INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal - 500 043, Hyderabad, Telangana

COURSE CONTENT

NETWORK ANALYSIS AND SYNTHESIS LABORATORY								
II Semester: ECE / EEE								
Course Code	**Category**	**Hours / Week**			**Credits**	**Maximum Marks**		
AEEE05	Core	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		0	0	2	1	40	60	100
Contact Classes: Nil	**Tutorial Classes: Nil**	**Practical Classes: 36**				**Total Classes: 36**		
Prerequisite: Electrical Circuits, Linear Algebra and Calculus								

I. COURSE OVERVIEW:

The Network Analysis and Synthesis Laboratory provide hands-on experience in analyzing, designing, and testing electrical networks. Students learn to apply fundamental circuit theories, network theorems, and synthesis techniques to real-time electrical and electronic circuits. It is designed to give hands-on experience on virtual instrumentation through digital simulation techniques. These techniques enable the students in examining characteristics of DC and AC circuits, filters, solution of differential equation, generation of three phase and complexwave forms using MATLAB.

II.COURSE OBJECTIVES:

The students will try to learn:

I. The RL, RC, and RLC circuits under different excitations using integro-differential and Laplace transform approaches.

II. The two-port electrical networks and apply network topology concepts like tie-set and cut-set matrices for circuit analysis.

III. The various filter and attenuator configurations for frequency-selective and signal conditioning applications.

IV. The electrical networks using driving-point functions, positive real functions, and classical synthesis methods such as Foster and Causer forms.

III.COURSE OUTCOMES:

After successful completion of the course, students should be able to:

CO1	Simulate and analyze electrical circuits to verify resonance phenomena, network theorems, and power measurement techniques.
CO2	Evaluate time and frequency domain responses of RL, RC, and filter circuits using circuit simulation tools.
CO3	Determine and interpret two-port network parameters for diverse interconnection configurations.
CO4	Measure active, reactive, and three-phase power in balanced star and delta connected systems.
CO5	Analyze coupling effects in magnetically coupled circuits through coefficient, self, and mutual inductance calculations.
CO6	Validate classical theorems and compensation principles for optimizing circuit performance.

Dos

1) For safety purpose the students should compulsory wear leather shoes.
2) Students should come in uniform prescribed.
   i. For boys, half sleeve shirts, tucked in trousers
   ii. For ladies, half sleeve overcoat, hair put inside the overcoat
3) After giving connections, staff members should be asked to verify the circuit connections.
4) Before staring the circuit connections check whether the circuit breaker is in OFF condition.
5) Circuit should be switched ON only after getting permission from the staff member.
6) To be careful with moving parts in the machine.
7) To come prepared with procedure relevant to the experiment.
8) Unplug electrical equipment after use.

Dont’s

1) Don’t assume that the power is disconnected.
2) Don’t attempt to repair electrical equipment.
3) Don’t come with any ornaments when working with electrical machines.
4) Don’t use an earth connection as a neutral.
5) Don’t touch any parts unnecessarily.
6) Don’t keep any fluids and chemicals nearing instruments and circuits.

IV. COURSE CONTENT:

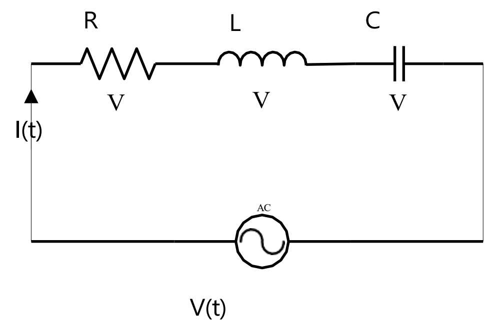
EXERCISES ON NETWORK ANALYSIS AND SYNTHESIS LABORATORY

Note: Students are encouraged to bring their own laptops for laboratory practice sessions.

1. Verification of Series and Parallel Resonance using any circuit simulation software.

1.1 Series resonance

Study the frequency response characteristics of series resonance circuit of Circuit – 19 and hence to determine the inductance, bandwidth and quality factor of the circuit. Where R = 1 kΩ, L = 10 mH, C = 0.1 μF.



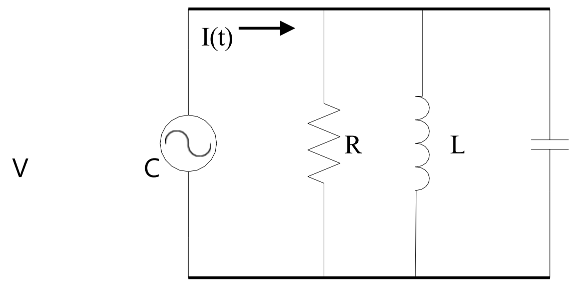
Circuit – 1

Try

1. Use Simulink to simulate Circuit–19 and plot the waveforms V_R, V_L and V_C verses frequency.
2. For series resonant circuit find I, V_R, V_L and V_C at resonance when V = 10 V, R = 2 Ω, X_L = 10 Ω and X_C = 10 Ω.
3. If the resonant frequency is 5000 Hz, find bandwidth.

1.2 Parallel resonance

Study the frequency response characteristics of parallel resonance circuit of Circuit – 20 and hence to determine the inductance, bandwidth and quality factor of the circuit. Where R = 1k Ω, L = 10 mH, C = 0.1 μF.



Circuit – 2

Try

1. Use Simulink to simulatiate parallel resonance circuit of circuit – 20 and plot magnitude of response vs frequency. Consider the frequency range 600 HZ to 2,500 HZ increments of 100 HZ and compare with the experimental plot

2. Determination of Time response of first order RL and RC circuit for periodic non – sinusoidal input- Time Constant and Steady-state error using any circuit simulation software.

Examine the time varying characteristics of series RL, RC and RLC circuits for given values of R, L and Cusing MATLAB software.

2.1 Time Response of Series RL

In this MATLAB program:

I. Circuit parameters (resistance, inductance, and applied voltage) are defined.

II. The time constant (τ) is calculated as L / R.

III. A symbolic variable **t** is defined for time.

IV. The differential equation representing the transient response of the RL circuit is defined using **diff** and ==.

V. The **dsolve** function is used to solve the differential equation symbolically.

VI. A time vector (**'t_span'**) is created for the simulation.

VII. IL(t) is numerically evaluated for the given time vector.

VIII. The transient response is plotted.

Hints

```
% Define circuit parameters
R = 100; % Resistance in ohmsL =
0.5; % Inductance in henrys
Vin = 10; % Applied voltage in volts

% Calculate the time constant τtau = L /
R;

% Define the symbolic variable for timesyms t

% Define the differential equation for the current IL(t)IL = sym('IL(t)');
eqn = L * diff(IL, t) + R * IL == Vin;

% Solve the differential equation IL_solution =
dsolve (eqn, IL); IL_solution=
simplify(IL_solution);

% Create a time vector and evaluate IL(t
time = 0:0.01:2*tau; % Adjust the time span as needed
IL_numeric = double (subs (IL_solution, t, time));

% Plot the transient responseplot
(time, IL_numeric); xlabel ('Time
(s)');
ylabel ('Current (A)');
title ('Transient Response of RL Circuit');grid on;
```

Try:

Circuit Parameters:

Given the following values:

I. Resistor (R): 150 ohms

II. Inductor (L): 0.3 H

III. Applied Voltage (V_in): 12

Calculate the time constant (τ) of the RL circuit.

Transient Analysis:

I. Write down the first-order differential equation governing the transient response of theRL circuit in terms of the current (I_L) through the inductor.

II. Solve the differential equation symbolically to obtain the expression for I_L(t) as afunction of time (t).

MATLAB Simulation:

Write MATLAB code to:

I. Define the circuit parameters (R, L, and Vin).

II. Simulate the transient response of the RL circuit using the 'lsim' function. Use a timevector from 0 to 5τ with a suitable time step.

III. Plot the transient response of the current ('lL) as a function of time. Label the axesappropriately.

Time Constant Verification:

I. Calculate the time constant τ numerically from the transient response obtained in question 2.1(c).

II. Compare the numerically calculated time constant with the value obtained in question.

III. Explain any differences or similarities.

Analysis:

Based on the transient response plot obtained in question iii, discuss the behavior of the currentin the RL circuit as it responds to the voltage step input. Specifically, explain the time constant, the initial current, and the behavior as time progresses.

2.2 Time Response of Series RC

In this MATLAB code:

1. Circuit parameters such as resistance (R), capacitance (C), and applied voltage (Vin) are defined.
2. The time constant (τ) is calculated as R * C.
3. A symbolic variable t is defined for time.
4. The differential equation representing the transient response of the RC circuit is defined usingdiff and ==.
5. The dsolve function is used to solve the differential equation symbolically.
6. A time vector (t_span) is created for the simulation.
7. Vc(t) is numerically evaluated for the given time vector.
8. The transient response is plotted, showing how the voltage across the capacitor changes overtime in response to a step input voltage.

You can adjust the time span in time span to observe the transient response for the desired duration.

Hints

```
% Define circuit parametersR =
100; % Resistance in ohms
C = 0.1; % Capacitance in farads Vin = 5; %
Applied voltage in volts

% Calculate the time constant τtau = R
* C;

% Define the symbolic variable for timesyms t

% Define the differential equation for the voltage VC(t) across the capacitorVC = sym('VC(t)');
eqn = R * diff (VC, t) + VC == Vin;

% Solve the differential equation symbolicallyVC_solution
= dsolve (eqn, VC);
VC_solution=simplify(VC_solution);

% Create a time vector
t_span = 0:0.01:5*tau; % Adjust the time span as needed

% Evaluate VC(t) for the given time vector VC_numeric = double
(subs (VC_solution, t, t_span));

% Plot the transient responseplot
(t_span, VC_numeric); xlabel
('Time (s)');
ylabel ('Voltage across Capacitor (V)');
title ('Transient Response of RC
Circuit');grid on;
```

Try

Circuit Parameters:

Given the following values:

I. Resistor (R): 220 ohms

II. Inductor (L): 0.01 F

III. Applied Voltage (Vin): 10 V

Calculate the time constant (τ) of the RC circuit.

b. Transient Analysis:

I. Write down the first-order differential equation governing the transient response of theRC circuit in terms of the voltage (Vc) across the capacitor.

II. Solve the differential equation symbolically to obtain the expression for Vc(t) as afunction of time (t).

c. MATLAB Simulation:

Write MATLAB code to:

I. Define the circuit parameters (R, C, and Vin).

II. Simulate the transient response of the RC circuit using the ‘lL’ function. Use a timevector from 0 to 5τ with a suitable time step.

III. Plot the transient response of the voltage across the capacitor (Vc) as a function oftime. Label the axes appropriately.

d. Time Constant Verification:

I. Calculate the time constant τ numerically from the transient response obtainedin question 2.2c.

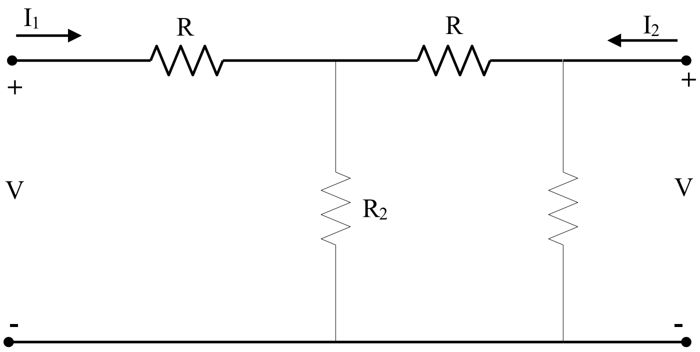
II. Compare the numerically calculated time constant with the value obtained inquestion 2.2(a). Explain any differences or similarities.

e. Analysis:

Based on the transient response plot obtained in question c(iii), discuss the behavior of the current in the RL circuit as it responds to the voltage step input. Specifically, explain the time constant, the initial current, and the behavior as time progresses.

3.1 Determination of Two port network parameters – Z, Y, Transmission and Hybrid parameters

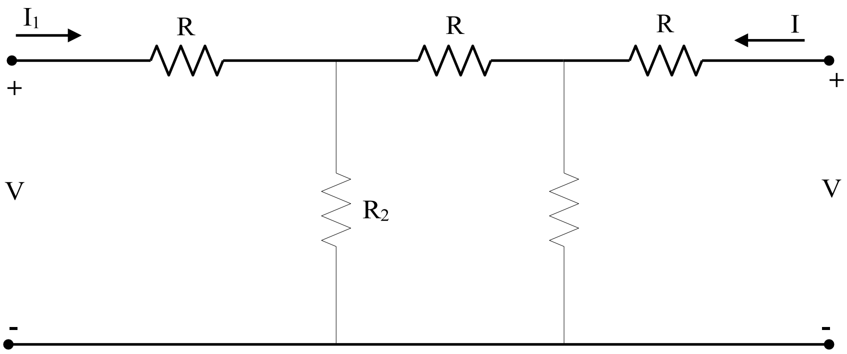
Calculate Z-parameters and Y-parameters for Circuit – 22, and analyze the network's response to different conditions (open and short circuits). Where R_1 = 100 Ω, R_2 = 220 Ω, R_3 = 1k Ω and R_4 = 150 Ω.



Circuit-3

Try

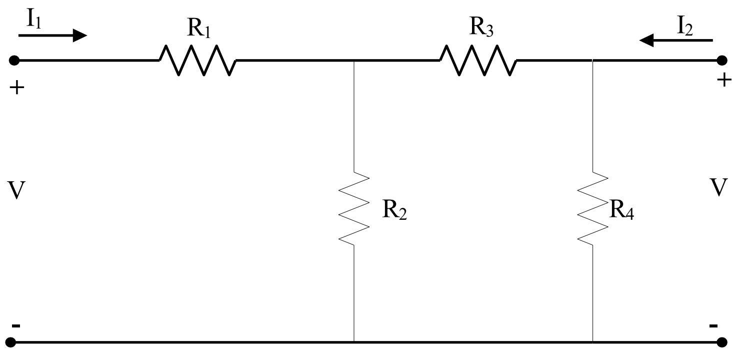
1. Use Simulink to simulate Circuit–22 for determining the Z and Y parameters and compare the values to those you obtained from your experiment.
2. Calculate Z parameters and convert in terms of Y parameters for Circuit – 23. Where R_1 = 100 Ω, R_2 = 220 Ω, R_3 = 1k Ω, R_4 = 150 Ω and R_5 = 82 Ω.



Circuit – 4

3.2 Exercise on H and ABCD Parameters

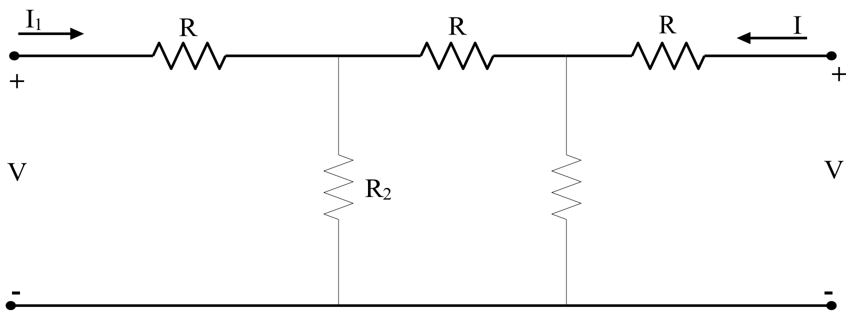
Calculate H and ABCD parameters for Circuit – 24, and analyze voltage gain and impedance transformation. Where R_1 = 100 Ω, R_2 = 220 Ω, R_3 = 1k Ω and R_4 = 150 Ω.



Circuit-5

Try

1. Use Simulink to simulate Circuit – 24 for determining the H and ABCD parameters and compare the values to those you obtained from your experiment.
2. Calculate H parameters and convert in terms of Y parameters for Circuit – 25. Where R_1 = 100 Ω, R_2 = 220 Ω, R_3 = 1k Ω, R_4 = 150 Ω and R_5 = 82 Ω.

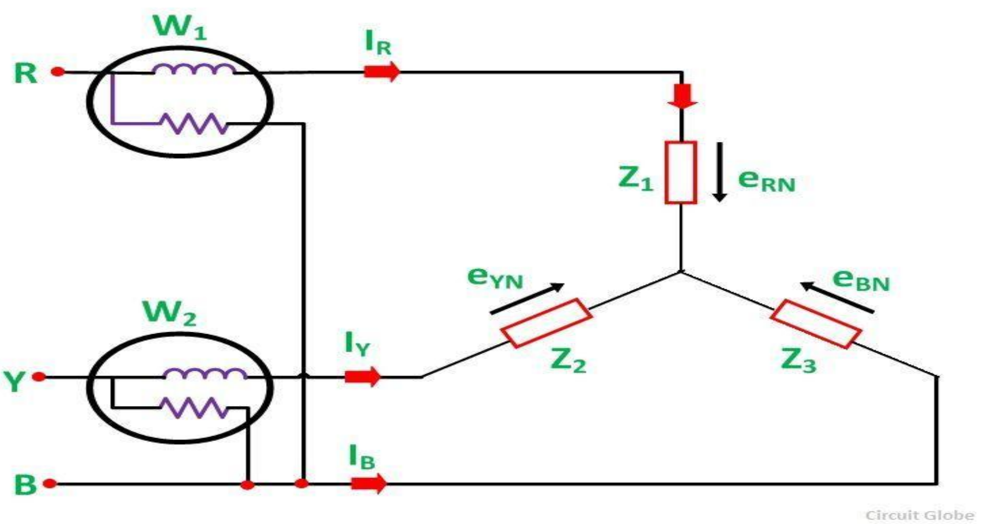


Circuit – 6

4. Measurement of 3-phase power in Balanced Star connected load using Two-Wattmeter method.

To simulate and measure the total power of a balanced 3-phase star-connected load using the two-wattmeter method.

a. balanced 3-phase star-connected load



Circuit – 7

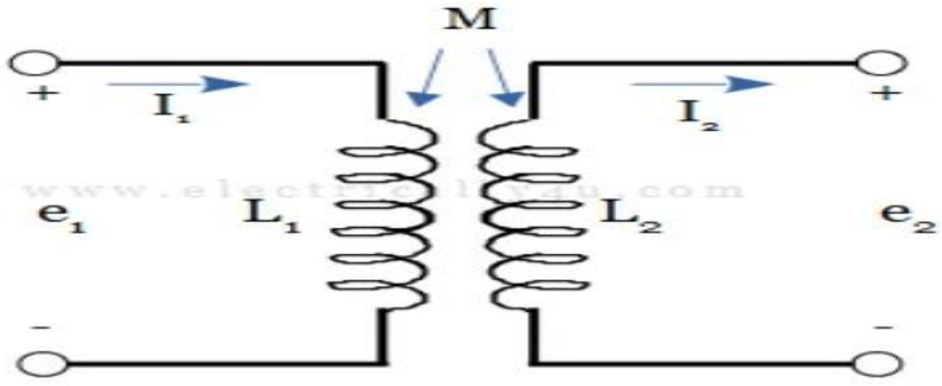
```
% Theoretical two-wattmeter values for balanced Y load
Vll = 400;          % V (line-to-line)
Zmag = 10;          % magnitude of per-phase impedance (Ohm)
phi_deg = 30;       % load power factor angle in degrees (positive = lagging)

Vph = Vll/sqrt(3);
I = Vph / Zmag;     % line current magnitude

% Wattmeter formulas
W1 = Vll * I * cosd(phi_deg - 30);
W2 = Vll * I * cosd(phi_deg + 30);
P_total = W1 + W2;  % should equal sqrt(3)*Vll*I*cos(phi_deg)

fprintf('I = %.4f A\n', I);
fprintf('W1 = %.3f W\nW2 = %.3f W\nP_total = %.3f W\n', W1, W2, P_total);
```

Expected printed result (approx):

```
I = 23.0940 A
W1 = 9237.604 W
W2 = 4618.802 W
P_total = 13856.406 W
```

5. Determination of Co-efficient of coupling, self and mutual inductance in magnetic Coupled Circuits

To determine the self-inductance (L1, L2) of two coils, the mutual inductance (M) between them, and the Coefficient of coupling (k) in a magnetically coupled circuit.



Circuit –8

MATLAB code

```
% coupled_inductor_test.m
% Simulate two coupled inductors, series aiding and series opposing,
% compute La, Lo, M, k from simulation.

clear; close all; clc;

% PARAMETERS (example)
L1 = 10e-3;    % H
L2 = 20e-3;    % H
k  = 0.8;      % coupling coefficient (true, for simulation)
M_true = k*sqrt(L1*L2);

R_series = 0.01;    % small series resistance to aid numeric stability (ohm)
Vamp = 5;           % source amplitude (V)
f = 50;             % frequency (Hz)
w = 2*pi*f;

% time vector
T = 1/f;
t = linspace(0,5*T,20000);   % simulate 5 cycles, high resolution

% Helper: simulate series connection with given sign (+1 aiding, -1 opposing)
simulate_series = @(sign) simulate_two_coils(L1,L2,M_true,sign,R_series,Vamp,w,t);

% Simulate aiding (+1) and opposing (-1)
[data_a] = simulate_series(+1);
[data_o] = simulate_series(-1);

% compute RMS voltages and currents after transients (last cycle)
last_cycle_idx = find(t >= 4*T);
```

```
Vrms = rms(Vamp*sin(w*t(last_cycle_idx)));
```

```
Irms_a = rms(data_a.i_series(last_cycle_idx));
Irms_o = rms(data_o.i_series(last_cycle_idx));

% Equivalent inductances:
La = Vrms/(w*Irms_a);
Lo = Vrms/(w*Irms_o);

% Compute M and k from La and Lo (and known L1,L2)
M_calc = (La - Lo)/4;
Lsum = (La + Lo)/2;
k_calc = M_calc / sqrt(L1*L2);

% Display results
fprintf('Given L1 = %.6f H, L2 = %.6f H, k(true) = %.3f\n', L1, L2, k);
fprintf('True M = %.6e H\n', M_true);
fprintf('Measured La = %.6e H\n', La);
fprintf('Measured Lo = %.6e H\n', Lo);
fprintf('Calculated M = %.6e H\n', M_calc);
fprintf('Calculated k = %.6f\n', k_calc);

% Plot steady waveforms (last cycle)
figure;
subplot(3,1,1);
plot(t(last_cycle_idx), data_a.i_series(last_cycle_idx));
title('Series Current (aiding) - last cycle'); xlabel('t (s)'); ylabel('I_A (A)');

subplot(3,1,2);
plot(t(last_cycle_idx), data_o.i_series(last_cycle_idx));
title('Series Current (opposing) - last cycle'); xlabel('t (s)'); ylabel('I_O (A)');

subplot(3,1,3);
plot(t(last_cycle_idx), Vamp*sin(w*t(last_cycle_idx)));
title('Source Voltage - last cycle'); xlabel('t (s)'); ylabel('V (V)');

% -------------------------
% Local function: simulate two coupled coils connected in series
function out = simulate_two_coils(L1,L2,M,sign,Rs,Vamp,w,t)
   % state variables: i1, i2  (currents through coil1, coil2)
   % Equations (flux linkages):
   % v1 = L1 di1/dt + M d i2/dt + R1 i1 (R1 assumed small)
   % v2 = M d i1/dt + L2 d i2/dt + R2 i2
   %
   % But for series connection we combine so that the single series
   % current i_series flows through coil1 and coil2 in a specific polarity.
   % We'll simulate loop with i_series and differential equations for flux.
   %
   % Simpler approach: Build coupled inductance matrix and compute d/dt currents
   %
   % Inductance matrix:
   Lmat = [L1, sign*M; sign*M, L2];
```

```
    Rmat = [Rs, 0; 0, Rs];
    % For series connection with single source across series combination:
    % The source drives series loop: v = [1 1] * (Lmat * di/dt + Rmat*i)
    % We'll simulate two currents satisfying same series current i1 = i2 = i_series
    % For small Rs we approximate i1 = i2 = i_series * [1;1]*0.5?
    % To properly capture mutual effect, simulate two-loop equations with a single
    % series source by enforcing i1 = i_series and i2 = i_series with sign on dot.
    % To avoid complexity, simulate two-winding circuit as single series branch with
    % equivalent L_eq = L1 + L2 + 2*sign*M and R = 2*Rs.
    Leq = L1 + L2 + 2*sign*M;
    Req = 2*Rs;
    % Solve di/dt = (1/Leq)*(v(t) - Req*i)
    dt = t(2)-t(1);
    i = zeros(size(t));
    for idx = 2:length(t)
        v = Vamp*sin(w*t(idx));
        di_dt = (v - Req*i(idx-1))/Leq;
        i(idx) = i(idx-1) + di_dt*dt;
    end
    out.i_series = i;
end

% RMS helper
function r = rms(x)
    r = sqrt(mean(x.^2));
end
```

Try:

- Build model per section 1, set L1/L2/k as above.
- Run for a few cycles with a 50 Hz sine, measure series current for aiding & opposing connections.
- Compute the equivalent inductance using $L_{\text{eq}} = V_{\text{rms}}/(\omega I_{\text{rms}})$ from your logged signals.
- Then compute M and k as shown.

6. Frequency domain analysis of Low-pass filter and High-pass filters using circuit simulation software

To plot the characteristics of low-pass and high-pass filters using MATLAB, you can follow these stepsand use the 'freqz' function for plotting the frequency response:

Hints

/** Design the Filters */

Design low-pass and high-pass filters using the **'fir1'** or **'fdesign'** functions.Here's an example using **'fir1'**:

```
% Design a low-pass filter
lowpass_order = 50; % Filter order
lowpass_cutoff_frequency = 0.2; % Cutoff frequency (normalized)lowpass_filter =
fir1(lowpass_order, lowpass_cutoff_frequency);

% Design a high-pass filter highpass_order
= 50; % Filter order
highpass_cutoff_frequency = 0.2; % Cutoff frequency (normalized)
highpass_filter = fir1(highpass_order, highpass_cutoff_frequency, 'high');
```

/** Plot the Frequency Response */

Plot the frequency response of the filters using the **'freqz'** function:

```
% Frequency response of the low-pass filterfigure;
freqz (lowpass_filter, 1, 1024);
title ('Low-Pass Filter Frequency Response');
xlabel ('Normalized Frequency (\pi radians/sample)');ylabel
('Magnitude (dB)');

% Frequency response of the high-pass filterfigure;
freqz (highpass_filter, 1, 1024);
title ('High-Pass Filter Frequency Response');
xlabel ('Normalized Frequency (\pi radians/sample)');ylabel
('Magnitude (dB)');
```

In this code:

‘freqz (lowpass_filter, 1, 1024)’ calculates and plots the frequency response ofthe low-pass filter.
‘freqz (highpass_filter, 1, 1024)’ calculates and plots the frequency response of the high-pass filter.

/** **Display the Phase Response (Optional)** */

If you also want to display the phase response, you can modify the **‘freqz’** functionas follows:

```
% Phase response of the low-pass filterfigure;
[h, w] = freqz (lowpass_filter, 1, 1024);plot (w,
unwrap(angle(h)));
title ('Low-Pass Filter Phase Response');
xlabel ('Normalized Frequency (\pi radians/sample)');ylabel ('Phase
(radians)');

% Phase response of the high-pass filter
figure;
[h, w] = freqz (highpass_filter, 1, 1024);plot (w,
unwrap(angle(h)));
title ('High-Pass Filter Phase Response');
xlabel ('Normalized Frequency (\pi radians/sample)');ylabel ('Phase
(radians)');
```

Try:

Problem 1: Filter Design

1. Design a low-pass filter with the following specifications:

- Filter Order: 40
- Cutoff Frequency: 0.2 (normalized frequency)

2. Design a high-pass filter with the following specifications:

- Filter Order: 30
- Cutoff Frequency: 0.3 (normalized frequency)

Problem 2: Frequency Response Plot

1. Plot the magnitude frequency response (in dB) of both the low-pass and high-pass filters.

- Use the **‘freqz’** function to calculate and plot the frequency response.
- Label the x-axis as "Normalized Frequency (\pi radians/sample)" and the y-axis as"Magnitude (dB)."

2. Plot the phase frequency response (in radians) of both filters.

- Use the freqz function to calculate and plot the phase response.
- Label the x-axis as "Normalized Frequency (\pi radians/sample)" and the y-axis as"Phase (radians)."

Problem 3: Filtered Signal

1. Generate a noisy input signal using MATLAB. You can use the **‘randn’** function to generaterandom noise and the **‘sin’** function to generate a sinusoidal signal.
2. Apply both the low-pass and high-pass filters to the noisy signal to obtain filtered signals.

3. Plot the noisy input signal, the low-pass filtered signal, and the high-pass filtered signal on thesame graph.

Problem 4: Frequency Response Analysis

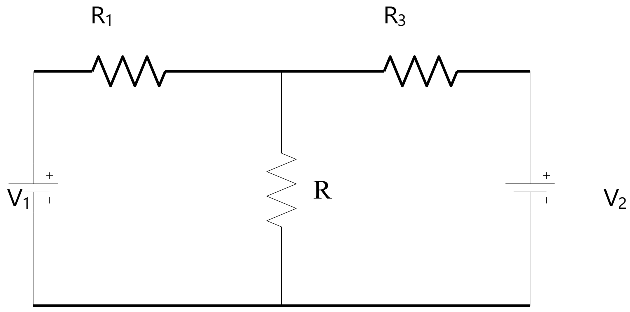
1. Analyze the frequency response plots and explain how the magnitude and phase characteristicsof the filters align with their designed specifications.
2. Describe the differences in filtering results between the low-pass and high-pass filters. Discusswhich frequencies are attenuated and which are passed through.

Problem 5: Submission

1. Prepare a report summarizing your filter design, frequency response analysis, and filtered signalresults.
2. Include the MATLAB code used for filter design, frequency response plotting, and signalprocessing in your report.
3. Provide explanations and interpretations of the results in the report.

7.1 Verification of Superposition and Maximum Power Transfer theorems using any circuit simulation software.

Investigate the current thought R_2 resistor using superposition theorem to multiple DC source



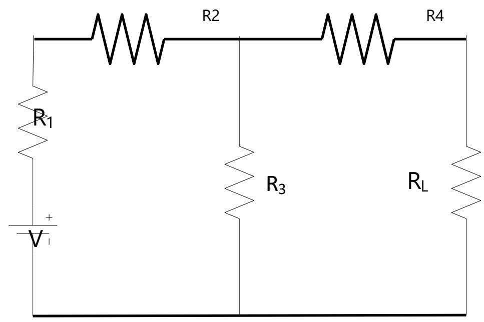
Circuit – 9

Try

1. Use Simulink to simulate the Circuit – 16 for determining the current through a resistor R_2 and compare this value to those you obtained from your experiment.
2. Find the current thought R_2 in Circuit – 16 using any circuit reduction technique and verify this value to those you obtained from the superposition theorem.
3. Find the current through the R_4 resistor using the superposition theorem for the Circuit – 17.

7.2 Exercise on Maximum Power Transfer Theorem

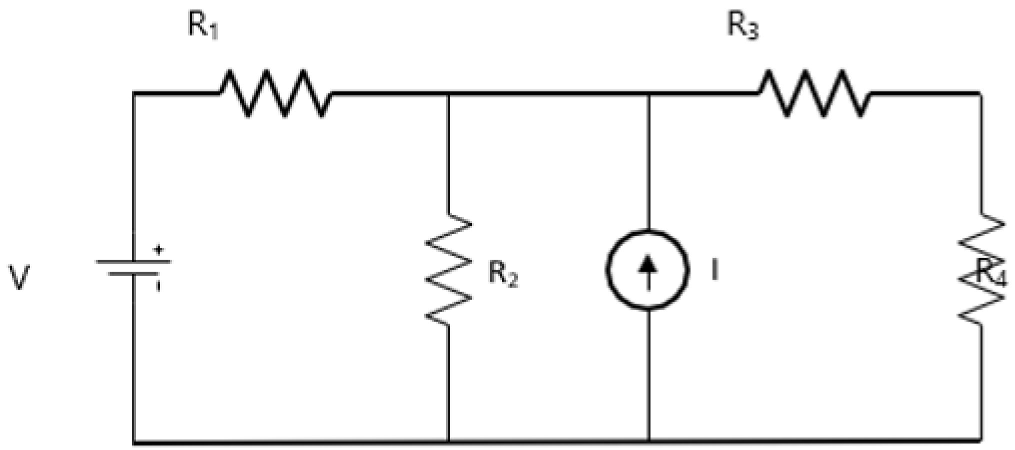
1. Find the value of R_L at which maximum power is transferred to the load in Circuit – 21.
2. Determine the maximum power transferred to the load. Where V = 10 V, R_1 = 100 Ω, R_2 = 220 Ω, R_3 = 1k Ω, R_4 = 150 Ω.



Circuit – 10

Try

1. If the internal resistance of a voltage source is adjustable, and the load resistor is fixed. Determine the range of source resistance values that will result in effective power transfer. Calculate the corresponding load power for each extreme of the source resistance range.
2. Use Simulink to simulate Circuit – 21 for determining the current thought R_L and compare this value to those you obtained from your experiment.



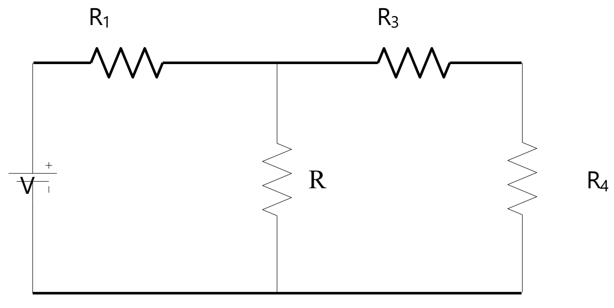
Circuit – 11

8. Verification of Thevenin's and Norton's theorems using any circuit simulation software.

8.1 Exercises on Thevenin's Theorem

1. Determine Thevenin's equivalent voltage (V_{th}) and resistance (R_{th}) at R_4 by applying Thevenin's theorem for Circuit – 13.
2. Determine load or unknown current through a R_4 resistor using Thevenin's equivalent circuit. Were

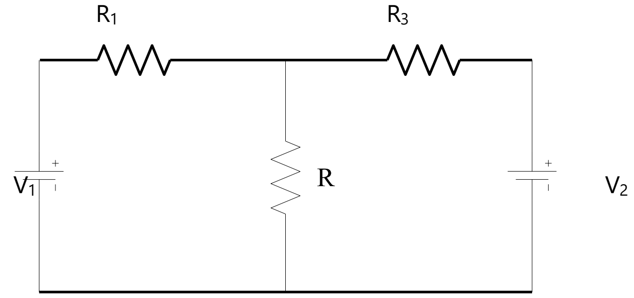
V = 10V, R_1 = 100 Ω, R_2 = 220 Ω, R_3 = 1k Ω and R_4 = 150 Ω.



Circuit – 12

Try

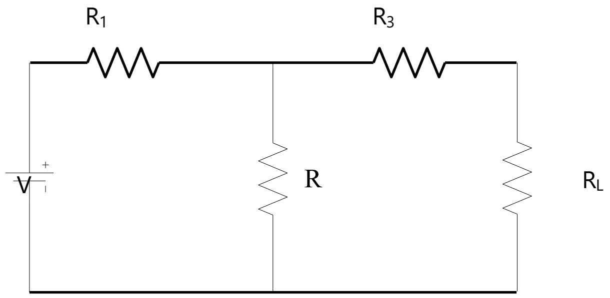
1. Use Simulink to simulate the Circuit – 13 for determining the current thought R_4 resistor using Thevenin's theorem and compare this value to those you obtained from your experiment.
2. Find the current thought R_4 resistor using any circuit reduction technique and verify this value to those you obtained from Thevenin's theorem.
3. Find the current flowing through R_2 resistor in Circuit – 14 using Thevenin's theorem for the below circuit. Where V_1 = 10 V, V_2 = 5 V, R_1 = 100 Ω, R_2 = 220 Ω, R_3 = 1k Ω.



Circuit – 13

8.2 Exercises on Norton's Theorem

1. Find Norton equivalent current (I_N) and resistance (R_N) by considering R_L = 150 Ω resistor for the Circuit – 15 by applying Norton's theorem.
2. Find load or unknown current through R_L resistor using Norton's equivalent circuit. Where V = 10 V, R_1 = 100 Ω, R_2 = 220 Ω and R_3 = 1k Ω.



Circuit – 14

Try

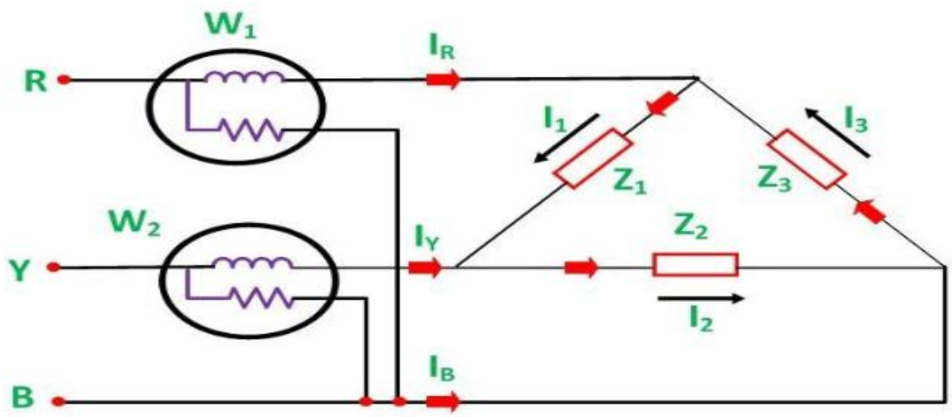
1. Use Simulink to simulate the Circuit – 15 for determining the current through R_L resistor and compare this value to those you obtained from your experiment.
2. Find the current thought R_L using any circuit reduction technique and verify this value to those you obtained from Norton's theorem.

9. Measurement of Active Power for Delta-connected balanced loads.

To measure the active power consumed by a delta-connected balanced load in a three-phase system using the Two-Wattmeter Method.

The two-wattmeter method is widely used for measuring the active power in a three-phase system (balanced). For a delta-connected balanced load, the total active power is the algebraic sum of the readings of the two wattmeters**:**

P=W1+W2



Circuit – 15

```
# Title: Measurement of Active Power in Delta Connected Balanced Load
# Method: Two Wattmeter Method
# Software: GLIMSim

BEGIN
 COMPONENTS:
  # Source
  Vsource3P  V1   R Y B  N  (3-phase balanced source, 400V, 50Hz)

  # Wattmeters
  Wattmeter   W1  R N LOAD1
  Wattmeter   W2  Y N LOAD2

  # Load - Delta Connected
  Impedance   Z1 R Y  (10+j5)   # Example load impedance per phase
  Impedance   Z2  Y B  (10+j5)
  Impedance   Z3  B R  (10+j5)

 CONNECTIONS:
  # Wattmeter W1: Current coil in R phase, potential coil across R-Y
```

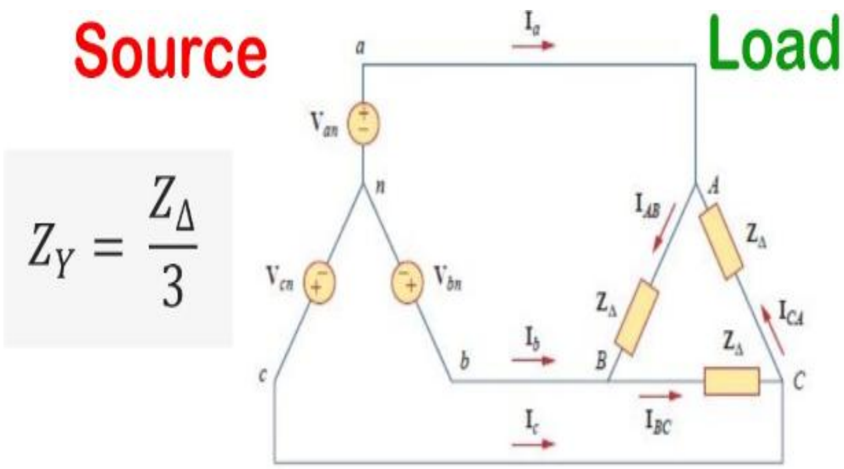
```
            W1.I coil  R N
            W1.P coil  R Y
# Wattmeter W2: Current coil in Y phase, potential coil across Y-B
    W2.Icoil  Y N
    W2.Pcoil  Y B

    # Source connections
    V1.R  -> R
    V1.Y  -> Y
    V1.B  -> B
    V1.N  -> N

    # Load connections (Delta)
    Z1.p -> R
    Z1.n -> Y

    Z2.p -> Y
    Z2.n -> B

    Z3.p -> B
    Z3.n -> R

END

SIMULATION:
  RUN  TIME = 0.1s
  PLOT  W1.Power  W2.Power  (Wattmeter readings)
  PLOT  V1.VR, V1.VY, V1.VB (Source voltages)
  PLOT  I(R), I(Y), I(B)    (Line currents)

RESULT:
  Total Power = W1 + W2
```

Try

1. Two wattmeters (W1, W2) measure power.
2. The load is balanced Delta connected with impedance Z=(10+j5) Ω (example).
3. You can change the impedance values or supply voltage as per requirement.
4. Total power = W1 + W2.

10. Measurement of Reactive Power for Star and Delta connected balanced load

To measure the reactive power of star and Delta connected balanced load



Circuit – 16

- In a star connection:
  $V_{ph} = \frac{V_{line}}{\sqrt{3}}$
- In a delta connection:
  $V_{ph} = V_{line}$
- Line current and phase current relationship in Delta load:
  $I_{line} = \sqrt{3} \cdot I_{ph}$

Also, for equivalent conversion:

$$Z_Y = \frac{Z_\Delta}{3}$$

```
% Simulation of Star Connected Source supplying Delta Connected Load

clc;
clear all;

% Source line voltage (rms)
V_ line = 400;   % Volts (3-phase balanced source)

% Phase voltage of star source
V_ phase_ Y = V_ line / sqrt(3);

% Load Impedance (Delta connected)
Z_ delta = 20 + 10j;  % example: 20 + j10 ohms

% Phase voltage across each delta branch
V_ phase_ delta = V_ line;  % In delta, Vph = Vline

% Phase current of delta load
I_ ph = V_ phase_ delta / Z_ delta;

% Line current of delta load
I_ line = sqrt(3) * I_ ph;

% Power calculations
P_ total = 3 * (V_ phase_ delta * conj(I_ ph)); % total 3-phase power
Q_ total = 3 * imag(V_ phase_ delta * conj(I_ ph));
S_ total = 3 * (V_ phase_ delta * conj(I_ ph));

% Display results
disp('--- Star Connected Source feeding Delta Load ---');
disp(['Source Line Voltage = ', num2str(V_ line), ' V']);
disp(['Load Impedance (Delta) = ', num2str(Z_ delta), ' ohm']);
disp(['Delta Phase Current = ', num2str(abs(I_ ph)), ' A']);
disp(['Line Current = ', num2str(abs(I_ line)), ' A']);
disp(['Total Active Power = ', num2str(real(P_ total)), ' W']);
disp(['Total Reactive Power = ', num2str(Q_ total), ' VAR']);
disp(['Total Apparent Power = ', num2str(abs(S_ total)), ' VA']);
```

11. Frequency domain analysis of Band-pass filters

Analyze the virtual instrumentation (VI) using control loops, arrays, charts and graphs in LabVIEW software

Hints

/** Open LabVIEW and create a new VI */

/** Place the following controls on the front panel */

- Numeric Control for Frequency (Hz)
- Numeric Control for Amplitude (V)
- Numeric Control for Duration (seconds)
- Waveform Chart (for displaying the original signal)
- Waveform Graph (for displaying the FFT result)

/ Create a FOR loop on the block diagram with the following elements inside it */**

- Numeric Constant (0) wired to the FOR loop iteration terminal
- Numeric Control for Duration wired to the count terminal of the FOR loop

/ Inside the FOR loop, generate a sinusoidal signal using the Sine Waveformfunction */**

- Place a Sine Waveform function inside the FOR loop
- Wire the Frequency control to the "Frequency (Hz)" input of the Sine Waveformfunction
- Wire the Amplitude control to the "Amplitude (V)" input of the Sine Waveformfunction
- Use the FOR loop iteration terminal to generate a time signal (0, 1 sampleperiod, 2 sample periods, etc.)
- Wire the time signal to the "Time (s)" input of the Sine Waveform function

/ Wire the output of the Sine Waveform function to a Build Array function tocreate an array of the generated signal */**

/ After the FOR loop, add a Peak Detector VI to detect peaks in the signal */**

- Place a Peak Detector VI on the block diagram
- Wire the output array to the "Input Array" input of the Peak Detector
- Set the "Threshold" parameter of the Peak Detector VI to a suitable value

/ Connect the output of the Peak Detector VI to the "Data" input of the WaveformChart to display the original signal with peaks highlighted */**

/ Place a Waveform Chart Clear function outside the FOR loop and wire it to thechart to clear the chart before displaying the new data */**

/ Create a WHILE loop with the following elements inside it */**

- A Waveform Graph Clear function to clear the graph for displaying FFT results
- An FFT function to compute the Fast Fourier Transform of the signal:
  - Wire the output array (from the Peak Detector) to the "Time Domain Signal"input of the FFT function
- A waveform chart to display the FFT result:
  - Wire the output of the FFT function to the "Amplitude Spectrum" input of thewaveform chart

/ Wire the output of the Waveform Chart Clear function inside the WHILE loop toclear the chart before displaying the new data */**

/ Run the VI to generate the sinusoidal signal, detect peaks, compute the FFT,and display the results on the charts */**

/** Analyze the displayed results, including peak positions and frequency componentsin the FFT result */

This LabVIEW example demonstrates the use of FOR and WHILE loops, charts (waveform chart and waveform graph), arrays, and analysis VIs (Peak Detector and FFT) in a signal processing application. Students can modify and extend this example to explore more complex signal processing and analysis tasks.

Try:

1. Obtain VI using for loop and While loop.
2. Obtain VI using charts and arrays.
3. Obtain VI using graphs.

12. Frequency domain analysis of Band-stop filters

Analyze the virtual instrumentation (VI) using control loops, arrays, charts and graphs in LabVIEW software

Hints

/** Design the Filters */

Design low-pass and high-pass filters using the **'fir1'** or **'fdesign'** functions.Here's an example using **'fir1'**:

```
% Design a low-pass filter
lowpass_order = 50; % Filter order
lowpass_cutoff_frequency = 0.2; % Cutoff frequency (normalized)lowpass_filter =
fir1(lowpass_order, lowpass_cutoff_frequency);

% Design a high-pass filter highpass_order =
50; % Filter order
highpass_cutoff_frequency = 0.2; % Cutoff frequency (normalized)
highpass_filter = fir1(highpass_order, highpass_cutoff_frequency, 'high');
```

/** Plot the Frequency Response */

Plot the frequency response of the filters using the **'freqz'** function:

```
% Frequency response of the low-pass filterfigure;
freqz (lowpass_filter, 1, 1024);
title ('Low-Pass Filter Frequency Response');
xlabel ('Normalized Frequency (\pi radians/sample)');ylabel
('Magnitude (dB)');

% Frequency response of the high-pass filterfigure;
freqz (highpass_filter, 1, 1024);
title ('High-Pass Filter Frequency Response');
xlabel ('Normalized Frequency (\pi radians/sample)');ylabel
('Magnitude (dB)');
```

In this code:
‘freqz (lowpass_filter, 1, 1024)’ calculates and plots the frequency response ofthe low-pass filter.
‘freqz (highpass_filter, 1, 1024)’ calculates and plots the frequency response of the high-pass filter.

/ Display the Phase Response (Optional) */**

If you also want to display the phase response, you can modify the **‘freqz’** functionas follows:

```
% Phase response of the low-pass filterfigure;
[h, w] = freqz (lowpass_filter, 1, 1024);plot (w,
unwrap(angle(h)));
title ('Low-Pass Filter Phase Response');
xlabel ('Normalized Frequency (\pi radians/sample)');ylabel ('Phase
(radians)');

% Phase response of the high-pass filter
figure;
[h, w] = freqz (highpass_filter, 1, 1024);plot (w,
unwrap(angle(h)));
title ('High-Pass Filter Phase Response');
xlabel ('Normalized Frequency (\pi radians/sample)');ylabel ('Phase
(radians)');
```

Try:

Problem 1: Filter Design

1. Design a low-pass filter with the following specifications:
   - Filter Order: 40
   - Cutoff Frequency: 0.2 (normalized frequency)
2. Design a high-pass filter with the following specifications:
   - Filter Order: 30
   - Cutoff Frequency: 0.3 (normalized frequency)

Problem 2: Frequency Response Plot

1. Plot the magnitude frequency response (in dB) of both the low-pass and high-pass filters.
   - Use the **‘freqz’** function to calculate and plot the frequency response.
   - Label the x-axis as "Normalized Frequency (\pi radians/sample)" and the y-axis as"Magnitude (dB)."
   - Plot the phase frequency response (in radians) of both filters.
   - Use the freqz function to calculate and plot the phase response.
   - Label the x-axis as "Normalized Frequency (\pi radians/sample)" and the y-axis as"Phase (radians)."

Problem 3: Filtered Signal

1. Generate a noisy input signal using MATLAB. You can use the **‘randn’** function to generaterandom noise and the **‘sin’** function to generate a sinusoidal signal.
2. Apply both the low-pass and high-pass filters to the noisy signal to obtain filtered signals.

3. Plot the noisy input signal, the low-pass filtered signal, and the high-pass filtered signal on thesame graph.

Problem 4: Frequency Response Analysis

1. Analyze the frequency response plots and explain how the magnitude and phase characteristicsof the filters align with their designed specifications
2. Describe the differences in filtering results between the low-pass and high-pass filters. Discusswhich frequencies are attenuated and which are passed through.

Problem 5: Submission

1. Prepare a report summarizing your filter design, frequency response analysis, and filtered signalresults.
2. Include the MATLAB code used for filter design, frequency response plotting, and signalprocessing in your report.
3. Provide explanations and interpretations of the results in the report.

13. Determination of Time response of first order RL, RC circuit for periodic non – sinusoidal inputs - Time Constant and Steady state error.

Examine the time varying characteristics of series RL, RC and RLC circuits for given values of R, L and Cusing MATLAB software.

13.1 Time Response of Series RL

In this MATLAB program:

1. Circuit parameters (resistance, inductance, and applied voltage) are defined.
2. The time constant (τ) is calculated as L / R.
3. A symbolic variable **t** is defined for time.
4. The differential equation representing the transient response of the RL circuit is defined using **diff** and ==.
5. The **dsolve** function is used to solve the differential equation symbolically.
6. A time vector (**‘t_span’**) is created for the simulation.
7. IL(t) is numerically evaluated for the given time vector.
8. The transient response is plotted.

Hints

```
% Define circuit parameters
R = 100; % Resistance in ohmsL = 0.5;
% Inductance in henrys
Vin = 10; % Applied voltage in volts

% Calculate the time constant τtau = L
/ R;

% Define the symbolic variable for timesyms t

% Define the differential equation for the current IL(t)IL = sym('IL(t)');
eqn = L * diff(IL, t) + R * IL == Vin;

% Solve the differential equation IL_solution =
dsolve (eqn, IL); IL_solution=
simplify(IL_solution);

% Create a time vector and evaluate IL(t
time = 0:0.01:2*tau; % Adjust the time span as needed
IL_numeric = double (subs (IL_solution, t, time));

% Plot the transient responseplot
(time, IL_numeric); xlabel ('Time
(s)');
ylabel ('Current (A)');
title ('Transient Response of RL Circuit');grid on;
```

Try:

Circuit Parameters:

Given the following values:

I. Resistor (R): 150 ohms

II. Inductor (L): 0.3 H

III. Applied Voltage (V_in): 12 V

Calculate the time constant (τ) of the RL circuit.

Transient Analysis:

a. Write down the first-order differential equation governing the transient response of theRL circuit in terms of the current (I_L) through the inductor.
b. Solve the differential equation symbolically to obtain the expression for I_L(t) as afunction of time (t).

MATLAB Simulation:

Write MATLAB code to:

a. Define the circuit parameters (R, L, and Vin).
b. Simulate the transient response of the RL circuit using the 'l_{sim}' function. Use a timevector from 0 to 5τ with a suitable time step.
c. Plot the transient response of the current ('l_L) as a function of time. Label the axesappropriately.

Time Constant Verification:

a. Calculate the time constant τ numerically from the transient response obtained inquestion 2.1(c).

b. Compare the numerically calculated time constant with the value obtained in question.

c. Explain any differences or similarities.

Analysis:

Based on the transient response plot obtained in question iii, discuss the behavior of the currentin the RL circuit as it responds to the voltage step input. Specifically, explain the time constant, the initial current, and the behavior as time progresses.

13.2 Time Response of Series RC

In this MATLAB code:

1. Circuit parameters such as resistance (R), capacitance (C), and applied voltage (Vin) are defined.
2. The time constant (τ) is calculated as R * C.
3. A symbolic variable t is defined for time.
4. The differential equation representing the transient response of the RC circuit is defined usingdiff and ==.
5. The dsolve function is used to solve the differential equation symbolically.
6. A time vector (t_span) is created for the simulation.
7. Vc(t) is numerically evaluated for the given time vector.
8. The transient response is plotted, showing how the voltage across the capacitor changes overtime in response to a step input voltage.

You can adjust the time span in t_span to observe the transient response for the desired duration.

Hints

```
% Define circuit parametersR =
100; % Resistance in ohms
C = 0.1; % Capacitance in farads Vin = 5; %
Applied voltage in volts

% Calculate the time constant τtau = R
* C;

% Define the symbolic variable for timesyms t

% Define the differential equation for the voltage VC(t) across the capacitorVC = sym('VC(t)');
eqn = R * diff (VC, t) + VC == Vin;

% Solve the differential equation symbolicallyVC_solution
= dsolve (eqn, VC);
VC_solution=simplify(VC_solution);

% Create a time vector
t_span = 0:0.01:5*tau; % Adjust the time span as needed

% Evaluate VC(t) for the given time vector VC_numeric = double
(subs (VC_solution, t, t_span));

% Plot the transient responseplot
(t_span, VC_numeric); xlabel ('Time
(s)');
ylabel ('Voltage across Capacitor (V)'); title ('Transient
Response of RC Circuit');grid on;
```

Try:

Circuit Parameters:

Given the following values:

I. Resistor (R): 220 ohms

II. Inductor (L): 0.01 F

III. Applied Voltage (Vin): 10 V

Calculate the time constant (τ) of the RC circuit.

f. Transient Analysis:

I. Write down the first-order differential equation governing the transient response of theRC circuit in terms of the voltage (Vc) across the capacitor.

II. Solve the differential equation symbolically to obtain the expression for Vc(t) as afunction of time (t).

g. MATLAB Simulation:

Write MATLAB code to:

I. Define the circuit parameters (R, C, and Vin).

II. Simulate the transient response of the RC circuit using the ‘lL’ function. Use a timevector from 0 to 5τ with a suitable time step.

III. Plot the transient response of the voltage across the capacitor (Vc) as a function oftime. Label the axes appropriately.

h. Time Constant Verification:

I. Calculate the time constant τ numerically from the transient response obtainedin question 2.2c.

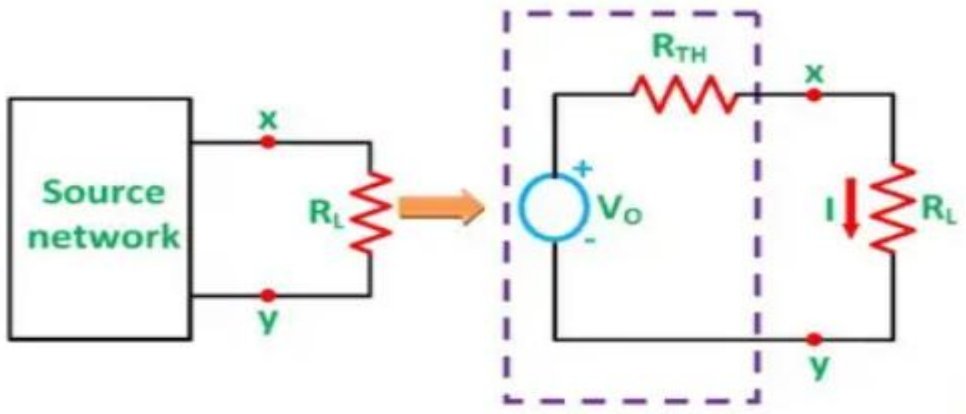
II. Compare the numerically calculated time constant with the value obtained inquestion 2.2(a). Explain any differences or similarities.

i. Analysis:

Based on the transient response plot obtained in question c(iii), discuss the behavior of the current in the RL circuit as it responds to the voltage step input. Specifically, explain the time constant, the initial current, and the behavior as time progresses

14. Verification of Compensation theorem.

To verify Compensation Theorem using MATLAB/Simulink by measuring currents before and after compensation in a simple resistive network.



Circuit – 17

```
% Verification of Compensation Theorem
clc; clear;

% Given values
Vs = 10;        % Source voltage
R1 = 5;         % Series resistor
R2 = 10;        % Branch resistor
R3 = 10;        % Load resistor
dR = 5;         % Change in resistance

% Case 1: Original circuit
Req1 = R1 + (R2*R3)/(R2+R3);   % Equivalent resistance
I_total1 = Vs / Req1;          % Total current
V_R2 = I_total1 * (R2*R3)/(R2+R3); % Voltage across parallel
I_R2 = V_R2 / R2;              % Current through R2

fprintf('Original Current through R2 = %.3f A\n', I_R2);
% Case 2: Direct change in R2
R2_new = R2 + dR;
Req2 = R1 + (R2_new*R3)/(R2_new+R3);
I_total2 = Vs / Req2;
V_R2_new = I_total2 * (R2_new*R3)/(R2_new+R3);
I_R2_new = V_R2_new / R2_new;

fprintf('Current through R2 after direct change = %.3f A\n', I_R2_new);

% Case 3: Compensation Theorem
Vc = I_R2 * dR;     % Compensating voltage
Vs_comp = Vs + Vc;  % Modified source
Req3 = R1 + (R2_new*R3)/(R2_new+R3);
I_total3 = Vs_comp / Req3;
V_R2_comp = I_total3 * (R2_new*R3)/(R2_new+R3);
I_R2_comp = V_R2_comp / R2_new;

fprintf('Current through R2 using Compensation theorem = %.3f A\n', I_R2_comp);
```

Expected Output (approx.):

```
Original Current through R2 = 0.400 A
Current through R2 after direct change = 0.333 A
Current through R2 using Compensation theorem = 0.333 A
```

1. Open **MATLAB Simulink** → New Model.
2. Drag blocks:
   - Voltage Source (DC)
   - Resistors (R1, R2, R3)
   - Ground
   - Current Measurement (for branches)
   - Scope / Display
3. Connect them as per circuit.
4. Run first case with **original circuit** ($R_2 = 10\Omega$).
5. Then **modify R2 = 15 Ω** and add a compensating voltage source $V_c = I \cdot \Delta R$ in series with R2, where I is the original branch current.
6. Compare currents in both methods.

V. TEXT BOOKS:

1. A Chakrabarthy, "Circuit Theory", Dhanpat Rai Publications, 6th edition, 2006.
2. A Sudhakar, Shyammohan S Palli, "Circuits and Networks", Tata McGraw Hill, 4th edition,2010.

VI. REFERENCES

1. William Hayt, Jack E Kemmerly S.M. Durbin, "Engineering Circuit Analysis", Tata McGrawHill, 7th edition, 2010.
2. K S Suresh Kumar, "Electric Circuit Analysis", Pearson Education, 1st edition, 2013.
3. Rudrapratap, "Getting started with MATLAB: A Quick Introduction for Scientists andEngineers", Oxford University Press, 1st edition, 19994.