# Appendix C: MATLAB's Simulink Tutorial

## C.1 Introduction

Readers who are studying MATLAB may want to explore the functionality and convenience of MATLAB's Simulink. Before proceeding, the reader should have studied Appendix B, the MATLAB Tutorial, including Section B.1, which is applicable to this appendix.

MATLAB's Simulink Version 7.4 and MATLAB Version 7.9(R2009b) are required in order to use Simulink.

The models described in this appendix, which are available at www.wiley.com/ college/nise, were developed on a PC using MATLAB Version 7.9 and Simulink Version 7.4. The code will also run on workstations that support MATLAB. Consult the MATLAB Installation Guide for your platform for minimum system hardware requirements.

Simulink is used to simulate systems. It uses a graphical user interface (GUI) for you to interact with blocks that represent subsystems. You can position the blocks, resize the blocks, label the blocks, specify block parameters, and interconnect blocks to form complete systems from which simulations can be run.

Simulink has block libraries from which subsystems, sources (that is, function generators), and sinks (that is, scopes) can be copied. Subsystem blocks are available for representing linear, nonlinear, and discrete systems. LTI objects can be generated if the Control System Toolbox is installed.

Help is available on the menu bar of the MATLAB Window. Under Help select **Product Help.** When the help screen is available, choose **Simulink** under the **Contents** tab. Help is also available for each block in the block library and is accessed either by right-clicking a block's icon in the **Simulink Library Browser** and selecting **Help for** . . . or by double-clicking the block's icon and then clicking the **Help** button. Finally, screen tips are available for some toolbar buttons. Let your mouse's pointer rest on the button for a few seconds to see the explanation.

## C.2 Using Simulink

The following summarize the steps to take to use Simulink. Section C.3 will present four examples that demonstrate and clarify these steps.

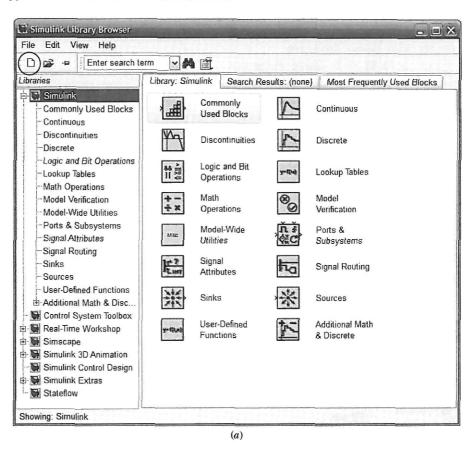


FIGURE C.1 MATLAB Window showing how to access Simulink. The Simulink Library Browser button is shown circled.

1. Access Simulink The Simulink Library Browser, from where we begin Simulink, is accessed by typing *simulink* in the MATLAB Command Window or by clicking on the Simulink Library Browser button on the toolbar, shown circled in Figure C.1.

In response, MATLAB displays the **Simulink Library Browser** shown in Figure C.2(a). We now create an **untitled** window, Figure C.2(b), by clicking on the **Create a new model** button (shown circled in Figure C.2(a)) on the toolbar of the **Simulink Library Browser**. You will build your system in this window. Existing models may be opened by clicking on the **Open a model** button on the **Simulink Library Browser** toolbar. This button is immediately to the right of the **Create a new model** button. Existing models may also be opened by selecting the **Current Folder** from the **Command Window Start** menu or the tab on the left side of the **Command Window as** shown in Figure C.1, selecting your file names, and then dragging them to the **MATLAB Command Window**.

2. Select blocks Figure C.2(*a*) shows the Simulink Library Browser from which all blocks can be accessed. The left-hand side of the browser shows major libraries, such as Simulink, as well as underlying block libraries, such as Continuous. The



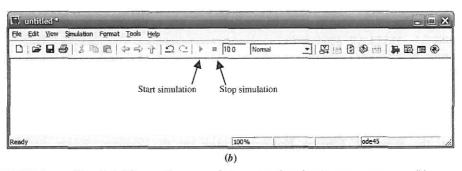


FIGURE C.2 a. Simulink Library Browser window showing the Create a new model button encircled b. resulting untitled model window

right-hand side of Figure C.2(a) also shows the underlying block libraries. To reveal a block library's underlying blocks, select the block library on the left-hand side or double-click the block library on the right-hand side. As an example, the **Continuous** library blocks under the **Simulink** major library are shown exposed in Figure C.3(a). Figures C.3(b) and C.3(c) show some of the **Sources** and **Sinks** library blocks, respectively.

Another approach to revealing the Simulink block library is to type *open\_ system ('simulink.mdl')* in the **MATLAB Command Window**. The window shown in Figure C.4 is the result. Double-clicking any of the libraries in Figure C.4

🕞 Simulink Library Browser			×
File Edit View Help			
🗋 🚅 🛥 Enter search t	erm 🔽 🚧 🛅	and the second second second	
Libraries	Library Simulink/Continuous	Search Results: (none) Most Frequently Used Blocks	
Commonly Used Blocks	du/dt> Derivative	$\frac{1}{s}$ Integrator $PD(s)$ PID Controller	
- Discontinuities - Discrete	PID Controller PID(R) (2DOF)	$x = A_{x+B_{u}}$ State-Space $\frac{1}{s+1}$ Transfer Fcn	
- Logic and Bit Operations - Lookup Tables - Math Operations	×2€ Transport Delay	Variable Time Variable Tool Delay Transport Delay	1
Model Verification Model-Wide Utilities	x (si) x(si) Zero-Pole		
Signal Attributes			
- Sinks - Sources			
User-Defined Functions     Additional Math & Disc     Gontrol System Toolbox			
Real-Time Workshop     Simscape			
Simulink 3D Animation			
Simulink Extras			
Showing: Simulink/Continuous	L		

(a)

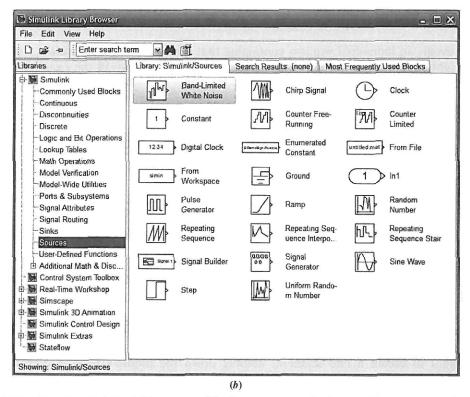


FIGURE C.3 Simulink block libraries: a. Continuous systems b. Sources (figure continues)

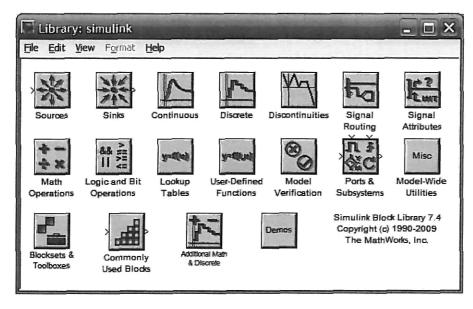
#### Appendix C MATLAB's Simulink Tutorial

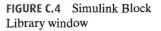
🖼 Simulink Library Browser		
File Edit View Help		
🗋 😂 🕫 Enter search te	erm 🖌 🙀 🛒	
Libraries	Library: Simulink/Sinks	Search Results: (none) Most Frequently Used Blocks
Libraries Simulink Commonly Used Blocks Continuous Discontinuities Discrete Lockup Tables Math Operations Model Venification Model Venification Model Vide Utilities Ports & Subsystems Signal Routing Sinks Sources User-Defined Functions B Additional Math & Disc Control System Toolbox Real-Time Workshop Simulink 3D Animation Simulink SD Animation	Display Coope	Search Results: (none) Most Frequently Used Blocks
🕀 🔛 Simulink Extras		
I- 🖬 Stateflow		
Showing. Simulink/Sinks		
		(c)

FIGURE C.3 (Continued) c. Sinks

reveals an individual window containing that library's blocks, equivalent to the right-hand side of the **Simulink Library Browser** as shown in the examples of Figure C.3.

- **3. Assemble and label subsystems** Drag required subsystems (blocks) to your model window from the browser, such as those shown in Figure C.3. Also, you may access the blocks by double-clicking the libraries shown in Figure C.4. You can position, resize, and rename the blocks. To position, drag with the mouse; to resize, click on the subsystem and drag the handles; to rename, click on the existing name, select the existing text, and type the new name. The text can also be repositioned to the top of the block by holding the mouse down and dragging the text.
- 4. Interconnect subsystems and label signals Position the pointer on the small arrow on the side of a subsystem, press the mouse button, and drag the resulting crosshair pointer to the small arrow of the next subsystem. A line will be drawn between the two subsystems. Blocks may also be interconnected by singleclicking the first block followed by single-clicking the second block while holding down the control key. You can move line segments by positioning the pointer on the line, pressing the mouse button, and dragging the resulting four-arrow pointer. Branches to line segments can be drawn by positioning the pointer where you want to create a line segment, holding down the mouse's right button, and dragging the resulting cross hairs. A new line segment will form. Signals can be labeled by double-clicking the line and typing into the resulting box. Finally, labels can be placed anywhere by double-clicking and typing into the resulting box.





- 5. Choose parameters for the subsystems Double-click a subsystem in your model window and type in the desired parameters. Some explanations are provided in the Block Parameters window. Press the Help button in the Block Parameters window for more details. The parameters can be read later without opening the block. Let your mouse's pointer rest on the block for a few seconds, and a screen tip will appear, identifying the block and listing its parameters. The information displayed in the screen tip first must be selected in the Block Data Tips Options in the model window's View menu. Explore other options by right-clicking on a block.
- 6. Choose parameters for the simulation Select Configuration parameters . . . under the Simulation menu in your model window to set additional parameters, such as simulation time. Press the Help button in the Configuration parameters window for more details.
- 7. Start the simulation Make your model window the active window. Double-click the Scope block (typically, the scope is used to view the simulation results) to display the Scope window. Select Start under the Simulation menu in your model window or click on the Start simulation icon on the toolbar of your model window as shown in Figure C.2(b). Clicking the Stop simulation icon will stop the simulation before completion.
- 8. Interact with the plot In the Scope window, using the toolbar buttons, you can zoom in and out, change axes ranges, save axis settings, and print the plot. Right-clicking on the Scope window brings up other choices.
- 9. Save your model Saving your model, by choosing Save under the File menu, creates a file with an .mdl extension, which is required.

## C.3 Examples

This section will present four examples of the use of Simulink to simulate linear, nonlinear, and digital systems. Examples will show the Simulink block diagrams as well as explain the settings of parameters for the blocks. Finally, the results of the simulations will be shown.

## Example C.1

#### Simulation of Linear Systems

Our first example develops a simulation of three linear systems to compare their step responses. In particular, we solve Example 4.8 and reproduce the responses shown in Figure 4.24. Figure C.5 shows a Simulink block diagram formed by following Steps 1 through 5 in Section C.2 as follows:

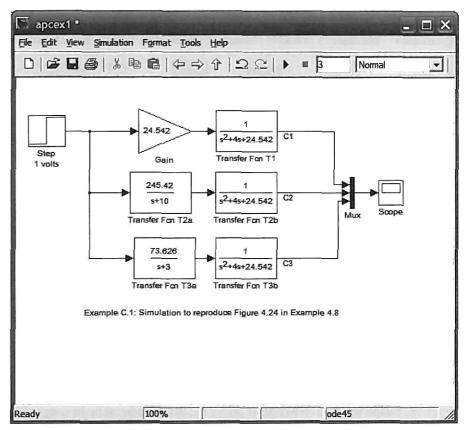


FIGURE C.5 Simulink block diagram for Example C.1

Access Simulink; select, assemble, and label subsystems The source is a 1-volt step input, obtained by dragging the **Step** block from the **Simulink Library Browser** under **Sources** to your model window.

The first system, T1, consists of two blocks, **Gain** and **Transfer Fcn**. Gain is obtained by dragging the **Gain** block from the **Simulink Library Browser** under **Math Operations** to your model window. Transfer function, T1, is obtained by dragging the **Transfer Fcn** block from the **Simulink Library Browser** under **Continuous** to your model window. Systems T2 and T3 are created similarly.

The three output signals, C1, C2, and C3, are multiplexed for display into the single input of a scope. The Mux (multiplexer) is obtained by dragging the **Mux** block from the **Simulink Library Browser** under **Signal Routing** to your model window.

The sink is a scope, obtained by dragging the **Scope** block from the **Simulink Library Browser** under **Sinks** to your model window.

Alternatively, all blocks can be dragged from the **Library: simulink** window shown in Figure C.4. The **Mux** can be found under **Signal Routing** in the **Library: simulink** window.

The labels for the blocks can be changed to those shown in Figure C.5 by following Step 3 in Section C.2.

**Interconnect subsystems and label signals** Follow Step 4 to interconnect the subsystems and label the signals. You must set the mux's parameters before the wiring can be completed. See the next paragraph.

Choose parameters for the subsystems Let us now set the parameters of each block using Step 5. The **Block Parameters** window for each block is accessed by double-clicking the block on your model window. Figure C.6 shows the **Block Parameters** windows for the 1 volt step input, gain, transfer function 1, and mux. Set the parameters to the required values as shown.

arameters tep time:	tep			
tep time: ) nitial value: nal value: L ample time: ) Interpret vector parameters as 1-D	output a step.		State 199	
) nitial value: nal value: L ample time: ) ] Interpret vector parameters as 1-D	arameters			
nitial value: nal value: L ample time: ) ] Interpret vector parameters as 1-D	itep time:			
nal value: L ample time: ) ] Interpret vector parameters as 1-D	0			
nal value: L ample time: ) ] Interpret vector parameters as 1-D	nitial value:			
ample time: ) ] Interpret vector parameters as 1-D	0			
ample time: ) ] Interpret vector parameters as 1-D	inal value:			
] Interpret vector parameters as 1-D	1			
Interpret vector parameters as 1-D	ample time:			
	0			
	Interpret vector pa	rameters as	1-D	
	7 Enable zero-crossin	a detection		
	Contraction of the second	OK	Cancel	He

FIGURE C.6 Block parameters windows for a. 1 volt step source; (figure continues)

#### Appendix C MATLAB's Simulink Tutorial

🗂 Fun	ction Block Parameters: Gain	×
Gain		· · · · · · · · · · · · · · · · · · ·
Element	t-wise gain (y = K.*u) or matrix gain (y = K*u or y = u*K).	
Main	Signal Attributes Parameter Attributes	
Gain:		
24.542		
Multiplic	ation: Element-wise(K.*u)	M
Sample	time (-1 for inherited):	
-1		
		(contraction like)
0	QK Cancel Help	Αρρίγ
e an on far somere	(b)	

📱 Function Block Parameters: Transfer Fcn T1	2
Transfer Fcn	
The numerator coefficient can be a vector or matrix expression. The denominator coefficient must be a vector. The output width equals the number of rows in the numerator coefficient. You should specify the coefficients in descending order of power of s.	s
Parameters	
Numerator coefficients:	
1	_
Denominator coefficients:	
[1 4 24.542]	
Absolute tolerance:	
auto	
State Name: (e.g., 'position')	
ч	
OK Cancel Help Apply	
(c)	

FIGURE C.6 b. gain; c. transfer function 1; (figure continues)

E Function B	ock Paramete	rs: Mux		
Multiplex scalar o	r vector signals.			
Parameters			*	
Number of input	:			
3				
Display option:	par			
2		Cance	a Hel	p Apply

FIGURE C.6 (Continued) d. mux

The scope requires further explanation. Double-clicking the **Scope** block in your model window accesses the scope's display, Figure C.7(a).

(d)

Clicking the **Parameters** icon on the **Scope** window toolbar, shown in Figure C.7(a), accesses the **'Scope' parameters** window as shown in Figure C.7(b). The **'Scope' parameters** window contains two tabs, **General** and **Data history**, as shown in Figure C.7(b) and (c), respectively.

Scope	
Parameters	
2	🥑 'Scope' parameters
1.8	General Data history Tip: try right clicking on axes
1.6	Axes
1.4	Number of axes: 1 floating scope
1.2	Time range: auto
1	Tick labels: bottom axis only
0.8	Sampling
0.6	Decimation 🖌 1
0.4	
0.2	OK Cancel Help Apply
	(b)
0 0.5 1 1.5 2 2.5 3	v- 2
Time offset: 0	
(a)	
FIGURE C.7 Windows for the scope: a. Scope; b. 'Scope' param	neters, General tab; (figure continues)

✓ 'Scope' parameters					
General Data history Tip: try right o	licking on axes				
✓ Limit data points to last: 5000		🧈 'Scope' p	roperties	: axis 1 (	<u>-                                    </u>
Save data to workspace		Y-min: 0		Y-max:2	
Variable name: ScopeData		Title ('% <sigr< td=""><td>alLabel&gt;' re</td><td>placed by signal r</td><td>name):</td></sigr<>	alLabel>' re	placed by signal r	name):
Format: Array	~		ОК	Cancel (	Apply
OK Cancel Help	Apply			(d)	
(c) FIGURE C.7 (Continued) c. 'Scope' paramet	<b>ers, Data history</b> tab	o; d. 'Scope' pro	perties: axi	s 1	
ters. Figure	ameters for the si C.8 shows the re neters, the simulat	sulting Confi	guration I	Parameters with	
	neters: apcex1/Configuration (Ac	tive)	-	the subscription of the local division of the	<u>×</u>
Selecti Solice - Solice - Data Import/Export - Data Import/Export - Optimization	Simulation time Start time: 0.0		Stop time: 3		Î
Diagnostics	Solver options Type: Variable-step		Solver:	ode45 (Dormand-Prince)	V
- Data Validity - Type Conversion	Max step size: auto		Relative tolerance:	le-3	
Connectivity Compatibility Model Referencing	Min step size: auto		Absolute tolerance:	1e-6	
- Saving Hardware Implementati	Initial step size: auto	251	Shape preservation:	Insable all	<u> </u>
- Model Referencing ⊖-Simulation Target	Tasking and sample time options				
- Symbols Custom Code	Tasking mode for periodic sample		SingleTasking		
	Automatically handle rate to Higher priority value indicate				
	Zero-crossing options Zero-crossing control: Use locz	l settings	Algorithm:	Nonadaptive	~
	Time tolerance: 10*128		Signal threshold	plant and an owner the second se	
	Number of consecutive zero cro	ssings:		1000	
FIGURE C.8 Configuration					(•)
Parameters window for Solver tab			<u>o</u> k	<u>Cancel</u> <u>Help</u>	<u>Apply</u>

#### Appendix C MATLAB's Simulink Tutorial

**Start the simulation** Now run the simulation by following Step 7. Figure C.9 shows the result in the **Scope** window. Plots are color coded in the order in which they appear at the mux input as follows: yellow, magenta, cyan, red, green, and dark blue. If the mux has more inputs, the colors recycle.

**Interact with the plot** The toolbar of the **Scope** window shown in Figure C.9 has several buttons that can be used to interact with the plot. Let us summarize the function and operation of each, starting with the left-most button:

- Button 1 executes a plot print.
- Button 2 has already been explained and is used to set scope parameters.
- **Button 3** permits zooming into the plot in both the x and y directions. Press the button and drag a rectangle over the portion of the curve you want to expand.
- **Button 4** allows zooming in the *x* direction only. Drag a horizontal line over the plot covering the extent of *x* you want to expand.
- **Button 5** allows zooming in the *y* direction only. Drag a vertical line over the plot covering the range of *y* you want to expand.
- Button 6 autoscales axis for use after zooming.
- Button 7 saves current axis settings.
- Button 8 restores saved axis settings.

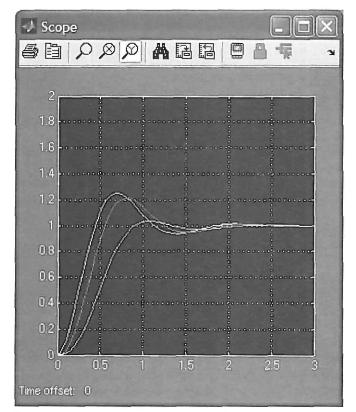
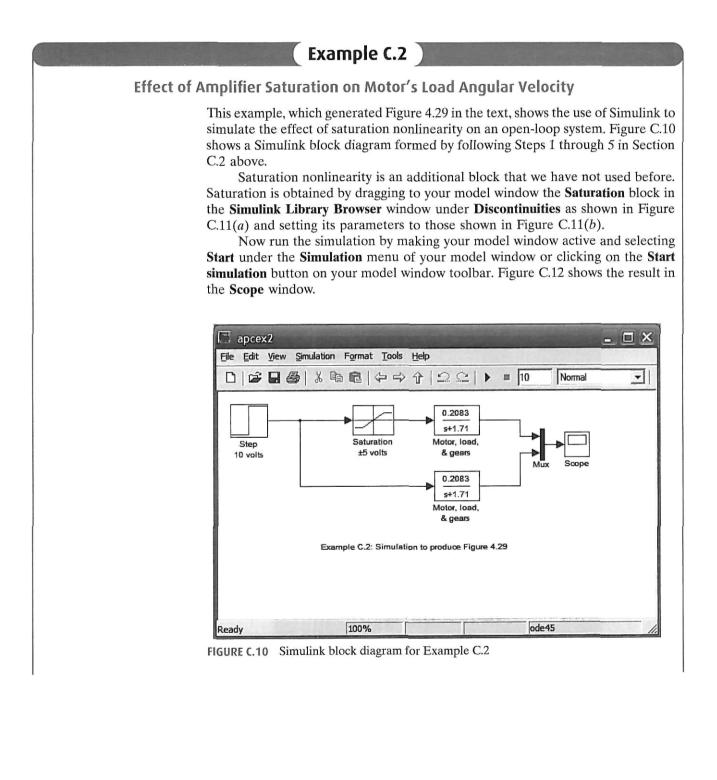


FIGURE C.9 Scope window after Example C.1 simulation stops

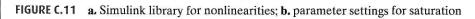
Button 9 toggles floating scope. It must be turned off to use zooming. See documentation for use of floating scopes.

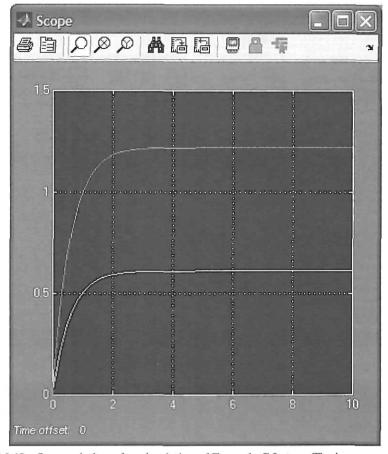
Button 10 toggles lock for current axis selection.

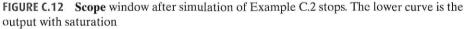
Button 11 allows selection of signals to view when using floating scope.



	erm MA					
branies	Library: Simuli	nk/Discontinuitie	s Search	Results: (none)	Most Frequ	ently Used Blocks
- Commonly Used Blocks - Continuous		Jacklash		Coulomb & Vi- scous Friction	册	Dead Zone
<ul> <li>Discontinuities</li> <li>Discrete</li> <li>Logic and Bit Operations</li> </ul>		Dead Zone Dynamic	ł	Hit Crossing	┢	Quantizer
- Lookup Tables - Math Operations		Rate Limiter		Rate Limiter Dynamic		Relay
Model Verification     Model-Wide Utilities     Ports & Subsystems     Signal Attributes     Signal Attributes     Sinks     Sources     User-Defined Functions     Control System Toolbox     Real-Time Workshop     Simulink 3D Animation     Simulink Control Design     Simulink Extras     Statellow		Saturation	<u>,</u> ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	Saturation Dynamic		Wrap To Zero
howing: Simulink/Discontinuitie	3	(a	2)			
	Paramete	ers: Satur	ation ±	· · · · · ·		\$
Function Block Saturation Limit input signal to th	Paramete ne upper and	ers: Satur	ation ±	· · · · · ·		3
Function Block	Paramete ne upper and	ers: Satur	ation ±	· · · · · ·		5
Function Block Saturation Limit input signal to th Main Signal Attrit	Paramete ne upper and	ers: Satur	ation ±	· · · · · ·		3
Function Block Saturation Limit input signal to th Main Signal Attrik Upper limit:	Paramete ne upper and	ers: Satur	ation ±	· · · · · ·		2
Function Block Saturation Limit input signal to th Main Signal Attri Upper limit: 5	Paramete ne upper and	ers: Satur	ation ±	· · · · · ·		2
Function Block Saturation Limit input signal to th Main Signal Attrit Upper limit: 5 Lower limit:	Paramete e upper and putes	ers: Satur	ation ±	· · · · · ·		5
Function Block Saturation Limit input signal to th Main Signal Attrit Upper limit: 5 Lower limit: -5	Paramete ne upper and putes	ers: Satur	ation ±	· · · · · ·		3
Function Block Saturation Limit input signal to th Main Signal Attrit Upper limit: 5 Lower limit: -5 I Treat as gain when Fnable zero-crossi	Paramete e upper and outes	ers: Satur	ation ±	Jes.		5
Function Block Saturation Limit input signal to th Main Signal Attrit Upper limit: 5 Lower limit: -5 Treat as gain when	Paramete e upper and outes	ers: Satur	ation ±	jes.		\$





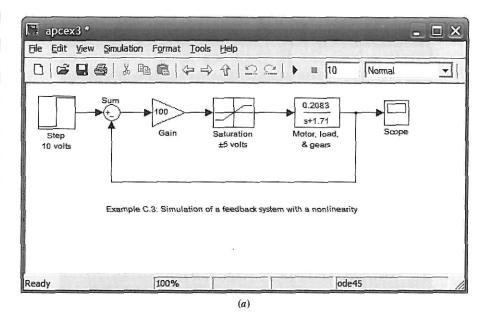


## Example C.3

### Simulating Feedback Systems

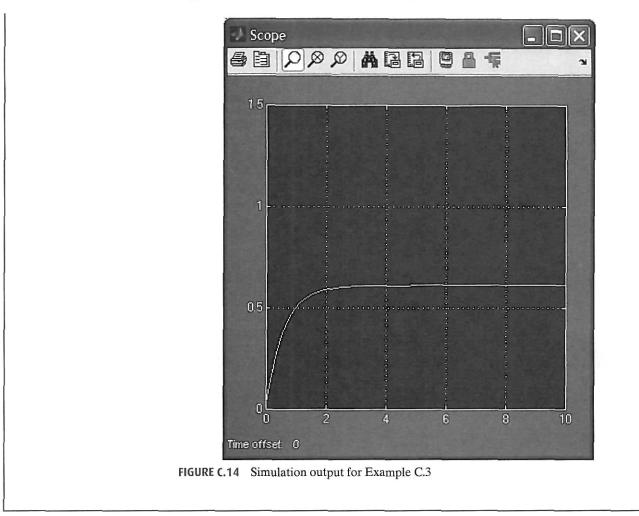
Simulink can be used for the simulation of feedback systems. Figure C.13(a) is an example of a feedback system with saturation.

In this example, we have added a feedback path (see Step 4 in Section C.2) and a summing junction, which is obtained by dragging the **Sum** block from the **Simulink Library Browser**, contained in the **Math Operations** library, to your model window. The **Function Block Parameters: Sum** window, Figure C.13(*b*), shows the parameter settings for the summer. You can set the shape as well as set the plus and minus inputs. In the list of signs, the "|"symbol signifies a space. We place it at the beginning to start the signs at "nine o'clock," conforming to our standard symbol, rather than at "12 o'clock." The result of the simulation is shown in Figure C.14.



C Function Block Parameters: Sum	>
Sum	-6
Add or subtract inputs. Specify one of the following: a) string containing + or - for each input port,   for spacer between ports (e.g. ++ -  ++) b) scalar, >= 1, specifies the number of input ports to be summed. When there is only one input port, add or subtract elements over all dimensions or one specified dimension	
Main Signal Attributes	
Icon shape: round	
List of signs:	
[+-	716
Sample time (-1 for inherited):	and a
-1	71
	16
OK Cancel Help Appl	y Y
(b)	

**FIGURE C.13 a.** Simulation block diagram for a feedback system with saturation; **b.** block parameter window for the summer



## Example C.4

### Simulating Digital Systems

This example demonstrates two methods of generating digital systems via Simulink for the purpose of simulation, as shown in Figure C.15.

The first approach uses a linear transfer function cascaded with a **Zero-Order Hold** block obtained from the **Simulink Library Browser** under the **Discrete** block library, shown on the right-hand side of Figure C.16. The second method uses a discrete transfer function also obtained from the **Simulink Library Browser** under the **Discrete** block library. The remainder of the block diagram was obtained by methods previously described.

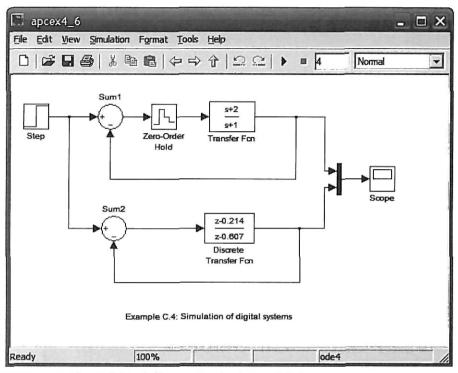


FIGURE C.15 Simulink block diagram for simulating digital systems two ways

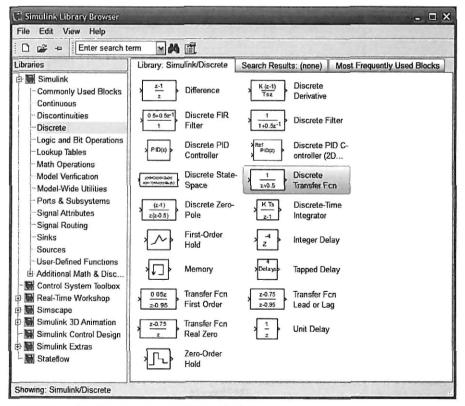


FIGURE C.16 Simulink library of discrete blocks

#### Appendix C MATLAB's Simulink Tutorial

Zero-order hold.					
Parameters					
Sample time (-1 fo	r inharitad):				
0.5	r innenteuj.				
		<u> </u>			A
	<u> </u>	Cancel	<u>H</u> e	P	Apply
		<i>(a)</i>			
		<i>(a)</i>			
Function Block Dara	notors: Discrete Tran				
Function Block Para	neters: Discrete Trar				
Discrete Transfer Fon Implement a z-transform tra	sfer function. Specify the	nsfer Fcn numerator and denom		ts in descend	ing powers of z.
Discrete Transfer Fcn	sfer function. Specify the	nsfer Fcn numerator and denom		ts in descend	ing powers of z.
Discrete Transfer Fon Implement a z-transform tran The order of the denominato The numerator coefficients n	sfer function. Specify the must be greater than or ust either be a scalar b0 o	nsfer Fcn numerator and denom equal to the order of t	he numerator,		
Discrete Transfer Fon Implement a z-transform trai The order of the denominato The numerator coefficients m a scalar a0 or a vector [a0 a	sfer function. Specify the must be greater than or ust either be a scalar b0 o a2].	nsfer Fcn numerator and denom equal to the order of t	he numerator,		
Discrete Transfer Fon Implement a z-transform tran The order of the denominato The numerator coefficients n a scalar a0 or a vector [a0 a Main Data Types St	sfer function. Specify the must be greater than or ust either be a scalar b0 o a2]. te Attributes	nsfer Fcn numerator and denom equal to the order of t	he numerator,		
Discrete Transfer Fcn Implement a z-transform trai The order of the denominato The numerator coefficients n a scalar a0 or a vector [a0 a Main Data Types Sta Numerator coefficients:	sfer function. Specify the must be greater than or ust either be a scalar b0 o a2]. te Attributes [1-0.214]	nsfer Fcn numerator and denom equal to the order of t	he numerator,		
Discrete Transfer Fon Implement a z-transform trai The order of the denominato The numerator coefficients n a scalar a0 or a vector [a0 a Main Data Types Sta Numerator coefficients: Denominator coefficients:	sfer function. Specify the must be greater than or ust either be a scalar b0 o a2]. te Attributes [1-0.214] [1-0.607]	nsfer Fcn numerator and denom equal to the order of t	he numerator,		
Discrete Transfer Fon Implement a z-transform tran The order of the denominato The numerator coefficients n a scalar a0 or a vector [a0 a Main Data Types St Numerator coefficients: Denominator coefficients: Initial states:	In the second se	nsfer Fcn numerator and denom equal to the order of t	he numerator,		
Discrete Transfer Fon Implement a z-transform trai The order of the denominato The numerator coefficients n a scalar a0 or a vector [a0 a Main Data Types Sta Numerator coefficients: Denominator coefficients:	sfer function. Specify the must be greater than or o ust either be a scalar b0 o a2]. te Attributes [1 -0.214] [1 -0.607] 0 0.5	numerator and denom equal to the order of t r a vector [b0 b1 b2	he numerator,		

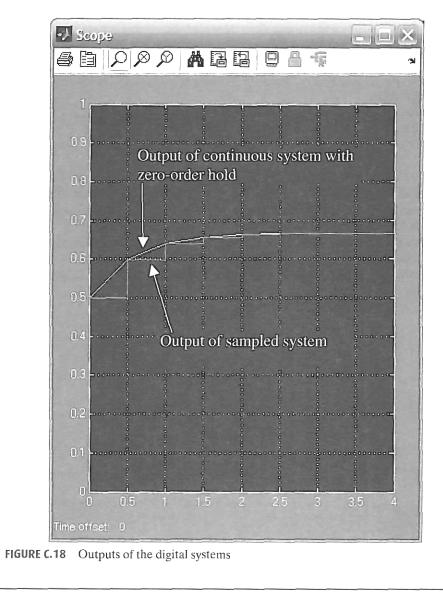
(b)

FIGURE C.17 Function Block parameter windows for: a. Zero-Order Hold block; b. Discrete Transfer Fcn block

The block parameters for the **Zero-Order Hold** and **Discrete Transfer Fcn** blocks are set as shown in Figures C.17(a) and (b), respectively.

Select **Configuration parameters**... under the **Simulation** menu in your model window and set the simulation stop time to 4 seconds, the type **to fixedstep**, and the solver to **ode4 (Runge-Kutta)**. The result of the simulation is shown in Figure C.18.

Summary



# Summary

This appendix explained Simulink, its advantages, and how to use it. Examples were taken from Chapters 4, 5, and 13 and demonstrated the use of Simulink for simulating linear, nonlinear, and digital systems.

The objective of this appendix was to familiarize you with the subject and get you started using Simulink. There are many blocks, parameters, and preferences that could not be covered in this short appendix. You are encouraged to explore and expand your use of Simulinkby using the on-screen help that was explained earlier. The references in the Bibliography of this appendix also provide an opportunity to learn more about Simulink.



- The MathWorks. Control System Toolbox <sup>TM</sup> 8 Getting Started Guide. The MathWorks. Natick, MA. 2000–2009.
- The MathWorks. Control System Toolbox <sup>TM</sup> 8 User's Guide. The MathWorks. Natick, MA. 2001–2009.
- The MathWorks. MATLAB<sup>(R)</sup> 7 Getting Started Guide. The MathWorks. Natick, MA. 1984–2009.

The MathWorks. MATLAB<sup>(1)</sup> 7 Graphics. The MathWorks. Natick, MA. 1984–2009.

- The MathWorks. MATLAB® 7 Mathematics. The MathWorks. Natick, MA. 1984-2009.
- The MathWorks. *MATLAB® 7 Programming Fundamentals*. The MathWorks. Natick, MA. 1984–2009.
- The MathWorks. Simulink<sup>®</sup> 7 Getting Started Guide. The MathWorks. Natick, MA. 1990–2009.

The MathWorks. Simulink® 7 User's Guide. The MathWorks. Natick, MA. 1990-2009.