SIMULINK is a graphical, block-diagram-oriented computer package for simulating dynamic systems. It is a product that is layered on top of MATLAB and uses many of MATLAB's functions. It can also interface with the MATLAB environment for maximum flexibility.

With SIMULINK, you can build a mathematical model of a system in much the same way as you might draw a block diagram. SIMULINK has a large library of component blocks from which you can select the basic components of your system model. In addition, there are numerous ways to add custom blocks and functionality into the system. Once the model has been constructed, time-domain simulations can be carried out in a variety of ways. Many integration methods are available, and the user has control over step size limits and convergence criteria. Simulation results can be viewed on SIMULINK "scopes" or ported to the MATLAB environment, where they can be plotted by MATLAB's powerful plotting routines.

The following pages are provided as a bare introduction to SIMULINK in the form of a simple tutorial. It should be sufficient for the student to become familiar with the basics. The intent here is not to provide a comprehensive manual for the package in this form. The new user is encouraged to experiment and make use of the on-line documentation provided by both SIMULINK and MATLAB.

SIMULINK TUTORIAL

In this tutorial, you will simulate a simple mass-spring-damper system, as is illustrated in Figure A3.1.
In this system, the input is the forcing function, \( f(t) \), and the output is the displacement of the mass, \( x(t) \). Table A3.1 summarizes the parameter values.

**Table A3.1 Parameter Values for Example System**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass ( (m) )</td>
<td>1</td>
<td>kg</td>
</tr>
<tr>
<td>Stiffness ( (k) )</td>
<td>100</td>
<td>N/m</td>
</tr>
<tr>
<td>Damping ( (b) )</td>
<td>1</td>
<td>N/m/s</td>
</tr>
</tbody>
</table>

Before proceeding to construct the model, a few observations about the system are in order. First, one should determine the order of the system. This is fundamental to the simulation because it determines how many integrators (states) are required. In modeling physical systems, it is appropriate to count the number of energy storing elements in the system and assign one state to each element. In the system considered here, there are two energy storing elements: the mass and the spring. Therefore, this is a second-order system the modeling of which will require two integrators. Further, it can be shown that a good choice of state variables is (a) displacement of the mass, which determines the amount of potential energy stored in the spring, and (b) velocity of the mass, which determines the amount of kinetic energy stored in the mass. The outputs of the two integrators will be the two state variables.

The state equations for this system are

\[
\dot{q}_1 = \frac{1}{m} [F(t) - kq_1 - bq_2]
\]

and

\[
\dot{q}_2 = q_1
\]

where \( q_1 \) is the velocity and \( q_2 \) is the displacement.

Construction of the simulation begins by starting the MATLAB program. From WINDOWS, double-click on the MATLAB icon in the Program Manager. The MATLAB command window will open, showing the >> prompt. At the >> prompt, you can enter any MATLAB command. To activate SIMULINK, simply type

\[
>> \texttt{simulink}
\]
A new window will open in the upper left-hand corner. Each icon in this window can be opened to reveal a group of blocks from which a simulation can be built. To start a new model, chose New from the File menu, and an empty window will open. This window will contain your model, which can be saved using the Save command from the File menu.

The File menu commands default to the MATLAB directory. Students are strongly advised to create their own directories for their own work to minimize confusion and the possibility of accidental loss of data.

Model construction begins by selecting appropriate block icons from the library and copying them to the new window by clicking and dragging. First, open the SOURCES library group and select the Signal Generator block. While holding down the left mouse button, drag the icon to your new window. Notice that the original Signal Generator is still there, and that a new copy has been placed in your window. This is a standard characteristic of SIMULINK. From the SINKS group, copy a “Scope” and a “To Workspace” block. The Scope block allows one to look at simulation variables as the simulation progresses, and the To Workspace block stores the data as a vector in the MATLAB workspace. This is useful for generating professional-looking plots using the MATLAB plot capabilities. Finally, open up the LINEAR library and copy over two integrators, three gains, and one summation block. To get multiple copies of a given block, you can either (a) repeat the drag and drop procedure outlined above for each block or (b) select the block you wish to copy in your window using the right mouse button and drag and drop it within your window. This will create a copy of the selected block.

Blocks are deleted by selecting the block and pressing the Delete key on the keyboard. After the above procedure, your window might look something like the assemblage of blocks shown in Figure A3.2.

Note that some of the blocks have only inputs, some have only outputs, and some have both. Note also that the blocks are designed assuming a general signal flow from left to right, so in Figure A3.2, the blocks are arranged with the source block on the left, the sink blocks on the right, and the other blocks placed in a somewhat arbitrary arrangement.

Now, begin to manipulate and connect the blocks so that they reflect the system as described mathematically above. The first state equation indicates that the derivative of the velocity (acceleration) is equal to a summation of three terms divided by the mass. In terms of simulations, it can be said that velocity (one of our states) is the integral of this summation divided by the mass. Our first step is to modify the summation block so that it can accept three inputs, not two. Double-click on the summation block and it will open into a dialog box with a single field that holds two plus signs: ++ . This indicates that the summation block will take two inputs, both with positive signs. The first state equation requires the summation of three signals: one positive and the other two negative. Therefore, modify the input field to + − − and click OK. The appearance of the summation block will change to reflect this modification. Now connect the signal generator to the first input of the summation, the output of the summation to a gain block, and the output of that gain block to the first integrator. Make the connections by positioning the cursor at the output of the first block, press and hold the left mouse button, and drag the connection to the desired input of the other block. After you make these connections, the model should look something like Figure A3.3.
Figure A3.2 Component blocks for the mass-spring-damper model, assembled to a new model window.

Figure A3.3 The first connections.
So far, so good. Now connect the output of the first integrator (which is velocity, of course) to the input of the second integrator. The output of this integrator is displacement (we trust you’re following along here). The displacement is what you wish to monitor, so connect the output of the displacement integrator to the input of the scope. It is also instructive to send the position to the MATLAB workspace as a vector. One can “tap” existing signal connections easily in SIMULINK. Position the cursor over an existing signal line at the location you need to “tap off.” Press the right-hand mouse button and drag the new connection. Using this method, make an additional connection for the displacement to the “To Workspace” block. Figure A3.4 shows what your model should look like now.

To complete the model, simply hook up the second two terms in the summation. Note that these are functions of velocity and displacement and, as such, are feedback loops in the integration. To make the diagram more readable, it would be nice if the other two gain blocks could be “turned around” so that they could allow signal flow from right to left. Fortunately, SIMULINK offers this possibility. Simply select the block, choose Orientation from the Style menu, and choose “Right to Left” from the submenu that pops up. Repeat this process for both of the remaining gain elements and hook them up between the velocity and summation blocks and the displacement and summation blocks to complete the simulation structure.

As a final step in building the simulation, rename the blocks with appropriate names so that the simulation is more easily understood. Click on the block name to highlight it, and edit the string. Figure A3.5 shows a completed simulation block diagram with appropriately named blocks.

The structure of the model is now complete. Take a moment to save the model file using the Save command from the File menu. Always save your work often to guard against

![Diagram](image_url)

**Figure A3.4** The model grows.
catastrophic crashes. There’s a wise saying in the computer business: “There are two kinds of computer users: those that have lost data and those that will.”

Now, let’s set up the parameters of the model. Refer to Table A3.1 to find the values of the gain blocks. The mass is 1 kg, so you can set up the 1/mass block to be 1.0 also. Double-click on the gain block to open the dialog box. The default value of this gain block (and all others) is 1.0. Click OK and move on to the Stiffness block. Table 1 indicates a value of 100 N/m. Modify this block accordingly and close it. The integrators can be modified to reflect nonzero initial conditions; double-click on them to verify that the initial conditions are both zero. Now, open up the signal generator and inspect its controls. It’s pretty self-explanatory, much like the signal generators one might find in an undergraduate dynamics or controls laboratory.

One last step before you are ready to simulate. Under the Simulation menu, choose parameters to open up the run-control dialog. In this box you can choose the simulation method (RK45 is probably the best general-purpose method), minimum and maximum step sizes, and error tolerance. Choosing a smaller value for tolerance will force the integration step size to smaller values until the minimum step size is met. Start and stop times can also be set here. For the purposes of this tutorial, the simulation will be terminated through keystrokes or mouse movements, so the defaults are fine here.

Now open up the scope and modify the scales so that the Horizontal Scale (time) is 10 (sec) and the Vertical Scale (displacement) is 0.05 (m). Open up the Signal Generator and set the Peak value to 1, Frequency to 0.2, and signal type to square-wave. Select START from the Simulation menu and watch the scope. You might have to click on the scope to bring it into the foreground of the simulation.
You should see the response of the mass to the alternating forcing function. You should observe that the system is very underdamped, with a large overshoot and gradual decay. After running for a few seconds, choose STOP from the simulation menu and move to the MATLAB command window. At the prompt, enter the command

```
>> who
```

to see what variables are present. You should see the variable `yout`, which is the output of your simulation, brought in through the "To Workspace" block. Examine the vector by entering the plot command

```
>> plot(yout)
```

Depending on the duration of your simulation, you should get a plot that looks something like Figure A3.6.

The x-axis of this plot is simply the index of the vector. In other words, there are 1000 elements in the `yout` vector plotted in Figure A3.6. Time is not explicitly recorded in this vector. To make sure you get the time-dependency correctly recorded, you must modify the simulation model using a "clock" block from the SOURCES library and a "mux" (multiplexer) block from the CONNECTIONS library. Figure A3.7 shows the approximate modification.

Run the model again with this modification and go back to the MATLAB command window. Enter the command

```
>> plot(yout(:,1),yout(:,2))
```
This command says "Plot the vector in the second column of \texttt{yout} as the $y$-coordinate versus the vector in the first column of \texttt{yout} as the $x$-coordinate." You can also add titles, labels, and grids to the plot with the following commands:

\begin{verbatim}
>> grid
>> title('A MATLAB PLOT')
>> xlabel('Time (sec)')
>> ylabel('Displacement (m)')
\end{verbatim}

The resulting plot should look something like Figure A3.8.

One final note: By default, the "To Workspace" block holds only the last 1000 data points and discards the oldest data. This is why the plot in Figure A3.8 does not start at zero. You can easily modify this limit by double-clicking on the block.
Figure A3.8  MATLAB plot with proper time scale.