**ENEL 441 Control Systems**

**Lab 1**

**Winter 2020**

**Introduction to Simulink**

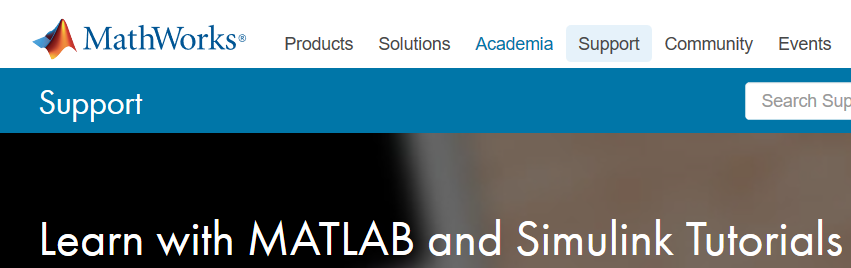
**Objective**

The primary objective of this lab is to introduce you to Simulink which is an analysis and simulation tool that runs under Matlab. Simulink is a graphical way of representing a general dynamic system. In our case we will be using Simulink for modelling LTI systems that can be described by interacting differential equations. It has effective tools for inputting such a system graphically and determining responses based on arbitrary input excitations. At the completion of this lab you should know how to:

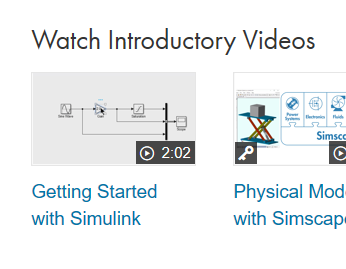
1. Open a new Simulink editor window and enter a model and save it  
2. Run the Simulink simulator for a continuous time model for a given simulation time and settings  
3. Porting data to the Matlab workspace for further analysis and plotting  
4. Copy and paste Simulink and Matlab models and responses into Word for documentation purposes

These are fundamental skills that you will be using for the ENEL441 labs which are all based on using the Matlab and Simulink tools for analysis of control systems.

If you have never used Simulink it is recommended that you look at some of the tutorial videos on the Mathworks.com site. You will see a webpage with the following menu. Open the support tab and get to the Simulink introduction video:

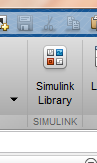


Scroll down the page and view this one:

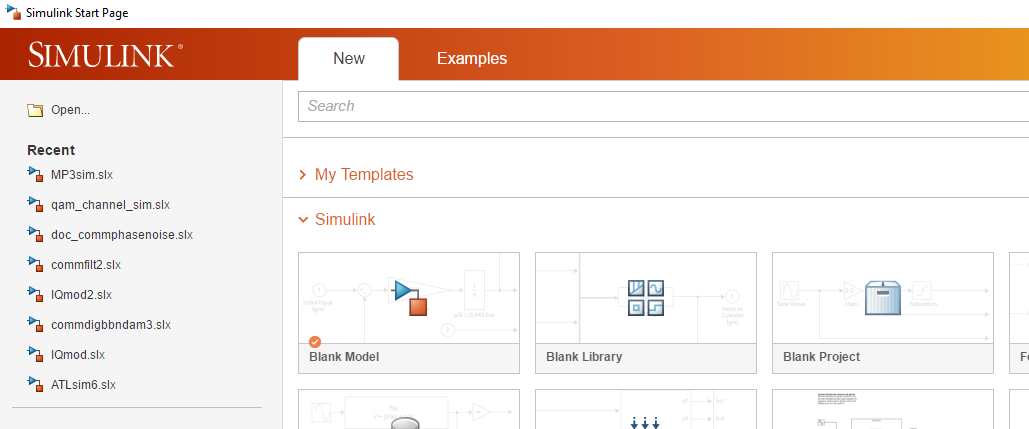


The Nise textbook has a tutorial on Simulink in the appendix at the end of the text that could be useful.

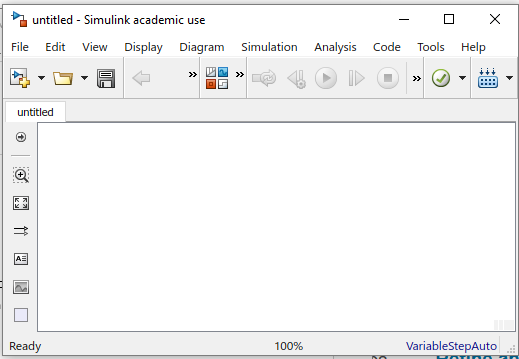
Next open up Matlab and then type “Simulink” in the command window then the Simulink GUI shows up with all of the available libraries. Or push the Simulink button on the main toolbar as shown below:



The initial page looks like this:

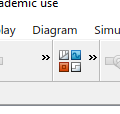


Select ‘Blank Model’ and a blank canvas will show up in which you can start adding functional blocks. You should get something like this:

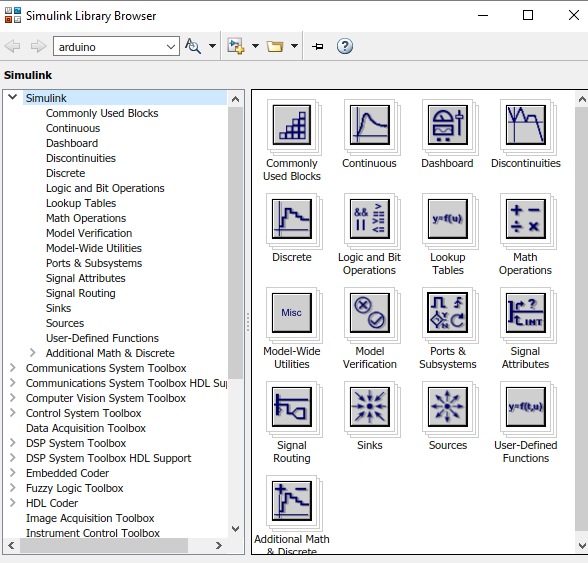


It could look a bit different depending on which version of Simulink you are running.

You should also open the ‘library browser’ by clicking the button



The library GUI looks a bit daunting but is very well organized and finding the blocks you need is straight forward.



On the left are all of the libraries of blocks and on the right are icons that you can open up and then select the specific models required. You just drag these around with the mouse and place them into your blank model window. Suggest you refer to Appendix C in your Nise textbook. This takes you through the initial steps of the Simulink library of blocks. Read sections C1, C2 and C3. Unfortunately, there are some changes in Simulink that are not documented in the text book. However the general work flow of building a model is there.

**Steps**

**A)** Open Simulink as instructed in the previous section and open a new Simulink model edit window (in the Simulink library browser window go to file->new->model). In the Simulink/sources directory find the ‘Sine Wave’ block and drag it into the model window. Then in the Simulink/sinks directory find the ‘Scope’ and ‘to workspace’ blocks. Connect these up as shown below:



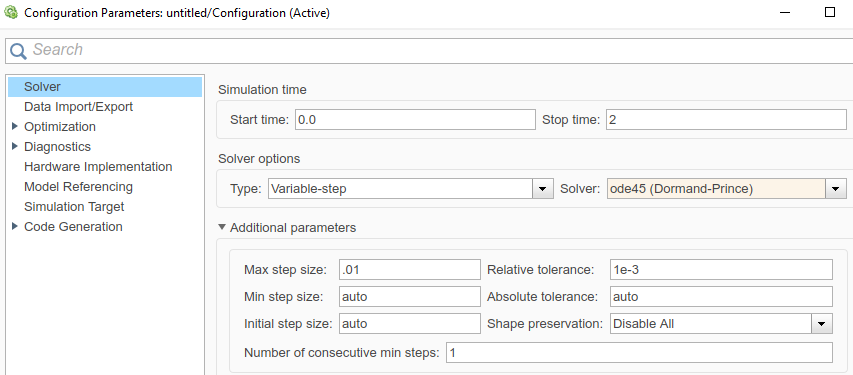
Open the sine generator dialog box (double click on the icon) and set:  
Time based, use simulation time, amplitude 1, frequency 10 (rad/sec)

In the ‘To Workspace block’, rename the variable name as ‘gen\_out’ . In the Save Format menu select ‘Array’.

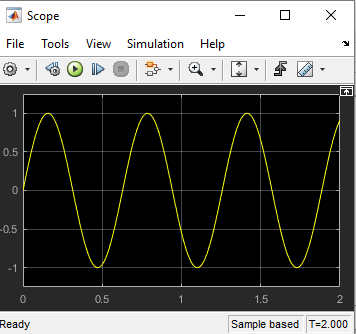
In the model window open the simulation tab-> model configuration parameters set the parameters:

Type: variable –step  
Max step size: .01 (sec)  
Solver: ode45  
start time 0 and stop time 2 (seconds)

The configuration screen will look like the following:

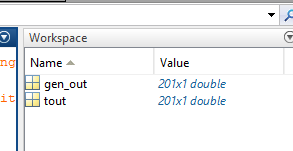


Close the simulation tab. Then save the model and then run the simulation and observe the scope output . Note you can scale time, response and zoom in on parts of the waveform. Make sure that the scope output corresponds to expectations based on the sine generator settings. It should look something like the following:



The exact output depends on the version of Matlab/Simulink you are running.

Note that the variables that were specified in the ‘To Workspace’ block appear as workspace variables after running the Simulink model



In the Matlab command space window and enter the command:

plot(tout,gen\_out)

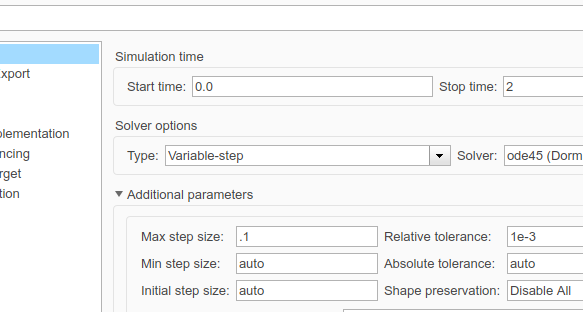
You can further annotate your graph with xlabel(), ylabel(), grid on, etc. You can type ‘help plot’ or ‘doc plot’ in the command window to get some help on the syntax details. The graph can be copied into this lab template by clicking the copy figure under the edit tap in the figure window. Alternately you can export it as an image. (edit->copy figure does not work on some Mac laptops )

**AQ1)** Plot the simulation output, label the axis, and then copy the graph image into your Word document for your lab report.

**(ans)**

**AQ2)** Try different settings of step size under the simulation->model configuration parameters tab. Describe what happens if the fixed step size simulation is used with a step size of 0.1 seconds.

Explain the results. (Simulink is based on representing continuous time domain solutions by a sequence of samples at a fixed time interval between samples. As this time interval becomes larger (lower sampling rate) then the signal represented in Simulink will appear to be more distorted). Determine the size of the array of gen\_out. Does it make sense?



**(ans)**

**B)** Use Simulink to determine the step response of a transfer function of

|  |  |
| --- | --- |
|  |  |

Use the *Pulse Generator* found in the Sources sub-library of the Simulink Library Browser as a source to replace the sinusoidal generator of part A. Before running the simulation use the variable step integration as shown

Also open the step block and set the ‘step time’ to 0. Your Simulink model should look something like the following.



Run the simulation from 0 to 2 seconds.

**BQ1)** Generate a matlab plot of the output signal using plot() in the command window. Also list the matlab commands used to do this.

**(ans)**

**BQ2)** Compare the Simulink result with the theoretical step response based on the inverse Laplace transform. You can use ilaplace() or residue() or just do the inverse calculation by hand. Superimpose the step response that Simulink generates and the theoretical step response based on using the Laplace transform. Write a short m-file in Matlab to do the plotting with a brief explanation.

(ans)

**C)** Next suppose that the step generator is not perfect but has a 3 dB bandwidth of 10 Hz. This bandwidth limitation is represented by a first order low pass filter with a DC gain of 1 and a -3 dB response at 10 Hz. That is it has a transfer function of



where  ,  and .

Use Simulink to plot the error by superimposing the step responses generated by the bandlimited square-wave generator and the infinite bandwidth square-wave generator.

**CQ1)** First create a model of the imperfect bandwidth limited step waveform generator and place it into a subsystem that is arbitrarily named ‘bandlimited step generator’. Subsystems are created by highlighting the functional boxes that you want to include and then right clicking and selecting ‘create subsystem from selection’. Note with the new version of Simulink you have to point to one of the wire connections when you right click.



The step generator subsystem should look something like the block below:



Verify that the transfer function does indeed represent a first order low pass filter with a 10 Hz 3 dB bandwidth. Determine the Simulink model that will do a comparison of the step response generated by the ideal generator (not bandlimited) and the bandlimited generator. Show the Simulink model and briefly describe how it works.

**(ans)**

**CQ2)** Produce a plot of the two simulated step responses superimposed on the same plot. Then in another figure, plot the difference of the two simulated step responses. Use Matlab for this graphing.

(ans)

**CQ3)** Determine the time domain response of the step response of the bandlimited pulse generator in part **CQ2** via the inverse Laplace transform with partial fraction expansion. Compare the theoretical with the Simulink output for the non-ideal bandlimited step function generator. That is evaluate the step response in time domain by taking the inverse laplace transform using residue() or by hand.

**(ans)**