# Module 8

## Basic Operation of Simulink for Dynamic Simulation

### Table of Contents

**Overview**

This Module will serve to introduce basic operation of Simulink for dynamic simulation. In a Simulink model scientific computation algorithms can be done by block diagrams.

**Learning Outcomes**

Upon completion of this Tutorial you will be able to:

- Create and run a Simulink model;
- Compute value of a function and display computed results with Simumulink;
- Use blocks in Simulink Library Browser;
- Configure system parameters;
- Configure simulation parameters; and
- Store simulated data to Workspace and save Workspace to files.

**Focus Questions**

- How is a Simulink model created from Simulink Library Browser?
- How are Simulink model configuration parameters configured?
- How are parameters of a dynamic system set in a Simulink model?
- How are data stored and saved within a Simulink model?

# 1. Getting Started with Simulink

## 1.1 Products of MathWorks

http://www.mathworks.com/

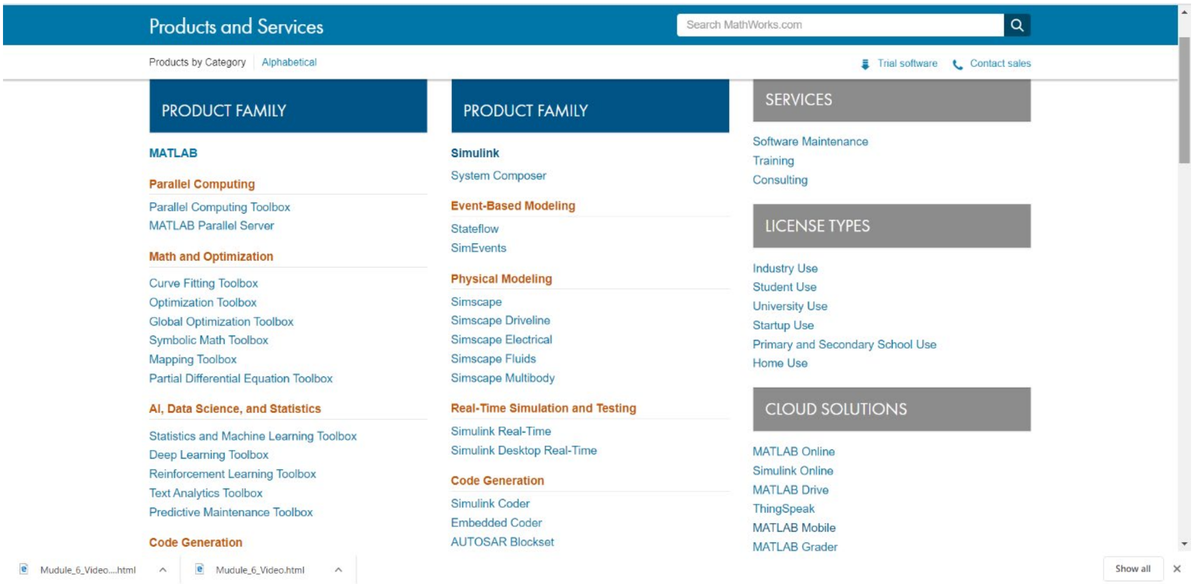



**Figure 1** Overview of MathWorks products

## 1.2 Toolboxes for Instrumentation and Process Control

List of https://au.mathworks.com/products.html?s_tid=gn_ps

**MATLAB (Current Version 2020a Released in March 2020)**

**Control System Design and Analysis**

- Control System Toolbox
- System Identification Toolbox
- Fuzzy Logic Toolbox
- Robust Control Toolbox
- Model Predictive Control Toolbox
- Aerospace Toolbox

**Test & Measurement**

Data Acquisition Toolbox
Instrument Control Toolbox
Image Acquisition Toolbox
SystemTest
OPC Toolbox
Vehicle Network Toolbox

**Simulink (R) (Current Version 10.1, March 2020)**

**Physical Modeling**

- Simscape
- SimMechanics
- SimPowerSystems
- SimDriveline
- SimHydraulics

- SimElectronics

**Control System Design and Analysis**

- Simulink Control Design
- Aerospace Blockset
- Simulink Design Optimization

**Code Generation**

- Real-Time Workshop
- Real-Time Workshop Embedded Coder
- Stateflow Coder
- Simulink HDL Coder

**Rapid Prototyping and HIL Simulation**

xPC Target
xPC Target Embedded Option
Real-Time Windows Target

**Control Engineering Lab (Old Cavitation Tunnel Building)**

**MATLAB (R2019b)**

**Math and Optimization**

Symbolic Math Toolbox

**Control System Design and Analysis**

- Control System Toolbox
- System Identification Toolbox

**Test & Measurement**

Data Acquisition Toolbox

**Simulink**

**Code Generation**

- Real-Time Workshop

**Rapid Prototyping and HIL Simulation**

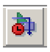
xPC Target

### 1.3 Background of Simulink®

#### 1.3.1 What Is Simulink?

Simulink, a companion program to MATLAB (the current version of MATLAB at AMC: R2010b), 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 multi-rate systems.

#### 1.3.2 Simulink Library Browser

The first step is to start up MATLAB on the computer you are using. Type `>>simulink` in the Command Window or click the  button in the Toolbar menu of the MATLAB Window. The Simulink Window will appear as that shown in **Fig. 1a**. Then click Blank Model you will have a Simulink blank model window as shown in **Fig. 1b**.

When clicking the Library Browser button in a Simulink blank model, the Simulink Library Browser including all built-in blocks will appear as win **Fig. 1c**. the Simulink® 10.0 has a number of additional options. There are several groups of Simulink blocks in the Simulink icon such as

Commonly Used Blocks, Continuous, Discontinuities, Math Operations, Sinks and Sources, etc. Selecting Commonly Used Blocks will provide a list of blocks shown in **Fig. 2**.

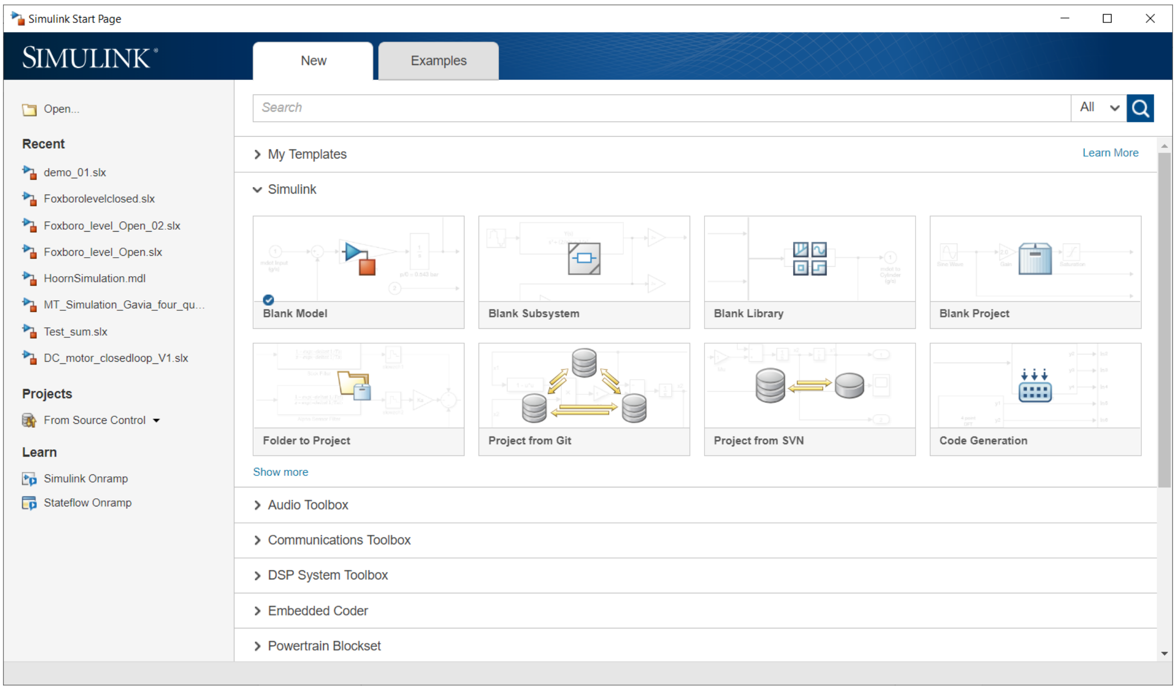



**Figure 1a** Simulink Window

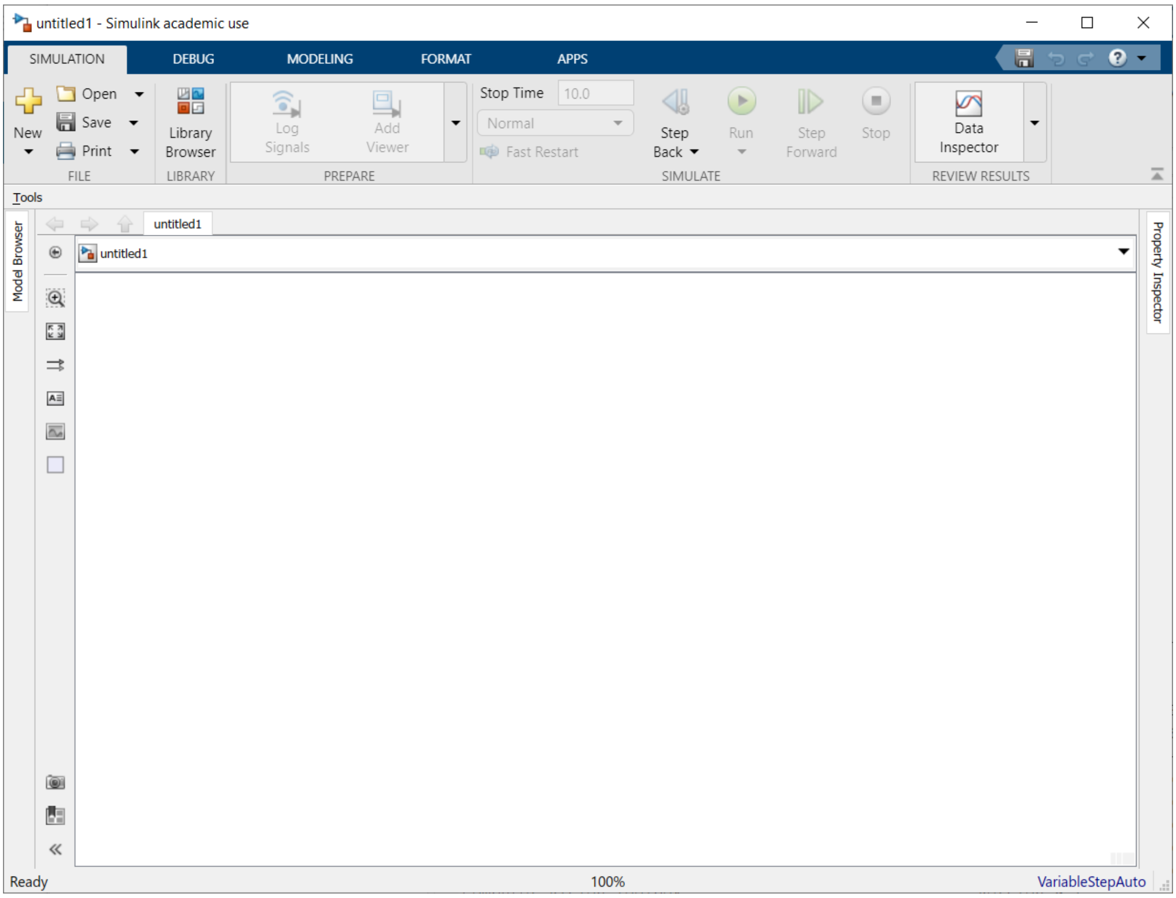



**Figure 1b** Simulink model window (Blank model)

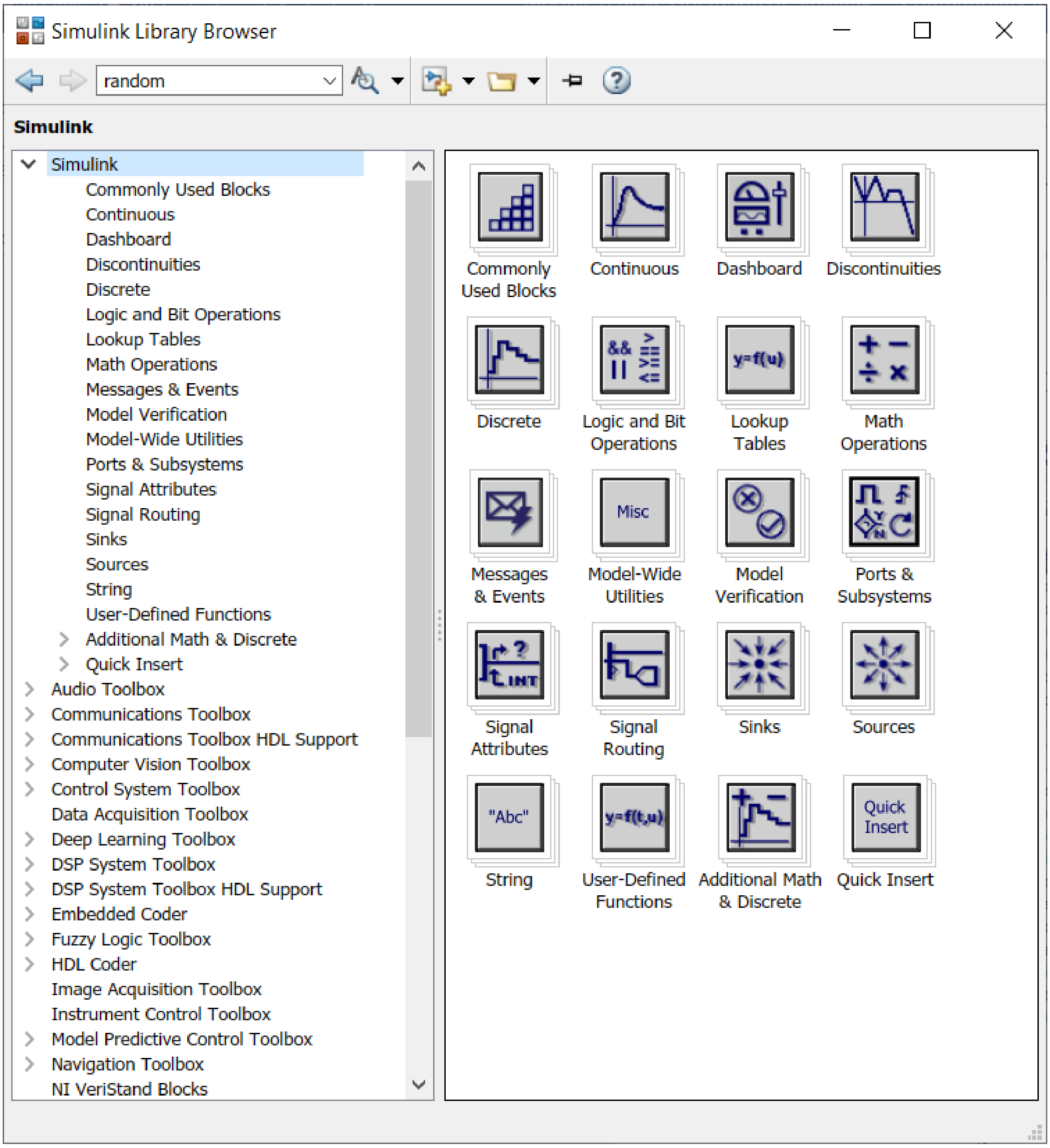



**Figure 1c** Simulink library browser (MATLAB R2019b (9.7), Simulink 10.0)

***Notes***: The current version of MATLAB/Simulink is 2019b for this Module.

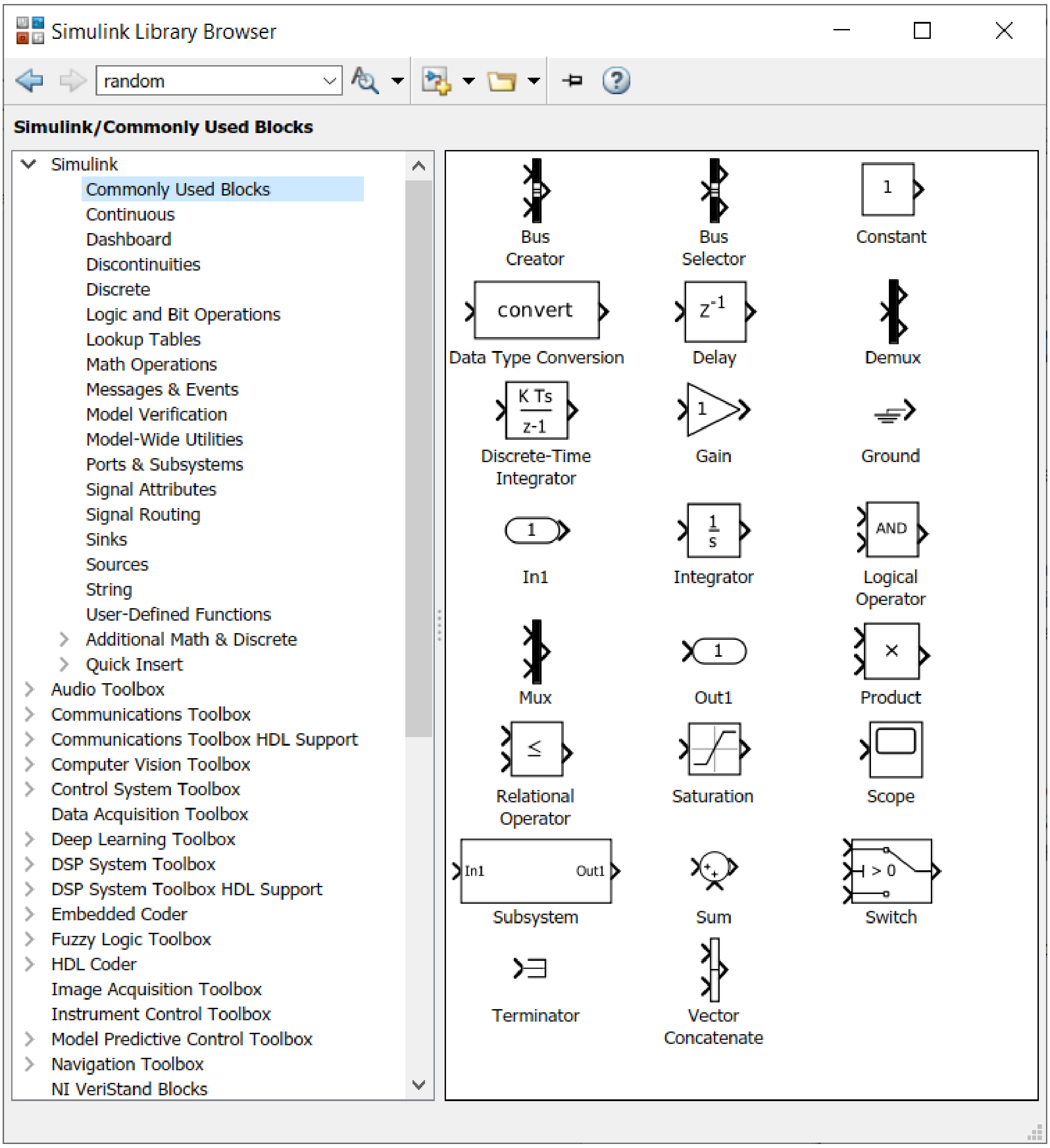



**Figure 2** A list of blocks in Commonly Used Blocks group

Selecting Continuous will provide a list of blocks shown in **Fig. 3**. The ones that we often use are Transfer Fcn, State-space and Integrator.

Selecting the Sources icon yields the library shown in **Fig. 4**. The most commonly used sources are Clock (which is used to generate a time vector), Step (which generates a step input), and Constant (that generate a constant function).

The Sinks icon as shown in **Fig. 5** provides a set of Sinks blocks that are used to display simulated results. The most often used blocks may be To Workspace (to which a variable passed is written to a vector in the MATLAB Workspace), Scope (to represent data graphically).

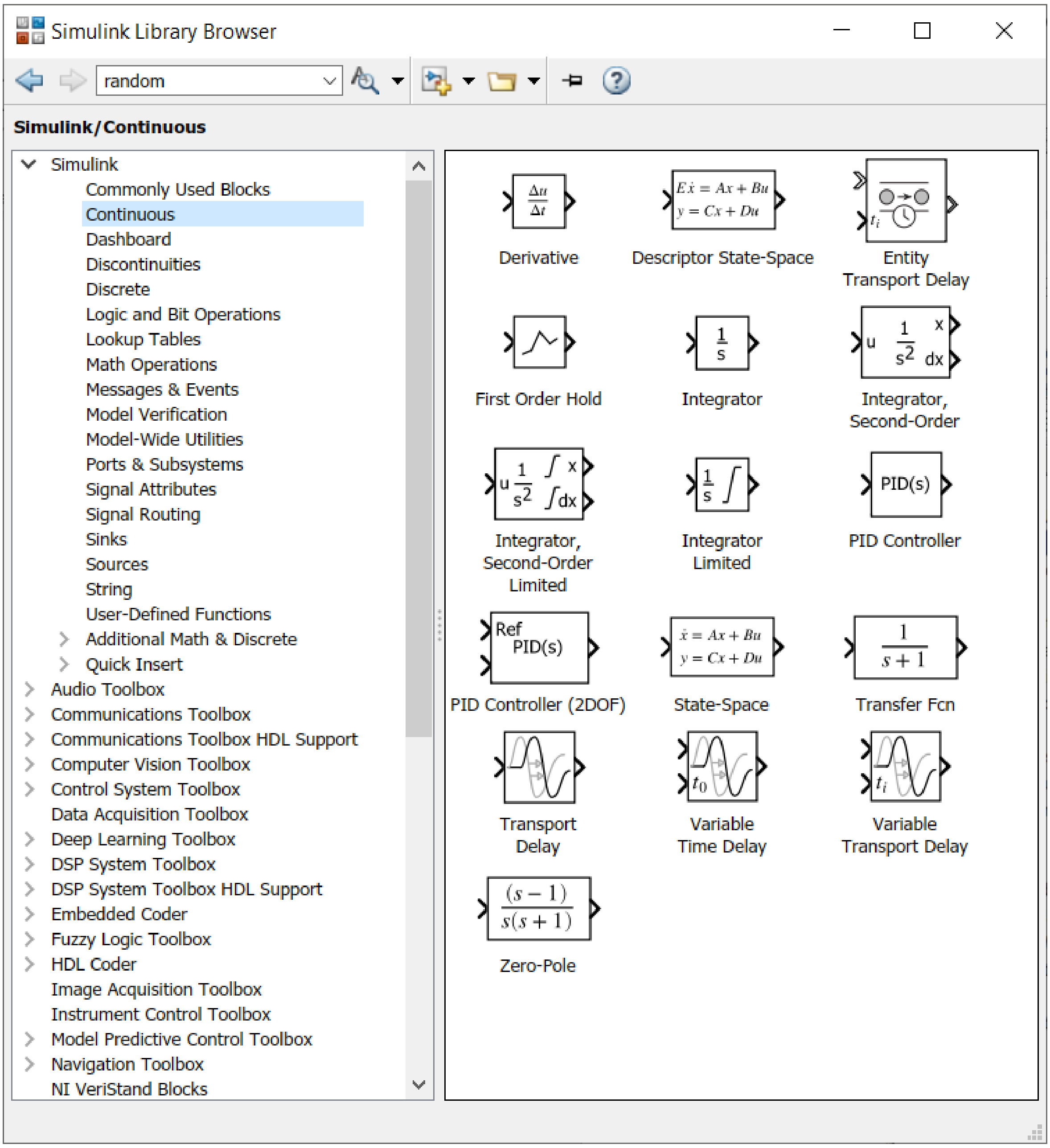



**Figure 3** A list of blocks in Continuous group

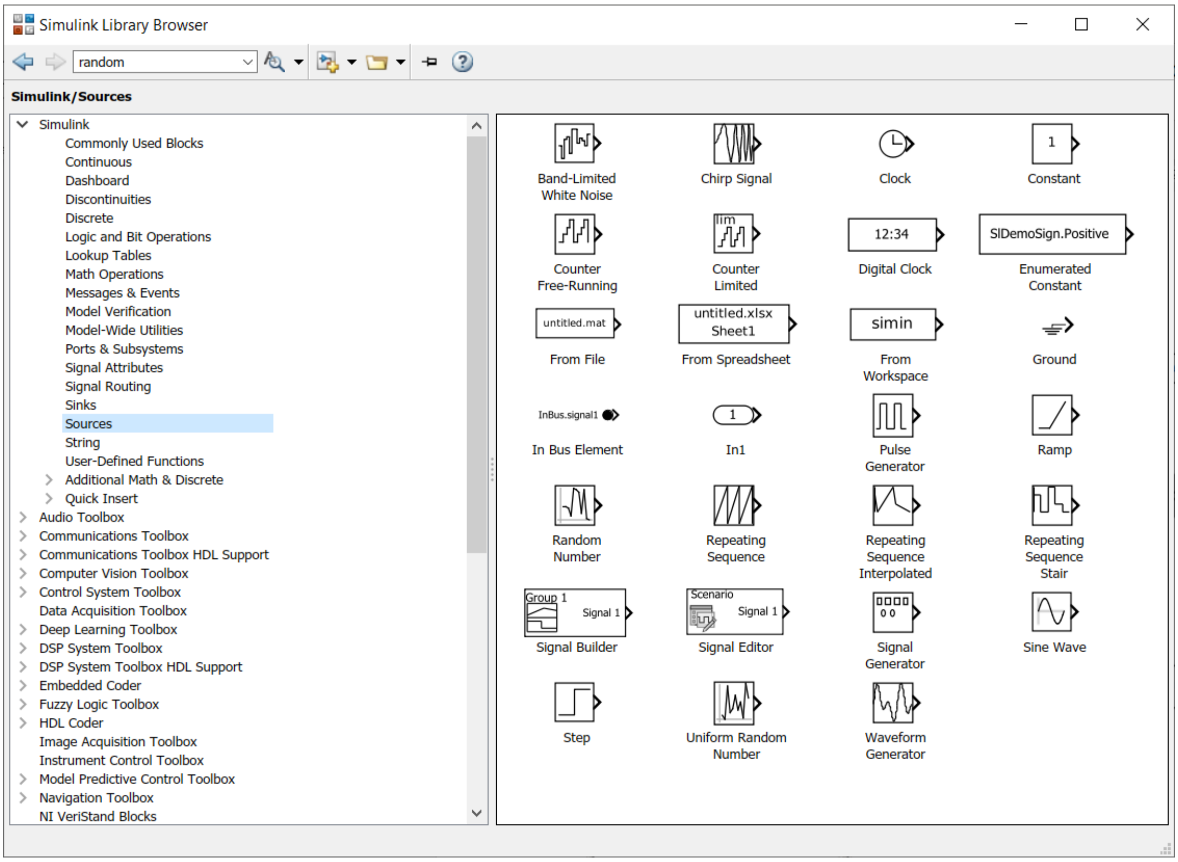



**Figure 4** A list of blocks in the Sources group

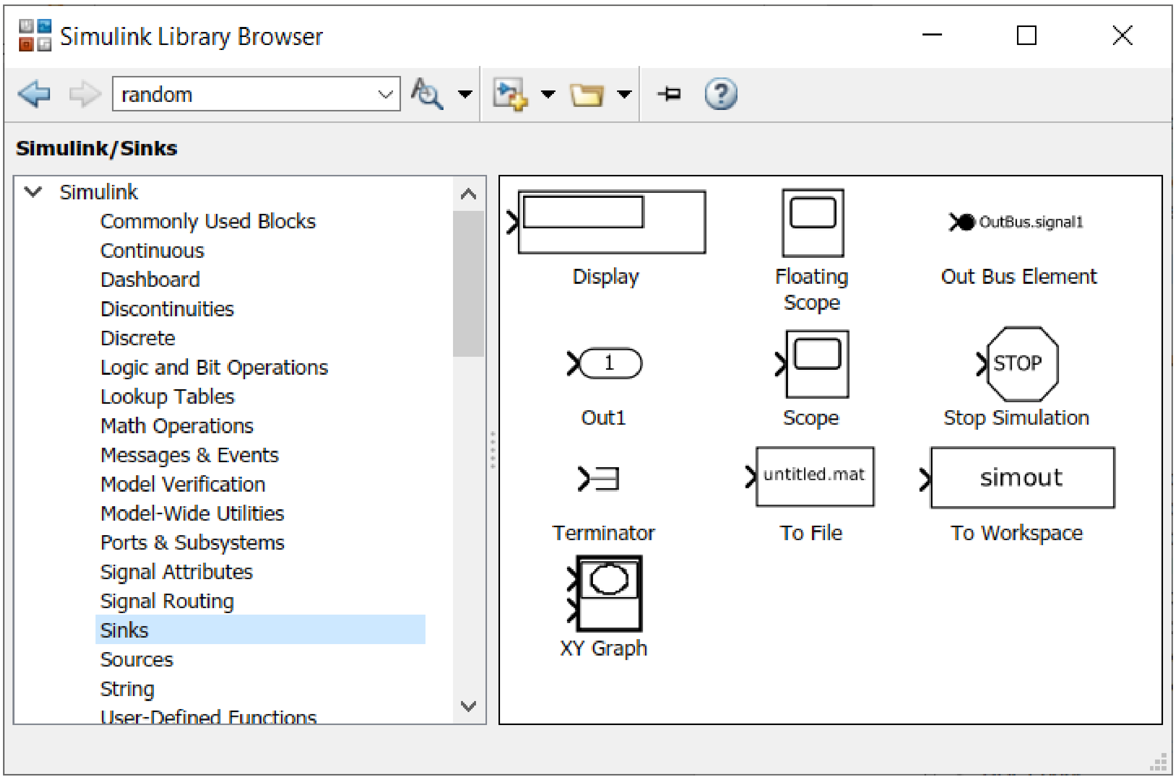



**Figure 5** A list of blocks in Sinks group

### 1.3.3 Concept of Signal and Logic Flow

In Simulink, data/information from various blocks is sent to another block by lines connecting the relevant blocks. Signals can be ***generated*** and fed into blocks (dynamic / static). Data can be fed into functions. Data can then be dumped into ***sinks,*** which could be scopes, displays or could be saved to a file. Data can be connected from one block to another, can be branched, multiplexed etc. In simulation, data is processed and transferred only at ***Discrete*** times, since all computers are discrete systems. Thus, a SIMULATION time step (otherwise called an INTEGRATION time step) is essential, and the selection of that step is determined by the fastest dynamics in the simulated system.

### 1.3.4 Connecting Blocks

A Simulink model is created based on the idea of block diagram and it consists of several blocks. Necessary blocks for a Simulink model can be selected from the Simulink Library Browser and they are connected together. To connect blocks, *left-click* and drag the mouse from the output of one block to the input of another block. **Figure 6** shows the steps involved. The Simulink model in **Fig. 6** consists of the following blocks: Clock, To Workspace1, Step Input, Transfer Fcn and To Workspace.

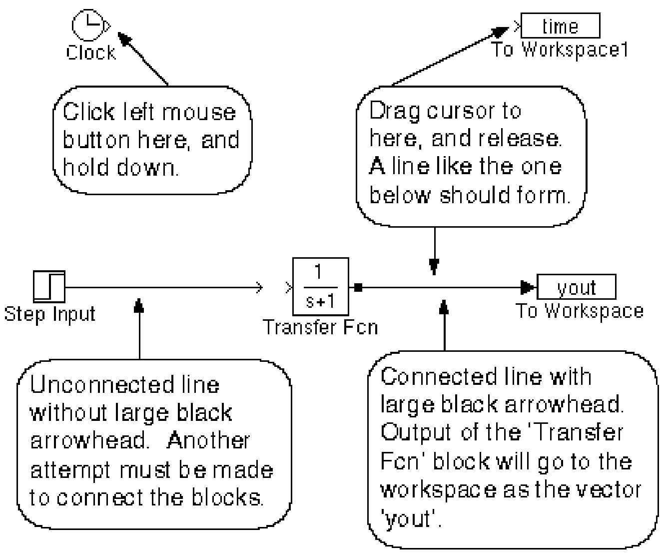



**Figure 6** Connecting blocks

## 1.4 Procedure for Simulink Simulation

With Simulink we can develop block diagram algorithms to solve ODEs or do scientific computation. Although computer simulations can be used to model a large variety of systems, it can be seen that all computer simulations must embody the following components:

**1. Structure of the mathematical model:** This is the complete set of differential equations that describe the system behaviour (dynamics) and reflect the fundamental physical laws governing the behaviour of the system.

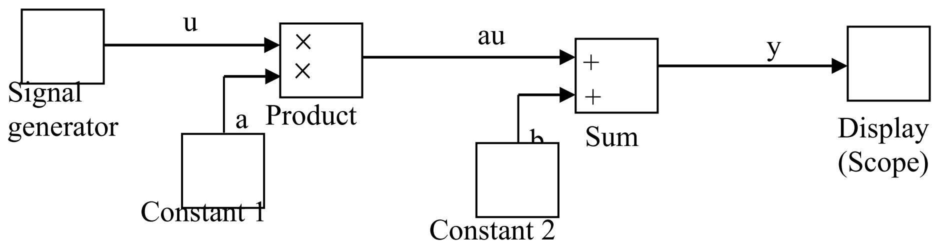
**2. Model parameter values:** Model parameters refer to numerical constants that usually do not change over the course of the simulation. Typical parameters for mechanical systems are mass, damping coefficient, and spring stiffness.
**3. Initial conditions:** Initial conditions are important for determination of the solution of ODEs. Therefore in the simulation program initial conditions must be set.

**<u>4. Inputs:</u>** Typically a system responds to one or more inputs. The simulation must embody the inputs as well. In simulation tools, there are some blocks that generate test input signals such as a step function, a ramp function or a sine-wave function.

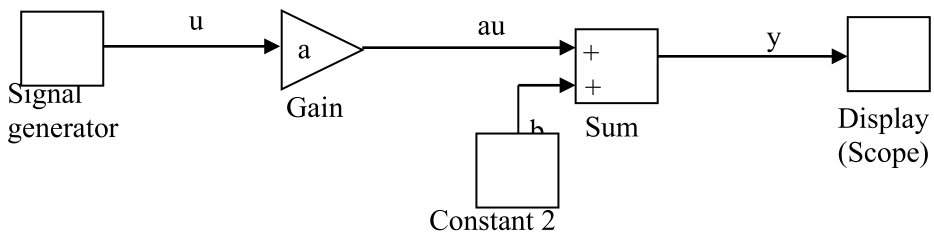
**<u>5. Outputs:</u>** Although a simulation does not require that the user explicitly define outputs, it is assumed that the goal of a computer simulation of a dynamic system is the time history of specific physical variables in the system under study. The time history of output variables can be stored to the computer hard drive for later analysis or displayed as graphs on the screen. The simulation tools often have blocks for displaying simulated outputs.

**<u>6. Simulation solution control parameters:</u>** Simulation solution control parameters define the values and choices made by the designer/engineer of the simulation tool who dictates how the numerical methods behind the simulation operate. These include values that determine the step size, output interval, error tolerance, and choice of numerical integration algorithm.

**Example:** in order to calculate y = au + b we develop a block diagram as follows:




(a) An example of block diagram algorithm with a “Product” block



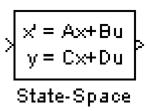

(b) An example of block diagram algorithm with a “Gain” block

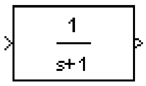
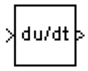
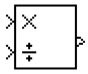
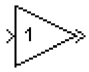
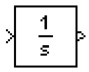
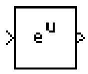
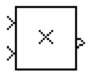
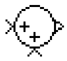
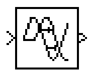
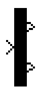
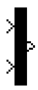
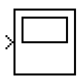
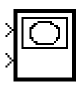
**Figure 7** Examples of block diagram algorithm

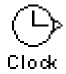
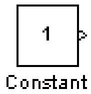
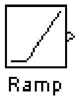
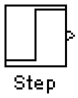
### 1.5 Commonly Used Simulink Blocks

**Table 1** summarises a list of commonly used Simulink blocks which we will the most often use in our course.

**Table 1** Summary of Commonly Used Simulink Blocks

| Block icon | Name | Use |
|---|---|---|
| **Continuous** | | |
|  | State-Space | Implement a linear state-space system |

| | | |
|---|---|---|
|  Transfer Fcn | Transfer Fcn | Implement a linear transfer function |
| **Math Operations** | | |
|  Derivative | Derivative | Merge scalar, vector or matrix signals |
|  Divide | Divide | Multiply or divide inputs |
|  Fcn | Function | Apply a specified expression to the input |
|  Gain | Gain | Multiplies the input by a constant value (gain) |
|  Integrator | Integrator | Integrate the input signal |
|  Math Function | Math Function | Perform a mathematical function |
|  Product | Product | Multiply inputs |
|  | Sum | Add or subtract inputs |
|  Transport Delay | Transport Delay | Delay the input by a given amount of time |
| **Signal Routing** | | |
|  | Demux | Split vector signals into scalars or smaller vectors |
|  | Mux | Extract and output the elements of a bus or vector signal |
| **Sinks** | | |
|  Scope | Scope | Display signals generated during a simulation |
|  To Workspace | To Workspace | Write data to the workspace |
|  XY Graph | XY Graph | Display an X-Y plot of signals using a MATLAB figure window |

| **Sources** | | |
|---|---|---|
|  | Clock | Generate a time vector |
|  | Constant | Generate a constant |
|  | Ramp | Output a ramp signal |
|  | Sine Wave | Generate a sine wave signal |
|  | Step | General a step signal |

### 1.6 Summary

Simulink is a very powerful block diagram simulation language. Simple simulations can be set up rapidly. The aim of this tutorial was to provide enough of an introduction to get you started on the development of simulations for dynamic systems. With experience, the development of these simulations will become second nature. It is recommended that you perform the simulations shown in these tutorials as well as the follow-up exercises, to rapidly acquire these simulation skills.

### 1.7 Further Information on Simulink and Mathworks Products

http://www.mathworks.com/help/toolbox/simulink/
http://www.mathworks.com/products/simulink/

## 2. Basic Operation of Simulink for Dynamic Simulation

### 2.1 Aims

- To use Simulink as a simulation and diagnostic tool for dynamic systems
- To simulate instrumentation systems with Simulink
- To do scientific computation with Simulink

### 2.2 Learning Outcomes

Upon completion of this Tutorial you will be able to:

- Create and run a Simulink model
- Compute value of a function and display computed results with Simumulink
- Use blocks in Simulink Library Browser
- Configure system parameters
- Configure simulation parameters
- Store simulated data to Workspace and save Workspace to files

### 2.3 Hands-on Exercises

#### 2.3.1 Hands-on Exercise 1 Create and Run a simple Simulink model

**Statement of Problem**

Create a Simulink model to simulate a signal that have the following form:

$$y = 2 + 3\sin\left(\omega t + \pi/2\right) \quad (t = 0 \text{ to } 10,\ \omega = 2 \text{ rad/s}) \tag{1}$$

In the Simulink simulation program, do the following:

- Compute value of y vs t (from 0 to 10 seconds)
- Display value of y (with digital display and graph)
- Save data to Workspace

**SOLUTION**
**Block diagram algorithm:**

Block diagram algorithm for function (1) is given in **Fig. 8.**

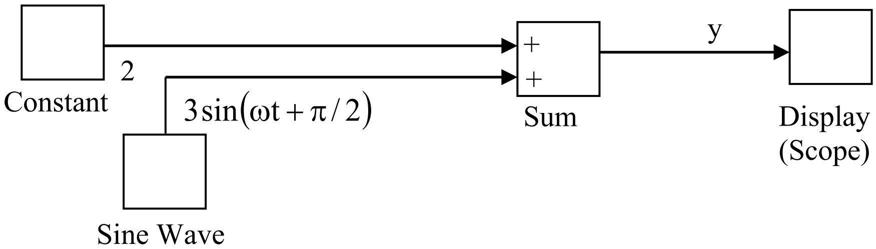



**Figure 8** Block diagram algorithm for equation (1)

**Create a Simulink model:**
Do the following steps:

- Start Simulink
- Open a new model – Save the Simulink model as M8Exericse1.mdl
- Select a Constant block (Sources)
- Select a Sine Wave block (Sources)
- Select a Sum block (Math Operations)
- Select a Clock block (Sources)
- Select a Display (Sinks)
- Copy Display to Display 1
- Select a Scope (Sinks). The Simulink model should look like the one in **Fig. 9**.
- Save the Simulink model.
- Connect/wire blocks as shown in **Fig. 10**.

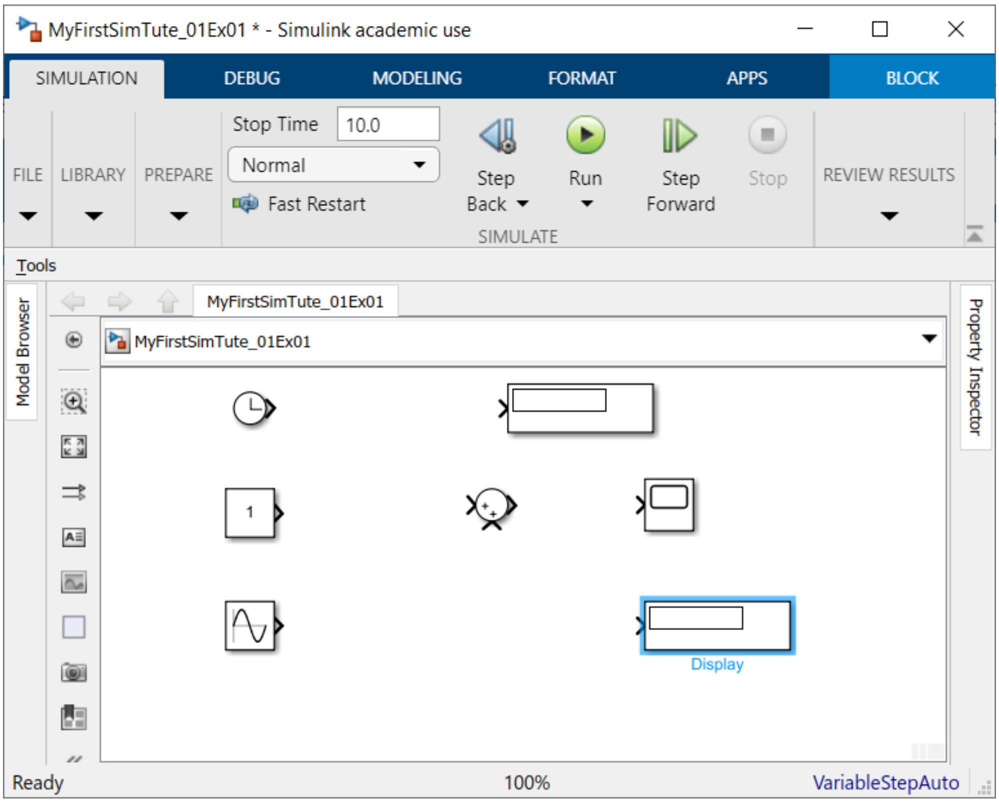



**Figure 9** Start a simple Simulink model

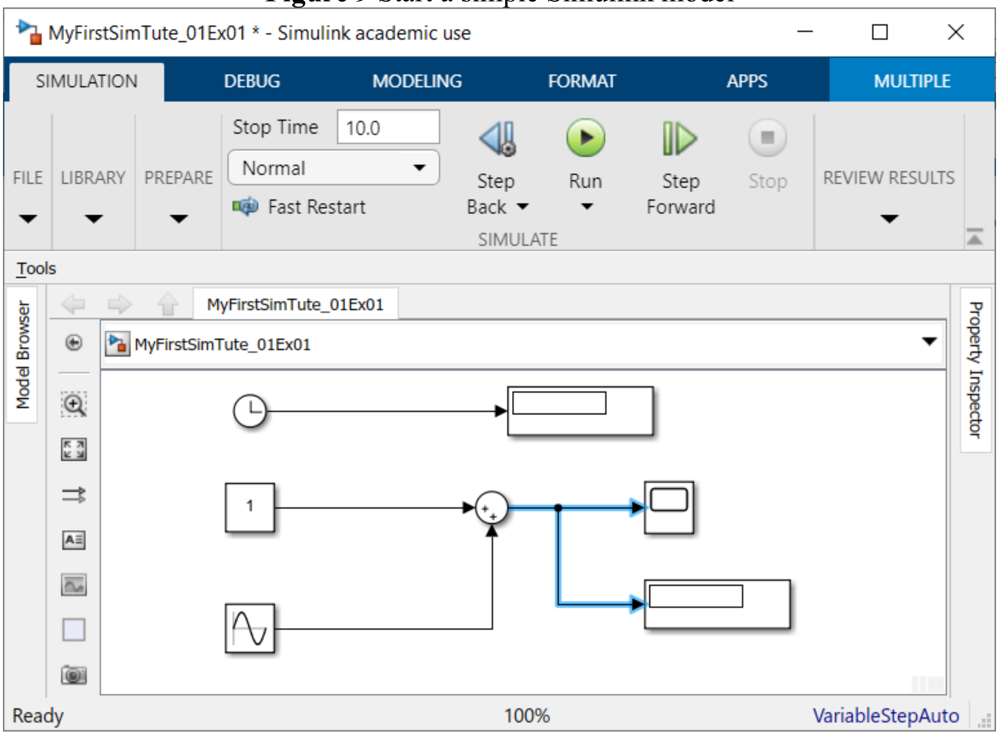



**Figure 10** The Simulink model 'MyFirstSimTute01Ex01.mdl'

**System Parameters:**

- Double-click the Constant, change the Constant value to 2 as shown in **Fig. 11**.
- Click the OK button.

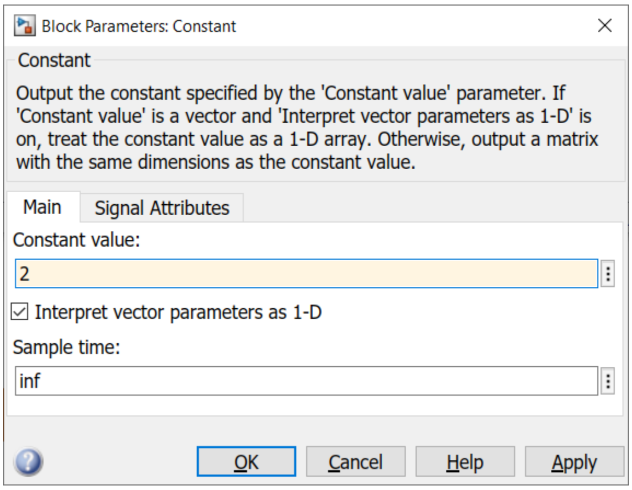



**Figure 11** Change parameter of the Constant block

- Double-click the SineWave block, do some changes as shown in **Fig. 12**.
- Click OK button.

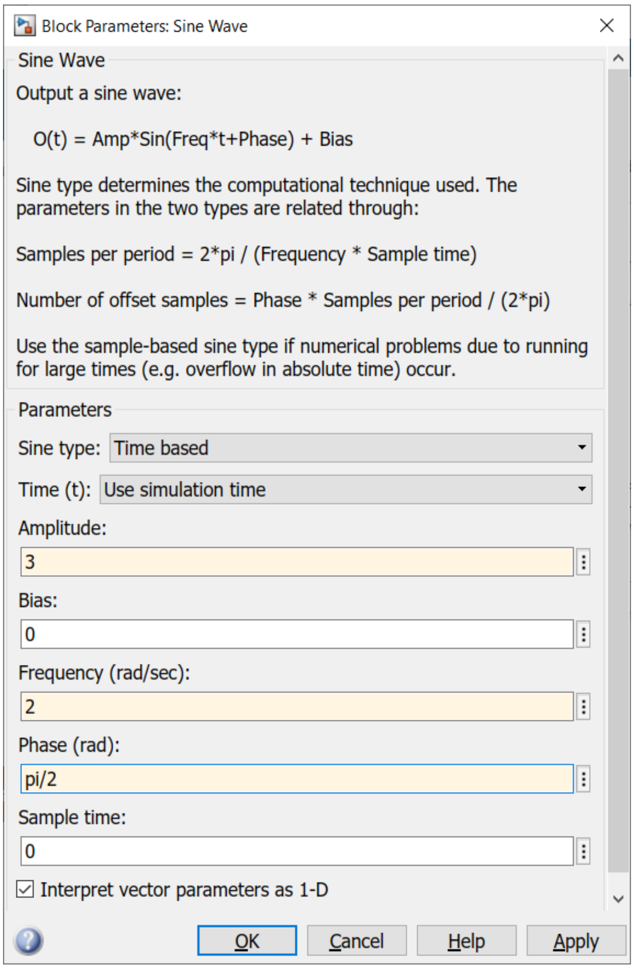



**Figure 12** Settings of sine wave signal

**Configuration of Simulation Parameters:**

- Before running the Simulink model, we need to set simulation parameters by selecting the Modelling Tab > Model Settings > Model Settings (see **Fig. 13**).

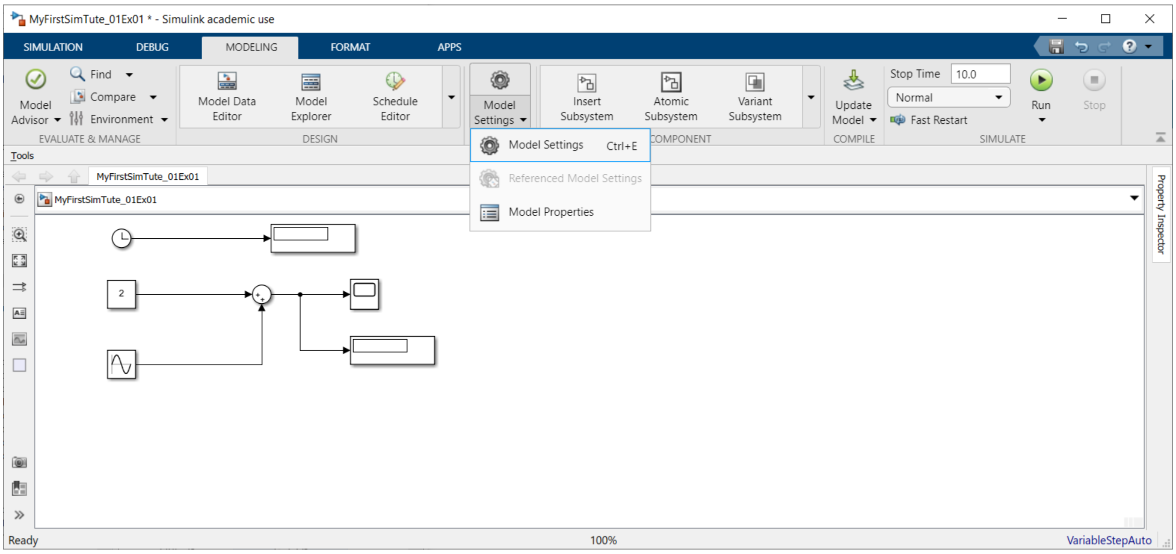



**Figure 13** Simulation tab > Step Back > Configurate simulation stepping

The Configurations dialog box will appear as shown in **Fig. 14**. In this dialog box we can set some parameters such as Start time 0.0, Stop time: 10.0 (seconds), and Solver: ode4 (Runge-Kutta). Fixed-step as 0.1, then Click OK button. The Simulink model is ready to run.

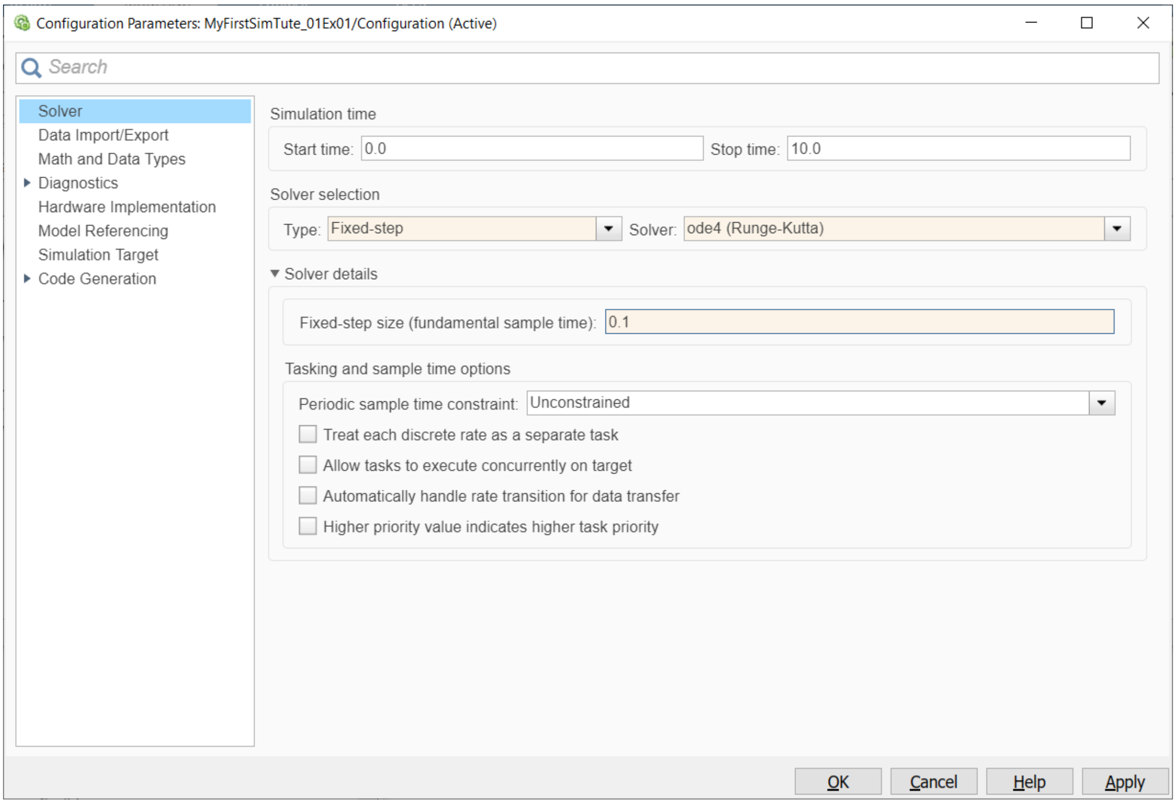
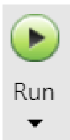



**Figure 14** Configuration Parameter dialog box

**Run Simulink model:**

- Double the Scope block and move it to a desired position on the screen.
- Run the Simulink model by selecting Simulation $\longrightarrow$ Start or click the 

 button.

The resulting Simulink model should look like...

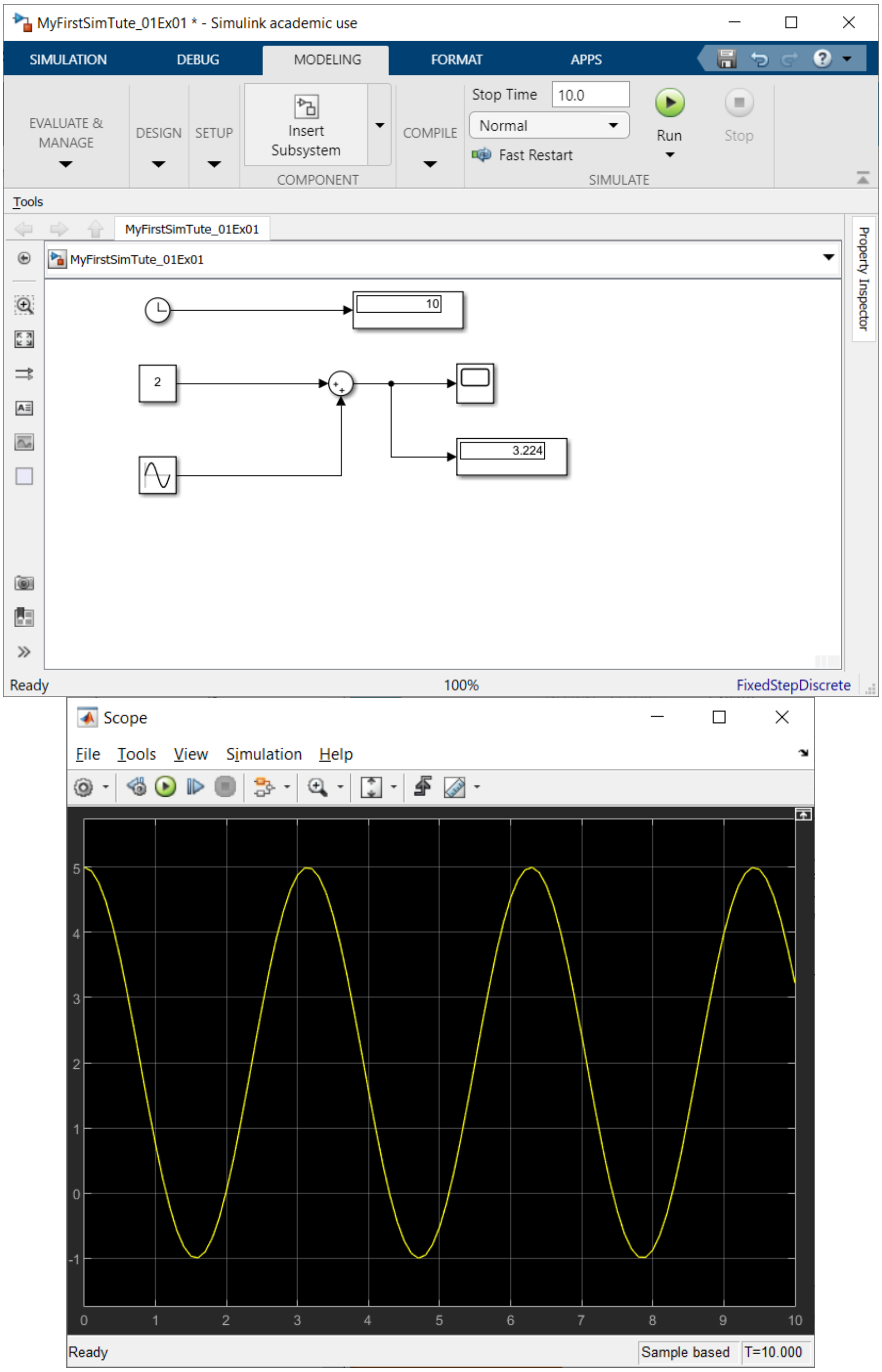



**Figure 15** Resulting Simulink model

**Store Data to Workspace**

**1. Use of To Workspace and a Mux**

There are some ways to save data. Simulated data (including time, t, and value of the function (1), y in a matrix)

- Launch a To Workplace block from the Simulink Library Browser > Sinks
- Launce a Mux block (from Commonly-Used Blocks or Signal Routing)

- Wire them as shown in the following figure:

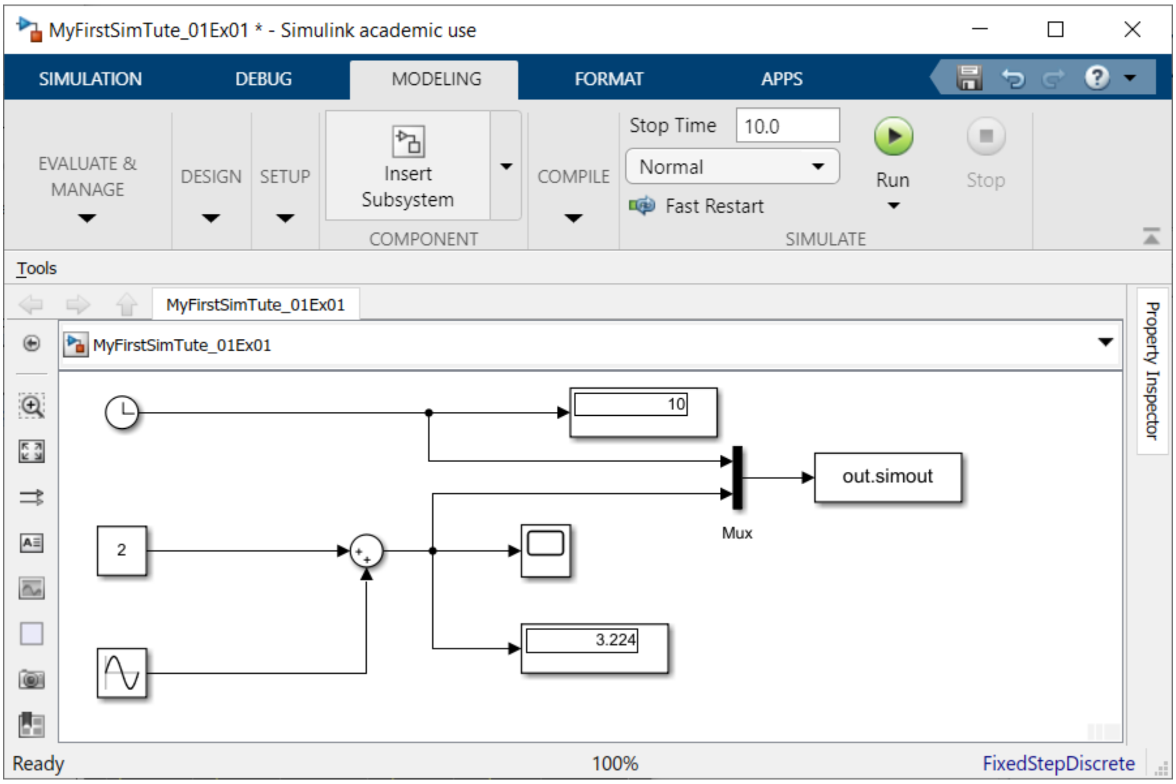



**Figure 16** Launching To Workspace and Mux blocks

- Double click the To Workspace, and set:
    - Variable name: data
    - Save format: Array
- Click the Apply button then the OK button.

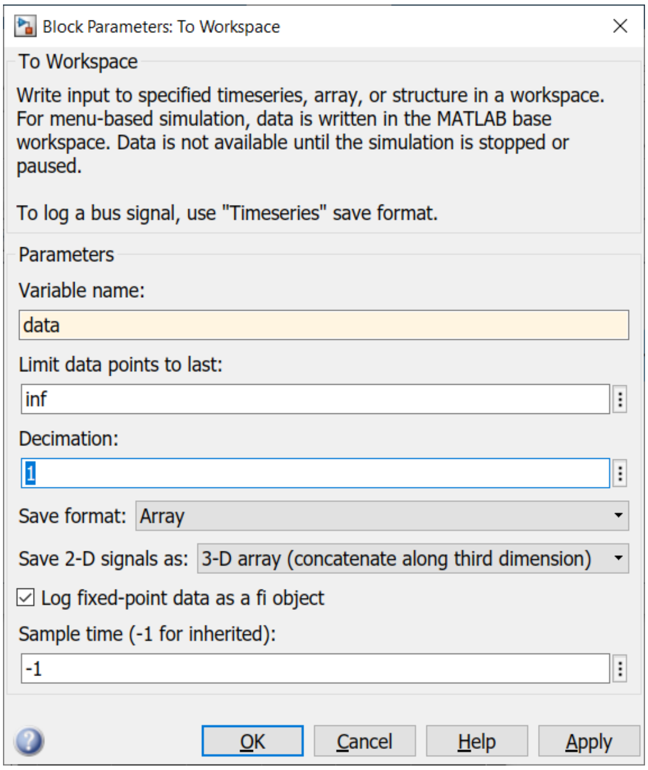



**Figure 17** Sink Block Parameters: To Workplace Window

- Save the Simulink model
- Run the Simulink model. The simulated data are in Workspace (matrix “data”).
- In the Command Window, type the following commands:

```
>>t=out.data(:,1)
>>y=out.data(:,2)
>>plot(t,y);grid
```

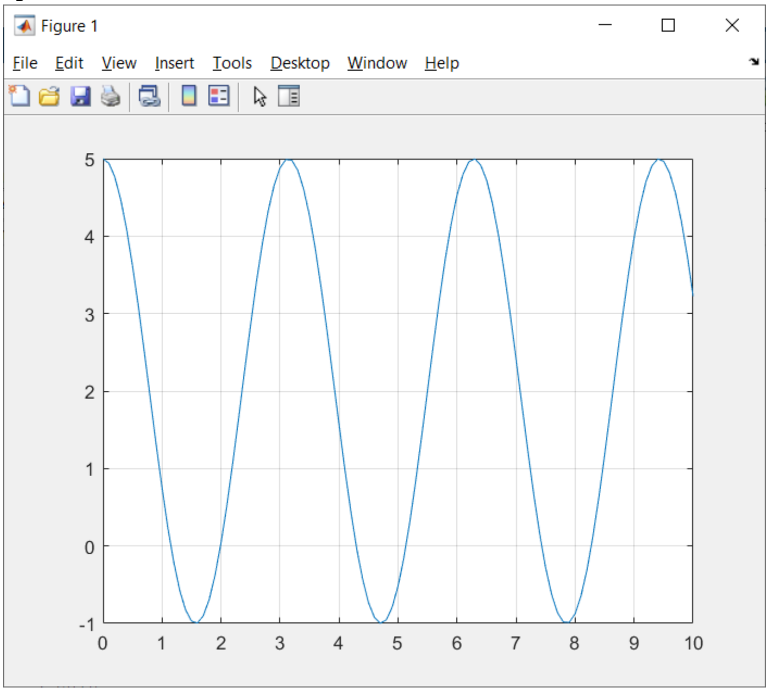



**Figure 18** Graph of $y = 2 + 3\sin(\omega t + \pi / 2)$

## 2. Use of Scope

- Save the Simulink model as "MyFirstSimTute01Ex0102.mdl".
- Double click the Scope block
- Click the Configuration Properties button:

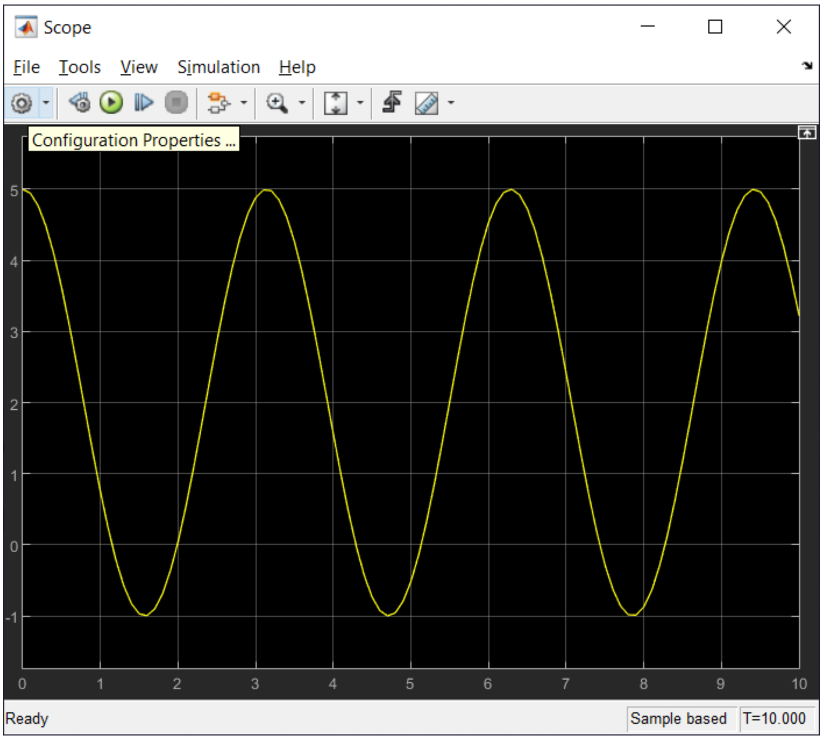



**Figure 19** Scope > Configuration Properties button

- Select the Logging tab
- Check Log data to workspace and set the 'ScopeData' parameters as in the following figure:

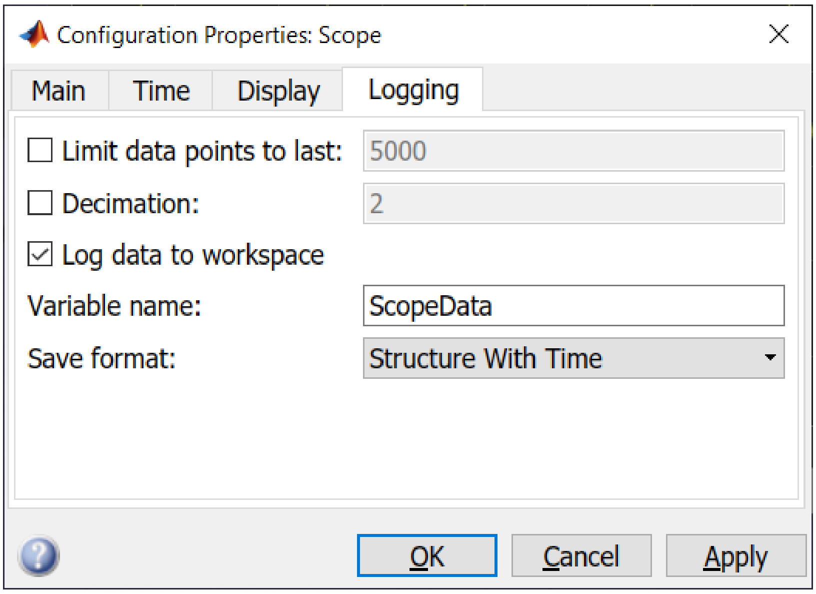



**Figure 20** Logging tab

- Click Apply button
- Click OK button
- Save the Simulink model
- Run the model
- View Workspace... there must be a variable named out | SimulationOutput which includes "ScopeData"

In the Command Window, type the following command:

```
>> out.ScopeData
```

that is resulting...

```
ans =

  struct with fields:

         time: [101×1 double]
      signals: [1×1 struct]
  blockName: 'MyFirstSimTute_01Ex01/Scope'
```

Let's extract signals from the ScopeData variable by typing the following commands in Command Window:

```
>>t=out.ScopeData.time
```
( to retrieve time)

```
>> y=out.ScopeData.signals.values
```
(to retrieve signal y)

```
>>plot(t,y);grid
```

The resulting plot looks like...

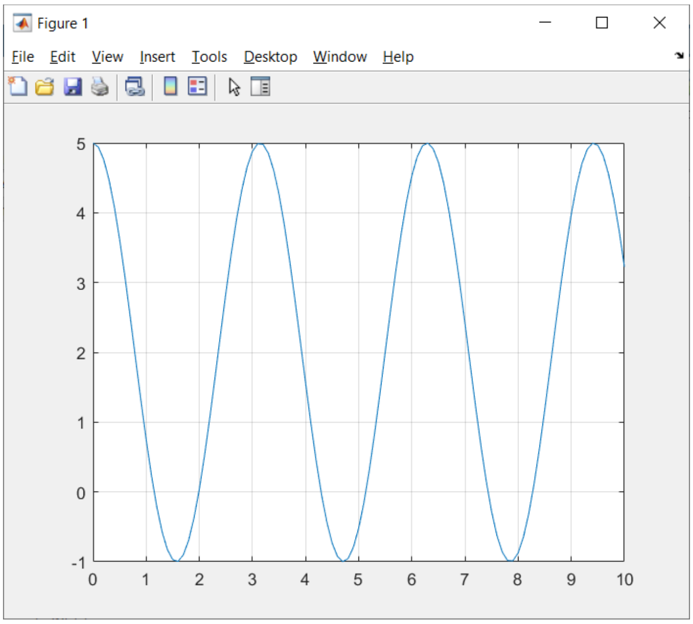



**Figure 21** Plotting data (y vs t) from ScopeData

**Selection of Solver and Step Size:**

- Model Settings Tab > Model Settings > see the following figure:

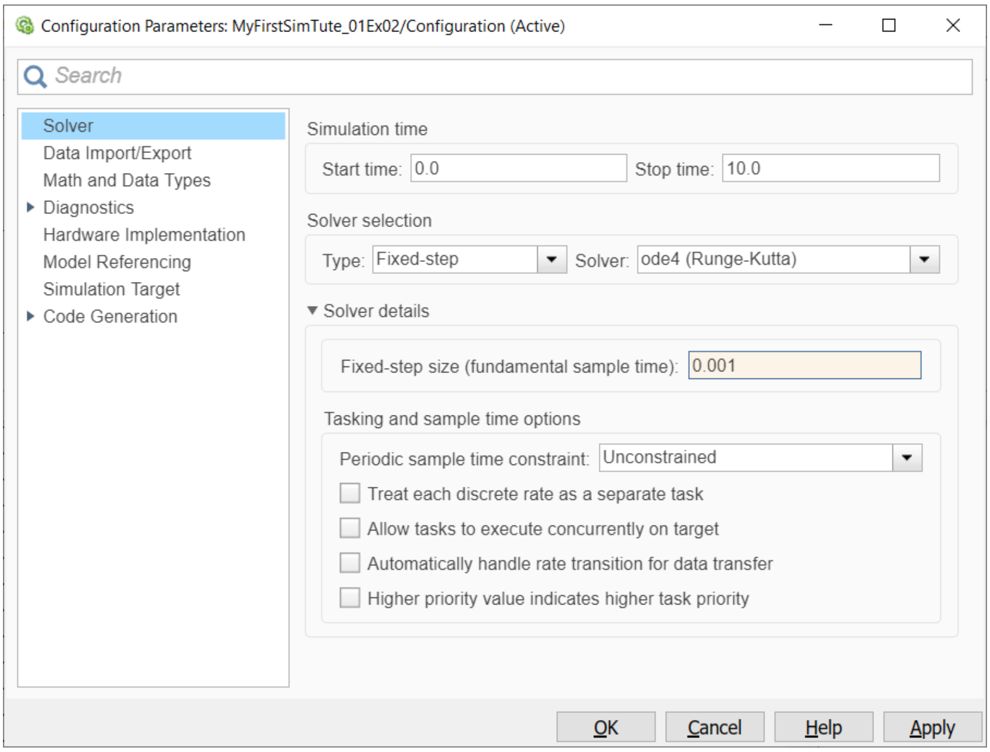



**Figure 22** Change Variable step to Fixed-step and Solver to Ode4 (Runge-Kutta)

- Click Apply button, then OK button. Run the Simulink model and plot y vs t:

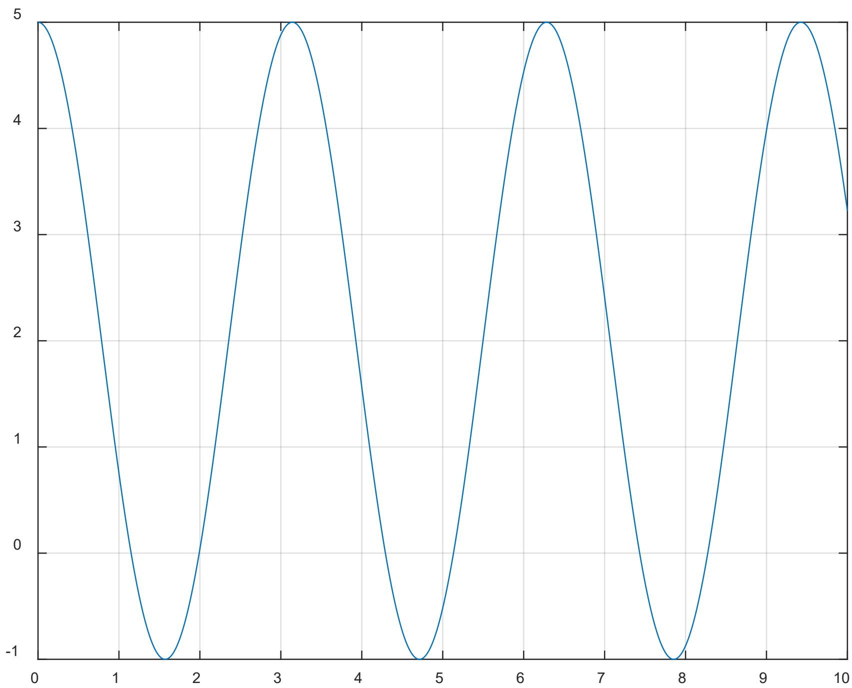



**Figure 23** Smoother graph of $y = 2 + 3\sin(\omega t + \pi/2)$

**Save Workspace to a MAT-formatted files:**

In some cases it is necessary to save all data in Workspace to files. In order to view the contents of Workspace, select the Workspace tab as follows:

Or type the following command in Command Window:

```
>>whos
```

That results in

```
>> whos
  Name          Size                Bytes  Class               Attributes

  ans         101x2                  1616  double
  out           1x1                408272  Simulink.SimulationOutput
  t         10001x1                 80008  double
  y         10001x1                 80008  double
```

Type the following command to save all data in Workspace to “Tute01Ex01.MAT’ file

```
>> save Tute01Ex01
```

The file will be in the Current Directory as follows:

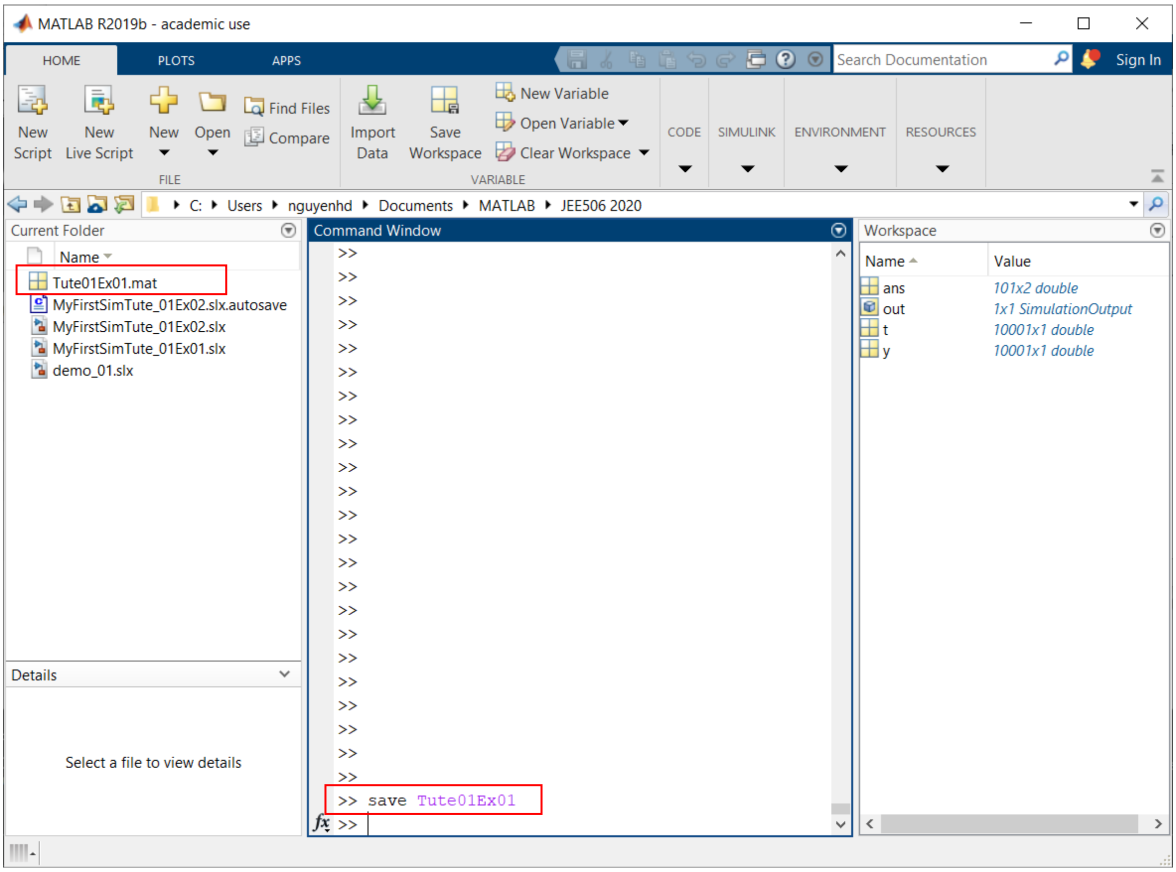


**Figure 24** Workspace tab and a new MAT file was created

To load the MAT file, type the following commands:

```
>>clear
```

```
>>whos
>>load Tute01Ex01
>>whos
```

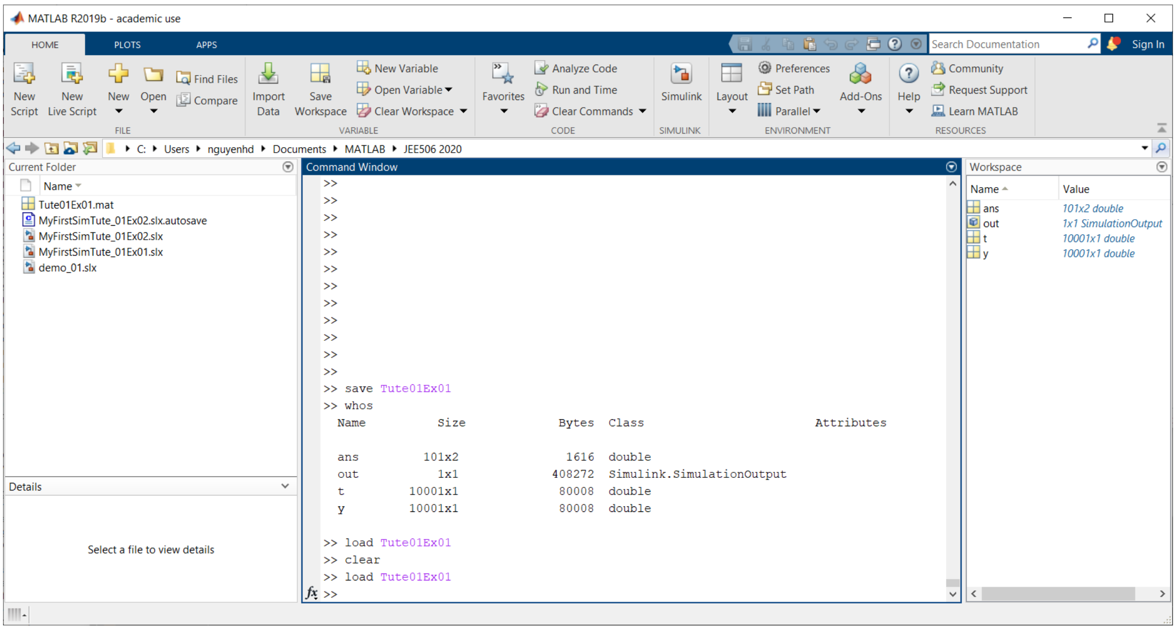


**Figure 25.** Load a MAT file

That results…

```
>> whos
  Name           Size                Bytes  Class                 Attributes

  ans          101x2                  1616  double
  out            1x1                408272  Simulink.SimulationOutput
  t          10001x1                 80008  double
  y          10001x1                 80008  double
```

### 2.3.2 Hands-on Exercise 2 (use of signal generators, manual switch, gain or slider gain and display blocks)

Simulate the following system:

A square wave $y_1 = K \times u_1$

A sine wave $y_2 = K \times u_2$

The simulation program is to switch between $u_1$ and $u_2$. Value of K can be changed during the simulation program is running.

**SOLUTION**

Block diagram for Exercise 2 is shown in **Fig. 26**.

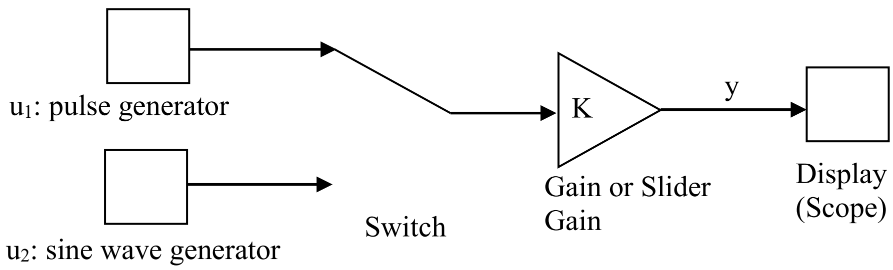



**Figure 26** Block diagram algorithm for equation (1)

Create a Simulink model by the following blocks

- Pulse generator
- Sinewave generator
- Manual switch
- Slider gain
- Scope

Save the model as “M8Exercise2.mdl”. The Simulink model looks like the following figure:

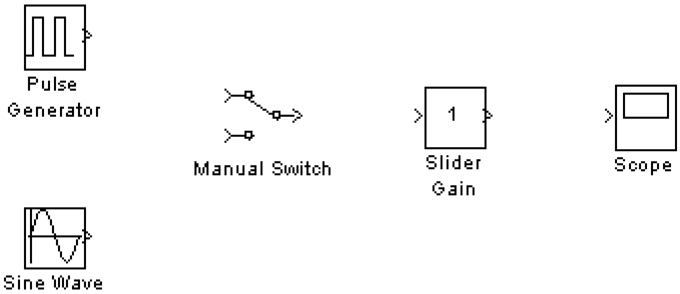



**Figure 27** Necessary blocks for Exercise 2

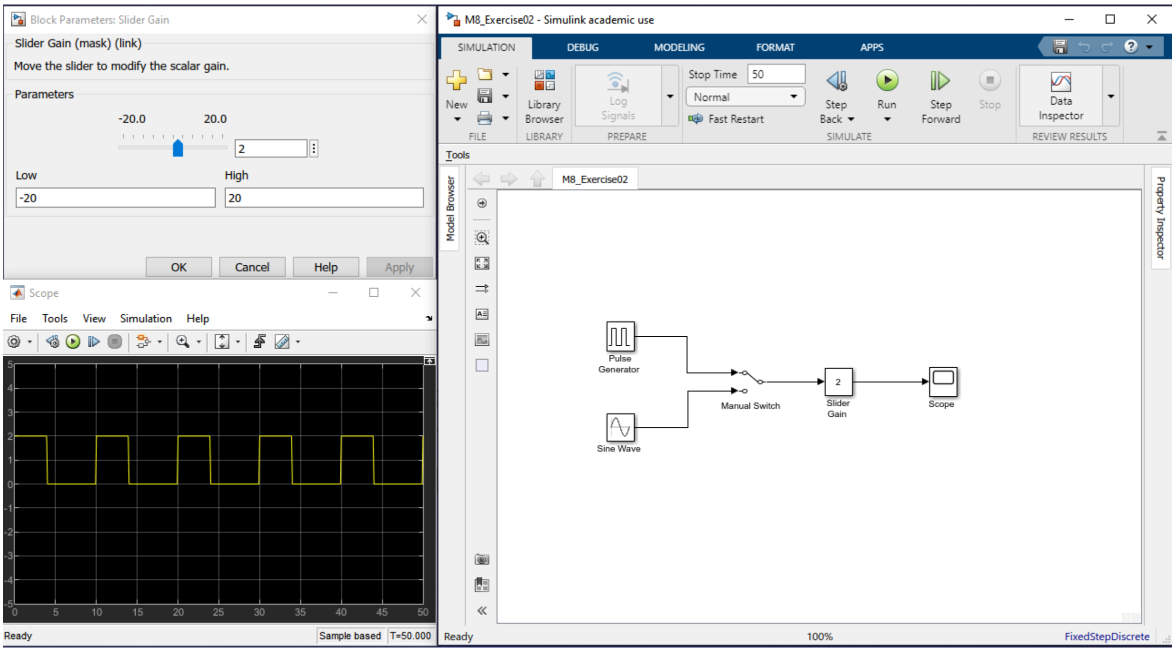
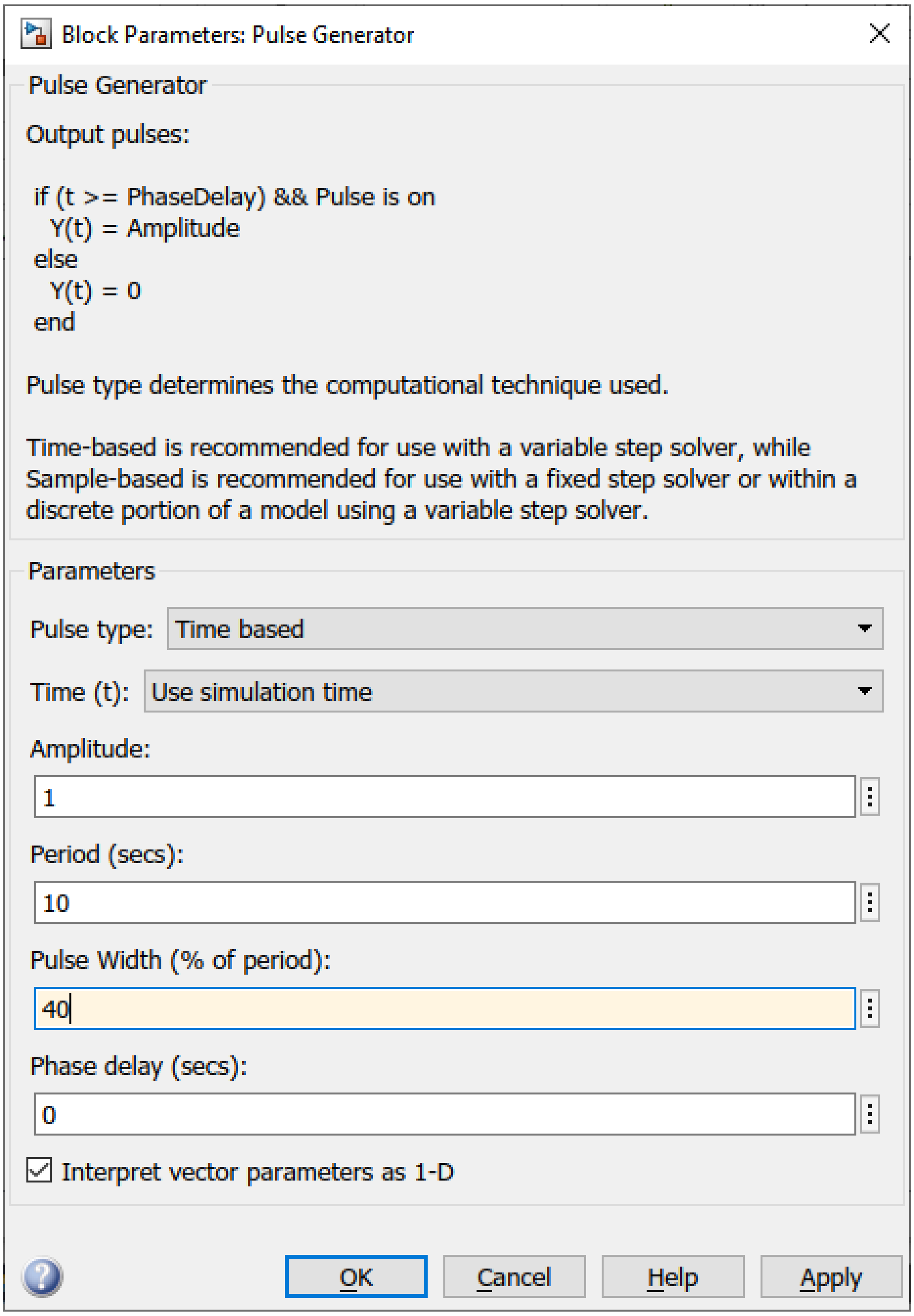
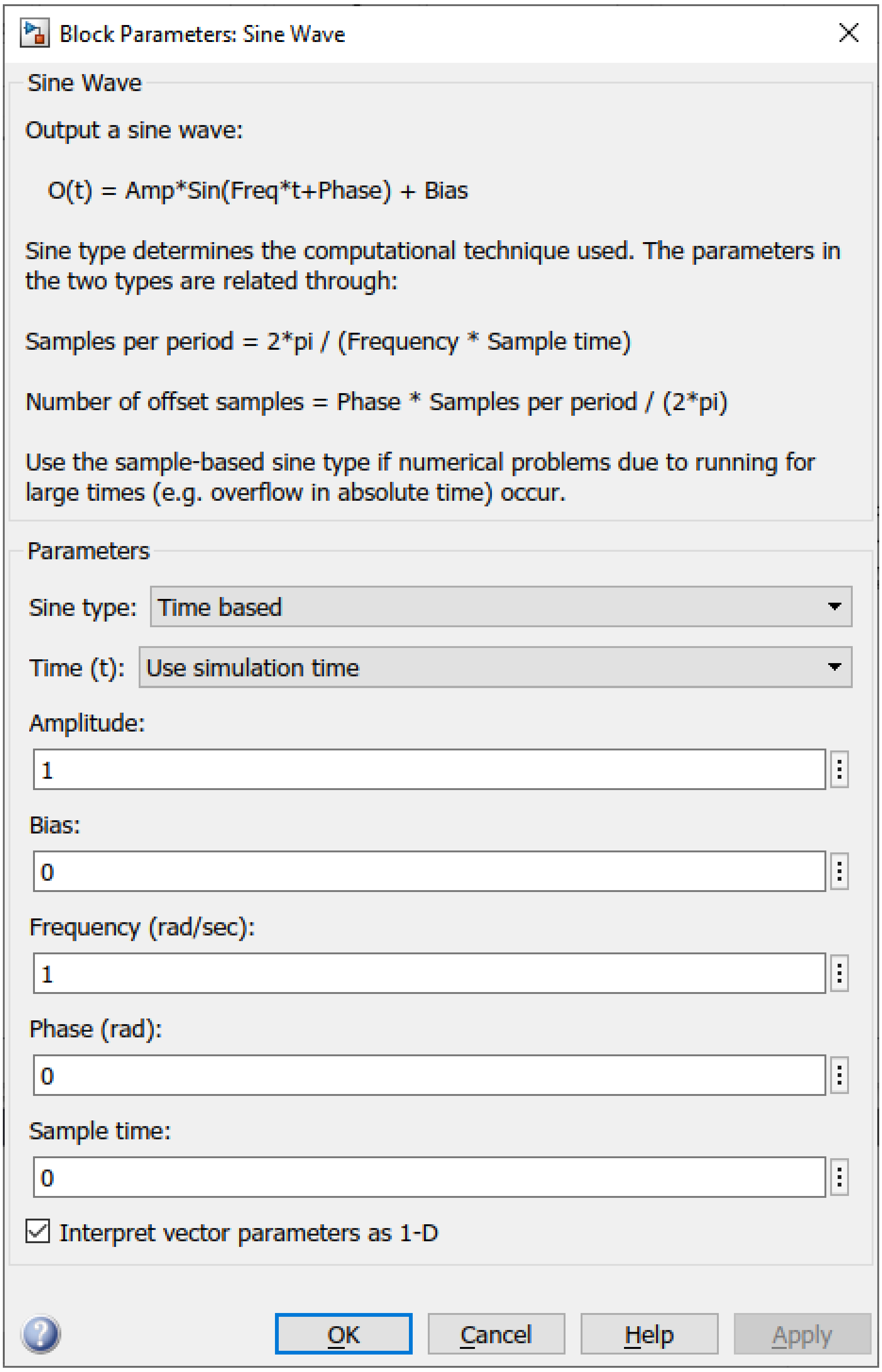



**Figure 28** Simulink model for Exercise 2

Do settings as shown in **Fig. 29**.




Block Parameters: Sine Wave
Sine Wave
Output a sine wave:
O(t) = Amp*Sin(Freq*t+Phase) + Bias
Sine type determines the computational technique used. The parameters in the two types are related through:
Samples per period = 2*pi / (Frequency * Sample time)
Number of offset samples = Phase * Samples per period / (2*pi)
Use the sample-based sine type if numerical problems due to running for large times (e.g. overflow in absolute time) occur.
Parameters
Sine type: Time based
Time (t): Use simulation time
Amplitude:
1
Bias:
0
Frequency (rad/sec):
1
Phase (rad):
0
Sample time:
0
Interpret vector parameters as 1-D
OK
Cancel
Help
Apply

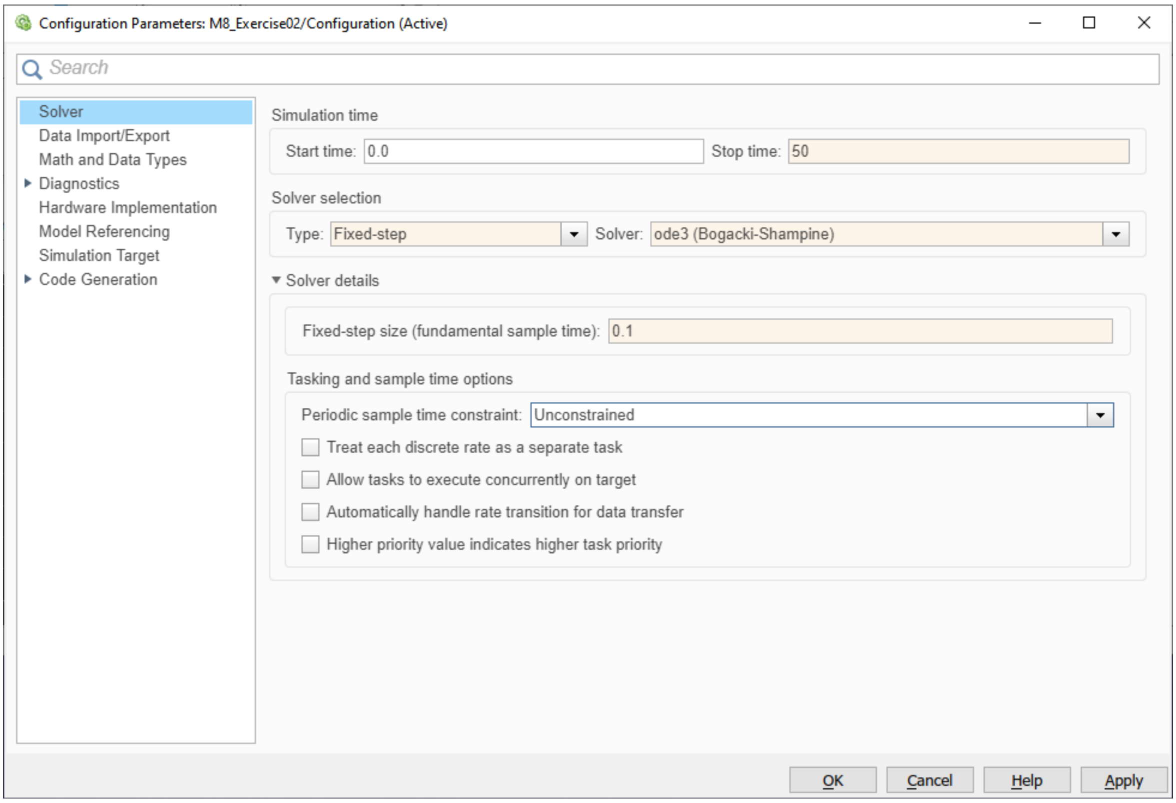



**Figure 29** Settings (Pulse generator, sinewave generator and Simulation parameters)

### 2.3.3 Hands-on Exercise 3: Simulation of a temperature measuring system

**Statement of Problem:** Temperature measurement

A PC-based temperature measuring system is shown in **Fig. 30** in which the signal is processed as: 0-100ºC > mV > mA (4-20 mA) > SC-2345: 1-5 [V]. The voltage signal (1-5V) is acquired by the DAQ (and software).

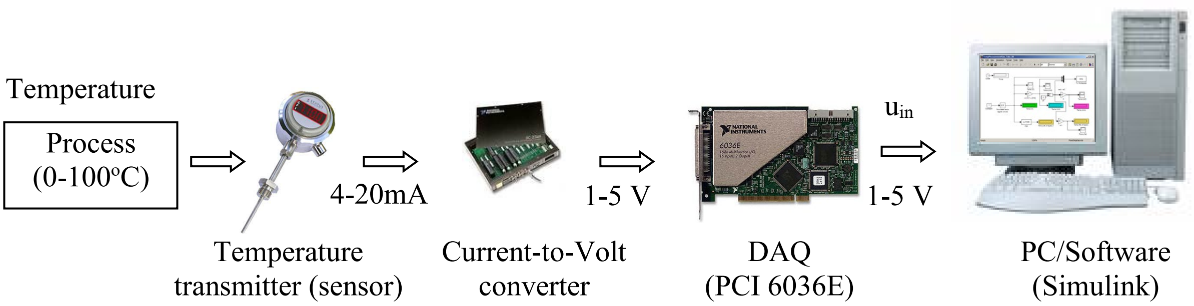



**Figure 30** PC-based temperature measuring system

Make a Simulink model to perform the following signal processing (conditioning) and to display the temperature signals in V, mA and ºC:

- Temperature in V (uin: 1-5 V) – simulated by a signal generator block
- Temperature in mA (u: 4-20 mA); and
- Temperature in ºC (y: 0-100ºC).

**SOLUTION**

The signal processing in PC/software will be:

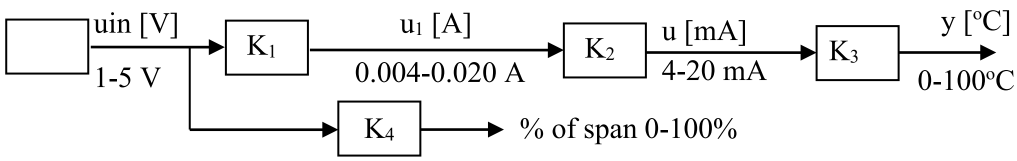



**Figure 31** Block diagram for the temperature transmitter

Therefore we have the following algorithm
Ohm's law:

$$u_1\ [A] = \frac{uin}{R} = uin \times K_1 \quad (R = 250\ \Omega,\ \text{thus } K_1 = 1/R = 1/250)$$

Convert A to mA:

$$u = u_1 \times K_2 \quad (K_2 = 1000)$$

Convert mA to °C:

$$K_3 = \frac{\Delta y}{\Delta u} = \frac{y_{max} - y_{min}}{u_{max} - u_{min}} = \frac{100^\circ C - 0^\circ C}{20mA - 4mA} = \frac{100^\circ C}{16mA} = 6.25\frac{^\circ C}{mA}$$

$$K_3 = \frac{\Delta y}{\Delta u} = \frac{y - y_{min}}{u - u_{min}} \Rightarrow y = K_3(u - u_{min}),\ u0 = u_{min} = 4\ \text{mA, thus: } y = K_3(u - u_0)$$

Convert V to % of span:

$$\%\ \text{of span} = uin_{max} - uin_{min} = 5V - 1V = 4V$$

$$K_4 = \frac{100\%}{4} = 25[\%]$$

or

$$\%\ \text{of span} = K_4\left(uin_{max} - uin_{min}\right) = 25\%(5V - 1V)$$

Block diagram algorithm is shown in **Fig. 32**.

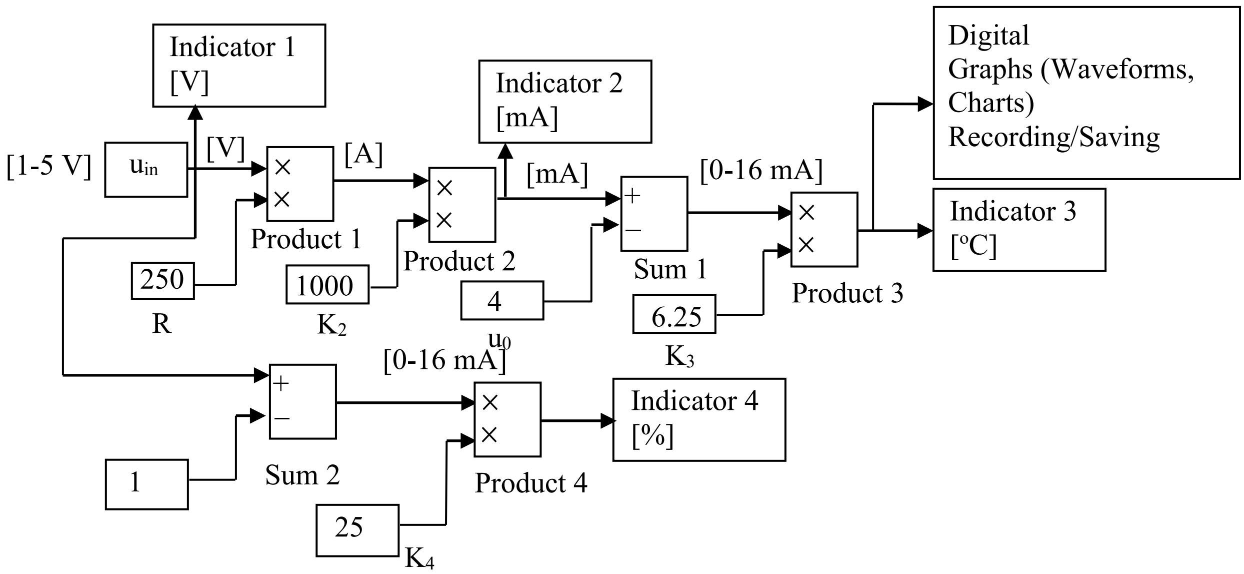



**Figure 32** Block diagram for temperature measurement

**Programming with Simulink (*sample codes were made with an older version*)**

- Open a new Simulink model
- Save as "M6Exericse3.mdl"
- Refer to the block diagram algorithm in **Fig. 32** select all necessary blocks from the Simulink Library Browser as in **Fig. 33**.
- Wire the blocks and set system parameters as in **Fig. 34** (demonstrate each step).

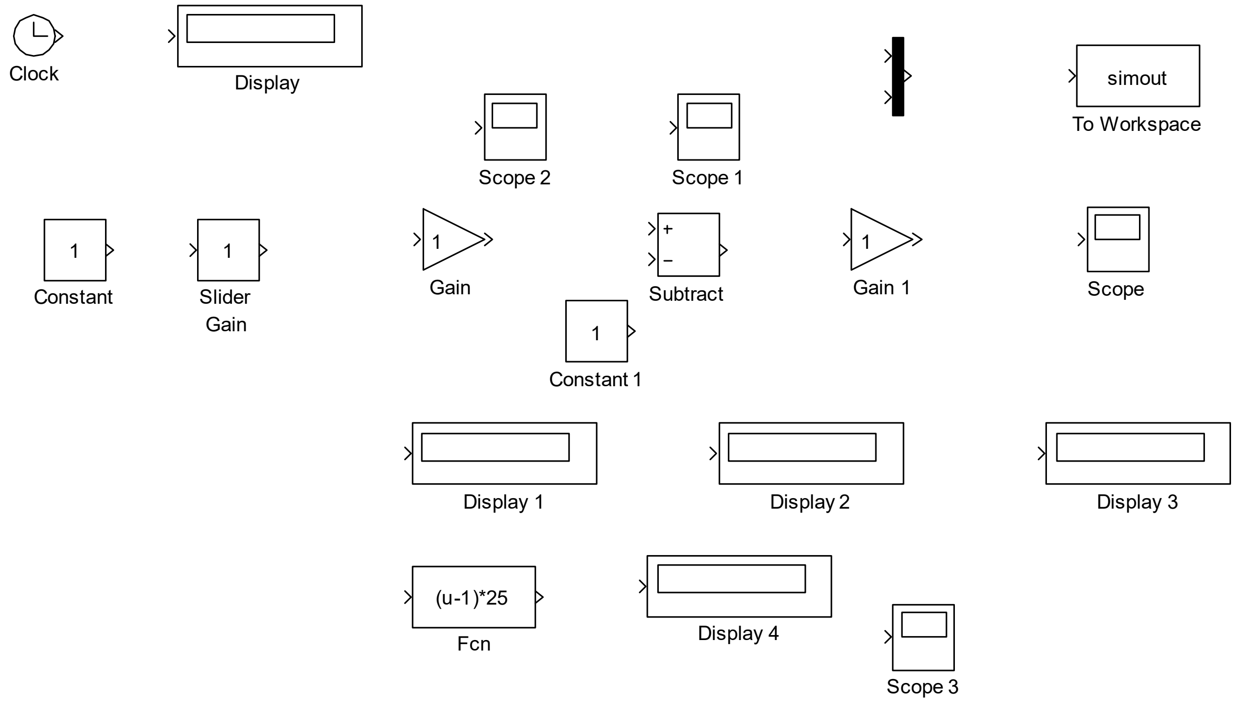



**Figure 33** Necessary blocks for block diagram algorithm in **Fig. 32**

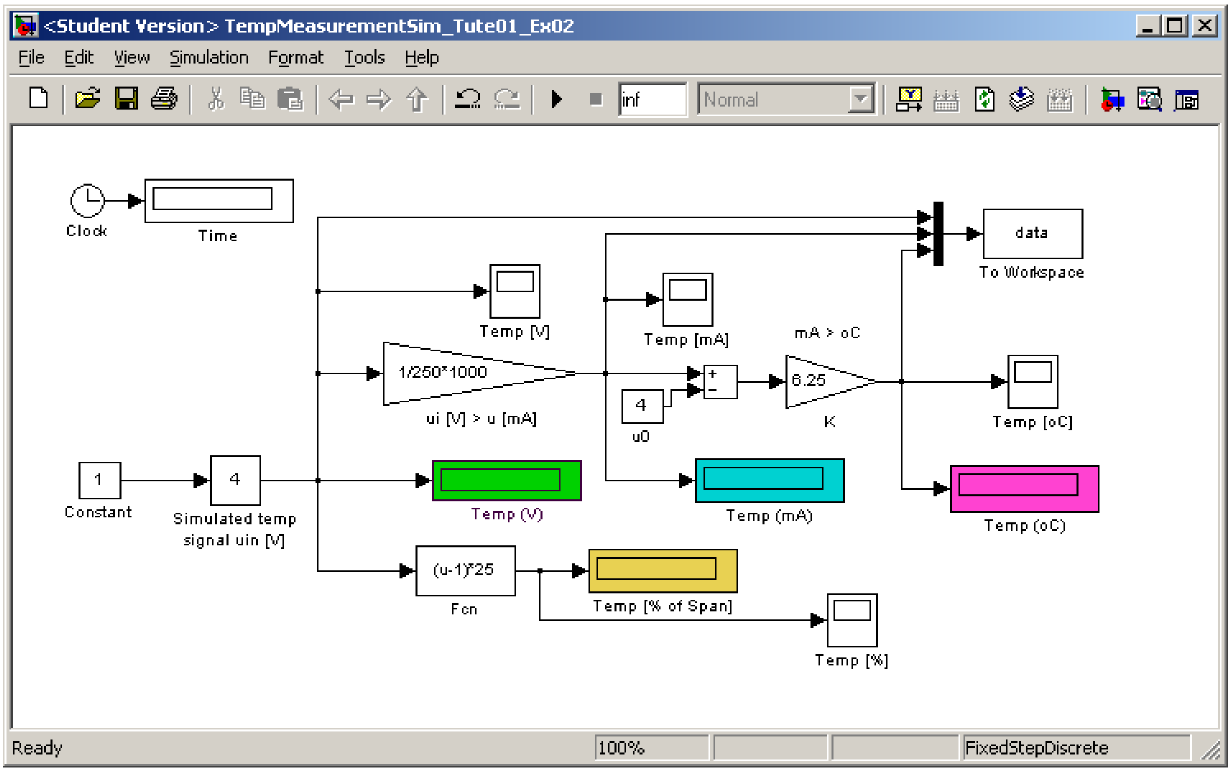



**Figure 33** System parameters

- Configure simulation parameters as in the following figure:

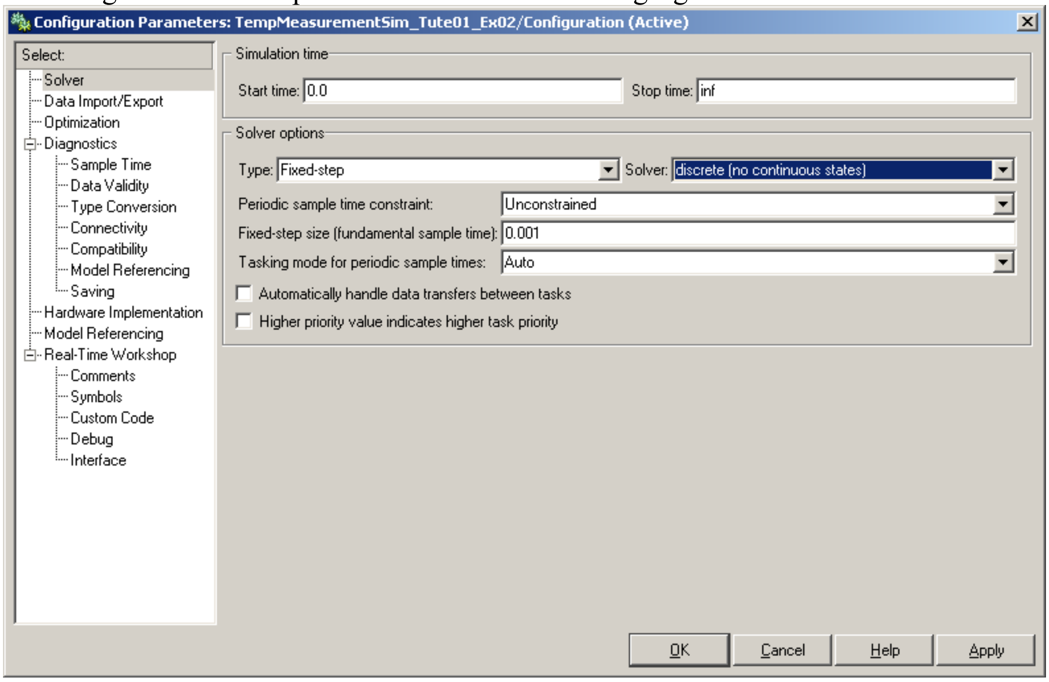
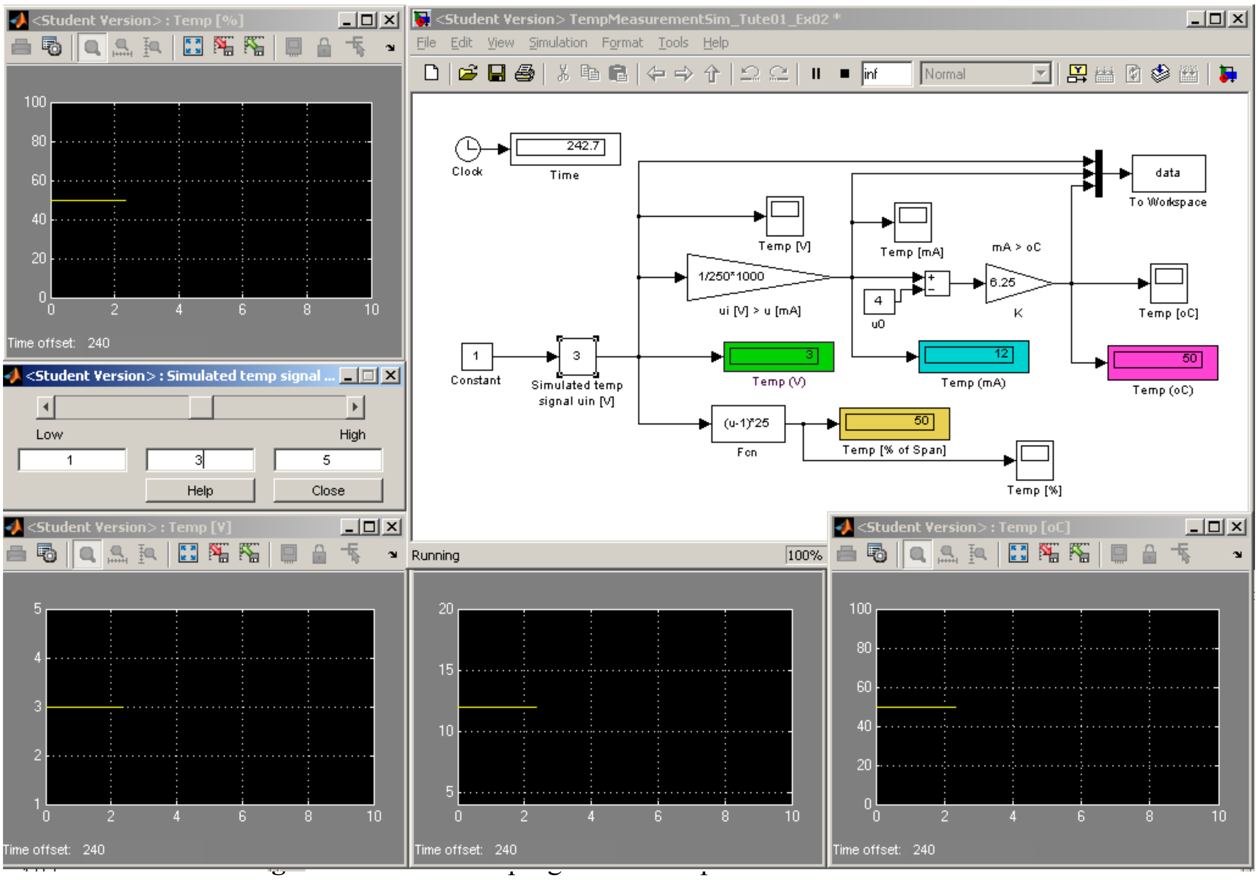



**Figure 34** Simulation parameters

- Save the program.
- Run the Simulink program… The resulting program looks like…




- Luckily so far there is no error!

You can test the functionality of the program by changing some values of uni below:

uin = 1 V $\rightarrow$ 4 mA $\rightarrow$ 0ºC $\rightarrow$ 0%
uin = 2 V $\rightarrow$ 8 mA $\rightarrow$ 25ºC $\rightarrow$ 25%
uin = 3 V $\rightarrow$ 12 mA $\rightarrow$ 50ºC $\rightarrow$ 50%
uin = 4 V $\rightarrow$ 16 mA $\rightarrow$ 75ºC $\rightarrow$ 75%
uin = 5 V $\rightarrow$ 20 mA $\rightarrow$ 100ºC $\rightarrow$ 100%

**3. Conclusions**

At this point the following learning objectives have been met:

- Create and run a Simulink model
- Compute value of a function and display computed results with Simumulink
- Use blocks in Simulink Library Browser
- Configure system parameters
- Configure simulation parameters
- Store simulated data to Workspace and save Workspace to files

**Follow-up Exercises**

**1. Level Measurement System**: A PC-based level measuring system is shown in **Fig. 30** in which the signal is processed as: 0-400 mm > mA (4-20 mA) > SC-2345: 1-5 [V] – DAQ : 1-5V (software). Make a Simulink model to simulate the level measuring system (you can modify the above simulation program).

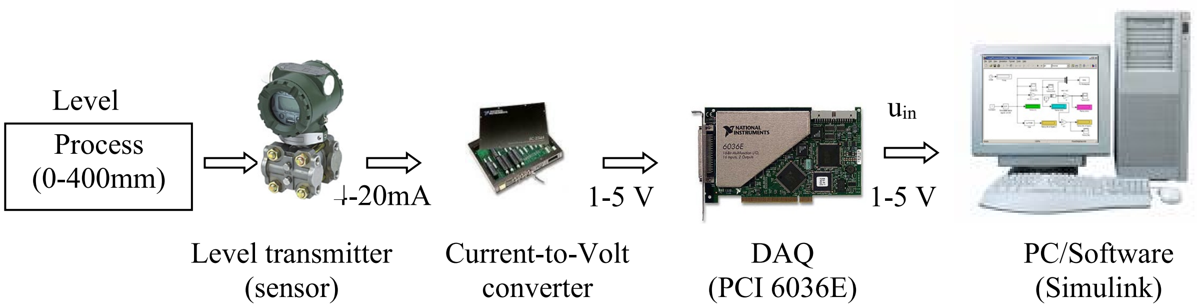



**Figure 36** PC-based level measuring system

**2. RTD Temperature Transmitter:** The following figure shows an RTD-100 temperature transmitter that consists of an RTD sensor, Wheatstone bridge and amplifier. The output voltage ($V_{out}$) is in the range of 0 to 10 V. The RTD is used to measure the temperature in the range of 0ºC to 100ºC. The RTD has a resistance of $R_x$ = 100 $\Omega$ (at 0ºC), $R_1 = R_2 = 100\,\Omega$ and $R_3$ is a variable resistor used to null the bridge at 0ºC ($R_3$ = 100), V = 10 V. $R_4$ = 1 k$\Omega$. Make a simulation program for the RTD-100 transmitter and select value of RF such that the output voltage is in the range of 0-10 V.

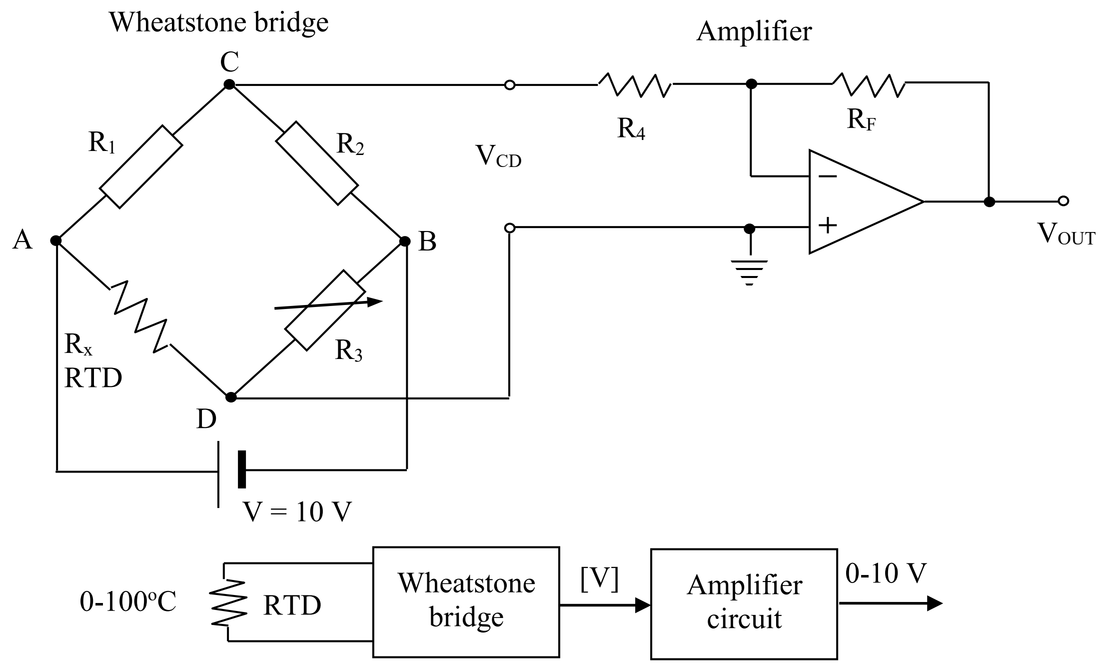



**Figure 37** A temperature measuring system

Make a Simulink model for the RTD temperature measuing system.

**Hints**

RTD sensor: The resistance is dependant on tempeature (coefficient = 0.00385/ºC)

$$R = R_0[1 + \alpha(T)]$$

Wheatstone bridge circuit:

$$V_{CD} = \frac{R_3R_2 - R_xR_4}{(R_x + R_3)(R_2 + R_4)}$$

Amplifier circuit:

$$V_{out} = -\frac{R_F}{R_4}V_{CD}$$

[Sample Simulink model]

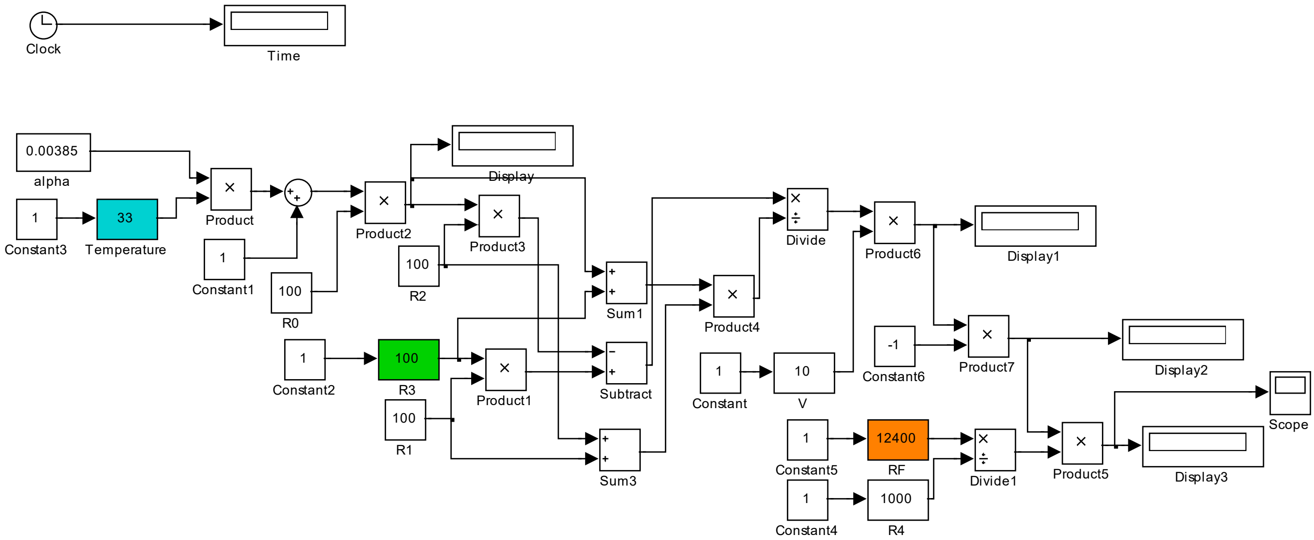



**Figure 38** Sample program for the RTD temperature measuring system