Electric Circuit Analysis Laboratory Manual

prepared by

Er. Vishal Kr. Mittal

Assistant Professor

(Electrical Engineering Department)

Contents

Experiment 1

To verify the Kirchhoff's current and voltage laws applied to ac circuits.

Theory

An ac circuit is the circuit which contains an alternating source. The elements such as resistor, inductor and capacitor provide different phase difference to the current, and thus, the voltage drop occurring across these elements is not in same phase. Kirchhoff's current law for ac circuits states that the phasor sum of currents at any node or junction is zero, and Kirchhoff's voltage law for ac circuits states that the phasor sum of excitation emfs and voltage drops in a mesh or loop is zero.

Circuit Diagram

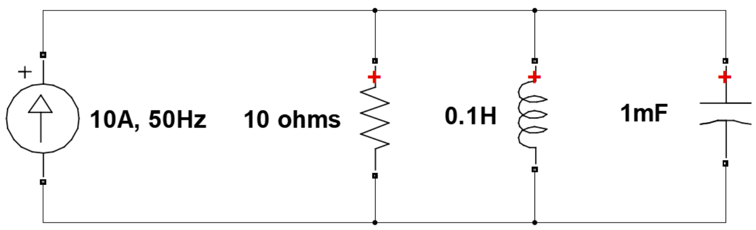

Fig. 1.1: Circuit diagram to verify Kirchhoff's current law

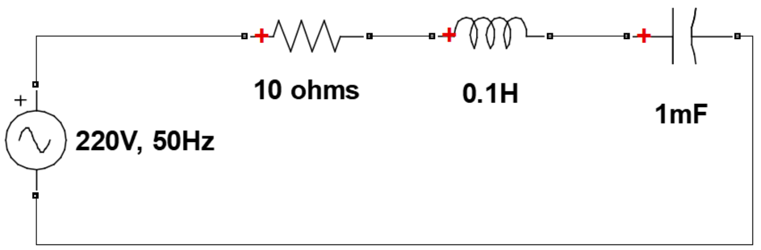

Fig. 1.2: Circuit diagram to verify Kirchhoff's voltage law

Procedure

PART A: Kirchhoff's Current Law

1. Open the *Simulink Library*, and open a new model by clicking on *File menu.*
2. Save this model.
3. Go to the *Simulink Library Browser* window, and select the *Simscape* library.
4. Under *Simscape*, go to *SimPowerSystems.*
5. Add the components given in Fig. 1.1 to the model. AC current source is available in *Electrical Sources*, and for R, L and C add *Series RLC Branch* from *Elements.*
6. To set the RLC branch to required R, L or C branch, double click on the component, and select the *Branch type* as per requirement, and set its value. Similarly, set the amplitude and the frequency of the current source block.
7. To measure current in each branch, add the *Current Measurement* block which is available under *Measurements* sublibrary of *SimPowerSystems.*
8. Connect the components as given in Fig. 1.3.
9. Add *powergui* block to the model which is available in *SimPowerSystems.* Double click on *powergui* block, set its solver simulation type to *phasor* and frequency to 50Hz.
10. Set the *output signal* of each of the current measurement block to *complex.*
11. To display the value of measured current, add *Display* block from the *Sinks* sublibrary of library *Simulink.*
12. To display magnitude and phase angle of the measured currents, add the block *Complex to Magnitude-Angle* which is found in the sublibrary *Math*

Operations of library *Simulink*. Use the *Gain* block also to convert the measured angle from radians to degrees, and amplitude of current to rms value. Create a *Subsystem* of these blocks to use it again as shown in Fig. 1.4.

13. To sum the currents $\bar{I}_1$, $\bar{I}_2$ and $\bar{I}_3$, add *Add* block from *Math Operations* sublibrary. Double clicking it and set the *List of Signs* to '+++' to sum three currents. Display this sum.
14. Compare the result of $\bar{I}_1$+ $\bar{I}_2$+$\bar{I}_3$ with the incoming current $\bar{I}$.

PART B: Kirchhoff's Voltage Law

1. Open a new model.
2. Assemble and connect the components as given in Fig. 1.5.
3. Compare the result of $\bar{V}_R + \bar{V}_L + \bar{V}_C$ with the applied voltage $\bar{V}$.

Observations

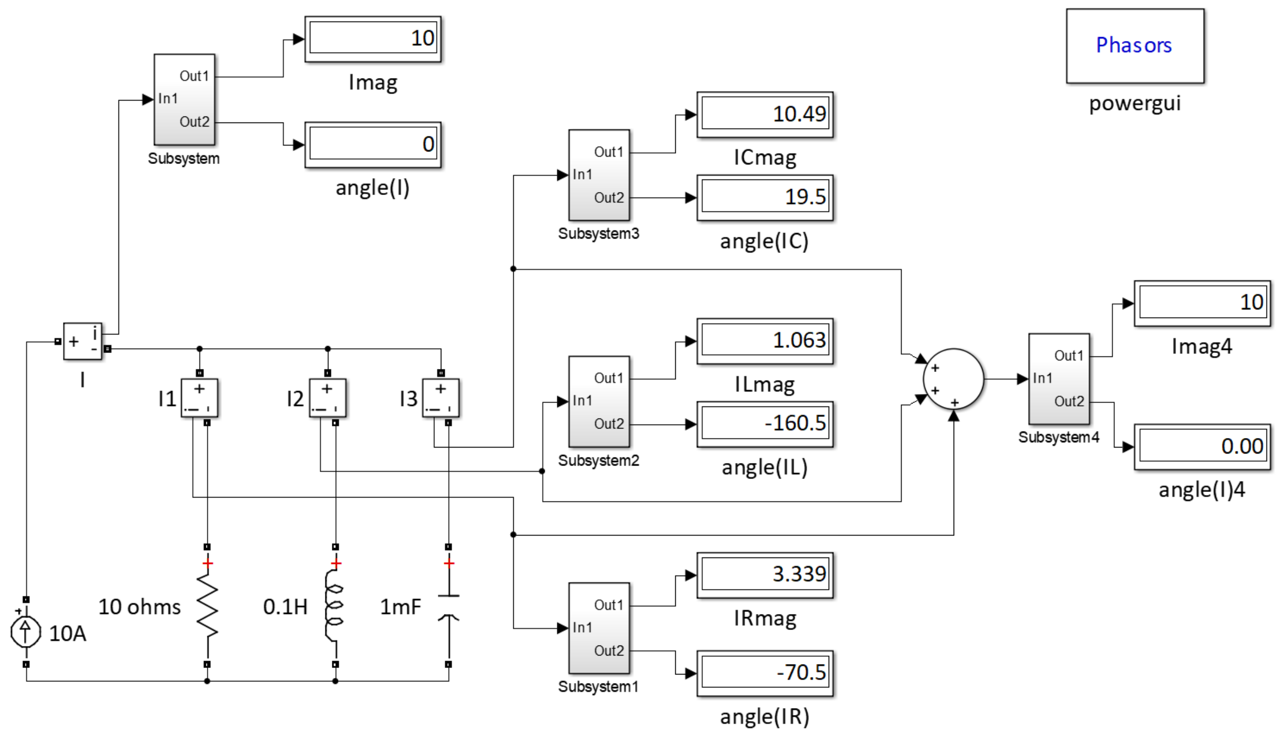

Fig. 1.3: Simulink model to verify Kirchhoff's current law

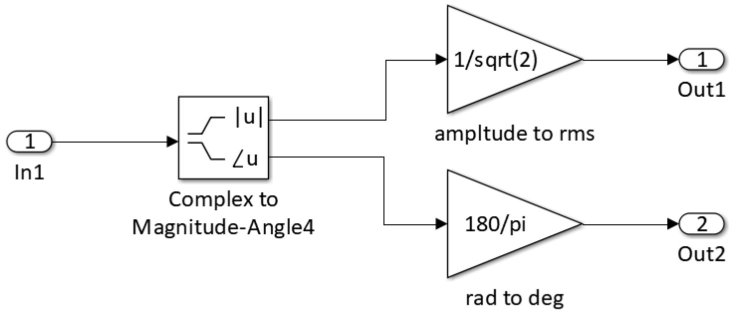

Fig. 1.4: Expansion of Subsystem utilized in Fig. 1.3 and 1.5

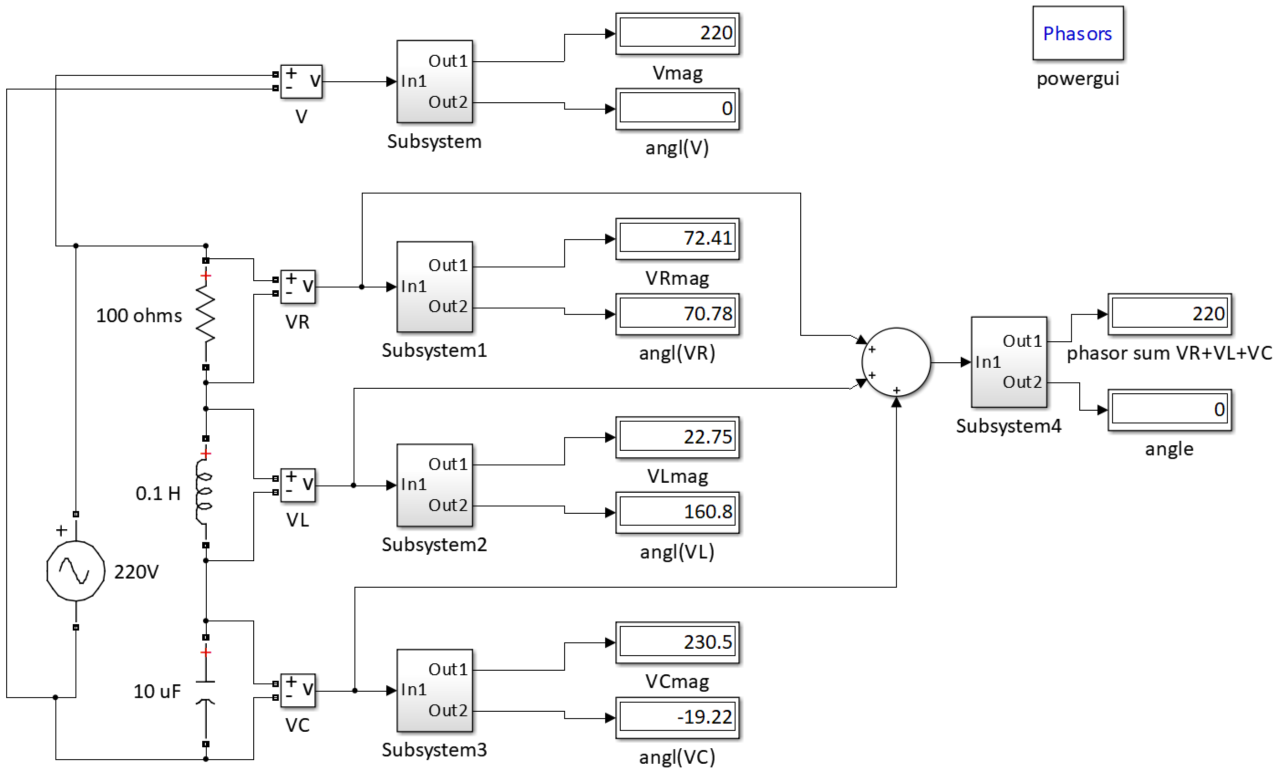

Fig. 1.5: Simulink model to verify Kirchhoff's voltage law

Result

As from the Fig. 1.3, it is seen that the phasor sum of currents leaving the junction is equal to the phasor current entering the junction, hence Kirchhoff's current law is verified.

Similarly from Fig. 1.5, it is seen that the phasor sum of voltage drops across the R, L and C is equal to the phasor value of excitation source, hence Kirchhoff's voltage law is verified.

Experiment 2

To verify the maximum average power transfer theorem applied to ac circuits.

Theory

The maximum average transfer theorem for ac circuits states that the condition for the maximum average power transfer to the load is that the load impedance must be equal to the complex conjugate of the internal impedance or Thevenin's impedance of the circuit, and the maximum average power transferred is given by $\frac{|V_{Th}|^2}{4R_{Th}}$, where V_{Th} is the Thevenin's voltage in rms and R_{Th} is the Thevenin's resistance of the internal circuit.

Circuit Diagram

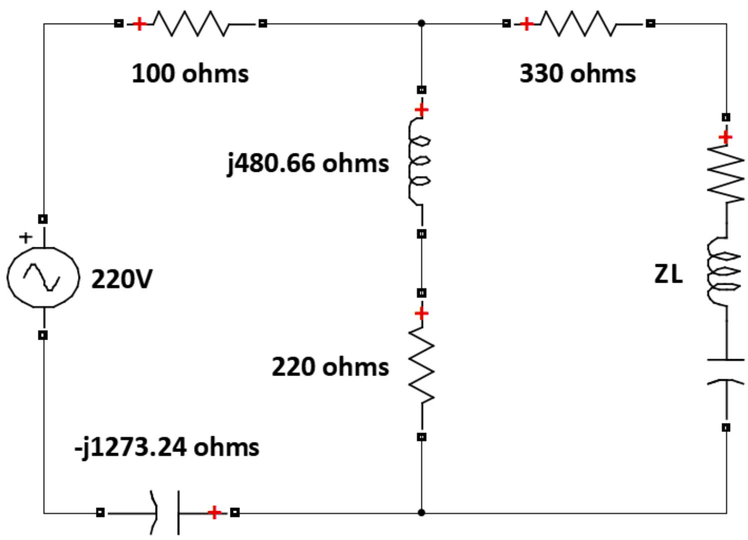

Fig. 2.1: Circuit diagram to verify maximum average power transfer theorem applied to ac circuits

Procedure

1. Open a new model in simulink to determine the Thevenin's voltage of the given circuit in Fig. 2.1.

2. Add the components required as per the given circuit diagram from the sublibrary *SimPowerSystems* as shown in Fig. 2.2.

3. Keep the load terminals open, and measure and display the open circuit voltage across these terminals as shown in Fig. 2.2. The description of subsystem is given in Fig. 2.4.

4. Open a new model in simulink to determine the Thevenin's impedance of the given circuit.

5. Add components required to this model as given in Fig. 2.4. Apply 1V, 50Hz ac voltage source at the open circuited terminals while short circuiting the 220V source as done in Fig. 2.3.

6. Measure the current drawn from the 1V source by this circuit. To compute Z_{Th}, divide the applied voltage and current drawn using *Division* block which is available in sublibrary *Math Operations* of library *Simulink*. The subsystem shown in Fig. 2.3 is described in Fig. 2.5.

7. Open a new script, and make a program of computing power transferred to the load Z_L for a range of zero to twice the magnitude of Z_{Th} for angle $\frac{-\pi}{2}$ to $\frac{\pi}{2}$. Obtain the surface and contour plots for the power transferred against the load impedance magnitude and angle. Determine the magnitude and angle of the load impedance corresponding to the maximum average power transferred.

8. Compare this obtained load impedance with the Z_{Th}.

```matlab
%Plotting the power delivered to load against varying load impedance
Vth_mag=136.1;              %Thevenin's voltage of circuit in volts
Vth_ang=133.4;              %Thevenin's voltage angle in deg
Zth_mag=1040;               %Thevenin's impedance magnitude
Zth_ang=34.3;               %Thevenin's impedance angle in deg
ZL_mag=0:0.01:Zth_mag*2;    %Load impedance magnitude vector
ZL_ang=-pi/2:0.01*pi:pi/2;  %Load impedance angle vector in rad, load angle lies between -90 to 90 deg
ZLmag_grid=meshgrid(ZL_mag,ZL_ang); %creating grid of ZL_mag of rows = elements in ZL_ang
ZL_phase=cos(ZL_ang)+1j.*sin(ZL_ang); %converting ZL_ang to phasors

%creating a grid of load impedance ZL = ZL_mag*ZL_phase
ZL=zeros(size(ZLmag_grid));
for k=1:1:length(ZL_phase)
    ZL(k,:)=ZL_phase(k).*ZLmag_grid(k,:);
end
%computing total impedance Z=Zth+ZL
Z=ZL+(Zth_mag*(cosd(Zth_ang)+1j*sind(Zth_ang))).*ones(size(ZLmag_grid));
%computing current grid: I=Vth/Z
Vth=Vth_mag*(cosd(Vth_ang)+1j*sind(Vth_ang));
I=Vth./Z;
%computing active power P=real(Vth*conj(I))
VL=I.*ZL;
P=real(VL.*conj(I));
%finding and displaying the maximum power transferred and corresponding
%value of ZL magnitude and ZL phase angle
[grid_x,grid_y]=find(P==max(max(P)));
y=ZL_ang(grid_x)*180/pi;
x=ZL_mag(grid_y);
fprintf('The maximum power transferred to load, Pmax = ');
fprintf('%4.3f ',max(max(P)));
fprintf('%s \n','watts');
fprintf('The Load impedance magnitude corresponsing to Pmax = ');
```

```
fprintf('%4.2f ',x);
fprintf('%s \n','ohms');
fprintf('The Load impedance angle corresponsing to Pmax = ');
fprintf('%4.2f ',y);
fprintf('%s \n','degrees');
%max avg. power transfer from formula
Pmax=(Vth_mag^2)/(4*real(Zth_mag*(cosd(Zth_ang)+1j*sind(Zth_ang))));
fprintf('The maximum power transferred to load using formula, Pmax = ');
fprintf('%4.3f ',Pmax);
fprintf('%s \n','watts');

%3D plot of power transferred variation with load impedance
fig1=figure(1);
set(fig1,'color','white');
m=mesh(ZL_mag,ZL_ang.*(180/pi),P);
title('Variation of power transferred to load with load impedance','fontsize',18,'fontweight','bold');
ylim([-90 90]);
xlabel('|Z_L| in \Omega','fontsize',14,'fontweight','bold');
ylabel('\angle{Z_L} in deg.','fontsize',14,'fontweight','bold');
zlabel('Load power (P_L) in watts','fontsize',14,'fontweight','bold');
set(gca,'fontsize',12,'fontweight','bold','gridlinestyle','--');
grid('minor');
%contour plot of power transferred variation with load impedance
fig2=figure(2);
set(fig2,'color','white');
[C,h]=contour(ZL_mag,ZL_ang.*(180/pi),P,linspace(0.1,max(max(P))-0.005,15));
set(h,'showtext','on','labelspacing',144*3,'linewidth',3);
xlabel('|Z_L| in \Omega','fontsize',14,'fontweight','bold');
ylabel('\angle{Z_L} in deg.','fontsize',14,'fontweight','bold');
title('Contour map showing variation of power transferred to load with load impedance','fontsize',18,'fontweight','bold');
set(gca,'fontsize',12,'fontweight','bold');
grid(gca,'minor');
```

Observations

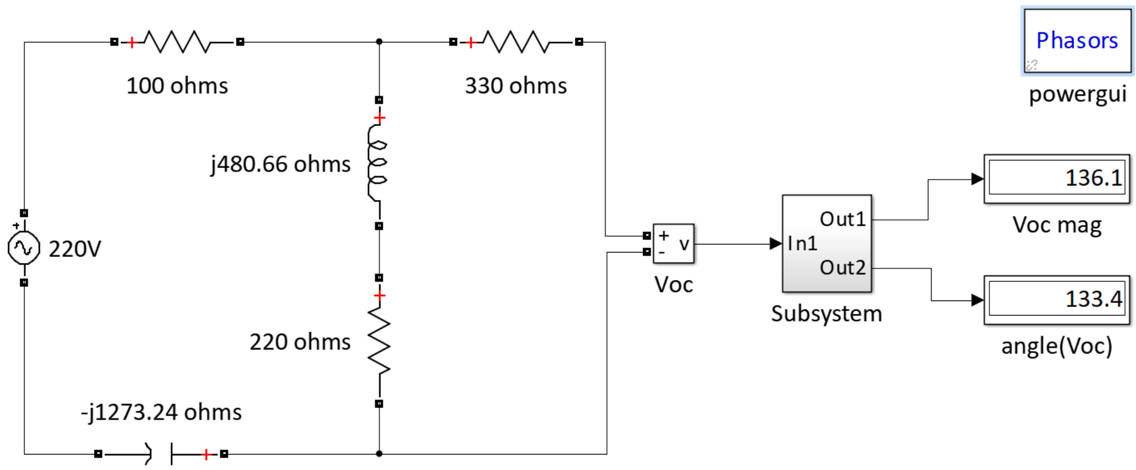

Fig. 2.2: Simulink model to determine Thevenin's voltage of the given circuit

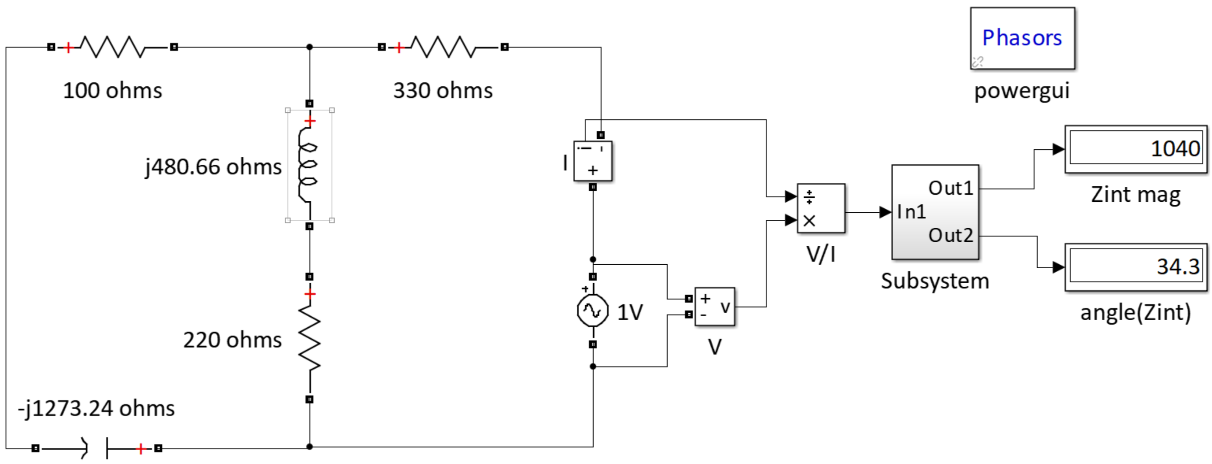

Fig. 2.3: Simulink model to determine Thevenin's impedance of the given circuit

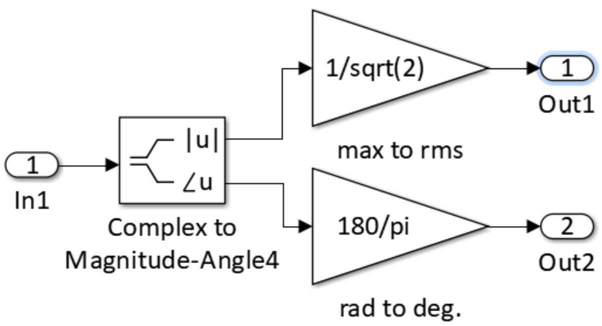

Fig. 2.4: Subsystem of Fig. 2.2

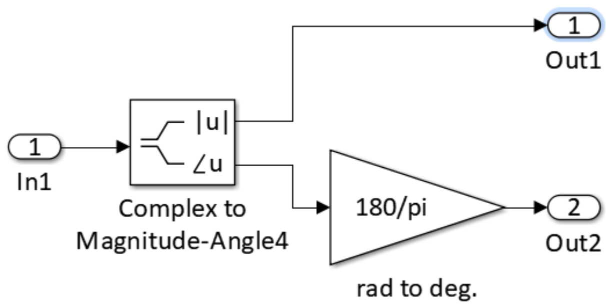

Fig. 2.5: Subsystem of Fig. 2.3

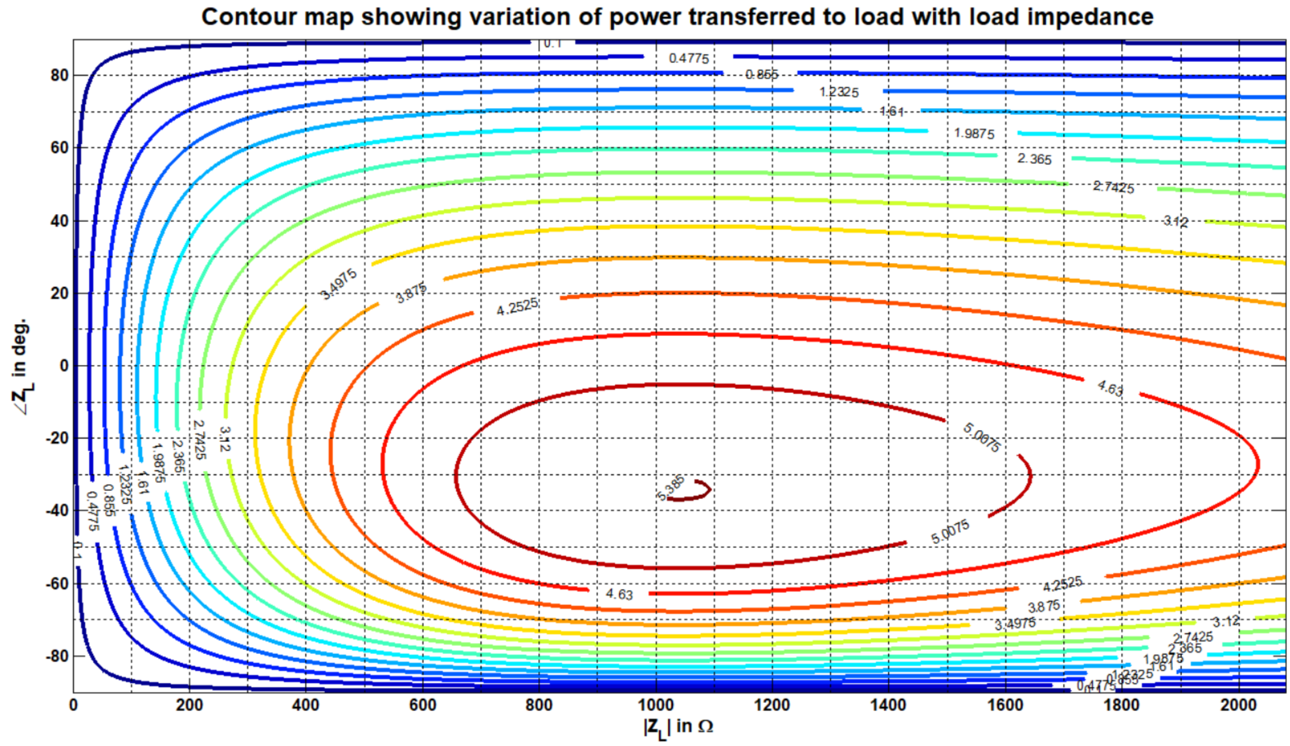

Fig. 2.6: Contour plot showing the average power transfer variation with load impedance for the circuit given in Fig. 2.1

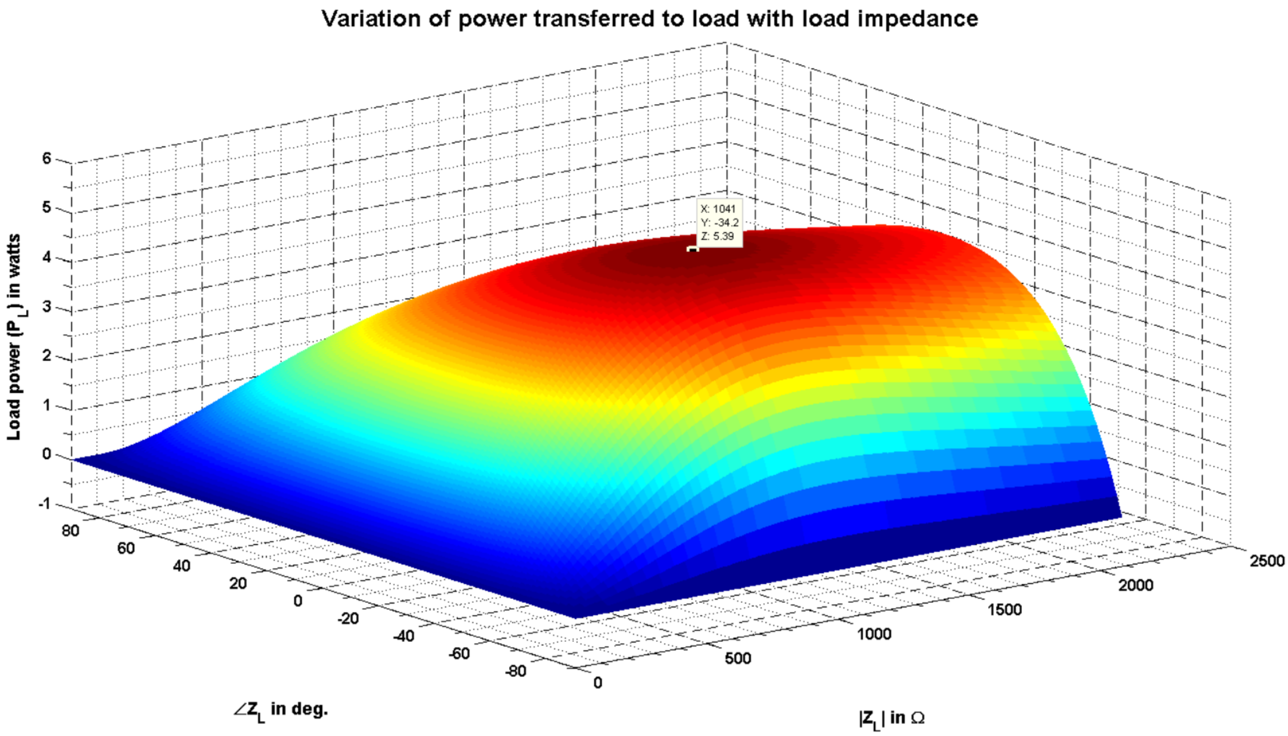

Fig. 2.7: Mesh plot showing the average power transfer transfer variation with load impedance for the circuit given in Fig. 2.1

Program Output

```
Command Window
>> Plot_P_Zint
The maximum power transferred to load, Pmax = 5.390 watts
The Load impedance magnitude corresponsing to Pmax = 1040.00 ohms
The Load impedance angle corresponsing to Pmax = -34.20 degrees
The maximum power transferred to load using formula, Pmax = 5.390 watts
fx >>
```

Result

From the computer program it can be seen that the maximum average power transferred occurs at $Z_L = 1040\angle{-34.2^o}\ \Omega$ which is same as Z^*_{Th}. Thus, in an ac circuit the maximum average power transferred occurs to the load impedance occurs when the load impedance is equal to the complex conjugate of the circuit's internal impedance.

Experiment 3

To verify the Tellegen's theorem.

Theory

Tellegen's theorem states that in a circuit, the total power is conserved, i.e., the amount of power absorbed is equal to the amount of power supplied. Thus, in case of dc circuit,

$$P^{dc}_{supplied} = P^{dc}_{absorbed}$$

For ac circuits,

$$P^{ac}_{supplied} = P^{ac}_{absorbed}$$

$$Q^{ac}_{supplied} = Q^{ac}_{absorbed}$$

where P and Q are active and reactive powers respectively.

Circuit Diagram

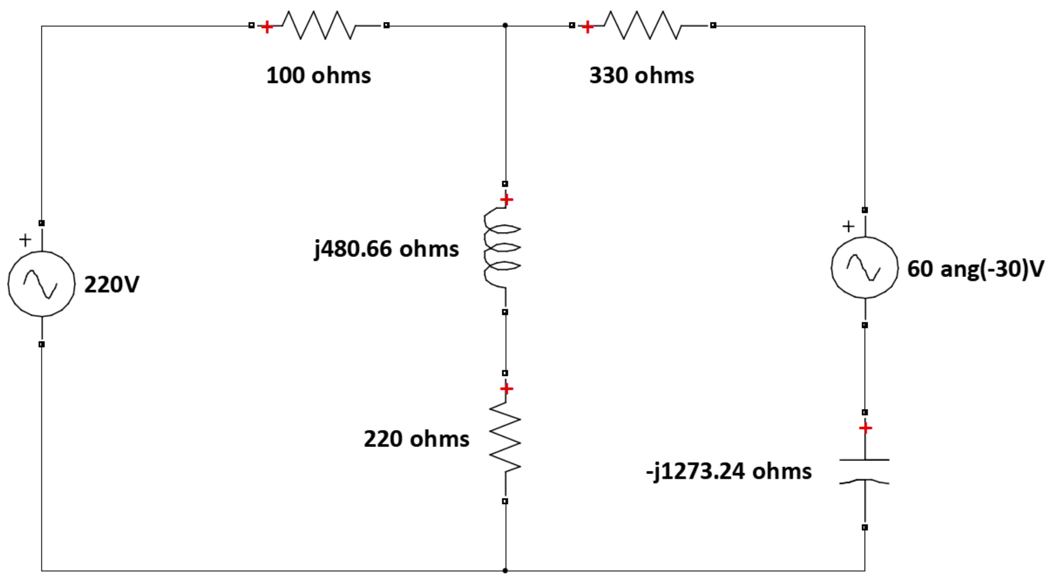

Fig. 3.1: Circuit diagram to verify Tellegen's theorem

Procedure

1. Open a new model in simulink to verify the Tellegen's theorem in the given circuit of Fig. 3.1.
2. Add the components required as per the given circuit diagram from the sublibrary *SimPowerSystems* as shown in Fig. 3.2.
3. Assuming that the power is delivered by each component, connect voltmeter across each element and ammeter in each series path to compute power.
4. Set the output signal of these meters to *complex*. Add *Out Port* blocks from the *Sinks* sublibrary of *Simulink*. Connect the output signal of each meter to respective *Out Port*.
5. Run this model. The output signals will get stored in the workspace.
6. Open a new script, and write a program to pass these output signals to *In Port* as shown in Fig. 3.3.
7. Open a new model to display the complex power delivered by each element.
8. Add *In Port* blocks from the *Sources* sublibrary of *Simulink*.
9. *In Port* takes the data from the workspace. For complex values, the block takes the signal values in structure format. This has been shown in Fig. 3.3. Open *Model Configuration Parameters* window, and set the *Input* in *Data Import/Export* section to the name of the structure as used while writing program in Fig. 3.3.
10. Add *Math Function* block from the sublibrary *Math Operations* of *Simulink*.
11. By double clicking this block, set the function to *conj* to take the complex conjugate of the current.
12. Add product block from *Math Operations*, and compute $\bar{V} \times \bar{I}^*$.
13. Add *Display* block to display the complex power delivered by each element.

14. Using *Add* block take the sum of each of the complex power, and display the result as shown in Fig. 3.4.

Observations

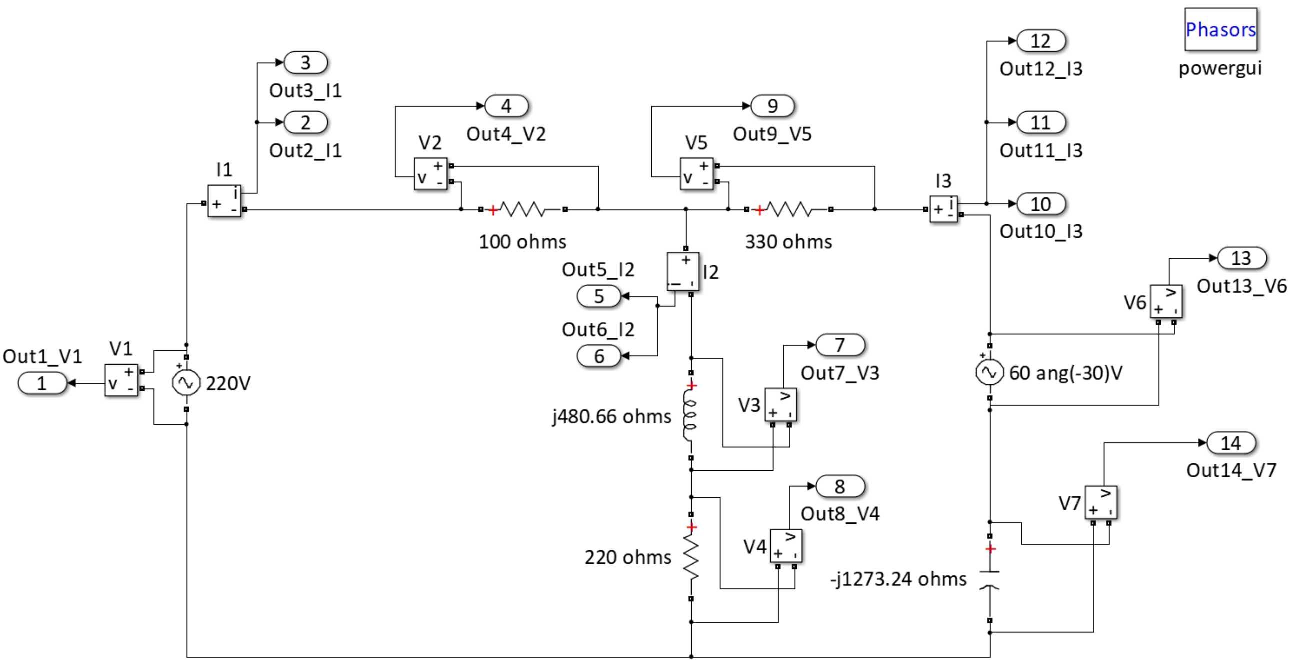

Fig. 3.2: Simulink model to verify Tellegen's theorem

```
%Input port programming for Exp 3: Tellegen's theorem
%The data is passed through structure of time and signals as data is
%complex in nature
%tout and yout are already in the workspace as model1 containing the output
%ports was run first

%passing elements to structure named pwr
pwr.time=tout;
for k=1:1:14            %there are 14 ports in the model containing circuit
                        %for Tellegen's theorem
    pwr.signals(k).values=yout(:,k);
end
```

Fig. 3.3: *In port* programming

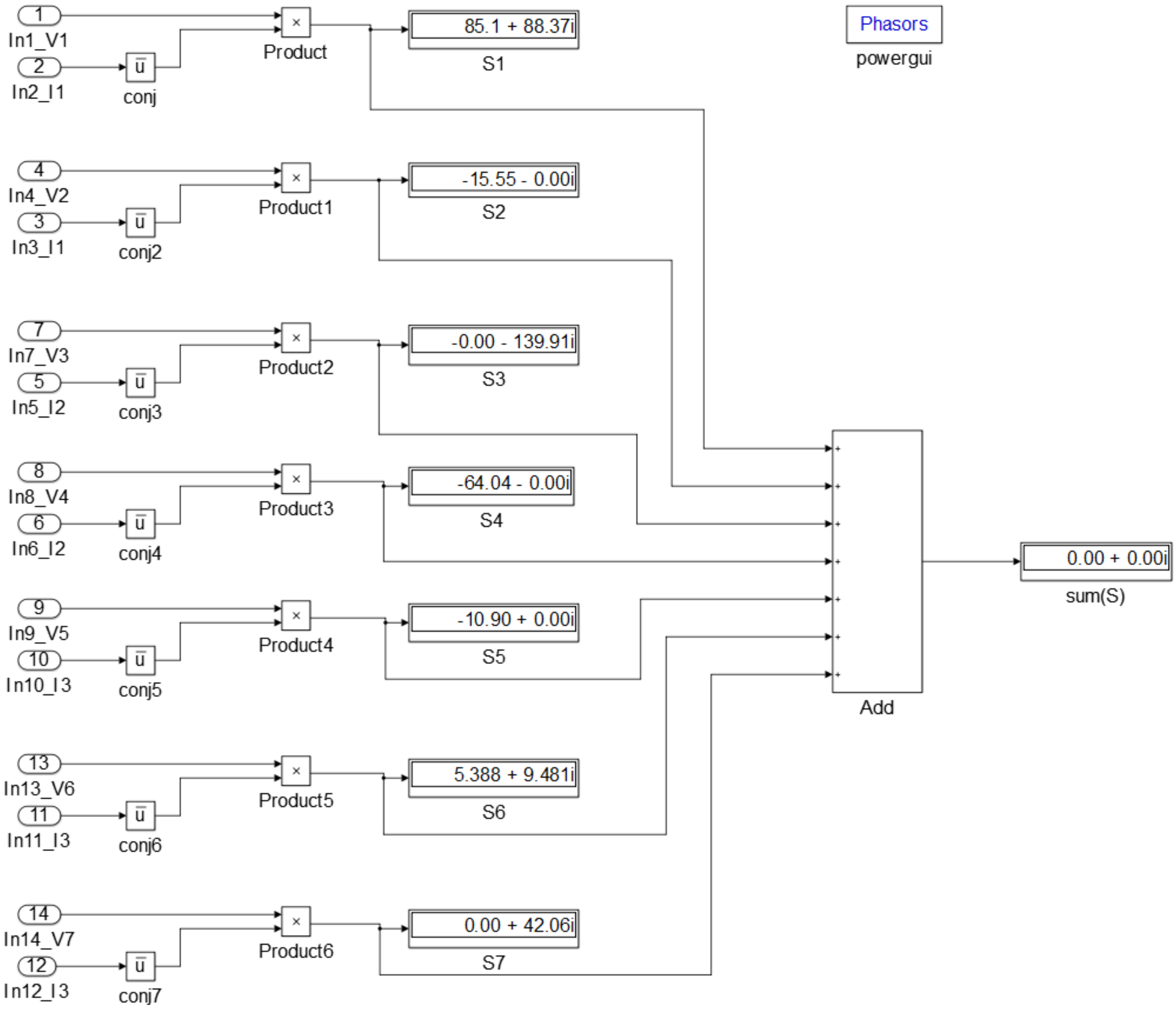

Fig. 3.4: Simulink model to display the complex power delivered by each element, and total complex power of the circuit.

Result

From the Fig. 3.4, it can be seen that the total sum of complex power delivered by each element in the given circuit is zero. Thus, the power in a circuit remains conserved. This verifies Tellegen's theorem.

Experiment 4

To verify the reciprocity theorem.

Theory

Reciprocity theorem states that in any branch of a linear and bilateral network, the current due to a single source elsewhere in the network is equal to the current through the branch in which the source was originally placed when the source is placed in the branch in which the current was originally obtained.

Circuit Diagram

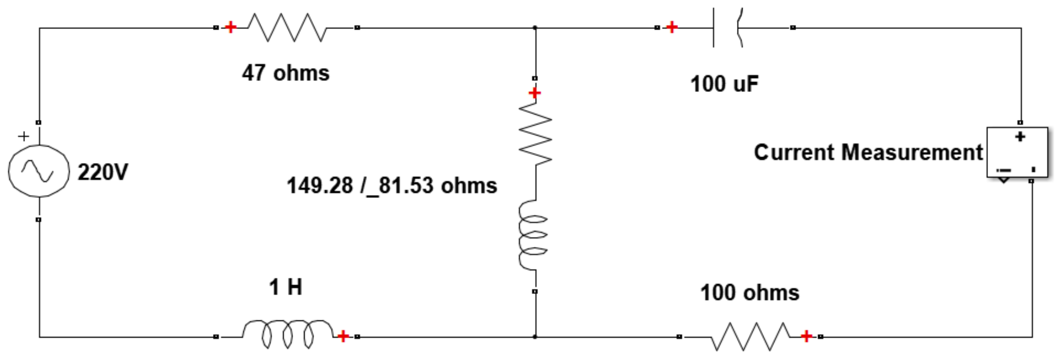

Fig. 4.1: Circuit diagram to verify reciprocity theorem

Procedure

1. Add the required components blocks from library *Simscape* sublibrary *SimPowerSystems* to simulate the circuit diagram given in Fig. 4.1.
2. Set the required values of the components.
3. Determine the current flowing in the branch given in Fig. 4.1 by setting the current measurement block output signal to complex. Use the *Complex to Magnitude-Angle* block from *Maths Operation* sublibrary of *Simulink*

library to display the magnitude and angle of the current in that branch. This has been presented in Fig. 4.2.

4. Now interchange the voltage source and current measurement block positions, as shown in Fig. 4.3.

5. Now run to determine the current in the interchanged branch and compare the current with the current obtained in the original circuit.

Observations

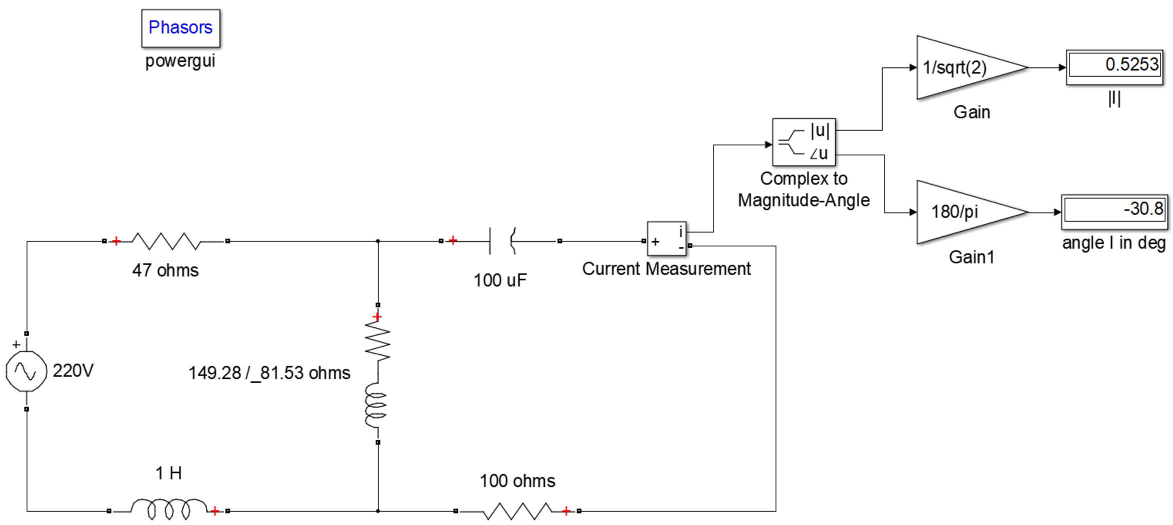

Fig. 4.2: Simulink model of original circuit to determine the current in a branch to verify the reciprocity theorem

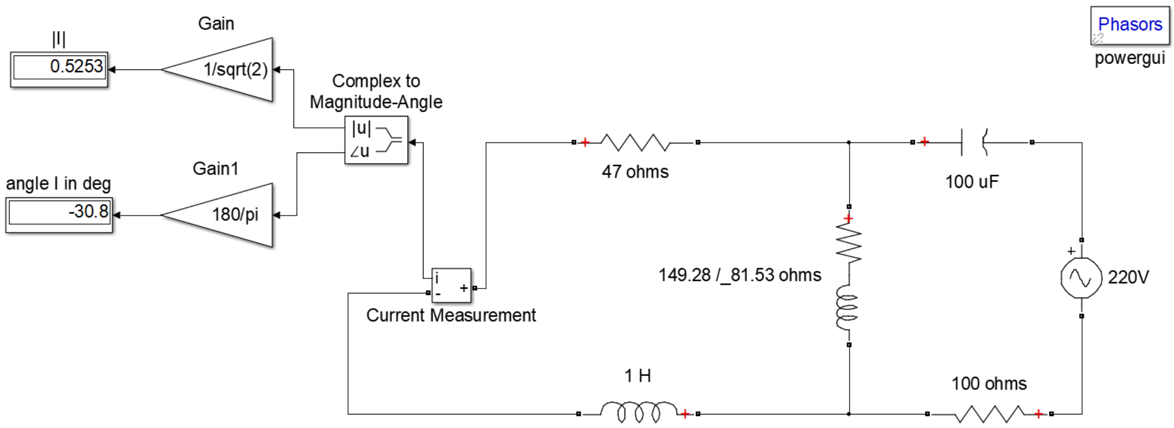

Fig. 4.3: Simulink model with interchanged positions of voltage source and current measurement blocks

Result

From Fig. 4.2 and Fig. 4.3, it can be seen that the magnitude and phase angle of the current in the given branch of the original circuit and the current with the interchanging the positions of the source and the current measurement blocks are same. Thus, the reciprocity theorem has been verified.

Experiment 5

To obtain sinusoidal steady state response of series RLC circuit.

Theory

The elements resistance, inductor and capacitor provide different phase difference to the ac current flowing through them. Inductor makes the ac current to lag behind voltage across it by 90^o, capacitor makes the current to lead the voltage across it by 90^o, whereas the resistor does not provide any phase difference. Thus, sinusoidal steady state study of a series RLC circuit is the study of behaviour of the circuit with the application of the sinusoidal input voltage at the steady state. Steady state is the state of the circuit when all the disturbances are died out.

Circuit Diagram

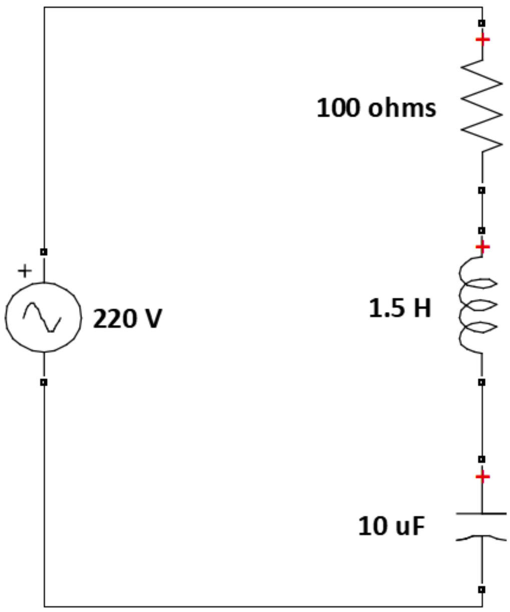

Fig. 5.1: Circuit diagram to study sinusoidal steady state response in series RLC circuit

Procedure

1. To simulate the circuit diagram given in Fig. 5.1, assemble the components in a simulink model as per the Fig. 5.2.
2. Double click on *powergui* block and configure the parameter to set the *simulation* type to continuous.
3. Get the voltage waveforms across the resistor, inductor and capacitor using *scope* block.
4. To display the above three waveforms in a single plot, use *Mux* block from *Commonly used block* sublibrary of *Simulink*.
5. Similarly measure the current and display its waveform.
6. Compute the phase difference of the current with respect to the applied voltage from the plot.

Observations

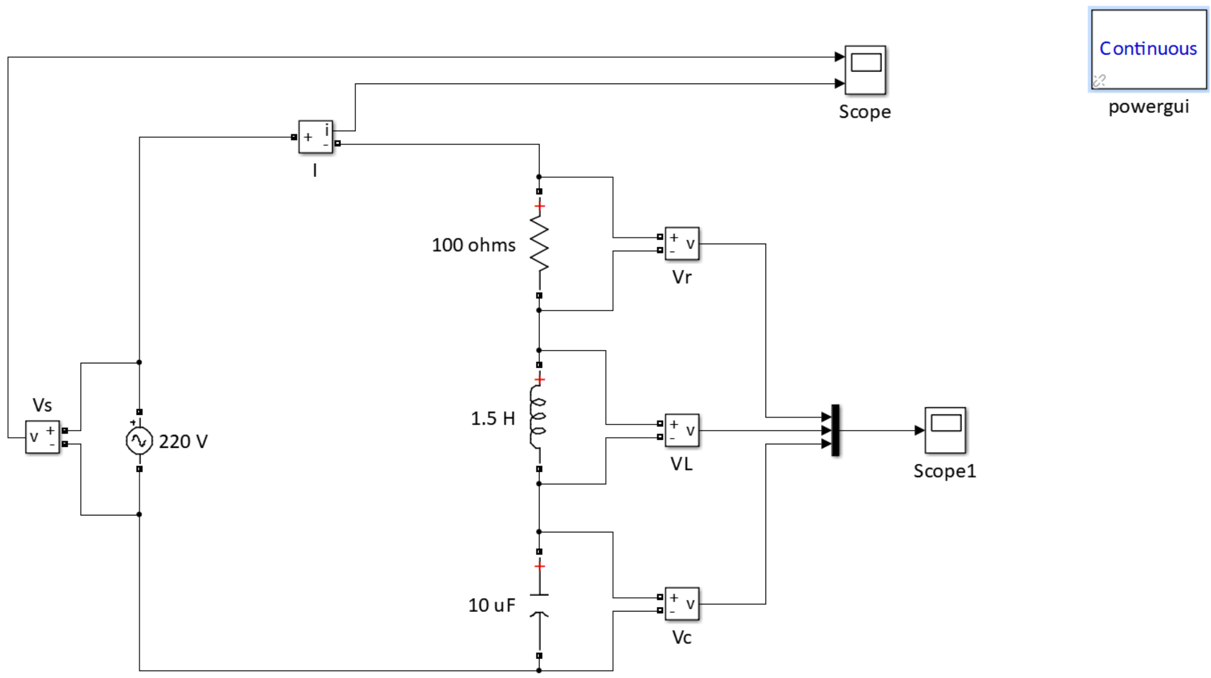

Fig. 5.2: Simulink model to study sinusoidal steady state response of circuit given in Fig. 5.1

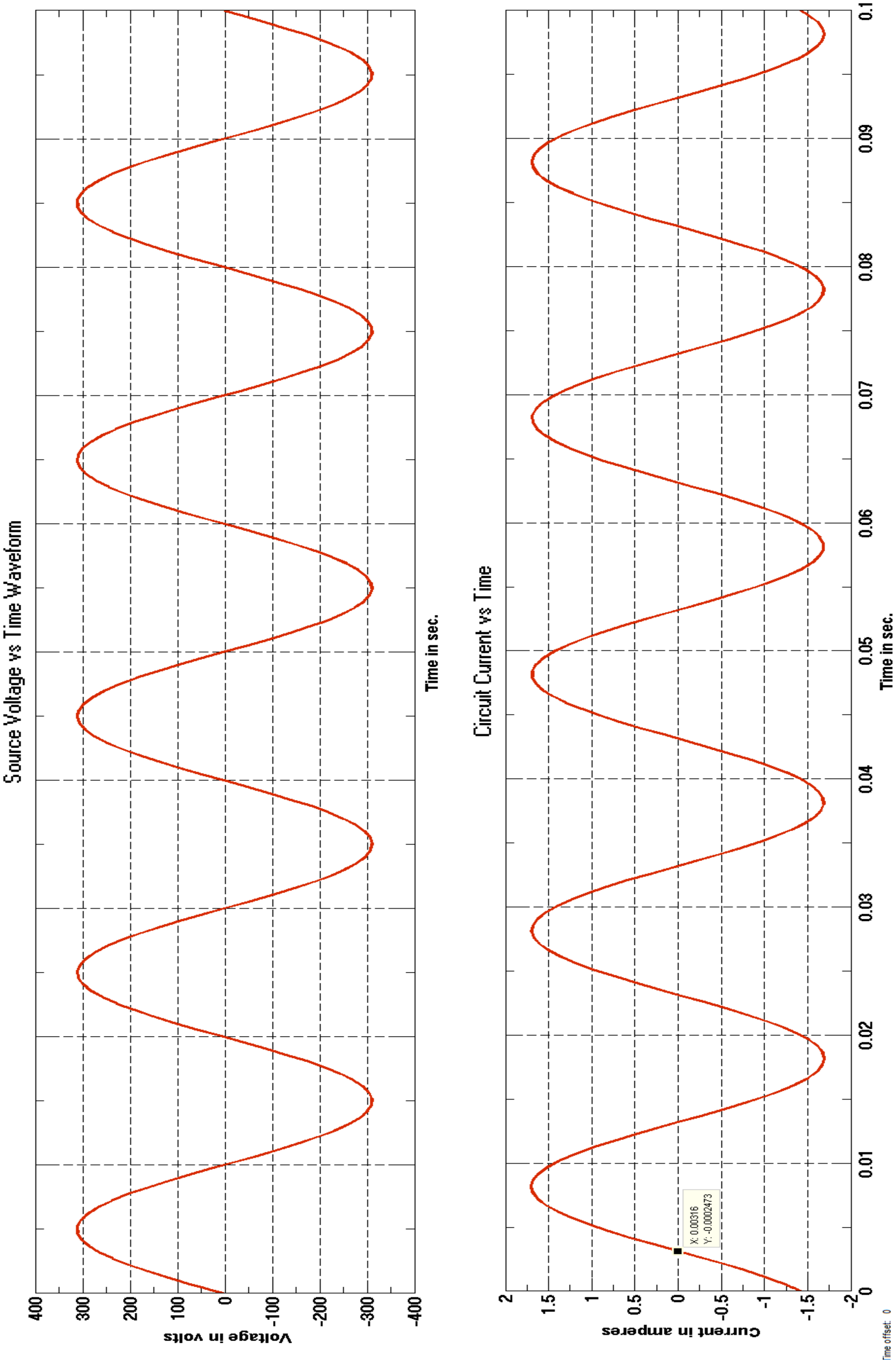

Fig. 5.3: Applied voltage and current waveforms

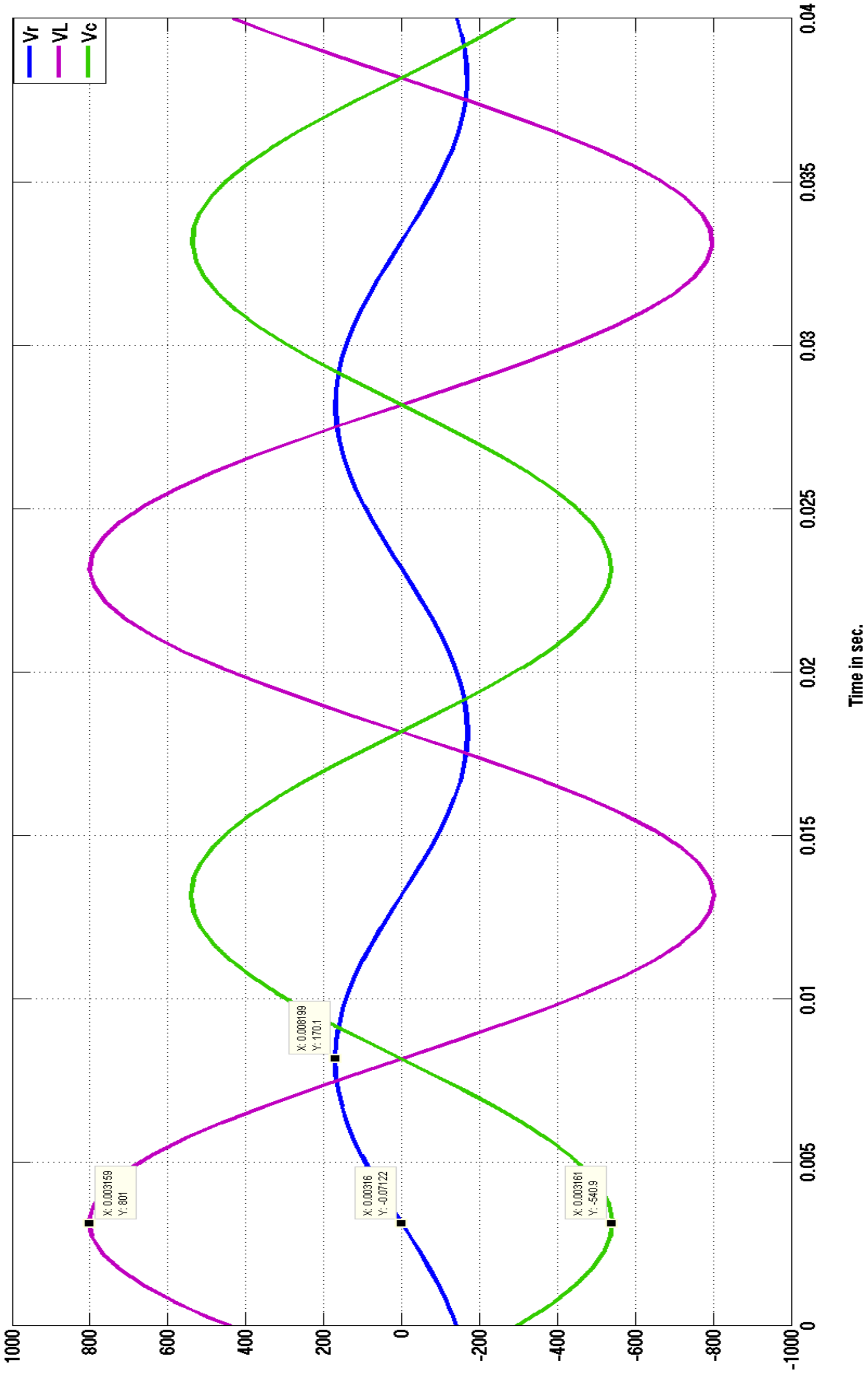

Fig. 5.4: Waveforms of voltages across resistor, inductor and capacitor

Result

From Fig. 5.3 it can be seen that the circuit is predominantly inductive as the current lags the voltage by $\frac{180}{0.01} \times 0.00316 = 56.88^o$. It can be seen from Fig. 5.4 that voltage across resistor remains in phase with the current flowing through it, whereas in case of inductor the voltage across it leads the current by 90^o and in case of capacitor the voltage lags the current by 90^o.

Experiment 6

To study transients in series RLC circuit excited from a step input.

Theory

RLC circuits are the second-order circuits. The responses of these circuits are described by second order differential equations. These circuits contain two energy storing elements - inductor and capacitor. Due to these energy storing elements, the circuit gives both the transient response and the steady-state response. The transient response due to step input may be oscillatory in nature if the circuit is underdamped, or may be critically damped response, or overdamped response.

Procedure

1. Add the components in a simulink model as are shown in Fig. 6.1 for different values of resistance R.
2. Set the *Source type* of controlled voltage source to *DC* and *initial voltage* to 10V.
3. Set the *step time* to 0 and *final value* to 1 of the *step input* block.
4. Set the *simulation type* of *powergui* block to *continuous* mode.
5. Use *Goto* and *From* blocks as shown in Fig 6.1 from the sublibrary *signal routing* of library *simulink* to display the waveform of current.
6. Set the *Goto tag* of *From* block by double clicking it to the required tag name as set for *Goto* block.
7. Run the simulation for 6 seconds, and obtain the waveform of current (response) using *scope*.

Observations

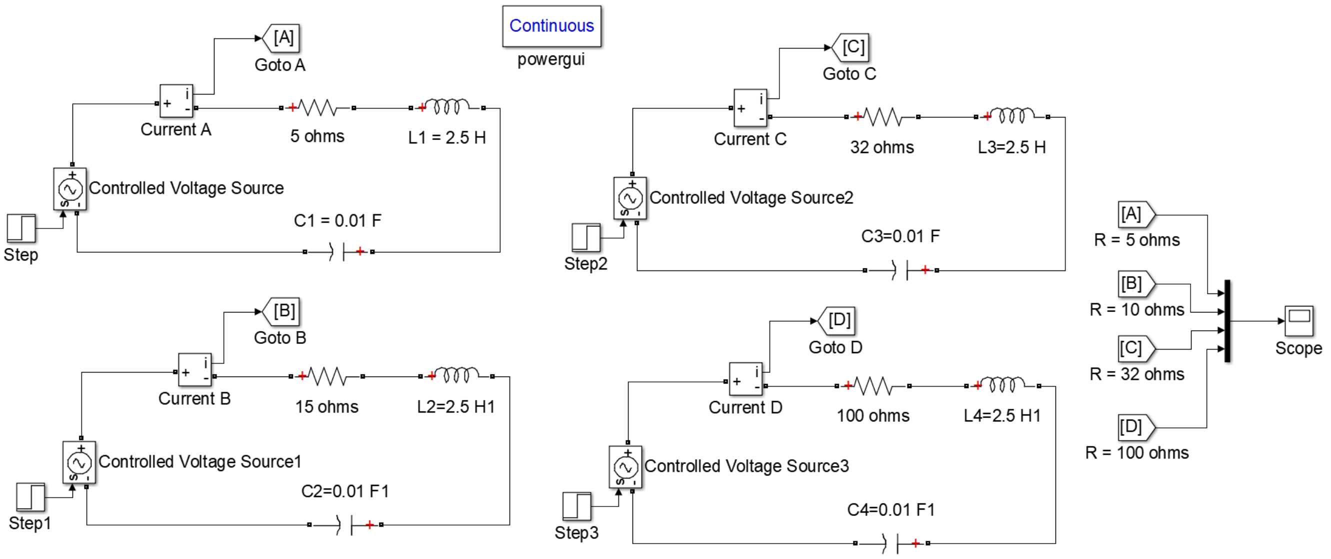

Fig. 6.1: Simulink model to study step response of series RLC circuit for different values of resistance

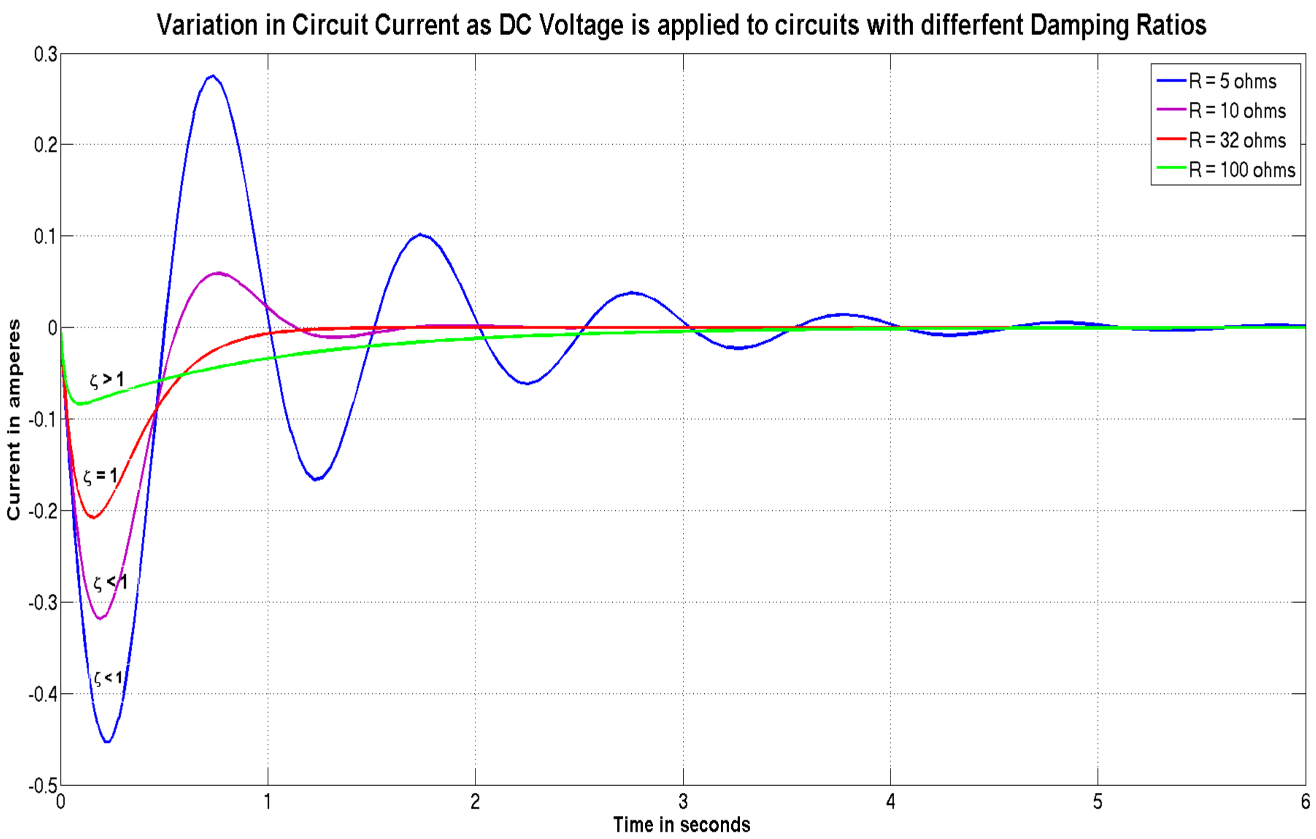

Fig. 6.2: Waveforms of response current due to step input for different values of resistance R in the circuit

Result

From the Fig. 6.2, it can be seen that as the resistance in the circuit increases, the current response shifted from underdamping nature, $\zeta < 1$ to overdamping nature, $\zeta > 1$. Overdamped circuit is sluggish in nature, i.e., it achieves the steady-state value very slowly, whereas the critically damped circuit achieves the steady-state value quicker.

Experiment 7

To study resonance in series RLC circuit.

Theory

Series Resonance is the condition in a series RLC circuit that the circuit behaves as purely resistive and it seems that reactive components are not there. This occurs at a particular frequency known as resonating frequency. At this frequency the inductive and capacitive reactances become equal and cancelled out. Series resonating circuits are used in a.c. filters, noise filters, radio an television tuning circuits etc. Resonance may be achieved either by varying the frequency or by varying inductance or capacitance in the circuit. In designing resonating circuits, the half-power frequencies play an important role. The frequency band between half-power frequencies is called bandwidth. The signals lying outside this bandwidth do not contain significant power, and hence bandwidth is a measure of selectivity.

Formulae Used

- $\bar{Z} = R + j(X_L - X_C) = \sqrt{R^2 + (X_L - X_C)^2} \angle tan^{-1}\left(\frac{X_L - X_C}{R}\right)$
- $\bar{I} = \frac{V\angle 0^o}{\bar{Z}}$
- $\left(\bar{I}\right)_{w_r} = \frac{V}{R}\angle 0^o$
- $(P)_{any\ \omega} = \left|\bar{I}\right|_{\omega}^{2} R$
- $\omega_r = \frac{1}{\sqrt{LC}}$
- Bandwidth, $B = \omega_2 - \omega_1$, where ω_2 and ω_1 are half power frequencies

```
%--------------Studying series resonance in series RLC circuit-------------
%*studying variation of circuit current with frequency
%*studying variation of impedance magnitude with frequency
%*studying variation of impedance angle with frequency
%*studying variation of power transferred with frequency
%*studying the effect of bandwidth and quality factor

%The given circuit is consisting of L of 25mH, C of 0.625uF, R of four
%different values: 4 ohms, 10 ohms, 20 ohms and 50 ohms, the magnitude of
%applied voltage is 100V
L=25*10^-3;
C=0.625*10^-6;
R=[10;25;50;100];
Vm=100;
w=0.001:0.01:10000;                %taking frequency range in rad/s
%computing reactances at different frequencies in ohms
XL=L.*w;
XC=1./(C.*w);
%creating meshgrid for computing Z at different freq. and resistances
XLgrid=meshgrid(XL,1:1:length(R));       %matrix R_length * XL_length
XCgrid=meshgrid(XC,1:1:length(R));       %matrix R_length * XL_length
Rgrid=meshgrid(R,1:1:length(XL))';       %matrix R_length * XL_length
Zgrid=Rgrid+1j.*(XLgrid-XCgrid);
%plotting impedances vs frequency plot for different values of R
fig1=figure(1);
set(fig1,'color','white','name','Impedance vs frequency plot');
clr=[0.8 0    0   ;
     0    0.7 0   ;
     0    0.6 0.8;
     0.8 0    0.8];
subplot(2,1,1);
hold on;
for k=1:1:length(R)
    hand=plot(w,abs(Zgrid(k,:)),'linewidth',2.5);
    set(hand,'color',clr(k,:));
end
title('|Z| variation with frequency','fontsize',20,'fontweight','bold');
xlabel('\omega in rad/s','fontsize',16,'fontweight','bold');
ylabel('|Z| in \Omega','fontsize',16,'fontweight','bold');
set(gca,'fontsize',14,'fontweight','bold');
xlim([5000 10000]);
grid on;
str=[];
for k=1:1:length(R)
    str=[str;cellstr(strcat('R=',num2str(R(k)),'\Omega'))];
end
legend(str);
subplot(2,1,2);
hold on;
for k=1:1:length(R)
    hand=plot(w,angle(Zgrid(k,:)).*(180/pi),'linewidth',2.5);
```

```
    set(hand,'color',clr(k,:));
end
title('\angle{Z} variation with frequency','fontsize',20,'fontweight','bold');
xlabel('\omega in rad/s','fontsize',16,'fontweight','bold');
ylabel('\angle{Z} in deg.','fontsize',16,'fontweight','bold');
set(gca,'fontsize',14,'fontweight','bold');
xlim([5000 10000]);
grid on;
legend(str,'location','southeast');
%computing current and plotting against freq. for difrerent values of R
Vmgrid=meshgrid(Vm.*ones(1,length(w)),1:1:length(R));
Igrid=Vmgrid./abs(Zgrid);
fig2=figure(2);
set(fig2,'color','white','name','Current vs frequency plot');
hold on;
for k=1:1:length(R)
    hand=plot(w,Igrid(k,:),'linewidth',2.5);
    set(hand,'color',clr(k,:));
end
title('|I| variation with frequency','fontsize',20,'fontweight','bold');
xlabel('\omega in rad/s','fontsize',16,'fontweight','bold');
ylabel('|I| in amperes','fontsize',16,'fontweight','bold');
set(gca,'fontsize',14,'fontweight','bold');
xlim([5000 10000]);
grid on;
str=[];
for k=1:1:length(R)
    str=[str;cellstr(strcat('R=',num2str(R(k)),'\Omega'))];
end
legend(str);
%computing power and plotting against freq. for different values of R
Pgrid=Igrid.^2.*Rgrid;
fig3=figure(3);
set(fig3,'color','white','name','Power vs frequency plot');
hold on;
for k=1:1:length(R)
    hand=plot(w,Pgrid(k,:),'linewidth',2.5);
    set(hand,'color',clr(k,:));
end
title('Power variation with frequency','fontsize',20,'fontweight','bold');
xlabel('\omega in rad/s','fontsize',16,'fontweight','bold');
ylabel('Power in watts','fontsize',16,'fontweight','bold');
set(gca,'fontsize',14,'fontweight','bold');
xlim([5000 10000]);
grid on;
str=[];
for k=1:1:length(R)
    str=[str;cellstr(strcat('R=',num2str(R(k)),'\Omega'))];
end
legend(str);
```

Circuit Diagram

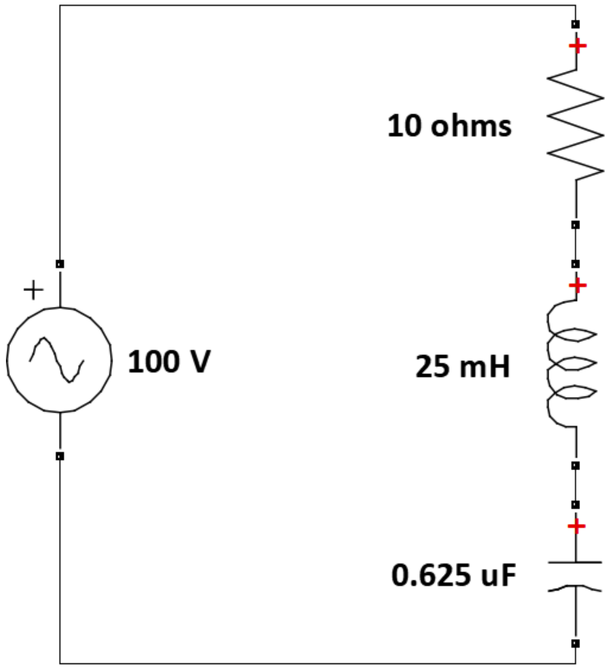

Fig. 7.1: Series RLC circuit to study resonance

Observations

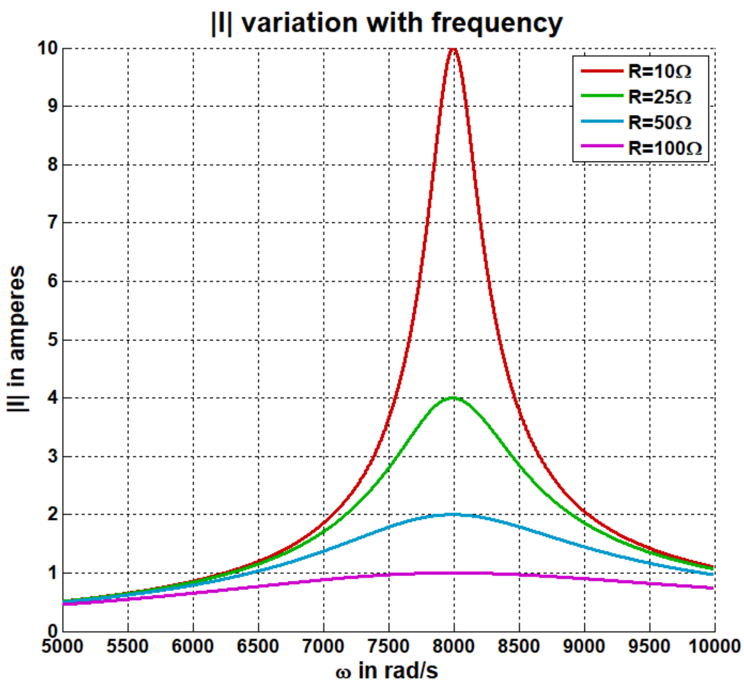

Fig. 7.2: Variation in $|\bar{I}|$ with the source frequency

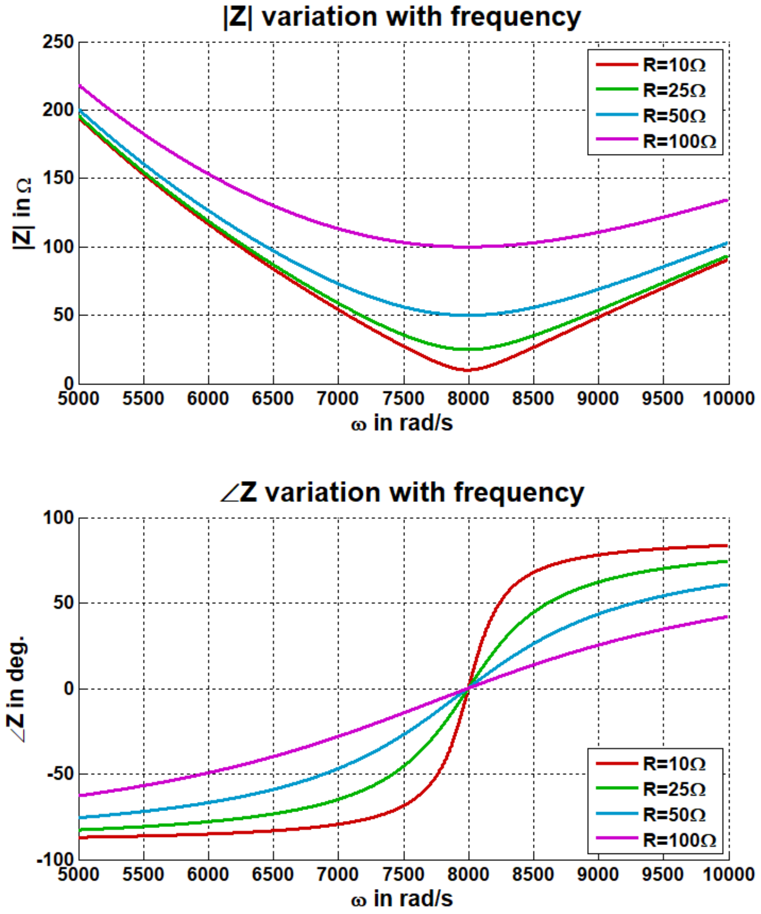

Fig. 7.3: Variation of impedance with the source frequency

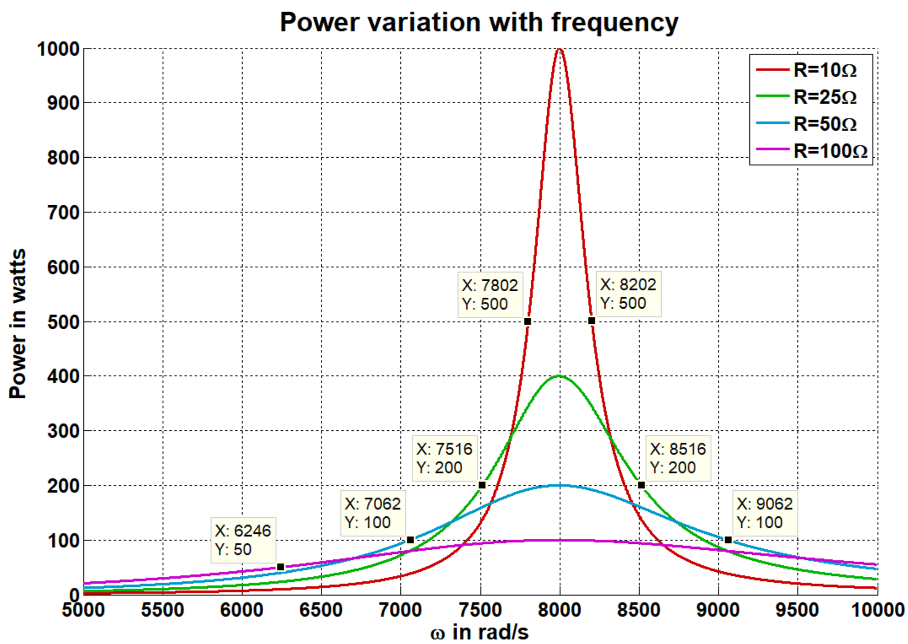

Fig. 7.4: Variation in power transfer with the source frequency

Table 7.1: Computing bandwidth from the half-power frequencies obtained from Fig. 7.4

R in Ω	ω_r in rad/s	ω_1 in rad/s	ω_2 in rad/s	$Bandwidth$ in rad/s
10	8000	7802	8202	400
25	8000	7516	8516	1000
50	8000	7062	9062	2000
100	8000	6246	10246	4000

Result

From Fig. 7.2 and Fig. 7.4, it can be seen that during the series resonance maximum rms current flows and maximum average power is transferred. From Fig. 7.3, it is seen that during series resonance the impedance is minimum and is equal to resistance of the circuit, and the power factor is unity. Table 7.1 shows that as the resistance of the circuit increases, the bandwidth also increases and selectivity decreases.

Experiment 8

To determine equivalent Z-parameters of two cascaded two-port networks.

Theory

A port is a pair of terminals through which a current may enter or leave a network. A two-port network contains two separate ports for input and output. A large and complex network may be divided into two-port subnetworks for the purposes of analysis and design. Impedance or Z - parameters are used in the synthesis of filters, analysis of impedance-matching networks and power distribution networks. The Z-paramters of a two-port network are given as,

$$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix}$$

Procedure

1. Open a simulink model and ad blocks of *resitor, current-controlled voltage source, dc voltage source, current sensor* and *voltage sensor* from *Foundation Library* of *Simscape*, as shown in Fig. 8.1
2. To use *Divide* and *Display* blocks of *Simulink* library, the *current* and *voltage sensors* signals are to be passed through *PS-Simulink Converter* block from *Simscape* → *Utilities*.
3. Electrical reference is must. Add and connect *Solver Configuration* block anywhere in the circuit from *Simscape* → *Utilities*.
4. The *Display* will display Z_{11} computed value.
5. In Fig. 8.1 to compute Z_{11}, the Port 2 is kept open-circuited and 1V source is applied at the Port 1. Measure voltage and current at Port 1 and divide to get Z_{11}.

6. Similarly, determine Z_{12} by applying 1V source at Port 2 and keep Port 1 open. To get Z_{12} measure voltage at Port 1 and current at Port 2 as shown in Fig. 8.2.
7. Similarly, obtain Z_{21} and Z_{22} as per Fig. 8.3 and 8.4.
8. Obtain Z-parameters for network 2 as described in Fig. 8.5-8.8.
9. For cascaded network, make subsytems of network 1 and network 2. Network 1 subsystem is shown in Fig. 8.9. Set the *Port location on parent subsystem* to either *left* or *right* by double clicking on the *conn port* block as per the Fig. 8.10.
10. Now obtain Z-parameters of the cascaded system as obtained in Fig. 8.10-8.13.

Observations

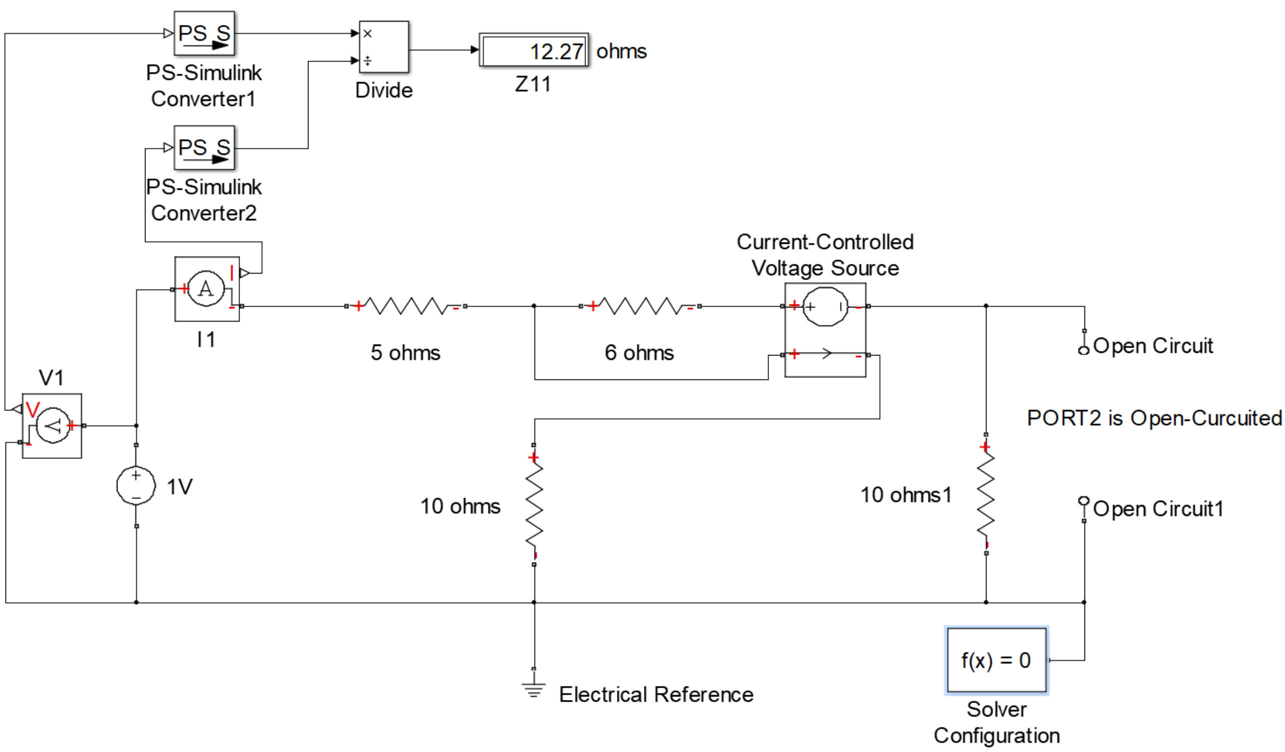

Fig. 8.1: Simulink model to obtain Z_{11} of network 1

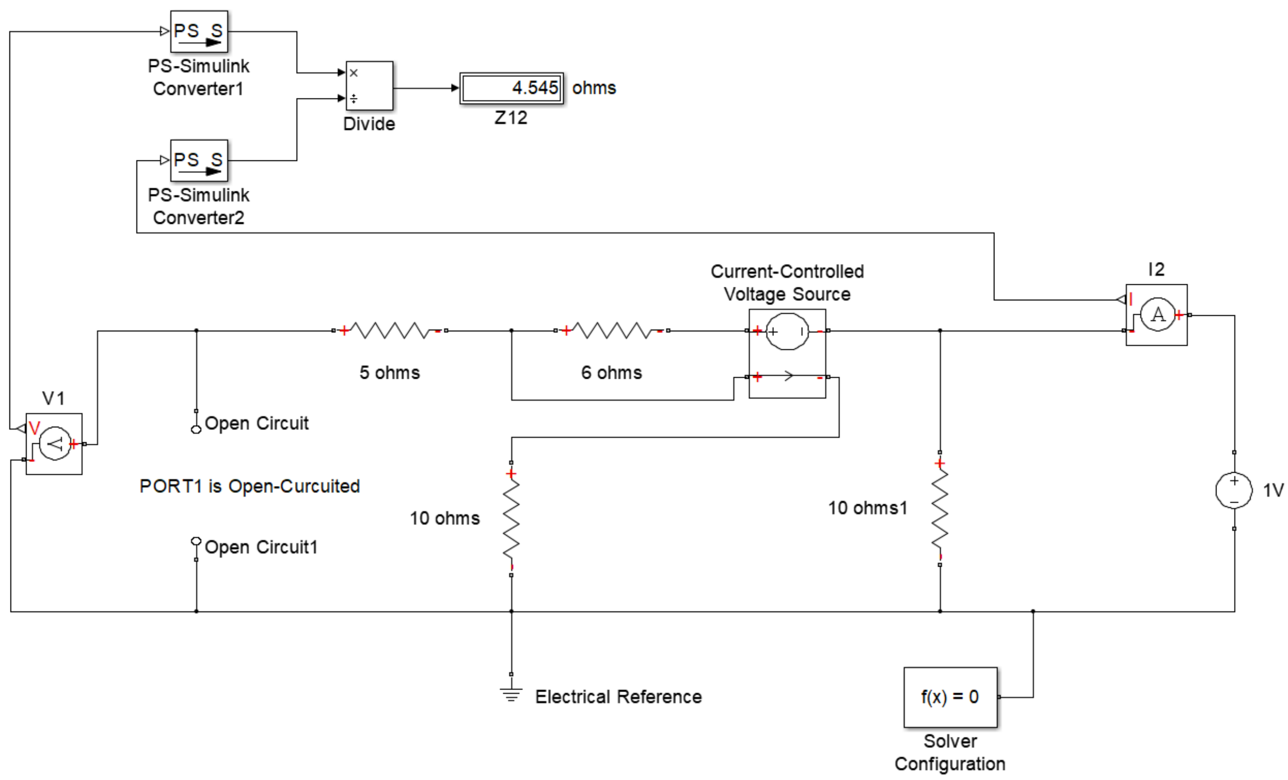

Fig. 8.2: Simulink model to obtain Z_{12} of network 1

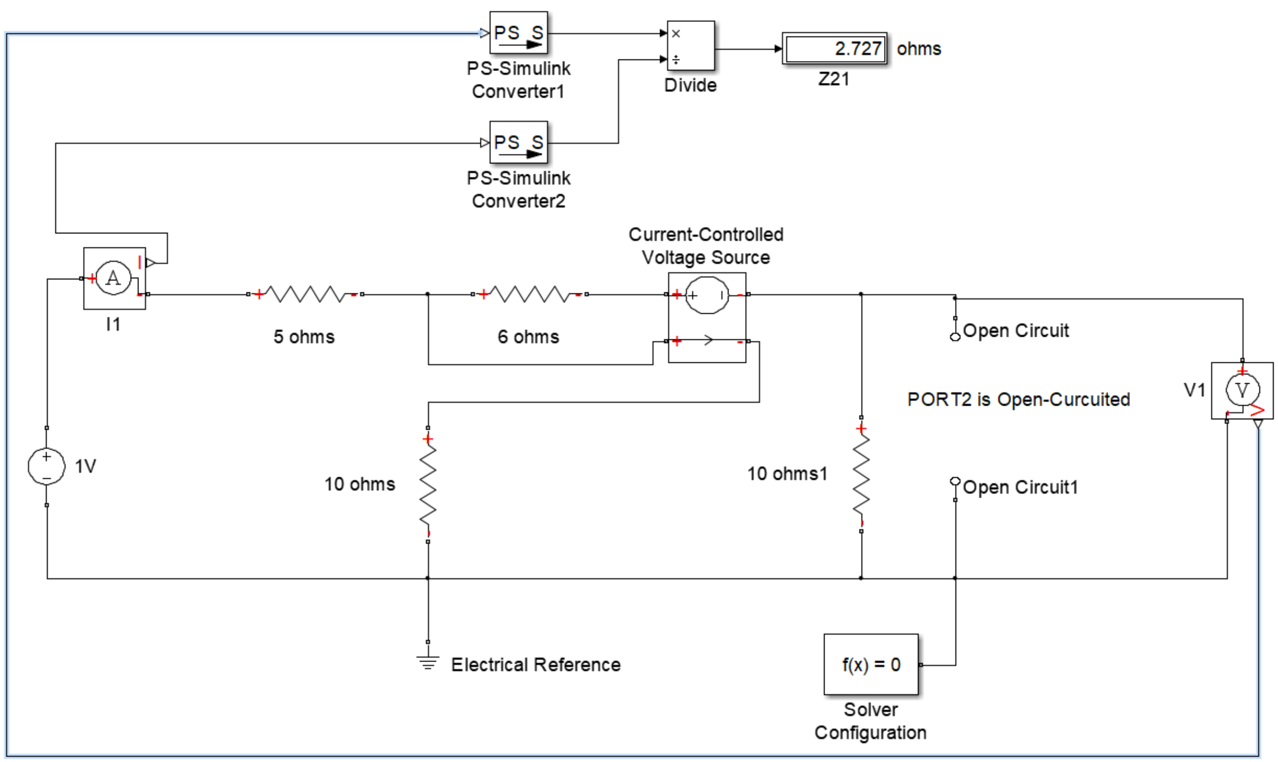

Fig. 8.3: Simulink model to obtain Z_{21} of network 1

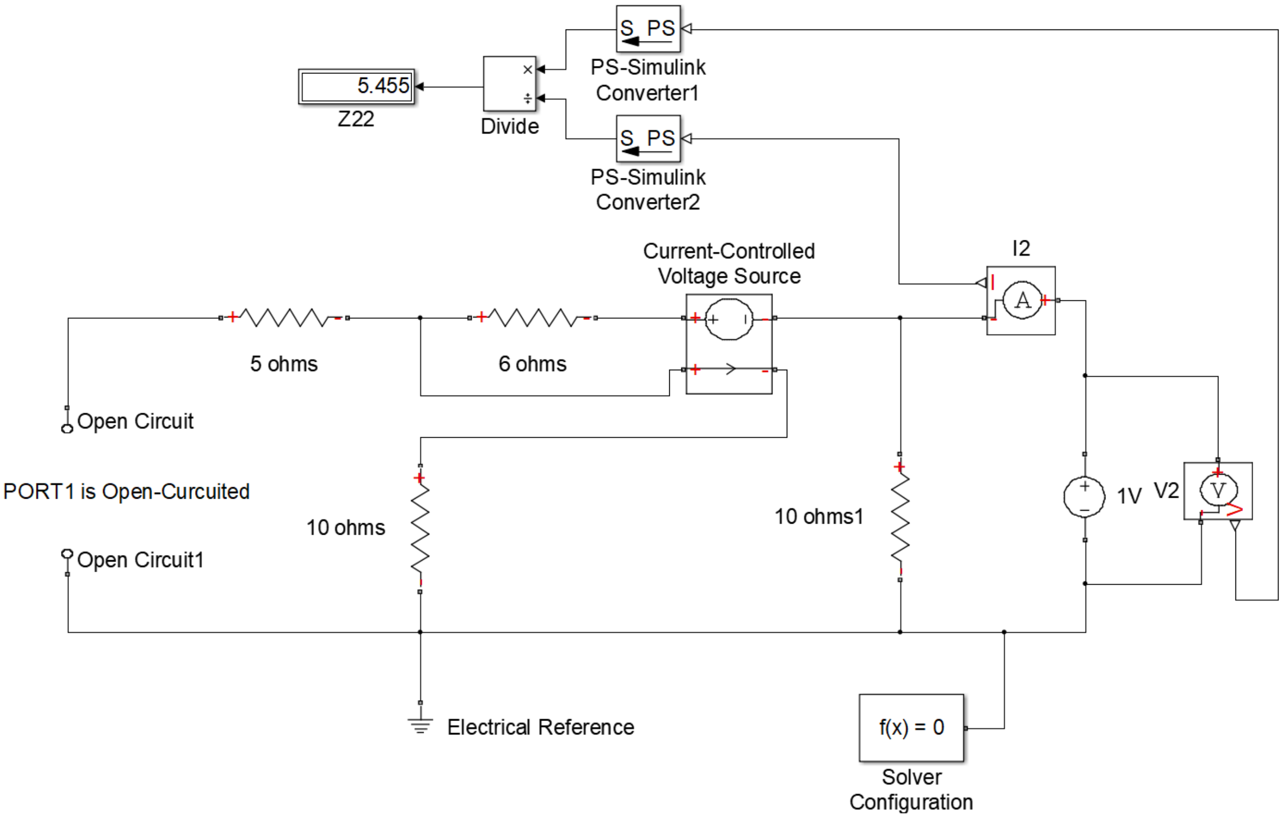

Fig. 8.4: Simulink model to obtain Z_{22} of network 1

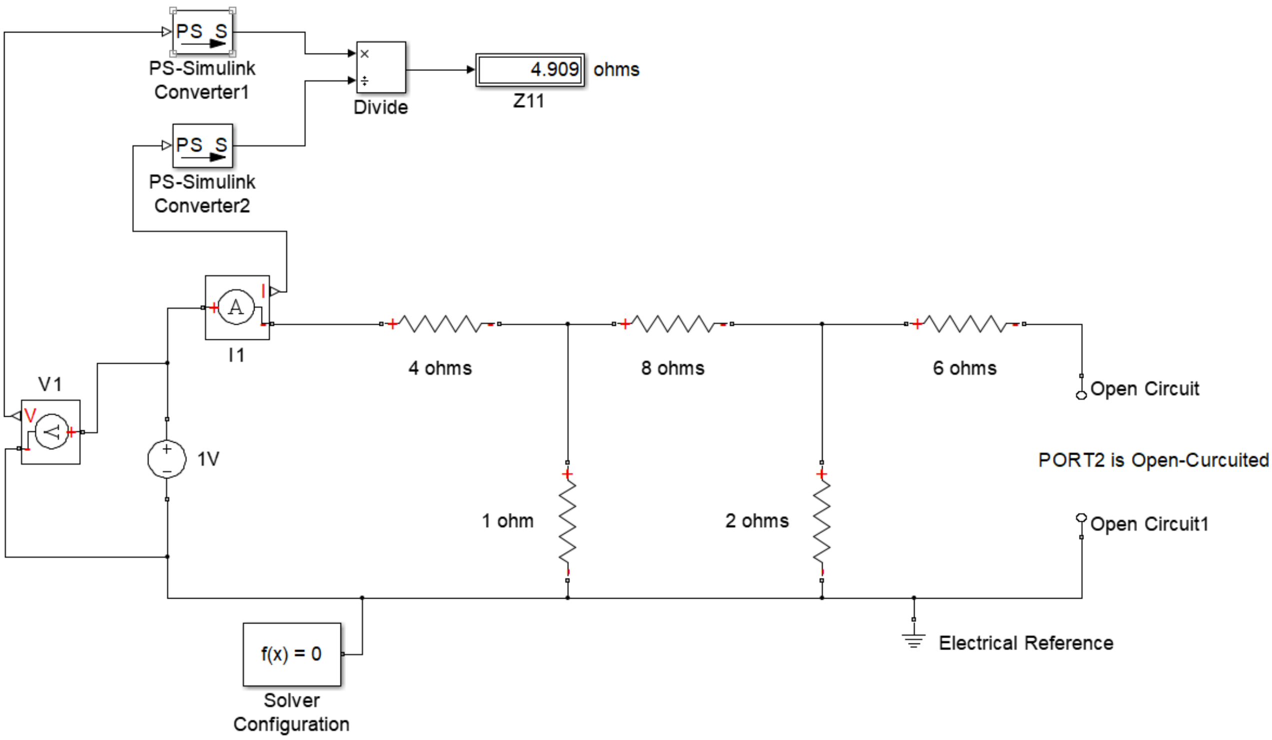

Fig. 8.5: Simulink model to obtain Z_{11} of network 2

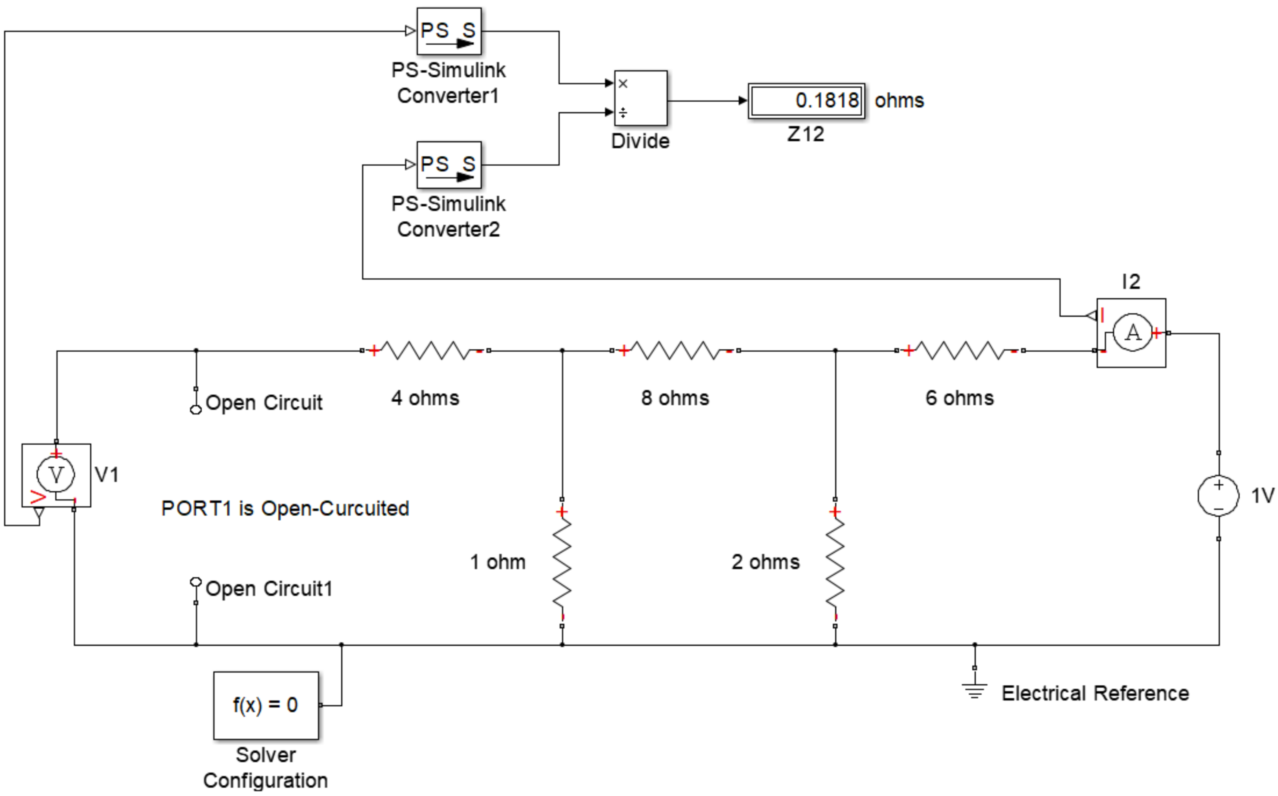

Fig. 8.6: Simulink model to obtain Z_{12} of network 2

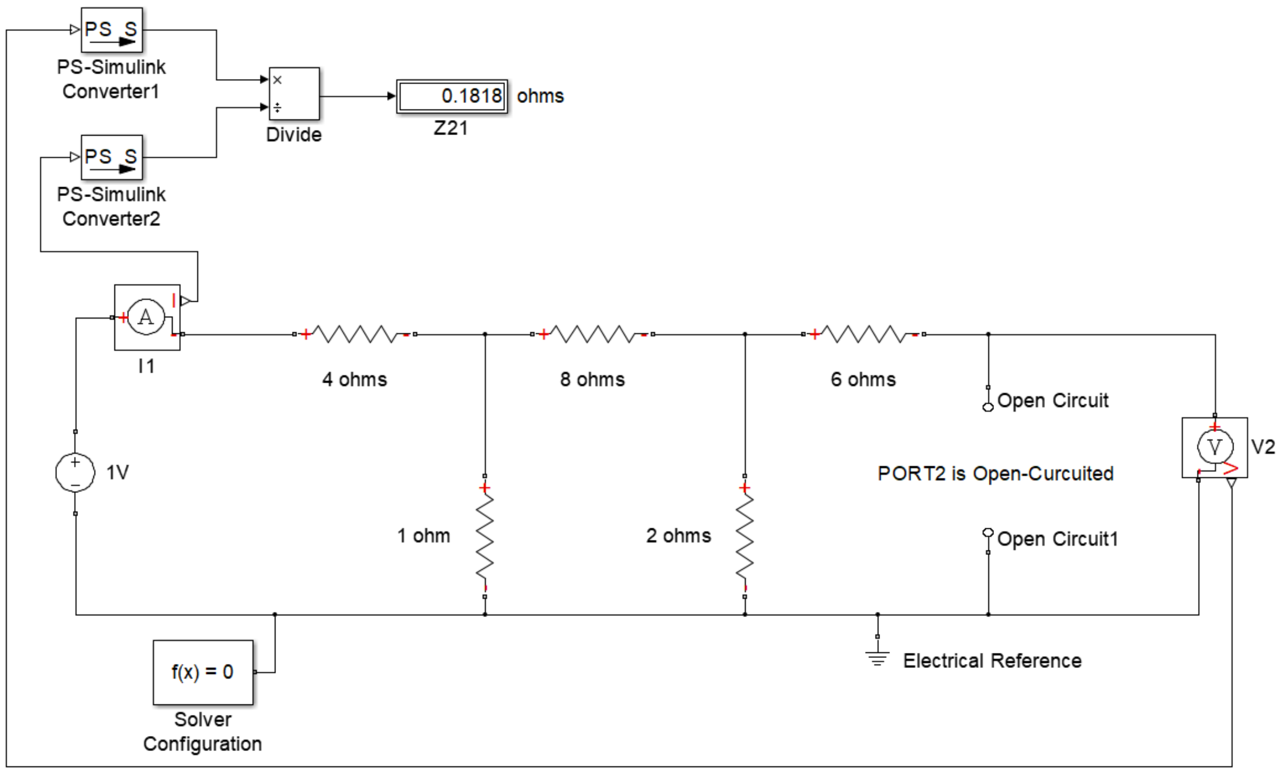

Fig. 8.7: Simulink model to obtain Z_{21} of network 2

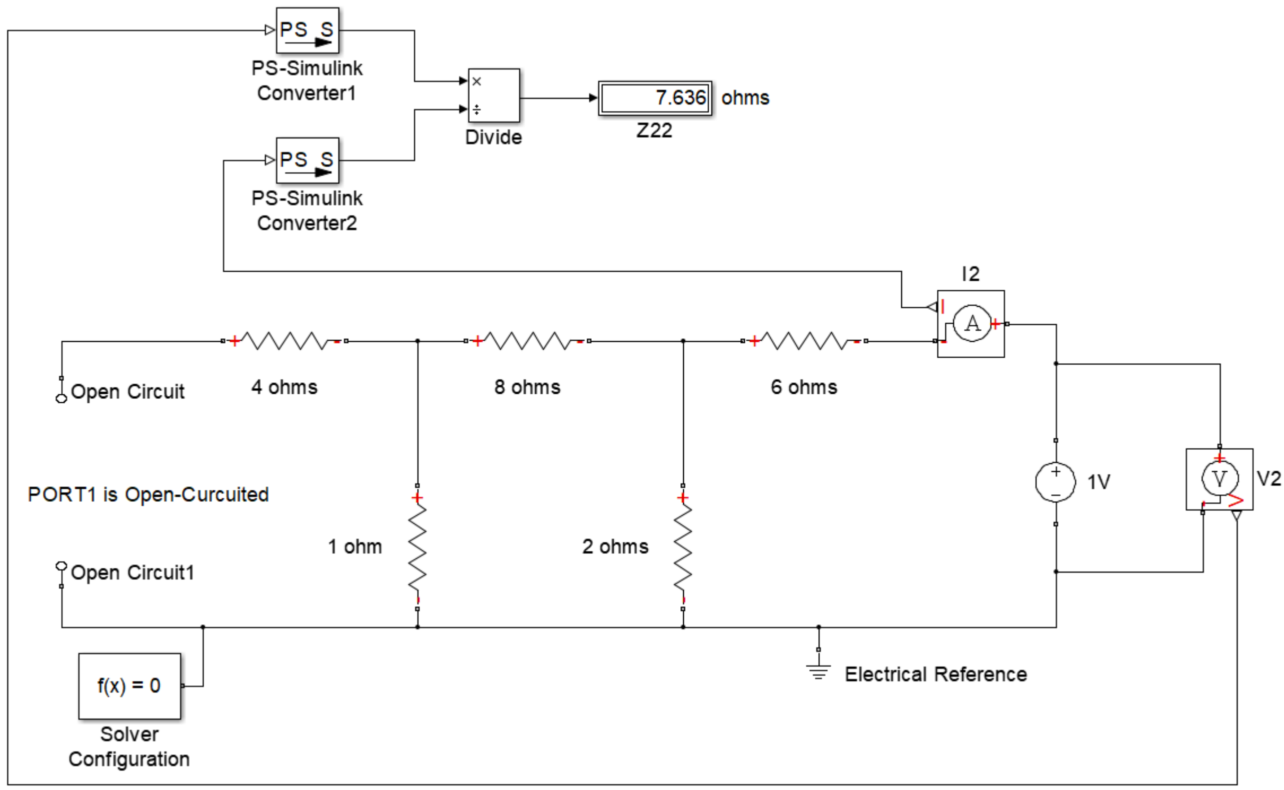

Fig. 8.8: Simulink model to obtain Z_{22} of network 2

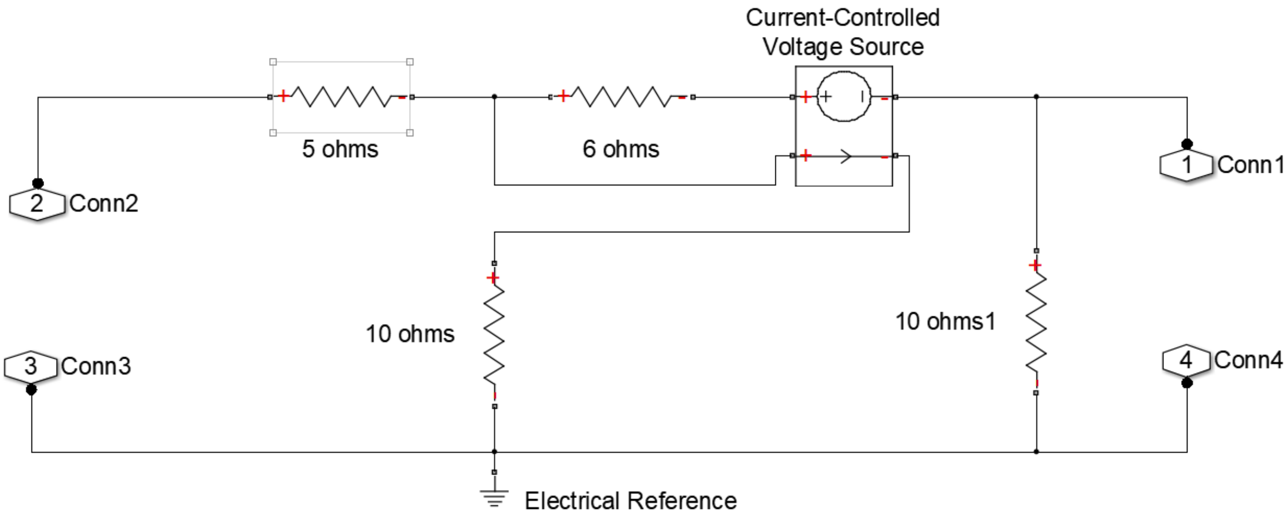

Fig. 8.9: Subsystem for network 1

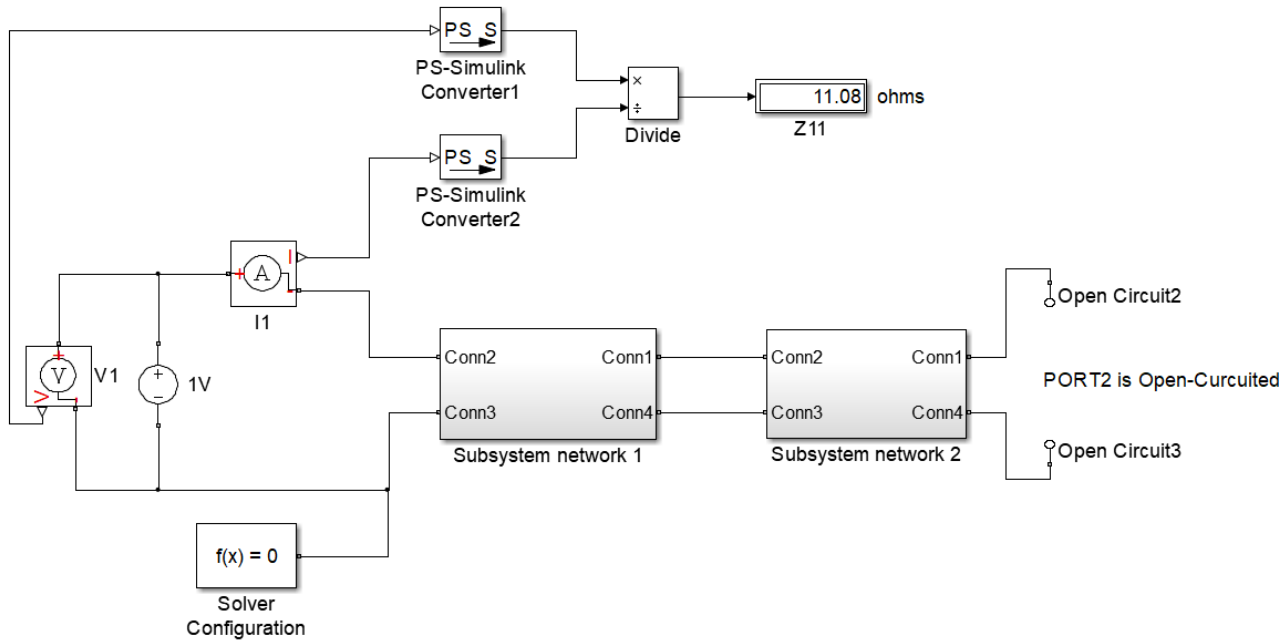

Fig. 8.10: Simulink model to obtain Z_{11} of cascaded network

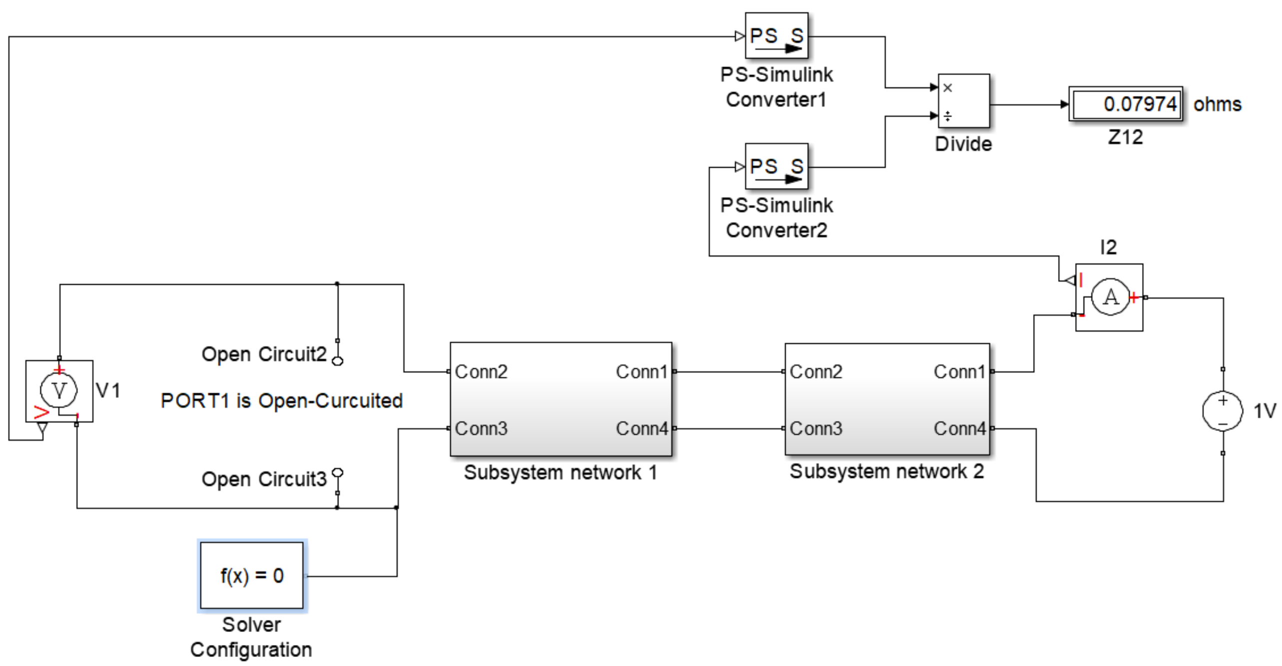

Fig. 8.11: Simulink model to obtain Z_{12} of cascaded network

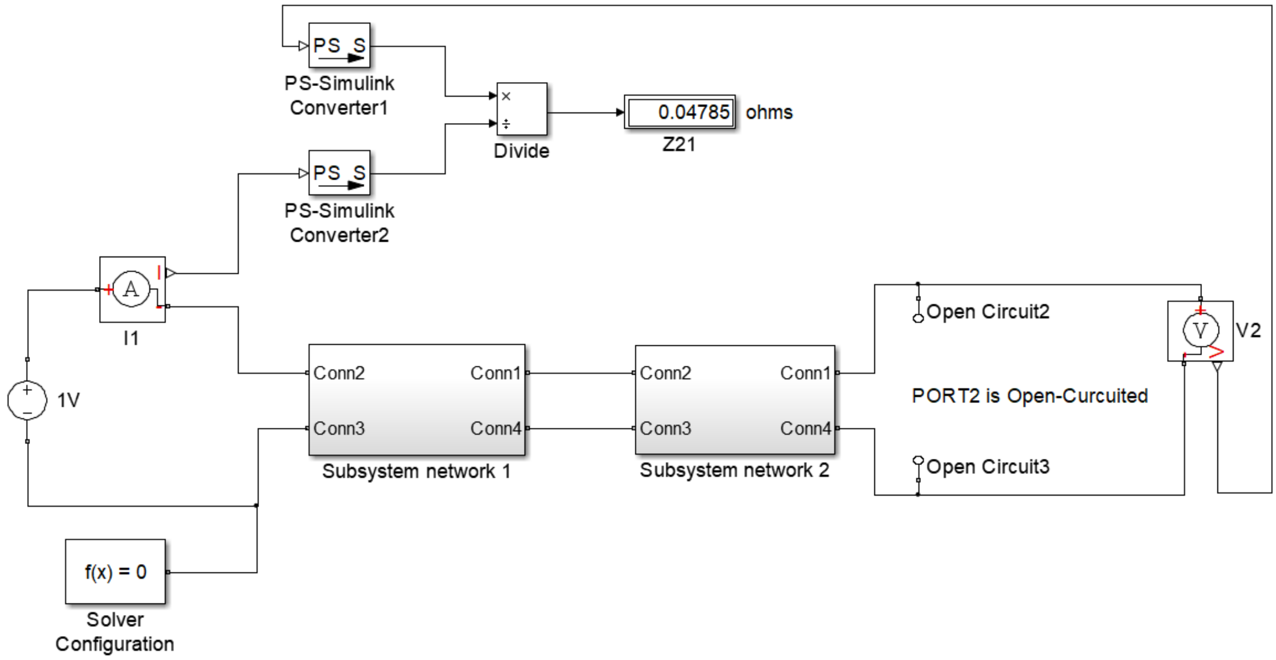

Fig. 8.12: Simulink model to obtain Z_{21} of cascaded network

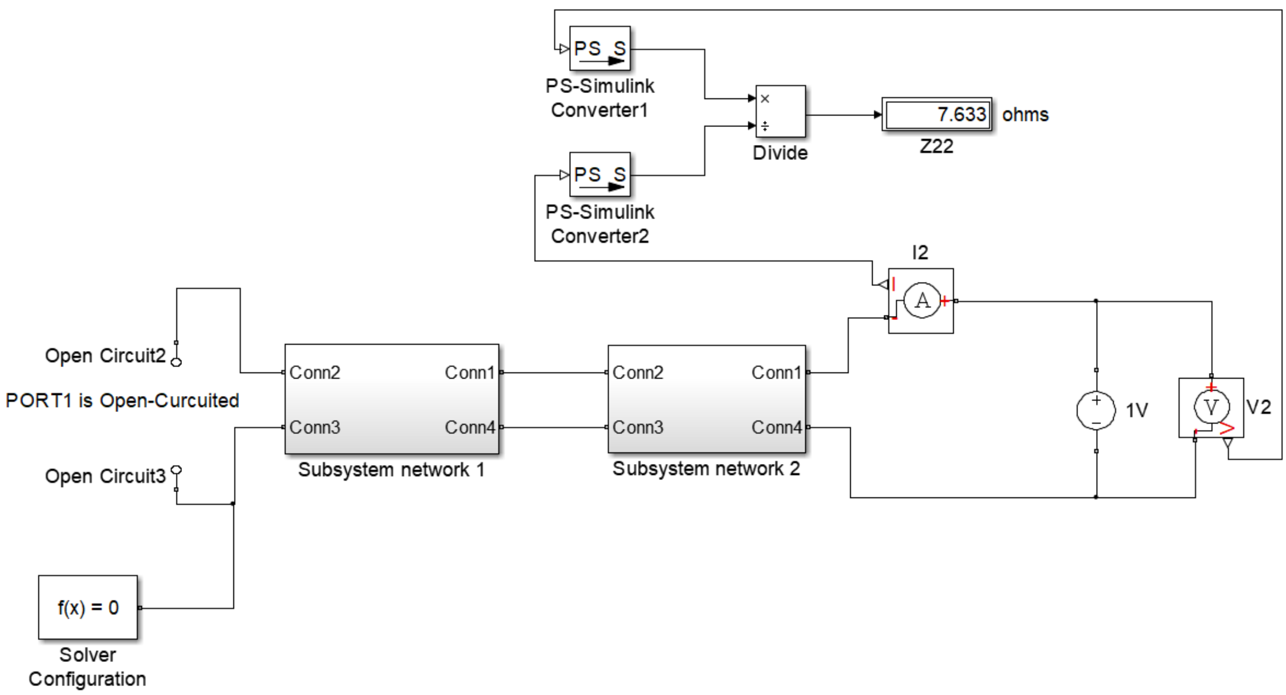

Fig. 8.13: Simulink model to obtain Z_{22} of cascaded network

$$network\ 1 : [Z] = \begin{bmatrix} 12.27 & 4.545 \\ 2.727 & 5.455 \end{bmatrix}$$

$$network\ 2 : [Z] = \begin{bmatrix} 4.909 & 0.1818 \\ 0.1818 & 7.636 \end{bmatrix}$$

$$cascaded\ network : [Z] = \begin{bmatrix} 11.08 & 0.07974 \\ 0.04785 & 7.633 \end{bmatrix}$$

Result

The equivalent Z-parameters of two cascaded two-port networks are obtained using *simulink*.

Experiment 9

To synthesize a network of a given network function as Foster-II form network and verify its response.

Theory

Network synthesis is a design technique for linear electrical circuits involving resistors, inductors and capacitors. Network synthesis techniques are used to synthesise network filters, impedance-matching networks, directional couplers, equalizers etc.

Procedure

1. Add *Transfer function* block from *Continuous* sublibrary of *Simulink*. Pass the driving-point admittance function to it in matrix form by double clicking this block.

2. Connect *step* input to it, and set its final value to 10 and step time to 0.

3. Using *scope* block obtain the waveform by configuring the *Relative tolerance* of *solver* to 10^{-6} (very small).

4. Obtain poles and residues of driving-point admittance function using MATLAB. The results of poles and residues are presented in Fig. 9.1.

5. Determine the Foster-II form network components and realize the network as shown in Fig. 9.2 to obtain the current waveform.

6. Compare the waveforms obtained from transfer function and from Foster-II form circuit.

Program

```
%Fidning the residues and poles of the driving point admittance function
%Y(s)=A(s)/B(s)
%A(s)=s^3+9s
%B(s)=10s^4+200s^2+640

A=[1 0 9 0];
B=[10 0 200 0 640];

[res,poles,k]=residue(A,B);
disp('residues are');
disp(res);
disp('corresponding poles are');
disp(poles);
disp('poles at infinity terms are');
if isempty(k)
    disp('nil');
else
    disp(k);
end
```

Observations

```
>> res_pole
residues are
   0.0292 + 0.0000i
   0.0292 - 0.0000i
   0.0208 - 0.0000i
   0.0208 + 0.0000i

corresponding poles are
  -0.0000 + 4.0000i
  -0.0000 - 4.0000i
   0.0000 + 2.0000i
   0.0000 - 2.0000i

poles at infinity terms are
nil
>>
```

Fig. 9.1: Residue and poles of the given transfer function as obtained using MATLAB program

$$Y = \frac{0.0208}{s - j2} + \frac{0.0208}{s + j2} + \frac{0.0292}{s - j4} + \frac{0.0292}{s + j4}$$

$$or,\ Y = \frac{0.0416s}{s^2 + 4} + \frac{0.0584s}{s^2 + 16}$$

$$or,\ Y = \frac{1}{24s + \frac{1}{0.01042s}} + \frac{1}{17.14s + \frac{1}{0.00365s}}$$

$$\therefore\ L_1 = 24\ H,\ C_1 = 0.01042\ F,\ L_2 = 17.14\ H,\ C_2 = 0.00365\ F$$

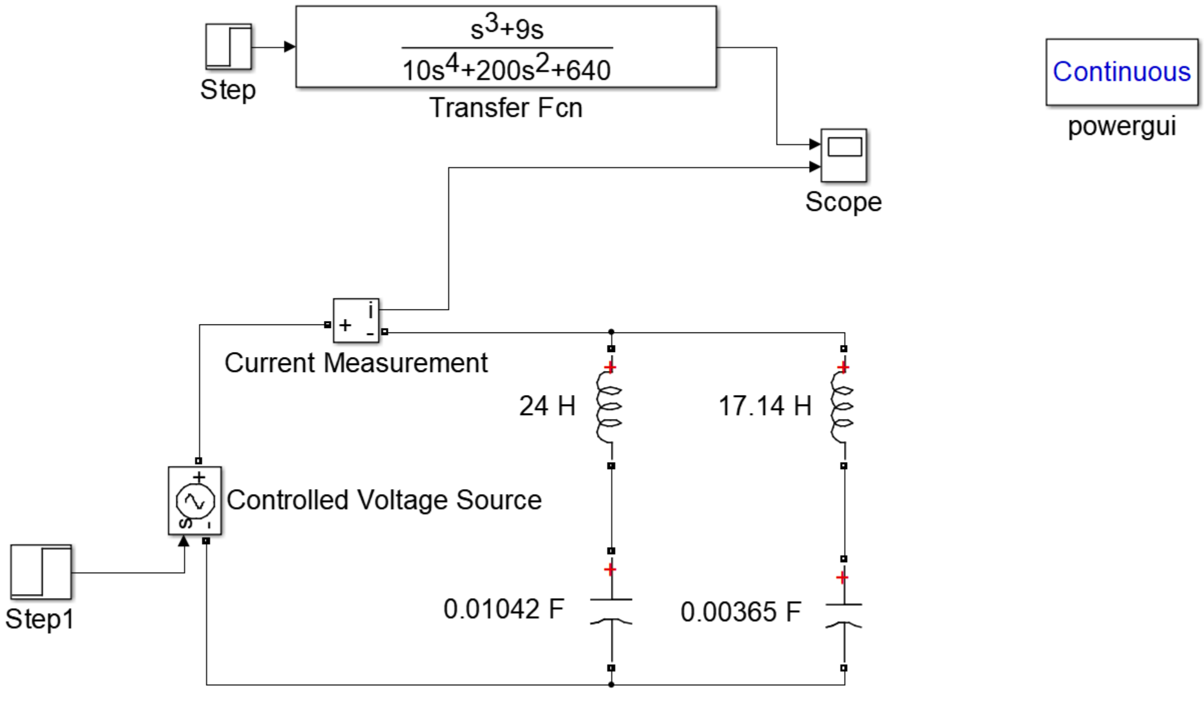

Fig. 9.2: Driving-point admittance function step response and its Foster-II form realization

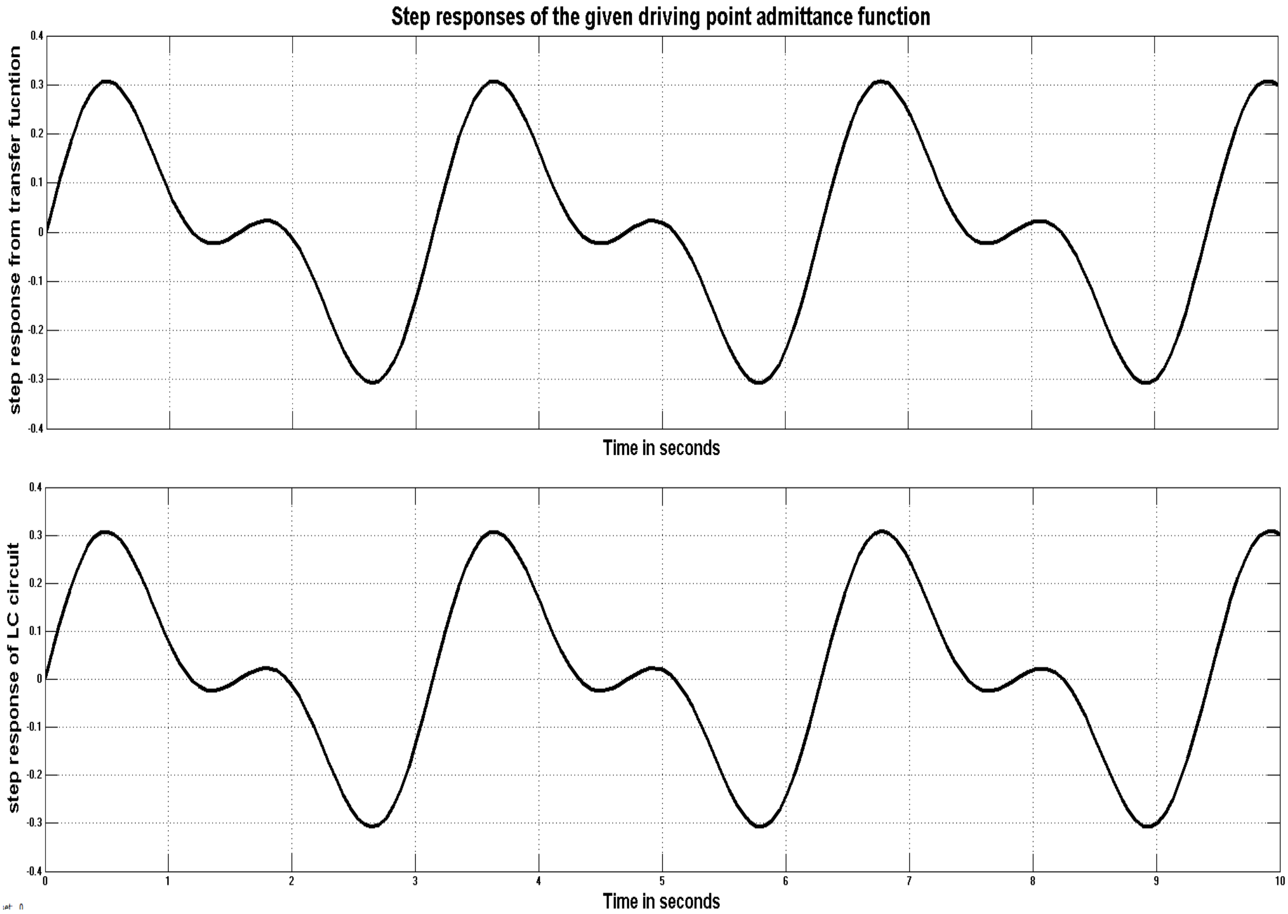

Fig. 9.3: Waveforms of step response of given driving-point admittance function and its corresponding Foster-II form circuit

Result

The step responses of given driving-point admittance function using transfer function approach and using Foster-II form realized network are same as presented in Fig. 9.3, hence response of synthesized Foster-II form network has been verified.

Experiment 10

To synthesize a network of a given network function as Cauer form network and verify its response.

Theory

Network synthesis is a design technique for linear electrical circuits involving resistors, inductors and capacitors. Network synthesis techniques are used to synthesise network filters, impedance-matching networks, directional couplers, equalizers etc.

Procedure

1. Add *Transfer function* block from *Continuous* sublibrary of *Simulink*. Pass the driving-point admittance function to it in matrix form by double clicking this block.
2. Connect *step* input to it, and set its final value to 10 and step time to 0.
3. Using *scope* block obtain the waveform by configuring the *Relative tolerance* of *solver* to 10^{-6} (very small).
4. Obtain continuous fraction expansion of given driving-point admittance function using MATLAB program.
5. Determine the Cauer-I form network components as shown in Fig. 10.1.
6. Now realize the Cauer-I form network and obtain the waveform of current as shown in Fig. 10.2 to obtain the current waveform.
7. Compare the waveforms obtained from transfer function and from Cauer-I form circuit.

```
% Program to find the continuous fraction of driving point admittance
% function Y(s)=A(s)/B(s) and hence determine Cauer-I form circuit
% A(s)=s^3+9s    % B(s)=10s^4+200s^2+640
A=[1 0 9 0];
B=[10 0 200 0 640];
quo=[];
while 1
    [q,r]=deconv(A,B);
    %eliminating the right zeros from the quotient term
    while 1
        if q(length(q))~=0
            break;
        end
        if length(q)==1
            break;
        end
        q(length(q))=[];
    end
    quo=[quo q];    %storing qoutient in quo
    %eliminating the inital zeros from the remainder term
    if r==0
        break;
    end
    while 1
        if r(1)~=0
            break;
        end
        r(1)=[];
    end
    A=B;
    B=r;
end
disp('Quotients are');
disp(quo);
%displaying L and C values of Cauer-I form circuit
i=1;
for k=1:2:length(quo)
    fprintf(strcat('C',num2str(i),'='));
    fprintf('%5.4f',quo(k));
    fprintf(' F');
    fprintf('\n');
    i=i+1;
end
i=1;
for k=2:2:length(quo)
    fprintf(strcat('L',num2str(i),'='));
    fprintf('%5.4f',quo(k));
    fprintf(' H');
    fprintf('\n');
    i=i+1;
end
```

Observations

```
>> continuous_fraction
Quotients are
           0    10.0000    0.0091    34.5714    0.0050

C1=0.0000 F
C2=0.0091 F
C3=0.0050 F
L1=10.0000 H
L2=34.5714 H
>>
```

Fig. 10.1: Components of Cauer-I form network as obtained by decomposition of given driving-point admittance function through continuous fraction

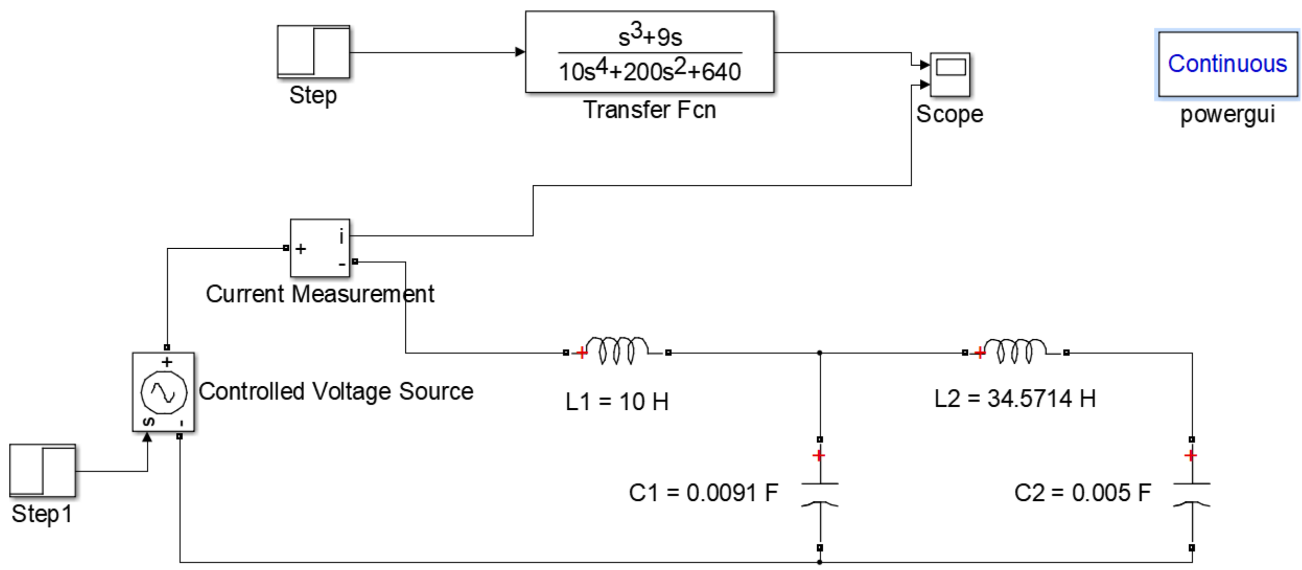

Fig. 10.2: Driving-point admittance function step response and its Cauer-I form realization

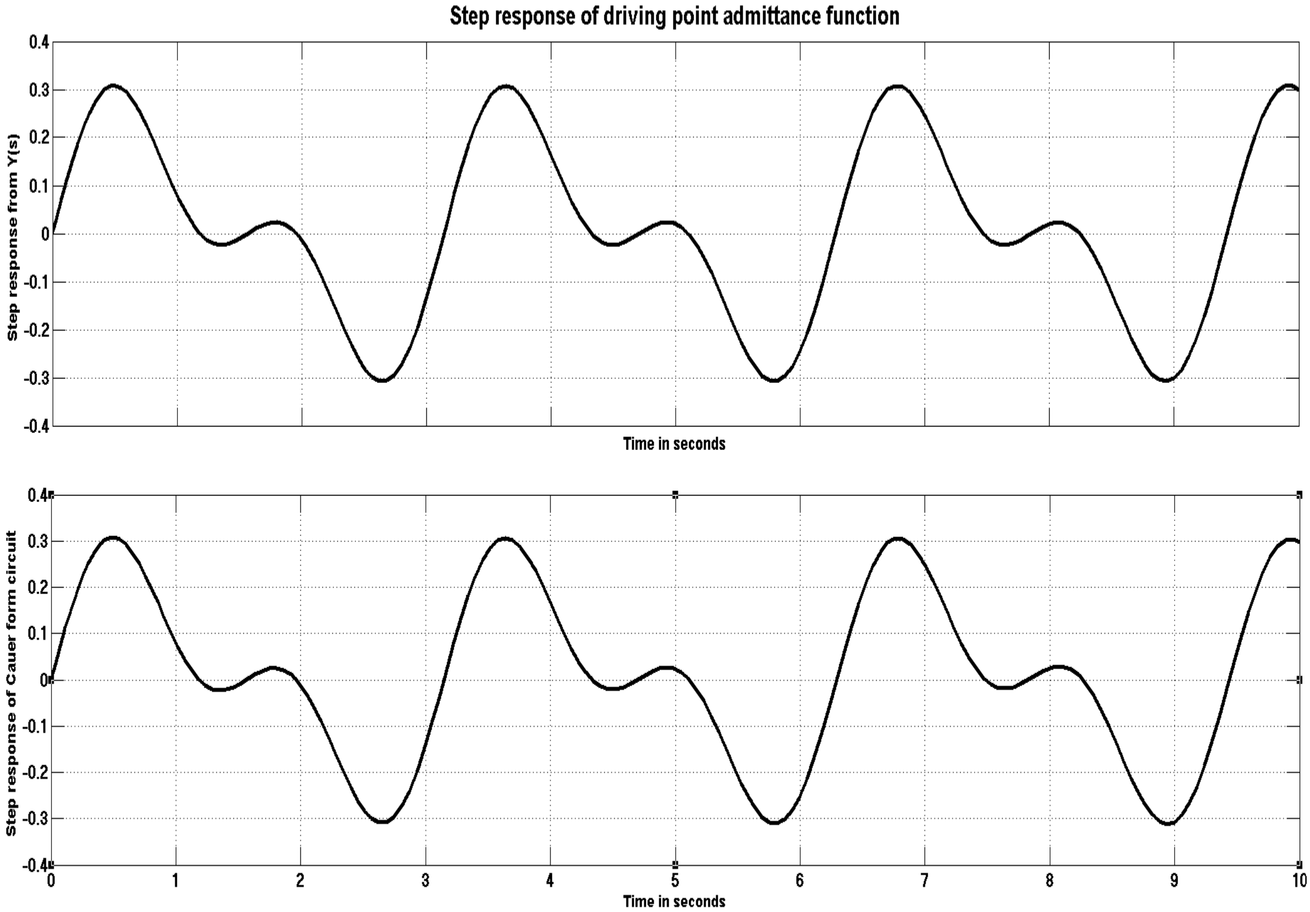

Fig. 10.3: Waveforms of step response of given driving-point admittance function and its corresponding Cauer-I form circuit

Result

The step responses of given driving-point admittance function using transfer function approach and using Cauer-I form realized network are same as presented in Fig. 10.3, hence response of synthesized Cauer-I form network has been verified.

Experiment 11

To verify superposition theorem.

Theory

Superposition theorem states that in any linear bilateral circuit consisting of two or more independent current or voltage sources, the current through any branch or voltage across any branch is equal to the algebraic sum of current through or voltage across that branch considering one source at a time and replacing the other sources with their internal resistance. This theorem is widely used in the circuits consisting of different frequencies sources.

Circuit Diagram

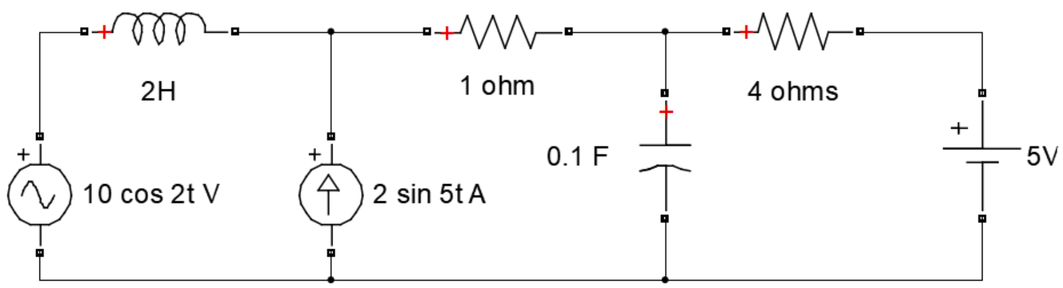

Fig. 11.1: Circuit Diagram to verify superposition theorem

Procedure

1. Add the various components into the model as per the circuit diagram given in Fig. 11.1 from *Simpowersystem* sub-library of *Simscape*.
2. Insert *Powergui* block and set its solver at continuous mode.
3. To find the voltage across 1Ω resistor, add the voltage measurement block from the *Measurement* sublibrary of *Simpowersystem* across the 1Ω resistor

and a *scope* block from *Simulink* library as shown in Fig. 11.2.

4. Open the settings of the *Scope* block, click on *History* menu and check the box *Save data to workspace.*

5. Run the simulation. Obtain the plot of the voltage by double clicking on the *scope* block. Note that at least one cycle of measured voltage should be there, otherwise increase the simulation time. Note down the time period of this voltage wave.

6. Now to perform the Fourier analysis on the measured voltage double click on *powergui* block and select *FFT Analysis* tool.

7. Under *Available signals* select the signal of the measured voltage which should be *ScopeData* unless it was changed during the settings done in *scope* history.

8. Under *FFT settings* Select *no. of cycles* to as per need or 1, *Fundamental frequency* to 1/time period of the voltage wave (i.e. the frequency of the input signal) and *Display style* to 'List relative to specified base'. Set the *Specified base* to 1, so that further calculations with the amplitudes of the harmonics will not be required.

9. Then click on *Display.* It will generate the amplitude and the phase angle of harmonics as shown in Fig. 11.3.

10. From here determine the voltage appearing across 1Ω resistance.

11. Now apply superposition theorem on the circuit by considering one source at a time and replacing the other sources with their internal resistance, and measure the voltage across 1Ω resistor as shown in Fig. 11.4-11.6.

12. For alternating source, the *powergui solver* is set at *phasor* with the frequency as per the source, and for dc source set *powergui solver* to *continuous.*

Observations

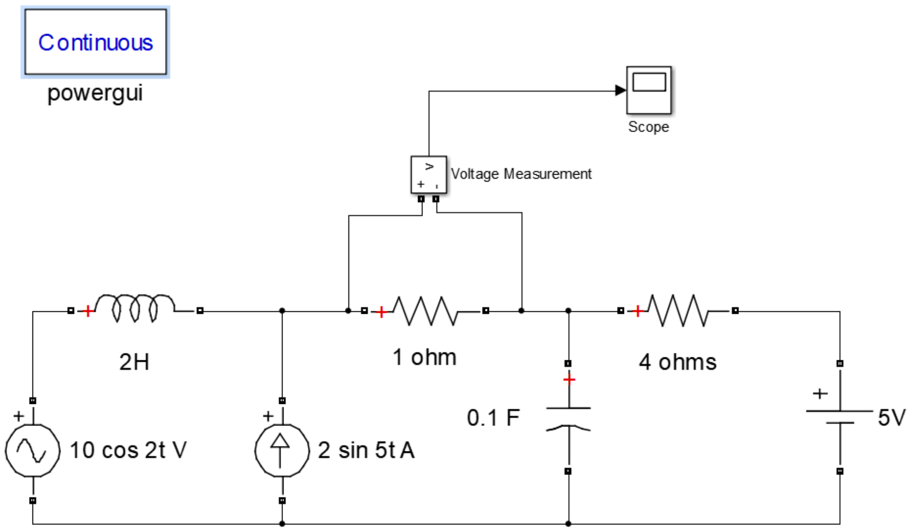

Fig. 11.2: Model to simulate the circuit

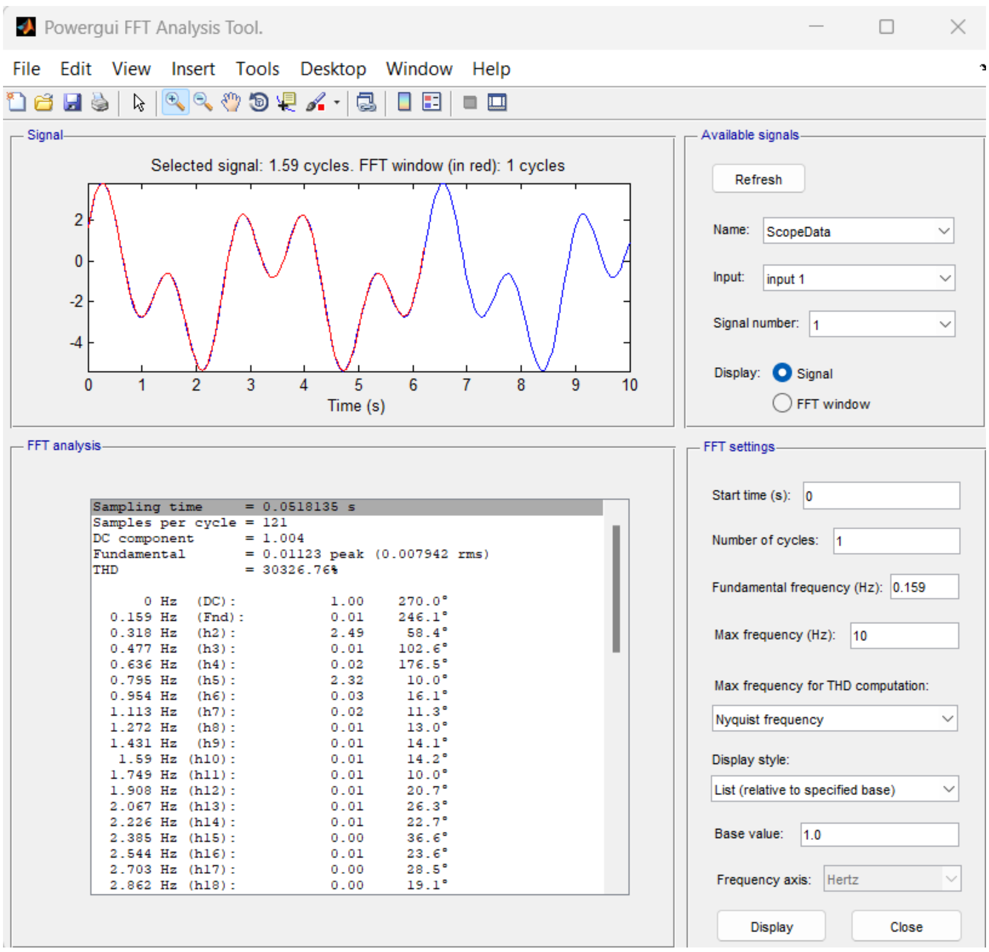

Fig. 11.3: Result of FFT analysis

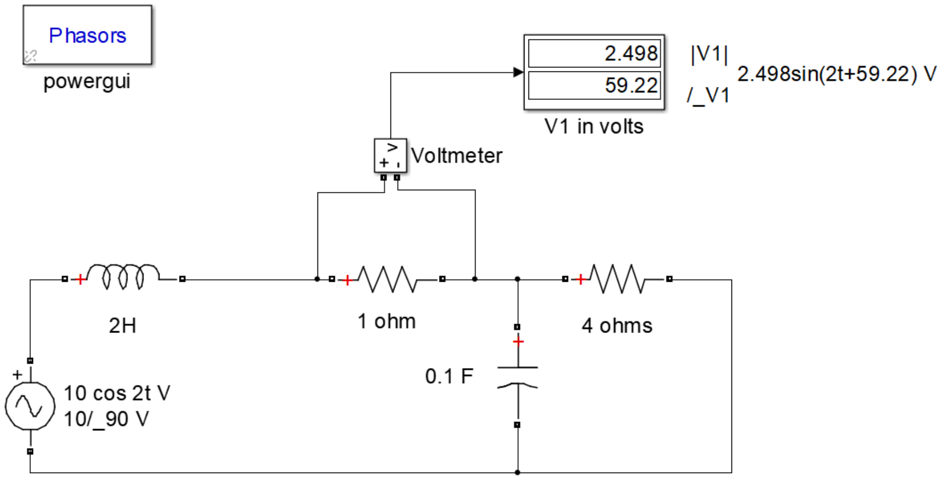

Fig. 11.4: Voltage across 1Ω resistor with source 1 acting alone

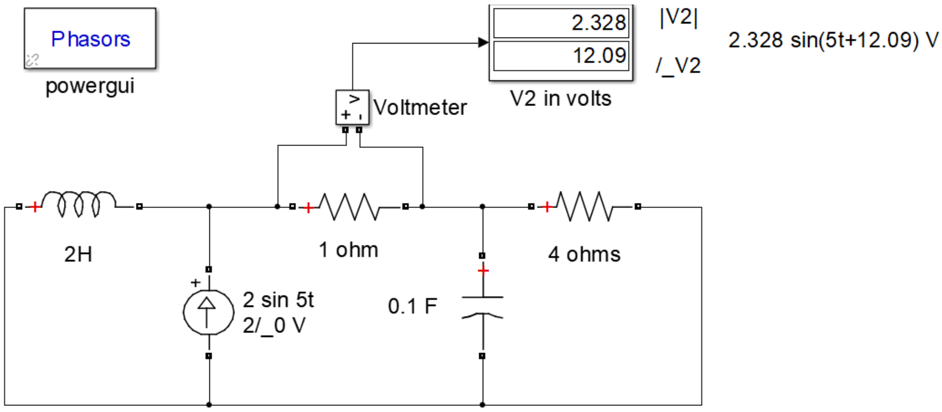

Fig. 11.5: Voltage across 1Ω resistor with source 2 acting alone

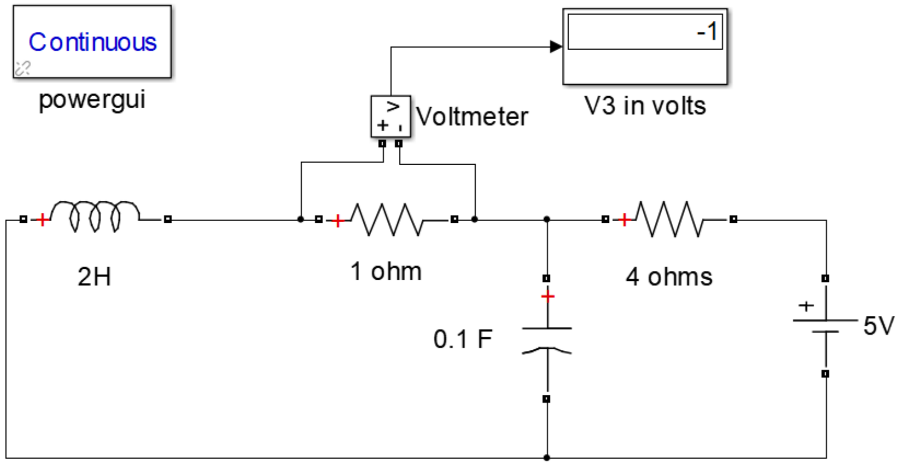

Fig. 11.6: Voltage across 1Ω resistor with source 3 acting alone

From *FFT*, voltage across 1Ω resistor,

$$V = 2.49sin\left(2t + 59.4^o\right) + 2.32sin\left(5t + 10^o\right) - 1\ volts$$

From *Superposition theorem*, voltage across 1Ω resistor,

$$V = 2.498sin\left(2t + 59.22^o\right) + 2.328sin\left(5t + 12.09^o\right) - 1\ volts$$

Result

The voltage across the 1Ω resistor obtained applying FFT on the circuit and applying superposition theorem are nearly same, thus the superposition theorem is verified.