An introduction to SIMULINK

Maarten Vergauwen

29 November 1999

Contents

1 Organization **2**
- 1.1 What . . . 2
- 1.2 Where . . . 2
- 1.3 When . . . 2
- 1.4 How . . . 2
- 1.5 Help ! . . . 2

2 An overview of Simulink **4**
- 2.1 What is Simulink ? . . . 4
- 2.2 Getting started . . . 4
- 2.3 The library . . . 4
- 2.4 The worksheet . . . 5
  - 2.4.1 Entering new blocks . . . 5
  - 2.4.2 Resizing and moving blocks . . . 5
  - 2.4.3 Copying, removing, rotating and flipping blocks . . . 5
  - 2.4.4 Connecting blocks . . . 5
  - 2.4.5 Grouping blocks . . . 6
  - 2.4.6 Saving and loading block diagrams . . . 6
  - 2.4.7 Simulation parameters . . . 6
  - 2.4.8 Simulating . . . 7
  - 2.4.9 The Scope block . . . 7
- 2.5 Exercises . . . 8
  - 2.5.1 Exercise 1: Sum of sines . . . 8
  - 2.5.2 Exercise 2: Feedback control . . . 9
  - 2.5.3 Exercise 3: Sampling . . . 9
  - 2.5.4 Exercise 4: Median filter . . . 10

Chapter 1

Organization

1.1 What

This exercise session is meant to give an introduction to Simulink, a software-tool that is needed throughout the education of an electrical engineer. The exercises are incorporated in a general design course which gives an overview of the different aspects of the design of an electrical/mechanical system and tries to point out the relations between these aspects. Simulink will be used later in the design course for simulating and testing control loops, behavior and stability of the systems, etc.

Exercise sessions of other courses will also use Simulink for various goals. It is therefore very important that you learn your way around the Simulink system and that you are proficient in using it. This text is meant as a tool to help you learn the basics of Simulink.

1.2 Where

The exercise sessions will be held on the HP workstations in room 91.56 in the basement of the ESAT building. To login on to the HP stations, please use your personal account.

1.3 When

For the correct date and time of the sessions, please consult the lists on the valvae. We urge you to come to the sessions at the correct moment, because the number of workstations is just enough to accommodate the number of students in each session.

1.4 How

There is a workstation available for every group of two students. You can choose your partner yourself. Although you will not be evaluated for these sessions, we urge you to take them very seriously because the following sessions are based on this introduction.

1.5 Help !

If you need help, ask the assistant during the exercise sessions or contact (during office hours):

Maarten Vergauwen	room 91.71
vergauwe@esat.kuleuven.ac.be	phone 1064

Chapter 2

An overview of Simulink

2.1 What is Simulink ?

Simulink, a companion program to Matlab, is an interactive system for simulating linear and nonlinear dynamic systems. It is a graphical mouse-driven program that allows you to model a system by drawing a block diagram on the screen and manipulating it dynamically. It can work with linear, nonlinear, continuous-time, discrete-time, multivariable, and multirate systems.
Getting acquainted with this toolbox and understanding how to use it to simulate easy and not-so-easy systems, are the two very important goals of these design sessions.

2.2 Getting started

- Before you start, close all applications (like Netscape) that allocate a lot of colors.
- Start a Matlab-session. Type `Simulink` at the Matlab prompt to start the Simulink toolbox.
- A window will appear on your screen: the **library** window.
- To start a worksheet select **New** → **Model** in the **File** menu or press <CONTROL>-n.

2.3 The library

- The default library, which is always opened when Simulink is started, consists of eight sub-libraries: **Sources, Sinks, Continuous, Discrete, Math, Functions and Tables, Nonlinear, Signals and Systems** and **Blocksets & Toolboxes**.
- By double-clicking on any sub-library, you open it and gain access to the blocks of that sub-library. For example, in the **Sources** sub-library, you can find often-used blocks like **Constant, Signal Generator, Pulse Generator** and **Clock**.
- It's also possible to load other (self-made) libraries. A special library was created with probably all the blocks you need in these design sessions. To open it, use **Open** in the **File** menu of the Library or Worksheet window. Open the library `sem838lib.mdl` in the directory `/users/visics/vergauwe/sem838`. If this directory is in your path, you can also start the library by typing `sem838lib` at the Matlab prompt.

2.4 The worksheet

2.4.1 Entering new blocks

- To put a block from the library into your worksheet, put the pointer on the block in the library window. Push the left mouse button and keep it down. Drag the block to your worksheet and release the mouse button.
- To change the parameters of a block, double-click on it with the left mouse button. A window pops up in which you can fill out the parameter fields. Click **OK** when ready. It is also possible to specify a variable-name instead of a value. The variable must be specified in your Matlab workspace before running the simulation. This can be very useful to try different initial conditions and parameters.
- If you're not sure how the block works or what all parameters do, click on the **Help** button. This will open a Netscape window with help for this specific block.

2.4.2 Resizing and moving blocks

- First select the block by clicking with the left mouse button. In every corner of the block a little filled square appears. Put the pointer on one of the corners , press the left mouse button and keep it down. Move the mouse and release the mouse button when the block has the desired size.
- To move a block, put the pointer inside it, click the left mouse button, hold it down, drag the block to its new position and release the mouse button.

2.4.3 Copying, removing, rotating and flipping blocks

- Position the pointer on the block to be copied. Click the right mouse button and keep it down. Move the mouse to the desired position of the copy of the block and release the mouse button. The new block will have the same properties as the original block. The name of the block is copied too but a "1" is added at the end.
- To remove a block, click on it once with the left mouse button to select it and press <CONTROL>-X or the <BACKSPACE> key. If this doesn't work, use **Cut** button in the **Edit** menu instead.
- To rotate a block, select it and press <CONTROL>-R. If this doesn't work, use the **Rotate Block** button in the **Format** menu.
- To flip a block (i.e. rotate it 180 degrees), select it and press <CONTROL>-F. If this doesn't work, use the **Flip Block** button in the **Format** menu.

2.4.4 Connecting blocks

- Put the mouse pointer on one of the outputs of the source block (indicated by a small triangle). Press the left mouse button and keep it down. Move the pointer towards the target block. A straight, dashed line will be drawn. If you release the mouse button an arrow will appear at the end of the line. From here you can start a new line. If you reach the input of the target block and release the mouse button, a bold arrow will appear to indicate that the connection is complete.

- For example: add a Sine-Wave (Sources library) and a Scope block Ssinks library) and connect them. The worksheet window should look like figure 2.1.

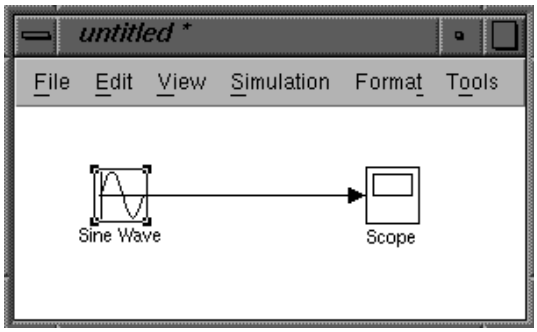

Figure 2.1: Sine wave + scope

- To branch from existing connections (for example to make multiple connections from one source block) use the right mouse button.

2.4.5 Grouping blocks

- You can group a set of blocks that belong together to make your worksheet more surveyable. Select the blocks you want to group. You can do so by pressing the left mouse button and keeping it down; then move the mouse: a rectangle will appear. When you release the mouse button all blocks inside the rectangle are selected. To group the selected blocks, press <CONTROL>-G or use **Create Subsystem** in the **Edit** menu. A subsystem is now created. If you want to see what's inside it, double-click on it. A new window will appear with the contents of the subsystem. Try this out on the system of figure 2.1.

2.4.6 Saving and loading block diagrams

- Use the **Save As** topic in the **File** menu to save your worksheet. A window will pop up. In the **Selection** information field you can specify the name of the file. Since the block diagram is saved as an "mdl" file, the extension `.mdl` will automatically be added if you do not specify it yourself. Click on **Done** to save your block diagram.

- From now on you can save this worksheet under the same name by using **Save** in the **File** menu. We urge you to do this very frequently.

- Use **Close** in the **File** menu to close the worksheet window.

- Use **Open** in the **File** menu to load a previously saved worksheet.

2.4.7 Simulation parameters

- A very important menu item is the **Parameters** item in the **Simulation** menu. If you click on it, a new window will appear in which you can specify all important simulation parameters.

- Among the parameters which you can alter are **Start-** and **Stop time**, **Maximum** and **Initial step size** and **Relative** and **Absolute tolerance**.
- Since setting the correct parameters is not an easy task, the correct values will be given for each exercise. If not all parameters are specified, it is assumed that the non-mentioned parameters take their default value, which is the value they get when Simulink is first started. You can return to the default values by clicking the **Revert** button in the **Simulation parameters** window.

2.4.8 Simulating

- Now we are ready to really start simulating a system. Let's take the example of the Sine-Wave connected to the Scope.
- Click twice on the Scope to open it.
- Set the stop-time of the simulation to 40 and the Max step size to 1e-2 in the **Simulation parameters** window.
- Start the simulation in the **Simulation** menu or click <CONTROL>-t.
- When the simulation is finished, you'll hear a beep.
- You can pause the simulation and later continue or stop it. Just click on the appropriate item in the **Simulation** menu.

2.4.9 The Scope block

- This block is very important and you will use it very often. It creates a plot of its input. If you want to plot two or more different signals in the same scope you have to combine the signals with a **Mux** block (Signal and Systems library) and connect the output of this block to the input of the scope.
- With the first three buttons on the left (showing a looking-glass) you can perform different zooming actions on the current plotted figure.
- The button with the binoculars "autoscales" the figure
- The fifth button on the left allows you to save the current axis settings.
- The second button from the right can be used to change some properties of the scope. The right button is for printing.
- If you right-click on the y-axis you can explicitly set its properties.

2.5 Exercises

2.5.1 Exercise 1: Sum of sines

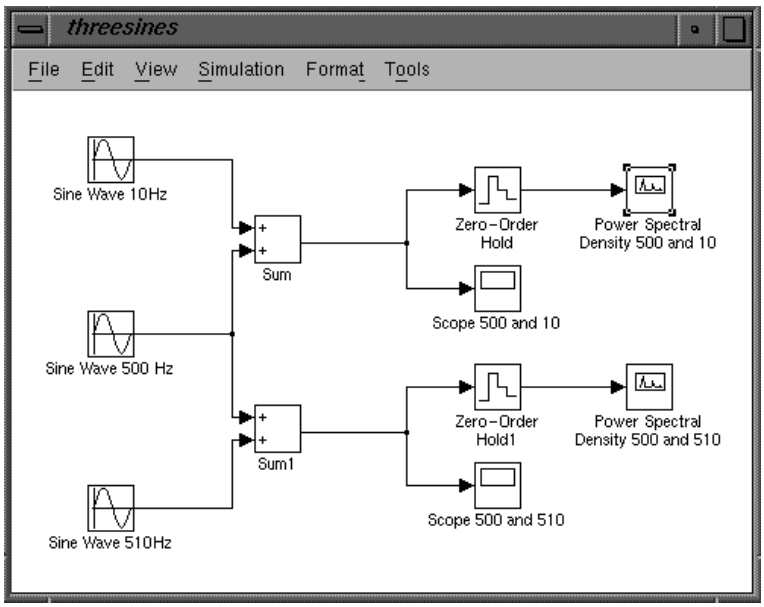

Figure 2.2: Three sines

- The first exercise in Simulink will demonstrate the basic commands and techniques of Simulink, as explained in the previous sections. This exercise will in fact produce the same outputs y and z as the exercise at the end of chapter 3 of the introduction to Matlab.
- Create the setup as shown in figure 2.2
- You see a two new blocks here:
  1. The **Zero-order hold** block. This is a block that performs sampling. Exercise 3 explains its function better. For now, make sure that the sampling time is 0.0005 sec.
  2. The **Power Spectral Density** block. This block performs the Fourier transform on the incoming digital data. Using the Fourier transform you can determine which frequencies are present in the incoming data. Set the parameters like this:
     - Length of buffer: 1024
     - Number of points for fft: 1024
     - Plot after how many points: 64
     - Sample time: 0.0005 (the same as the zero-order hold)
- To simulate correctly, you need to set the following parameters: The **Stop time** is 1e8 sec and the **Max step size** is 0.0001 sec. Select "Variable-step" and "discrete (no continuous states)" for the **Solver options**. The default values for the other parameters will suffice.

- Start the simulation. The PSD windows will open automatically.
- To view the results in the scopes, double-click on them and make the window that appears big enough. Set the time range to 1 sec.

2.5.2 Exercise 2: Feedback control

- The next system to simulate is the feedback system with transfer functions

$$G(s) = \frac{10}{s+10} \qquad \text{and} \qquad H(s) = \frac{s^2}{s^2+3s+2}$$

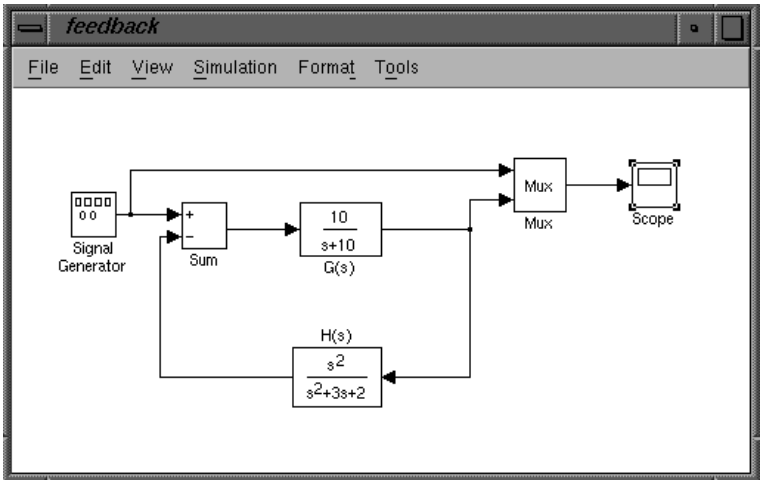

Figure 2.3: Feedback system

- Create the setup as shown in figure 2.3.
- Adjust the simulation parameters such that the **Max step size** is 0.001 sec and the **Stop Time** is 1e8 sec.
- Let the Signal Generator produce a sine-wave with an amplitude of 1 and a frequency of 1 rad/sec.
- Specify a time range of 10 sec in the scope.
- Simulate the system for different values of the frequency of the sine-wave, going from 1 to 100 rad/sec.

2.5.3 Exercise 3: Sampling

- To show the results of sampling with different sampling frequencies below and above the Nyquist frequency, this exercise only needs a simple setup as shown in figure 2.4.
- The Signal Generator should create a sine-wave with a frequency of 10 Hz. The **Max Step Size** in the simulation parameters window must be set to 0.001 sec and the **Stop Time** to 1e8 sec.

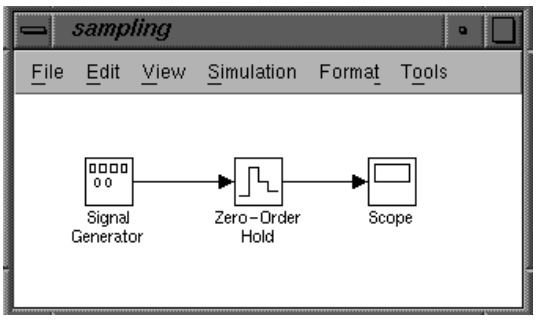

Figure 2.4: Sampling

- Specify a time range of 1 sec in the scope.
- Simulate the system with the following sampling frequencies (i.e. the frequency in the **Zero-Order Hold** block): 9 Hz, 10 Hz, 12 Hz, 20 Hz, 40 hz and 100 Hz. Can you draw conclusions ?

2.5.4 Exercise 4: Median filter

- As explained in exercise 6 of "An introduction to Matlab" the median filter is a non-linear filter that is often used in image-processing to smooth images while at the same time preserving the edges.
- We will implement the 1D-version with a masksize of 3.
- Create the setup you can see in figure 2.5.

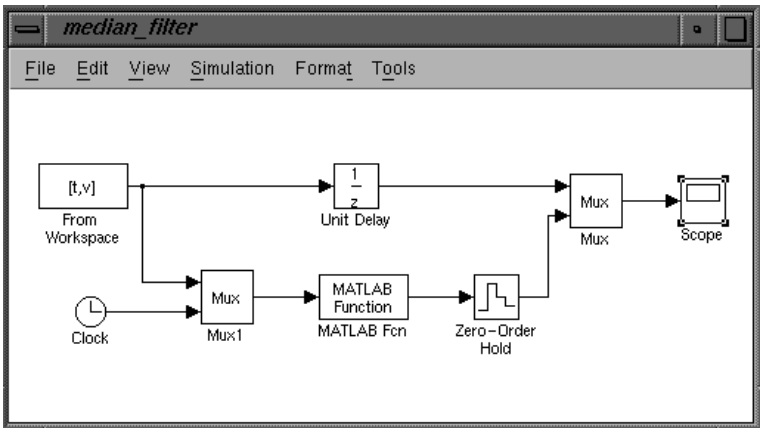

Figure 2.5: Median filter

- You recognize two blocks you already know: the scope and the Mux.
- Four other blocks are also used. These are:

1. The **Clock** block. This block outputs the current time. Set the decimation value to 1.
2. The **From Workspace** block. This block reads data from variables in the Matlab workspace.
   It needs two vectors: a time vector (t) and a value vector (v).
   Make sure the two vectors have the same length and are **column vectors**.
3. The **MATLAB function** block. This block executes the specified Matlab function every timestep. For information on how to write a Matlab-function we refer to the previous session.
4. The **Unit Delay** block. This block takes the input, holds it for one timestep and then sends it to its output. This block is needed here because the output of the median filter has a delay of (masksize-1)/2 timesteps. In our case the masksize is three so the output is delayed one timestep. If we want to compare the output of the filter to the input, we need to delay the input one timestep as well to synchronize the two.

- Load the file `/users/visics/vergauwe/sem838/medianvector.mat`. This loads the variables t and v in the workspace.
- Type `[t,v]` in the **Data** field of the **From Workspace** block. Set the sample time to 1 and switch off **Interpolate data** and **Hold final data value**.
- Set the **stop time** to 50 and the **maximum step size** to 1 in the **Simulation Parameters** window.
- Write a Matlab function which implements the median filter. You can find a template in `/users/visics/vergauwe/sem838/median_for_simulink.m`.
  Adjust the function so it works.
- Call the function with `median_for_simulink([3,u(1),u(2)])` in the **Matlab Function** block.



Figure 2.6: Result of the median filter

- Visualize the output. It should look like figure 2.6