SIMULINK FOR BEGINNERS*

Heikki Koivo (updated by Alpo Kuusisto)

*These notes are revised using MATLAB and SIMULINK version 2020b. Compared with earlier versions some modifications have occurred.

SIMULINK FOR BEGINNERS:

To begin your Simulink session, start by clicking MATLAB ICON button (Figure 1.1).



📣 MAT	MATLAB R2020b - academic use –														
но	ME	PLO	TS	APPS							ti ti	5 e 🗗 🕐	Search	n Docum	nentation 🛛 🔎 🜻
New Script	New Live Script	New File	Open	G Find Files	Import Data	Save Workspace VA	→ Open Variable → Open Variable ▼ → Clear Workspace ▼ RIABLE	Favorites	Analyze Code Code Clear Commands CODE	Simulink	Layout	Preferences Set Path ENVIRONMENT	Add-Ons	? Help	Community Request Support Learn MATLAB RESOURCES
Cu 🕤	🖸 🔼 🦨	or - C:\	Users\A	lpo\Desktop\sir	n_DC.m	p •						() × ()	Workspace		
	sir	n_DC.m	X	+									Name 🔺		Value
	2 - 3 -	c	lear lose	all all	and var	labies									

Figure 1. 1 MATLAB window is first opened. For a new Simulink session click the Simulink button.



Figure 1. 1 Select Blank Model and when it opens, the Library Browser

You will now see the whole *Simulink* block library as in Fig.1.2.

© Heikki Koivo (updated by Alpo Kuusisto)



Figure 1. 2 Simulink consists of block libraries Continuous, Dashboard, Discontinuities, Discrete, etc, as shown above. Note that there is a separate Simulink Extras, where you can find special blocks.

Browse through different block libraries. E.g., if you click Continuous, you will see the following:



Figure 1. 3 Blocks of *Continuous* block library.

Observe the description inside the Integrator block: $\frac{1}{s}$. This operator notation comes from linear

2

systems domain, where *s* represents Laplace variable. Roughly speaking, *s* corresponds to derivative operator $\frac{d}{dt}$, and its inverse $\frac{1}{s}$, to integration operation \int We will discuss the differences later.

Math Operations library is shown next



Figure 1. 4 Blocks in Math Operations block library (not all shown).

After browsing through other block libraries, we are now ready to start generating *Simulink* diagrams.

You can now construct a *Simulink* configuration of your system using different blocks in the library. You do not need to know how to solve e.g. a differential equation, *Simulink* will do it for you. After solving the problem the result is in numerical form, which can also be presented graphically. **EXAMPLE 1:** Given an input signal u(t) = sin(t).

It is fed into an amplifier with gain 2. Simulate the output of the amplifier y(t). (The answer is $y(t) = 2 \sin(t)$)

Solution: First choose the amplifier block. This you can find under *Math Operations* and it is called *Gain* (Figure 1.6). Move the cursor on top of *Gain*. Press the left hand side of your mouse down and drag the *Gain* block to the **untitled** file and release it. If you fail, try again. The result is shown below.



Figure 1.6 Gain block in Math Operations block library. The Gain has only one parameter, default value of 1. Change gain to 2.

SI	MULAT	NON	DEBUG	MODELING	FORM	AT AP	PS	BLOCK					¢
New		Open ▼ Save ▼ Print ▼	Library Browser	Log Signals	Add Viewer	 ✓ Stop Time Normal Imp Fast Res 	10.0 • tart	G Step Back ▼	► Run ▼	Step Forward	Stop		D
del Browser	e e	FILE	untitled	PREP	Gain Elemer	k Parameters: Gain ht-wise gain (y =	י K.*u) or ma	atrix gain (y	y = K*u	or y = u*K)		×	EVIE
Mo	€. 53	(1) Quick	ly insert block	s by double clickir	n <mark>g in</mark> Main Gain:	Signal Attribut	tes Parar	meter Attrik	outes				
	⇒				2							:	
	A II			Gair	Multipli	cation: Element	-wise(K.*u)					•	
					0		<u>О</u> К	<u>C</u> ance	el 📃	<u>H</u> elp	<u>A</u> ppl	y	

Figure 1. 7 Changing the gain value in the Gain block from 1 to 2. The gain value can be changed by double-clicking the Gain block. Once finished, click OK. If you do not, the value will not be changed. The numerical value inside the Gain block will now become 2. Next find input signals. Input block library is called *Sources*, where you can find the needed input: *Sine Wave*. Move the cursor on top of the *Sine Wave* block, drag the block to your model (Figure 1. 9).



Figure 1. 9 Sine Wave block is taken from the Sources block library.

To see the result, you need to install a *Sink* from *Sinks* library. In the beginning, the easiest *Sink* device is *Scope*. Move the *Scope* block to your model in the same way as before (The result is shown below Figure 1. 10).

*	untitle	d * - Simulink	academic use	e								_		Х
SII	MULAT	NON	DEBUG	MODELING	FORMAT	APPS	SCOPE				- • ¢	20. ▼	? •	•
New		Open 👻 Save 👻 Print 👻	Library Browser	Log Signals	Add Viewer	Stop Time 10.0 Normal • Fast Restart	Step Back V	Run	Step Forward	Stop	Data Inspector	•		_
G	4		untitled	PREPA	RE		SIMULATE	-			REVIEW RE	JULIS	2005	
Brows	۲	🎦 untitled											•	roperty
Model	Ð	Quickly	/ insert block	s by double clickin	ig in the diagram	and typing part of the	block name.	More inf	ormation.				×	Inspec
	エゴ													tor
	⇒				6									
	AΞ			A⇒		, <u>,</u>								
	~					Scope								

Figure 1. 10 Scope block is taken from the Sources block library.

The only thing missing from the system is to connect the subsystems together. Take the cursor to the output of the *Sine Wave* block. Cursor will change to a hairline. Now press left mouse button and drag your mouse close to the input of the Gain block. You will see a dashed black line forming from the

output side of the block which you started. Once you reach the input of the other block, a solid black line is formed to indicate that connection is made. If the connection is not complete, you will see a red dotted line (Figure 1.11). To complete the connection, you can e.g. drag the receiving block so that its input will meet the arrowhead.



Figure 1. 11 Connection from sine wave to gain is good. The connection from gain to scope is not complete (red dashed line)

After connections your configuration of the system is complete. Before simulation, you should check that the parameter values in the *Sine Wave* block are correct. Open it by double-clicking the block.

1	untitle	d * - Simulini	cacademic use		🔁 Block Parameters: Sine Wave	×		_		\times
SI	MULAT	NON	DEBUG	MODELING	Sine Wave	^	9 0	₽, -	? -	•
		Open 🔻		<u>_</u>	Output a sine wave:					
New	, 8	Save 🔻	Library	Log	O(t) = Amp*Sin(Freq*t+Phase) + Bias		Data			
-		Print 🔻 FILE	Browser LIBRARY	PREPARE	Sine type determines the computational technique used. The parameters in the two types are related through:		REVIEW RE	SULTS		Ā
Ser	$\langle \neg \rangle$		untitled							Pa
Brov	۲	🎦 untitled			Samples per period = 2*pi / (Frequency * Sample time)				•	r Perty
Model	Q	Quickl	y insert block	s by double clicking	Number of offset samples = Phase * Samples per period / (2*pi)				-	Inspec
: {	K Z				Use the sample-based sine type if numerical problems due to running for large times (e.g. overflow in absolute time) occur.					tor
,	⇒				Parameters					
1	A≣			Δ_{-}						
-	\sim			Sine Wave	Sine type: Time based					
1				One wave	Time (t): Use simulation time •					
1					Amplitude:					
					1					
					Bias:					
					0					
1					Frequency (rad/sec):					
					1					
					Phase (rad):					
					0					
2						~				
					OK Cancel Help Appl	y				

Figure 1. 12 *Block Parameters* of the *Sine Wave (last parameter: sample time, not shown).* For this example *Amplitude* is 1, *Frequency* (rad/s) is also 1, *Phase* and *Bias* are 0, so default values are OK. Sampling time is not an issue here.

Begin simulation by clicking the **Run** button (Figure 1.13).



Figure 1. 13 Starting simulation with the *Run* button. The simulation result is shown in the scope. The amplitude should be equal to 2 as seen in the Figure.

The solution is seen in the *Scope*. If the *Scope* does not open automatically, double-click it. Observe from the *x*-axis that the default simulation time is 10 s. If the figure is not scaled properly, use the different scale buttons in the Menu above the figure.

An important thing in learning to use SIMULINK is to actively play and experiment!

Example 2 – First order differential equation: Solve numerically the differential equation with SIMULINK. Think also what the exact solution is.

$$\frac{dx}{dt} = -2x + 1, t > 0$$
$$x(0) = 0$$

Solution 1: Consider the integrator block in Continuous block library. The input to the block is $\frac{dx}{dt}$. The integrator integrates it from t = 0s up to t. Default value is t = 10 s.

$$\int_0^t \frac{dx}{dt} dt = x(t) - x(0)$$

What is shown is only the input $\frac{dx}{dt}$ and output x(t). The initial condition is given in the *Block* parameters (Figure 1.14).



Figure 1. 14 Operation of the integrator block: Input is $\frac{dx}{dt}$ and output x(t). Initial condition x(0) is given by double-clicking the block. Default value is x(0)=0.

Continuing our configuration we observe that in Figure 1.14 we already have the left hand side of equation (1.1), which is $\frac{dx}{dt}$. We still need to generate the right hand side of the equation -2x+1 and equate it with $\frac{dx}{dt}$.

The integration in Figure 1.14 produces x(t). This needs to be multiplied by 2 (or -2), minus sign can also be added later to produce -2x. What is needed is the gain block from *Math operations* block library. Finally, a constant from the *Sources* block library is picked. For summing we need a summing block, which is found from *Math operations* block library. All the mentioned blocks are shown in Figure 1.15.



Figure 1. 15 Blocks needed to configure system (1.1).

Fastest way to add blocks is instead of Library Browser to double-click an empty area of your model and type first letters of the needed block, then select it from a list

Activate *Gain* block by clicking once and apply *Flip block*, which is found by right-clicking the block or under *Format* menu. Open the *Gain* block to change the gain value to 2. Remember to click OK. With the *Sum* block, click it open and change one sign to a minus sign. Again, remember to click OK.

S	IMULAT	ION	DEBUG	;	MODELI	NG	FORMAT	APPS		BLOCK		6	1
∛	Format Screen	t Painter Ishot 🔻		Arial		•	皆 Block Paramet	ers: Sum				· · ×	
	Show M	Markup	Auto Arrange	В		• 8, DAD	Sum Add or subtract i	inputs. Specif	fy one (of the following:		^	
Ser			untitled		TON	d ran	a) character vec between ports (tor containing e.g. ++ - ++)) + or -	for each input p	ort, for spacer		
Brow	۲	🎦 untitle	ed				b) scalar, >= 1, When there is or	specifies the i nly one input p	numbei port, ac	d or subtract ele	o be summed. ements over all		
Mode	Ð	Qui	ickly insert	blocks	by double	clickin	dimensions or or	ne specified di	imensio	n			
	K 7						Main Signal	Attributes					
	⇒			<u></u>	1	٦	Icon shape: rou	nd				•	ł
	AE			1	S	- 1	List of signs:						1
	\sim						+-						1
			1	}	\mathbf{x}							~	
					Sum		0		ОК	Cancel	Help	Apply	

Figure 1. 16 Block parameters of Sum. Change list of signs to +-.

Default value for *Integrator*'s *Initial condition* x(0) is 0, so we do not need to change it.

The final configuration is given in Figure 1.17. Scope has been added to observe the simulation result.

s	IMULAT	ION	DEBUG	MODELING	FORMAT	APPS	
 <th>Format Screen Show M</th><th>: Painter shot ▼ Markup</th><th>Auto Arrange</th><th></th><th>• A* A*</th><th>Background Foreground Shadow</th><th></th>	Format Screen Show M	: Painter shot ▼ Markup	Auto Arrange		• A* A*	Background Foreground Shadow	
ser			untitled		GRAPH	SITLE	
Brow	۲	🎦 untitle	ed				
Mode	Đ,	Qui	ickly insert	blocks by double clicking	in the diagram and	typing part of the blo	ck name. More information.
	K N K N						
	⇒						
	ΑΞ						
				2			

Figure 1.17 The final SIMULINK configuration of system (1.1).

Start the simulation with *Run* button. The result is shown in Figure 1.18.



Figure 1. 18 The simulation result on scope. (after double-clicking the scope block)

The result is a typical exponential response of a forced first order linear system. You can connect several signals to the same scope as shown in Figure 1.19.



Figure 1. 19 Both input and output signals in the same Scope

Solution 2 (Transfer function):

Another way to configure system (1.1) is to use a differential operator D and derive a transfer function for the system. Let $D = \frac{d}{dt}$ and denote the input as u(t)=1. Then the differential equation becomes

$$Dx = -2x + u(t)$$

or

$$Dx + 2x = (D+2)x = u$$

Solving for *x* leads to

$$x = \frac{1}{D+2} \quad u$$

 $output \ transfer f \ input$

The configuration is now shown in Figure 1.20, where *D* has been replaced by *s*. The coefficients in the denominator polynomial are changed from default values [1 1] to [1 2]

SI	MULAT	ION		DEBUG	MODELING	FOI	📔 Block Parameters: Transfer Fcn	×
New		Open Save Print ILE	• • •	Library Browser	Log Signals	Add Viewer PARE	Transfer Fcn The numerator coefficient can be a vector or matrix expression. The denominator coefficient must be a vector. The output width equals the number of rows in the numerator coefficient. You should specify the	
rowser	(i - 🔁	itled	untitled		_	coefficients in descending order of powers of s.	-
Model B	Đ,	(i) c	Quickly	insert bloc	ks by double click	king in the d	Numerator coefficients:	
	<u>кя</u>			л Г			Denominator coefficients:	
	A II		1	} 1	<u>s+2</u>		Absolute tolerance:	
							State Name: (e.g., 'position')	Ţ
							OK Cancel Help Apply	/

Figure 1. 20 System (1.1) in block diagram form.

The Block Parameters in Transfer Function are [1 1] and have to be changed: Denominator multipliers are [1 2]. This system configuration is equivalent to that of Figure 1.17,

except here the initial condition is equal to zero.

Step input: The input in system (1.1) is 1 after $t \ge 0$. A function, that first has a constant value say 0, and then at some later time instant changes to another constant value, is called a *Step function u*(*t*). It is defined as

$$u(t) = \begin{cases} 1, \ t > 0\\ 0, \ t < 0 \end{cases}$$

Here the change happens at t = 0 s. Note that it does not matter what value the function assumes at t = 0 s.

Step function can be found from the **Sources** block library as *Step* block. Note however that the default stepping time is not t=0 s but t=1 s. This can easily be changed as seen in Figure 1.21.

a Si	MULAT	ION		DEBUG	MODEL	ING		ADDC
New		Open Save Print	• • •	Library Browser	Log Signals	Ada View	Step Outp	p put a step.
2	F	ILE		LIBRARY		PREPARE	Main	in Signal Attributes
vser		P 1		untitled			Step t	o time:
Brov	۲	🍋 unti	tled					
Mode	Đ,	(1) Q	uickly	/ insert bloc	ks by double	clicking in t	ne (Initial	al value:
3	K N N						0	
	⇒						Final	ıl value:
1	AE	Γ	Г	>	$\frac{1}{r+2}$		1	
Ł	\sim	Ŀ		L	3+2		Samp	ple time:
							0	
1							🗹 Int	nterpret vector parameters as 1-D
							🗹 En	nable zero-crossing detection
							<	>
							0	OK Cancel Help Apply

Figure 1.21 Changing Block Parameters in Step Function, especially the Step Time.

Instead of using *Constant* block, it is common to use *Step* block as shown in Figure 1.21. The simulation result is exactly the same as in Fig. 1.17, if the change of Step time is made.

MATLAB Command window

Once you have defined your system in the SIMULINK window, you can simulate it also on the MATLAB Command window. For Matlab MATLAB version 2019a or newer, the variables from Simulink will be exported differently to the workspace of MATLAB than in the older versions. Change the Simulink settings so that the provided scripts will work for newer MATLAB versions.



Then make an output for Matlab by adding *'to Workspace'* block. It will be named first **'simout'**. Double-click it to change name to a more meaningful one, like **'acceleration'** or **'amperes'**, depending on your simulation. In this example we will just use the default **'simout'**



Save your model – it has first the name untitled, which may be used here.

Go to the MATLAB command window and type *help sim*. This will output a quick manual of simcommand. It is useful as a reference when you know the basics of using the sim-command. *Help plot* is another often useful reference.

Run the simulation by typing:
>> sim('untitled')

The only thing needed here is the name of the system you want to simulate. Next you can use plot command to see the result in x (time) - y (data) graph by typing:

```
>> plot(simout.time, simout.data)
```

Or plot (acceleration.time, acceleration.data) etc. depending on the name you chose in Simulink for the output.



Figure 1. 23 Simulation result is plotted with *plot* command.

The default time for simulation is 10 s. If you wish to simulate longer or shorter duration you can change Simulink window's *Simulation stop time*



Figure 1. 24 Simulation stop time can be changed from 10 seconds to a larger or smaller nonnegative value.

Complete the exercise. Run the simulation from SIMULINK and also from Command window.

Example 1.3 Damped oscillator

Solve the damped oscillator problem

$$\frac{d^2x}{dt^2} + 5\frac{dx}{dt} + 9x = 0$$

 $\frac{dx}{dt} = -2$

$$x(0) = 2$$

Assume that there is no input, that is, input u(t) = 0.

PURPOSE: To illustrate how to configure a SIMULINK diagram for a higher order differential equation and how to introduce initial conditions into it.

SOLUTION: Solve equation first with respect to the highest order derivative to obtain



To set up the left-hand side two integrators are needed:

The input to the first integrator is the second derivative $\frac{d^2x}{dt^2}$

and its output is $\frac{dx}{dt}$. The latter is the input to the second integrator producing x(t) at its output. In this

way we have constructed the left-hand side of the equation. Since the second derivative $\frac{d^2x}{dt^2}$

is equal to the right hand side, we collect it term by term. In order to do that we need $\frac{dx}{dt}$ from the

output of the first integrator, and x(t) from the output of the second integrator. Then x(t) must be multiplied by 9, so a *Gain* block is required. All the items are to be summed up so a *Sum* block is also needed. The final configuration is given below. The initial values are added to the integrators. The resulting configuration is given in Figure 1.25.



Figure 1. 25 SIMULINK configuration of system (1.7) Note that one way to take information to MATLAB command window is using Outport block (See Figure 1.26 about details).

Finally, set up the initial conditions by clicking the integrators one at a time and make the changes according to (1.8)

S	IMULATION	DEBUG	MODELING	FORMAT	APPS	🔁 Block Parameters: Integrator	×	
Nev	v 🛱 Open 👻	Library	Log	Add -	Stop Time 10.0 Normal	Integrator Continuous-time integration of the input signal.		
Ser	FILE	Browser LIBRARY untitled	PREPA	Viewer	🕬 Fast Restart	External reset: none		
Model Brow		d				Initial condition:]	
			Integrator			Limit output Wrap state		
						Show saturation port Show state port Absolute tolerance:		
						OK Cancel Help App	ly	

Simulink can output results to Matlab using 'To Workspace' block or 'Outport'. To Workspace is better suited for Matlab output. Outport's main usage is when dividing a large Simulink model to subsystems and exchanging data between subsystems and parent systems.

To see variable names *Outport* sends to *Matlab* open Simulation => Prepare => Model Settings. Select *Data Import/Export*, which is second on the list. By default Time *tout* and Output *yout* are saved to workspace and can be used in MATLAB command window (Figure 1.26).

SIMU	ILATION	DEBUG	MODELING	FORMAT	APPS	BLOCK		4	Configuration Parameters: untitle	d/Configuration (Active)			_	
FILE	LIBRAF	Y PREPARE	Stop Time 10.0 Normal Fast Restart	Step Back V	Run •	Step Forward	Stop F	R	Q Search Solver Data Import/Export	Load from workspace			Connect In	out
vser	-	Search					■ == ×		Math and Data Types Diagnostics	Initial state: xInitial			- oonneer my	put
odel Brov	•	SIGNAL	MONITORING	_	_		<u> </u>		Hardware Implementation Model Referencing	Save to workspace or file				
ź		Lo Sign Normal Visib INPUTS Conr	Add als Viewer Mode liity & PARAMETER TUI ect Signal	Signal Table	View Mana	ers Co ger L	onfigure .ogging		Simulation Target	 Time: States: Output: Final states: Signal logging: Data stores: Log Dataset data to file: 	tout xout yout xFinal logsout dsmout out.mat	Format: Da	taset al operating p Signals to Log	▼ oint g
		CONFIG Mor Setti	JRATION & SIMUL/ JRATION & SIMUL/ iel ngs Reference Model Settin	Arion Arion Property Inspector Inspector Inspector Inspector	S Upda Mod	s ate lel ution param	eters			Simulation Data Inspector Record logged workspar Additional parameters	: out	Logging inte	ervals: [-inf, in	11]
											ОК	Cancel	Help	Арр

Figure 1. 26 Configuration Parameters: Select Data Import/Export to see what is saved to workspace.

As seen from Figure 1.25, the *Outport* block corresponding to *yout*, is connected to x(t) or *yout* = x. For instance, plot command can now be utilized as follows

>> plot(yout)

The result is displayed in Figure 1.27.



Figure 1. 27 The simulation result yout=x is plotted with *plot* command. The same result can be seen with SIMULINK scope. Here below with both x(t) and dx/dt



The sharpness of the yellow curve around t = 0.4 s is not real, it should be smooth. First you might suspect numerical difficulties due to too large a step size. This is not the case. It is due to display graphics, i.e., not enough points have been saved to have a smooth presentation.

The damping factor can be changed by changing the coefficient 5 in front of $\frac{dx}{dt}$. If the coefficient is zero (no damping), the result is a sinusoidal. Increasing the damping will result in damping oscillations. Complete the study to obtain the following responses. Use Matlab command **'hold on'** to plot several graphs to the same window.



Figure 1. 29 Simulation results vary, when damping is changed. When damping constant is small, oscillation is large and when damping constant is bigger, there is no oscillation.

Let us also plot a **phase plane** plot (x vs dx/dt). Note that the time has been eliminated and it only appears as a parameter. To see the effect better, start with less damping. Change the coefficient 5 to 1.



Figure 1. 30 XY Graph scope is used to draw a phase plane plot. The result is shown below.



Figure 1. 31 Phase plane plot of (1.7). The resulting figure should be smooth. *XY Graph* does not adjust the scales automatically. In order to see the whole picture in Figure 1.31, click the XY Graph block in Simulink and adjust the scales. Adjusting also the *Sample time* from -1 to 0.1 (or Simulation => Model Settings => Solver => *Relative tolerance* from 1e-3 to 1e-6) results in a smoother Figure.

SIMULATION	DEBUG	MODELING	FORMAT		APPS	Diocy	
New I Save I Sav	Library Browser LIBRARY untitled	Log Signals	Add Viewer PREPARE	Signal Table	▼ Stop Norm	Y scope. (mask) (link) Plots second input (Y) against first input (X) at each time step to create an X-Y plot. Ignores data outside the ranges specified by x-min, x-max, y-min, y-max. Parameters	5
B Image: Constraint of the second					XY Graph	X-min: -2 : X-max: 2 : Y-min: -1 : Y-max: 1 : Sample time: 0.1 : 0.1 :	

Figure 1.32 Change the scaling *Parameters* and the sample time, in order to obtain a better figure than the one in Figure 1.31. The result is shown in Figure 1.33.



Figure 1.33 The whole, smooth phase plane plot is displayed. A stable system. It converges towards the origin. Physical interpretation: In the origin both position and velocity are zero.