The McGraw-Hill Companies

PowerPoint to accompany

Introduction to MATLAB for Engineers

Third Edition

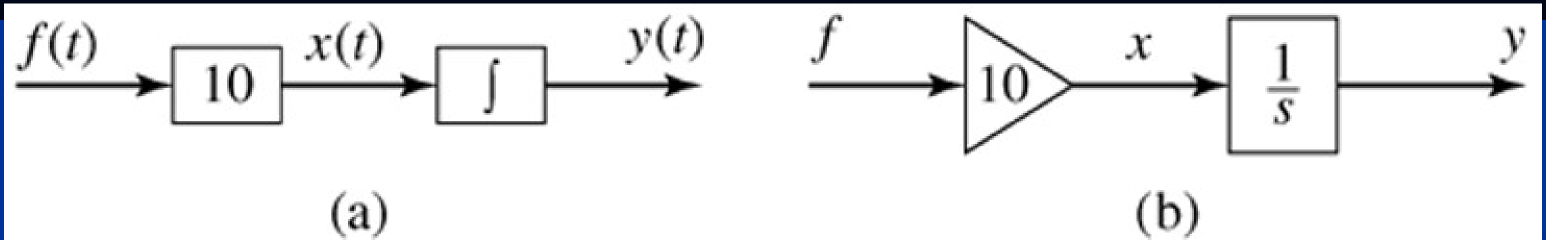
William J. Palm III

Chapter 10
Simulink

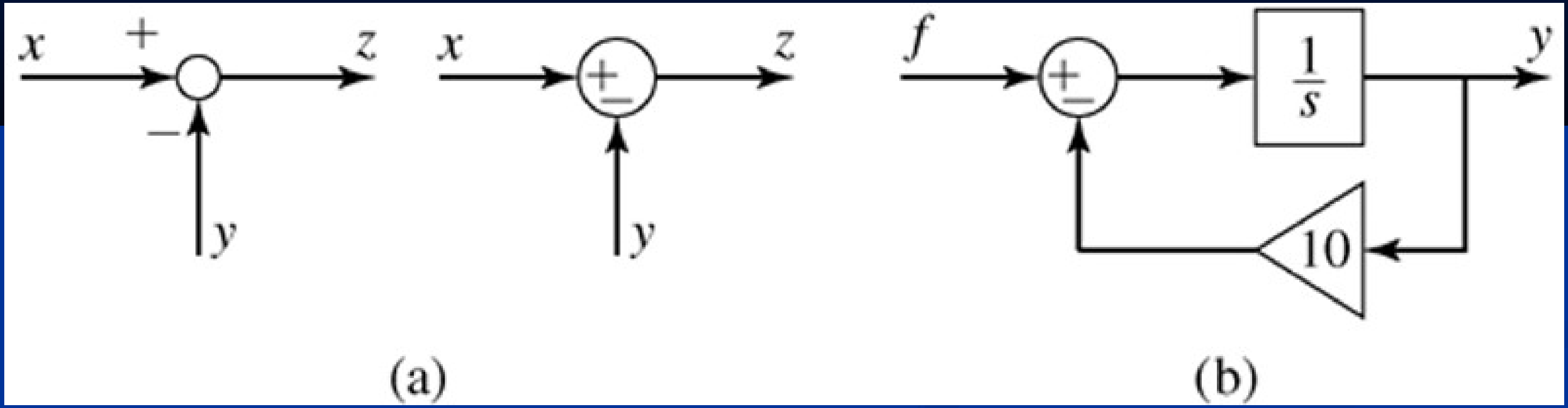
Mc Graw Hill

.

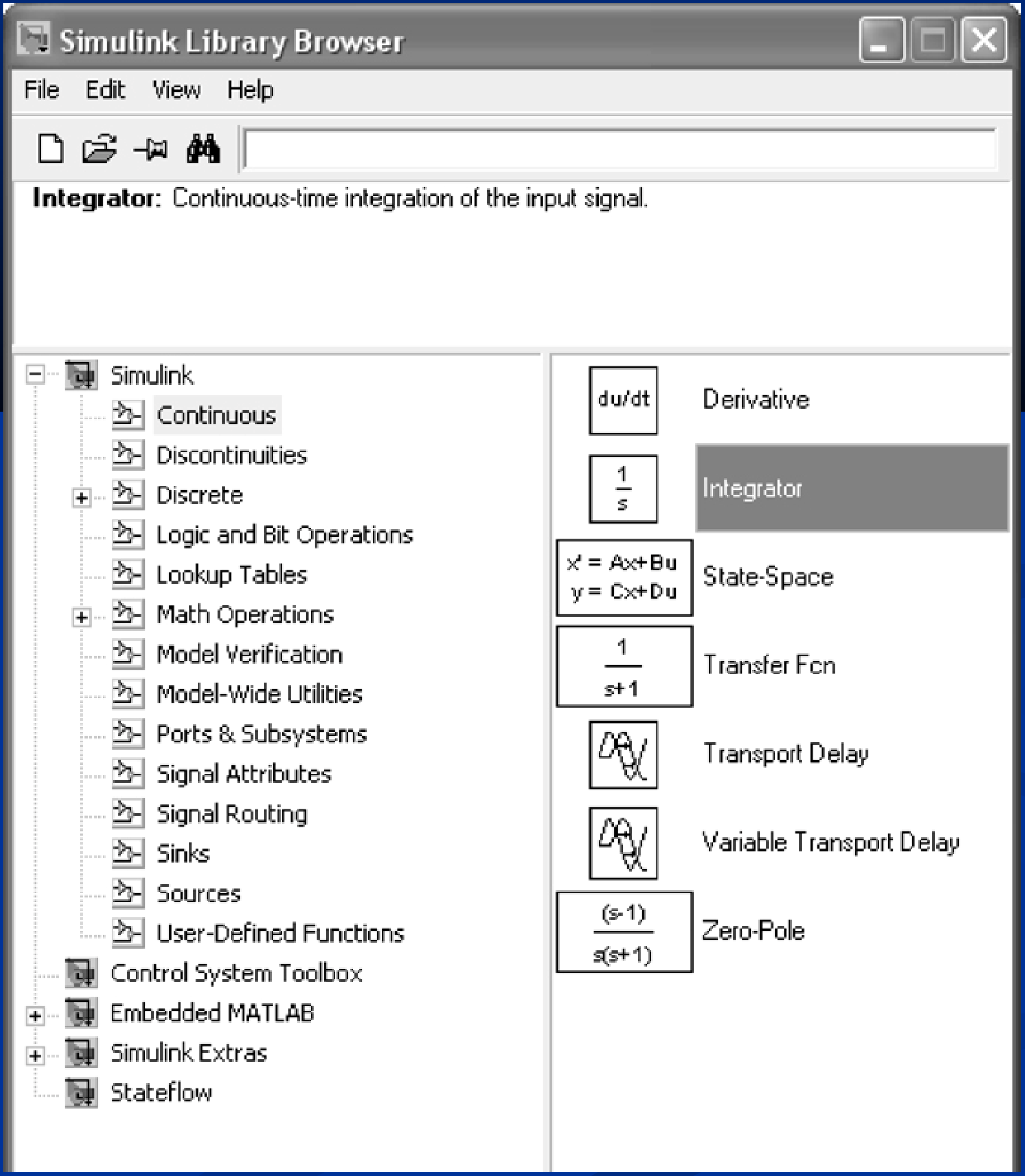
Simulation diagrams for $y = 10 f(t)$. Figure 10.1–1 on page 420



(a) The summer element. (b) Simulation diagram for
y **=** ***f*** **(*t*) – 10*y*.** Figure 10.1–2 on page 421



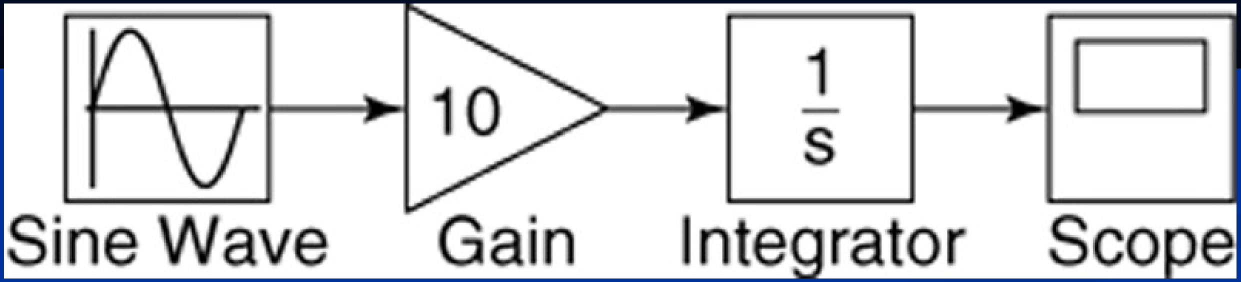
The Simulink Library Browser. Figure 10.2–1 on page 422



Integrator: Continuous-time integration of the input signal.

.

Simulink model for *y* = 10 sin *t*. Figure 10.2–2 on page 423



For the steps needed to construct this model, see Example 10.2-1 on pages 422-424.

Note that blocks have a Block Parameters window that opens when you double-click on the block.

This window contains several items, the number and nature of which depend on the specific type of block.

In general, you can use the default values of these parameters, except where we have explicitly indicated that they should be changed.

You can always click on **Help** within the Block Parameters window to obtain more information.

See page 424 for more information.

Note that most blocks have default labels.

You can edit text associated with a block by clicking on the text and making the changes.

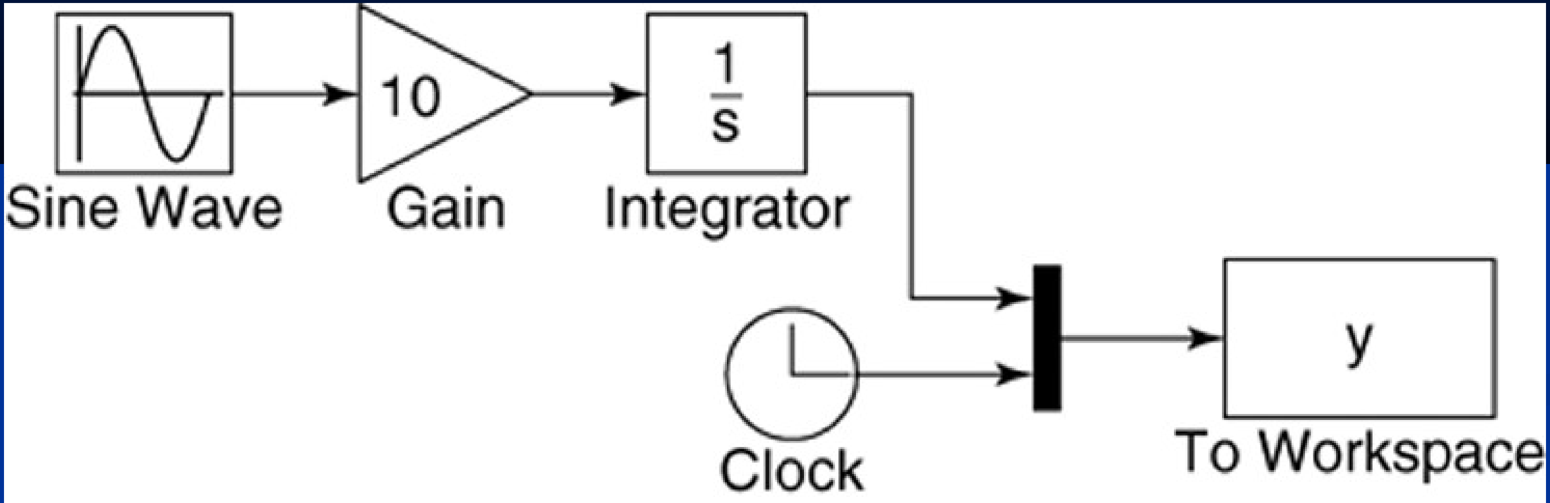
You can save the Simulink model as an `.mdl` file by selecting **Save** from the **File** menu in Simulink.

The model file can then be reloaded at a later time.

You can also print the diagram by selecting **Print** on the **File** menu.

Simulink model using the Clock and To Workspace blocks.

Figure 10.2–5 on page 425



For the steps needed to construct this model, see Example 10.2-2 on pages 425-426.

Double-click on the To Workspace block. You can specify any variable name you want as the output; the default is `simout`. Change its name to `y`.

The output variable `y` will have as many rows as there are simulation time steps, and as many columns as there are inputs to the block.

The second column in our simulation will be time, because of the way we have connected the Clock to the second input port of the Mux.

Specify the Save format as Array. Use the default values for the other parameters (these should be `inf`, `1`, and `-1` for Maximum number of rows, Decimation, and Sample time, respectively). Click on **OK.**

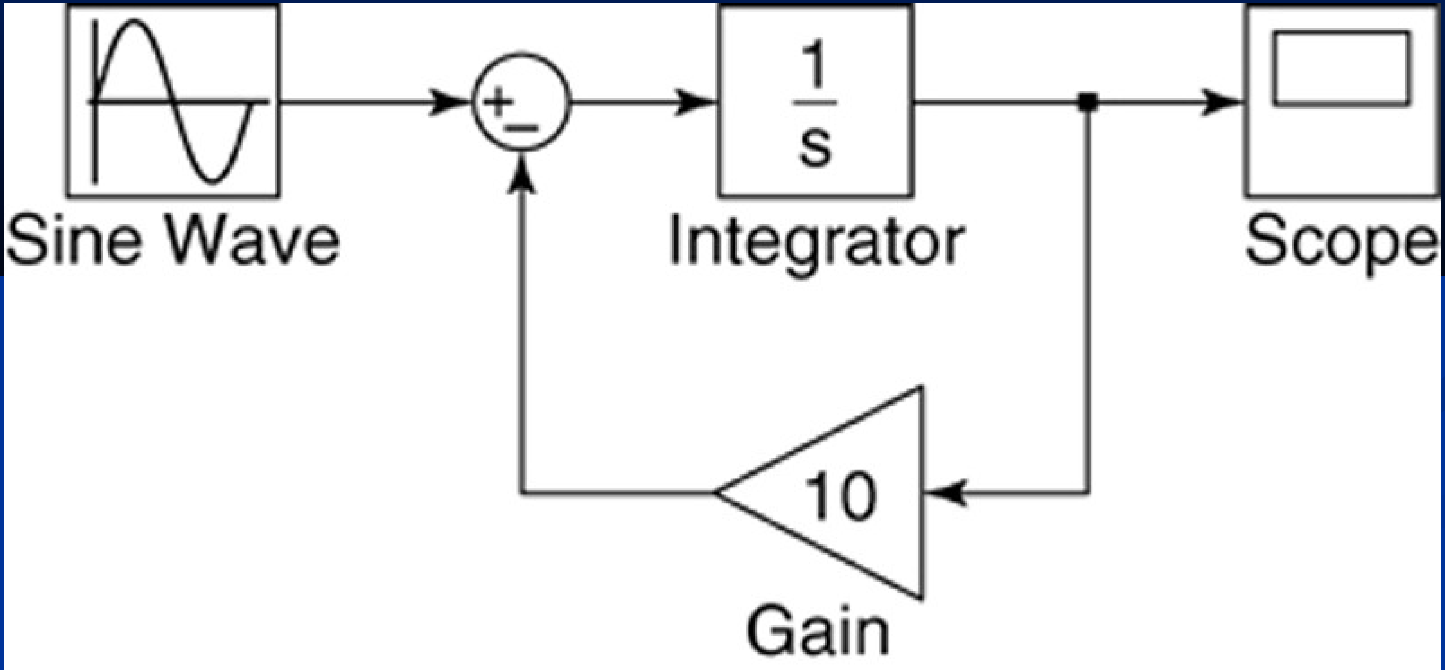
Simulink can be configured to put the time variable `tout` into the MATLAB workspace automatically when you are using the To Workspace block.

This is done with the Data I/O tab under **Configuration Parameters** on the Simulation menu.

The alternative is to use the Clock block to put `tout` into the workspace.

The Clock block has one parameter, Decimation. If this parameter is set to 1, the Clock block will output the time every time step; if set to 10 for example, the block will output every 10 time steps, and so on.

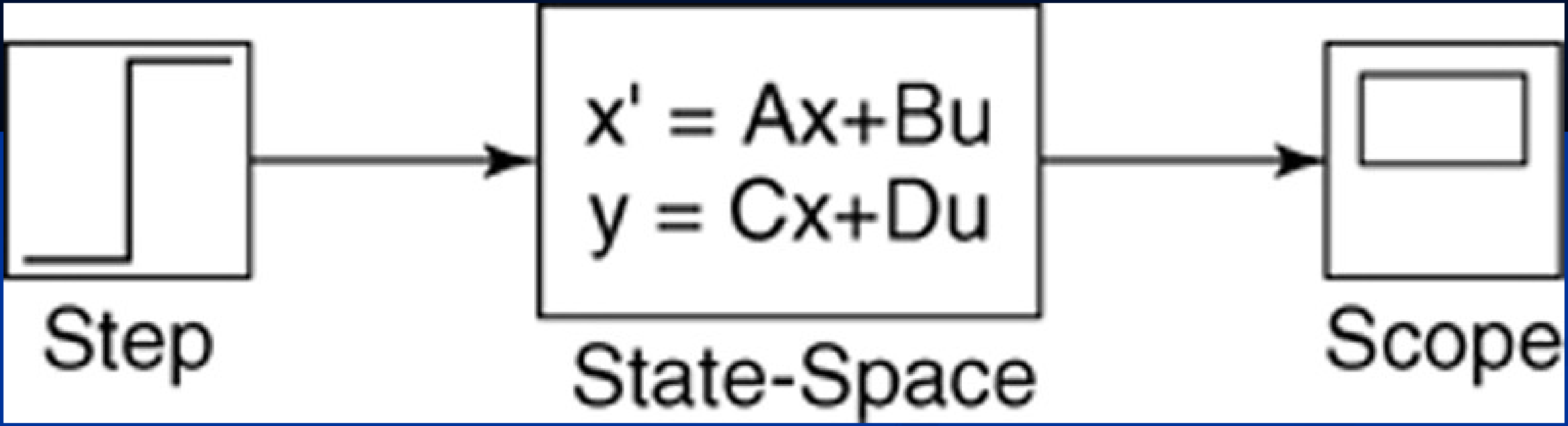
Simulink model for *y* = –10*y* + *f* (*t*). Figure 10.2–6 on page 426



For the steps needed to construct this model, see Example 10.2-3 on pages 426-427.

A vibrating system having two masses. Figure 10.3–1 on page 428. The state-variable model and simulation construction steps are given in Example 10.3-1 on pages 427-430.

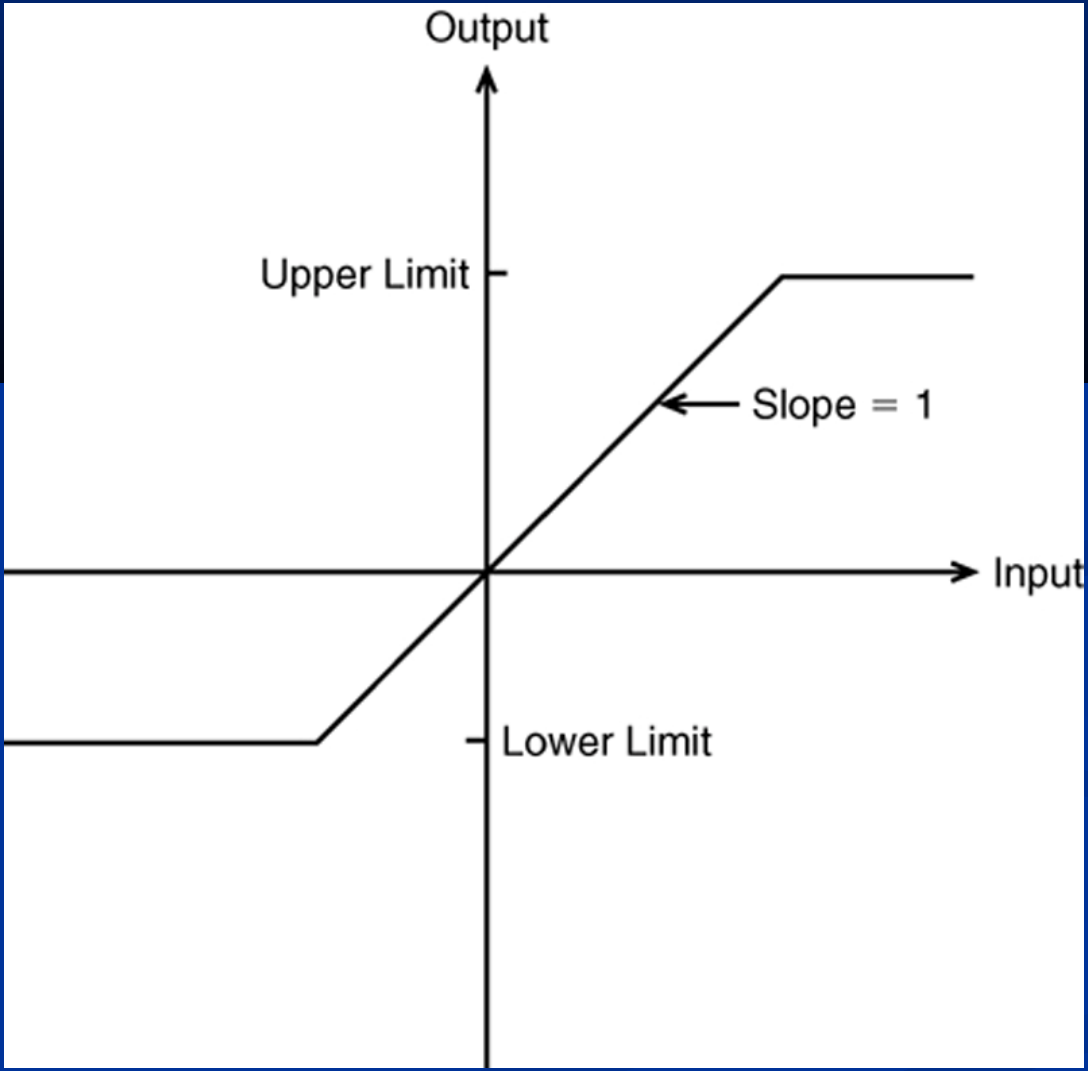
Simulink model of the system having two masses. Use of the State-Space block and the Step block. Figure 10.3–2 on page 429.



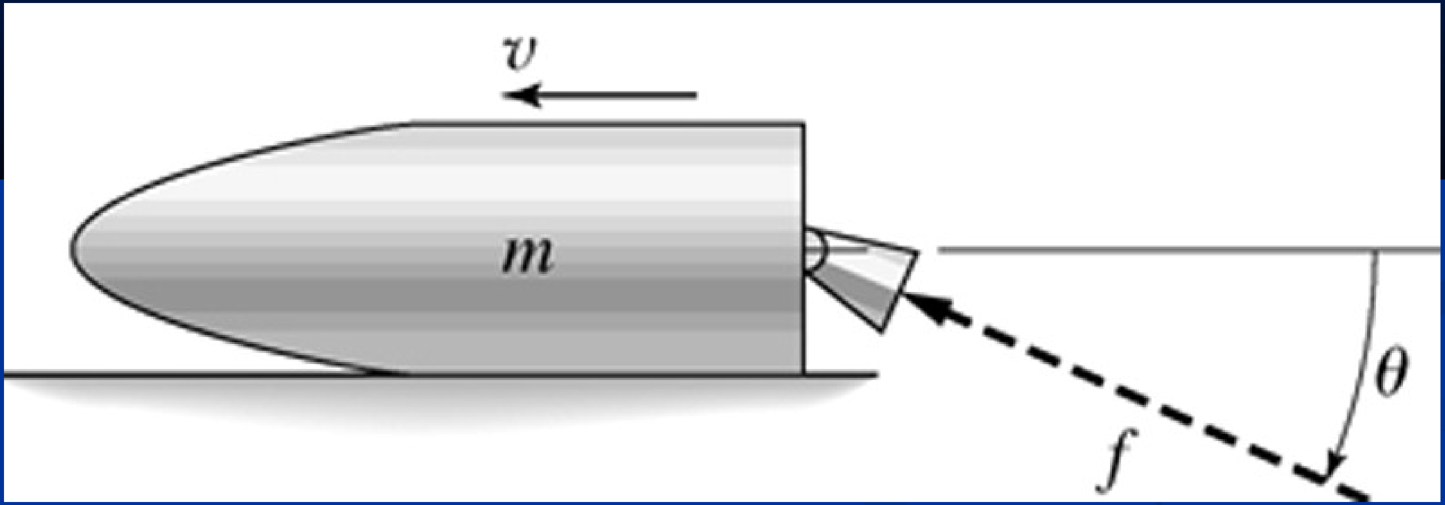
When you are connecting inputs to the State-Space block, care must be taken to connect them in the proper order.

Similar care must be taken when connecting the block’s outputs to another block.

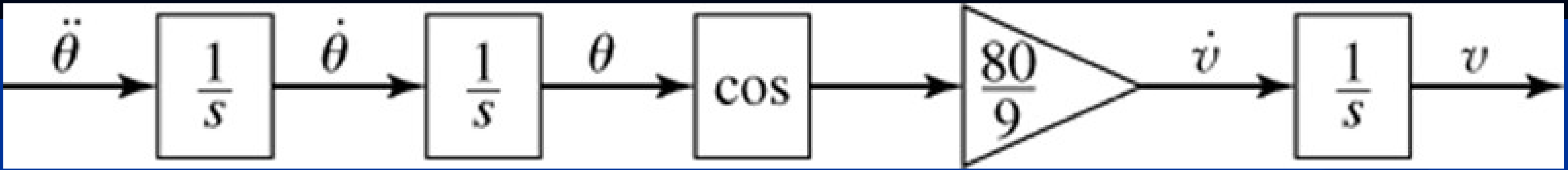
The saturation nonlinearity. Figure 10.4–1 on page 431.



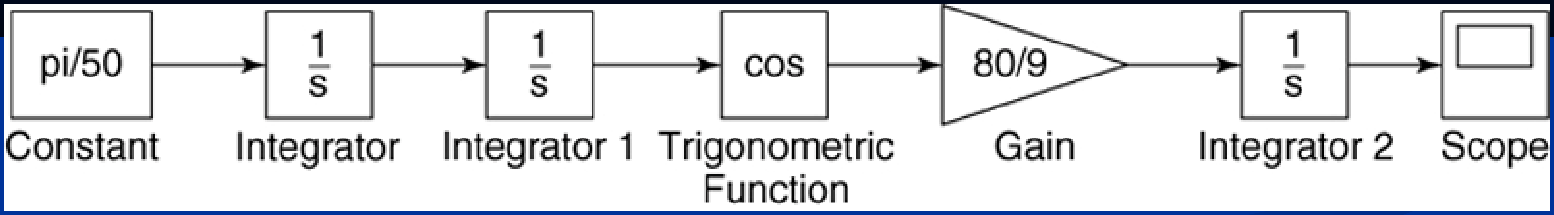
Example 10.4-1: A rocket-propelled sled. Figure 10.4–2 on page 431.



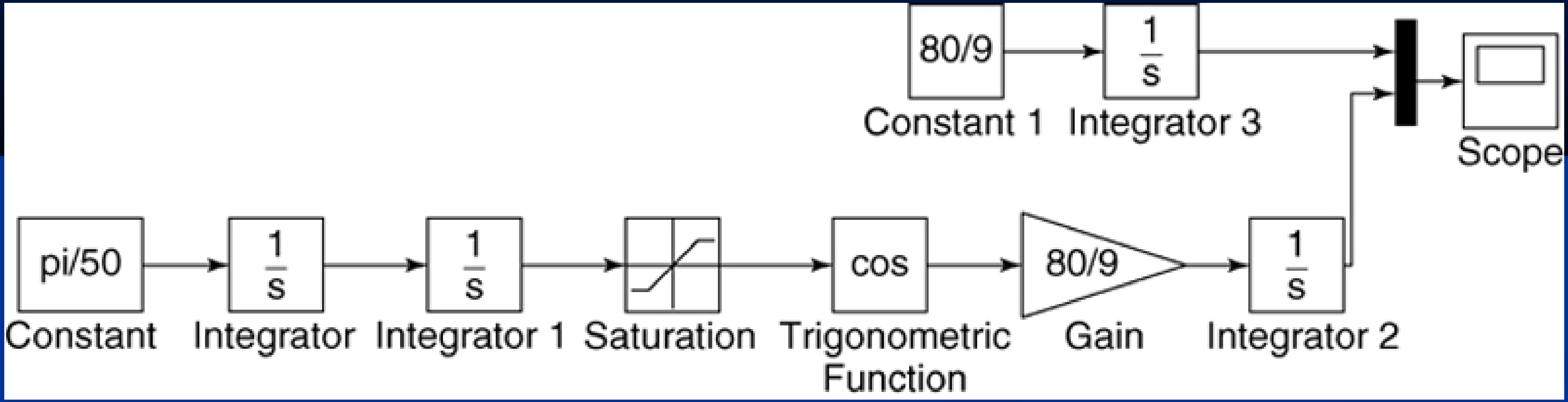
Rocket-propelled sled example. Simulation diagram for $\upsilon = (80/9)\cos(\pi t^2 / 100)$. Example 10.4-1 and Figure 10.4–3 on page 432.



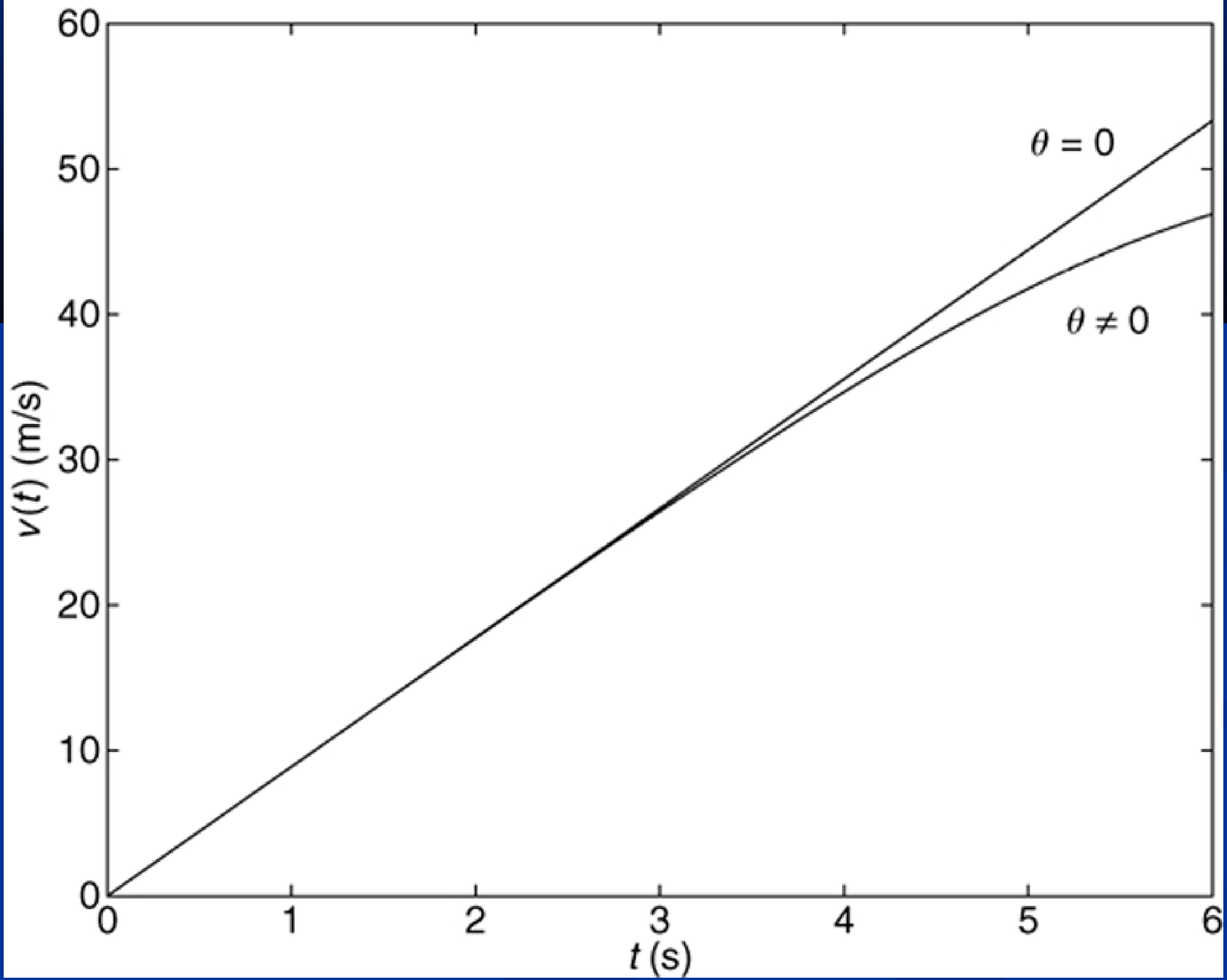
Simulink model for $v = (80/9)\cos(\pi t^2/100)$. Figure 10.4–4 on page 433.



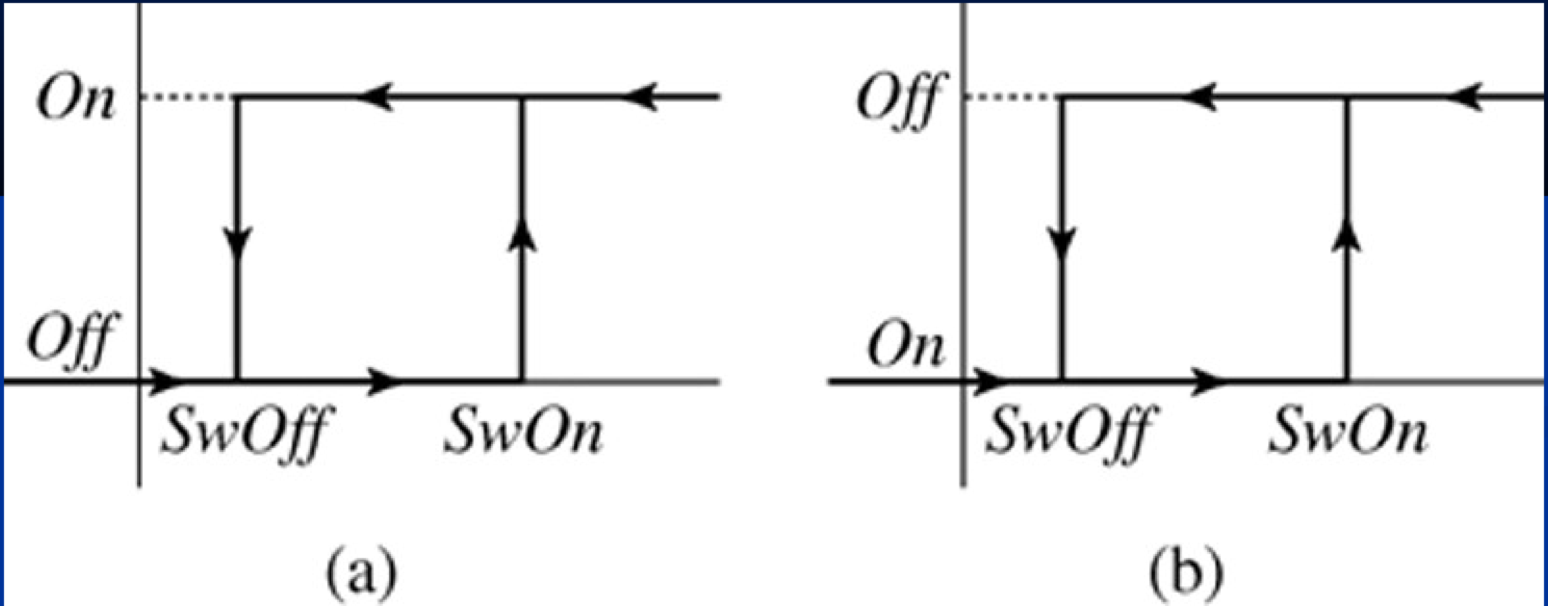
Simulink model for $v = (80/9)\cos(\pi t^2/100)$ with a Saturation block. Figure 10.4–5 on page 433.



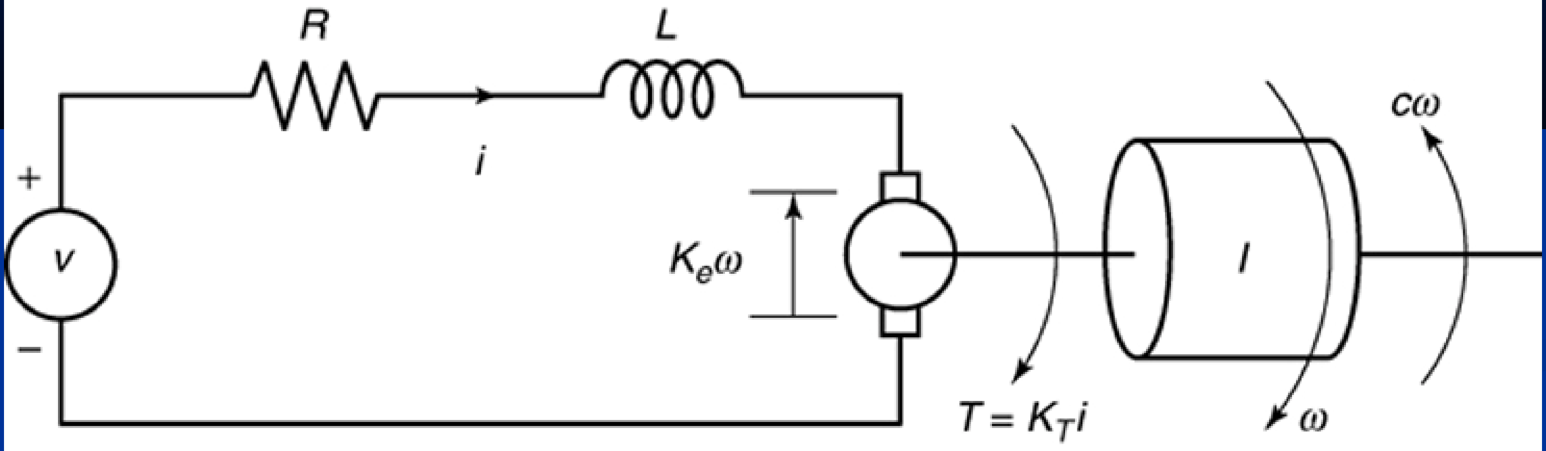
Speed response of the sled for $\theta = 0$ and $\theta \neq 0$. Figure 10.4–6 on page 434.



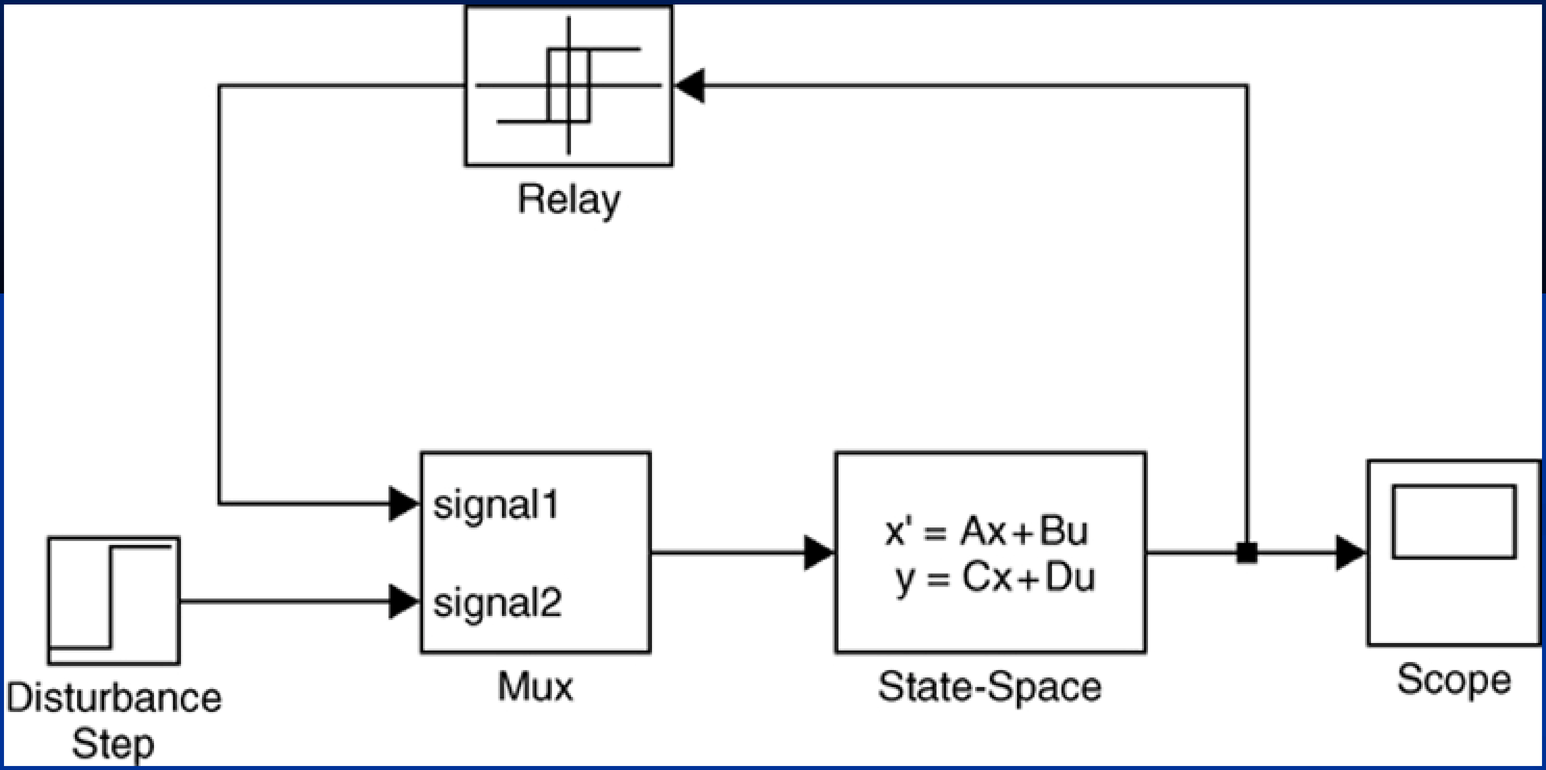
The relay function. (a) The case where *On* > *Off*. (b) The case where *On* < *Off*. Figure 10.4–7 on page 434.



An armature-controlled dc motor, Example 10.4-2. Figure 10.4–8 on page 435.

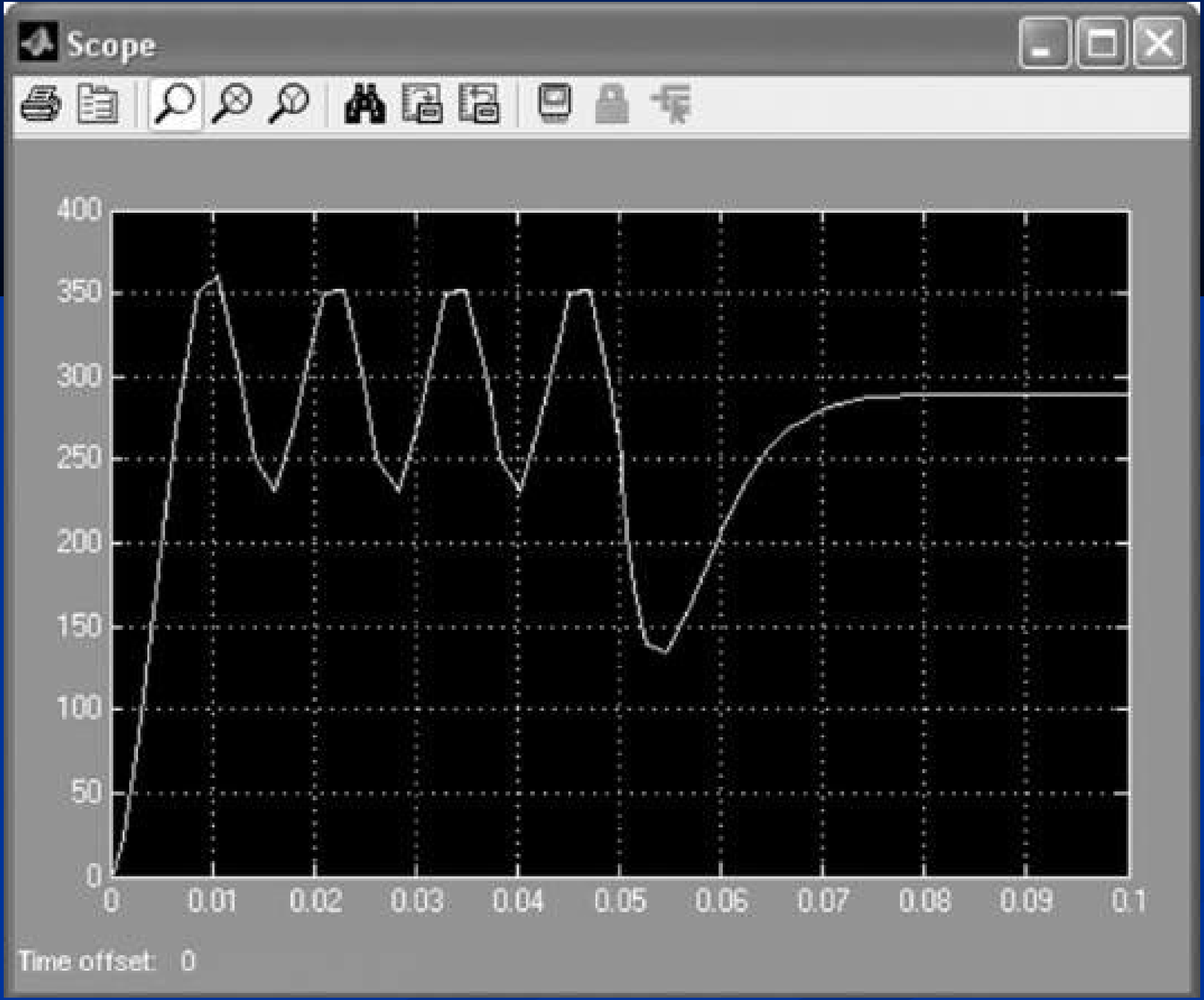


Simulink model of a relay-controlled motor. Figure 10.4–9 on page 436.

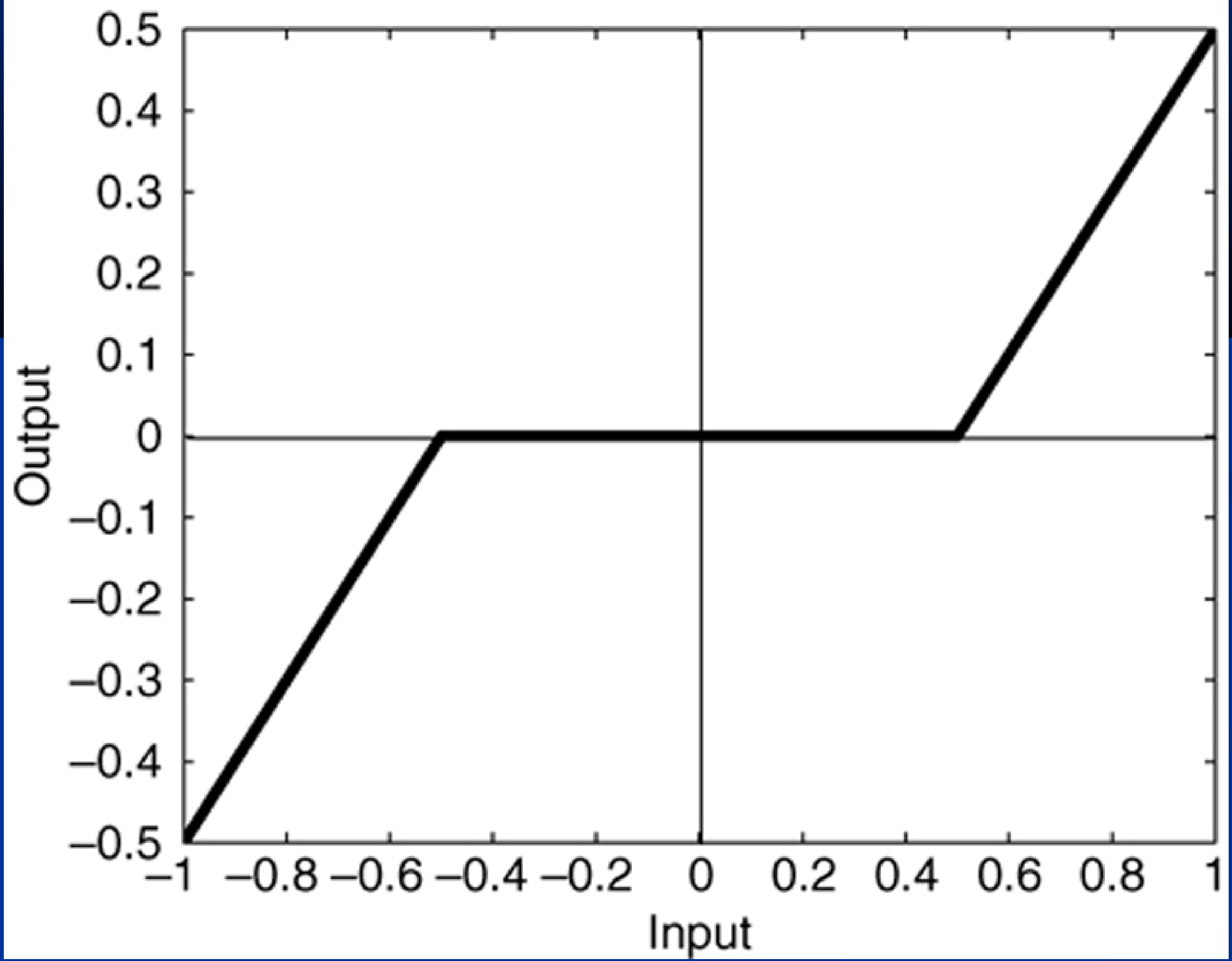


For the steps needed to construct this model, see Example 10.4-2 on pages 434-437.

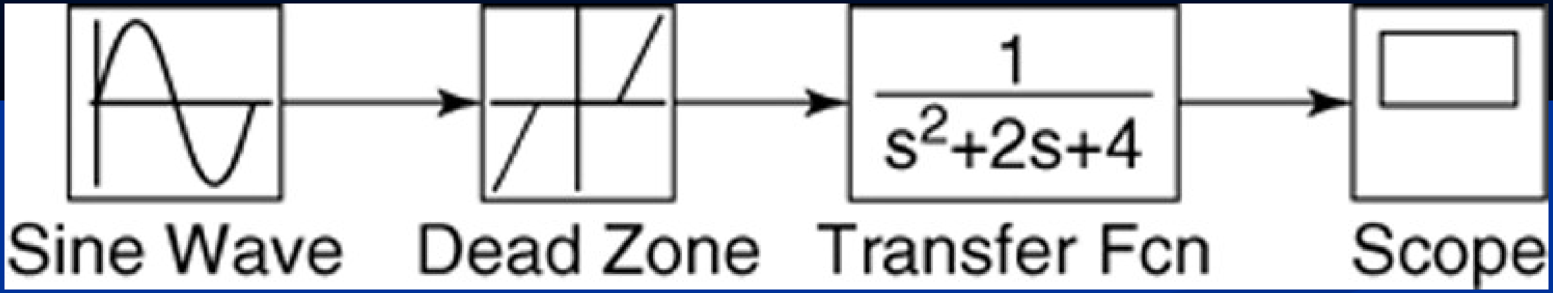
Scope display of the speed response of a relay-controlled motor. Figure 10.4–10 on page 437.



A dead-zone nonlinearity. Figure 10.5–1 on page 438.

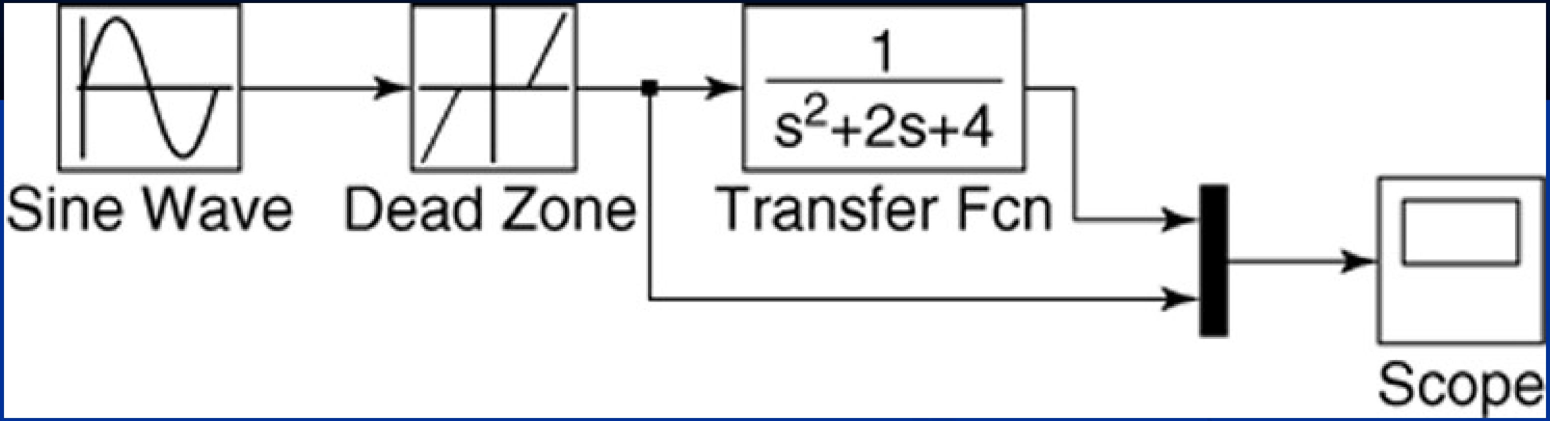


The Simulink model of dead-zone response. Figure 10.5–2



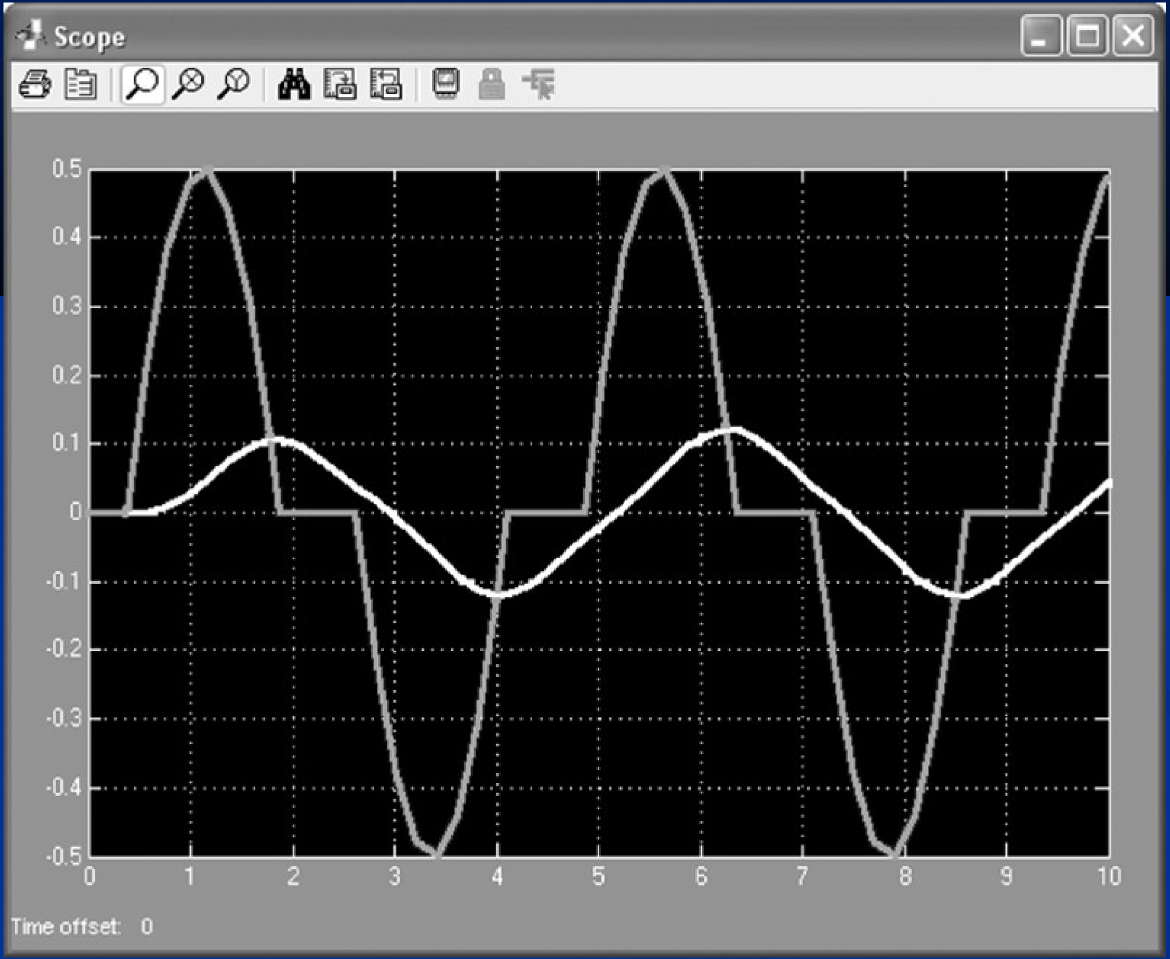
For the steps needed to construct this model, see Example 10.5-1 on pages 438-439.

Modification of the dead-zone model to include a Mux block.
Figure 10.5–3 on page 439.

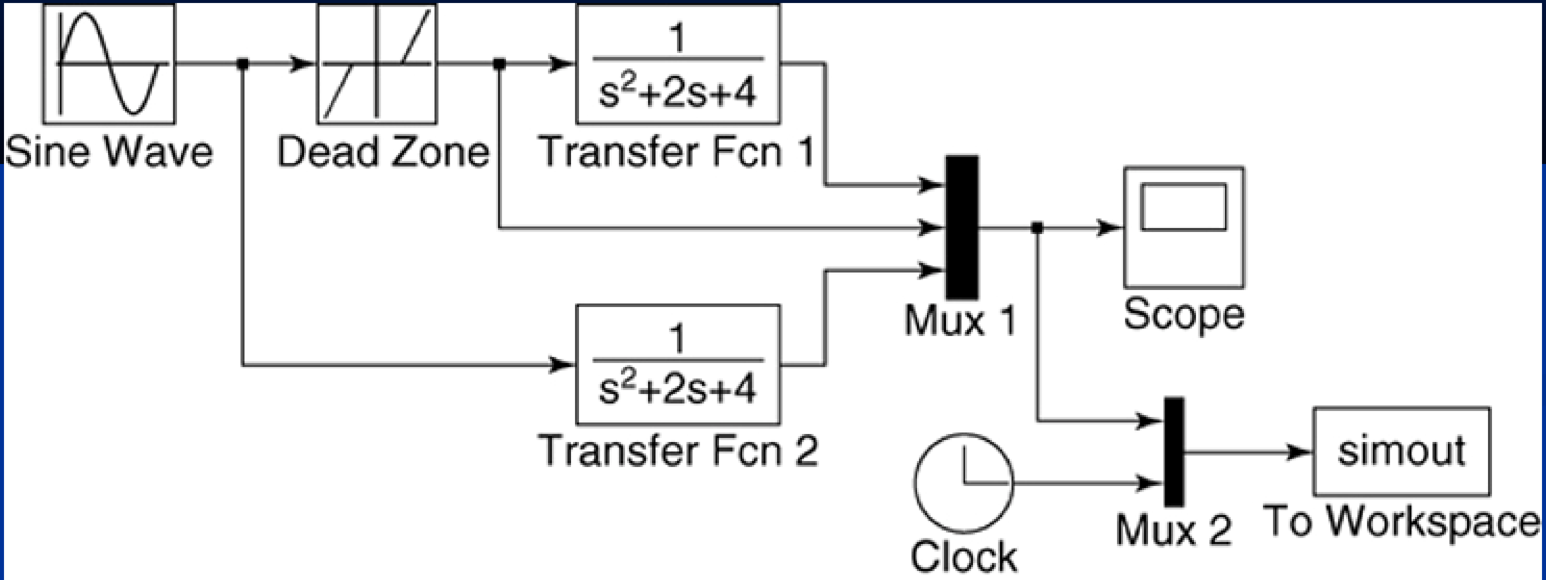


For the steps needed to construct this model, see Example 10.5-1 on page 439.

The response of the dead-zone model. Figure 10.5–4 on page 440.

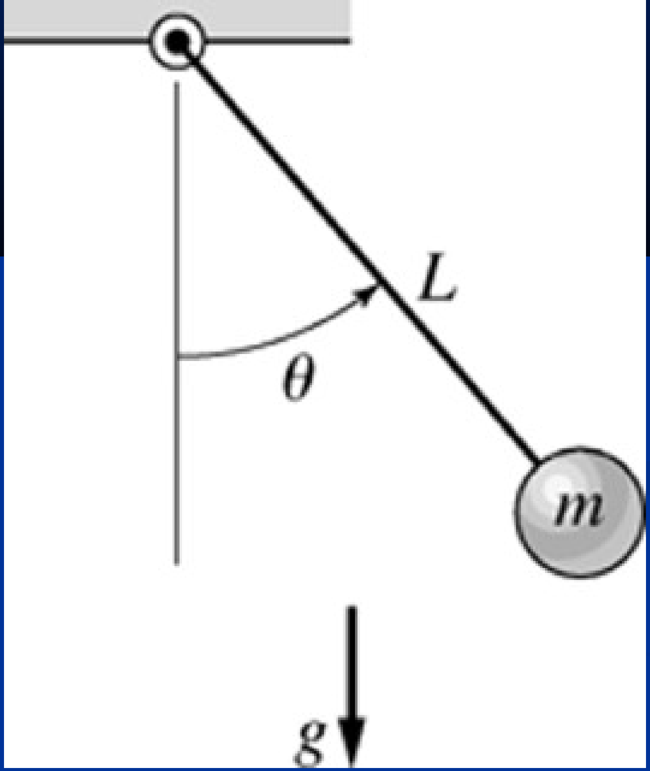


Modification of the dead-zone model to export variables to the MATLAB workspace. Figure 10.5–5 on page 440.

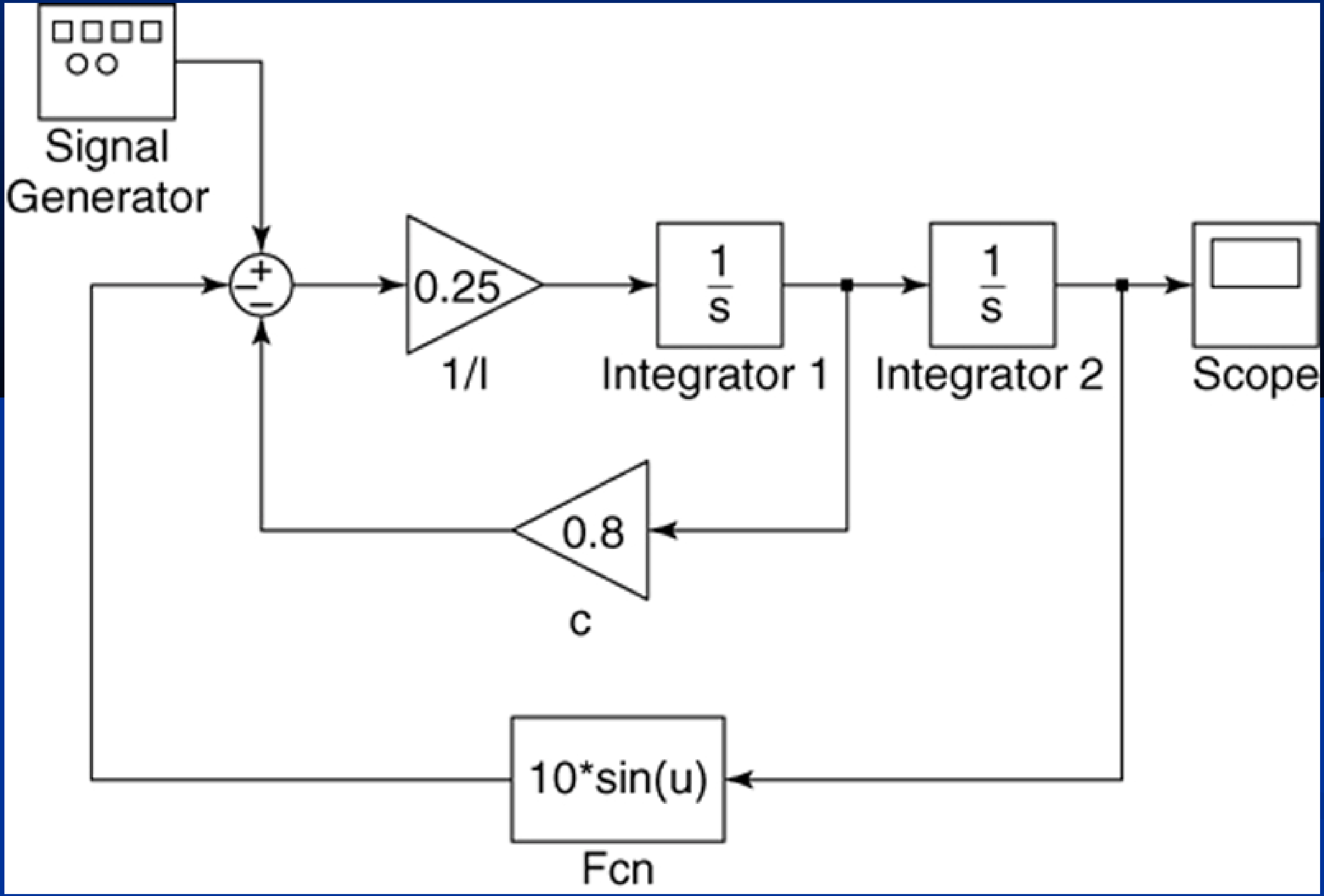


For the steps needed to construct this model, see pages 440-441.

A pendulum. Figure 10.6–1 on page 441.

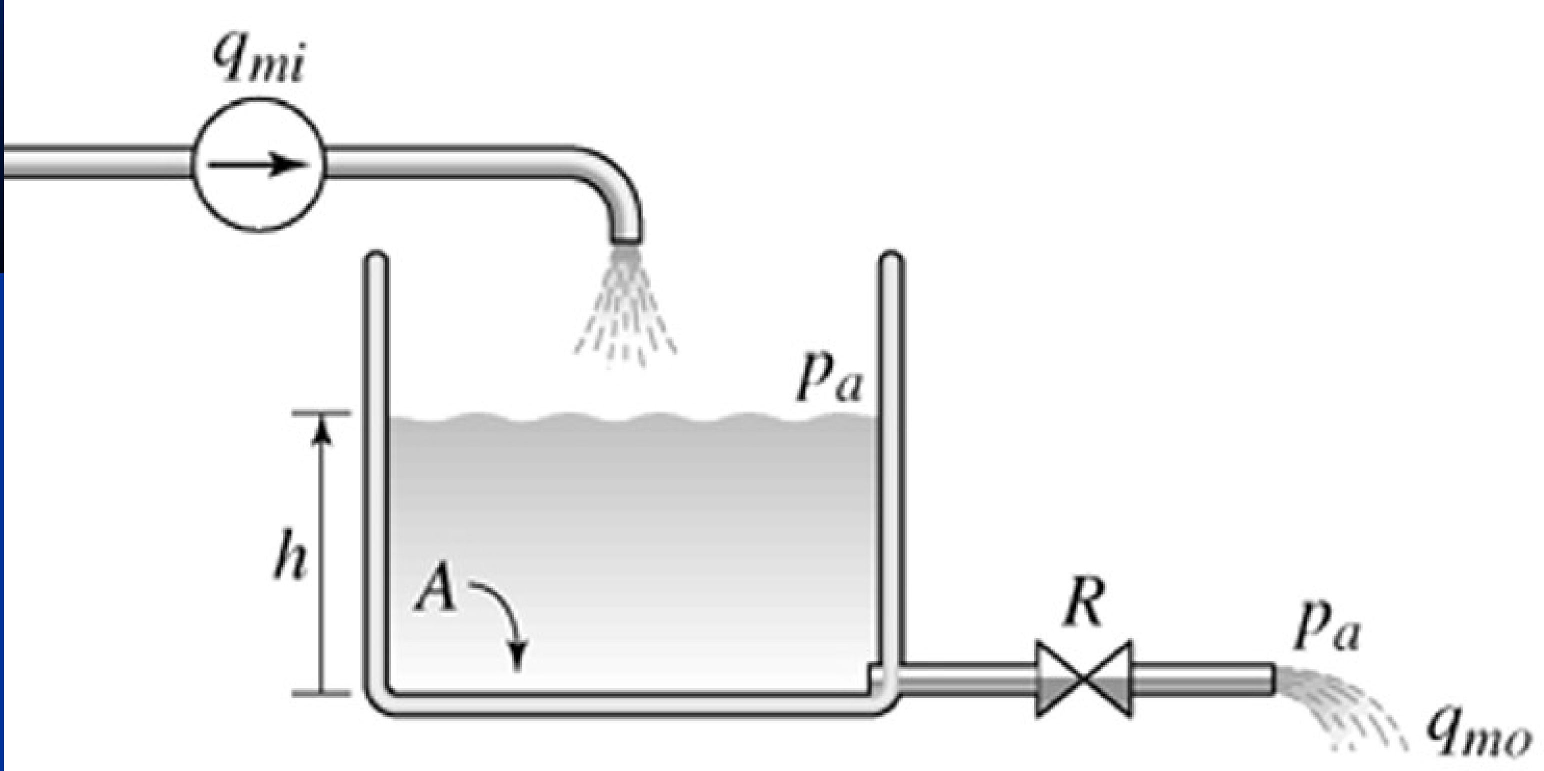


Simulink model of nonlinear pendulum dynamics. Figure 10.6–2 on page 442.



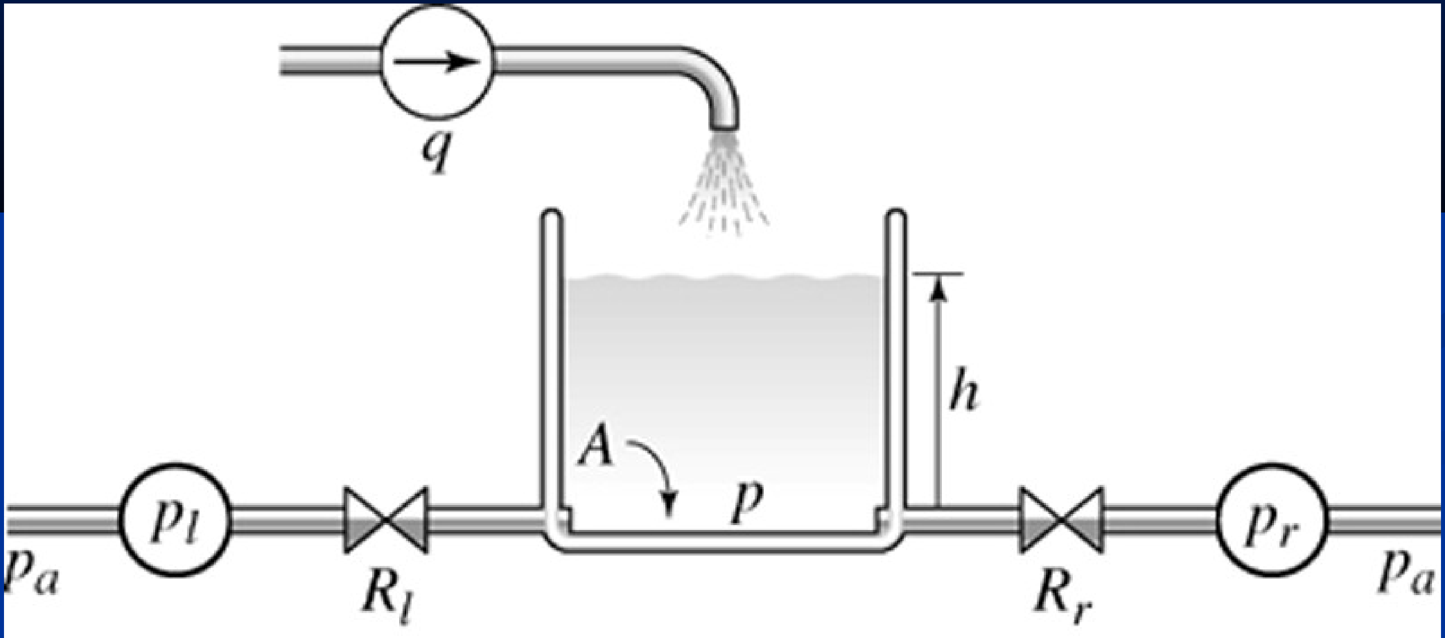
For the steps needed to construct this model, see pages 442-443.

Subsystem blocks. A hydraulic system with a flow source. Figure 10.7–1 on page 443.



This model is discussed on pages 443-444.

A hydraulic system with a flow source and two pumps. Figure 10.7–2 on page 444.



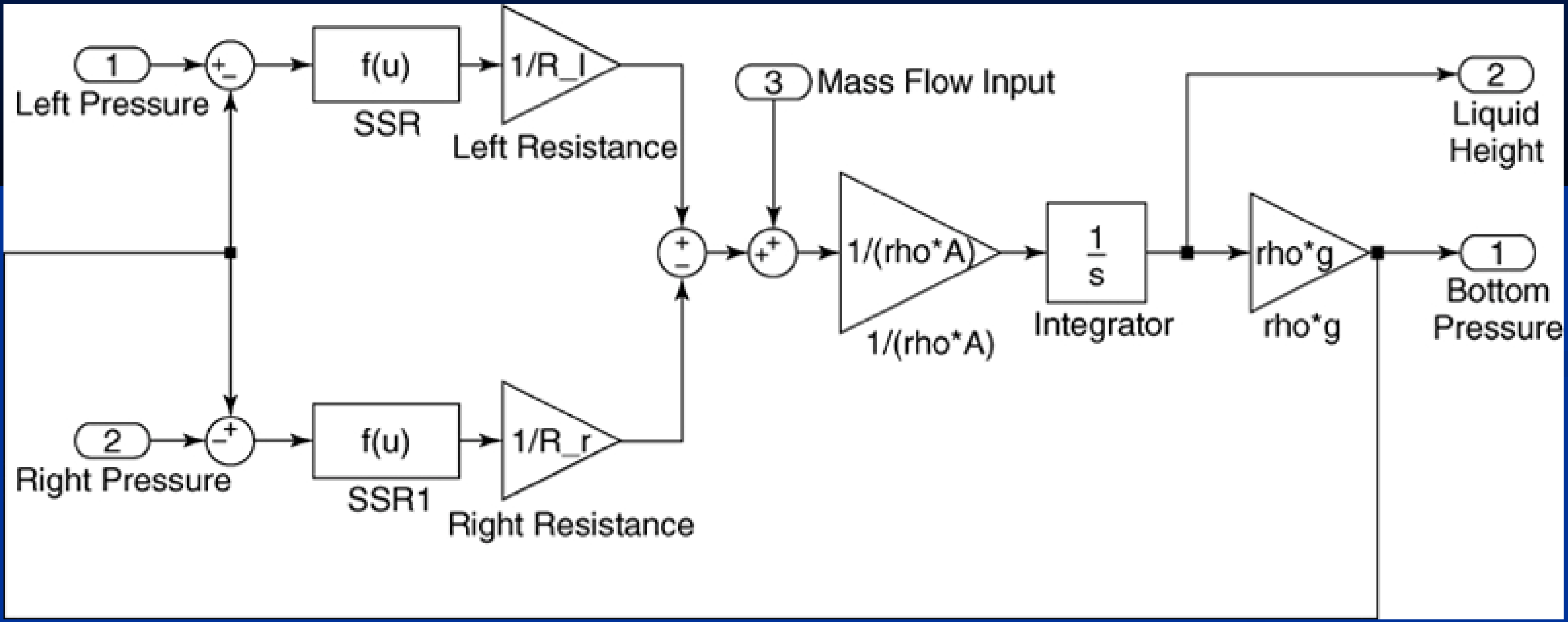
This model is discussed on pages 444-445.

You can create a subsystem block in one of two ways:

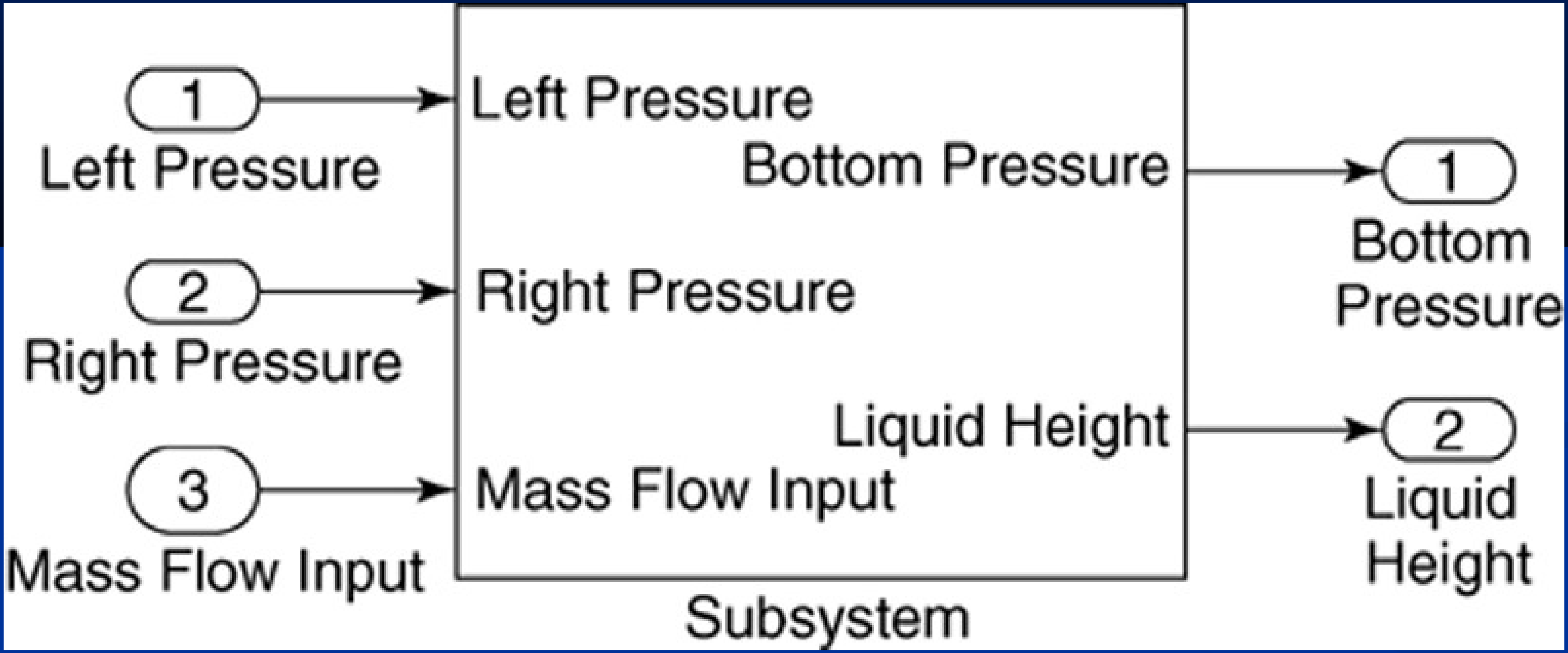
1. by dragging the Subsystem block from the library to the model window, or

2. by first creating a Simulink model and then “encapsulating” it within a bounding box.

Simulink model of the hydraulic system shown in Figure 10.7–2. Figure 10.7–3 on page 445.



The Subsystem block. Figure 10.7–4 on page 446.



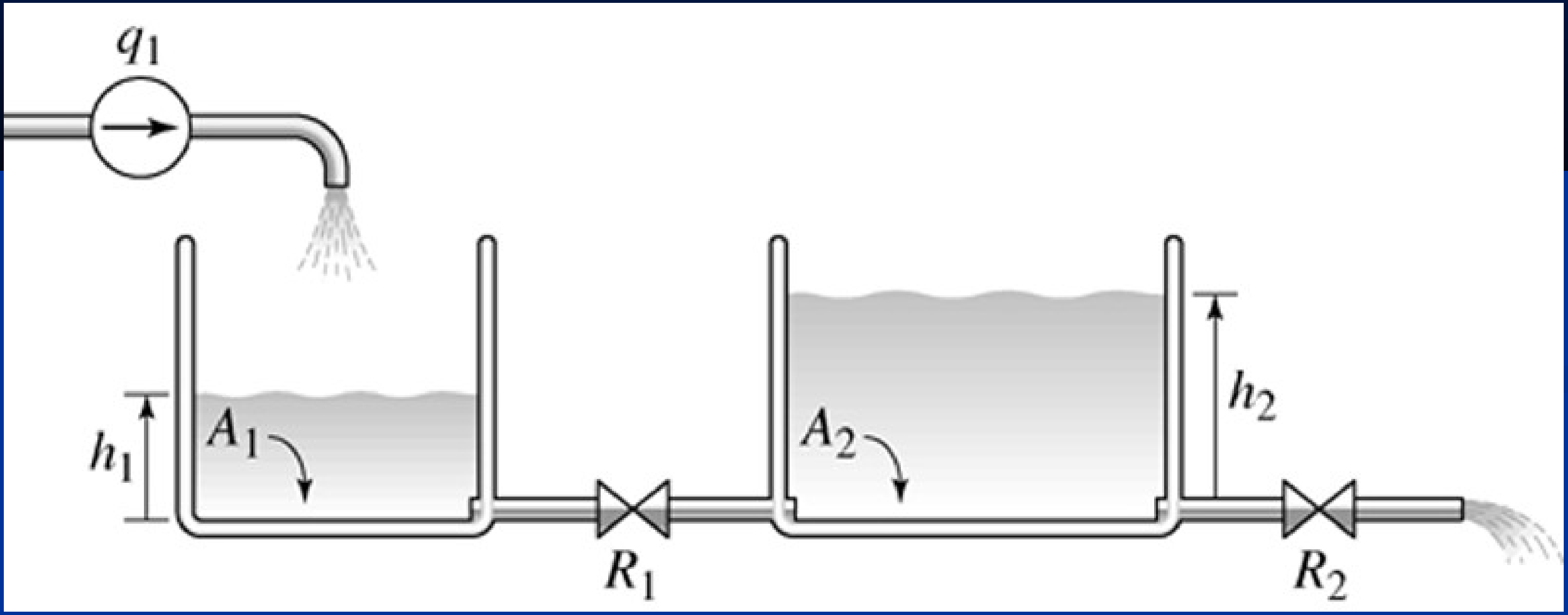
To create a subsystem block, create a “bounding box” surrounding the diagram.

Do this by placing the mouse cursor in the upper left, holding the mouse button down, and dragging the expanding box to the lower right to enclose the entire diagram.

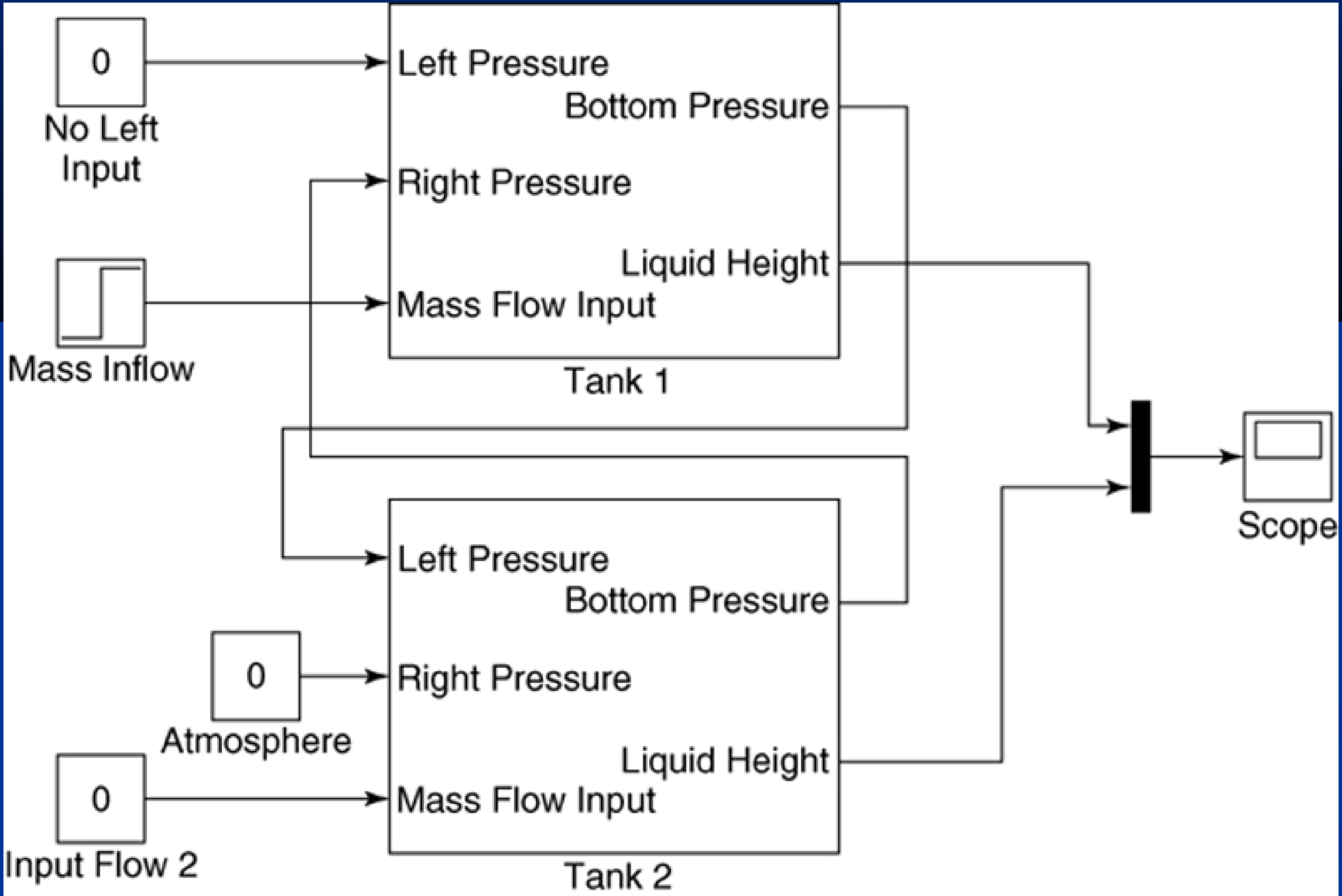
Then choose **Create Subsystem** from the **Edit** menu. Simulink will then replace the diagram with a single block having as many input and output ports as required and will assign default names.

You can resize the block to make the labels readable. You may view or edit the subsystem by double-clicking on it.

A hydraulic system with two tanks. Figure 10.7–5 on page 446.



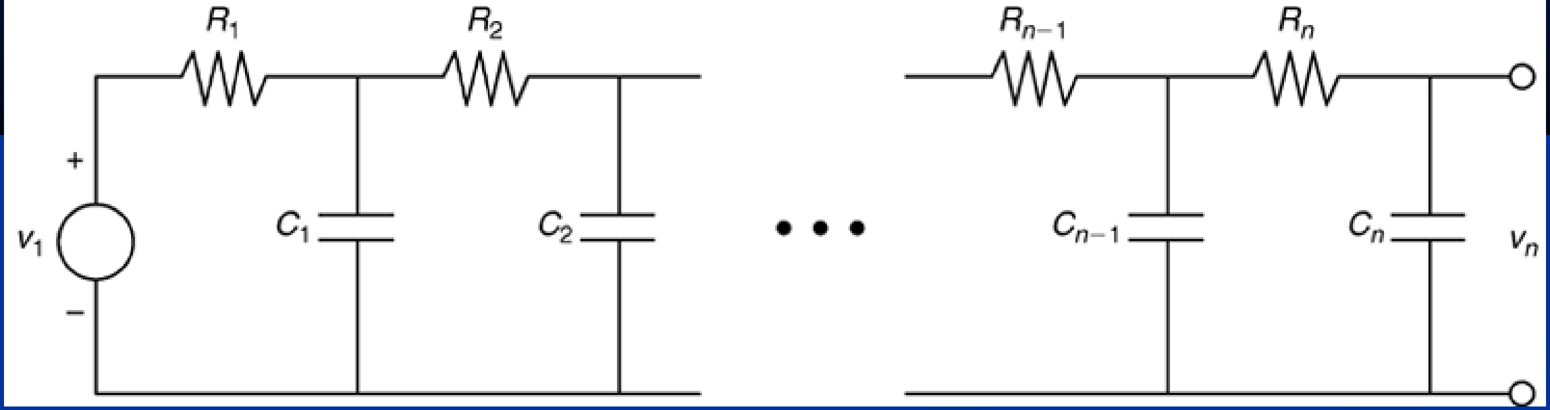
Simulink model of the system shown in Figure 10.7–5. Figure 10.7–6 on page 447.



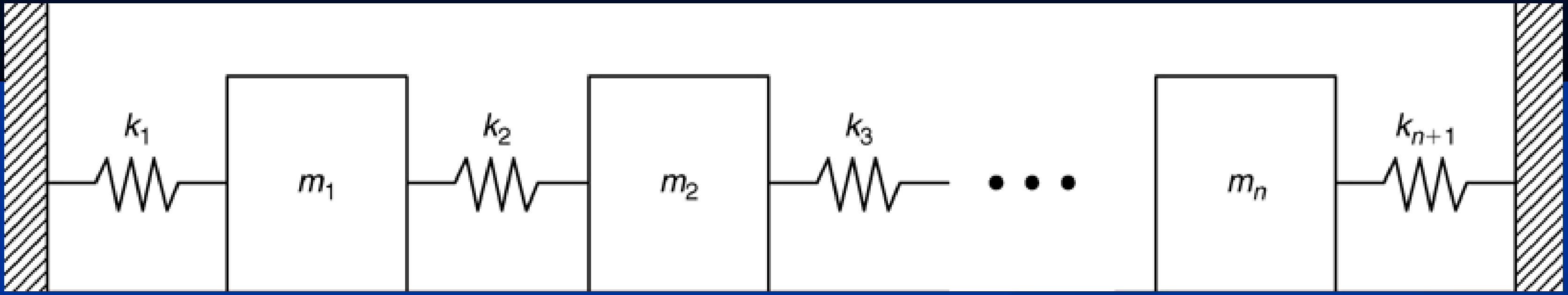
More? See pages 445-447.

Other applications of subsystem blocks.

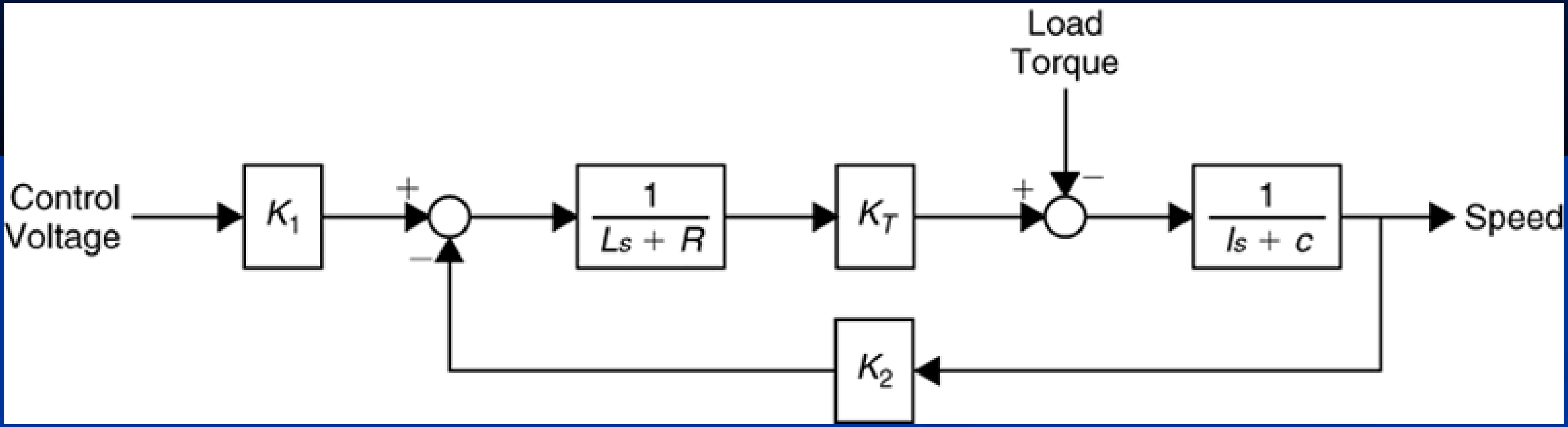
A network of RC loops. Figure 10.7–7 on page 448.



A vibrating system. Figure 10.7–8 on page 448.



An armature-controlled dc motor. Figure 10.7–9 on page 448.

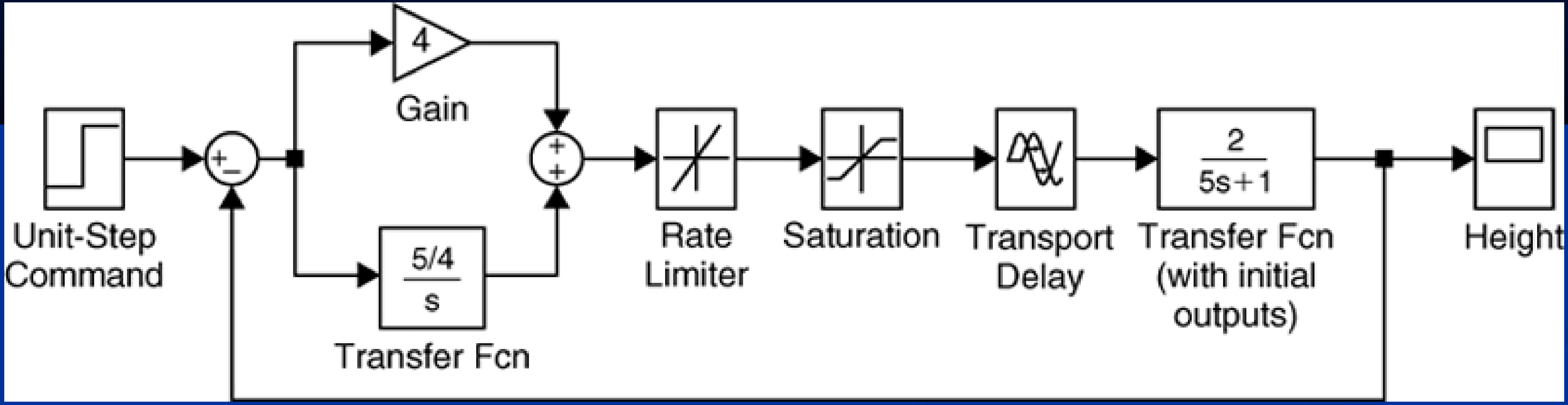


The “Transfer Fcn (with initial outputs)” block

The “Transfer Fcn (with initial outputs)” block, so-called to distinguish it from the Transfer Fcn block, enables us to set the initial value of the block output.

The “Transfer Fcn (with initial outputs)” block is equivalent to adding the free response to the block output, with all the block’s state variables set to zero except for the output variable.

Simulink model of a hydraulic system with dead time. Figure 10.8–1 on page 450.



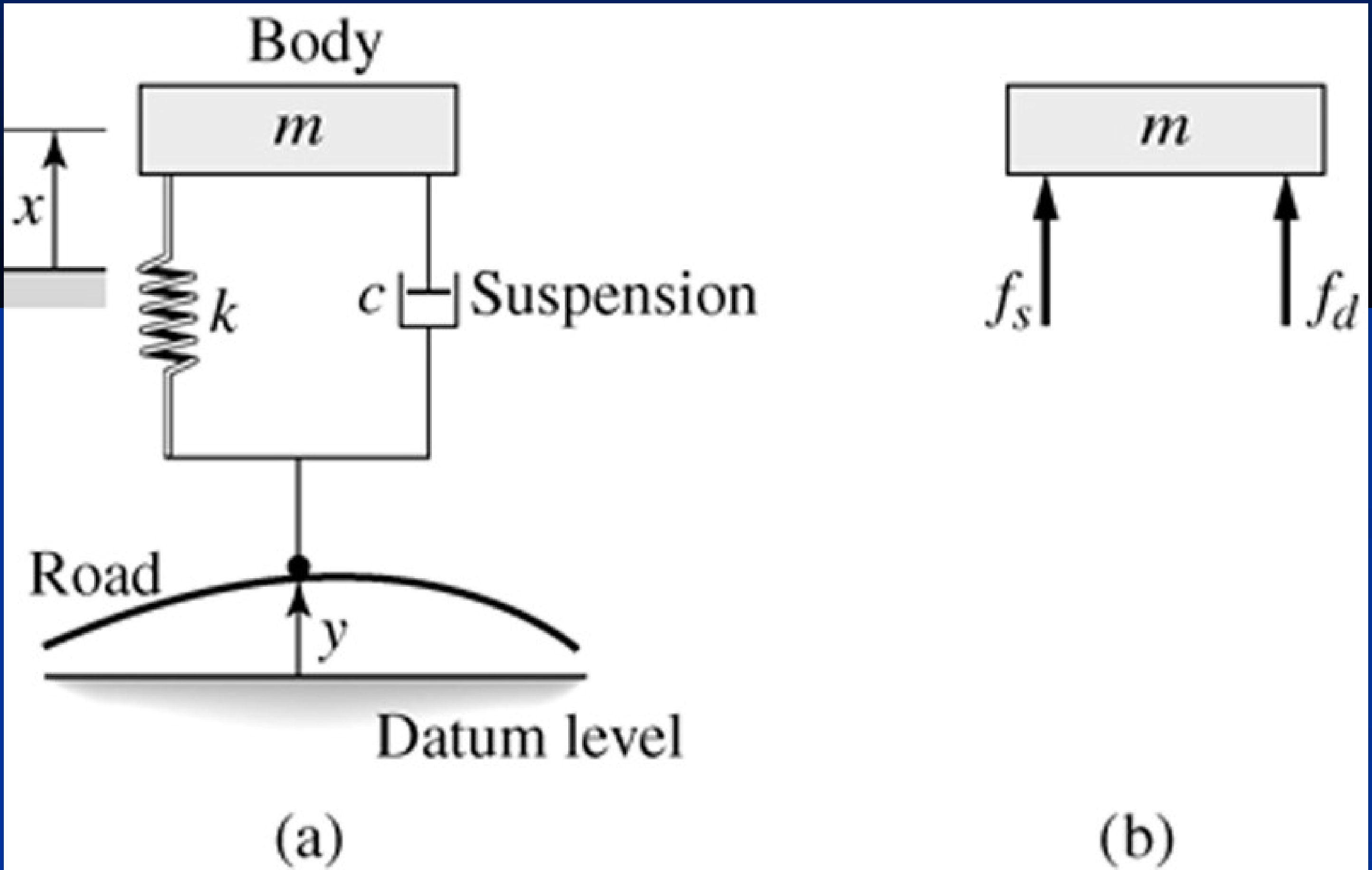
More? See pages 448-451.

Four additional Simulink elements that enable us to model a wide range of nonlinearities and input functions, namely,

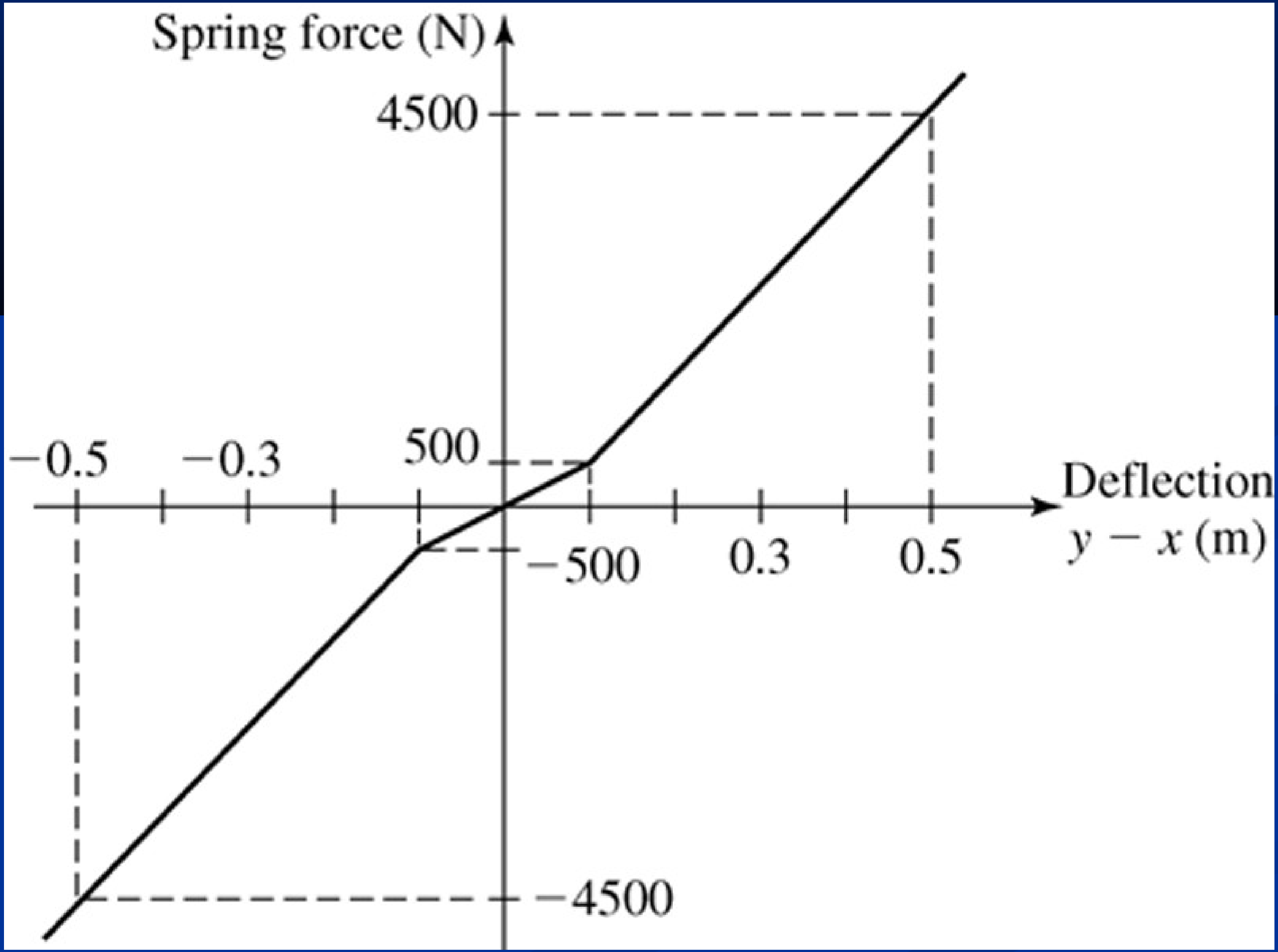
- the Derivative block,
- the Signal Builder block,
- the Look-Up Table block, and
- the MATLAB Fcn block.

- These are discussed in Section 10.9 along with a simulation of a nonlinear vehicle suspension model (pages 451-455).

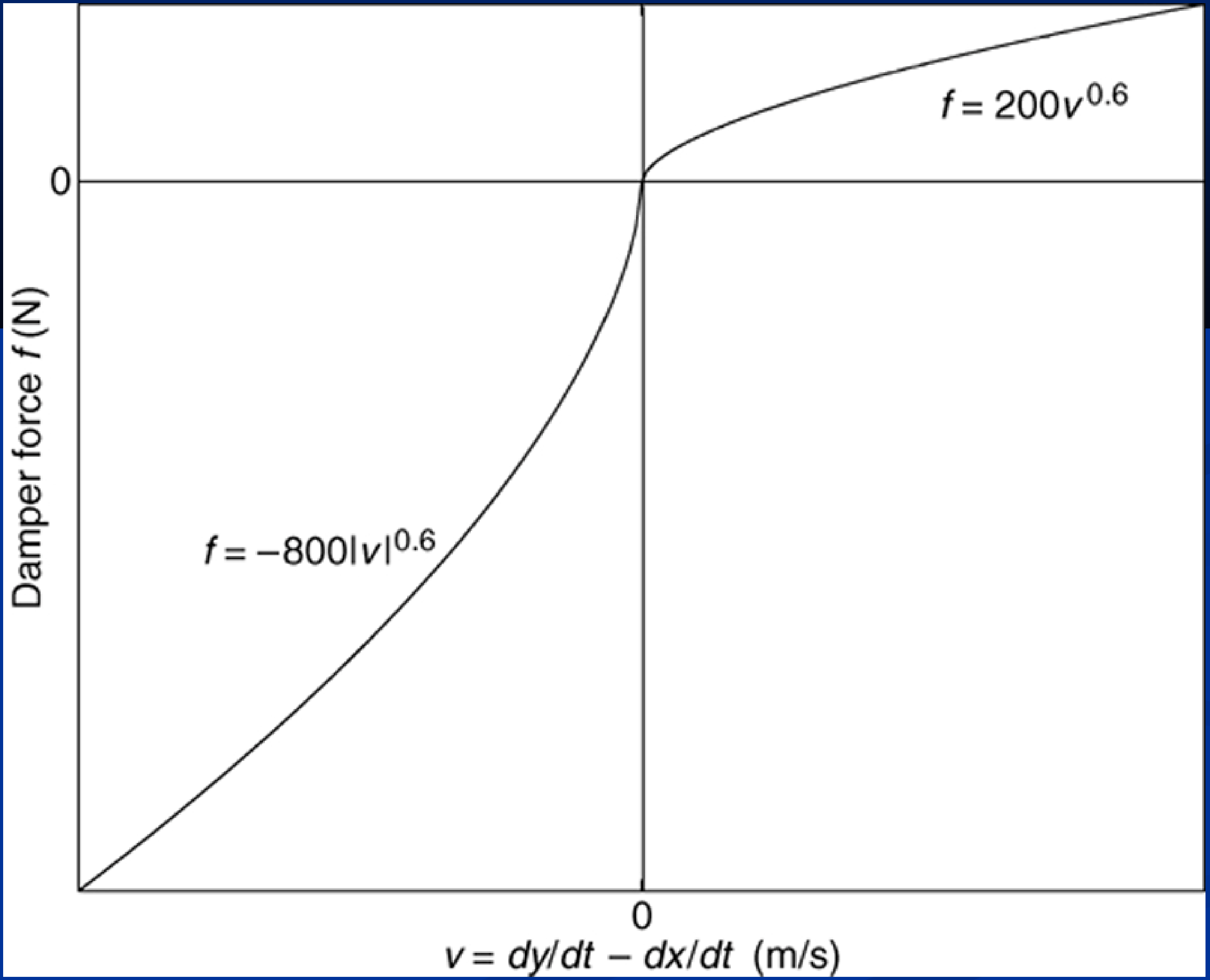
Single-mass model of a vehicle suspension. Figure 10.9–1 on page 451.



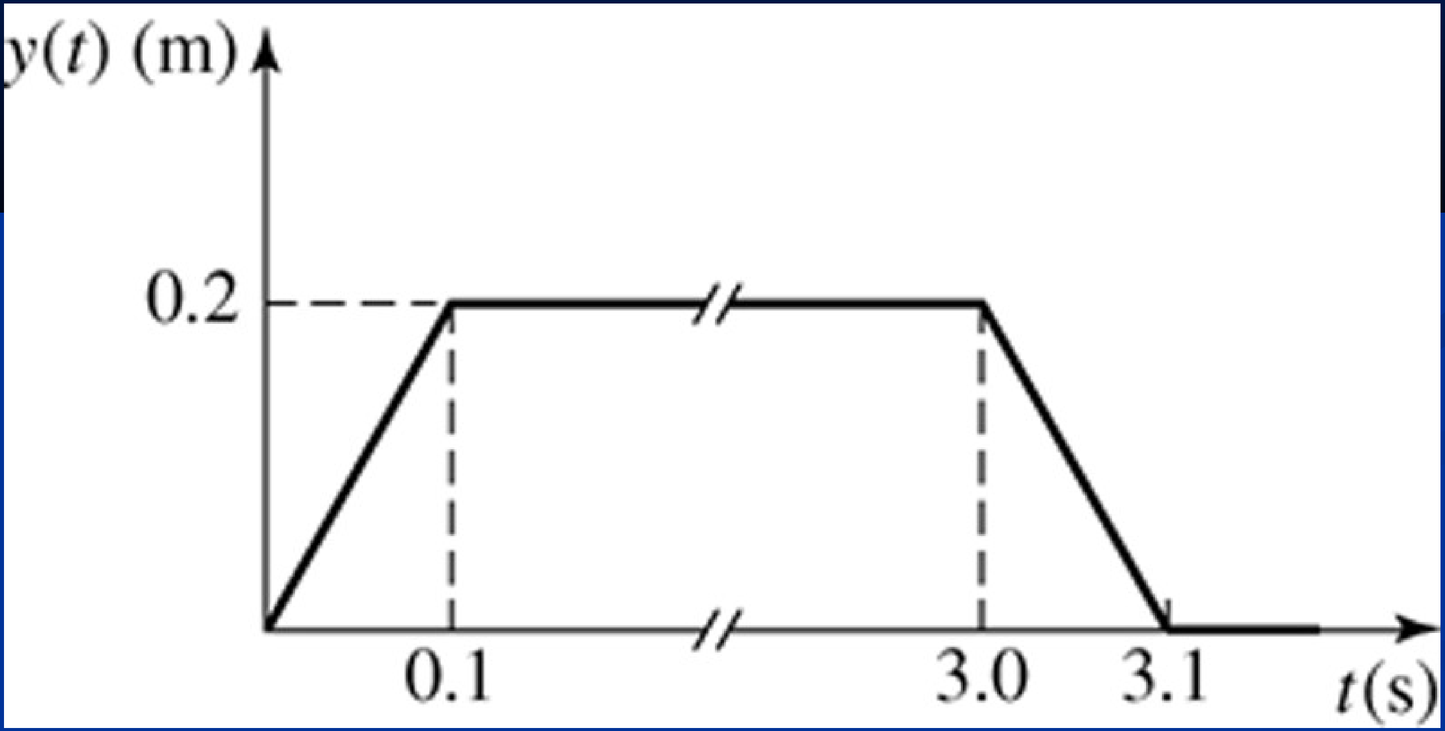
Nonlinear spring function. Figure 10.9–2 on page 452.



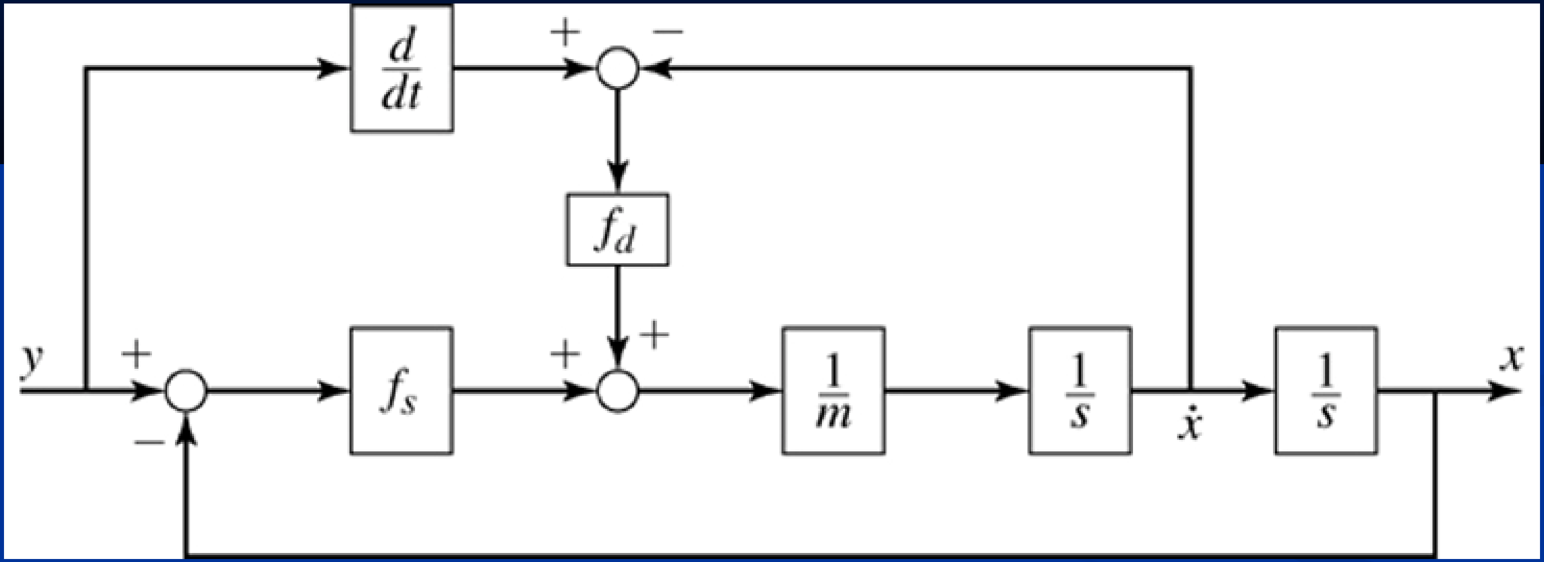
Nonlinear damping function. Figure 10.9–3 on page 452.



Road surface profile. Figure 10.9–4 on page 452.



Simulation diagram of a vehicle suspension model. Figure 10.9–5 on page 453.



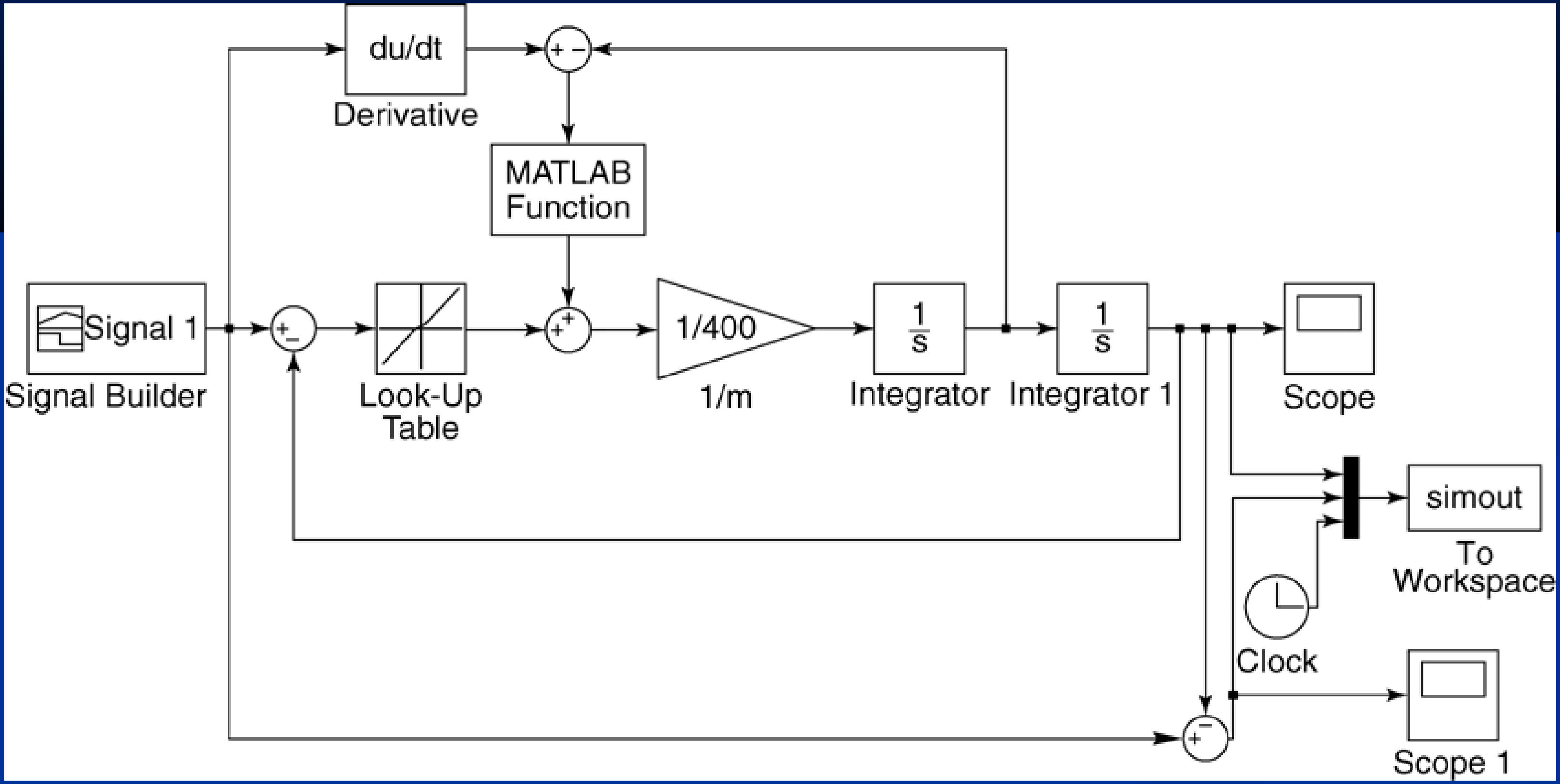
The simulation diagram shows that we need to compute the derivative *dydt*.

Because Simulink uses numerical and not analytical methods, it computes derivatives only approximately, using the Derivative block.

We must keep this approximation in mind when using rapidly changing or discontinuous inputs.

Simulink model of a vehicle suspension system.

Figure 10.9–6 on page 453.



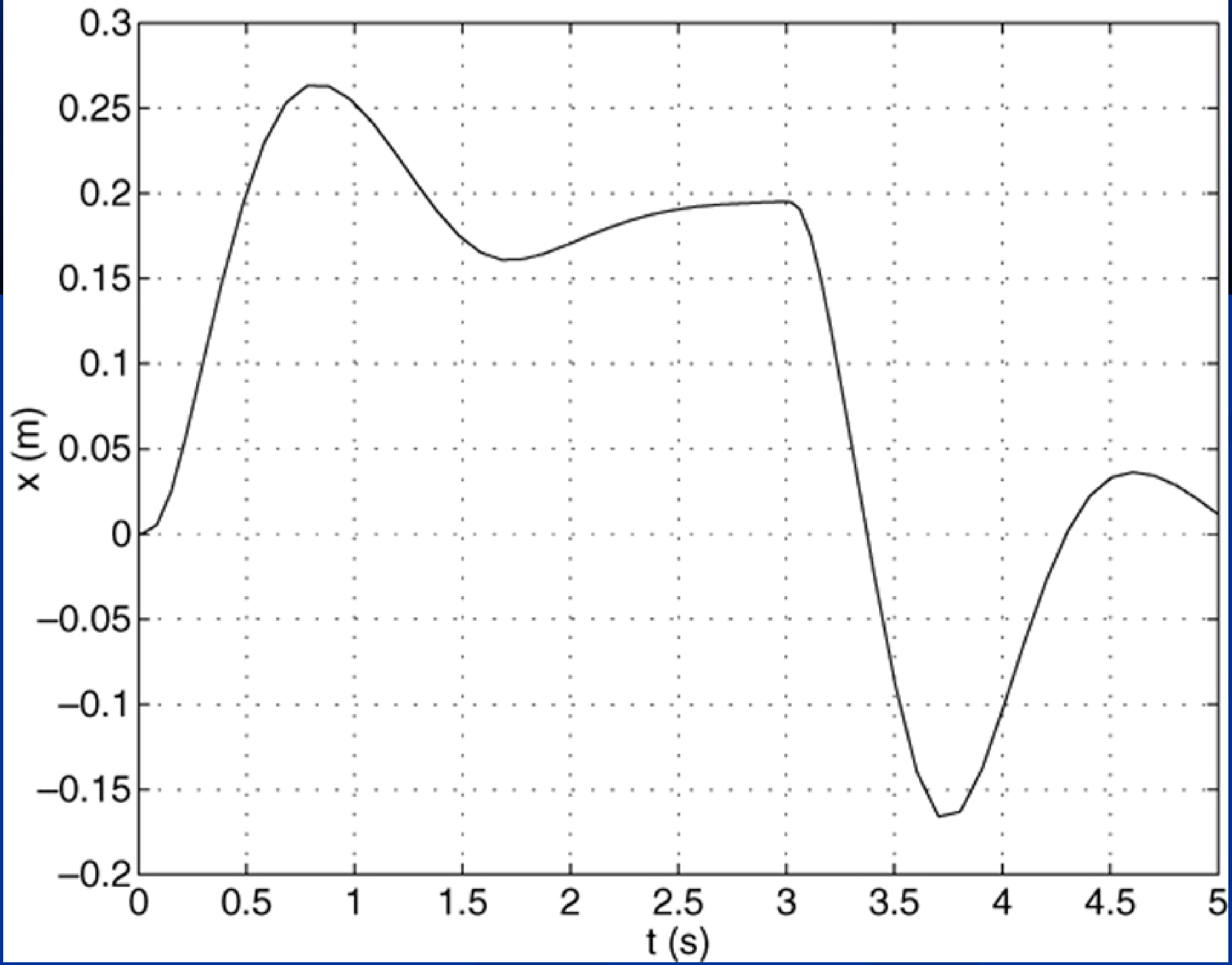
The Fcn, MATLAB Fcn, Math Function, and S-Function blocks can be used to implement functions, but each has its advantages and limitations.

The Fcn block can contain an expression, but its output must be a scalar, and it cannot call a function file.

The MATLAB Fcn block is slower than the Fcn block, but its output can be an array, and it can call a function file.

The Math Function block can produce an array output, but it is limited to a single MATLAB function and cannot use an expression or call a file. The S-Function block provides more advanced features, such as the ability to use C language code.

Output of the Simulink model shown in Figure 10.9–6. Figure 10.9–7 on page 455.



The following slides are figures from the chapter’s homework problems.

Figure P26 on page 460.

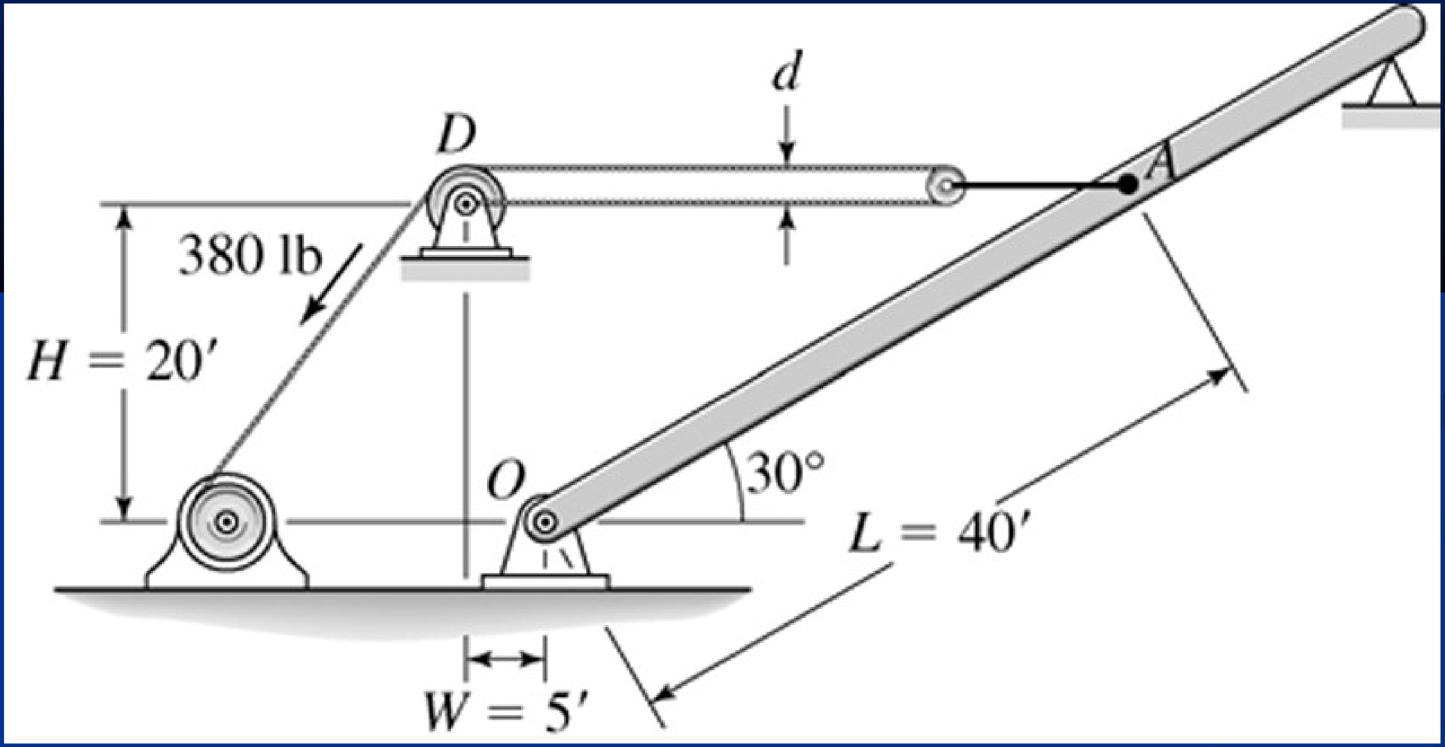


Figure P30 on page 462.

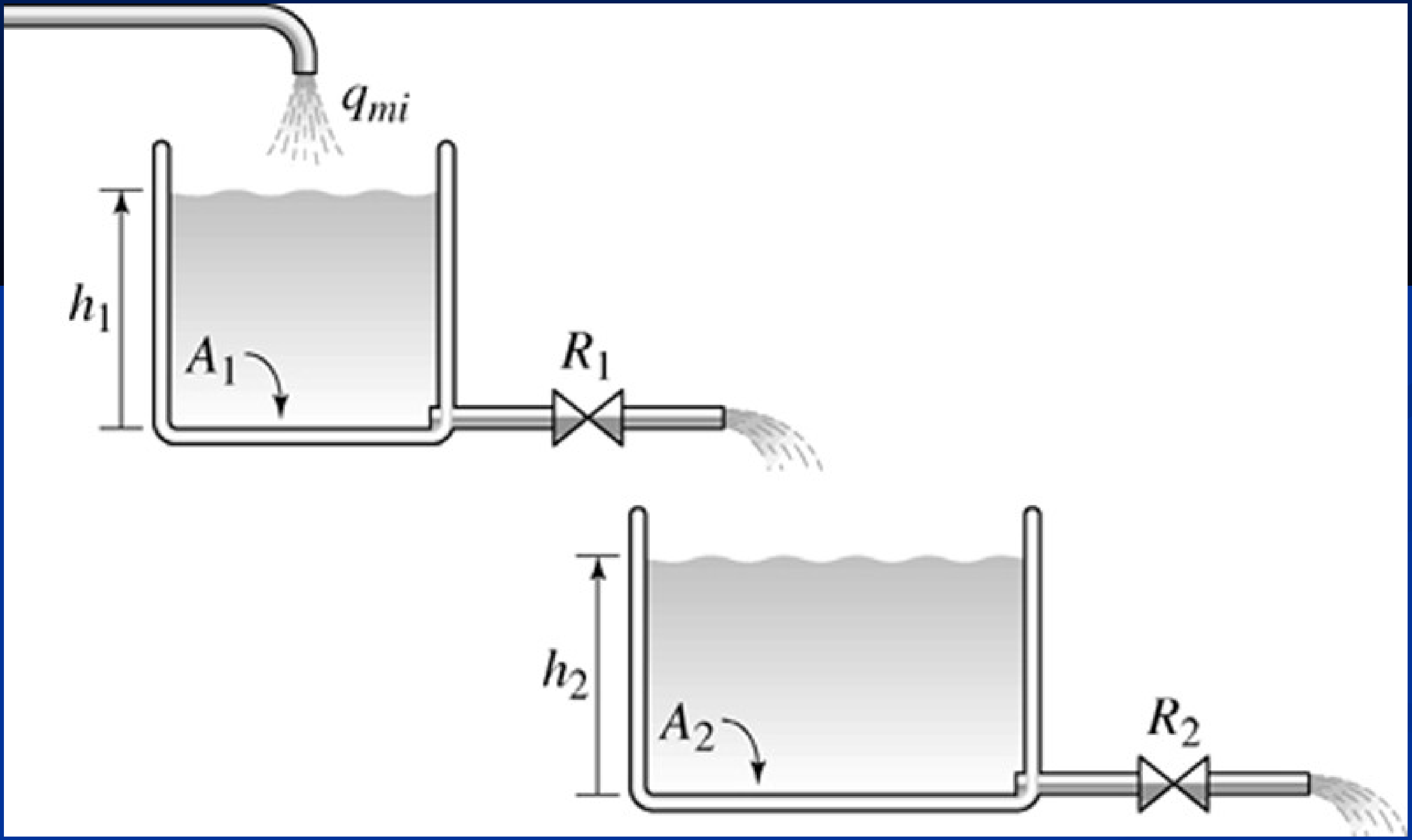


Figure P34 on page 463.

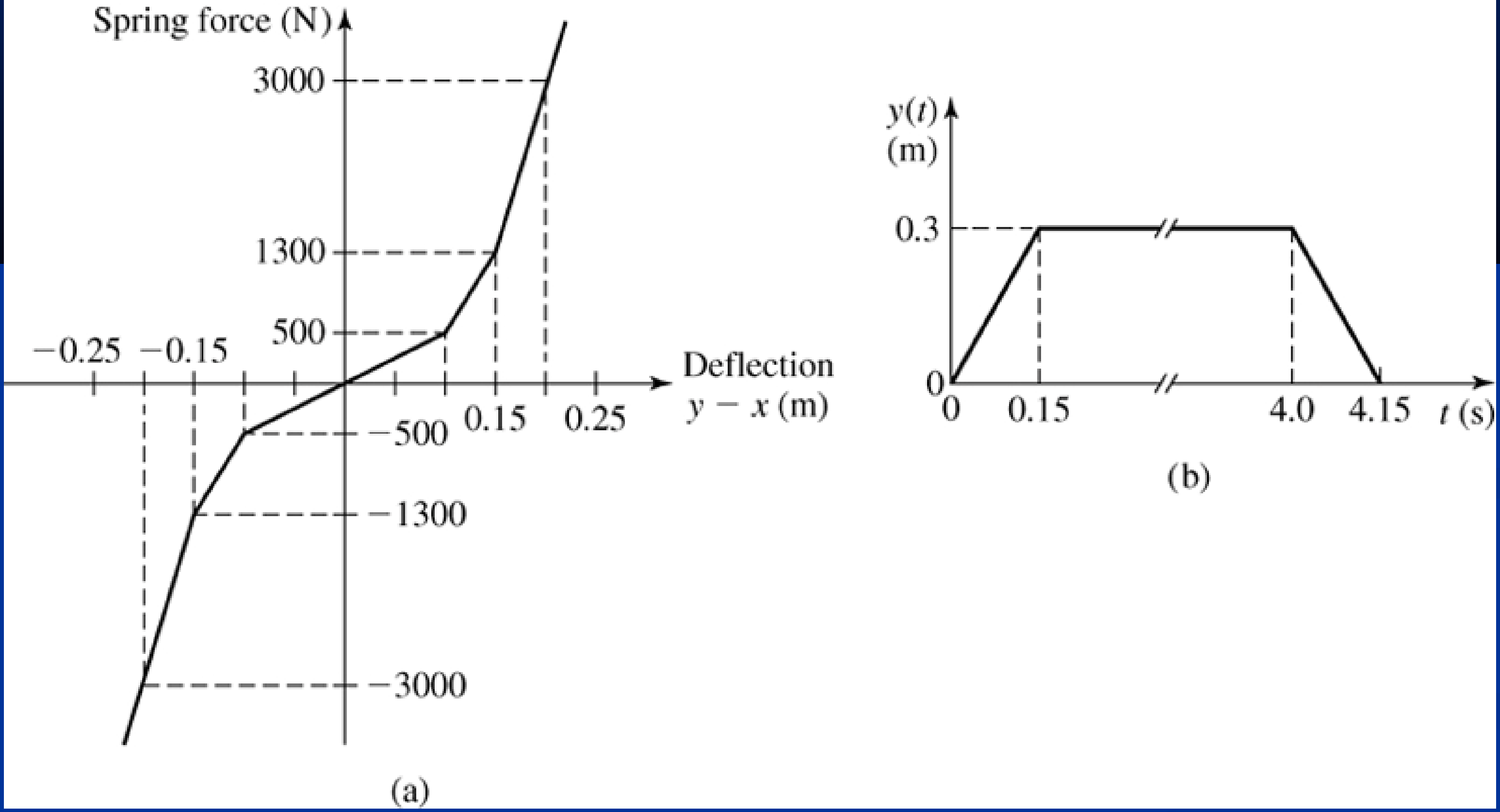
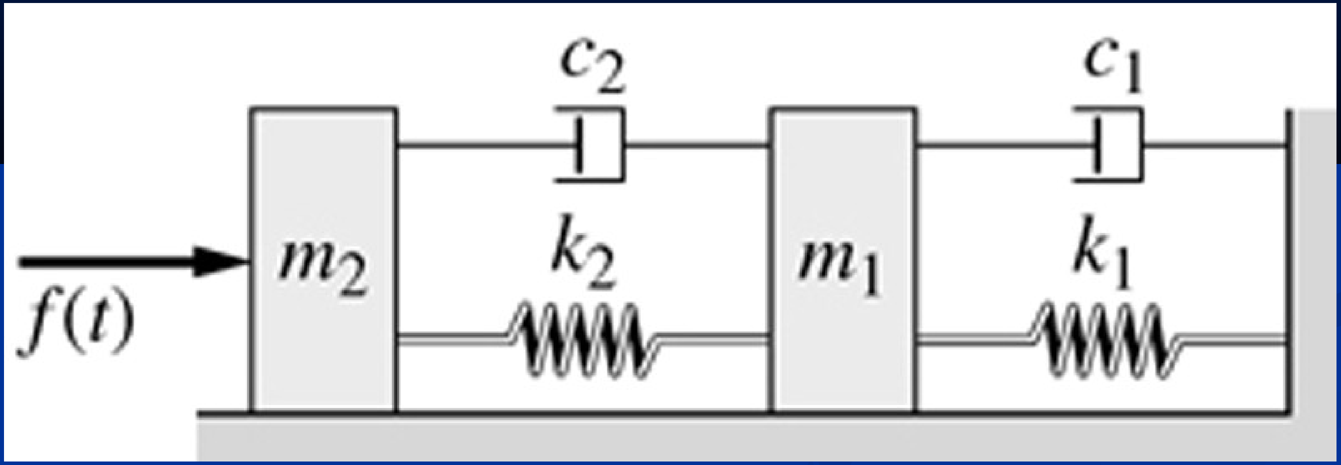


Figure P35 on page 463.


c_2
c_1
$f(t)$
m_2
k_2
m_1
k_1