E102 MATLAB Simulink

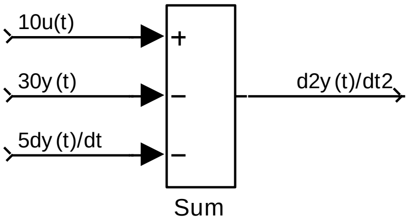
The simulation of any linear constant coefficient differential equation (LCCDE) can be represented by a block diagram. First rearrange the differential equation with the highest derivative on the left hand side of the equation. Represent this equation with a summing junction. For example, the equation

$$\frac{d^2y}{dt^2}+5\frac{dy}{dt}+30y=10u$$

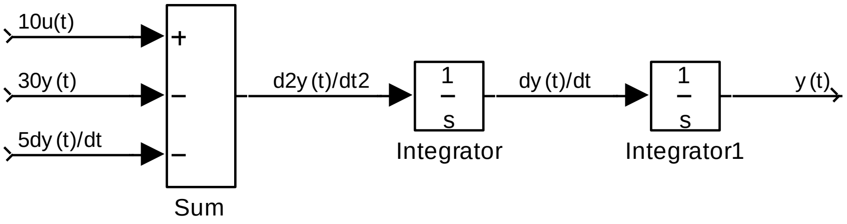
is rearranged to:

$$\frac{d^2y}{dt^2}=-5\frac{dy}{dt}-30y+10u$$

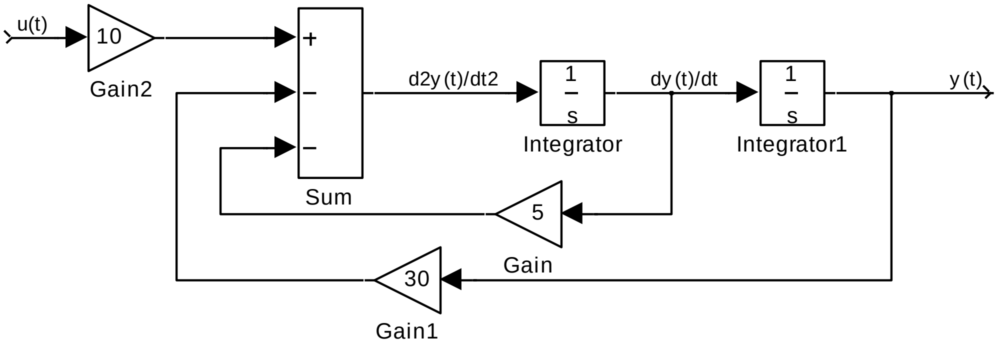
and represented by:



Next form a cascade of integrators to yield the lower order derivatives.



Finally, feedback the integrator outputs to the summing block and insert gain blocks



Simulink

Simulink, which runs under MATLAB allows you to simulate dynamic systems in the time domain using block diagrams.

Simulink is introduced by means of an example of simulating the step response of a causal LTI system described by a second order differential equation:

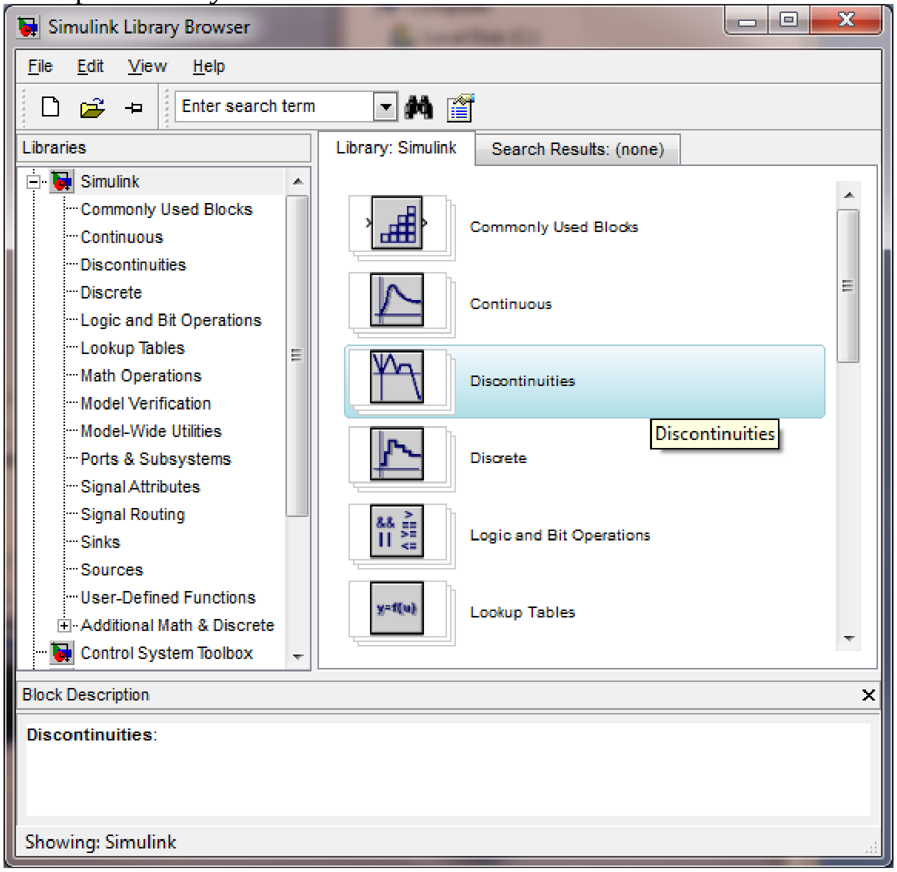
$$\frac{d^2y}{dt^2}+\frac{dy}{dt}+4y=8u$$

We shall work through the following steps for simulating the system:

A Build the block diagram
B Choose the simulation parameters
C Run the model
D Save the model

A Build the block diagram

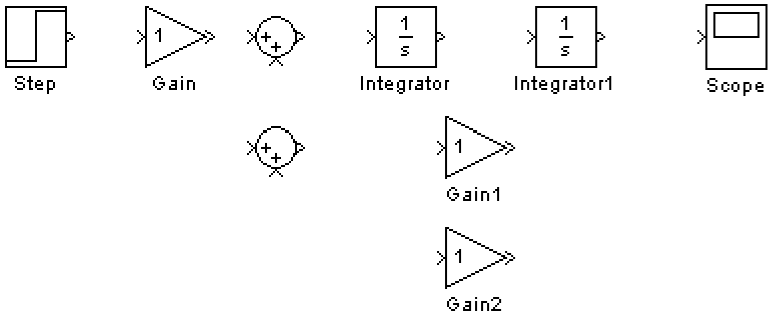
- Start MATLAB
- Click on the Simulink icon on the MATLAB toolbar or type `>>Simulink`
- Open *Blank Model* from the Start Page to open a model window
- Open *Library Browser* from the menu icons in the model window



The Library Browser displays a tree-structured view of the Simulink block libraries. You build models by copying blocks from the library into a model window.

- Move the model window so that you can see the Library Browser
- Select *Simulink => Commonly Used Blocks* from the Library Browser
- Drag the **Integrator** icon to the model window. When it opens it will show an Integrator block with input and output ports.
- Drag a second **Integrator** block to the model window.
- Drag 3 **Gain** blocks and 2 **Sum** blocks to the model window.
- Drag a **Scope** block to the model window.
- Select *Simulink => Sources*. Drag a **Step** block to the model window.

You now have all the blocks for the model. Arrange your blocks in the window to look like this:

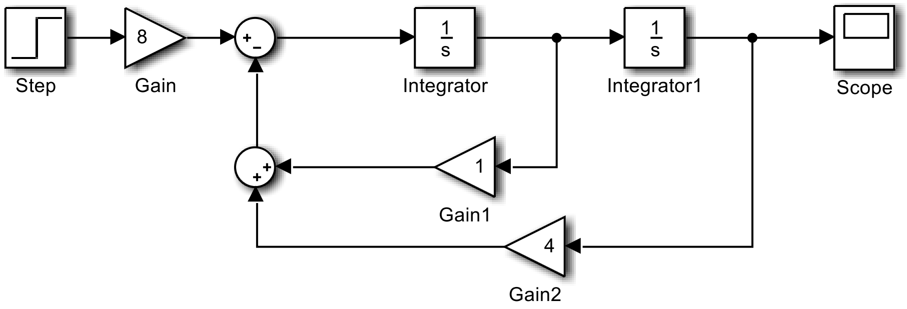


By default, signals flow through a block from left to right. You can change the orientation of a block by choosing the **Rotate and Flip** commands from the *Diagram* menu in the model window.

- Select each of the two **Gain** blocks Gain1 and Gain2 and flip them to point from right to left (*Diagram=>Rotate and Flip-=>Flip Block*)
- Select the lower **Sum** block so that the output port points up (*Diagram=>Rotate and Flip=>Counterclockwise*)
- Select the **Gain** blocks and change their values to 8, 1 and 4.
- Select the upper **Sum** block and change the signs to |+-
- Select the **Integrator** blocks and check that the initial conditions are set to 0
- Select the **Step** block and set the step time to 0

To connect the blocks together, click over the output port of the first block and drag the pointer to the input port of the second block. To branch from an existing connection line, click the right mouse button over the line and drag the pointer to the input port of the target block.

- Connect the blocks together to produce the block diagram shown below:



B Choose the simulation parameters

Simulink solves differential equations by putting them into finite difference form and stepping in time through the simulation until a stop time is reached. The algorithm for solving the equations can be chosen by the user as well as the step size and other parameters. You set the simulation parameters and select the solver by choosing **Model Configuration Parameters** from the **Simulation** menu in the model window. Simulink displays the Configuration Parameters dialog box, which uses several pages to manage the simulation. The two key pages are:

1. The **Solver** page, which allows you to set the start and stop times, to choose the solver and to specify numerical step times and solution tolerance limits.
2. The **Data Import/Export** page, which manages input from and output to the MATLAB workspace from which you started.

There are also **Diagnostics** pages which allow you to select the level of warning messages displayed if you make a mistake (which of course never happens). For this first run, you will accept all the default values that Simulink has chosen in its wisdom. So you may proceed directly to the next step.

C Run the model

- Open the **Scope** block.
- Select **Run** from the **Simulation** menu in the model window.

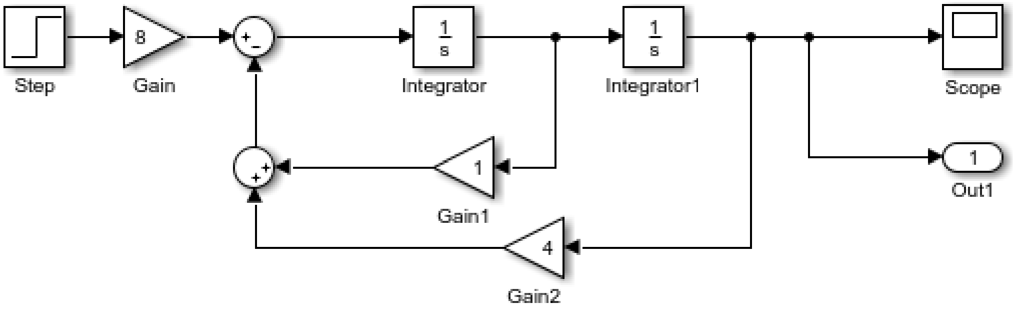
Your computer beeps to signal the completion of the simulation. The scope block should display the result …. a typical step response of an underdamped system.

D Save the model

- Save your model in your own directory as **example1.slx**. Simulink generates a specially formatted model file with the .slx extension that contains all the properties of the model.

The results of the simulation can also be made available as vectors of response values that you can access from the MATLAB command window.

- Add an output to your model from the Simulink Library Browser. *Library Browser => Sinks =>Out1*



- Select **Model Configuration Parameters** from the **Simulation** menu and choose **Data Import/Export**. Make sure that the time and output boxes in the "Save to Workspace" area are checked and the "Format" is set to Array.
- Run the simulation again.
- Return to the MATLAB command window and list all the current variables in the workspace:

```
>> whos
```

  You should find both a vector of time values **tout** and output values **yout** listed

```
 >> plot(tout,yout)
```

That completes the example. You should now be able to build and simulate your own block diagrams in Simulink. Before we finish however, here are some tips for running your Simulink models:

Increasing Resolution

You may want to look at a specific part of the response and increase the number of simulation points

- Choose *Model Configuration Parameters=>Solver=>Stop time* =>3
- Choose *Model Configuration Parameters=>Data Import/Export=>Additional Parameters=>Refine Factor* =>4

This gives you four times as many points over a smaller time range.

- Run the simulation again
- Return to the MATLAB workspace, plot the results and use the plot editor to find the peak overshoot (2.889)

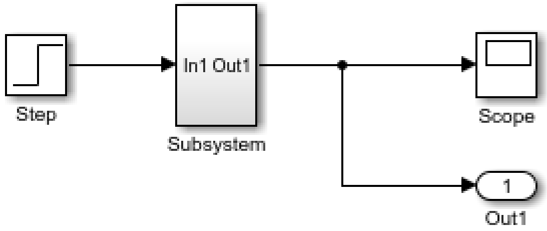
Creating Subsystems

As you add additional blocks for more complex systems, the model can become cluttered. You can simplify your block diagram and make it easier for yourself and others to read by grouping blocks into a new subsystem:

- Enclose all the blocks in **example1.slx** except for the **Step**, **Scope** and **Out1** blocks within a bounding box. (Click on empty space and drag to form a box)

- Choose *Diagram => Subsystem and Model Reference => Create Subsystem From Selection* from the menu.

After resizing and rearranging for aesthetic delight, your model should look like:



If you open the subsystem block, the underlying system is shown beneath. Simulink adds Inport and Outport blocks to represent the inputs and outputs to the subsystem.

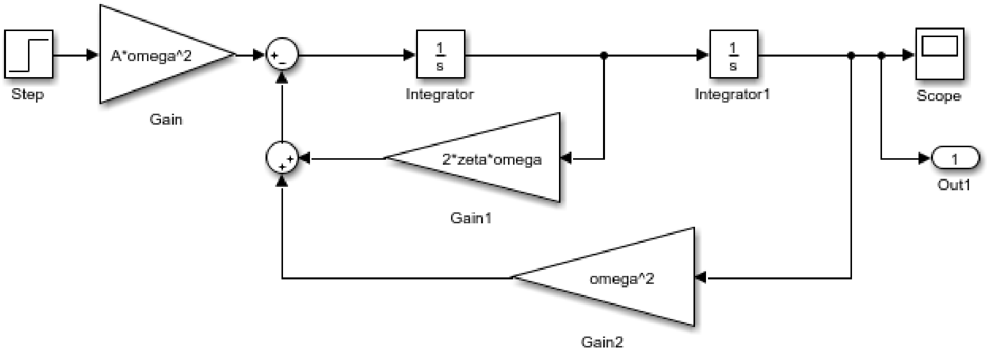
Running your SIMULINK model from an M-file

When you need to try a range of different parameters in your simulation models you can enter parameters from an M-file. For example to examine the step response of the canonical second order system:

$$\frac{d^2y(t)}{dt^2}+2\zeta\omega_n\frac{dy(t)}{dt}+\omega_n^2y(t)=A\omega_n^2x(t)$$

where ω_n is the undamped natural frequency, ζ is the damping ratio and A is the steady state gain.

Open **example1.slx** and change the constants in the gain blocks to variables:

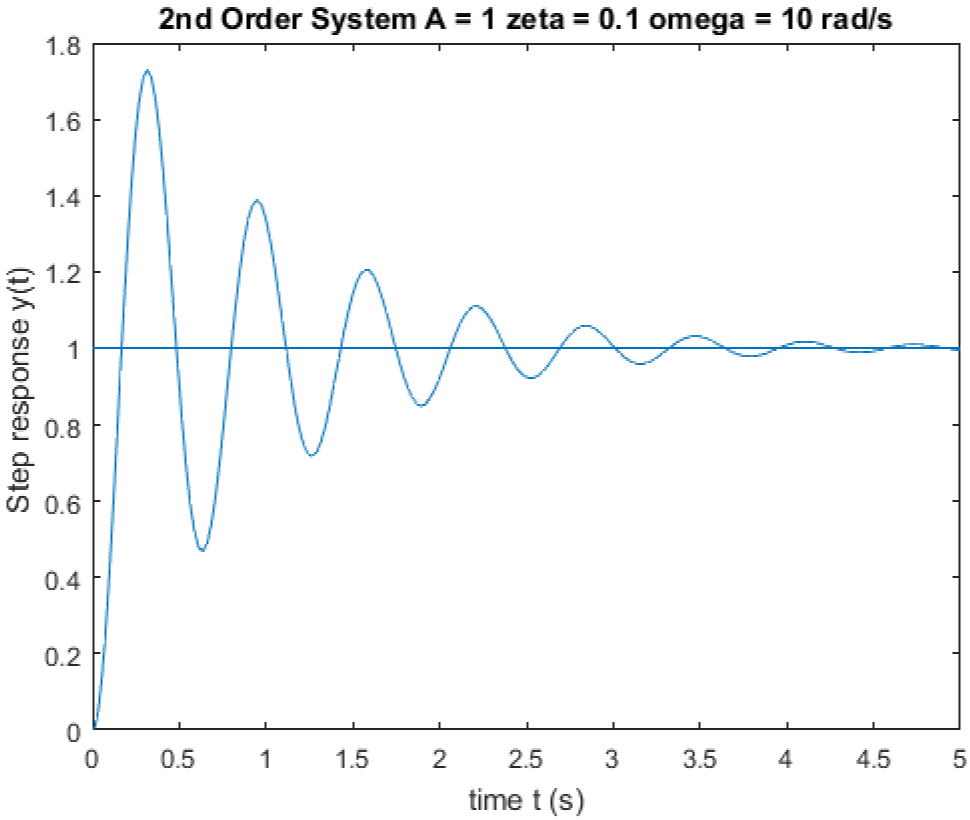


- Save as **canon2.slx**
- Copy, paste and run the following M-file (save as **canon2script.m**).

```
% canon2script.m
% plots the step response of a canonical second order system with
% natural frequency: omega
% damping ratio: zeta
% steady state gain: A

% set the parameters
omega=10;zeta=0.1;A=1;

% run the simulation
tstop=5;
[t,x,y]=sim('canon2',tstop);

% plot
plot(t,y)
str=sprintf('2nd Order System A = %g zeta = %g omega = %g rad/s', ...
         A,zeta,omega);
title(str)
ylabel ('Step response y(t) ')
xlabel ('time t (s)')
line([0 tstop],[1 1])
```

The M-file sets the parameters, runs the simulation and controls the subsequent data analysis and plotting. The MATLAB command **sim** runs the simulation model. By default, the time t, the state x and the output y are saved. The output y contains results from the output blocks in the model. The output x contains results from the integrator outputs in the model. For detailed syntax, type **help sim**.

Your output should look like:



This completes the MATLAB Simulink Tutorial