SIMULINK

Simulink provides access to an extensive set of blocks that accomplish a wide range of functions useful for the simulation and analysis of dynamic systems. The blocks are grouped into libraries, by general classes of functions.

- Mathematical functions such as summers and gains are in the Math library.
- Integrators are in the Continuous library.
- Constants, common input functions, and clock can all be found in the Sources library.
- Scope, To Workspace blocks can be found in the Sinks library.

Simulink is a graphical interface that allows the user to create programs that are actually run in MATLAB. When these programs run, they create arrays of the variables defined in Simulink that can be made available to MATLAB for analysis and/or plotting. The variables to be used in MATLAB must be identified by Simulink using a ***To Workspace*** block, which is found in the Sinks library. (When using this block, open its dialog box and specify that the save format should be Matrix, rather than the default, which is called Structure.) The Sinks library also contains a ***Scope***, which allows variables to be displayed as the simulated system responds to an input. This is most useful when studying responses to repetitive inputs.

Simulink uses blocks to write a program. Blocks are arranged in various libraries according to their functions. Properties of the blocks and the values can be changed in the associated dialog boxes. Some of the blocks are given below.

CLASSES

There are two major classes of items in Simulink: ***Blocks*** and ***Lines***. Blocks are used to generate, modify, combine, output, and display signals. Lines are used to transfer signals from one block to another.

LINES

Lines transmit signals in the direction indicated by the arrow. Lines must always transmit signals from the output terminal of one block to the input terminal of another block. On exception to this is a line can tap off of another line, splitting the signal to each of two destination blocks, as shown in Figure 1.

Lines can never inject a signal *into* another line; lines must be combined through the use of a block such as a ***Sum*** block, which will be described shortly..

A signal can be either a scalar signal or a vector signal. For Single-Input, Single-Output (SISO) systems, scalar signals are generally used. For Multi-Input, Multi-Output (MIMO) systems, vector signals are often used, consisting of two or more scalar signals. The lines used to transmit scalar and vector signals are identical. The type of signal carried by a line is determined by the blocks on either end of the line.

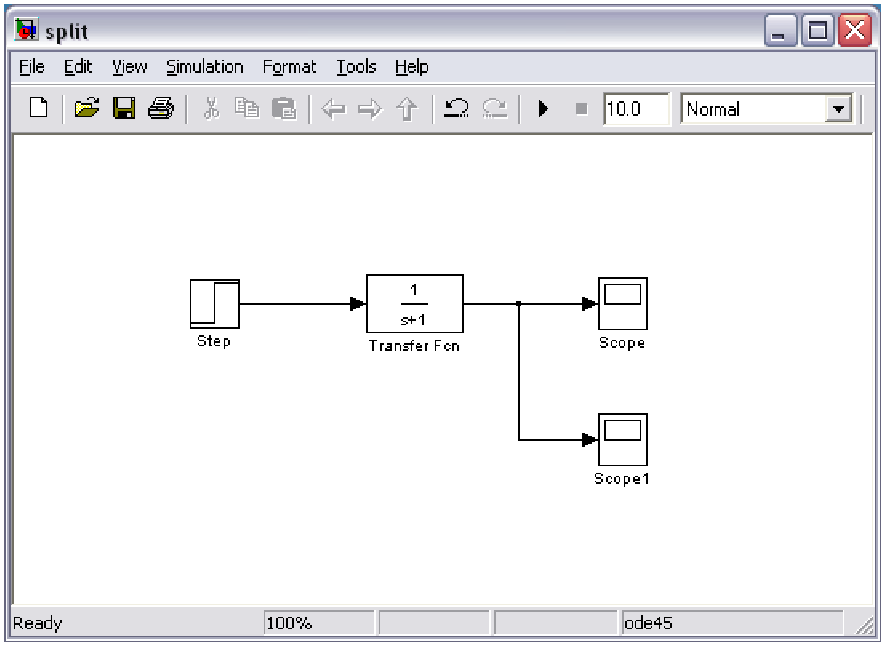

Figure 1: Example of lines that are split.

<u>BLOCKS</u>

There are several general classes of blocks within the Simulink library:

- Sources: used to generate various signals
- Sinks: used to output or display signals
- Continuous: continuous-time system elements (transfer functions, state-space models, PID controllers, etc.)
- Discrete: linear, discrete-time system elements (discrete transfer functions, discrete state-space models, etc.)
- Math Operations: contains many common math operations (gain, sum, product, absolute value, etc.)
- Ports & Subsystems: contains useful blocks to build a system

Blocks have zero to several input terminals and zero to several output terminals. Unused input terminals are indicated by a small open triangle. Unused output terminals are indicated by a small triangular point. The block shown in Fig. 2 has an unused input terminal on the left and an unused output terminal on the right.

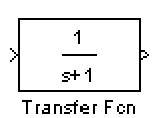

Figure 2: A transfer block with an unused input and output terminal.

SUM (Math library)

A dialog box obtained by double-clicking on the ***Sum*** block performs the configuration of the ***Sum*** block, allowing any number of inputs and the sign of each. The sum block can be represented in two ways in Simulink, by a circle or by a rectangle. Both choices are shown

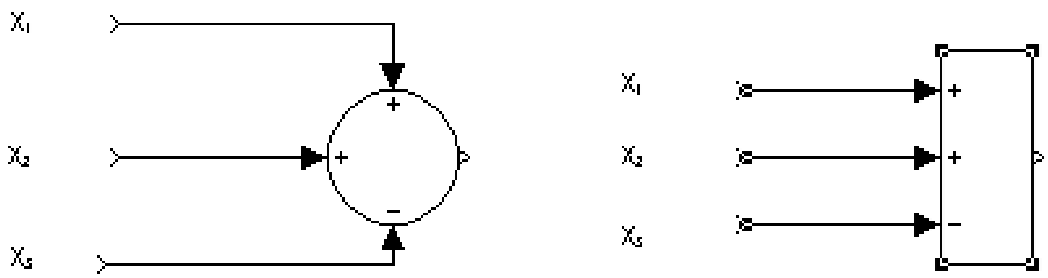

Figure 3: Two Simulink blocks for a summer representing $y = x_1 + x_2 - x_3$

GAIN (Math library)

A ***Gain*** block is shown by a triangular symbol, with the gain expression written inside if it will fit. If not, the symbol - K - is used. The value used in each gain block is established in a dialog box that appears if the user double-clicks on its block.

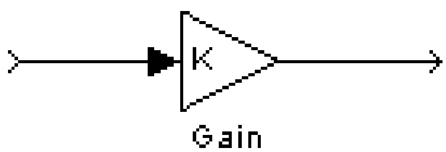

Figure 4: Simulink block for a gain of K.

CONSTANTS (Source library)

Constants are created by the ***Constant*** block, which closely resembles Figure 5. Double-clicking on the symbol opens a dialog box to establish the constant's value. It can be a number or an algebraic expression using constants whose values are defined in the workspace and are therefore known to MATLAB.

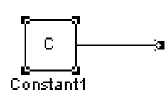

Figure 5: A constant block.

INTEGRATOR (Continuous library)

The block for an ***Integrator*** as shown below looks unusual. The quantity 1/s comes from the Laplace transform expression for integration. When double-clicked on the symbol for an integrator, a dialog box appears allowing the initial condition for that integrator to be specified. It may be implicit, and not shown on the block, as in Figure (a). Alternatively, a second input to the block can be displayed to supply the initial condition explicitly, as in part (b) of Figure 6. Initial conditions may be specific numerical values, literal variables, or algebraic expressions.

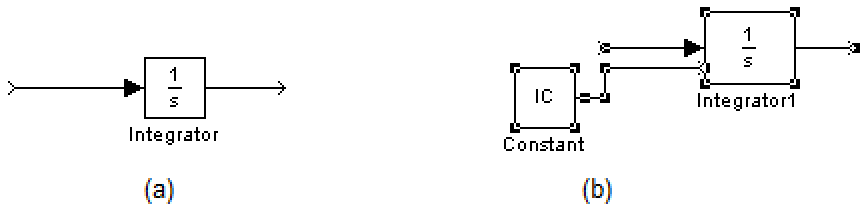

Figure 6: Two forms of the Simulink block for an integrator: (a) an implicit initial condition and (b) an explicit initial condition.

DIFFERENTIATOR (Continuous library)

The block for a ***Differentiator*** is shown in Figure 7. The Derivative block approximates the derivative of the input signal *u* with respect to the simulation time *t*. The result is an approximation of du/dt that is computed a numerical difference $\Delta u/\Delta t$, where Δu is the change in input value and Δt is the change in time since the previous simulation (major) time step. This block accepts one input and generates one output. The initial output for the block is zero.

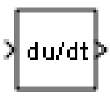

Figure 7: A differentiator block

The precise relationship between the input and output of this block is:

$$y(t) = \frac{\Delta u}{\Delta t} = \frac{u(t) - u(t_{previous})}{t - t_{previous}}$$

where t is the current simulation time and $t_{previous}$ is the time of the last output time of the simulation. The latter is the same as the time of the last major time step.

TRANSFER FUNCTION (Continuous library)

The ***Transfer Function*** block models a linear system by a transfer function of the Laplace-domain variable *s*. The block can model single-input single-output (SISO) and single-input multiple output (SIMO) systems.

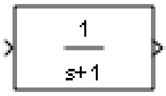

Figure 8: A transfer function block where the transfer function H(s) = $\frac{1}{S+1}$

STEP (Source library)

A Simulink block is provided for a ***Step*** input, a signal that changes (usually from zero) to a specified new, constant level at a specified time. These levels and time can be specified through the dialog box, obtained by double-clicking on the ***Step*** block.

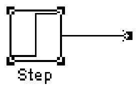

Figure 9: A step block

SIGNAL GENERATOR (Source library)

One source of repetitive signals in Simulink is called the ***Signal Generator***. Double-clicking on the ***Signal Generator*** block opens a dialog box, where a sine wave, a square wave, a ramp (sawtooth), or a random waveform can be chosen. In addition, the amplitude and frequency of the signal may be specified. The signals produced have a mean value of zero. The repetition frequency can be given in Hertz (Hz), which is the same as cycles per second, or in radians/second.

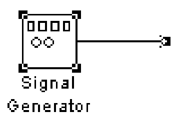

Figure 10: A signal generator block

SCOPE (Sinks library)

The system response can be examined graphically, as the simulation runs, using the ***Scope*** block in the sinks library. This name is derived from the electronic instrument, oscilloscope, which performs a similar function with electronic signals. Any of the variables in a Simulink diagram can be connected to the ***Scope*** block, and when the simulation is started, that variable is displayed. It is possible to include several Scope blocks. Also it is possible to display several signals in the same scope block using a MTJX block in the signals & systems library. The Scope normally chooses its scales automatically to best display the data.

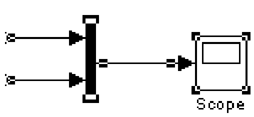

Figure 11: A scope block with MUX block

Two additional blocks will be needed if we wish to use MATLAB to plot the responses versus time. These are the ***Clock*** and the ***To Workspace*** blocks.

CLOCK (Sources library)

The clock produces the variable "time" that is associated with the integrators as MATLAB calculates a numerical (digital) solution to a model of a continuous system. The result is a string of sample values of each of the output variables. These samples are not necessarily at uniform time increments, so it is necessary to have the variable "time" that contains the time corresponding to each sample point. Then MATLAB can make plots versus "time." The clock output could be given any arbitrary name; we use "t" in most of the cases.

Figure 12: A clock block

To Workspace (Sinks library)

The ***To Workspace*** block is used to return the results of a simulation to the MATLAB workspace, where they can be analyzed and/or plotted. Any variable in a Simulink diagram can be connected to a ***ToWorkspace*** block. In our exercises, all of the state variables and the input variables are usually returned to the workspace. In addition, the result of any output equation that may be simulated would usually be sent to the workspace. In the block parameters drop down window, change the save format to 'array'.

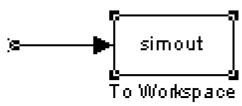

Figure 13: A To Workspace block

In the Simulink diagram, the appearance of a block can be changed by changing the foreground or background colours, or by drop shadow or other options available in the format drop down menu. The available options can be reached in the Simulink window by highlighting the block, then clicking the right mouse button. The ***Show Drop Shadow*** option is on the format drop-down menu.

Simulink provides scores of other blocks with different functions. Consider browsing the Simulink libraries and consult the online Help provided by MathWorks.

Compiled from:

- CISE 302 Linear Control Systems Laboratory Manual, Systems Engineering Department, King Fahd University of Petroleum & Minerals. http://www.kfupm.edu.sa/departments/se/SiteAssets/Lab%20Manuals/CISE-302-Linear-Control-Systems-Lab-Manual.pdf

- Controls Tutorials for MATLAB and Simulink, College of Engineering, University of Michagin http://ctms.engin.umich.edu/CTMS/index.php?aux=Home

- MathWorks website www.mathworks.com/