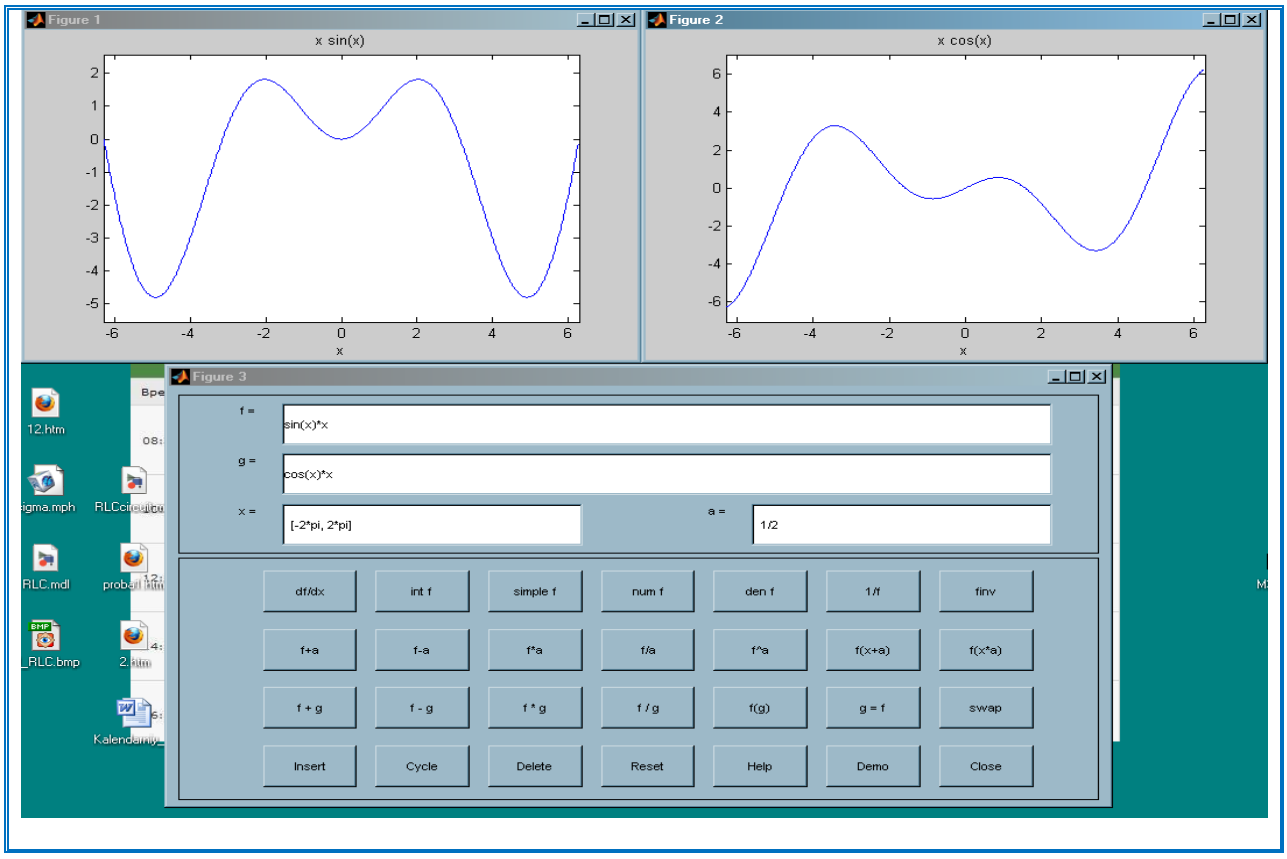
LECTURE 6

MATLAB

MATLAB (**mat**rix **lab**oratory) is a numerical computing environment and fourth-generation programming language. Developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, and Fortran.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing capabilities. An additional package, Simulink, adds graphical multi-domain simulation and Model-Based Design for dynamic and embedded systems.

In 2004, MATLAB had around one million users across industry and academia. MATLAB users come from various backgrounds of engineering, science, and economics. MATLAB is widely used in academic and research institutions as well as industrial enterprises. Let us start by considering function calculator- **funtool**



Description: funtool is a visual function calculator that manipulates and displays functions of one variable. At the click of a button, for example, funtool draws a graph representing the sum, product, difference, or ratio of two functions that you specify. funtool includes a function memory that allows you to store functions for later retrieval.

At startup, **funtool** displays graphs of a pair of functions, f(x) = x and g(x) = 1. The graphs plot the functions over the domain [-2*pi, 2*pi]. funtool also displays a control panel that lets you save, retrieve, redefine, combine, and transform f and g.

Control Buttons: The bottom part of the control panel contains an array of buttons that transform f and perform other operations. The first row of control buttons replaces f with various transformations of f.

df/dx Derivative of f

int f Integral of f

simple f Simplified form of f, if possible

num f Numerator of f

den f Denominator of f

1/f Reciprocal of f

finv Inverse of f

The operators **intf** and **finv** may fail if the corresponding symbolic expressions do not exist in closed form.

The second row of buttons translates and scales f and the domain of f by a constant factor. To specify the factor, enter its value in the field labeled **a=** on the calculator control panel. The operations are

f+a Replaces f(x) by f(x) + a.

f-a Replaces f(x) by f(x) - a.

f*a Replaces f(x) by f(x) * a.

f/a Replaces f(x) by f(x) / a.

f^a Replaces f(x) by f(x) ^ a.

f(x+a) Replaces f(x) by f(x + a).

f(x*a) Replaces f(x) by f(x * a).

The first four buttons of the third row replace f with a combination of f and g.

f+g Replaces f(x) by f(x) + g(x).

f-g Replaces f(x) by f(x)-g(x).

f*g Replaces f(x) by f(x) * g(x).

f/g Replaces f(x) by f(x) / g(x).

The remaining buttons on the third row interchange f and g.

g=f Replaces g with f.

swap Replaces f with g and g with f.

The first three buttons in the fourth row allow you to store and retrieve functions from the calculator's function memory.

Insert Adds f to the end of the list of stored functions.

Cycle Replaces f with the next item on the function list.

Delete Deletes f from the list of stored functions.

The other four buttons on the fourth row perform miscellaneous functions:

Reset Resets the calculator to its initial state.

Help Displays the online help for the calculator.

Close Closes the calculator's windows.

taylortool- Taylor series calculator

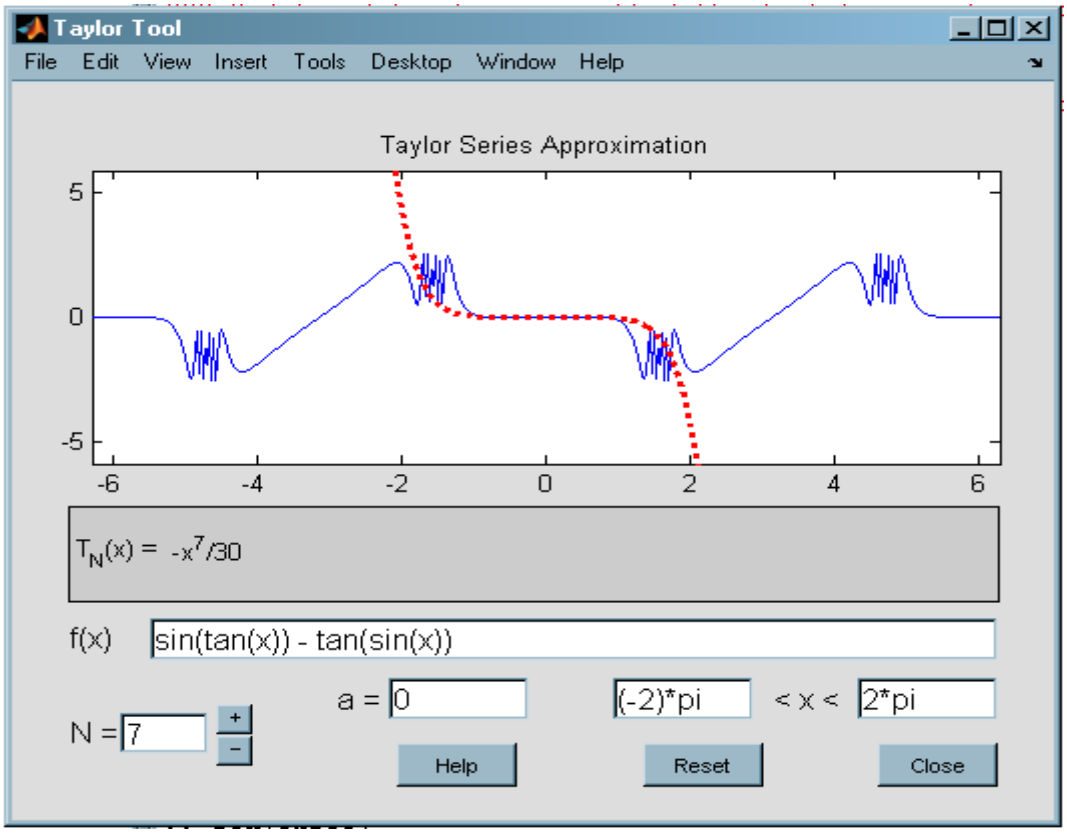
Syntax taylortool

taylortool('f')

Description: taylortool initiates a GUI that graphs a function against the Nth partial sum of its Taylor series about a base point x = a. The default function, value of N, base point, and interval of computation for taylortool are f = x*cos(x), N = 7, a = 0, and [-2*pi,2*pi], respectively.

taylortool('f') initiates the GUI for the given expression f.

Examples: taylortool('sin(tan(x)) - tan(sin(x))')



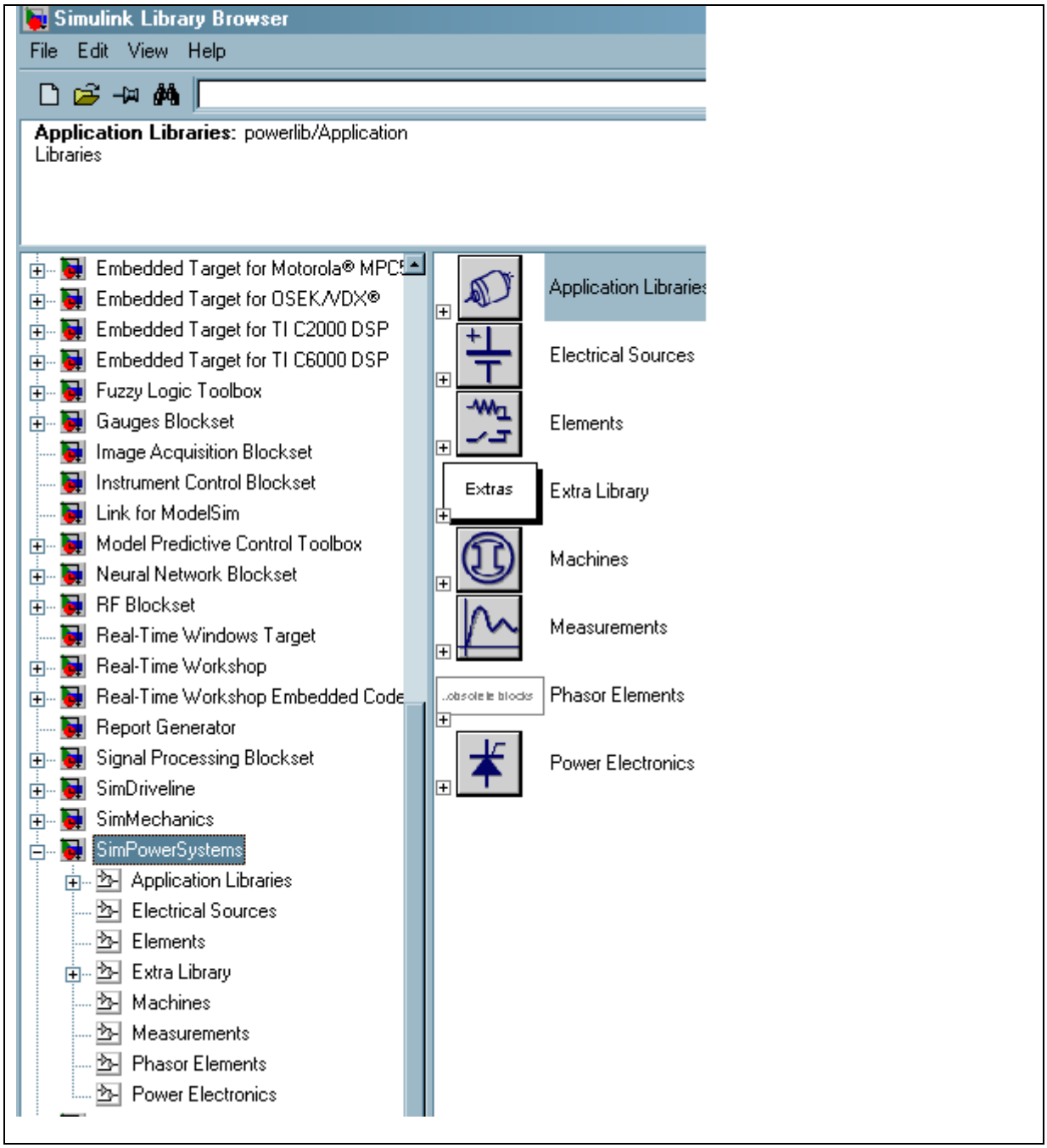
Simulink

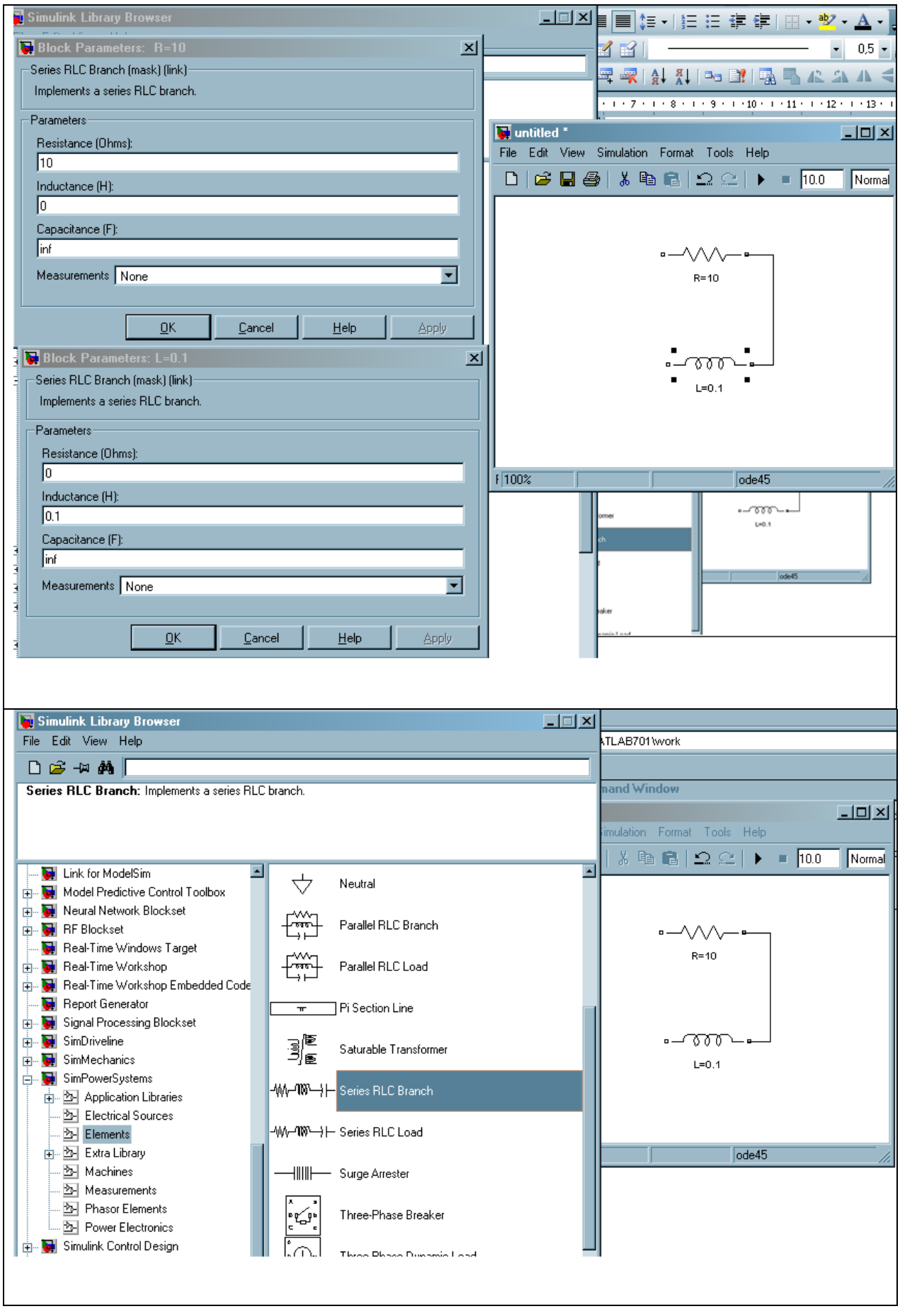
Simulink is a block diagram environment for multidomain simulation and Model-Based Design. It supports system-level design, simulation, automatic code generation, and continuous test and verification of embedded systems. Simulink provides a graphical editor, customizable block libraries, and solvers for modeling and simulating dynamic systems. It is integrated with MATLAB®, enabling you to incorporate MATLAB algorithms into models and export simulation results to MATLAB for further analysis

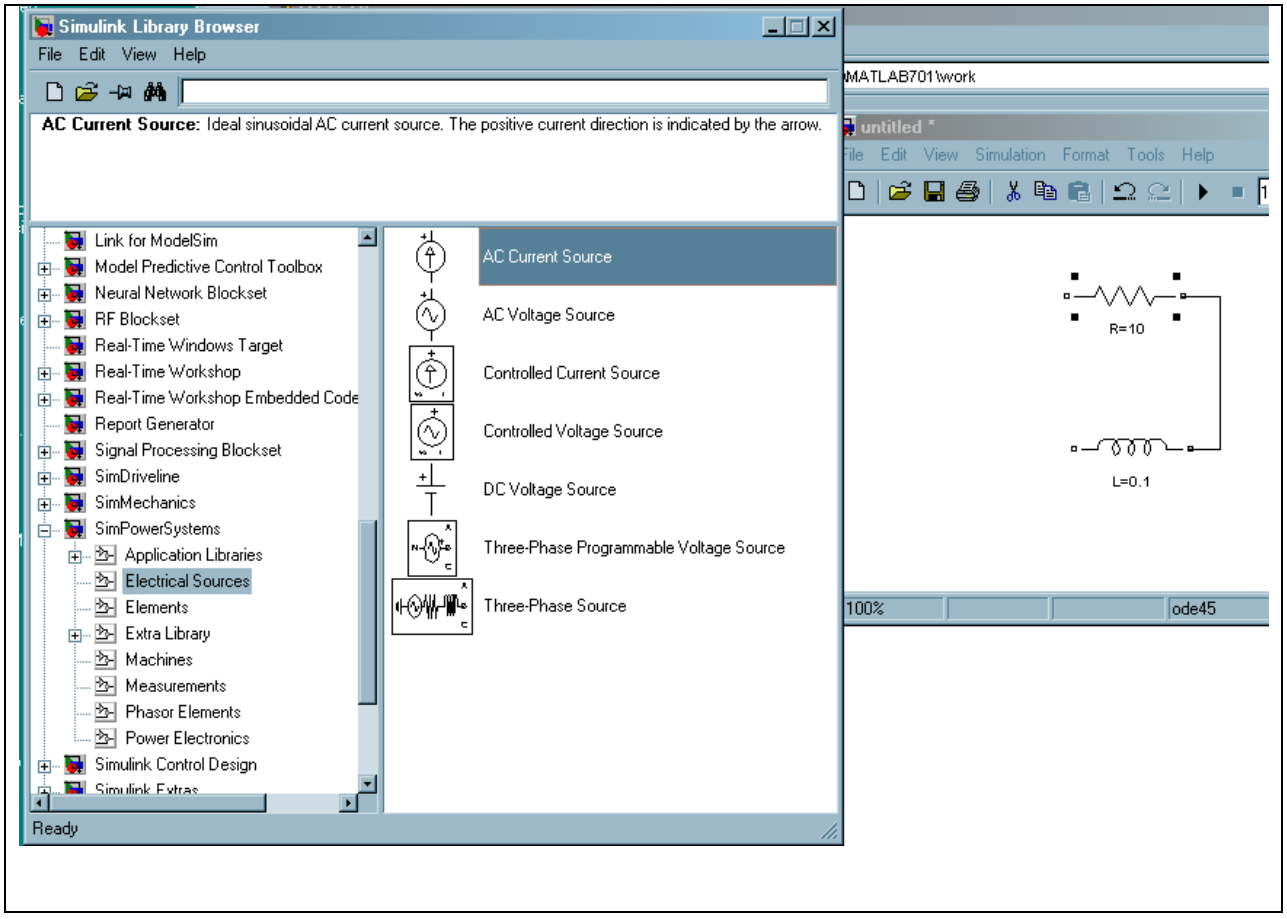
Model and simulate electrical power systems

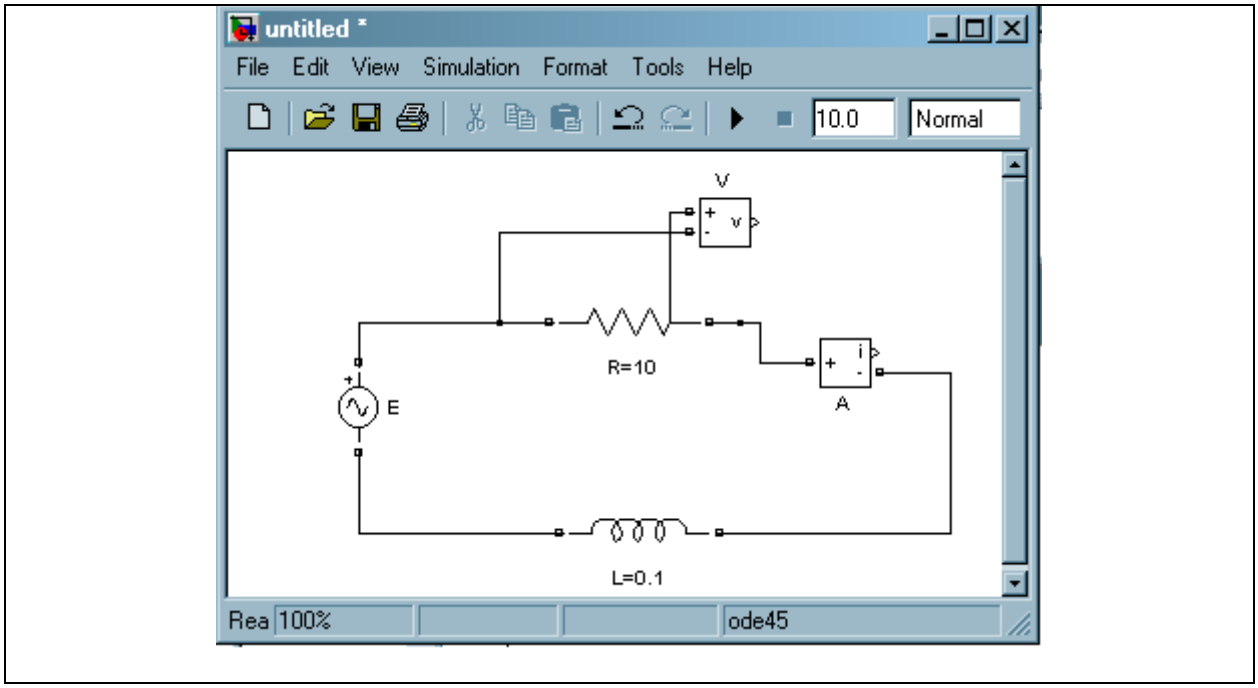
SimPowerSystems provides component libraries for modeling and simulating electrical power systems. It includes models of three-phase machines, electric drives, flexible AC transmission systems (FACTS), and wind power generators. Abstracted models of power electronics components are also included, enabling you to assess the impact of switching events on system-level behavior. You can use these components to model the generation, transmission, distribution, and

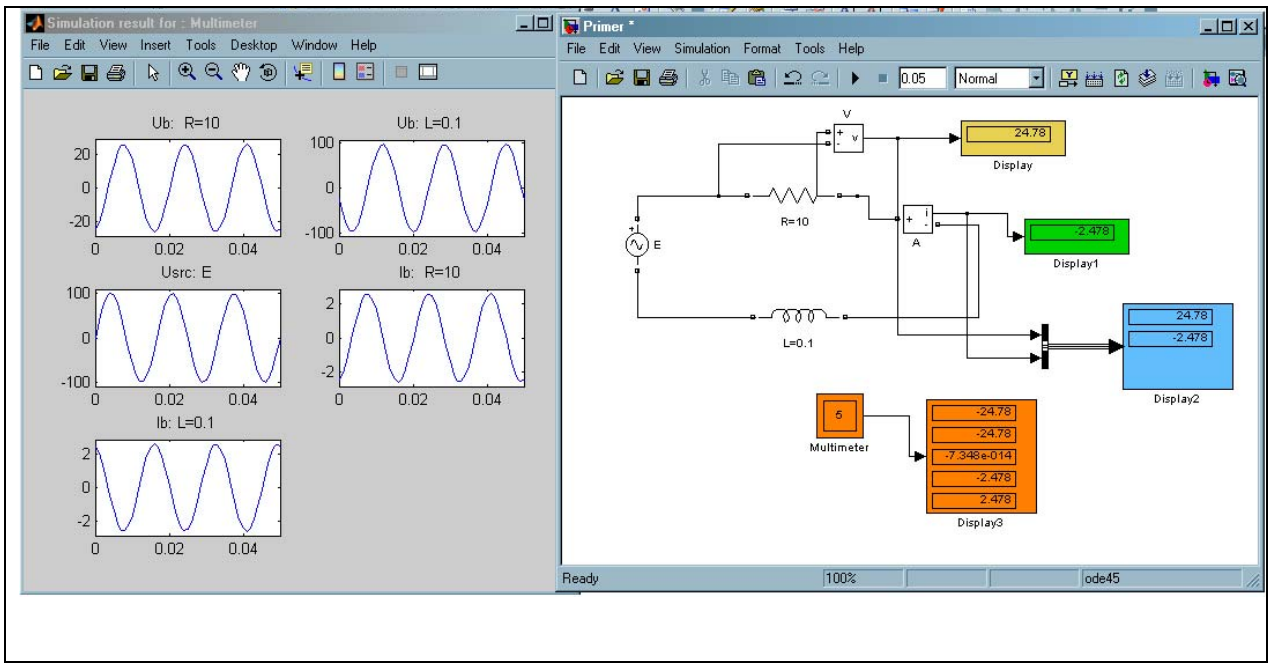
consumption of electrical power. Harmonic analysis, calculation of total harmonic distortion (THD), load flow, and other key electrical power system analyses are automated. SimPowerSystems models can be discretized to speed up simulations and configured for phasor simulation, which helps you determine the transient stability of electrical power systems. Here are given same examples.

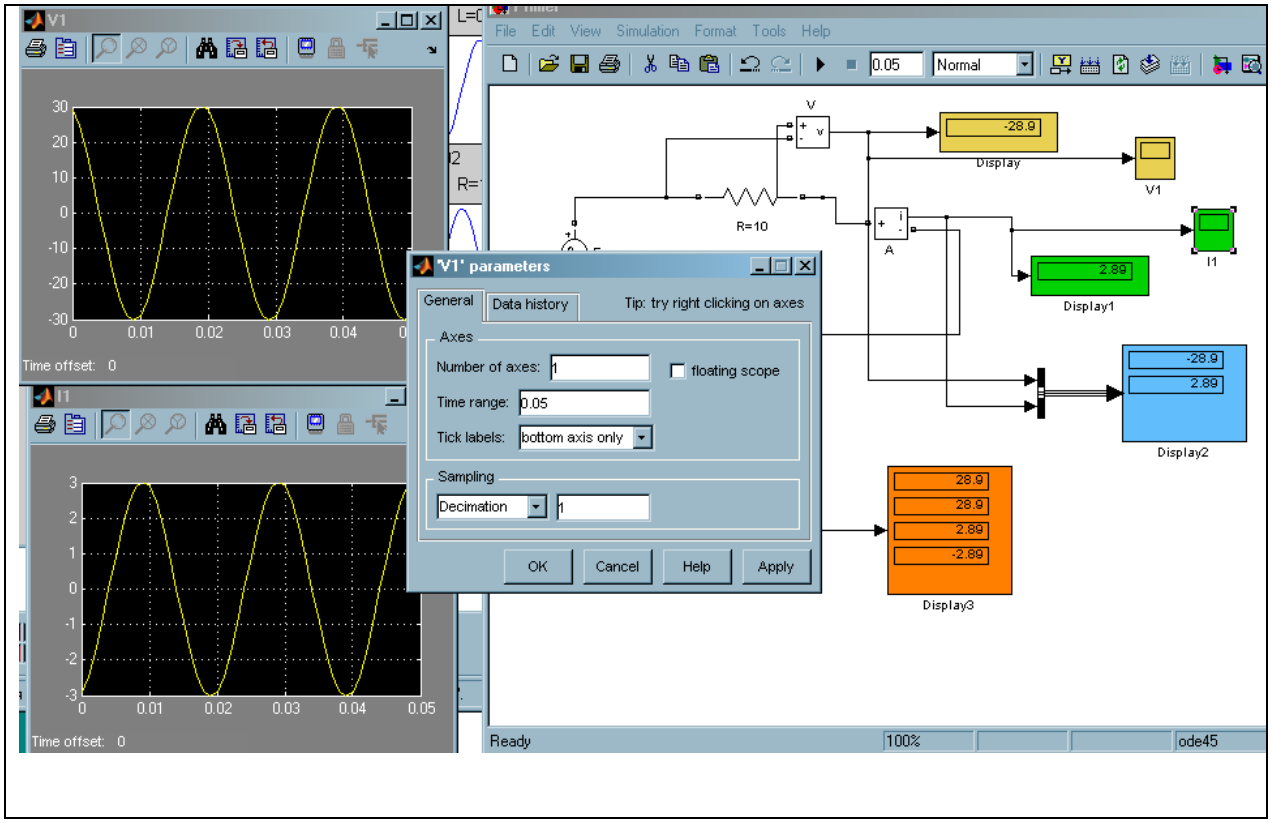


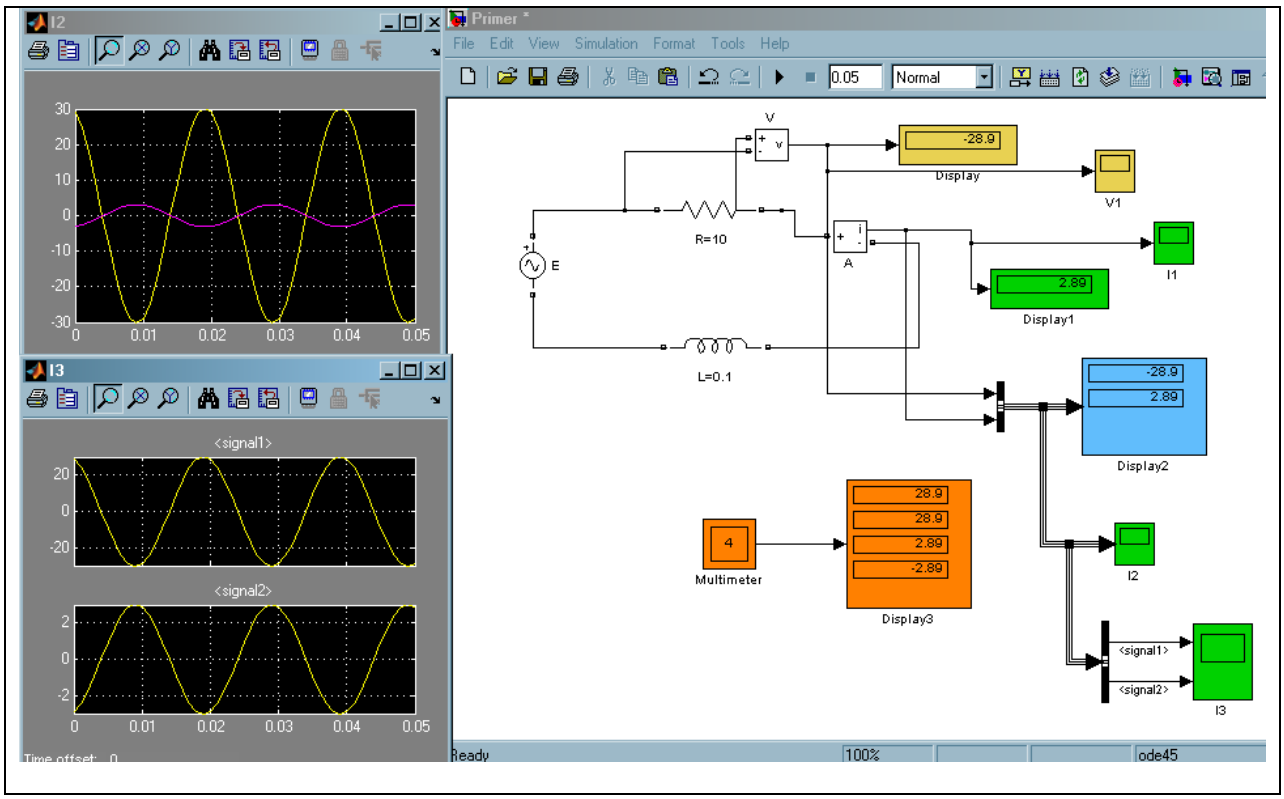

Simulink Library Browser
Block Parameters: R=10
Series RLC Branch (mask) (link)
Implements a series RLC branch.
Parameters
Resistance (Ohms):
10
Inductance (H):
0
Capacitance (F):
inf
Measurements None
OK
Cancel
Help
Apply
Block Parameters: L=0.1
Series RLC Branch (mask) (link)
Implements a series RLC branch.
Parameters
Resistance (Ohms):
0
Inductance (H):
0.1
Capacitance (F):
inf
Measurements None
OK
Cancel
Help
Apply
untitled *
File Edit View Simulation Format Tools Help
10.0
Normal
R=10
L=0.1
100%
ode45
Simulink Library Browser
File Edit View Help
Series RLC Branch: Implements a series RLC branch.
Link for ModelSim
Model Predictive Control Toolbox
Neural Network Blockset
RF Blockset
Real-Time Windows Target
Real-Time Workshop
Real-Time Workshop Embedded Code
Report Generator
Signal Processing Blockset
SimDriveline
SimMechanics
SimPowerSystems
Application Libraries
Electrical Sources
Elements
Extra Library
Machines
Measurements
Phasor Elements
Power Electronics
Simulink Control Design
Neutral
Parallel RLC Branch
Parallel RLC Load
Pi Section Line
Saturable Transformer
Series RLC Branch
Series RLC Load
Surge Arrester
Three-Phase Breaker
ATLAB701\work
nand Window
imulation Format Tools Help
10.0
Normal
R=10
L=0.1
ode45


Simulink Library Browser
File Edit View Help
AC Current Source: Ideal sinusoidal AC current source. The positive current direction is indicated by the arrow.
Link for ModelSim
Model Predictive Control Toolbox
Neural Network Blockset
RF Blockset
Real-Time Windows Target
Real-Time Workshop
Real-Time Workshop Embedded Code
Report Generator
Signal Processing Blockset
SimDriveline
SimMechanics
SimPowerSystems
Application Libraries
Electrical Sources
Elements
Extra Library
Machines
Measurements
Phasor Elements
Power Electronics
Simulink Control Design
AC Current Source
AC Voltage Source
Controlled Current Source
Controlled Voltage Source
DC Voltage Source
Three-Phase Programmable Voltage Source
Three-Phase Source
Ready
MATLAB701\work
untitled *
File Edit View Simulation Format Tools Help
R=10
L=0.1
100%
ode45

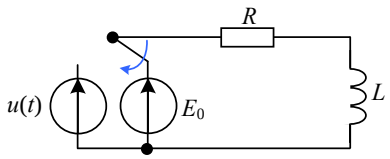

untitled *
File Edit View Simulation Format Tools Help
10.0
Normal
V
R=10
E
A
L=0.1
Rea
100%
ode45


Simulation result for : Multimeter
File Edit View Insert Tools Desktop Window Help
Ub: R=10
Ub: L=0.1
Usrc: E
Ib: R=10
Ib: L=0.1
Primer *
File Edit View Simulation Format Tools Help
0.05
Normal
V
R=10
E
A
L=0.1
Display
Display1
Display2
Display3
Multimeter
24.78
-2.478
-24.78
-7.348e-014
2.478
Ready
100%
ode45


V1
I1
Time offset: 0
'V1' parameters
General
Data history
Tip: try right clicking on axes
Axes
Number of axes: 1
floating scope
Time range: 0.05
Tick labels: bottom axis only
Sampling
Decimation
1
OK
Cancel
Help
Apply
Primer *
File Edit View Simulation Format Tools Help
0.05
Normal
V
R=10
A
-28.9
Display
V1
I1
2.89
Display1
Display2
Display3
28.9
-2.89
Ready
100%
ode45



Transient process problem is one of the main problems at electrical engineering problems solution. Transient processes are described by differential equations. Differential equations can be of the first, second etc order. Inhomogeneous differential equations are solved more often. External power supplies serve as inhomogeneous sources. Dependence can be various: sinusoidal, impulse, sawtooth etc. Thus you should use differential equations. State space method is the most modern calculation method.



Let's consider the example of transient process in first order circuit.

$u(t) = 10\sin(\omega t)B$, $R = 10Ом$, $L = 0{,}1Гн$, $E_0 = 10B$.

Form Kirchhoff equation for the given loop

$$L\frac{di_L}{dt} + i_L R = u(t)$$

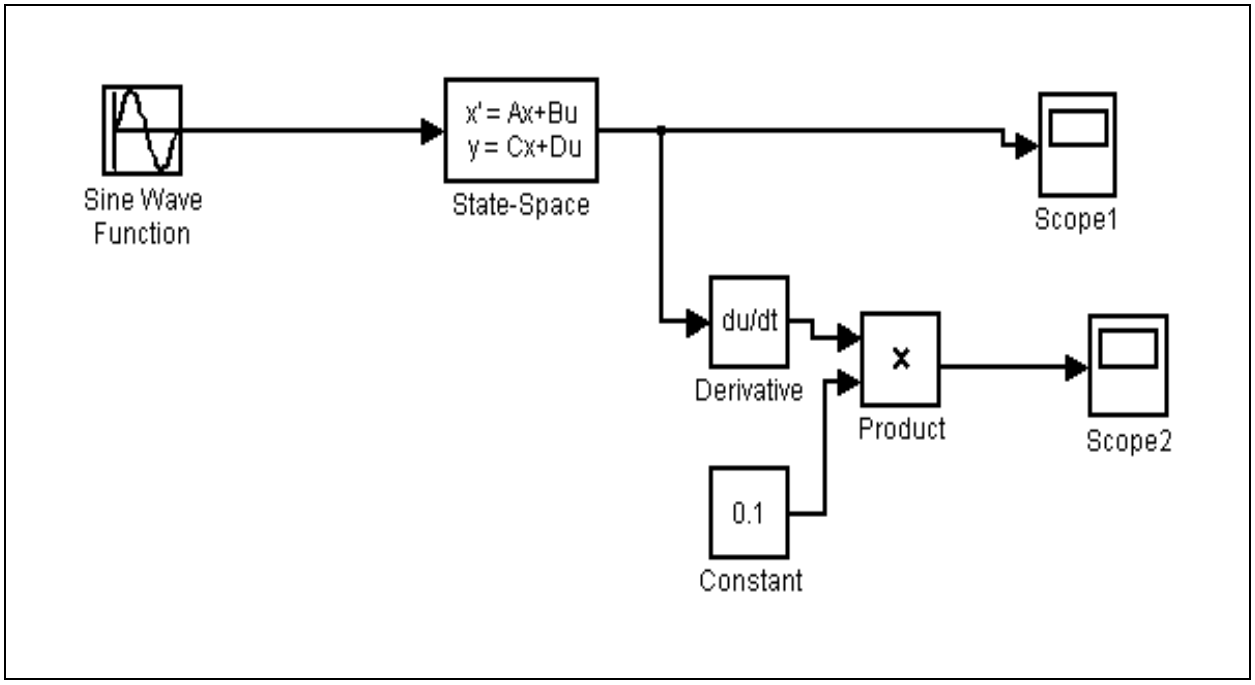
Let's solve it relating to derivative

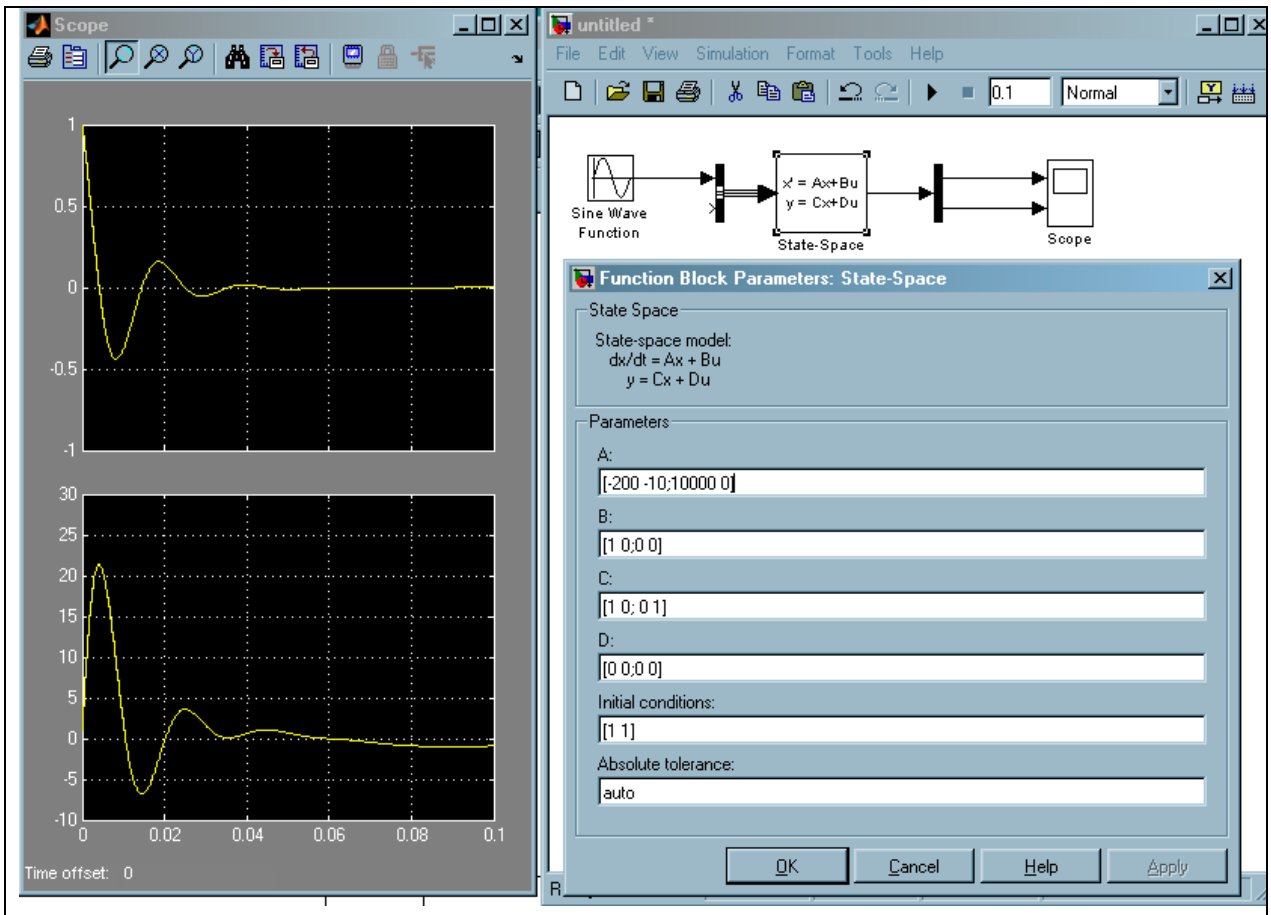
$$\frac{di_L}{dt} = -i_L \frac{R}{L} + \frac{u(t)}{L}$$

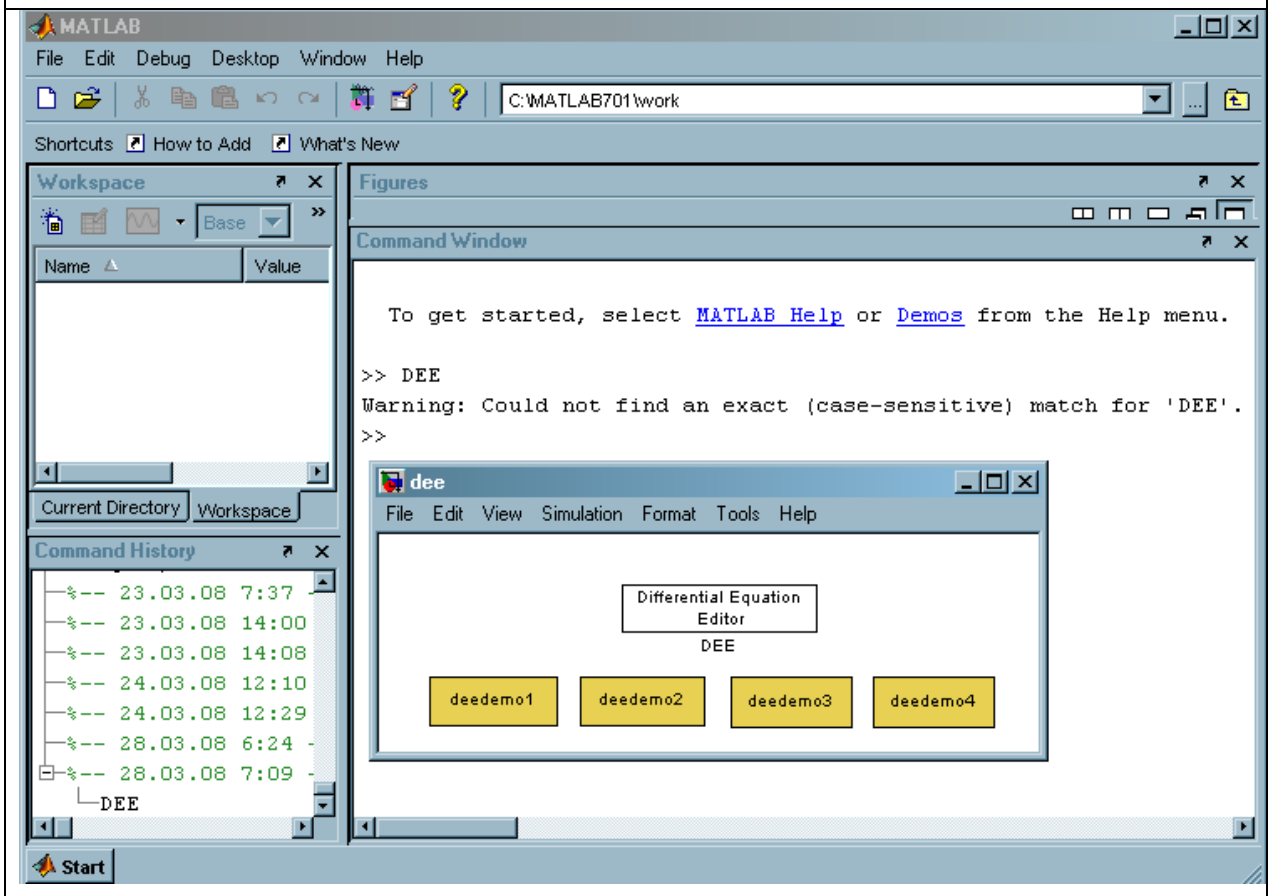
Write the coefficients at unknown function $i(t)$ and at external source of $u(t)$

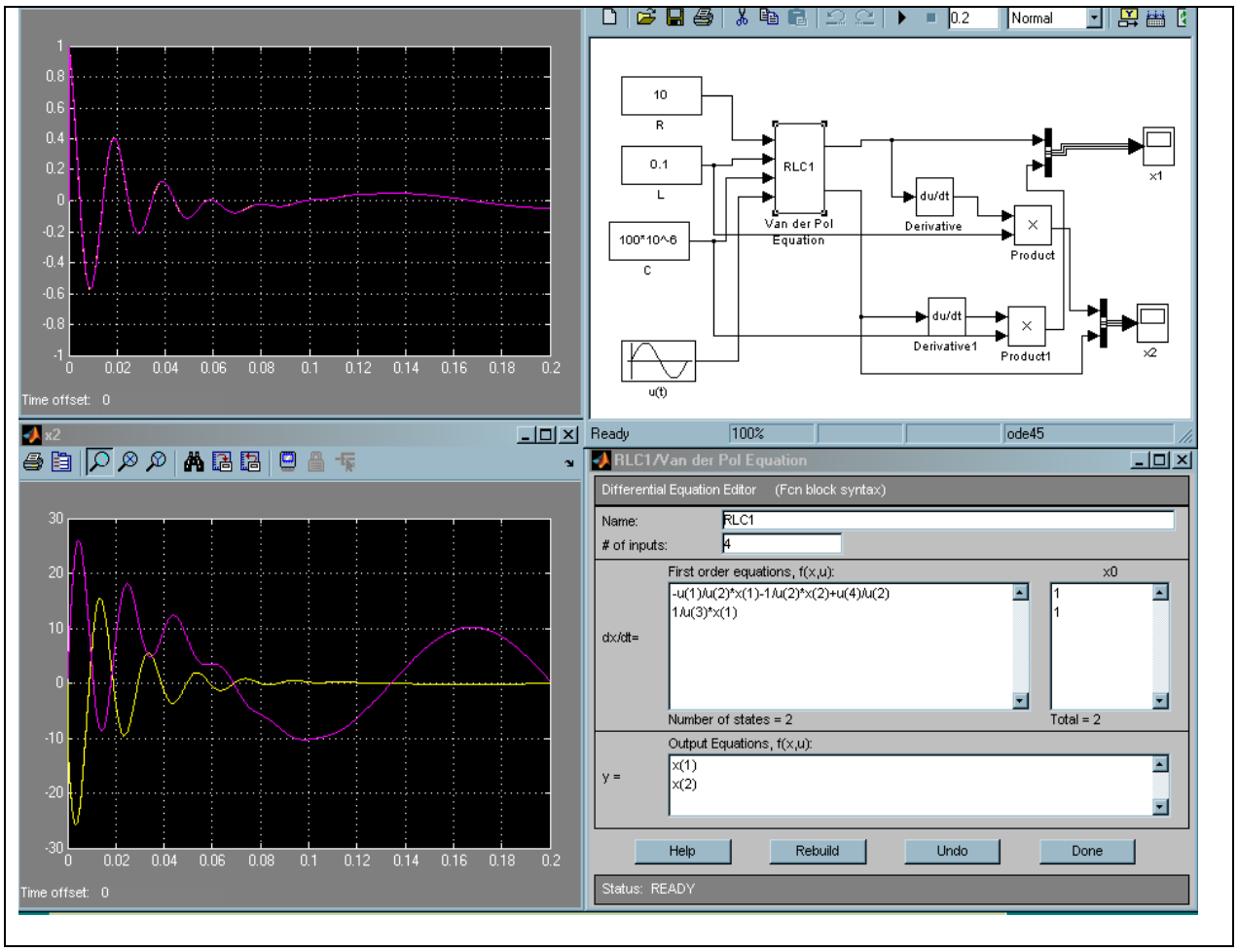
$$\frac{di_L}{dt} = -i_L \frac{R}{L} + \frac{u(t)}{L}$$
$$A = -\frac{R}{L},\ B = \frac{1}{L}$$

Let's work in ***MATLAB***. Solve the problem in ***Simulink.*** In ***Simulink*** browser select ***Continuous,*** then ***State-Space***.




Scope
Time offset: 0
untitled *
File Edit View Simulation Format Tools Help
Normal
Sine Wave Function
x' = Ax+Bu
y = Cx+Du
State-Space
Scope
Function Block Parameters: State-Space
State Space
State-space model:
dx/dt = Ax + Bu
y = Cx + Du
Parameters
A:
[-200 -10;10000 0]
B:
[1 0;0 0]
C:
[1 0; 0 1]
D:
[0 0;0 0]
Initial conditions:
[1 1]
Absolute tolerance:
auto
OK
Cancel
Help
Apply


MATLAB
File Edit Debug Desktop Window Help
C:\MATLAB701\work
Shortcuts How to Add What's New
Workspace
Base
Name Value
Current Directory Workspace
Command History
%-- 23.03.08 7:37
%-- 23.03.08 14:00
%-- 23.03.08 14:08
%-- 24.03.08 12:10
%-- 24.03.08 12:29
%-- 28.03.08 6:24
%-- 28.03.08 7:09
DEE
Figures
Command Window
To get started, select MATLAB Help or Demos from the Help menu.
>> DEE
Warning: Could not find an exact (case-sensitive) match for 'DEE'.
>>
dee
File Edit View Simulation Format Tools Help
Differential Equation Editor
DEE
deedemo1
deedemo2
deedemo3
deedemo4
Start

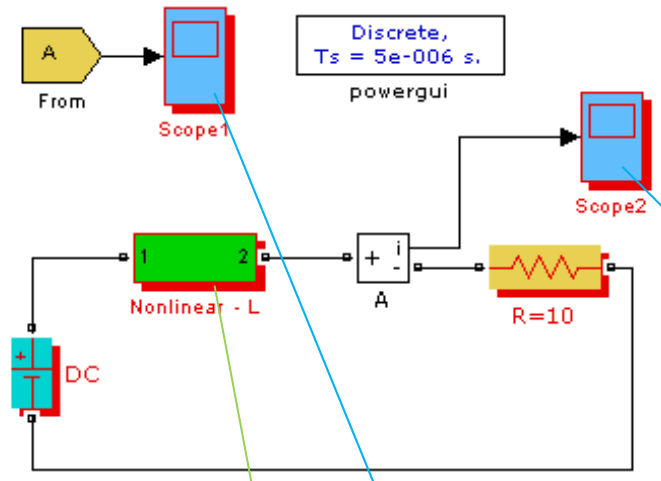


Nonlinear inductance modeling

$$\frac{d\psi}{dt} + iR = E$$

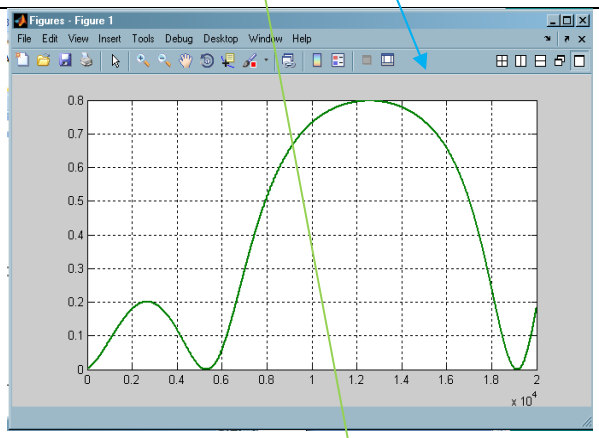
$$i = \frac{\psi}{L(\psi)},\ L(\psi) = \frac{1}{\sin(20\psi) + 1}$$

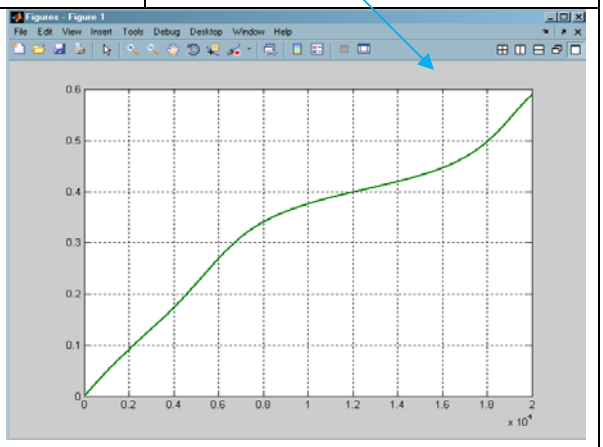
MATLAB

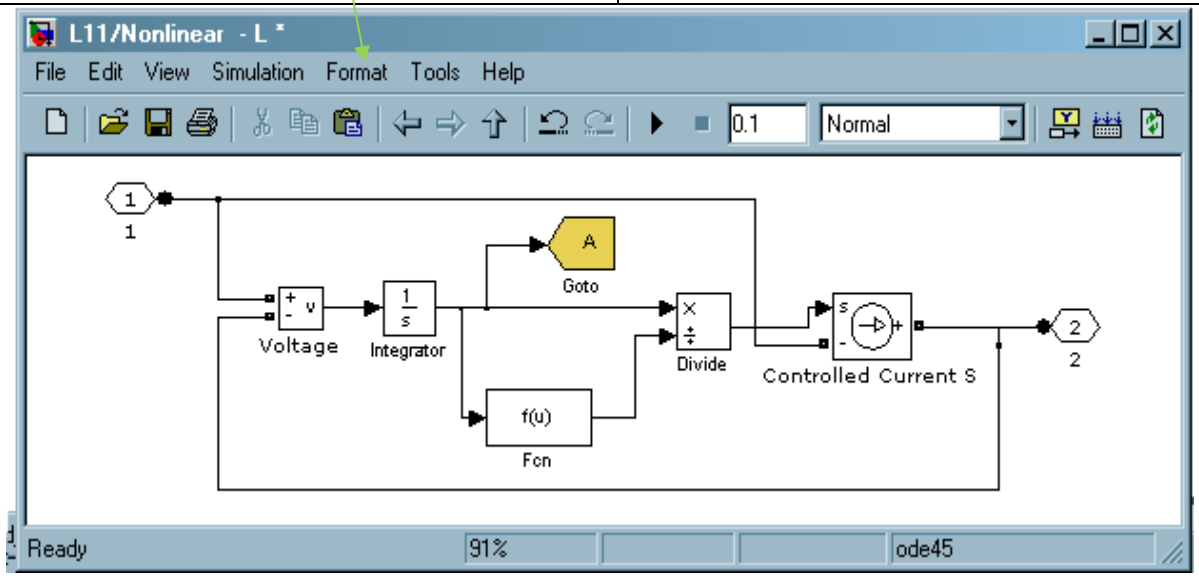


$$\frac{d\psi}{dt} + iR = E$$

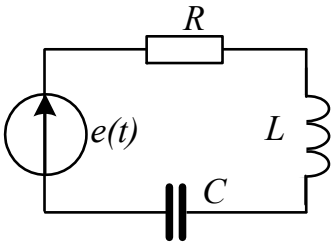
$$i = \frac{\psi}{L(\psi)},\ L(\psi) = \frac{1}{\sin(20\psi)+1}$$







ELECTRIC DIAGRAM MASKING



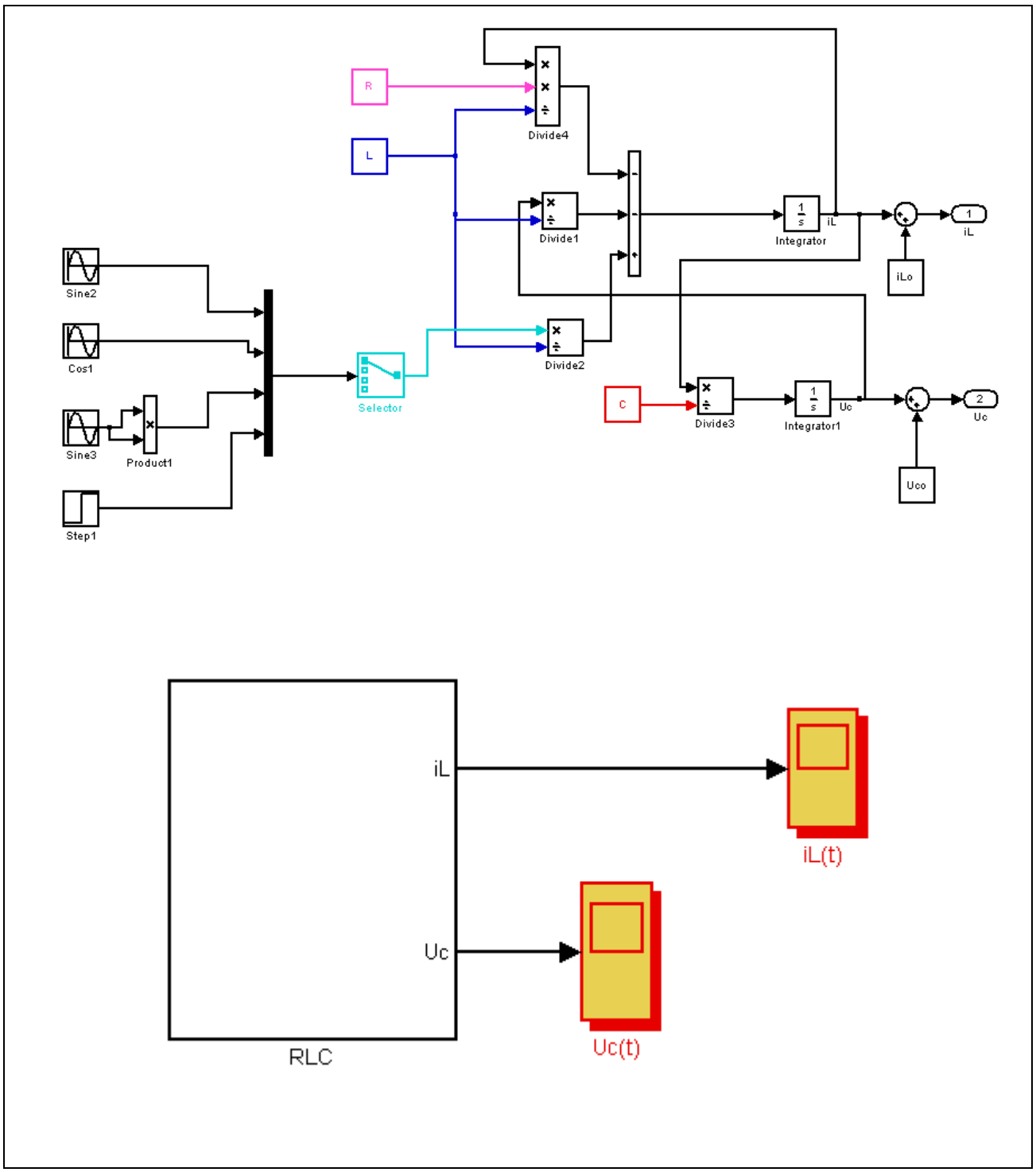
The diagram is given, diagram masking should be done, which help investigate the transient process at different diagram parameters.

Let's write state-space equation

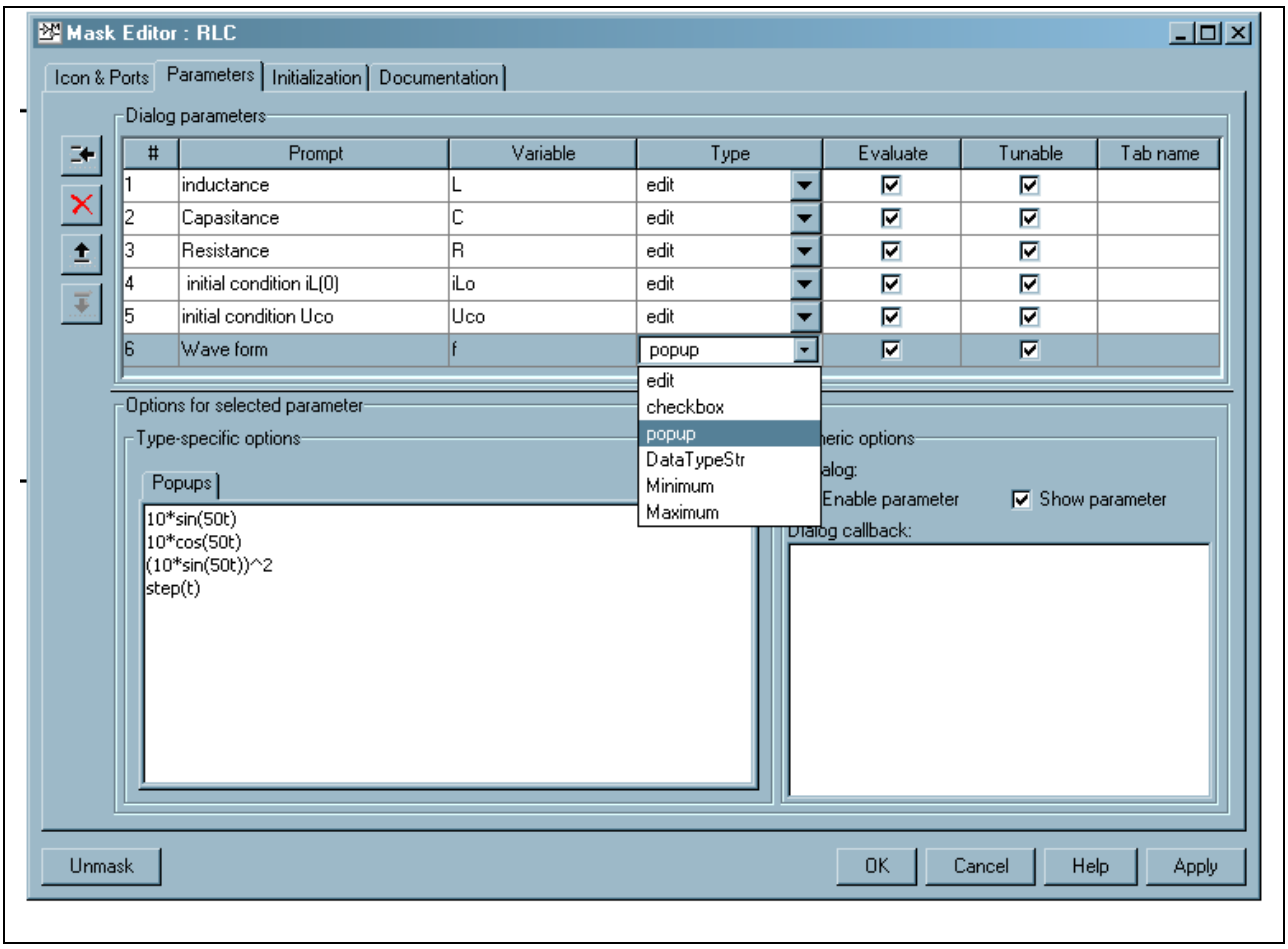
$$\begin{cases} \dfrac{di}{dt} = -\dfrac{R}{L}i - \dfrac{1}{L}u + \dfrac{1}{L}e \\ \dfrac{du}{dt} = \dfrac{1}{C}u \end{cases}$$

$$A = \begin{pmatrix} -\dfrac{R}{L} & \dfrac{1}{L} \\ \dfrac{1}{C} & 0 \end{pmatrix}, \quad B = \begin{pmatrix} \dfrac{E}{L} \\ 0 \end{pmatrix} \rightarrow D(t.x) = Ax + B$$

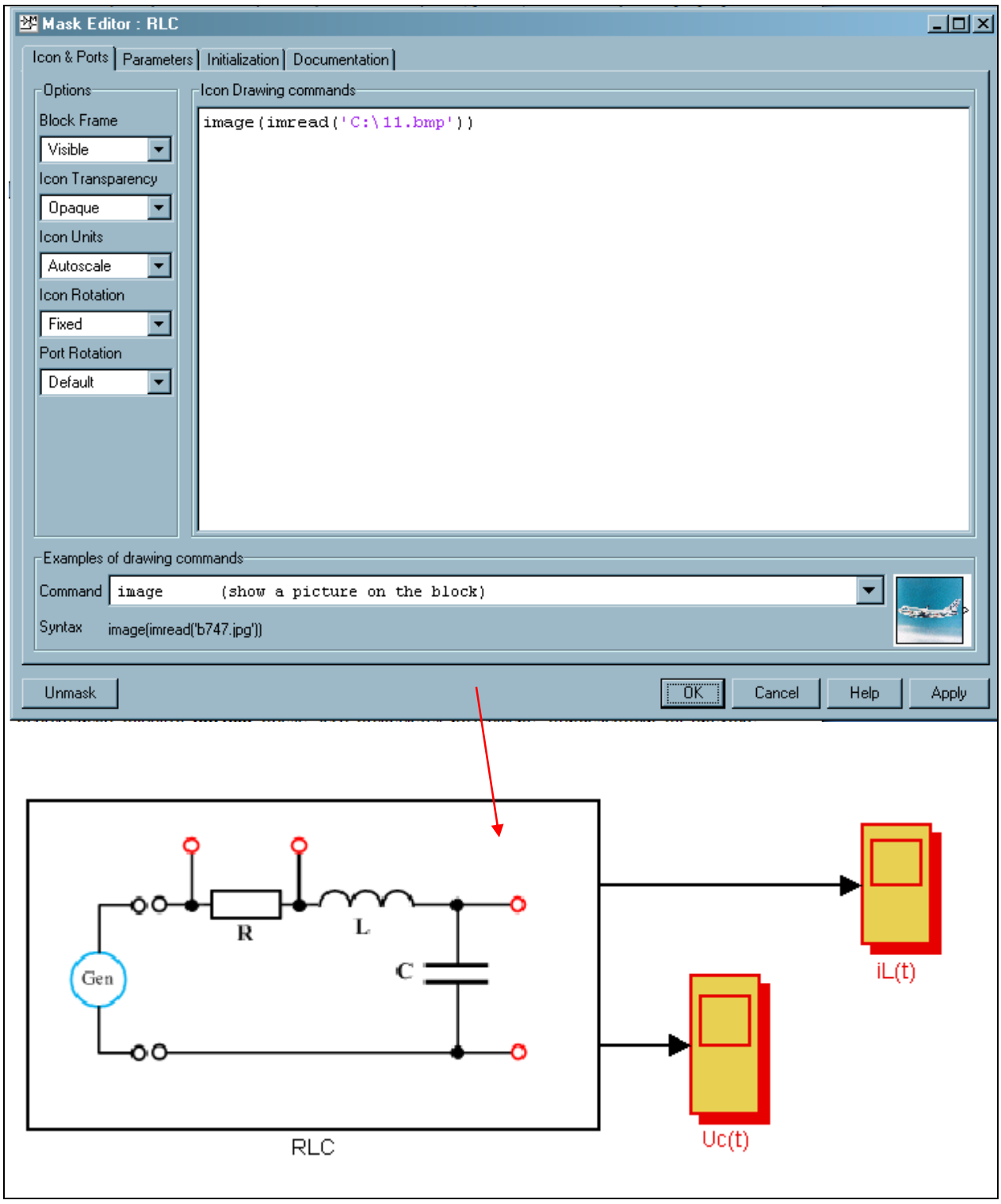
Let's plot the structure diagram: construct the structure diagram given in Figure below in **Simulink.** Select the given structure diagram (**Edit => Select All**) in **Edit** position choose **Create Subsystem-** form the subsystem.



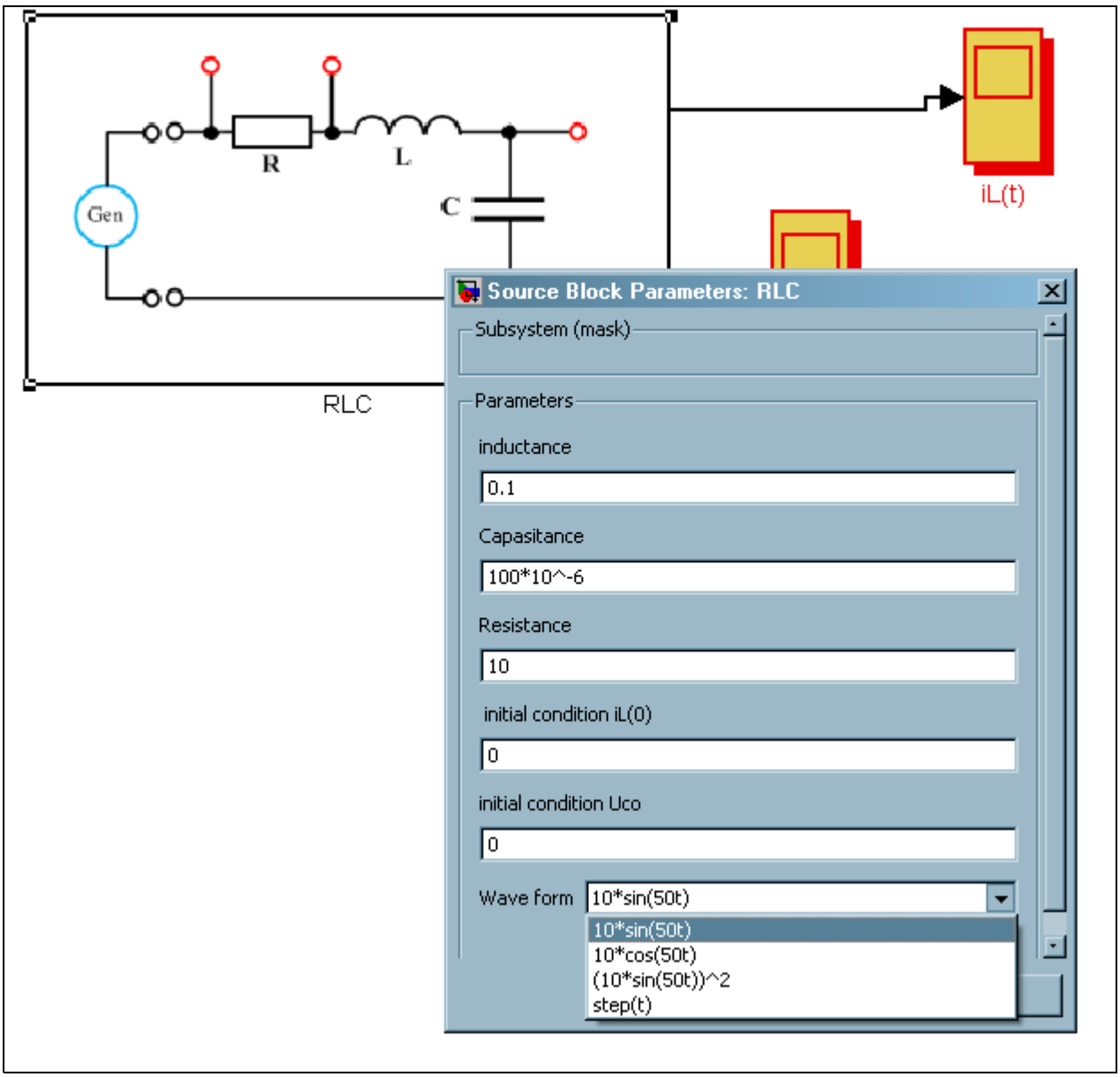
Put the cursor to subsystem area and by left clicking get pop-up window in which choose **Edit Mask** position.

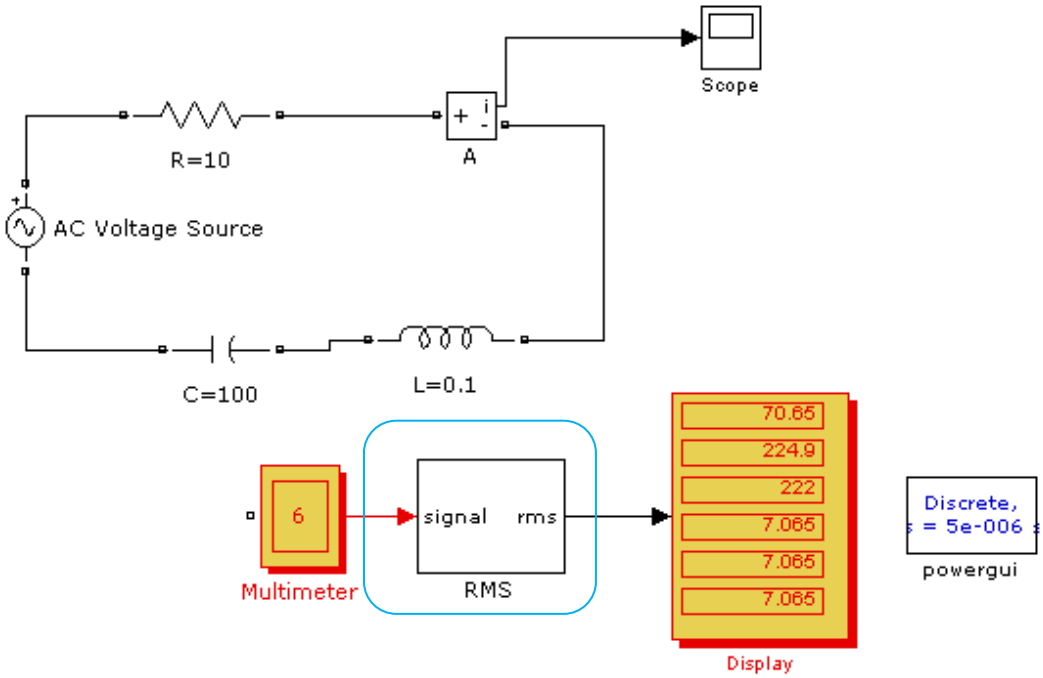


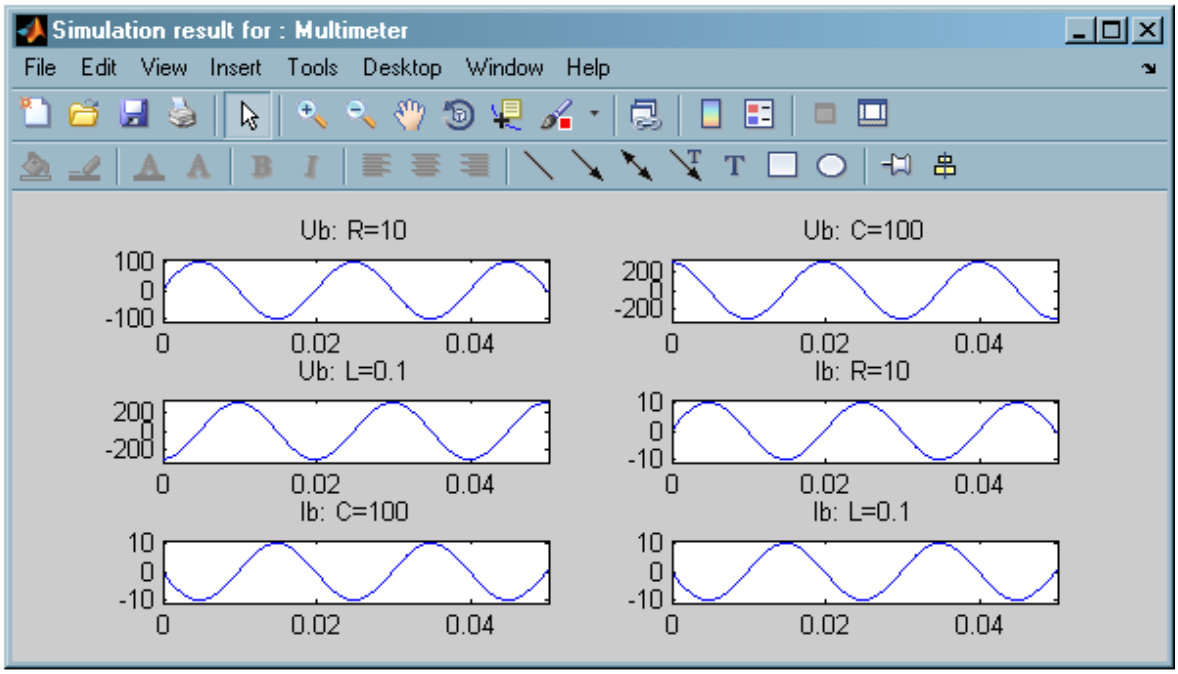
In pop-up window choose **Parameters** position and fill it in according to the given diagram parameters. In source position (position 5 in Figure) in **Type** column choose **popup** and fill it in according to given sources. You can add the required picture for the diagram looks better. In **Icon&Ports** position show the pass for the required picture.

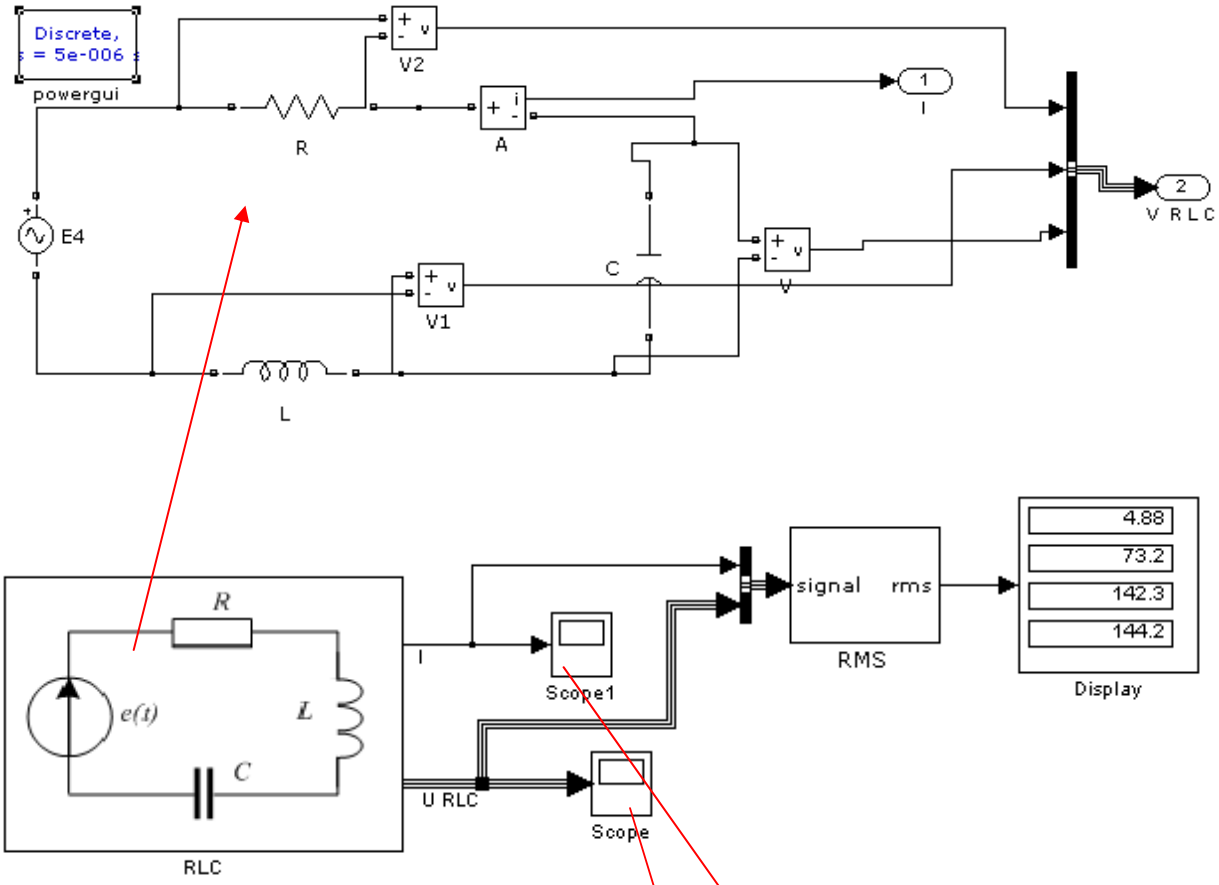


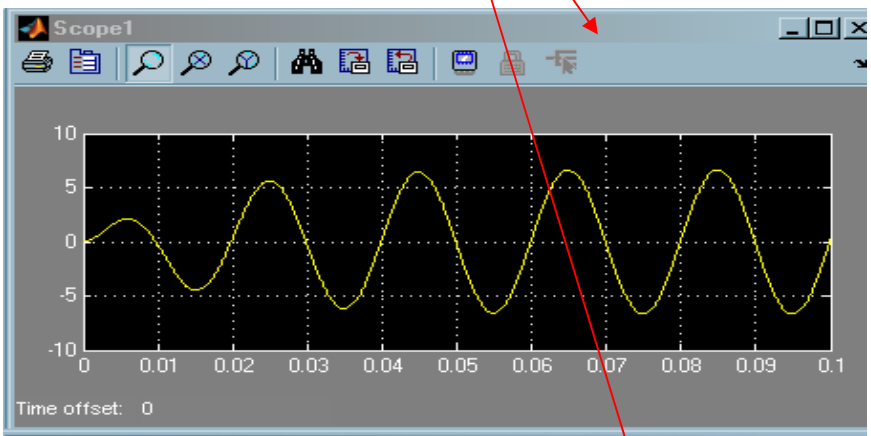
After diagram masking being completed, popup menu appears after left clicking the diagram area where you can change diagram parameters.

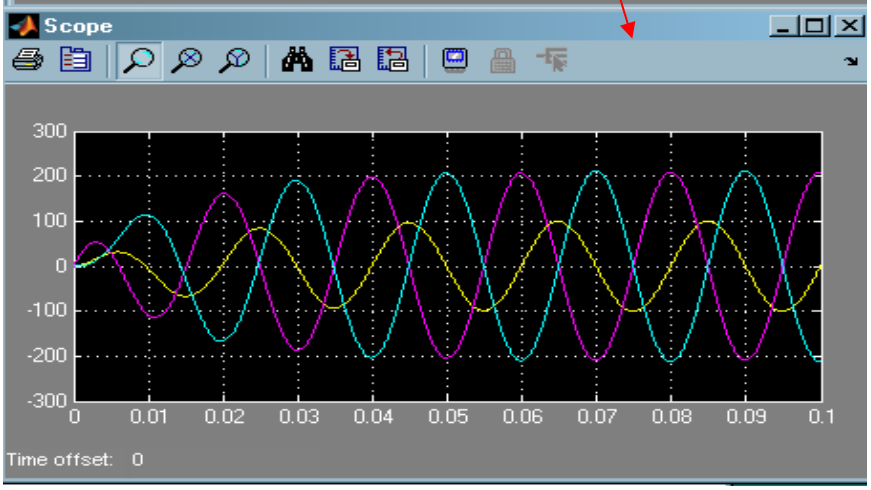

R
L
C
Gen
iL(t)
RLC
Source Block Parameters: RLC
Subsystem (mask)
Parameters
inductance
0.1
Capasitance
100*10^-6
Resistance
10
initial condition iL(0)
0
initial condition Uco
0
Wave form
10*sin(50t)
10*sin(50t)
10*cos(50t)
(10*sin(50t))^2
step(t)

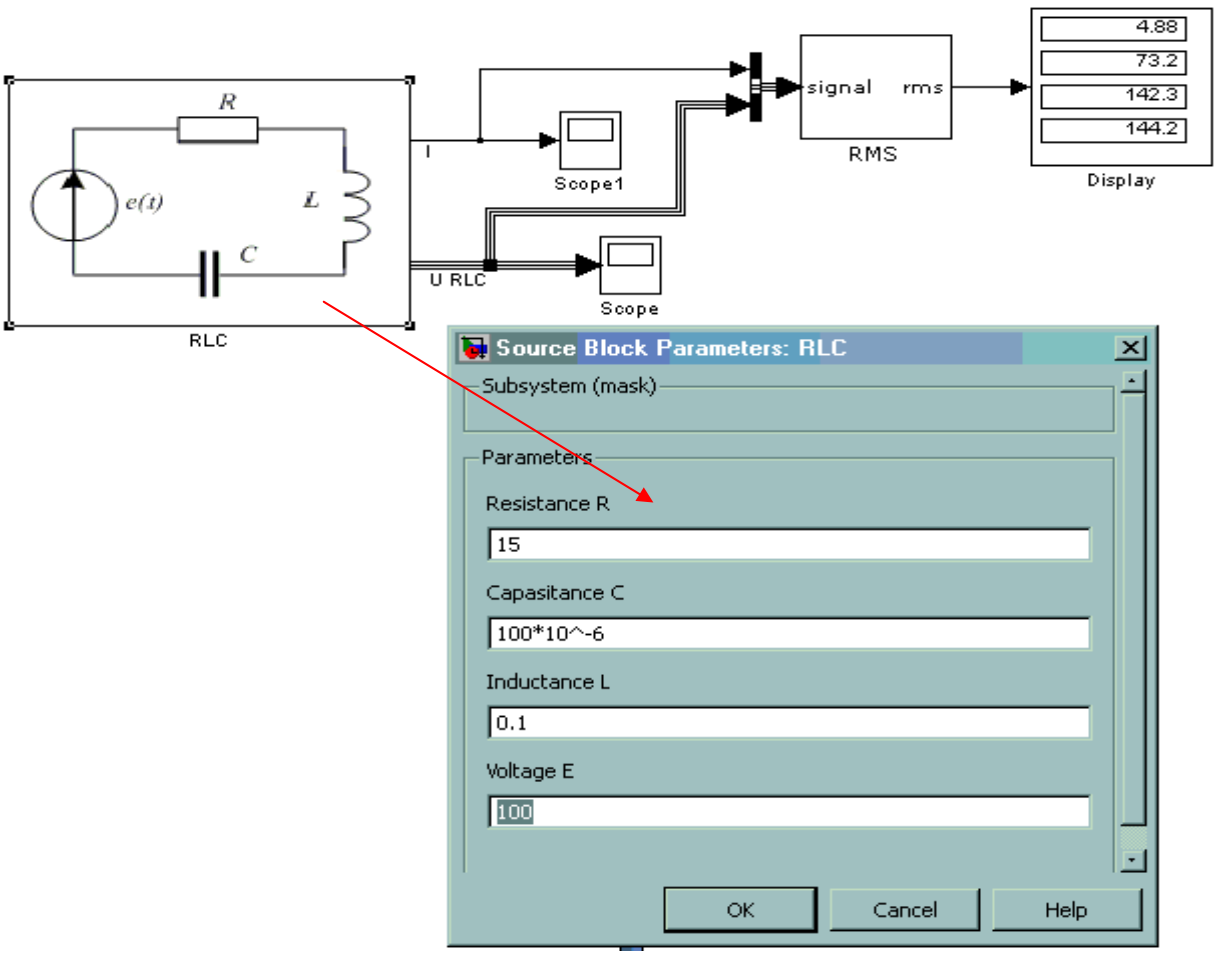

Scope
R=10
A
AC Voltage Source
C=100
L=0.1
70.65
224.9
222
7.065
7.065
7.065
6
signal
rms
Multimeter
RMS
Display
Discrete,
= 5e-006
powergui


Simulation result for : Multimeter
File Edit View Insert Tools Desktop Window Help
Ub: R=10
Ub: C=100
Ub: L=0.1
Ib: R=10
Ib: C=100
Ib: L=0.1
0
0.02
0.04


Discrete,
= 5e-006
powergui
V2
R
A
1
I
2
V R L C
E4
C
V
V1
L
R
e(t)
L
C
RLC
I
Scope1
signal
rms
RMS
4.88
73.2
142.3
144.2
Display
U RLC
Scope


Scope1
10
5
0
-5
-10
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1
Time offset: 0

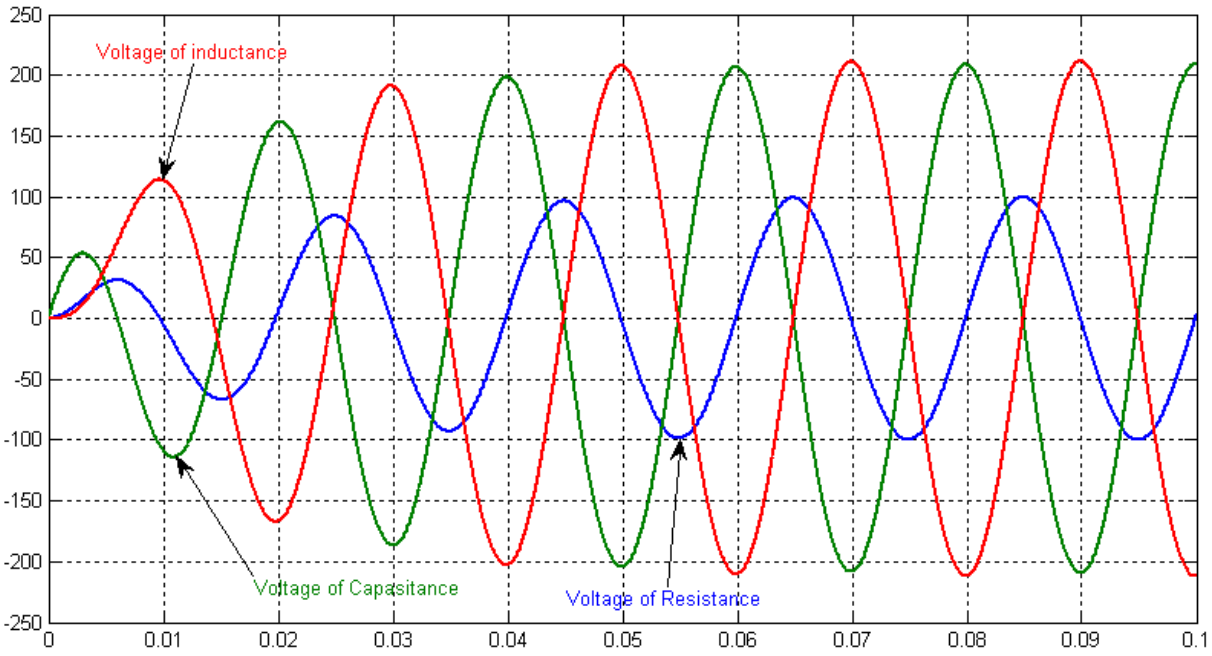

Scope
300
200
100
0
-100
-200
-300
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1
Time offset: 0



In command line we type the following,

```
>> plot(RLC(:,1),RLC(:,2),RLC(:,1),RLC(:,3),RLC(:,1),RLC(:,4))
```

then get the plot



Form the plop