Appendix C
Simulink Tutorial

C.1 Introduction

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

Simulink Version 9.0(R2017b) and MATLAB Version 9.3(R2017b) are required in order to use Simulink. In addition, if you wish to pursue the design of PID controllers discussed in Section C.4, you will need the Simulink Control Design Version 5.0(R2017b) add-on.

The models described in this appendix are available in the Control Systems Engineering Toolbox folder. 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 (i.e., function generators), and sinks (i.e., 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 at the top of the **MATLAB R2017b** window. Click the circled question mark and select **Simulink**. 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.

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

   In response, MATLAB displays the **Simulink Start Page** shown in Figure C.2(*a*). We now create an **untitled** window, Figure C.2(*b*), by clicking on the **model** button (shown circled in Figure C.2(*a*)) on the **Simulink Blank Model Start Page**. You will build your system in this window. Existing models may be opened by clicking on the **Open** file button on the **Simulink Start Page** toolbar. This button is in the left-hand column of the Simulink **Start Page**. Existing models may also be opened from the **Searchbar** below the **Toolstrip** on the **MATLAB** window.

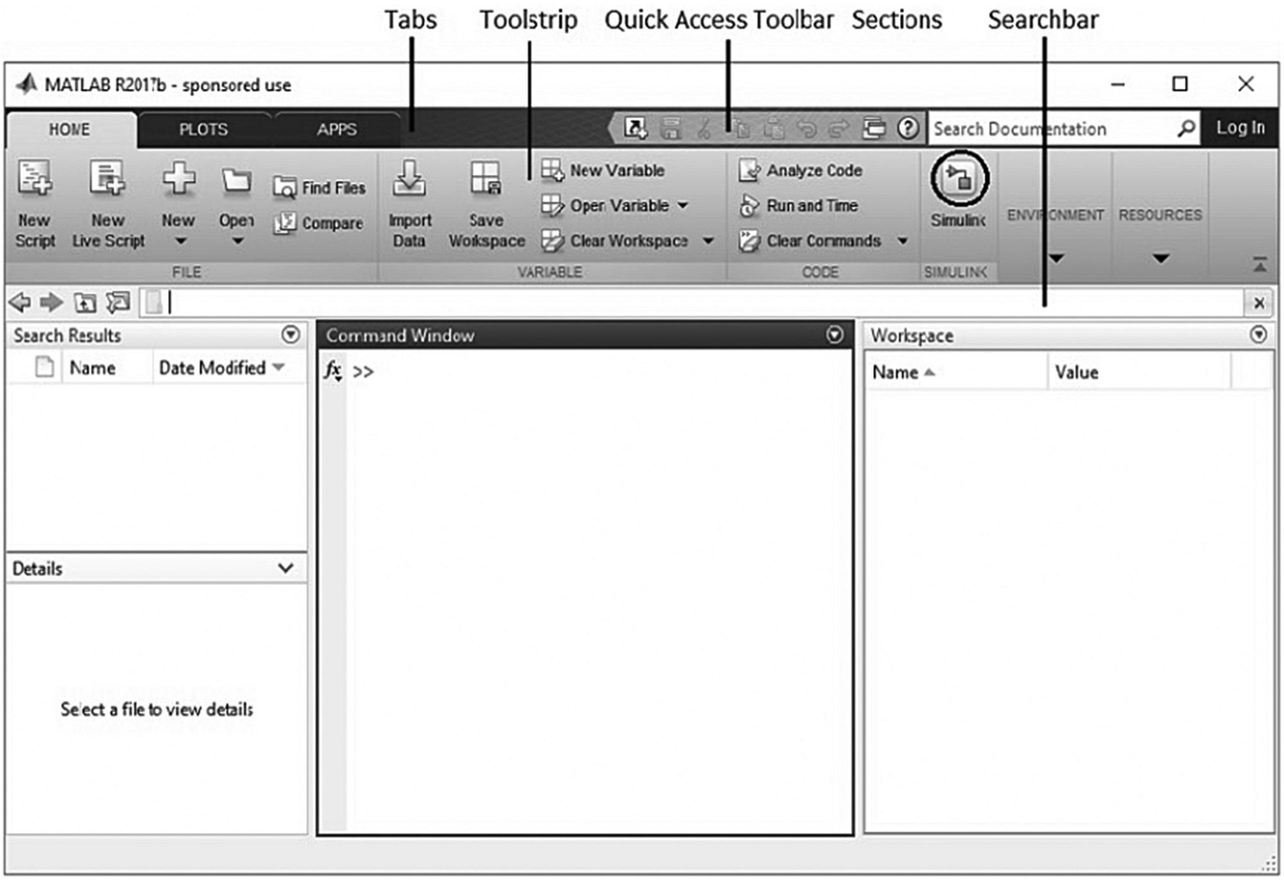
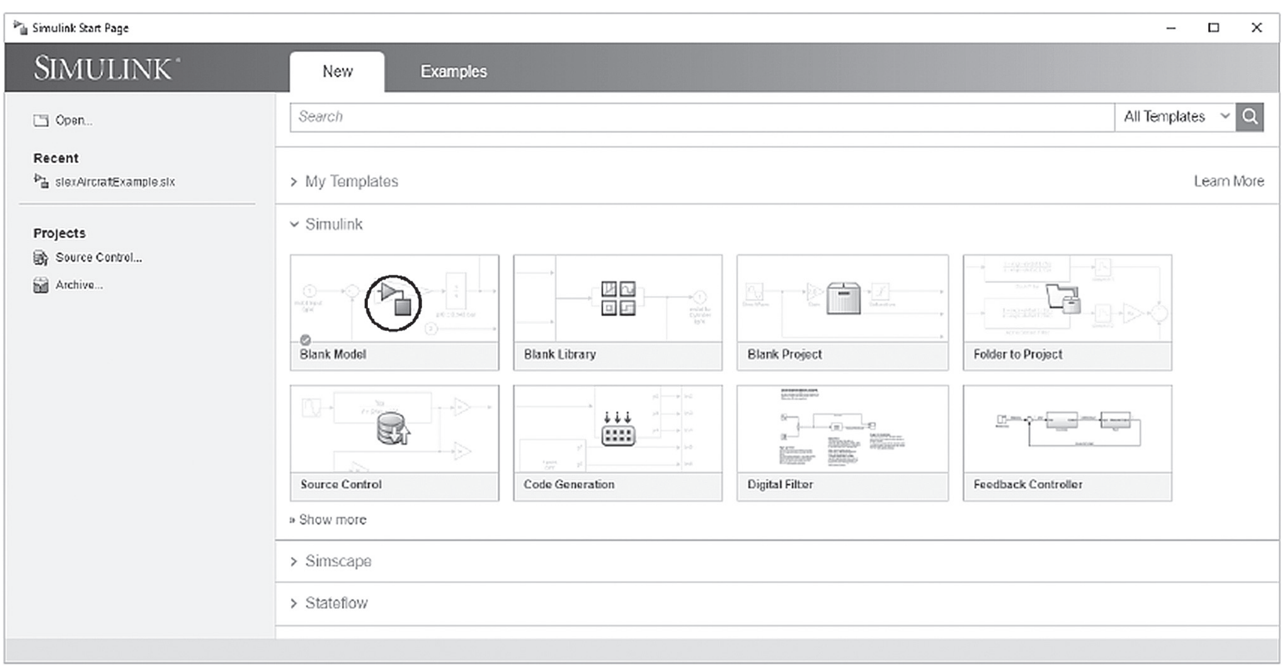
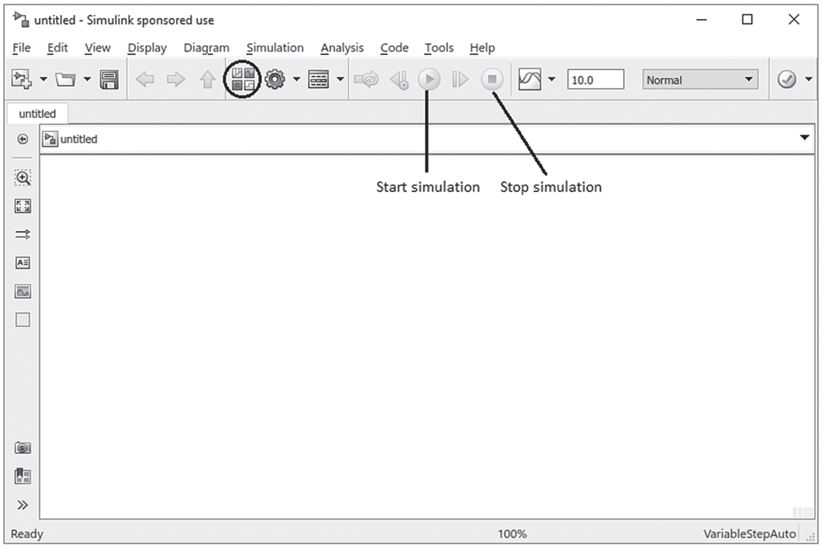


FIGURE C.1 **MATLAB** Window showing how to access Simulink. The **Simulink Start Page** button is shown circled



(*a*)

FIGURE C.2 **a. Simulink Start Page** window showing the **Blank Model** button encircled; (*figure continues*)



(*b*)

FIGURE C.2 (*continued*) **b.** resulting **untitled** model window

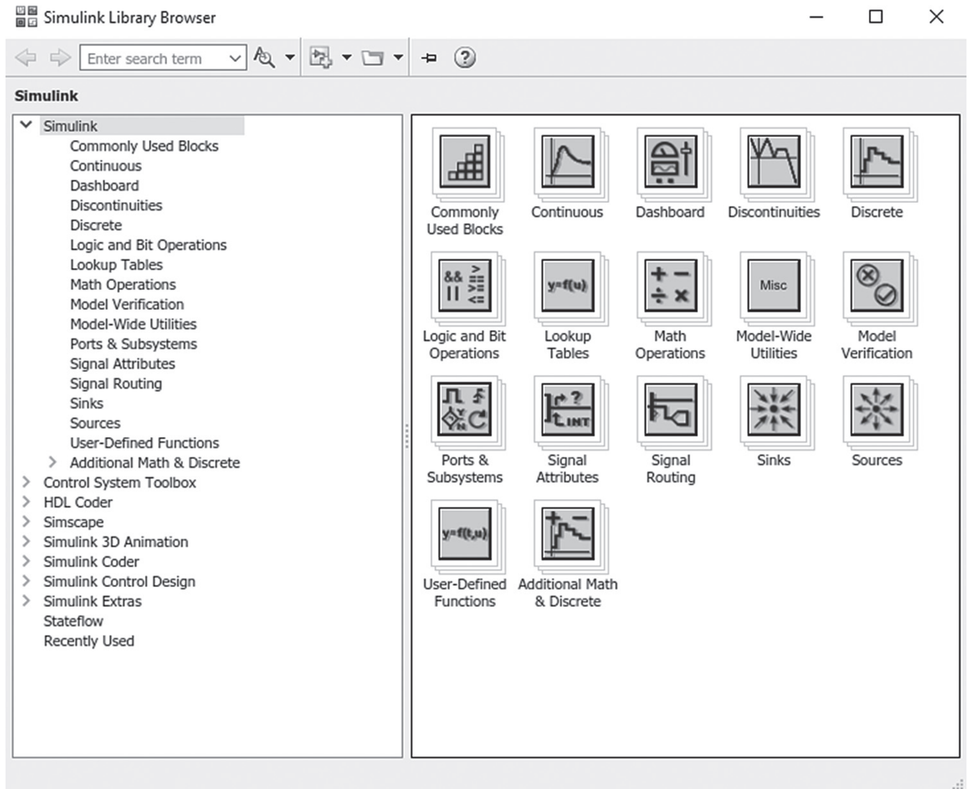
2. **Select blocks** Press the Simulink **Library Browser** button shown encircled in Figure C.2(*b*). Figure C.3(*a*) shows the **Simulink Library Browser** from which all blocks can be accessed. The left-hand side of the browser shows major libraries, such as **Simulink**, as well as underlying block libraries, such as **Continuous**. The right-hand side of Figure C.3(*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(*b*). Figure C.3(*c*) and C.3(*d*) shows some of the **Sources** and **Sinks** library blocks, respectively.

   Another approach to revealing the Simulink block library is to type *open_system (simulink)* 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 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.

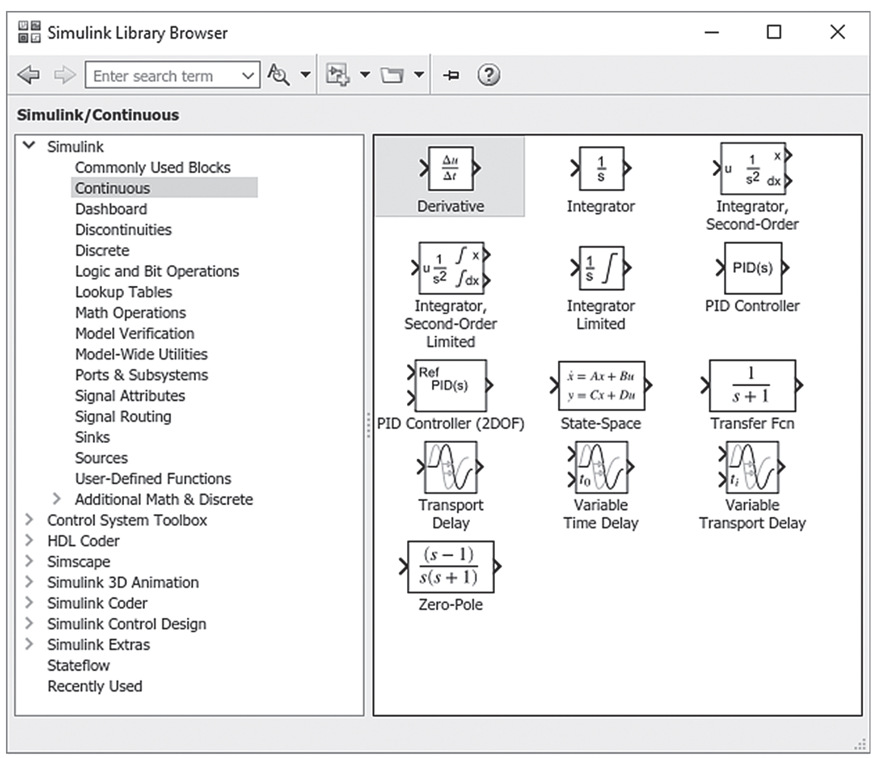
3. **Assemble and label subsystems** Drag required subsystems (blocks) to your model window from the browser, such as those shown in Figure C.3. Also, you may access the blocks by double-clicking the libraries shown in Figure C.4. You can position, resize, and rename the blocks. To position, drag with 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.

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 cross-hair pointer to the small arrow of the next subsystem. A line will be drawn between the two subsystems. Blocks may also be interconnected by single-clicking 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. 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. 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**

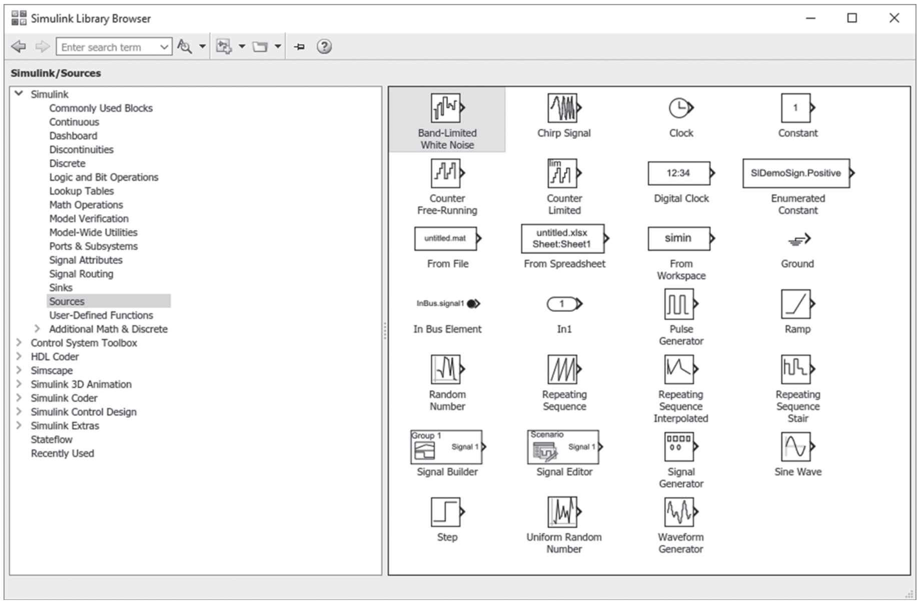


(*a*)

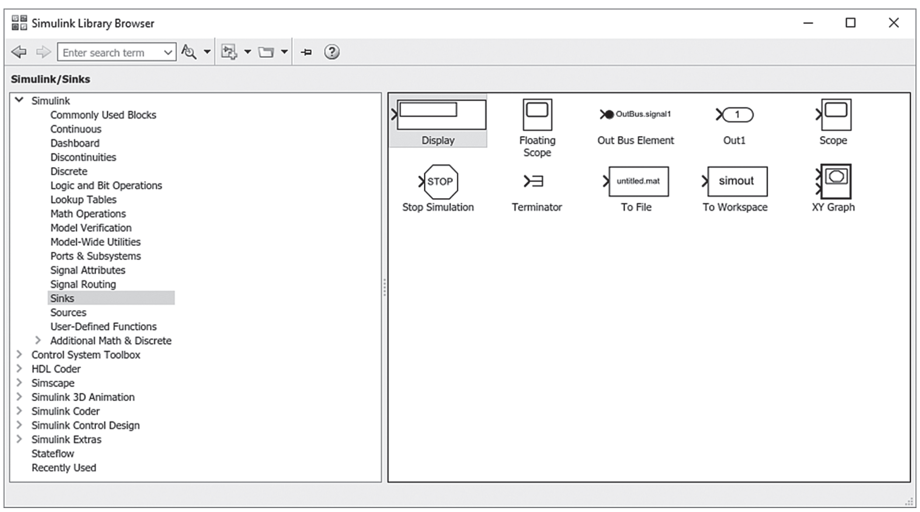


(*b*)

FIGURE C.3 Simulink block libraries: **a. Simulink Library Browser; b.** Continuous systems; (*figure continues*)



(*c*)



(*d*)

FIGURE C.3 (*continued*) **c.** Sources **d.** Sinks

Parameters window. Press the Help button in the **Block Parameters** window for more details. Explore other options by right-clicking on a block.

6. **Choose parameters for the simulation** Select **Model 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 **Run** under the **Simulation** menu in your model window or click on the **Run** icon on the toolbar of your model window as shown in Figure C.2(*b*). Clicking the **Stop** 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.

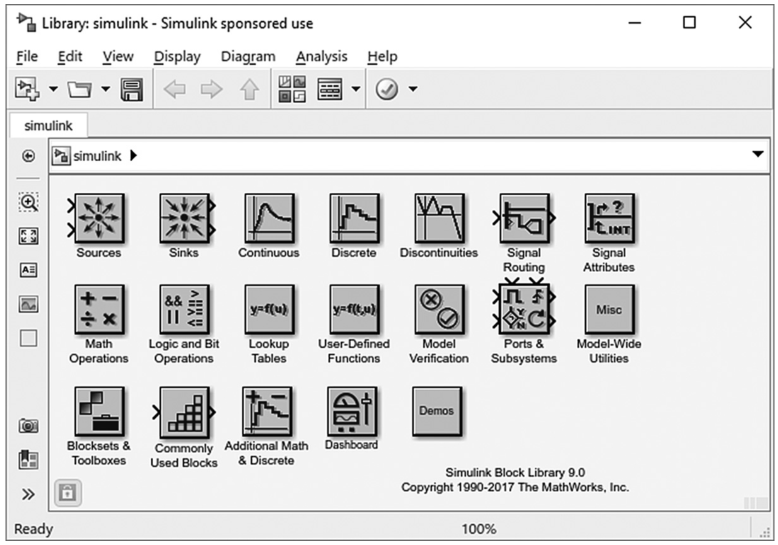


FIGURE C.4 Simulink Block Library window

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:

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.

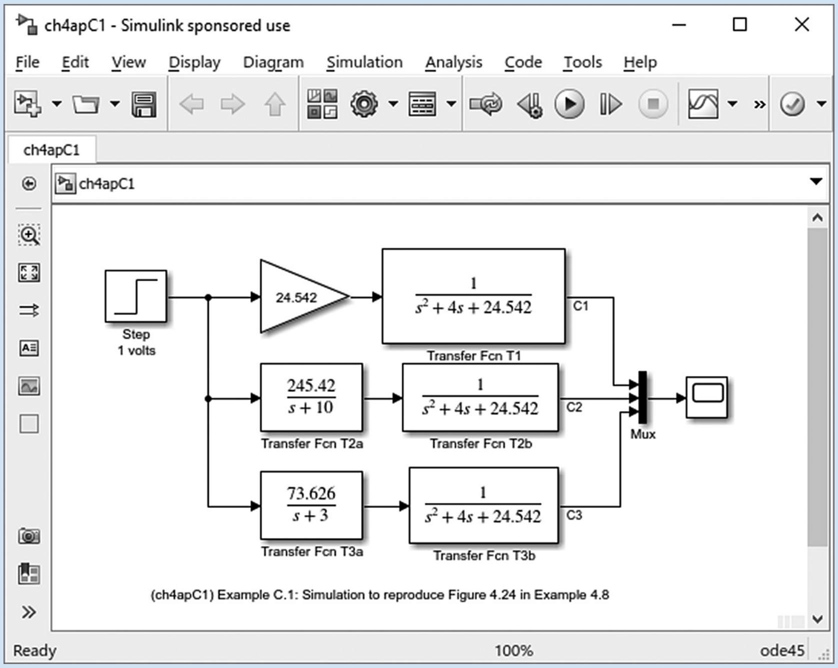
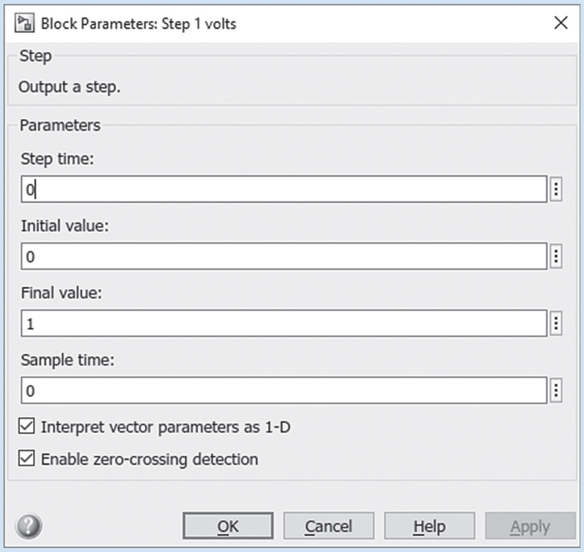


FIGURE C.5 Simulink block diagram for Example C.1

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

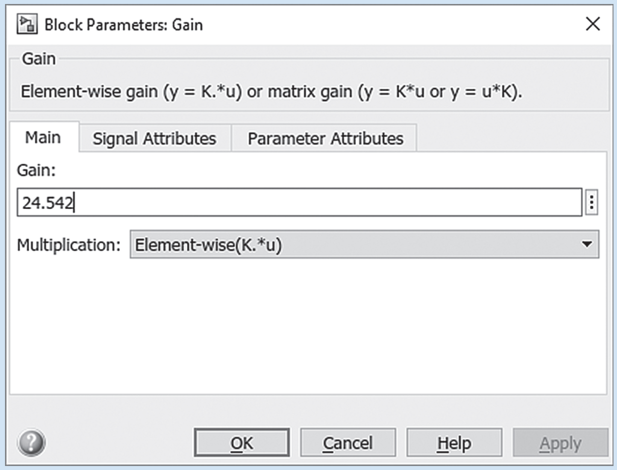
Interconnect subsystems and label signals Follow Step 4 to interconnect the subsystems and label the signals. You must set the mux's parameters before the wiring can be completed. See the next paragraph.

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.

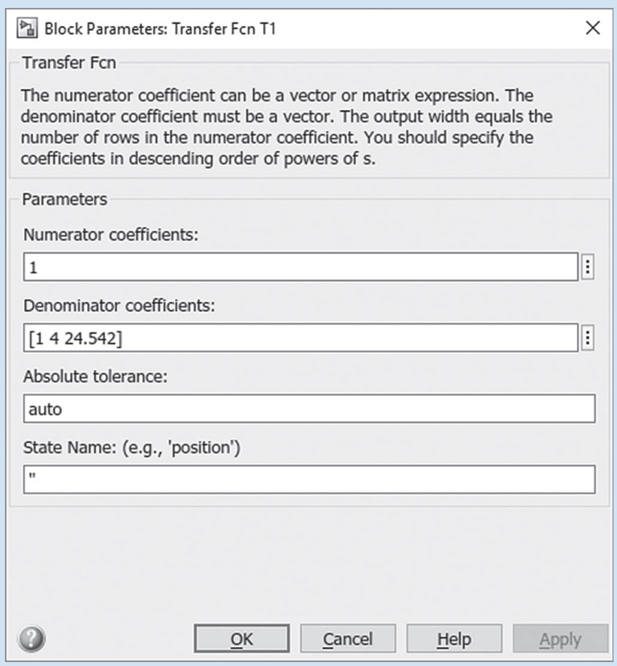


(*a*)

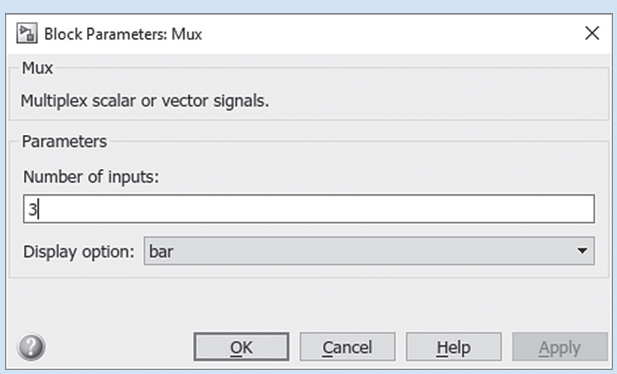
FIGURE C.6 **Block parameters** windows for **a.** 1-Volt step source; (*figure continues*)



(*b*)



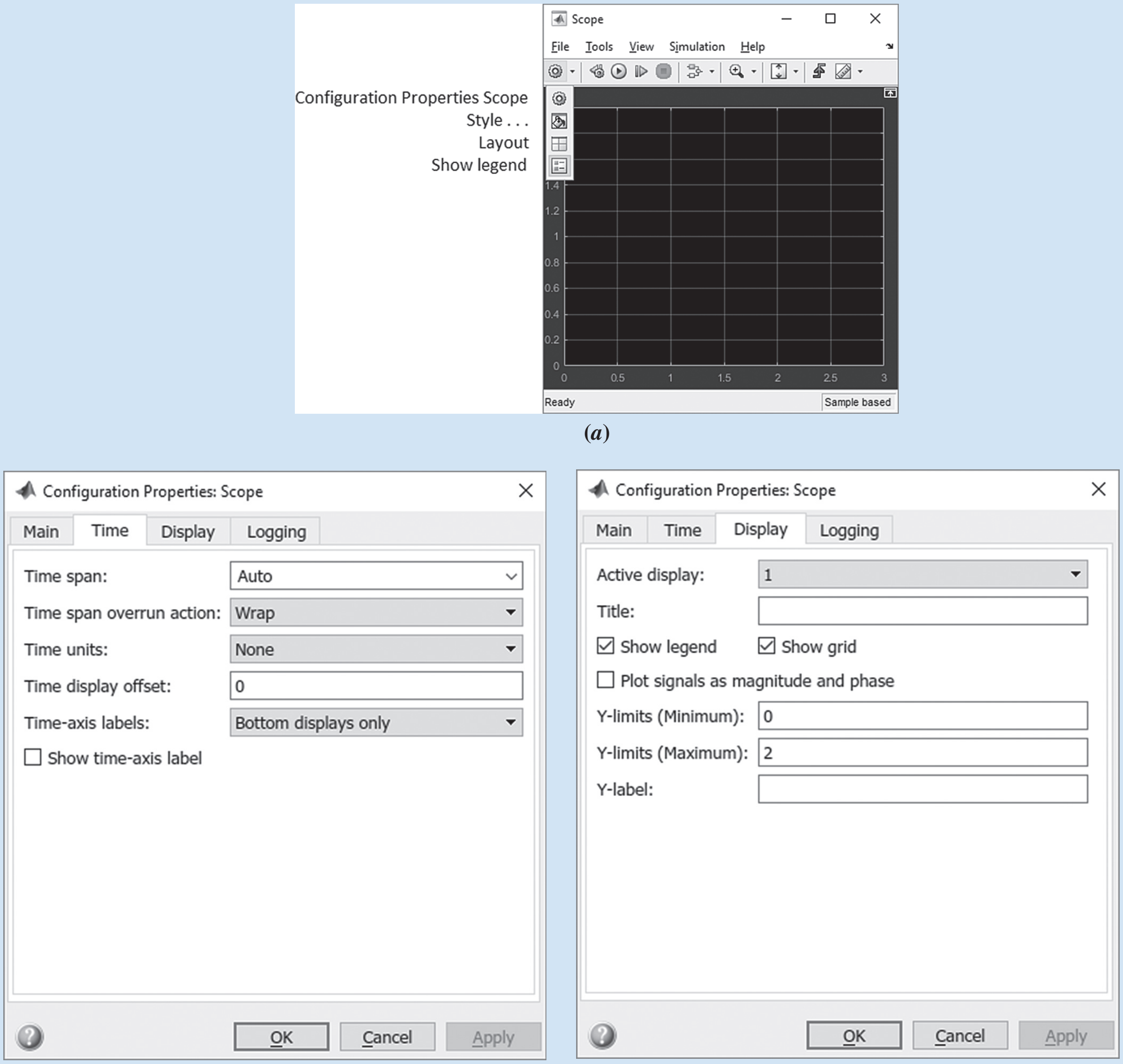
(*c*)



(*d*)

FIGURE C.6 (*continued*) **b.** gain; **c.** transfer function 1; **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*).



(*a*)

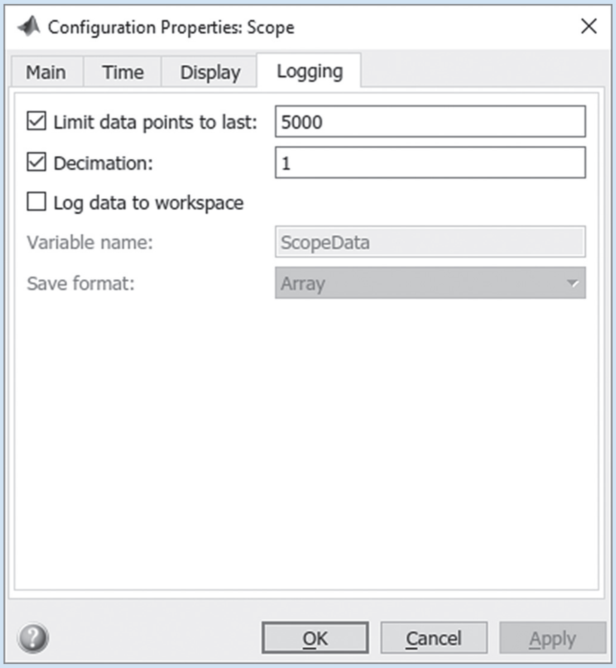
(*b*) (*c*)

FIGURE C.7 Windows for **Configuration Properties Scope** and **Style**. . . : **a. Scope**; **b.** Scope parameters, **Time** tab; **c.** Scope parameters, **Display** tab; d. Scope parameters, **Logging** tab; (*figure continues*)

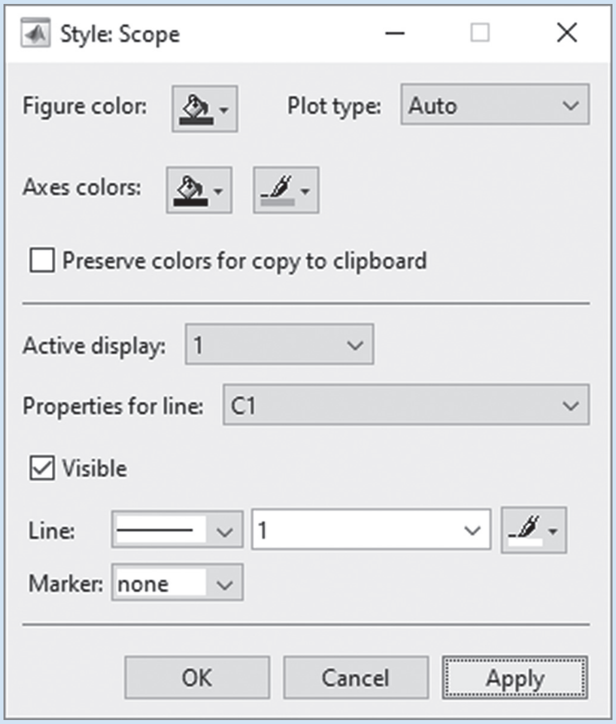
Clicking the **Configuration Properties Scope** tab on the **Scope** drop-down menu, shown in Figure C.7(*a*), accesses the **Configuration Properties Scope** window as shown in Figure C.7(*b*). The **Configuration Properties Scope** window contains four tabs, **Main, Time, Display**, and **Logging**, as shown in Figure C.7(*b*), C.7(*c*), and C.7(*d*).

Finally, clicking the **Style**. . . tab on the Scope drop-down menu reveals the **Style Scope** window, Figure C.7(*e*).

Choose parameters for the simulation Follow Step 6 to set simulation parameters. Figure C.8 shows the resulting **Configuration Parameters** window. Among other parameters, the simulation start and stop times can be set.



(*d*)



(*e*)

FIGURE C.7 (*continued*) **d.** Scope parameters, **Logging** tab; **e. Style Scope**

Start the simulation Now run the simulation by following Step 7. Figure C.9 shows the result in the **Scope** window. Color can be changed in the **Style Scope** window for each plot.

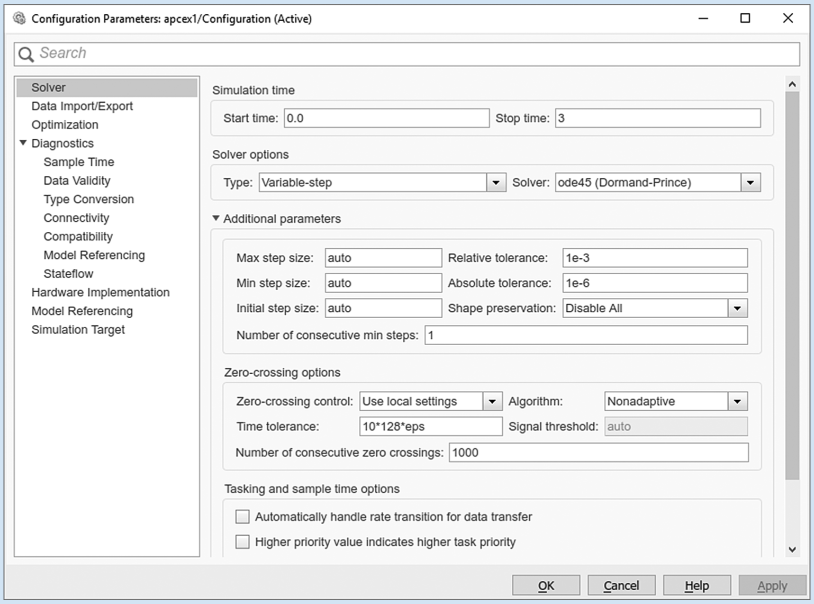


FIGURE C.8 Configuration Parameters window

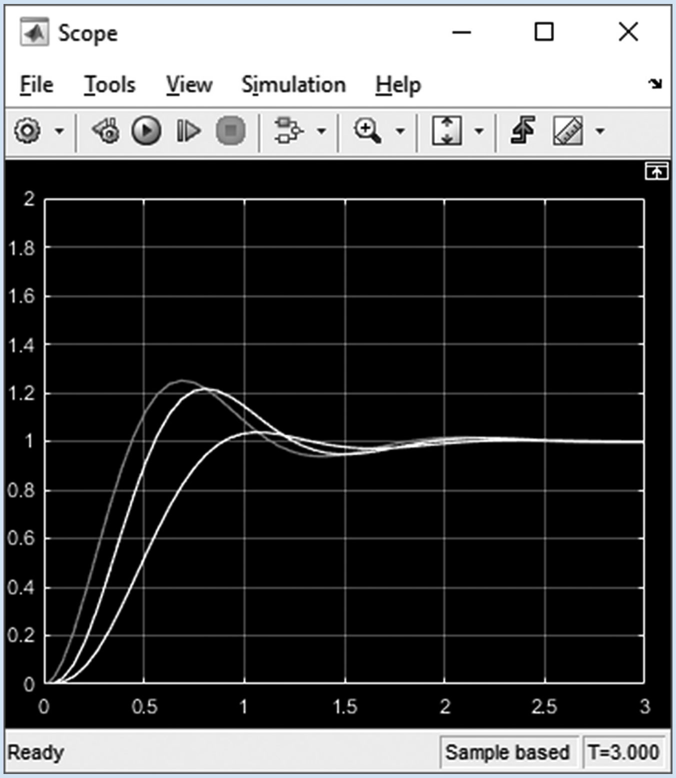


FIGURE C.9 Scope window after Example C.1 simulation stops

Interact with the plot The toolbar of the **Scope** window shown in Figure C.9 has several buttons and drop-down menus that can be used to interact with the plot. Explore the function and operation of each.

Example C.2

Effect of Amplifier Saturation on Motor's Load Angular Velocity

This example, which generated Figure 4.29 in the text, shows the use of Simulink to simulate the effect of saturation nonlinearity on an open-loop system. Figure C.10 shows a Simulink block diagram formed by following Steps 1 through 5 in Section C.2 above.

Saturation nonlinearity is an additional block that we have not used before. Saturation is obtained by dragging to your model window the **Saturation** block in the **Simulink Library Browser** window under **Discontinuities** as shown in Figure C.11(*a*) and setting its parameters to those shown in Figure C.11(*b*).

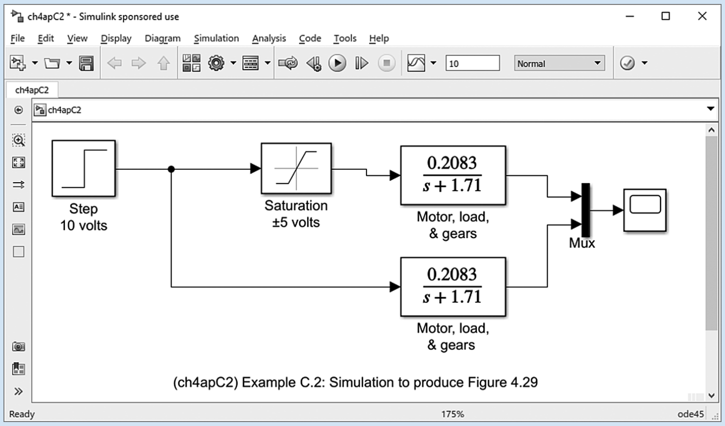
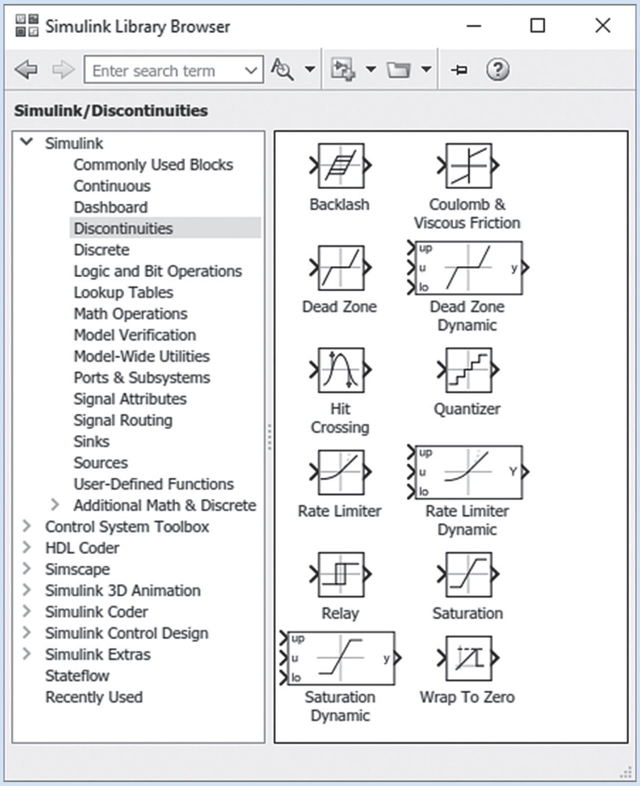
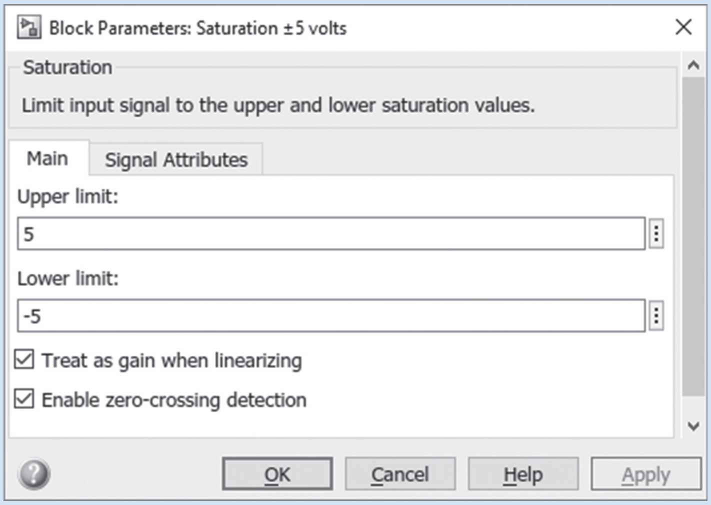


FIGURE C.10 Simulink block diagram for Example C.2



(*a*)

FIGURE C.11 **a.** Simulink library for nonlinearities; (*figure continues*)



(*b*)

FIGURE C.11 (*continued*) **b.** parameter settings for saturation

Now run the simulation by making your model window active and selecting **Run** under the **Simulation** menu of your model window or clicking on the **Run** button on your model window toolbar. Figure C.12 shows the result in the **Scope** window.

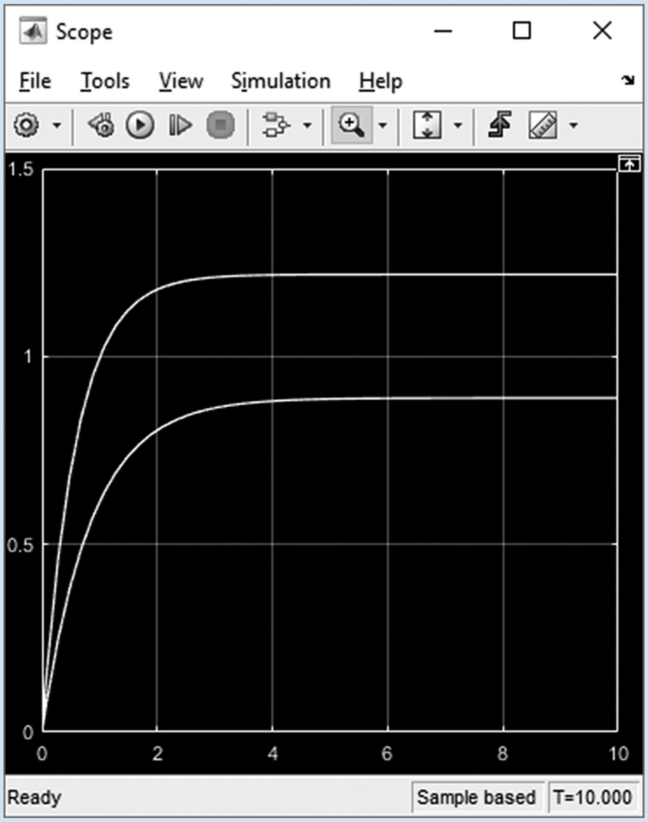


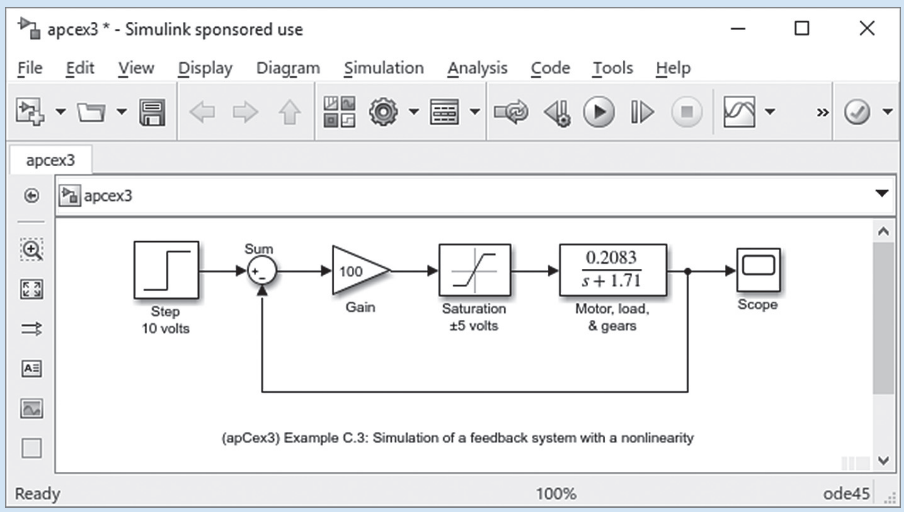
FIGURE C.12 **Scope** window after simulation of Example C.2 stops. The lower curve is the output with saturation

Example C.3

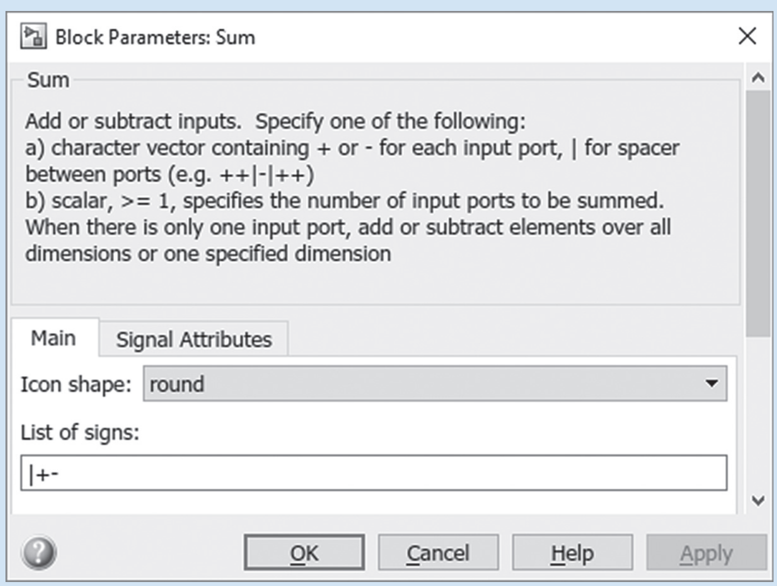
Simulating Feedback Systems

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.



(*a*)



(*b*)

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

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.

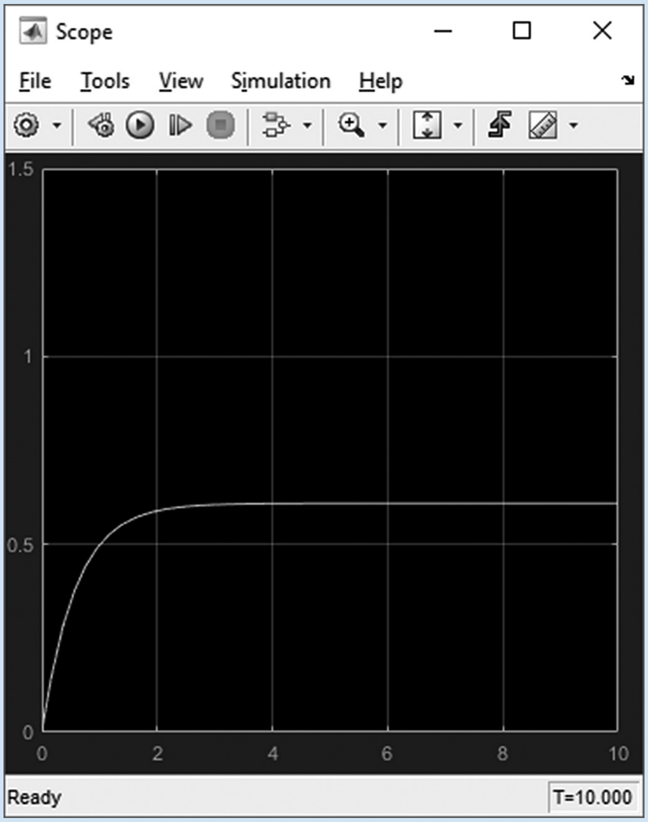


FIGURE C.14 Simulation output for Example C.3

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.

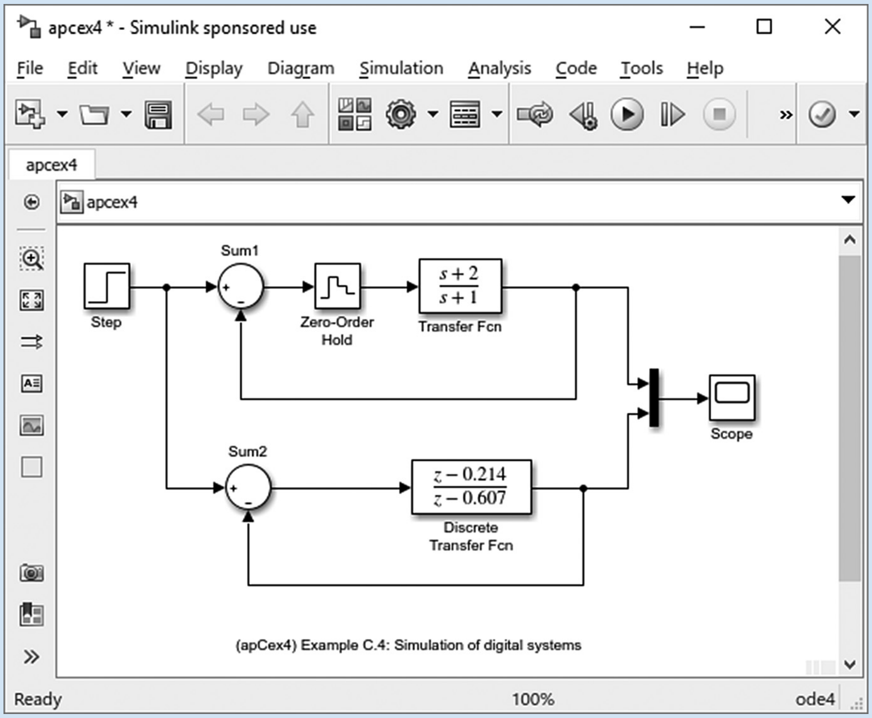


FIGURE C.15 Simulink block diagram for simulating digital systems two ways

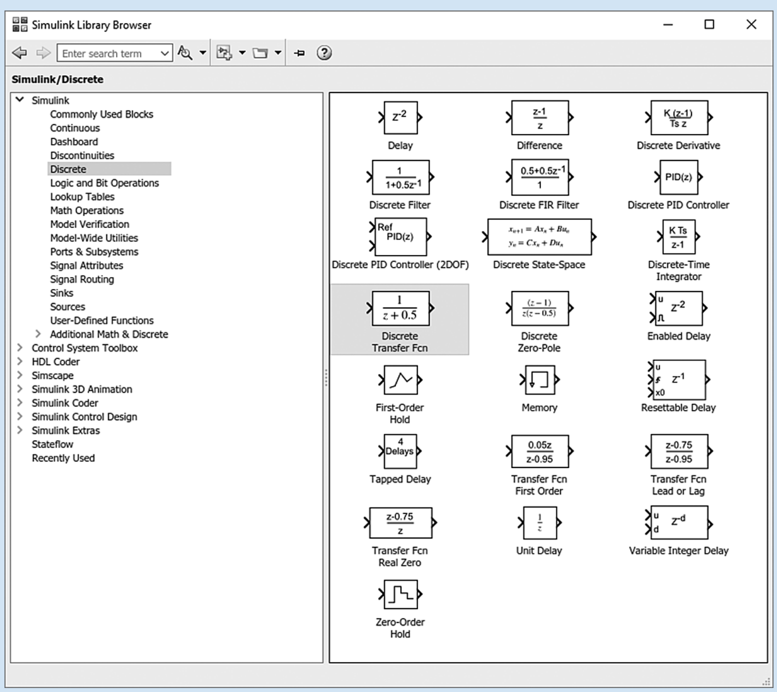
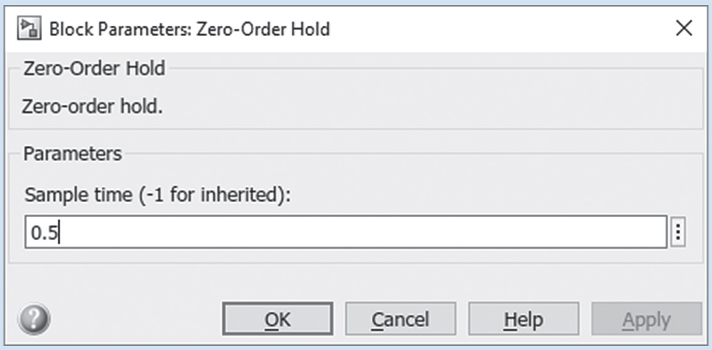


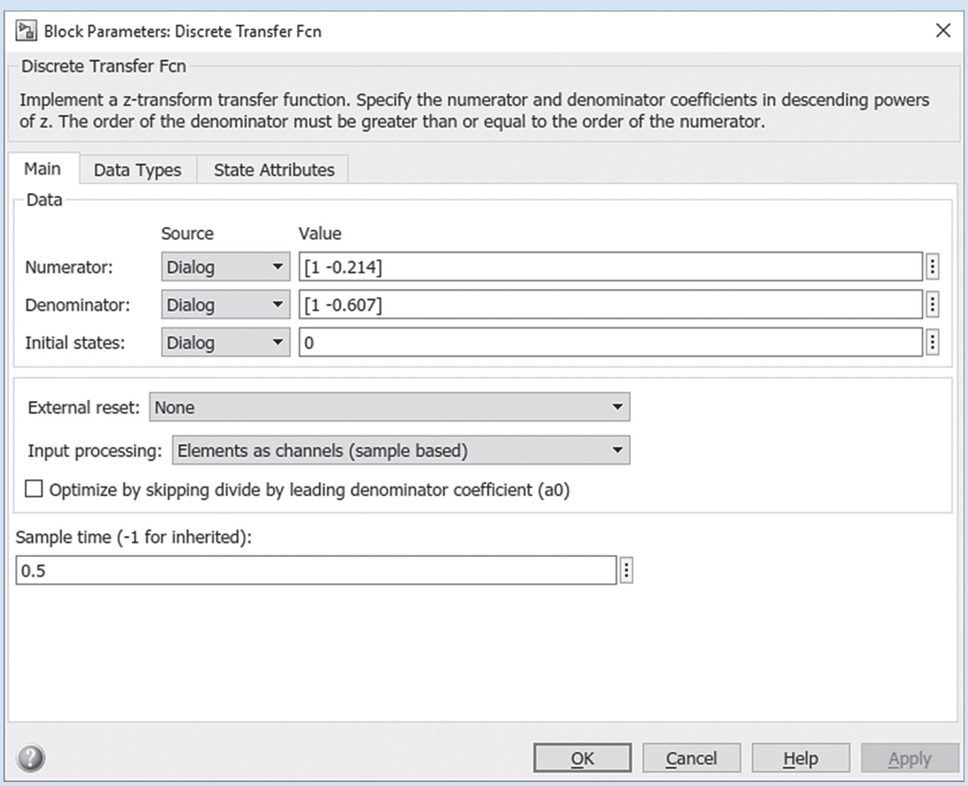
FIGURE C.16 Simulink library of discrete blocks

The block parameters for the **Zero-Order Hold** and **Discrete Transfer Fcn** blocks are set as shown in Figures C.17(*a*) and C.17(*b*), respectively. AQ1

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



(*a*)



(*b*)

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

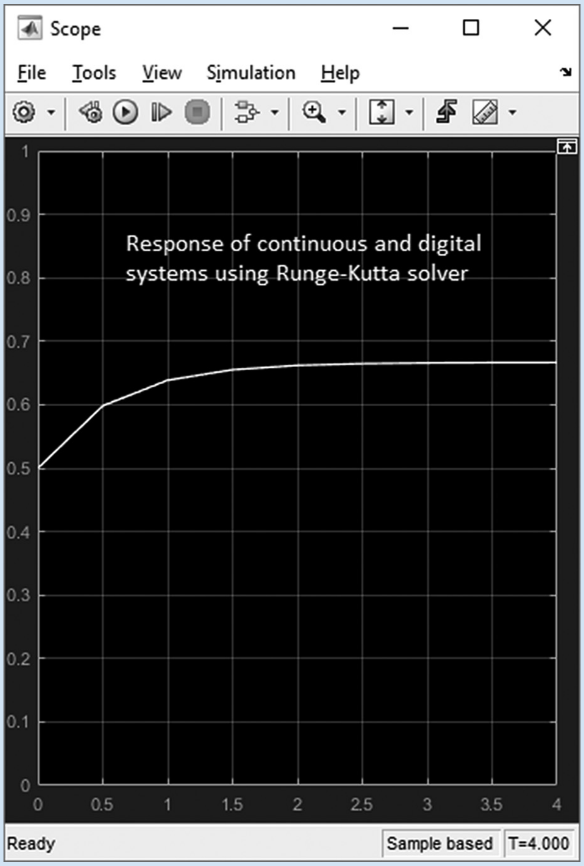


FIGURE C.18 Outputs of the digital systems

C.4 Using Simulink for Control System Design

In this section we show how to use Simulink to design control systems to meet specifications previously discussed in this book. We can make gain adjustments and design compensators using our Simulink system along with other windows that give us instant verification of our design. Specifically, we will concentrate on the design of PID compensators. We will show that PID compensators can be designed automatically or by adjusting design tools, such as response time and transient behavior. As we make adjustments, we see the immediate result of our design in lists of specifications or time responses along with the automatic calculation of the PID gains.

In order to perform control system design, you will need to add the Simulink® Control Design™ module Version 5.0 (R2017b), which contains all the necessary tools. Simulink designs PID controllers using derivative control with a low-pass filter to reduce noise. The design requires negative values of derivative gain, which you will notice in the design examples. We first cover the automated design of PID controllers. Next, we cover PID design using graphical methods.

Automated Design of PID Controllers

The automated design of PID gains generates reasonable robustness and response time. After the initial design, further adjustments are available, including response time, bandwidth, and phase margin. Let us first enumerate the steps involved, followed by an example.

1. **Create a Simulink diagram** Begin with a linear or nonlinear feedback control system containing a PID controller.

2. **Set initial values for the PID controller** Double-click the PID controller and launch the **Block Parameters: PID Controller** window. On the **MAIN** tab, input nominal values for the **Controller parameters.** Click **Apply**.
3. **Tune the PID controller** Click **Tune**… in the **Block Parameters: PID Controller** window. The system is linearized and the **PID Tuner** window is launched showing the nominal values response (**Block response**) and the designed response (**Tuned response**). Click the **Show parameters** button on the **Toolstrip** of the **PID Tuner** window to expose performance data, including the designed PID gains. If the response meets requirements, then click the **Update Block** button on the **Toolstrip** of the **PID Tuner** window to write the PID parameters to the controller.
4. **Modify the design via interactive tuning** If required, change the performance by moving the **Response Time** and **Transient Behavior** sliders in the middle of the **Toolstrip** of the **PID Tuner** window. Click the **Update Block** to write the PID parameters to the controller. If the system is nonlinear, you should run the simulation to see the effect of the nonlinearity on the designed response.

Example C.5

Automated Design of a PID Controller

In this example we follow the previously enumerated steps to automate the design of a PID controller for the system of Figure C.19. The requirements are (1) less than 1 second settling time; and (2) less than 5% overshoot.

Create a Simulink diagram We create the Simulink block diagram of Figure C.19 where the PID controller block is found in the **Simulink Library Browser** as shown in Figure C.3(*a*).

Set initial values for the PID controller Double-clicking the PID controller results in Figure C.20. Choose initial values as shown and click **Apply**.

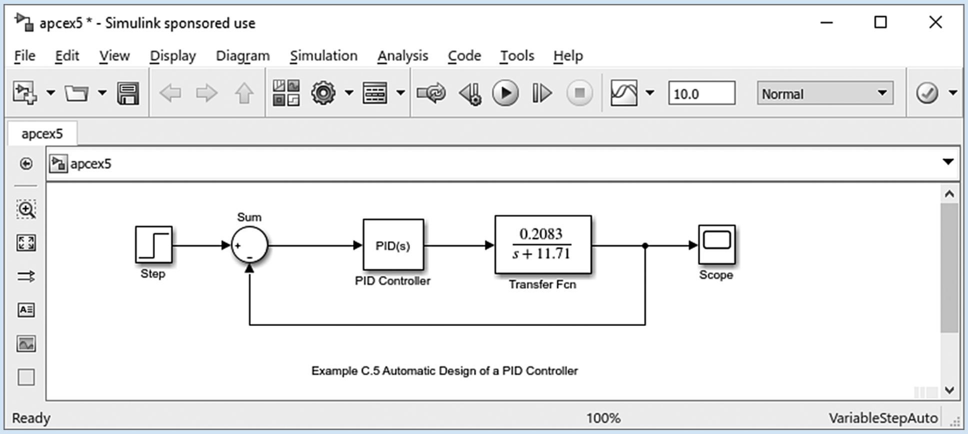


FIGURE C.19 Simulink block diagram for automated design of a PID controller

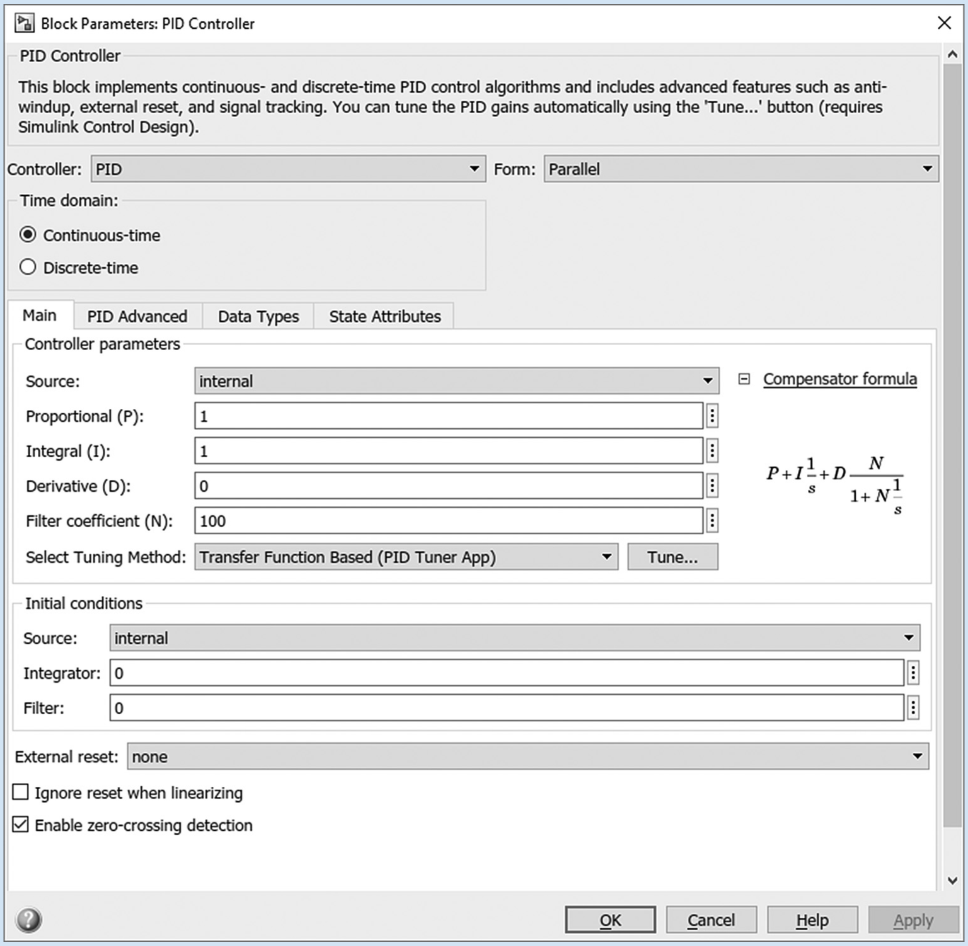
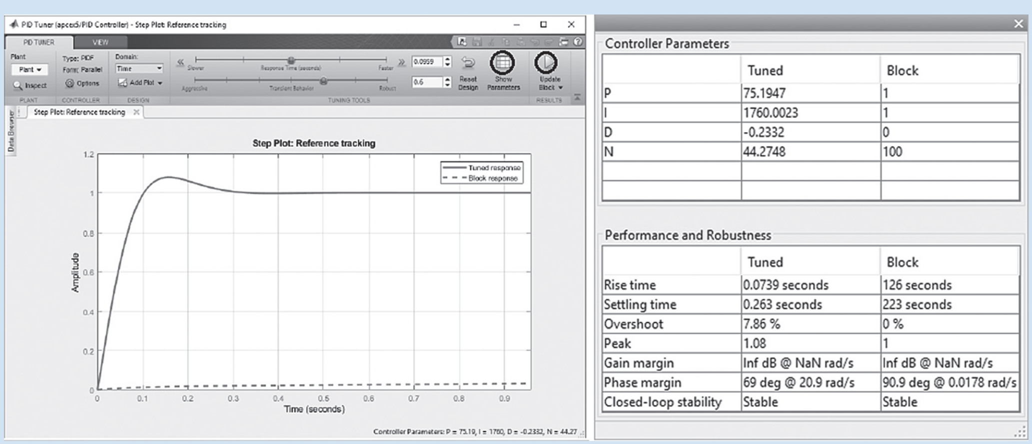


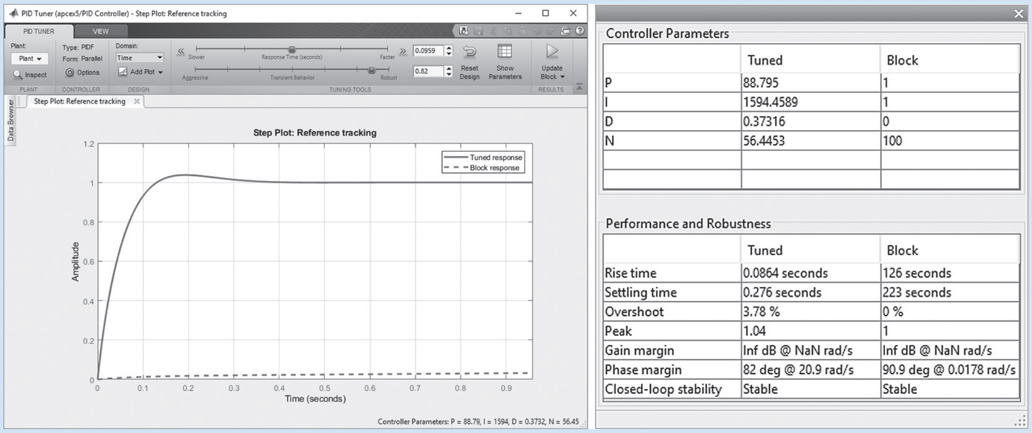
FIGURE C.20 **Block Parameters** window for the PID controller

Tune the PID controller Click **Tune**... in Figure C.20 to launch the **PID Tuner** window shown in Figure C.21(a). Click the **Show Parameters** button, shown encircled, on the **Toolstrip** of the **PID Tuner** window to display parameters and performance.



(a)

FIGURE C.21 **PID Tuner** window: **a.** before additional tuning; (*figure continues*)



(***b***)

FIGURE C.21 (*continued*) **b.** after additional tuning

Modify the design via interactive tuning Since the percent overshoot requirement is not met, we follow the instructions in **Step 4** on page C-19. We continue tuning the controller using the **Response Time** and **Transient Bahavior** sliders. Finally, click the encircled **Update Block** button to write the PID parameters to the controller. The final design is shown in Figure C.21(*b*).

Automated Tuning of PID Controllers and Graphical Design

In this subsection, we begin PID design with automated tuning followed by graphical design of our choice using Bode plots, root locus, etc. Let us first enumerate the steps involved followed by an example.

1. **Create a Simulink diagram** Begin with a linear or nonlinear feedback control system containing a PID controller.
2. **Begin compensator design by first performing automatic tuning** From the menu bar of your Simulink block diagram select **Analysis/Control Design/Control System Tuner**… which launches the **Control System Tuner** window. Select the **Tuning** tab.
3. **Select block to tune** In the **Control System Tuner** click **Select Blocks**. The **Select Tuned Blocks** window appears. Click on **Add Blocks** and select the **PID controller**. Click **OK**. The result appears in the **Data Browser** in the **Control System Tuner** window.
4. **Choose the closed-loop response that will be analyzed** Click on **New Goal** and select **Transient response matching**. A **Transient Goal** window results. Specify response inputs, response outputs, **Input Signal Selection**, **Desired Transient Response**, and **Options**. Click **Apply** and **OK**. Your choice appears in the **Data Browser** and in a plot of the actual and desired response. Right-click on the graph for more information about the plots. Click **Tune** to see the effect of the tuning on the actual response. Select the **Control System** tab in the **Control System Tuner** and click on **Update Blocks** to

transfer the automated tuning design to the actual system. Click **Apply** in the **Block Parameters: PID Controller** to complete the transfer.

5. **Continue with graphical design setup if response requirements were not met with automatic tuning** From the menu bar of your Simulink block diagram select **Analysis/Control Design/Control System Designer**… In the resulting **Edit Architecture — Simulink Configuration** window click **Add Blocks**… under the **Blocks** tab and select the block to tune. Click **OK**. Next, select the **Signals** tab and select the output signal of the block to tune. Select **Tuning Methods** from the **Control System** tab followed by a choice of the **Root Locus Editor**. Click **Plot** and the root locus is presented.
6. **Perform root locus design** Right-click the root locus and select **Design Requirements/New**. From the resulting window, make a selection from the drop-down menu under **Design requirements type**. Repeat for additional requirements. The design boundaries then appear on the root locus. You now can move closed-loop points along the root locus to change gain or move controller poles and zeros to effect closed-loop poles that meet the design requirements.
7. **Test the design** Under the **Control System** tab choose **New Plot** and select **New Step**. In the drop-down menu for **Select Response to Plot** choose **New Input-Output Transfer Response**. In the **New Step to Plot** resulting window specify the closed-loop input and output signals. Click **Plot**. Right-click the plot and choose **Characteristics**. The characteristics chosen will show up on the plot from which a determination may be made that the design requirements have been achieved. If further design is not required, click **Update Blocks** under the **Control System** tab to load your design into the Simulink model. Click **Apply** in the **Block Parameters: PID Controller** to complete the transfer.
8. **Run a simulation on the Simulink model**

Example C.6

Automated Tuning of a PID Controller and Graphical Design

In this example we follow the previously enumerated steps to automatically design a PID controller for the system of Figure C.19 and follow with further improvement in performance using root locus. The requirements are: (1) less than 1 second settling time; and (2) less than 4% overshoot.

Create a Simulink diagram We create the Simulink block diagram of Figure C.19 where the PID controller block is found in the **Simulink Library Browser** as shown in Figure C.3(*b*). We first perform automatic tuning.

Begin Compensator design In Figure C.19, we select **Analysis/Control Design/Control System Tuner**…, launching the **Control System Tuner** window shown in Figure C.22.

Click on **New Goal** in Figure C.22 and select **Transient response matching**. In the resulting **Transient Goal** window shown in Figure C.23, specify response inputs and outputs, input signal selection, and **Desired Transient Response**. In Figure C.23 the **Desired Transient Response** is indicated by a second-order system that has the desired settling time and percent overshoot. Finally, input any other **Options**. Click **Apply** and

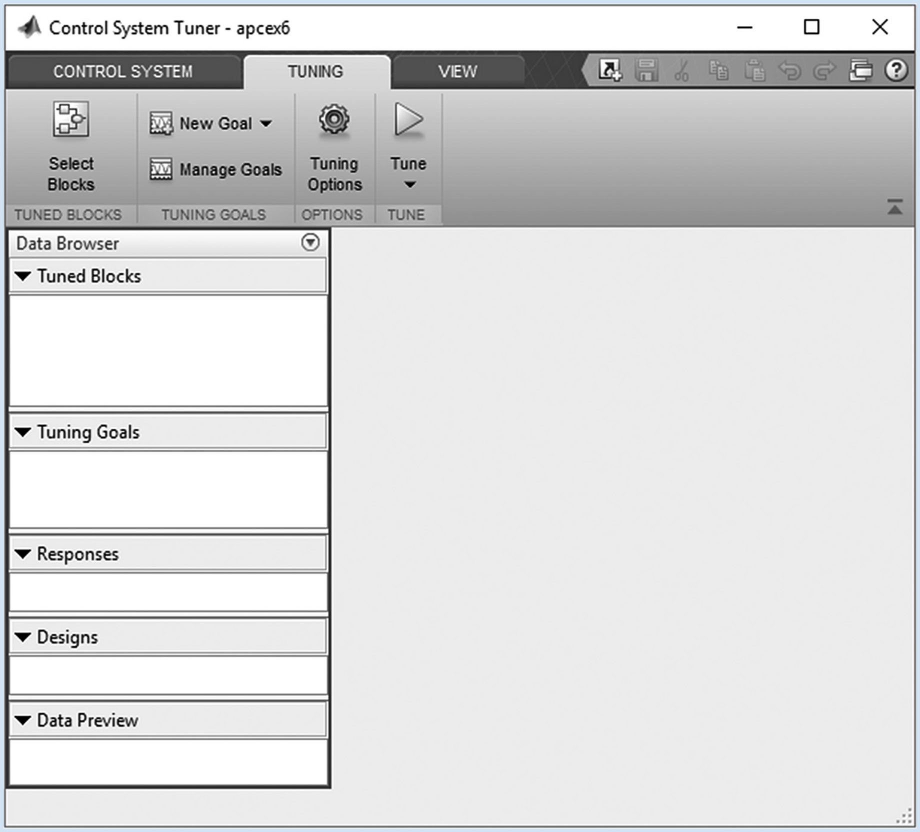


FIGURE C.22 **Control System** Tuner window before selecting block to tune

OK. Actual and Desired plots are then displayed in the **Control System Tuner** window shown in Figure C.24. Right-click on the graph and select **Characteristics** to help you evaluate performance. If you are unsatisfied with desired performance and are ready to make further improvements through graphical design, then click **Tune** under the **Tuning** tab followed by clicking the **Update Blocks** button under the **Control System** tab to port your automatically designed values to your Simulink diagram. Click **Apply** in the **Block Parameters: PID Controller** window if necessary.

Continue improvement with graphical design using root locus From the menu bar of your Simulink block diagram select **Analysis/Control Design/Control System Designer**... In the resulting **Edit Architecture - Simulink Configuration** window click **Add Blocks**... under the **Blocks** tab and select the PID controller as the block to tune as shown in Figure C.25(a). Next, select the **Signals** tab and select the output of the PID controller. Click **OK** in Figure C.25(b). Now select **Tuning Methods** under the **Control System** tab of the **Control System Designer** window. Choose the **Root Locus Editor** from the drop-down menu, Select the response to edit, and click **Plot** as shown in Figure C.26. Right-click the resulting root locus plot and select **Design Requirements/New**... Choose settling time and percent overshoot. Figure C.27(a) shows the root locus with the settling time and percent overshoot boundaries. We can improve the settling time by moving the PID pole at -8 and keeping the closed-loop poles within the boundaries shown. The improved root locus is shown in Figure C.27(b)

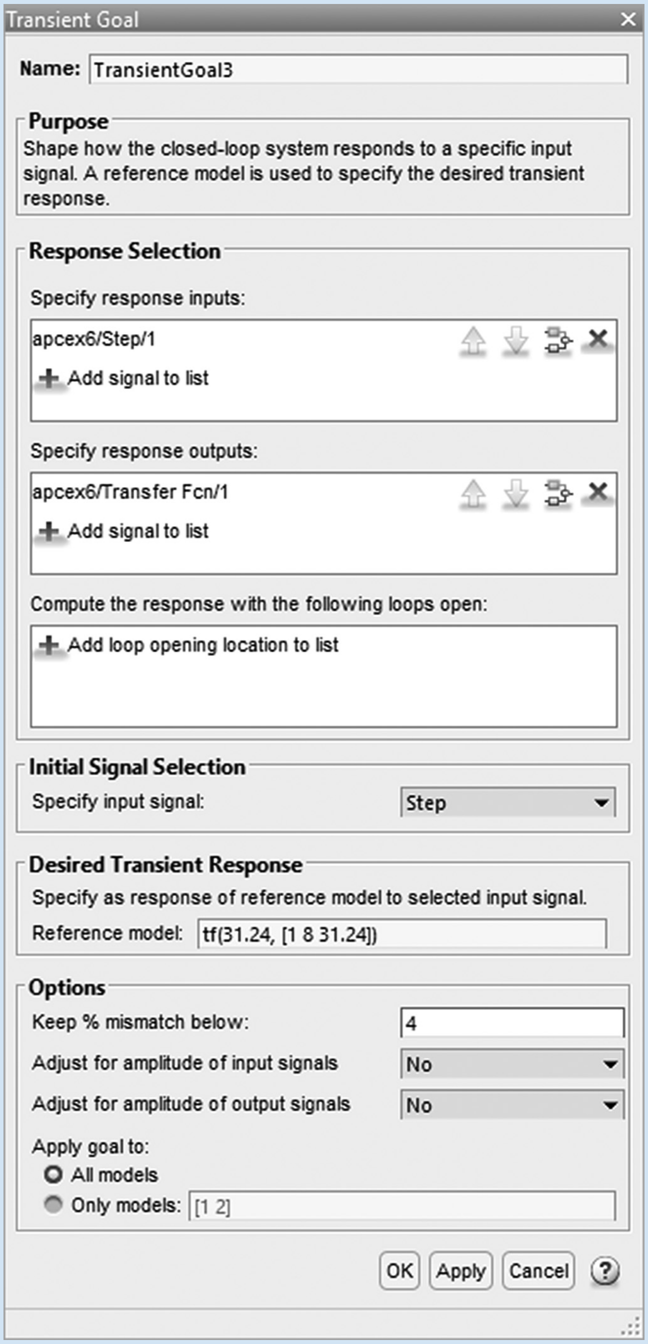


FIGURE C.23 **Transient Goal** window

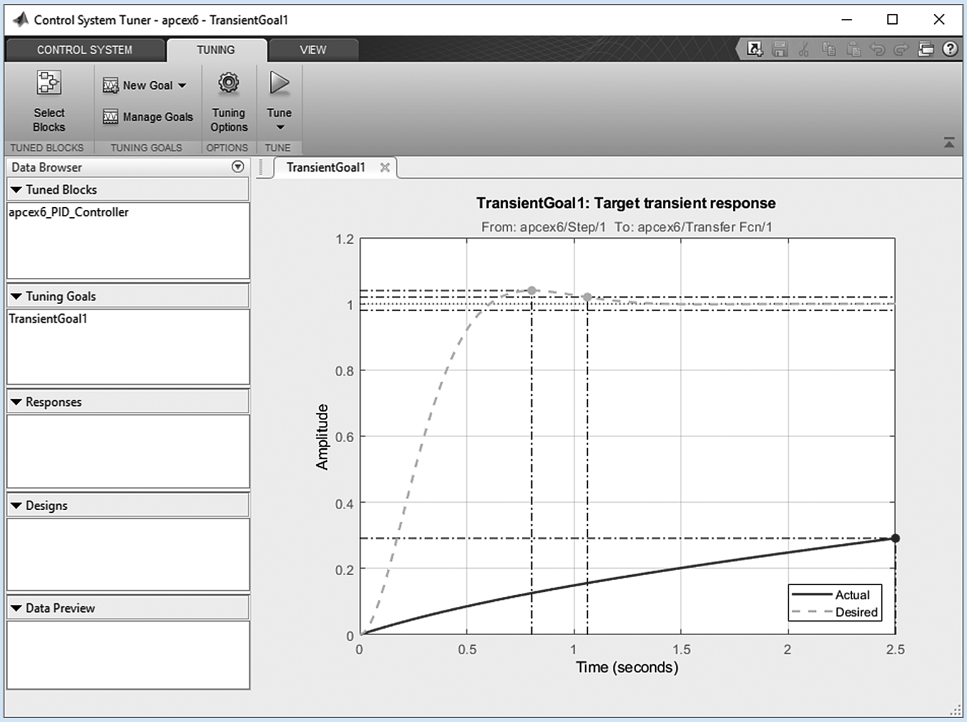


FIGURE C.24 **Control System Tuner** window with **Target transient response plots** with **Control System** tab selected

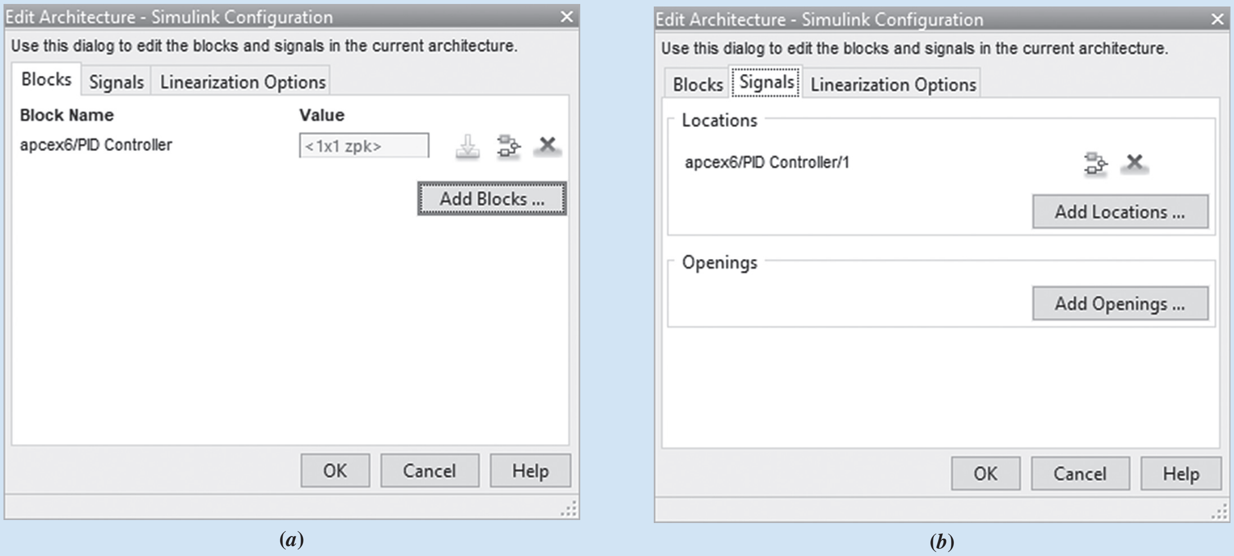


FIGURE C.25 **Edit Architecture — Simulink Configuration**: **a. Blocks** tab; **b. Signals** tab

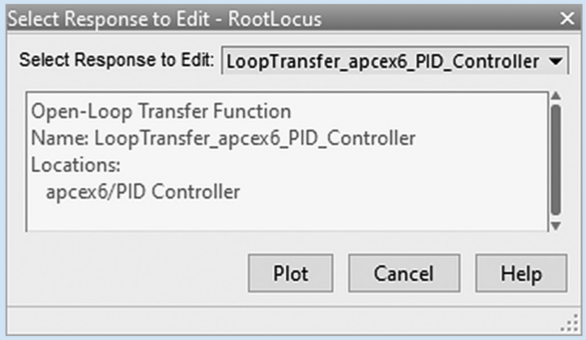
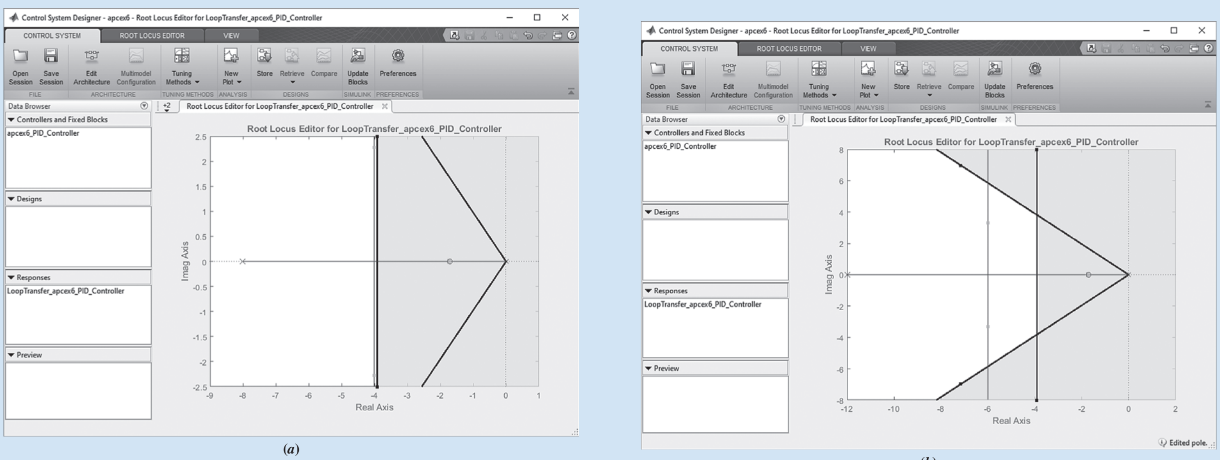


FIGURE C.26 **Select Response to Edit - Root Locus** window



(*a*) (*b*)

FIGURE C.27 Root locus plots with design boundaries: **a**. Original ; **b**. After moving compensator pole

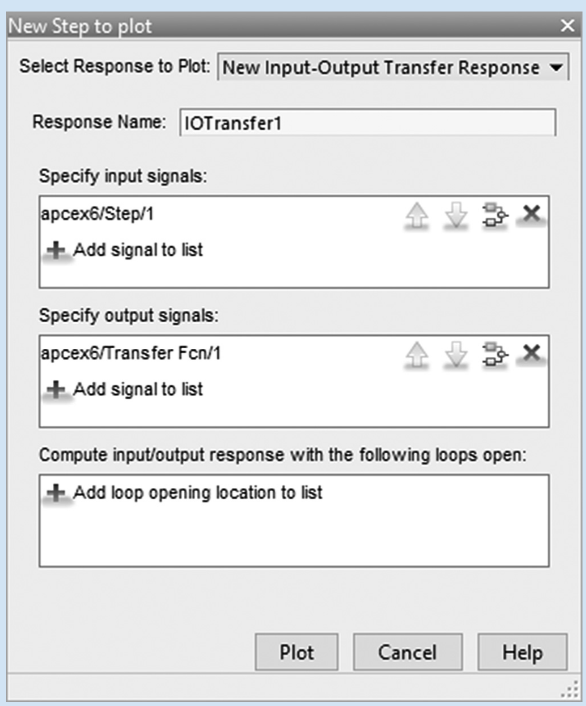


FIGURE C.28 **New Step to Plot** window

Test the design Under the Control System tab choose **New Plot** and select **New Step**. The **New Step to plot** window results and is shown filled out in Figure C.28. Click Plot. Right click the plot and select **Characteristics** to put settling time and percent overshoot on the plot. Let your mouse pause over the indicated point to get a label showing the numerical results. Figure C.29 shows the final successful design.

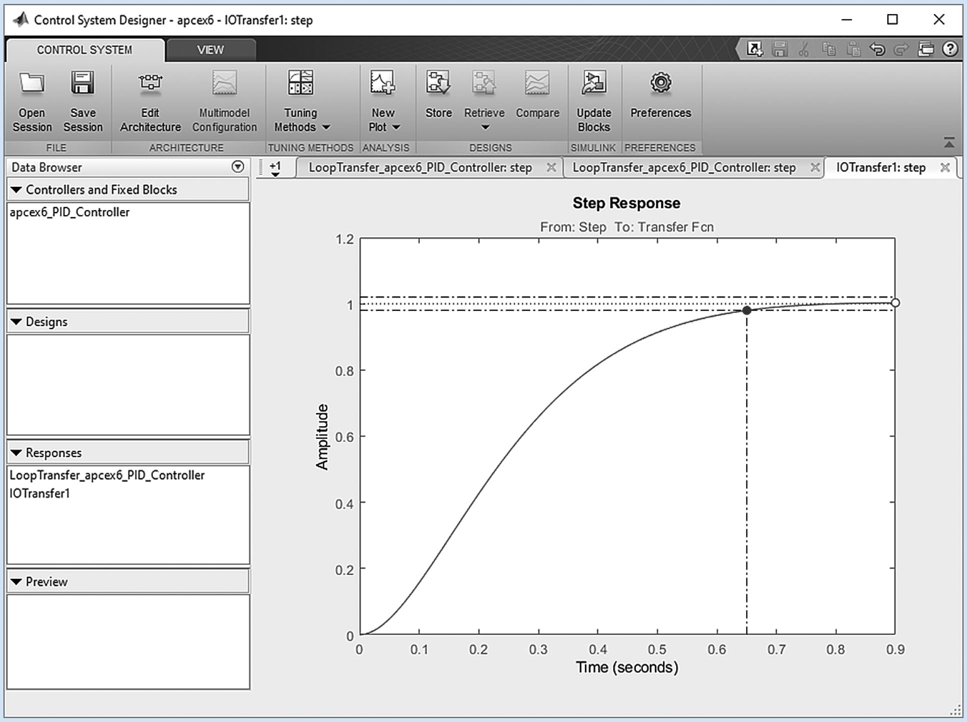


FIGURE C.29 Final designed response showing settling time

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.

In addition, we showed how to use the Simulink Control Design add-on to automatically tune PID controllers and perform shaping of graphical design tools in order to meet performance requirements.

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 Simulink by 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.

Bibliography

MathWorks. *Control System Toolbox* ™ *Getting Started Guide R2017b*. MathWorks, Natick, MA, 2000–2017.

MathWorks. *Control System Toolbox* ™ *User's Guide R2017b*. MathWorks, Natick, MA, 2001–2017.

MathWorks. *MATLAB® Primer R2017b*. MathWorks, Natick, MA, 1984–2017.

MathWorks. *MATLAB® Graphics R2017b*. MathWorks, Natick, MA, 1984–2017.

MathWorks. *MATLAB® Mathematics R2017b*. MathWorks, Natick, MA, 1984–2017.

MathWorks. *MATLAB® Programming Fundamentals R2017b* . MathWorks, Natick, MA, 1984–2017.

MathWorks. *Simulink® Getting Started Guide R2017b*. MathWorks, Natick, MA, 1990–2017.

MathWorks. *Simulink® User's Guide R2017b*. MathWorks, Natick, MA, 1990–2017.

MathWorks. *Simulink® Control Design™ Getting Started Guide R2017b*. MathWorks, Natick, MA, 2004–2017.

MathWorks. *Simulink® Control Design™ User's Guide R2017b*. MathWorks, Natick, MA, 2004–2017.