050915/ Thomas Munther
Halmstad University
School of Information Technology, Computer Science and Electrical Engineering

Simple exercises in Matlab/Simulink II

The goal with the exercises is to get started with easy problem solving in Simulink.
This is a graphical tool where most of the things also can be done in Matlab or its other toolboxes. But the need of remembering and writing all commands is not necessary.
It is rather simple to get started. Simulink has several application areas, such as control theory or signal processing.

Start Matlab and write **simulink** in the command window or click the colourful symbol !

This will probably open one or more windows. Depending on which version of matlab you are currently using .
One window is named **Simulink Library Browser.**
In this window you will find two columns. The left one has the sublibraries and the right shows a graphical symbol of what each library contains. Now we are going to build a simulink model file, approximately the same as a m-file but using symbols instead.
Open up a editor file to create simulink model (mdl-file).
Look for **File->New ->Model**. Have both these windows opened without any overlapping. Then enter **Simulink Library Browser ->Simulink->Sinks** and on the right symbol column. Mark the **Scope** symbol. Drag it to your open model window.
Also get a signal generator which can be found under **Simulink->Sources** choose **Pulse generator**. Drag it over to your model file.
Finally connect the two blocks by clicking the Pulse generator and using the left mousebutton. It should then look like the figure below.

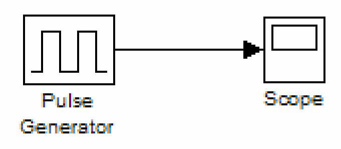

figure 1

Exercise 1:
Run the model by clicking the arrow in the tools menu.
Double click the **Scope** symbol in order to see a simulation.
It seems to bee a square wave with amplitude 1 and frequency 1 Hz.
This can bee changed by double click the **Pulse generator**.
Run the model once again . Does it seem correct ?
Maybe its hard to see. Use the three different zoom possibilities that can be found in the tools menu of the Scope. Zoom, zoom x-scale and zoom y-scale.

Also try to change pulse width to 2%. Run the model !

Exercise 2:
Mark the **Pulse generator** and delete it. Instead get a Signal Generator under **Simulink->Sources.** It shoult be connected as the previous example.
Here we have the possibility to choose different signals as square wave, sawtooth, sine wave or a random waveform.
Select a sine wave with amplitude 2 and frequency 1000 Hz. Run it !
It looks a bit strange does it not. I can see no resemblance with a sine wave what so ever. Every signal being simulated is using more or less the same number of points. Suppose we would like simulate a sine signal (1000 Hz). It would be sufficient to simulate this one during 0.002 sec. The simulation time can be altered under **Simulation->Simulation Parameters->Stop Time.** Change the **Stop Time** and start the simulation.
Look on the curve , now we have 2 periods (if nothing can be seen press the binocular).

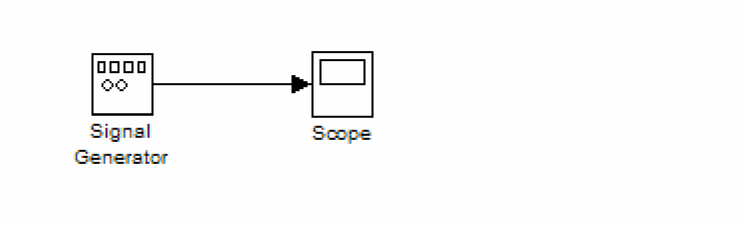

figure 2

Exercise 3:
Replace the Signal Generator above with a **Signal Builder** and try to construct your very own signal. Open the signal builder, click and drag with the mouse. Perhaps you can get something similar to the figure below.

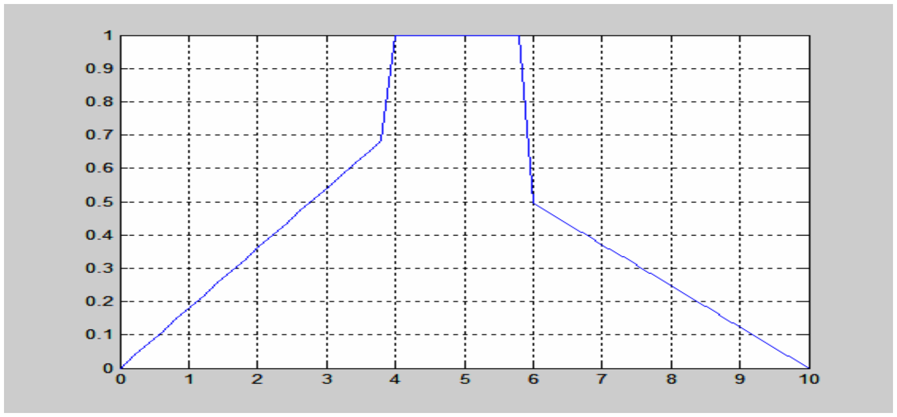

figure 3

Exercise 4:
We are now going to construct our own function. We get **Matlab Fcn** from **Simulink->User defined Functions.** Put it together with a **Scope** and use **Ramp** as an input signal. We can find it in **Simulink->Sources**

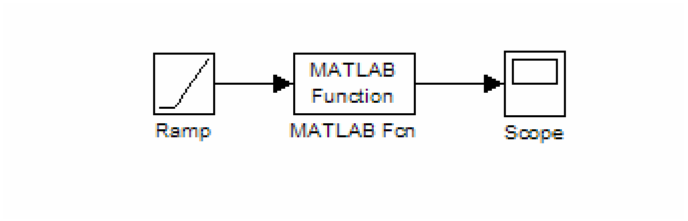

figure 4

Click on Ramp and in this symbol you can find three argument: slope, start time and initial output. Use the following settings, slope=1, start time=2 and initial output=2. Afterwards set the **Matlab Function** to a sine function. Just write **sin**.
Run the model !

Add more functions to the model above. Enter **Simulink->Math Operations** and drag the blocks **Abs** and **Gain** to the current model window. Double click the Gain and change the value to 3. Add also some extra Scopes in order to see precisely what happens in every step. Mark Scope and press the right mouse button and drag how many scopes you desire.

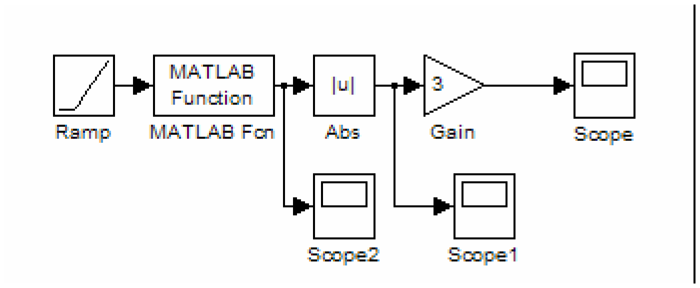

figure 5

What we have here is a full wave rectifier with an amplifier.
Run the model ! Does it work as expected ?
In scope 2 we can see a sine wave though with a time delay of 2 seconds.
Thereafter we take the absolute value of the sine and get a rectified signal as can be seen in scope 1. Finally we amplify the amplitude with gain 3 and get the output signal in scope.

Exercise 5:
Remove all blocks except **Ramp** and **Scope**.
Now we are going to take the time derivative of an input signal and to integrate the same signal. Get an integrator from **Simulink->Continuous** and in the same library we also find **Derivative**. If you double click on the Integrator you can set a parameter

initial condition. From the start it has a default value 0, but it can be altered to whatever you want. Integrate and take the time derivate for 10 seconds.

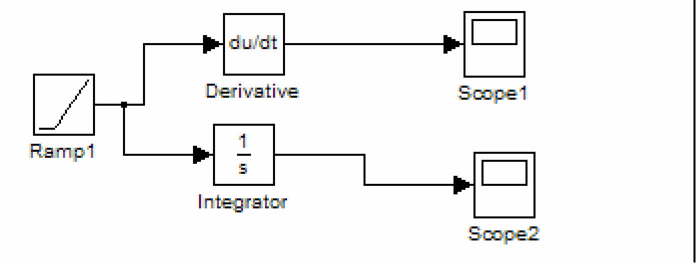

figure 6

Run the model file and see the answers in scopes.
Also change the input signal to a **Step**. Can be found under **Simulink->Sources**.

You are surely familiar with the step function. There are three parameters in the block: Initial value, Step time and Final value. Set the Step time=2 and the rest should be as it is. What value does the integration and the time derivative give if we simulate for 10 seconds ?

Sometimes you want the signals to appear in the same Scope. This can be arranged by choosing a multiplexer. Get a Mux from **Simulink->Signal Routing.** Delete one of the scopes. See figure 7 !

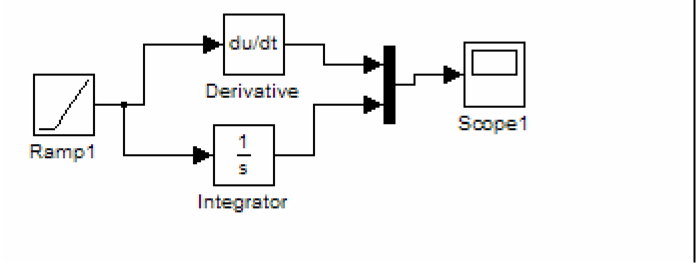

figure 7

Simulate once again !

Exercise 6:
Lets introduce the idea of subsystem. A subsystem makes it possible to hide details of a model and to have a overview of the model.
It can be found in: **Simulink->Ports&Subsystems**
Hide the time derivative and the integrator in the subsystem and switch **Step** to a **Sine wave** (**Simulink->Sources**).

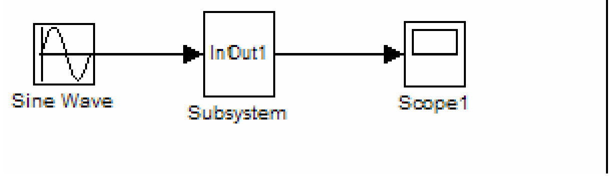

figure 8

Execute !
We can also export variables to matlab command window from our model.
The opposite is also possible.
But here and now we are going to export and therefore we look for **simout**
in **Simulink->Sinks** and **clock** from **Simulink-> Sources**.
In both imported blocks you should open them and select format array.
This means that they are saved in format of a vector. Change the block names to time and out respectively. Everything should look like according to figure 9.

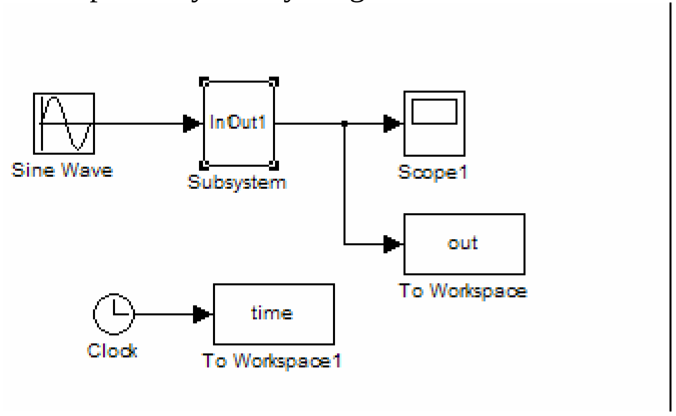

figure 9

Start simulation, now the variables time and out are accessable in the command window. Activate the command window and write:

plot(time,out), grid

It looks rather familiar.

Exercise 7:
In this example we are going to investigate a differential equation and look on its solution. We have a simple model of bacteria growth in a jam pot. Assume that the number of ”born ”bacteria is increasing proportional to the existing number (x) of bacteria and the number dying is proportional to the existing number in square.
$((dx)/(dt))=bx-px^2$
Birth growth= bx   death growth= px^2
The constants b =1/hour and p=0.5/(bacteria*hour).
We have 100 bacteria from the start.

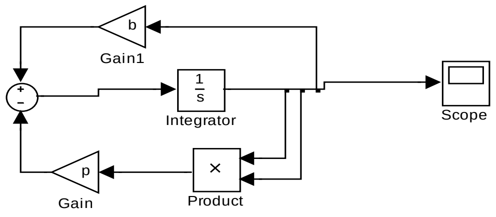

figure 10

The blocks **Gain, Product and Sum** can all be found in **Simulink-> Math Operations**.
Use the blocks above to show how the solution to the differential equation looks like.
Make a graphical reading from the **Scope**.
The initial number of bacteria is written in the integrator. The constants b and p can be assigned in the matlab command window or by just replacing the letters by their numerical value. Double click on the Gains and switch value.
After how long time will only half the initial bacteria remain ?

Exercise 8: In this example we will try to illustrate how a difference equation will look like: y[n]= -0.5*y[n-1]+x[n] and a initial condition y[-1]=1.

Further we know that our input signal is a step applied at n=0 with amplitude 1.

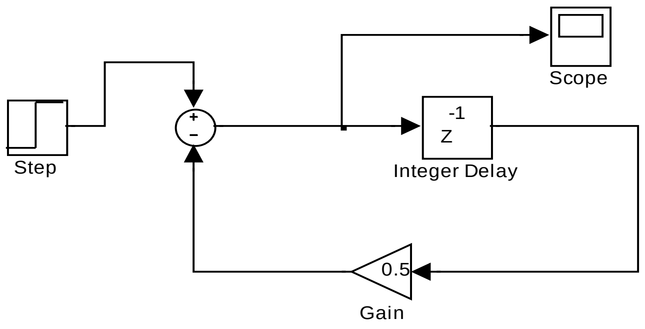

Figure 11

In the scope we can find the solution y[n]. The initial condition is inserted in the Delay block. I can also select the sampling time in seconds. This can be inserted in both the Delay and Step block. The output is shown below in figure 12. The Integer Delay can be found in library: **Simulink-> Discrete**
Before you run the simulation. Change in **Simulation-> Configuration Parameters**.
Choose the following: **Solver Options-> Fixed-step** and Solver-> Discrete

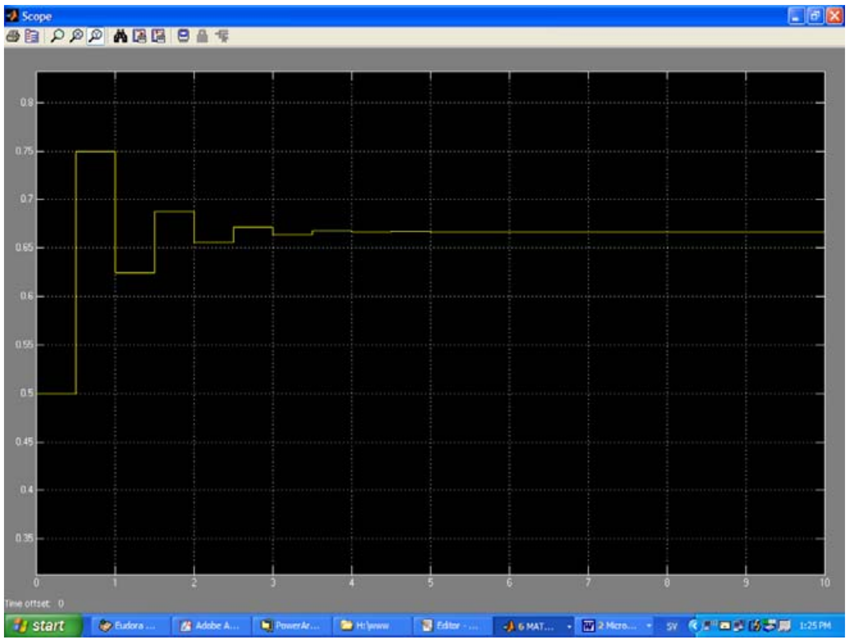

Figure 12

Please compare the output with your own calculations !

Exercise 9: We are going to look on some demo application models using simulink. This is for somewhat more advanced users. As you can see in these they are frequently using transfer functions. This is more or less the same as the frequency response for the system. There are numerous examples to be found and from different application areas. Simulink can easily deal both linear and unlinear expressions.
Study the following models.

penddemo
thermo

Run these by writing the name in the command window followed by enter.

Exercises to be solved

The exercises should be handed in as mdl-files and sent to my emailadress. These exercises can be handed in individually or as a group, but notice that the group must not be larger than 2. Can give bonus points on the written exam. They mdl-files should be zipped and sent to me as a package.

1. In this problem we will try to illustrate the following: a waggon is linked to a wall with a spring and a damper. The position of the waggon is given by x(t). The waggon can move without any friction on the ground. The mass is M.
The differential equation is the following: $M(d^2x(t))/(dt^2)+b(dx(t)/(dt)+cx(t)=F$

M= 5 [kg] mass of the waggon
b= 1 [Ns/m] damper constant
c= 2 [N/m] spring constant
F- force acting on the waggon, zero from the beginning and 2[N] after 5 [sec].
x(t)- position
dx(t)/dt-velocity

Assume velocity and position are zero from the start.
Decide the stationary position of the waggon after a long time by reading from the **Scope** in your model.

2. Determine the output of the system described by the following difference equation.
Input- x[n], output- y[n]
y[n]-0.75*y[n-1]+0.125*y[n-2]= 2*x[n], y[-1]=1, y[-2]=-1 and x[n]=2*u[n]

Give me a Simulink model from which I can easily obtain the solution !