, the Sine Wave block in the Simulink Sources library can generate a continuous sine wave.

1.8 Discrete Signals and Sample Times

The Sine Wave block in the Signal Processing Blockset generates a discrete signal. This means that it updates the signal at integer multiples of a fixed time interval, called the sample time. You can set the length of this time interval in the **Sample time** parameter in the block's dialog.

The sample time T_s can be calculated based on Nyquist theorem that me be stated simply as follow: sampling frequency f_s should be at least twice of the highest frequency contained in the message signal f_m or in mathematical terms

$$f_s \ge 2f_m$$

<u>Note</u>

Many blocks in the Communications Blockset accept only discrete signals. To find out whether a block accepts continuous signals, consult the reference page for the block.

1.9 Frames and Frame-Based Processing

A *frame* is a sequence of samples combined into a single vector. By setting **Samples per frame** to 100 in the Sine Wave block, you set the frame size to 100, so that each frame contains 100 samples. This enables the Vector Scope block to display enough data for a good picture of the sine wave.

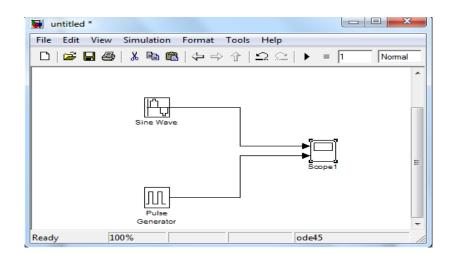
Another important reason to set the frame size is that many Communications Blockset blocks require their inputs to be vectors of specific sizes. If you connect a source block, such as the Sine Wave block, to one of these blocks, you can set the input size correctly by setting **Samples per frame** to the required value. The model described in "Reducing the Error Rate Using a Hamming Code" shows how to do this.

In frame-based processing all the samples in a frame are processed simultaneously. In sample-based processing, on the other hand, samples are processed one at a time. The advantage of frame-based processing is that it can greatly increase the speed of a simulation. If you see double lines between blocks, the model uses frame-based processing.

Time and Spectral Analysis

This section deals with looking at the waveforms of simple sine wave and pulse wave.

Figure 2.1 shows the design for viewing the waveforms of toe signals.

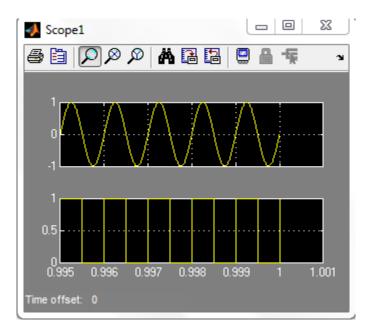


The blocks parameters:

📣 'Scope1' parameters 📃 🗆 🖾
General History
Axes
Number of axes: 2 🔲 floating scope
Time range: 0.05
Tick labels: bottom axis only
Sampling Decimation
OK Cancel Help Apply

Source Block Parameters: Pulse Generator				
Pulse type determines the computational technique used.				
Time-based is recommended for use with a variable step solver, while Sample-based is recommended for use with a fixed step solver or withi a discrete portion of a model using a variable step solver.				
Parameters				
Pulse type: Time based -				
Time (t): Use simulation time -				
Amplitude:				
1				
Period (secs):				
1e-3				
Pulse Width (% of period):				
50				
Phase delay (secs):				
0				
Interpret vector parameters as 1-D				
OK Cancel Help				

The results:



Spectrum Scope

This section deals with looking at the spectrum of simple waves. We first look at the spectrum of a simple sine wave.

Spectrum of a simple sine wave: - Figure 1.2 shows the design for viewing the

spectrum of a simple sine wave.

🙀 ex2 *				23
File Edit V	iew Simulation F	ormat Tools Help		
🗅 🖉 🖬	🚭 👗 🖻 💼	⇔⇒↑ Ω⊆	▶ ■ .1	- [i
	Sine Wave	ζ— Spec Sα	FT	
Ready 100%		ode45		

Parameters:

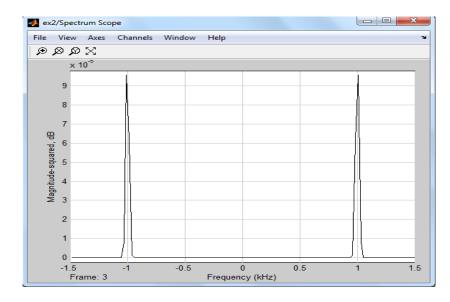
The sine wave block parameters as shown in

🙀 Source Block Parameters: Sine Wave				
Use the sample-based sine type if numerical problems due to running fo A 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):				
2*pi*1000				
Phase (rad):				
0				
Sample time:				
1/3000				
✓ Interpret vector parameters as 1-D				
OK Cancel Help				

The frequency domain spectrum is obtained through a buffered-FFT scope, which comprises of a Fast Fourier Transform of 128 samples which also has a buffering of 128 of them in one frame. The property block of the B-FFT is also displayed in figure 1.5.

駴 Sink Block Paramete	rs: Spectrum Scope		×		
Spectrum Scope					
Compute and display the mean-square spectrum or power spectral density of each input signal.					
Scope Properties	Display Properties	Axis Properties	Line Pr 🔹 🕨		
Parameters					
Spectrum units: N	/atts/Hertz		-		
Spectrum type: Tv	vo-sided ((-Fs/2Fs/2])	-		
Buffer input					
Buffer size: 128					
Buffer overlap: 0					
Treat Mx1 and unoriented sample-based signals as: One channel 🔹					
Window: Hann			•		
Window sampling: Symmetric					
Specify FFT length					
Number of spectral	averages: 128				
	OK Can	cel Help	Apply		

Result:



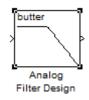
The below figure shows the frequency domain corresponding of the sine wave

Notes:

- From the property box of the B-FFT scope the axis properties can be changed and the Line properties can be changed. The line properties are not shown in the above. The Frequency range can be changed by using the frequency range pop down menu and so can be the y-axis the amplitude scaling be changed to either real magnitude or the dB (log of magnitude) scale. The upper limit can be specified as shown by the Min and Max Y-limits edit box. The sampling time in this case has been set to 1/3000.
- The sampling frequency of the B-FFT scope should match with the sampling time of the input time signal.
- Also as indicated above the FFT is taken for 128 points and buffered with half of them for an overlap.
- Calculating the Power: The power can be calculated by squaring the value of the voltage of the spectrum analyzer.
- Note: The signal analyzer if chosen with half the scale, the spectrum is the single-sided analyzer, so the power in the spectrum is the total power.
- Similar operations can be done for other waveforms like the square wave, triangular. These signals can be generated from the signal generator block.

Analog Filter design

This section we will use the analog filters implementation tool that founded at DSP Toolbox.



1. Design low pass filter

Now we will implement low pass filter with the following criteria

Type: low pass filter

Design Method : Butterworth

Order:8

Cut off frequency : 2kHz

🙀 Function Block Parameters: Analog Filter Design				
Analog Filter Design (mask) (link)				
Design one of several standard analog filters, implemented in state- space form.				
Parameters				
Design method: Butterworth				
Filter type: Lowpass				
Filter order:				
8				
Passband edge frequency (rad/s):				
2*pi*2000				
OK Cancel Help Apply				



2. Design High Pass Filter

In the same manner we will implement high pass filter with the following criteria : Type: high pass filter

Design Method : Butterworth

Order : 8

Cut off	frequency	1:	10kHz
---------	-----------	----	-------

Function Block Parameters: Analog Filter Design	
Analog Filter Design (mask) (link)	
Design one of several standard analog filters, implemented in state- space form.	
Parameters	
Design method: Butterworth	B
Filter type: Highpass 🔹	butter
Filter order:	Analog
8	Filter Desig
Passband edge frequency (rad/s):	
2*pi*10000	
OK Cancel Help Apply	

3. Design Band Pass Filter

Finally we will implement band pass filter with the following criteria :

Type: band pass filter

Design Method : Butterworth

Order : 8

lower Cut off frequency : 2KHz

upper cut off frequency :10kHz

Function Block Parameters: Analog Filter Design				
Analog Filter Design (mask) (link)				
Design one of several standard analog filters, implemented in state- space form.				
Parameters				
Design method: Butterworth 🔹				
Filter type: Bandpass 🔹				
Filter order:				
8				
Lower passband edge frequency (rad/s):				
2*pi*2000				
Upper passband edge frequency (rad/s):				
2*pi*10000				
OK Cancel Help Apply				

