

Lab 2 Pre-Lab

Running Simulink from a MATLAB m-file

Please complete the following tutorial before coming to the lab period this week. This prelab should take not more than 1 hour. The last page has what you will be required to bring to class.

Introduction

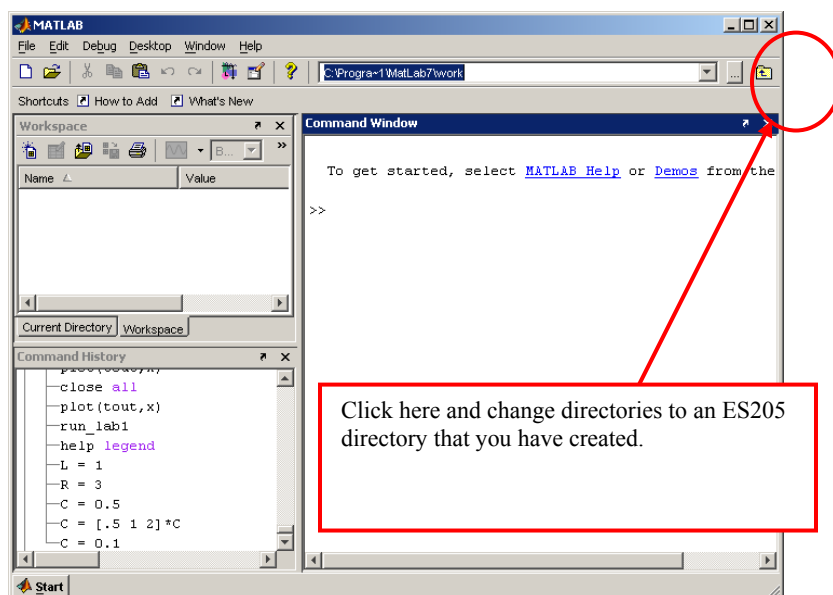
MATLAB (MATrix LABoratory) is a software package that operates as an interactive programming environment that can be used to run Simulink model files. While control of the simulation models can be done from within Simulink itself, sometimes, it may be more efficient to run the simulation from the MATLAB command window. The script file in MATLAB, called an *M-file*, can be used to more efficiently vary Simulink parameters, provide simulation results for further analysis/display and less frequently generate signals that are then used as inputs in a Simulink model.

This tutorial consists of 3 parts:

1. Creating a Simulink model to solve a first order ODE
2. Creating and using an M-file to run the Simulink model
3. Becoming familiar with plotting in MATLAB

Getting started

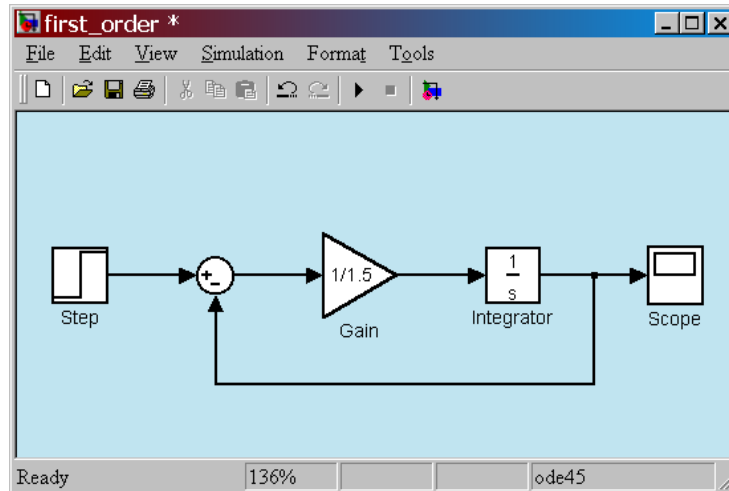
The first thing I would suggest doing is changing the directory to a folder you create for ES205. To do this just click on the three dots as shown below.



Set up a Simulink file to solve the ODE given by

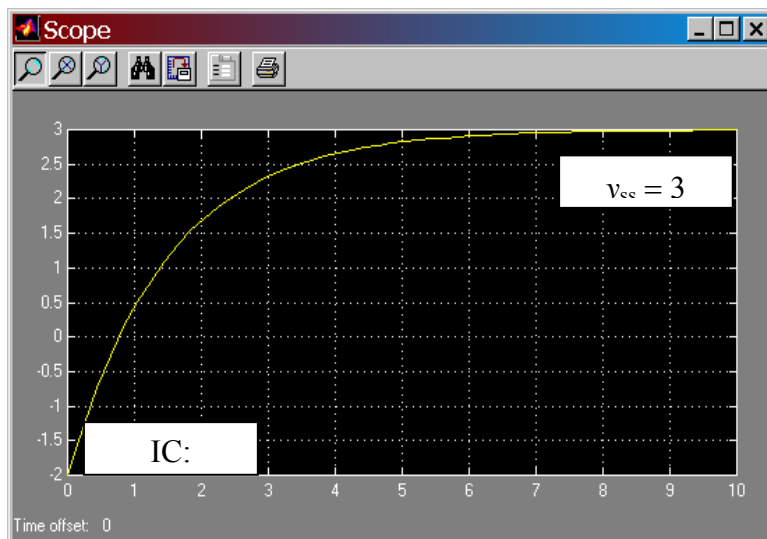
$$1.5\dot{y} + y = 3u,$$

where $y(0) = -2$ and $u(t)$ is a unit step input. Save the model under the filename **first_order.mdl**. Your simulation file should look like:



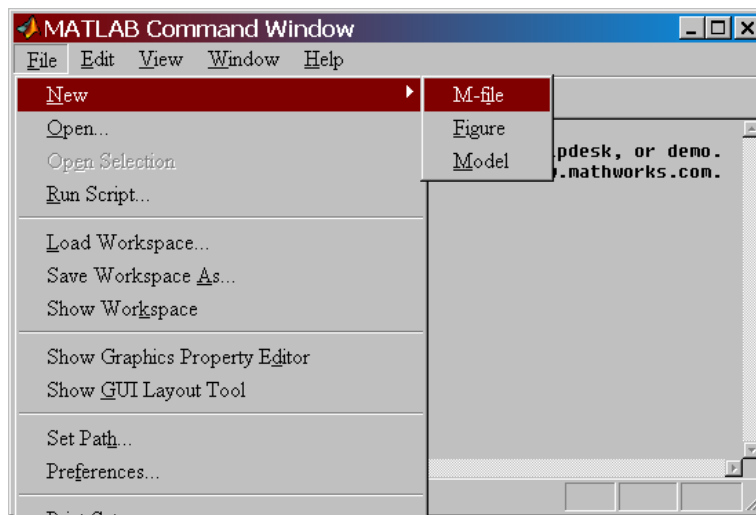
Every time you make a change to a MATLAB *M-file* or a Simulink *model file*, you have to **File** → **Save** before running the new simulation.

The solution of the ODE should look like:

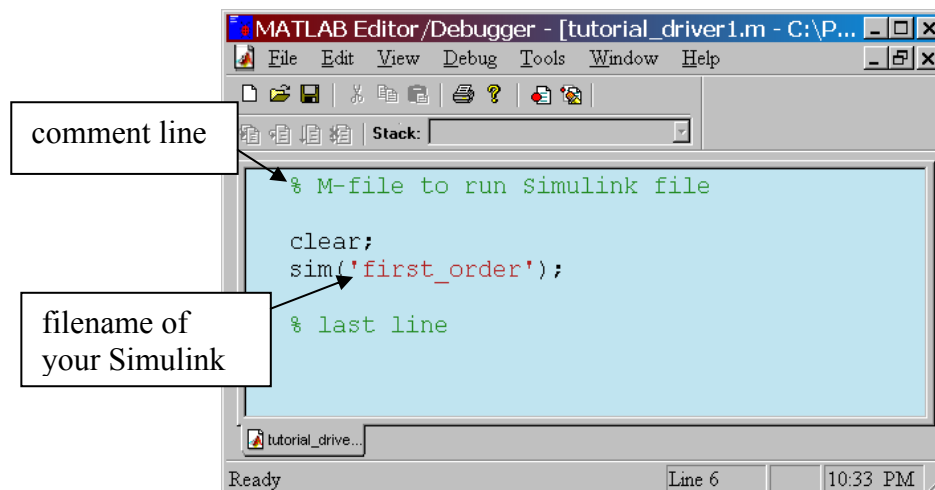


To run the simulation from MATLAB

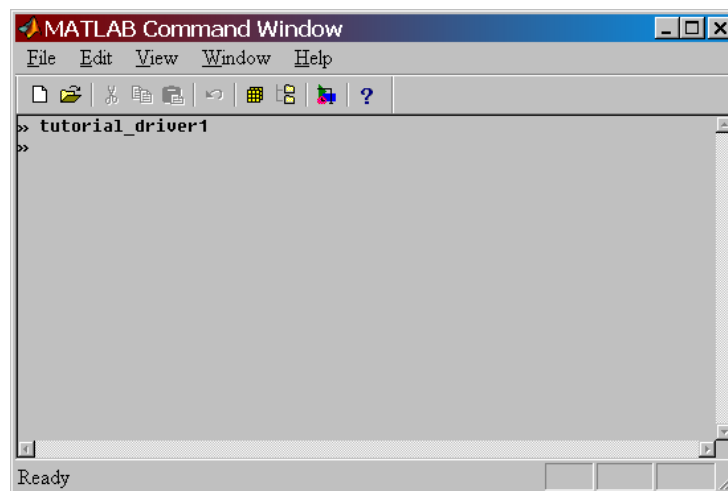
We need to create an M-file. In the MATLAB Command Window, select **File → New → M-file**.



Save the file that opens as **tutorial_driver1.m**. Type the following commands in the M-file. **File → Save**. This file is the MATLAB program you will use to control your simulation, change variables, and gain greater control over plotting results.



Run the program by typing its filename **tutorial_driver1** ↵ (return) in the MATLAB *Command Window* as shown below. Select the *Scope Window* to see the simulation results. Nothing will appear to happen, but the Simulink model will run unless you get an error message. **If you get an error message, read the next section, other wise skip to the next section.**



'Undefined function or variable' error message

If you type the filename in the MATLAB Command window, you might get an error message that says *Undefined function ...* (see below):

```
>> tutorial_driver1
??? Undefined function or variable 'tutorial_driver1'.
```

Or you might get an error regarding the use of the `sim` command.

First check that you spelled the filename correctly. If spelled correctly, you probably have to

- 1) Change directories so that the m-file and the Simulink model are in the same folder or
- 2) Use the *Set Path* command to tell MATLAB the directory to look in so it can find your file. Go to **File** → **Set Path ...**, then select **Add Folder** to find the directory in which you saved your M-file. Once you've selected the correct folder in the *Path Browser* window, click **OK**. Select **Save**, and close the *Path Browser*. Repeat for the directory in which your Simulink models are stored.

Return to the MATLAB Command window, type the filename of your M-file. Your file should run. To see results you need to go to your Simulink model and open the scope. We'll talk about plotting in MATLAB later in this tutorial.

To change variables from the m-file

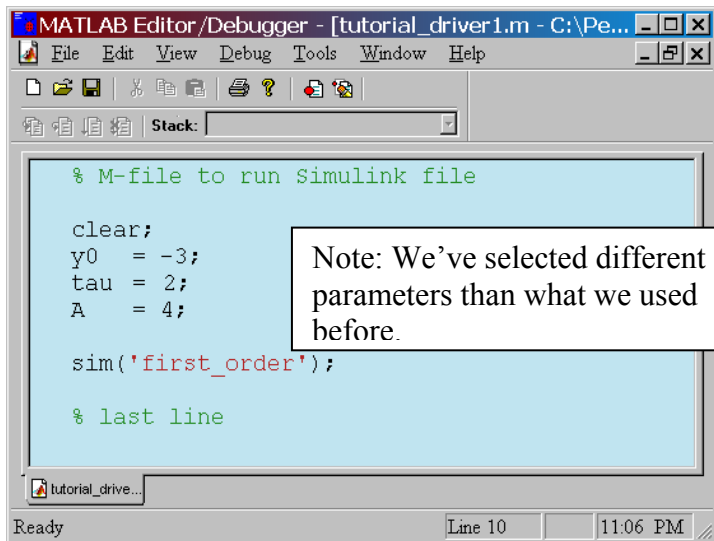
Put variable names in the Simulink simulation diagram. Replace the gain with $1/\tau$. Change the IC on the integrator to the variable name y_0 . Change the magnitude of the step input to A . Don't forget to **File** → **Save**.

The image shows a Simulink model window titled 'first_order'. The model consists of a Step block, a summing junction (+), a Gain block labeled '1/tau', an Integrator block labeled '1/s', and a Scope block. The Step block is connected to the summing junction, which is connected to the Gain block, which is connected to the Integrator block, which is connected to the Scope block. The status bar at the bottom of the window shows 'Ready', '136%', and 'ode45'.

Below the main window are two block parameter windows:

- Block Parameters: Step**: This window shows the parameters for the Step block. The 'Final value' field is circled in red and contains the variable name A .
- Block Parameters: Integrator**: This window shows the parameters for the Integrator block. The 'Initial condition' field is circled in red and contains the variable name y_0 .

Now, let's assign values to these variables in the M-file.



```
% M-file to run Simulink file

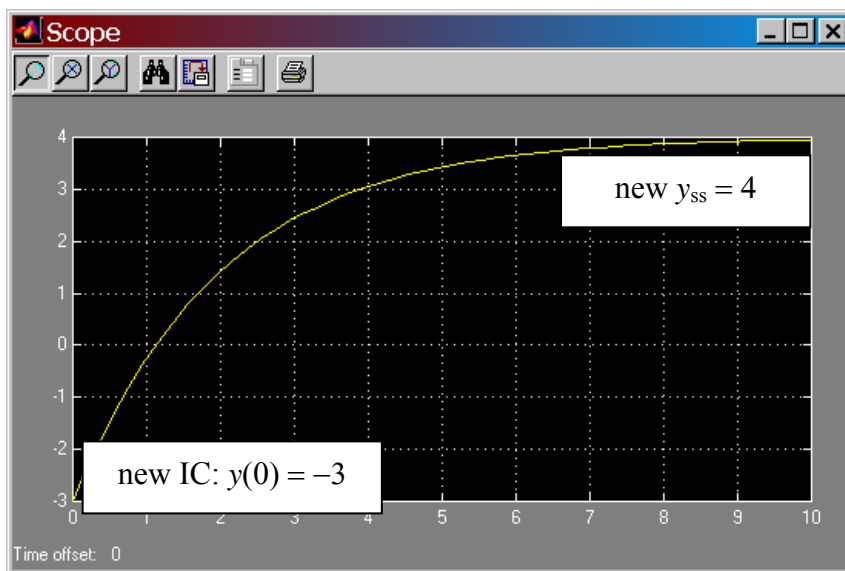
clear;
y0 = -3;
tau = 2;
A = 4;

sim('first_order');

% last line
```

Note: We've selected different parameters than what we used before.

File → Save. Again, run the program by typing `tutorial_driver1` in the MATLAB Command Window. Nothing will appear to happen. Go back into your Simulink model and open the Scope. It should show a new plot, with a new IC and a new final value.

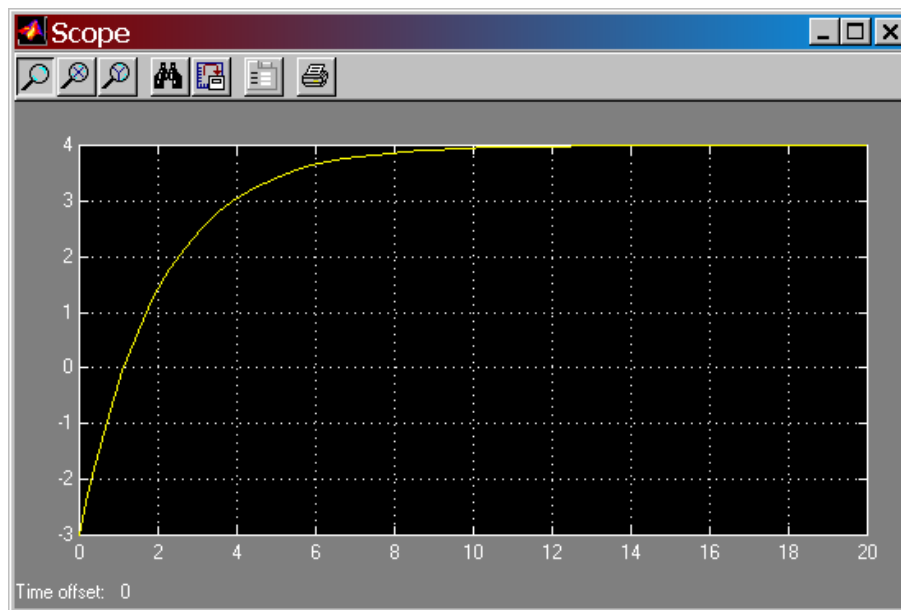


Changing the simulation time span

In the M-file, add a new argument to the **sim** command. The bracketed expression [0 20] tells Simulink to run the simulation for the time interval $0 \leq t \leq 20$. The M-file should now look like:

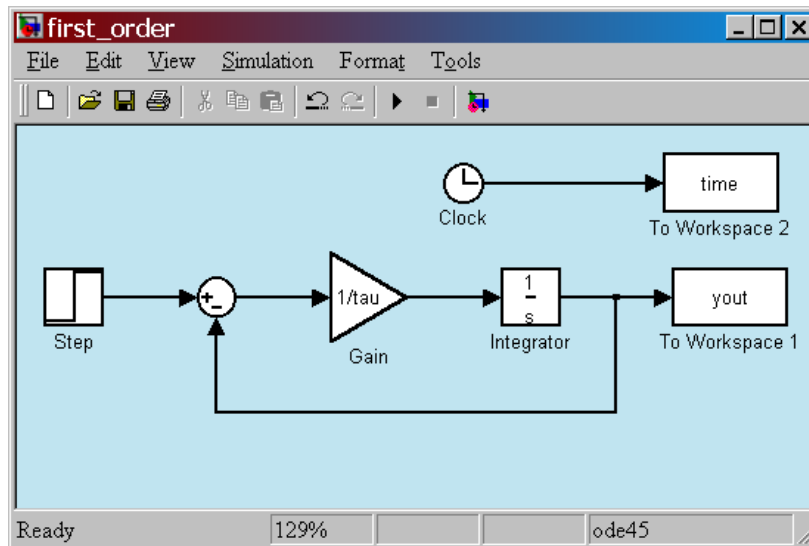
```
% M-file to run Simulink file
clear;
y0    = -3;
tau   = 2;
A     = 4;
sim('first_order',[0 20]);
% last line
```

The solution should look as follows. Note that time now runs from 0 to 20 sec.

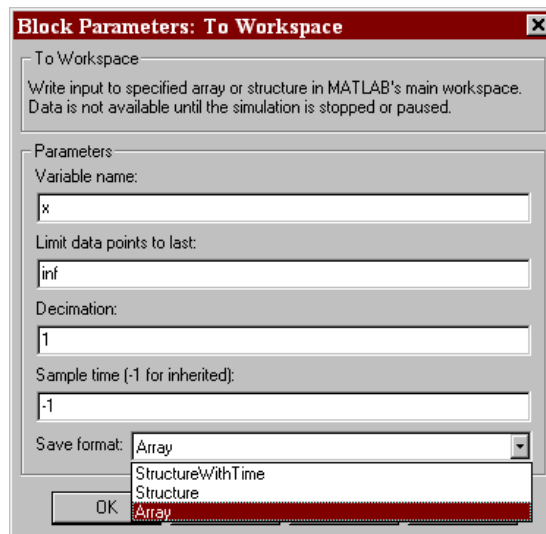


Plotting in MATLAB

To bring the variables from the Simulink workspace into the MATLAB workspace for better control of plotting, we have to assign variable names to the output variables. In the Simulink window, delete the *Scope* block and replace it with a *To Workspace* block from the *Sinks* library. In the *Block parameters* window, change the name of the variable name to **yout**. Add a *Clock* from the *Sources* menu connected to a second *To Workspace* block. Name this variable **time**. You have now created two new variables, **time** and **yout**, which are available for manipulation in the MATLAB environment.



You have to assign a “**Save format**” to these two output variables. In the *Block parameters* window of both *To Workspace* blocks, set the Save format selector to *Array* format. See the example below. (In MATLAB version 5, select the format called *Matrix*.)



Don't forget to **File** → **Save** the Simulink model.

Add a plot command to the M-file as follows.

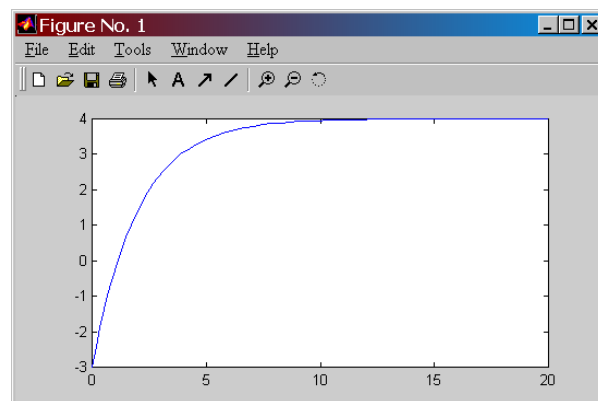
```
% M-file to run Simulink file
clear;
y0 = -3;
tau = 2;
A = 4;
sim('first_order',[0 20]);

plot(time, yout)

% last line
```

Don't forget to **File** → **Save** the M-file.

Again, run the program by typing **tutorial_driver1** ↵ in the MATLAB Command Window. The resulting plot is in the *Figure* window.



Add a title and label the axes by adding the following commands to the M-file. The plot is shown.

```
% M-file to run Simulink file

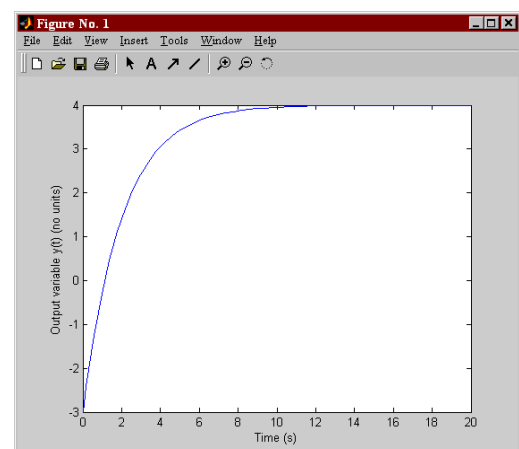
clear;

y0 = -3;
tau = 2;
A = 4;

sim('first_order',[0 20]);
plot(time,yout)

xlabel('Time (s)')
ylabel('Output variable y(t) (no units)')

% last line
```



Deliverable

Bring to lab your laptop with your file that solves this pre-lab activity (**tutorial_driver.m**). We will continue to work on this file during the lab period.