THEORY: MATLAB is a programming language developed by Math Works. It started out as amatrix programming language where linear algebra programming was simple. MATLAB (matrix laboratory) is a fourth-generation high-level programming language and interactive environment for numerical computation, visualization and programming.

MATLAB is developed by Math Works.

It allows matrix manipulations; plotting of functions and data; implementation of algorithms; creation of user interfaces; interfacing with programs written in other languages, including C, C++, Java, and FORTRAN; analyze data; develop algorithms; and create models and applications.

It has numerous built-in commands and math functions that help you in mathematical calculations, generating plots, and performing numerical methods.

In MATLAB environment, every variable is an array or matrix.
Variables can be defined in MATLAB in the following ways:

```
x = 3          % defining x and initializing it with a value
x = sqrt(16)         % defining x and initializing it with an expression
```

The clear command deletes all (or the specified) variable(s) from the memory.

```
clear x    % it will delete x, won't display anything
clear         % it will delete all variables in the workspace
         % peacefully and unobtrusively
```

A vector is a one-dimensional array of numbers. MATLAB allows creating two types of vectors:

- Row vectors
- Column vectors

Row vectors are created by enclosing the set of elements in square brackets, using space or comma to delimit the elements.
For example,

```
r = [7 8 9 10 11]
```

Column vectors are created by enclosing the set of elements in square brackets, usingsemicolon(;) to delimit the elements.

```
c = [7; 8; 9; 10; 11]
```

CreatingMatrices

A matrix is a two-dimensional array of numbers.

In MATLAB, a matrix is created by entering each row as a sequence of space or comma separated elements, and end of a row is demarcated by a semicolon. For example a 3-by-3 matrix is created as:

```
m = [1 2 3; 4 5 6; 7 8 9]
```

MATLAB ENVIRONMENT:

- Command Window
- Command History
- Workspace
- Current Directory
- Figure Window

Command Window:

Whenever MATLAB is invoked, the main window called command window is activated. The command window displays the command prompt'>>' and a cursor where commands are enteredand are executed instantaneously.

Command History Window:

Command history window consists of a list of all the commands that are entered at the commandwindow. These commands remain in the list until they are deleted. Any command may be executed by selecting and double clicking it with the mouse.

Workspace:

A workspace is a collection of all the variables that have been generated so far in the current MATLAB session and shows their data type and size. All the commands executed from Command Window and all the script files executed from the Command Window share commonworkspace, so they can share all the variables.

Current Directory:

The Current Directory window contains all the files and folders present in the Current Directory.To run any file, it must either be in the Current Directory or on the search path.

Edit Window:
An Edit Window is used to create a new program file, or to modify existing files. In this window, programs can be written, edited and saved. The programs written using the MATLAB editor are automatically assigned an extension (.m) by the editor and are known as M- files.

Figure Window:
A Figure Window is a separate window with default white background and is used to display MATLAB graphics. The results of all the graphic commands executed are displayed in the figurewindow.

MATLAB allows two different types of arithmetic operations −

- Matrix arithmetic operations
- Array arithmetic operations

Matrix arithmetic operations are same as defined in linear algebra. Array operations are executedelement by element, both on one dimensional and multi-dimensional array.

The matrix operators and arrays operators are differentiated by the period (.) symbol. However, as the addition and subtraction operation is same for matrices and arrays, the operator is same forboth cases.

The following table gives brief description of the operators −

Operator	**Description**
+	Addition or unary plus. A+B adds the values stored in variables A and B. A and B must have the same size, unless one is a scalar. A scalar can be added to a matrix of any size.
-	Subtraction or unary minus. A-B subtracts the value of B from A. A and B must have the same size, unless one is a scalar. A scalar can be subtracted from a matrix of any size.
*	Matrix multiplication. C = A*B is the linear algebraic product of the matrices A and B. More precisely, $C(i,j) = \sum_{k=1}^{n} A(i,k)B(k,j)$ For non-scalar A and B, the number of columns of A must be equal to the numberof rows of B. A scalar can multiply a matrix of any size.
.*	Array multiplication. A.*B is the element-by-element product of the arrays A and

	B. A and B must have the same size, unless one of them is a scalar.
/	Slash or matrix right division. B/A is roughly the same as B*inv(A). More precisely, B/A = (A'\B')'.
./	Array right division. A./B is the matrix with elements A(i,j)/B(i,j). A and B musthave the same size, unless one of them is a scalar.
\	Backslash or matrix left division. If A is a square matrix, A\B is roughly the same as inv (A)*B, except it is computed in a different way. If A is an n-by-n matrix and B is a column vector with n components, or a matrix with several such columns, then X = A\B is the solution to the equation *AX = B*. A warning messageis displayed if A is badly scaled or nearly singular.
.\	Array left division. A.\B is the matrix with elements B(i,j)/A(i,j). A and B musthave the same size, unless one of them is a scalar.
^	Matrix power. X^p is X to the power p, if p is a scalar. If p is an integer, the power is computed by repeated squaring. If the integer is negative, X is inverted first. For other values of p, the calculation involves eigenvalues and eigenvectors,such that if [V,D] = eig(X), then X^p = V*D.^p/V.
.^	Array power. A. ^B is the matrix with elements A (i,j) to the B(i,j) power. A andB must have the same size, unless one of them is a scalar.
'	Matrix transpose. A' is the linear algebraic transpose of A. For complex matrices,this is the complex conjugate transpose.
.'	Array transpose. A' is the array transpose of A. For complex matrices, this doesnot involve conjugation.

Functions for Arithmetic Operations:-

Apart from the above-mentioned arithmetic operators, MATLAB provides the following commands/functions used for similar purpose −

Function	**Description**
uplus(a)	Unary plus; increments by the amount a
plus (a,b)	Plus; returns a + b
uminus(a)	Unary minus; decrements by the amount a
minus(a, b)	Minus; returns a - b

times(a, b)	Array multiply; returns a.*b
mtimes(a, b)	Matrix multiplication; returns a* b

rdivide(a, b)	Right array division; returns a ./ b
ldivide(a, b)	Left array division; returns a.\ b
mrdivide(A, B)	Solve systems of linear equations $xA = B$ for x
mldivide(A, B)	Solve systems of linear equations $Ax = B$ for x
power(a, b)	Array power; returns a.^b
mpower(a, b)	Matrix power; returns a ^ b

PROGRAM:-

```
a = 10;
b = 20;
c = a + b
d = a - be
= a * bf
= a / b g
= a \ bx =
7;
y = 3;
z = x ^ y
```

TheMFiles

MATLAB allows writing two kinds of program files −

- **Scripts** − script files are program files with **.m extension**. In these files, you write series of commands, which you want to execute together. Scripts do not accept inputs and do not return any outputs. They operate on data in the workspace.

- **Functions** − functions files are also program files with **.m extension**. Functions can accept inputs and return outputs. Internal variables are local to the function.

You can use the MATLAB editor or any other text editor to create your **.m** files.

1. **Hello World**

```
disp('Hello, World!');
```

2. **Simple Addition**

```
a = 5;
b = 10;
c = a + b;
disp(c);
```

3. **For Loop**

```
for i = 1:10
    disp(i);
end
```

4. **While Loop**

```
i = 1;
while i <= 10
    disp(i);
    i = i + 1;
end
```

5. **If-Else Statement**

```
a = 5;
if a > 0
    disp('Positive number');
else
    disp('Negative number');
end
```

6. **Function Definition**

```
function y = square(x)
    y = x^2;
end
```

7. **Plotting a Sine Wave**

```
x = linspace(0, 2*pi, 100);
y = sin(x);
plot(x, y);
title('Sine Wave');
xlabel('x');
ylabel('sin(x)');
```

8. **Matrix Addition**

```
A = [1 2; 3 4];
B = [5 6; 7 8];
C = A + B;
disp(C);
```

9. **Matrix Multiplication**

```
A = [1 2; 3 4];
B = [5 6; 7 8];
C = A * B;
disp(C);
```

10. **Element-wise Multiplication**

```
A = [1 2; 3 4];
B = [5 6; 7 8];
C = A .* B;
disp(C);
```

11. **Transpose of a Matrix**

```
A = [1 2; 3 4];
B = A';
disp(B);
```

12. **Inverse of a Matrix**

```
A = [1 2; 3 4];
B = inv(A);
disp(B);
```

13. **Solving Linear Equations**

```
A = [1 2; 3 4];
B = [5; 6];
X = A\B;
disp(X);
```

14. **Finding Eigenvalues**

```
A = [1 2; 3 4];
eigvals = eig(A);
disp(eigvals);
```

15. **Creating a 3D Plot**

```
[X, Y] = meshgrid(-2:0.1:2, -2:0.1:2);
Z = X.^2 + Y.^2;
surf(X, Y, Z);
title('3D Plot');
```

16. **Saving a Variable to File**

```
a = 10;
save('myVariable.mat', 'a');
```

17. **Loading a Variable from File**

```
load('myVariable.mat');
disp(a);
```

18. **Reading Data from a File**

```
data = load('data.txt');
disp(data);
```

19. **Writing Data to a File**

```
data = [1 2 3; 4 5 6; 7 8 9];
save('output.txt', 'data', '-ascii');
```

20. **Calculating the Mean**

```
data = [1, 2, 3, 4, 5];
mean_val = mean(data);
disp(mean_val);
```

21. **Calculating the Standard Deviation**

```
data = [1, 2, 3, 4, 5];
std_val = std(data);
disp(std_val);
```

22. **Finding the Maximum Value**

```
data = [1, 2, 3, 4, 5];
max_val = max(data);
disp(max_val);
```

23. **Finding the Minimum Value**

```
data = [1, 2, 3, 4, 5];
min_val = min(data);
disp(min_val);
```

24. **Sorting an Array**

```
data = [5, 3, 1, 4, 2];
sorted_data = sort(data);
disp(sorted_data);
```

25. **Creating a Histogram**

```
data = randn(1000, 1);
histogram(data);
title('Histogram');
```

26. **Generating Random Numbers**

```
random_numbers = rand(1, 10);
disp(random_numbers);
```

27. **Solving a System of Equations**

```
syms x y;
eq1 = 2*x + 3*y == 6;
eq2 = 4*x - y == 5;
sol = solve([eq1, eq2], [x, y]);
disp(sol.x);
disp(sol.y);
```

28. **Symbolic Differentiation**

```
syms x;
f = x^2 + 3*x + 2;
f_prime = diff(f);
disp(f_prime);
```

29. **Symbolic Integration**

```
syms x;
f = x^2 + 3*x + 2;
f_int = int(f);
disp(f_int);
```

30. **Creating a Subplot**

```
x = linspace(0, 2*pi, 100);
y1 = sin(x);
y2 = cos(x);
subplot(2,1,1);
plot(x, y1);
title('Sine Wave');
subplot(2,1,2);
plot(x, y2);
title('Cosine Wave');
```

Sample Experiments

Basic Mathematics and Operations

Experiment 1.1: Matrix Operations

Theory: Matrix operations are fundamental in MATLAB. They include addition, subtraction, multiplication, and division .

Code:

```
A = [1 2; 3 4];
B = [5 6; 7 8];

% Matrix Addition
C = A + B;

% Matrix Subtraction
D = A - B;

% Matrix Multiplication
E = A * B;

% Element-wise Multiplication
F = A .* B;

% Matrix Division
G = A / B;

% Display Results
disp('Matrix Addition:'), disp(C)
disp('Matrix Subtraction:'), disp(D)
disp('Matrix Multiplication:'), disp(E)
disp('Element-wise Multiplication:'), disp(F)
disp('Matrix Division:'), disp(G)
```

Experiment 1.2: Solving Linear Equations

Theory: Solving a system of linear equations can be achieved using matrix inversion or MATLAB's backslash operator .

Code:

```
A = [2 1; 3 4];
b = [5; 6];

% Solve using inverse
x1 = inv(A) * b;

% Solve using backslash operator
x2 = A \ b;

% Display Results
disp('Solution using inverse:'), disp(x1)
disp('Solution using backslash operator:'), disp(x2)
```

2. Data Analysis and Visualization

Experiment 2.1: Plotting Data

Theory: Visualizing data using different types of plots helps in understanding the data better.

Code:

```
x = 0:0.1:10;
y = sin(x);

figure;
plot(x, y);
title('Sine Wave');
xlabel('x');
ylabel('sin(x)');
grid on;
```

Experiment 2.2: Histogram

Theory: Histograms represent the distribution of data.
Code:

```
data = randn(1000, 1);

figure;
histogram(data);
title('Histogram of Random Data');
xlabel('Data Values');
ylabel('Frequency');
grid on;
```

3. Signal Processing

Experiment 3.1: FFT of a Signal

Theory: Fast Fourier Transform (FFT) is used to compute the frequency spectrum of a signal.
Code:

```
Fs = 1000;              % Sampling frequency
T = 1/Fs;               % Sampling period
L = 1000;               % Length of signal
t = (0:L-1)*T;          % Time vector

% Create a signal
S = 0.7*sin(2*pi*50*t) + sin(2*pi*120*t);

% Compute the FFT
Y = fft(S);

% Compute the two-sided spectrum P2 and the single-sided spectrum P1
P2 = abs(Y/L);
P1 = P2(1:L/2+1);
P1(2:end-1) = 2*P1(2:end-1);
```

```
% Define the frequency domain f
f = Fs*(0:(L/2))/L;

% Plot single-sided amplitude spectrum P1
figure;
plot(f,P1)
title('Single-Sided Amplitude Spectrum of S(t)')
xlabel('f (Hz)')
ylabel('|P1(f)|')
```

4. Control Systems

Experiment 4.1: Step Response of a Transfer Function

Theory: The step response of a system is an important characteristic in control systems.

Code:

```
num = [1];
den = [1 3 2];
sys = tf(num, den);

figure;
step(sys);
title('Step Response');
```

5. Optimization

Experiment 6.1: Minimizing a Function

Theory: Optimization involves finding the minimum or maximum of a function.

Code:

```
fun = @(x) (x-2)^2 + 3;
x0 = 0; % Initial guess
x = fminsearch(fun, x0);

disp('The minimum value is at:'), disp(x)
```

Bangabandhu Sheikh Mujibur Rahman Aviation and Aerospace University (BSMRAAU)
Modelling and Simulation Sessional

6. Numerical Methods
Experiment 6.1: Solving ODEs

Theory: Solving ordinary differential equations (ODEs) is essential in many scientific fields. Code:

```
% Define the ODE
ode = @(t, y) -2*y;

% Initial condition
y0 = 1;

% Time span
tspan = [0 5];

% Solve ODE
[t, y] = ode45(ode, tspan, y0);

% Plot results
figure;
plot(t, y);
title('Solution of ODE');
xlabel('Time');
ylabel('y(t)');
grid on;
```

7.Simulations
Experiment 7.1: Monte Carlo Simulation

Theory: Monte Carlo simulations use random sampling to solve problems.

Code:

```
% Number of simulations
N = 10000;

% Generate random samples
samples = rand(N, 1);

% Compute estimate of pi
inside_circle = sum(samples.^2 + rand(N, 1).^2 <= 1);
pi_estimate = 4 * inside_circle / N;

disp('Estimated value of pi:'), disp(pi_estimate)
```

10. File I/O and Data Import/Export

Experiment 10.1: Reading and Writing CSV Files

Theory: Importing and exporting data is crucial for data analysis and sharing.
Code:

```
% Generate sample data
data = rand(10, 3);

% Write data to CSV
csvwrite('data.csv', data);

% Read data from CSV
data_read = csvread('data.csv');

disp('Data read from CSV:'), disp(data_read)
```

THEORY:-

a) Impulse Function

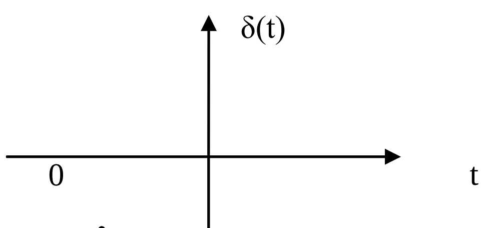
The impulse function is defined as

$$\int_{-\infty}^{\infty} \delta(t)dt = 1$$

and

$\delta(t = 0 \text{ for } t \neq 0)$

That is the impulse function has zero amplitude everywhere expect at t = 0.

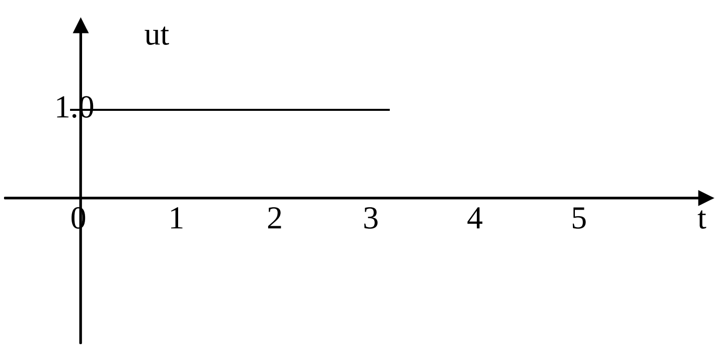


b) Step Function

The unit step function is defined as

$U(t) = 1 \quad \text{for } t \geq 0$

$\quad\;\; = 0 \quad \text{for } t > 0$



Program:

```
t=(-2:0.01:10);
impulse = t==0;
unitstep = t>=0;
plot(t, impulse)
plot(t, unitstep)
ramp = t.*unitstep;
plot(t,ramp)
xlabel('Time')
ylabel('Amplitude')
title('impulse  function')
title('unit step function')
title('ramp function')
```

c) 2-D plot

Define x as a vector of linearly spaced values between 0 and 2π. Use an increment of $\pi/10$ between the values. Define y as sine values of x.

Program:

```
clc
clear all
x=(0:pi/10:2*pi)
y=sin(x)
plot(x,y)
title('2D Plot')
xlabel('Time')
ylabel('Amplitude')
```

d) 3-D plot

A three-dimensional plot may refer to

a) A graph or plot embedded into a three-dimensional space

b) The plot of a function of two variables, embedded into a three-dimensional space

Program:

```
clc
clear all
t=(-4:0.01:4)
x=t.^2
y=4*t
plot3(x,y,t)
grid on
xlabel('x-axis')
ylabel('y-axis')
```

```
zlabel('z-axis')
title('3D Plot')
```

RESULTS:

Results have been seen on the command window.

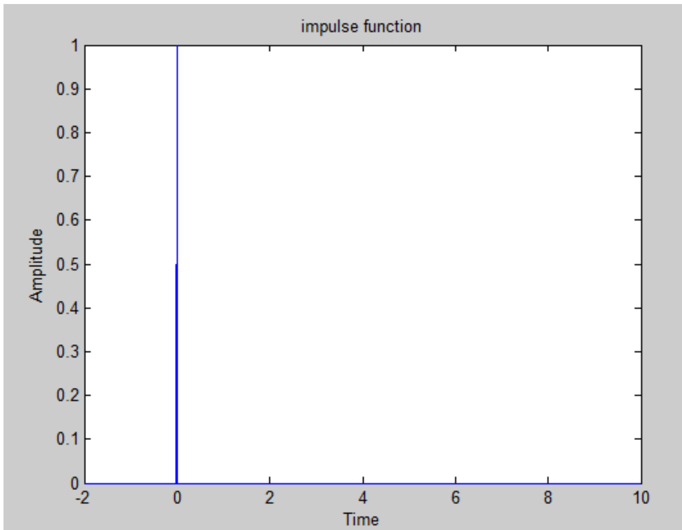


Figure-1

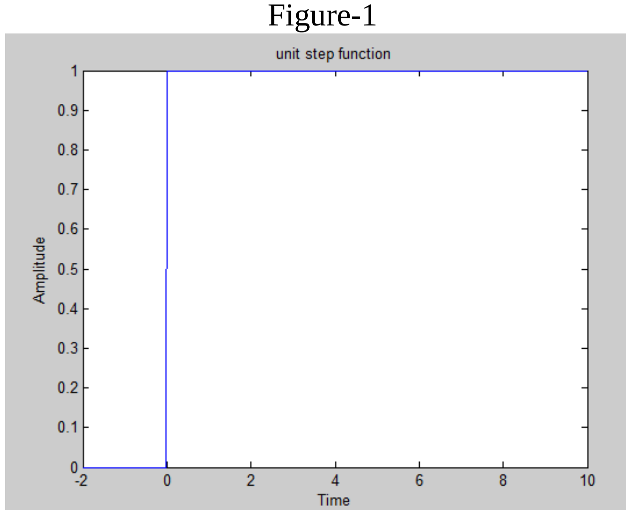


Figure-2

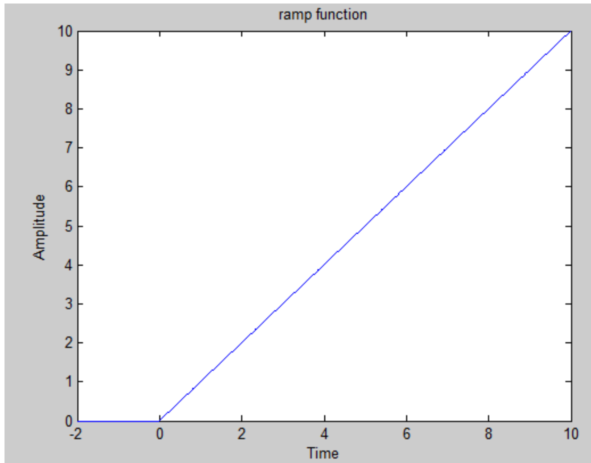


Figure-3

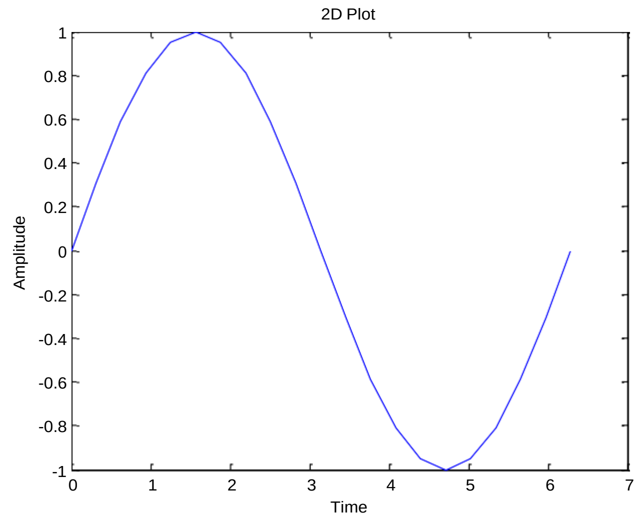
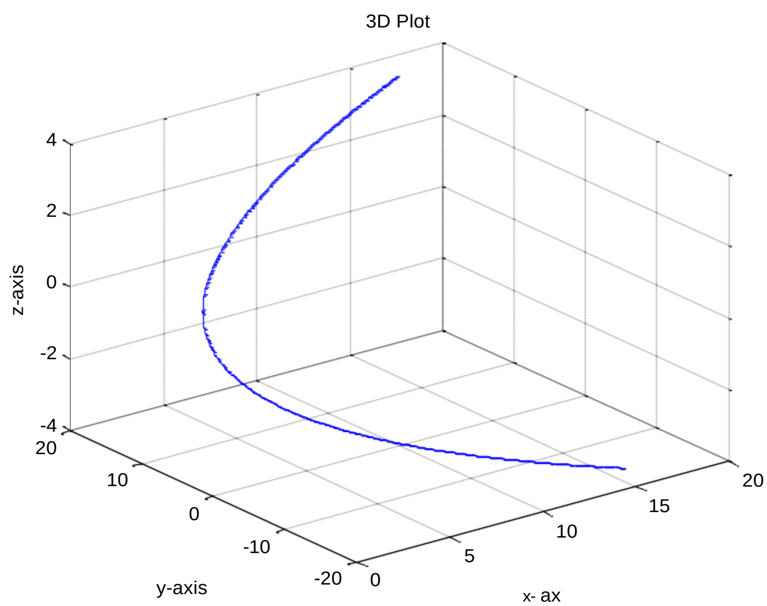

2D Plot
Amplitude
Time
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
1
2
3
4
5
6
7

Figure-4


3D Plot
z-axis
y-axis
x- ax
4
2
0
-2
-4
20
10
0
-10
-20
0
5
10
15
20

THEORY:

Convolution is a mathematical operation used to express the relation between input and output of an LTI system. It relates input, output and impulse response of an LTI system as

$$y(t)=x(t)*h(t)$$

Where y (t) = output of LTI

x (t) = input of LTI

h (t) = impulse response of LTI

There are two types of convolutions:

a) Continuous Convolution

$$\begin{aligned} y(t) &= x(t) * h(t) \\ &= \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau \end{aligned}$$

b) Discrete Convolution

$$\begin{aligned} y(n) &= x(n) * h(n) \\ &= \Sigma_{k=-\infty}^{\infty} x(k)h(n-k) \end{aligned}$$

PROGRAM:

```
clc;
clear all;
close;
disp('enter the length of the first sequence m=');
m=input('');
disp('enter the first sequence x[m]=');
for i=1:m
   x(i)=input('');
end
disp('enter the length of the second sequence n=');
n=input('');
disp('enter the second sequence h[n]=');
for j=1:n
   h(j)=input('');
end
y=conv(x,h);
figure;
subplot(3,1,1);
stem(x);
ylabel ('amplitude---->');
xlabel('n--- >');
title('x(n) Vs n');
subplot(3,1,2);
stem(h);
ylabel('amplitude --->');
xlabel('n--- >');
title('h(n) Vs n');
subplot(3,1,3);
stem(y);
ylabel('amplitude --->');
xlabel('n--- >');
title('y(n) Vs n');
disp('linear convolution of x[m] and h[n] is y');
```

<u>INPUT</u>:--

Enter the length of the first sequence m=

Enter the length of first sequence x[m]=

1

2

3

4

5

6

Enter the length of the second sequence n=

6

Enter the length of second sequence h[n]=

1

2

3

4

5

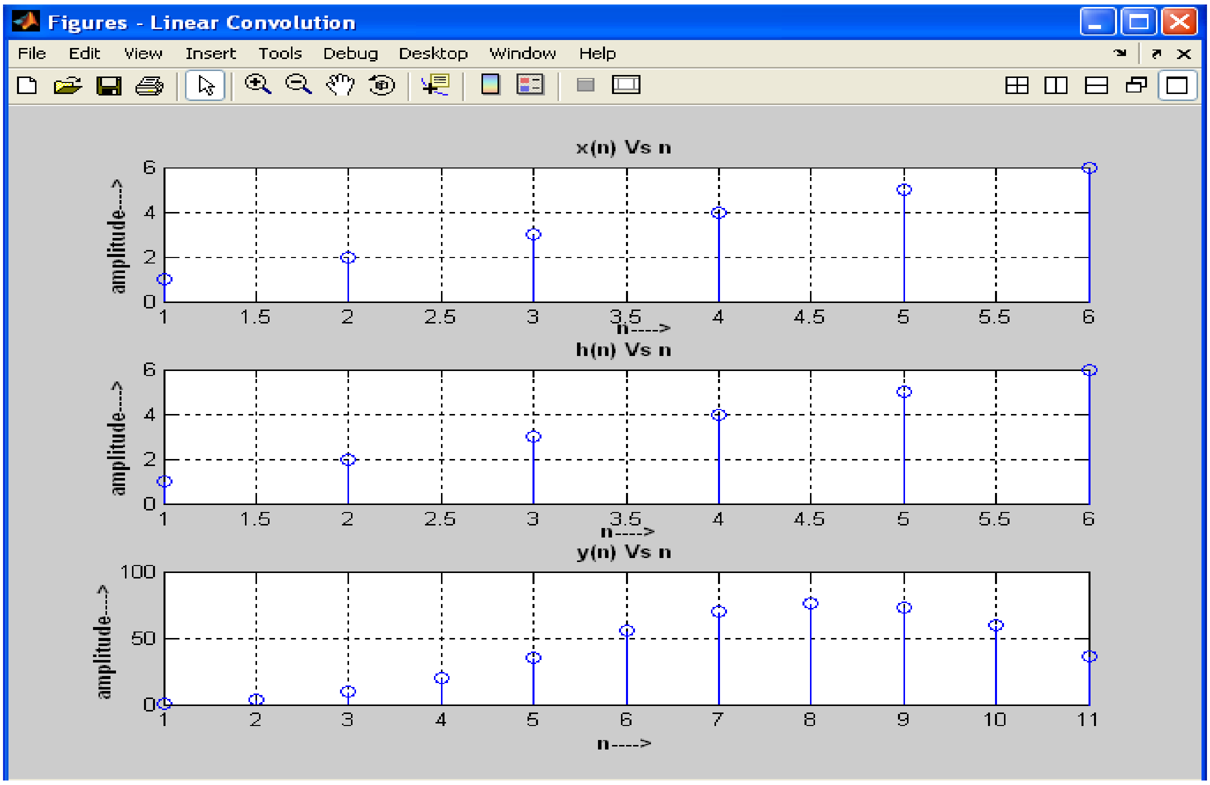
6

<u>OUTPUT</u>:-

Linear convolution of x[m] and h[n] is y=

1 4 10 20 35 56 70 76 73 60 36

<u>RESULTS: -</u> Thus the program for linear convolution is written using MATLAB and verified.

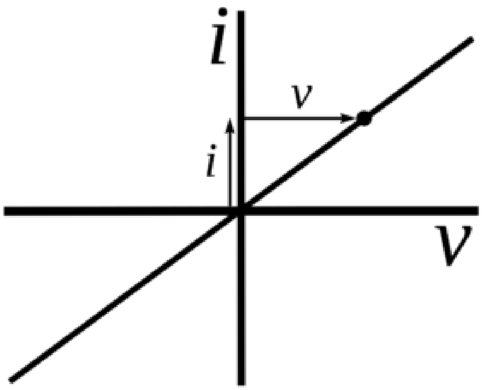


1. *Ohm's Law Verification*

Ohm’s law states that the voltage or potential difference between two points is directly proportional to the current or electricity passing through the resistance, and directly proportional to the resistance of the circuit. The formula for Ohm’s law is V=IR. This relationship between current, voltage, and relationship was discovered by German scientist Georg Simon Ohm. Let us learn more about Ohms Law, Resistance, and its applications.

Ohm’s Law Definition

Most basic components of electricity are voltage, current, and resistance. Ohm’s law shows a simple relation between these three quantities. **Ohm’s law** states that the current through a conductor between two points is directly proportional to the voltage across the two points.



Ohm's Law Formula

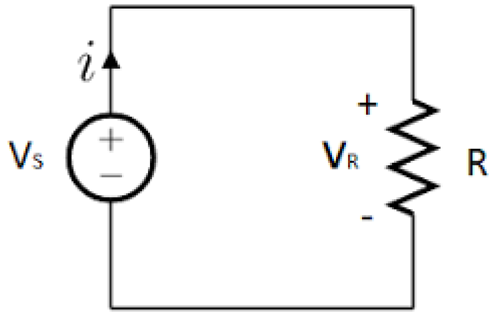
Voltage= Current× Resistance

V= I×R

V= voltage, I= current and R= resistance

The SI unit of resistance is **ohms** and is denoted by **Ω**

This law is one of the most basic laws of electricity. It helps to calculate the power, efficiency, current, voltage, and resistance of an element of an electrical circuit.



Objective: Verify Ohm's Law using MATLAB.

Steps:

1. Define the resistance and current.
2. Calculate the voltage using Ohm's Law.
3. Display the result.

Code:

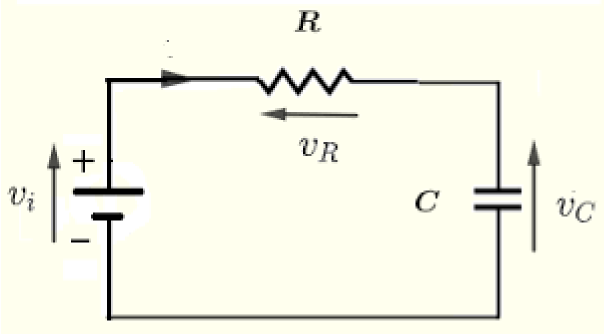
```
% Step 1: Define the resistance and current
R = 10; % resistance in ohms
I = 2; % current in amperes

% Step 2: Calculate the voltage
V = R * I;

% Step 3: Display the result
disp(['Voltage (V) = ', num2str(V), ' V']);
```

2. Series RC Circuit Response

Objective: Analyze the step response of a series RC circuit.

Steps:

1. Open Simulink and create a new model.
2. Add a resistor, capacitor, and step input to the model.
3. Connect the components in series.
4. Run the simulation and observe the voltage across the capacitor.



Simulink Setup:

1. Drag and drop a Resistor, Capacitor, and Step block from the library.
2. Connect them in series.
3. Add a Scope block to visualize the voltage across the capacitor.
4. Configure the parameters (e.g., R = 1 ohm, C = 1 F).

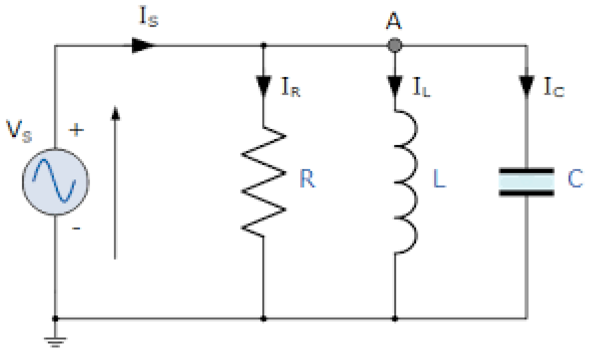
Steps:

1. Set up the Simulink model as described.
2. Run the simulation.
3. Observe the voltage across the capacitor on the Scope.

3. Parallel RLC Circuit

Objective: Study the resonance frequency of a parallel RLC circuit.
Steps:

1. Define the values of R, L, and C.
2. Calculate the resonance frequency.
3. Plot the impedance vs. frequency.

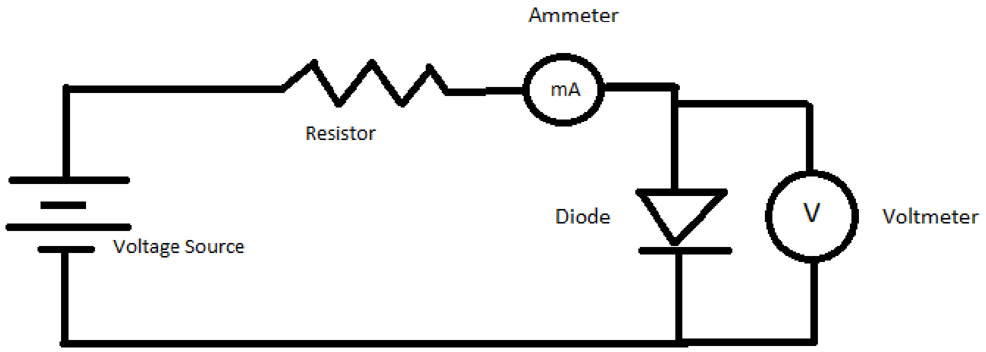


Code:
```
% Step 1: Define the values of R, L, and C
R = 10; % resistance in ohms
L = 1e-3; % inductance in henries
C = 1e-6; % capacitance in farads

% Step 2: Calculate the resonance frequency
f_res = 1 / (2 * pi * sqrt(L * C));
```

```
disp(['Resonance Frequency (Hz) = ', num2str(f_res)]);

% Step 3: Plot the impedance vs. frequency
f = linspace(0, 2*f_res, 1000);
Z = abs(R + 1./(1i*2*pi*f*C) + 1i*2*pi*f*L);
plot(f, Z);
xlabel('Frequency (Hz)');
ylabel('Impedance (Ohms)');
title('Impedance vs Frequency');
grid on;
```

4. Diode Characteristics

Objective: Plot the I-V characteristics of a diode.



Steps:

1. Define the diode equation.
2. Plot the current vs. voltage.

Code:

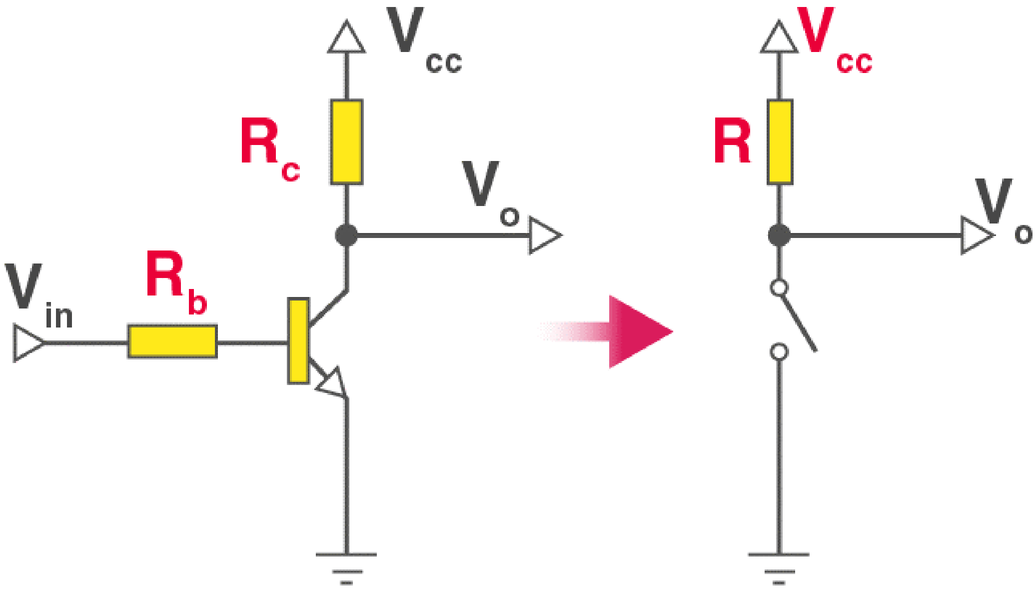
```
% Step 1: Define the diode equation
Is = 1e-12; % saturation current in amperes
Vt = 0.025; % thermal voltage in volts

% Step 2: Plot the current vs. voltage
V = linspace(-0.7, 0.7, 1000);
I = Is * (exp(V/Vt) - 1);
plot(V, I);
xlabel('Voltage (V)');
ylabel('Current (I)');
title('I-V Characteristics of a Diode');
grid on;
```

5. Transistor as a Switch

Objective: Demonstrate the operation of a transistor in switch mode.

Steps:

1. Open Simulink and create a new model.
2. Add an NPN transistor, resistor, and DC voltage source.
3. Connect the components to form a switch circuit.
4. Run the simulation and observe the output.



Simulink Setup:

1. Drag and drop an NPN Transistor, Resistor, DC Voltage Source, and Scope block.
2. Connect the transistor base to a pulse generator, the collector to the voltage source via a resistor, and the emitter to ground.
3. Connect the Scope to the collector-emitter voltage.
4. Configure the pulse generator to provide a switching signal.

Steps:

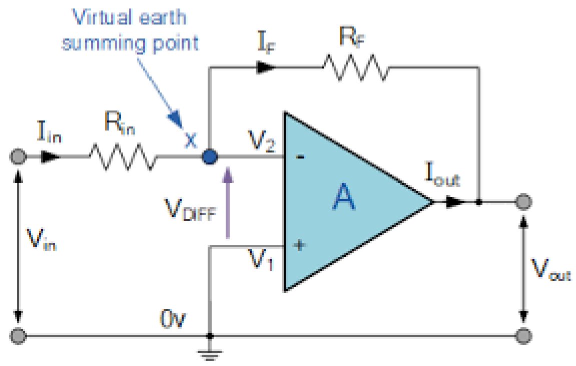
1. Set up the Simulink model as described.
2. Run the simulation.
3. Observe the collector-emitter voltage on the Scope.

6. Operational Amplifier (Op-Amp) as an Inverting Amplifier

Objective: Verify the inverting amplification using Op-Amp.

Steps:

1. Open Simulink and create a new model.
2. Add an Op-Amp, resistors, and DC voltage source.
3. Connect the components to form an inverting amplifier circuit.
4. Run the simulation and observe the output.



Simulink Setup:

1. Drag and drop an Op-Amp block, Resistors, and DC Voltage Source.
2. Connect the inverting input of the Op-Amp to the input signal via a resistor.
3. Connect the feedback resistor between the output and inverting input.
4. Ground the non-inverting input.
5. Connect a voltage source to the input and a Scope to the output.

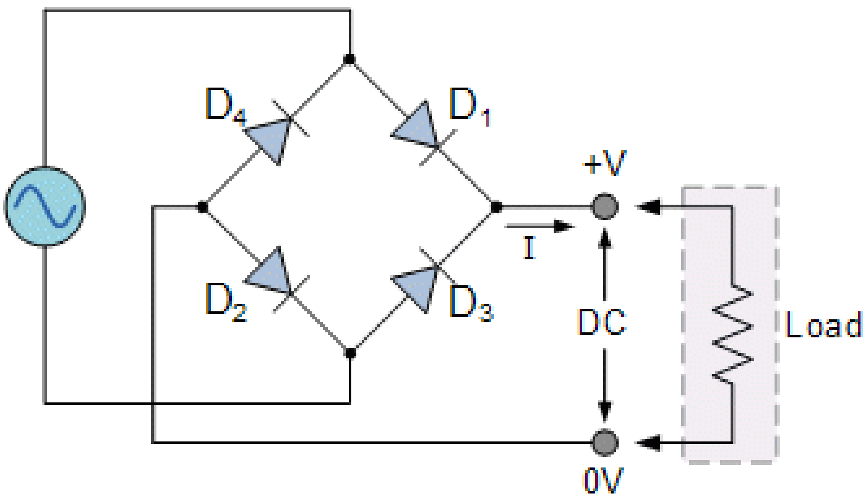
Steps:

1. Set up the Simulink model as described.
2. Run the simulation.
3. Observe the output voltage on the Scope.

7. Full-Wave Rectifier

Objective: Simulate a full-wave rectifier and analyze the output waveform.
Steps:

1. Open Simulink and create a new model.
2. Add diodes, resistors, and a sine wave generator.
3. Connect the components to form a full-wave rectifier circuit.
4. Run the simulation and observe the output.



Simulink Setup:

1. Drag and drop Diodes, Resistors, Sine Wave Generator, and Scope block.
2. Connect the diodes in a bridge configuration.
3. Connect the load resistor to the bridge output.
4. Connect the Sine Wave Generator to the bridge input.
5. Connect a Scope to the output.

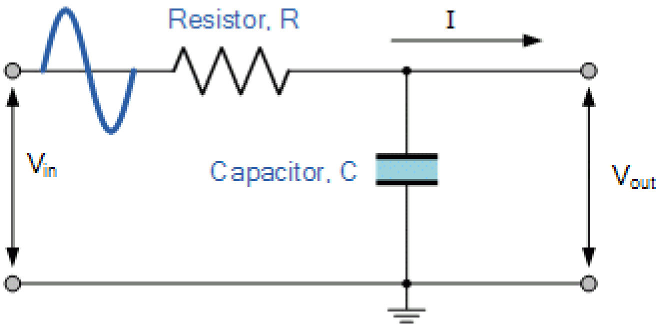
Steps:

1. Set up the Simulink model as described.
2. Run the simulation.
3. Observe the rectified output on the Scope.

8. RC Low-Pass Filter

Objective: Design and analyze the frequency response of an RC low-pass filter.
Steps:

1. Define the values of R and C.
2. Calculate the transfer function.
3. Plot the frequency response.



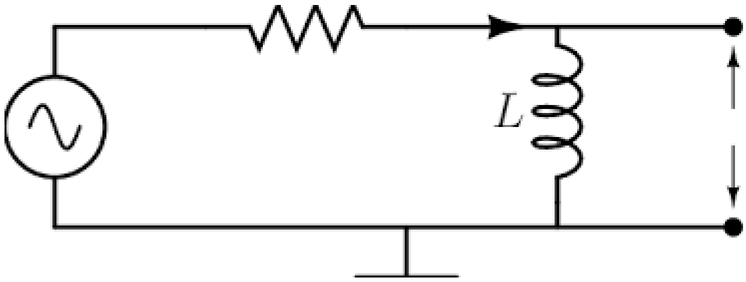
Code:

```
% Step 1: Define the values of R and C
R = 1e3; % resistance in ohms
C = 1e-6; % capacitance in farads

% Step 2: Calculate the transfer function
H = tf([1], [R*C 1]);

% Step 3: Plot the frequency response
bode(H);
title('Frequency Response of RC Low-Pass Filter');
grid on;
```

9. RL High-Pass Filter

Objective: Design and analyze the frequency response of an RL high-pass filter.
Steps:

1. Define the values of R and L.
2. Calculate the transfer function.
3. Plot the frequency response.



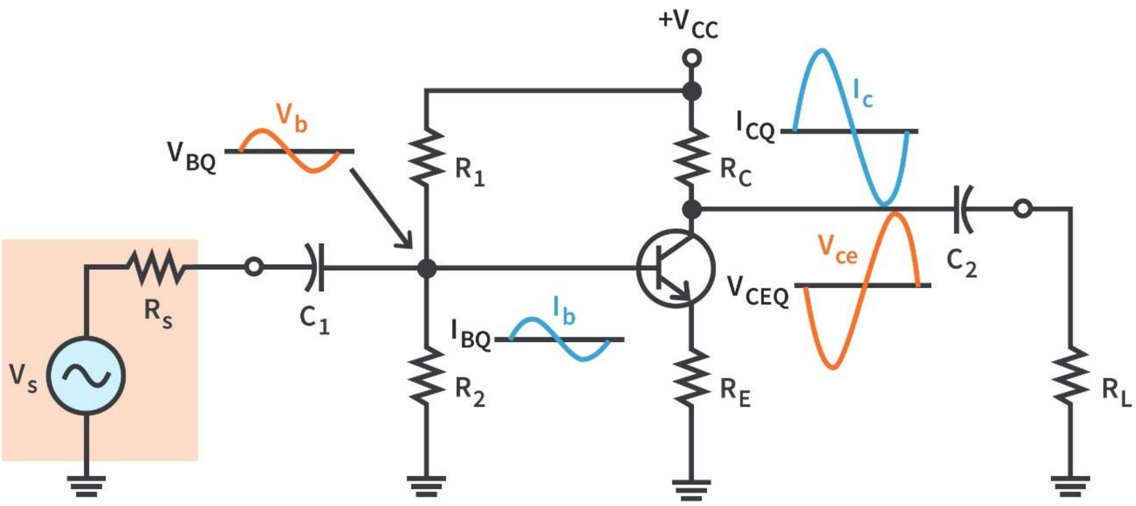
Code:

```
% Step 1: Define the values of R and L
R = 1e3; % resistance in ohms
L = 1e-3; % inductance in henries

% Step 2: Calculate the transfer function
H = tf([L 0], [L R]);

% Step 3: Plot the frequency response
bode(H);
title('Frequency Response of RL High-Pass Filter');
grid on;
```

10. BJT Amplifier

Objective: Simulate the frequency response of a BJT amplifier.
Steps:

1. Open Simulink and create a new model.
2. Add a BJT transistor, resistors, capacitors, and a sine wave generator.
3. Connect the components to form a common-emitter amplifier circuit.
4. Run the simulation and observe the output.



Simulink Setup:

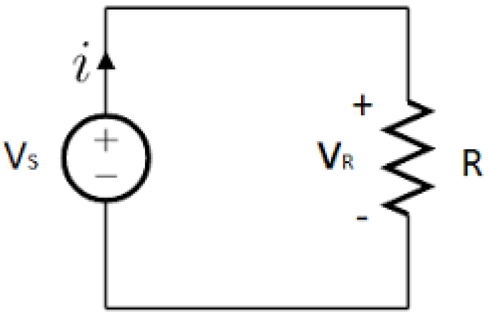
1. Drag and drop a BJT Transistor, Resistors, Capacitors, Sine Wave Generator, and Scope block.
2. Connect the BJT in a common-emitter configuration.
3. Add a bypass capacitor and load resistor to the collector.
4. Connect the Sine Wave Generator to the base through a coupling capacitor.
5. Connect a Scope to the output.

Steps:

1. Set up the Simulink model as described.
2. Run the simulation.
3. Observe the amplifier output on the Scope.

1. Ohm's Law

Theory: Ohm's Law states that the current through a conductor between two points is directly proportional to the voltage across the two points.

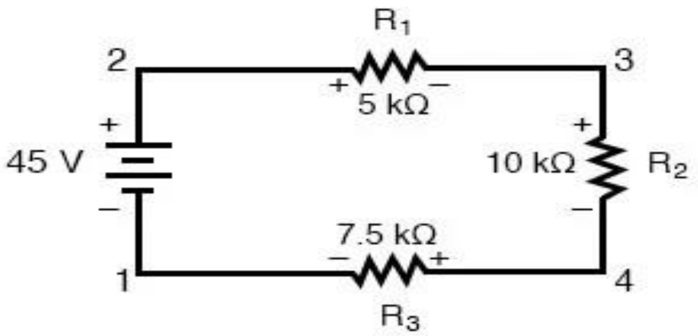


MATLAB Code:

```
% Ohm's Law: V = I * R
R = 10; % resistance in ohms
I = 2;  % current in amperes
V = I * R; % voltage in volts
disp(['Voltage: ', num2str(V), ' V']);
```

2. Kirchhoff's Voltage Law (KVL)

Theory: The sum of all electrical voltages around a loop is zero.



MATLAB Code:

```
% Kirchhoff's Voltage Law example
V1 = 5; % Voltage source 1 in volts
V2 = 10; % Voltage source 2 in volts
V3 = -3; % Voltage drop in volts
total_voltage = V1 + V2 + V3; % Should be zero in a closed loop
disp(['Total Voltage: ', num2str(total_voltage), ' V']);
```

3. Series RLC Circuit Response

Theory: Analyze the response of a series RLC circuit to a step input.
MATLAB Code:

```
% RLC Circuit Response
R = 5; % Resistance in ohms
L = 1e-3; % Inductance in henrys
C = 100e-6; % Capacitance in farads
sys = tf([1], [L*C R*C 1]);
step(sys);
```

```
title('Series RLC Circuit Step Response');
```

4. Fourier Transform

Fourier Transform is a mathematical model which helps to transform the signals between two different domains, such as transforming signal from frequency domain to time domain or vice versa. Fourier transform has many applications in Engineering and Physics, such as signal processing, RADAR, and so on. In this article, we are going to discuss the formula of Fourier transform, properties, tables, Fourier cosine transform, Fourier sine transform with complete explanations.

What is Fourier Transform?

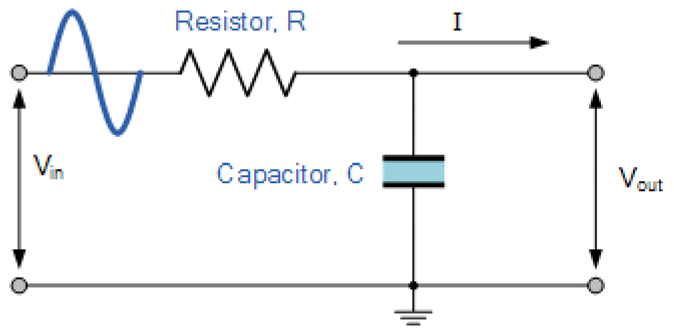
The generalisation of the complex Fourier series is known as the Fourier transform. The term "Fourier transform" can be used in the mathematical function, and it is also used in the representation of the frequency domain. The Fourier transform helps to extend the Fourier series to the non-periodic functions, which helps us to view any functions in terms of the sum of simple sinusoids.

MATLAB Code:

```
% Fourier Transform of a sine wave
Fs = 1000; % Sampling frequency
t = 0:1/Fs:1-1/Fs;
f = 5; % Frequency of sine wave
x = sin(2*pi*f*t);
X = fft(x);
n = length(x);
f = (0:n-1)*(Fs/n); % Frequency range
magnitude = abs(X);
plot(f,magnitude);
title('Fourier Transform');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
```

5. RC Low Pass Filter

Theory: An RC low-pass filter allows signals with a frequency lower than a certain cutoff frequency to pass through.



MATLAB Code:

```
% RC Low Pass Filter
R = 1e3; % Resistance in ohms
C = 1e-6; % Capacitance in farads
fc = 1/(2*pi*R*C); % Cutoff frequency
sys = tf([1], [R*C 1]);
bode(sys);
title('RC Low Pass Filter');
```

6. AC Voltage Analysis

In our tutorial about the AC Waveform we looked briefly at the **RMS Voltage** value of a sinusoidal waveform and said that this RMS value gives the same heating effect as an equivalent DC power and in this tutorial we will expand on this theory a little more by looking at RMS voltages and currents in more detail.

The term "RMS" stands for "Root-Mean-Squared". Most books define this as the "amount of AC power that produces the same heating effect as an equivalent DC power", or something similar along these lines, but an RMS value is more than just that.

The RMS value is the square root of the mean (average) value of the squared function of the instantaneous values. The symbols used for defining an RMS value are V_{RMS} or I_{RMS}.

The term RMS, ONLY refers to time-varying sinusoidal voltages, currents or complex waveforms where the magnitude of the waveform changes over time and is not used in DC circuit analysis or calculations where the magnitude is always constant.

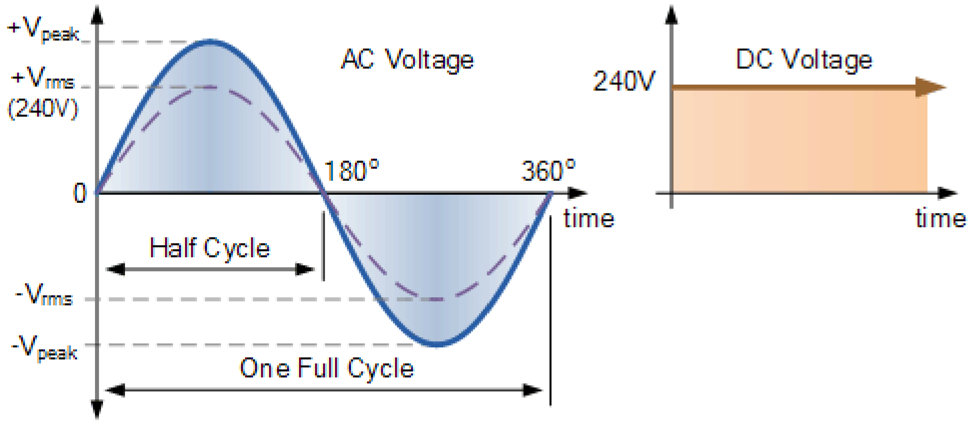
When used to compare the equivalent RMS voltage value of an alternating sinusoidal waveform that supplies the same electrical power to a given load as an equivalent DC circuit, the RMS value is called the "effective value" and is generally presented as: V_{eff} or I_{eff}.

In other words, the effective value is an equivalent DC value which tells you how many volts or amps of DC that a time-varying sinusoidal waveform is equal to in terms of its ability to produce the same power.

For example, the domestic mains supply in the United Kingdom is 240Vac. This value is assumed to indicate an effective value of "240 Volts rms". This means then that the sinusoidal rms voltage from the wall sockets of a UK home is capable of producing the same average positive power as 240 volts of steady DC voltage as shown below.

RMS Voltage Equivalent

Bangabandhu Sheikh Mujibur Rahman Aviation and Aerospace University (BSMRAAU)
Modelling and Simulation Sessional



So how do we calculated the **RMS Voltage** of a sinusoidal waveform. The RMS voltage of a sinusoid or complex waveform can be determined by two basic methods.

- Graphical Method – which can be used to find the RMS value of any non-sinusoidal time-varying waveform by drawing a number of mid-ordinates onto the waveform.
- Analytical Method – is a mathematical procedure for finding the effective or RMS value of any periodic voltage or current using calculus.

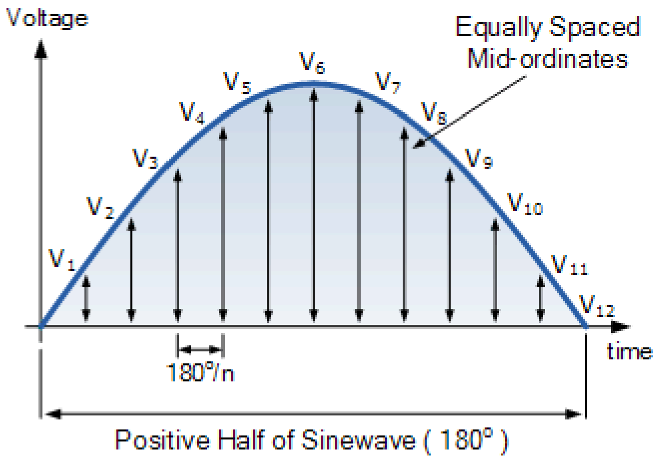
RMS Voltage Graphical Method

Whilst the method of calculation is the same for both halves of an AC waveform, for this example we will consider only the positive half cycle. The effective or rms value of a waveform can be found with a reasonable amount of accuracy by taking equally spaced instantaneous values along the waveform.

The positive half of the waveform is divided up into any number of "n" equal portions or *mid-ordinates* and the more mid-ordinates that are drawn along the waveform, the more accurate will be the final result.

The width of each mid-ordinate will therefore be n° degrees and the height of each mid-ordinate will be equal to the instantaneous value of the waveform at that time along the x-axis of the waveform.

Graphical Method



Each mid-ordinate value of a waveform (the voltage waveform in this case) is multiplied by itself (squared) and added to the next. This method gives us the "square" or **Squared** part of the RMS voltage expression.

Next this squared value is divided by the number of mid-ordinates used to give us the **Mean** part of the RMS voltage expression, and in our simple example above the number of mid-ordinates used was twelve (12). Finally, the square root of the previous result is found to give us the **Root** part of the RMS voltage.

Then we can define the term used to describe an rms voltage (V_{RMS}) as being "the square root of the mean of the square of the mid-ordinates of the voltage waveform" and this is given as:

$$V_{RMS} = \sqrt{\frac{\text{sum of mid-ordinate (voltages)}^2}{\text{number of mid-ordinates}}}$$

and for our simple example above, the RMS voltage will be calculated as:

$$V_{RMS} = \sqrt{\frac{V_1^2 + V_2^2 + V_3^2 + V_4^2 \dots + V_{11}^2 + V_{12}^2}{12}}$$

So let us assume that an alternating voltage has a peak voltage (V_{pk}) of 20 volts and by taking 10 mid-ordinate values is found to vary over one half cycle as follows:

Voltage	6.2V	11.8V	16.2V	19.0V	20.0V	19.0V	16.2V	11.8V	6.2V	0V
Angle	18^o	36^o	54^o	72^o	90^o	108^o	126^o	144^o	162^o	180^o

The **RMS voltage** is therefore calculated as:

$$V_{RMS} = \sqrt{\frac{6.2^2 + 11.8^2 + 16.2^2 + 19^2 + 20^2 + 19^2 + 16.2^2 + 11.8^2 + 6.2^2 + 0^2}{10}}$$

$$V_{RMS} = \sqrt{\frac{2000}{10}} = \sqrt{200} = 14.14\,\text{Volts}$$

Then the RMS Voltage value using the graphical method is given as: 14.14 Volts.

RMS Voltage Analytical Method

The graphical method above is a very good way of finding the effective or RMS voltage, (or current) of an alternating waveform that is not symmetrical or sinusoidal in nature. In other words the waveform shape resembles that of a complex waveform.

However, when dealing with pure sinusoidal waveforms we can make life a little bit easier for ourselves by using an analytical or mathematical way of finding the RMS value.

A periodic sinusoidal voltage is constant and can be defined as $V_{(t)} = V_{max}*\cos(\omega t)$ with a period of T. Then we can calculate the **root-mean-square** (rms) value of a sinusoidal voltage ($V_{(t)}$) as:

$$V_{RMS} = \sqrt{\frac{1}{T}\int_0^T V_m^2 \cos^2(\omega t)\, dt}$$

Integrating through with limits taken from 0 to 360° or "T", the period gives:

$$V_{RMS} = \sqrt{\frac{V_m^2}{2T}\left[t + \frac{1}{2\omega}\sin(2\omega t)\right]_0^T}$$

Where: Vm is the peak or maximum value of the waveform. Dividing through further as ω = 2π/T, the complex equation above eventually reduces down too:

RMS Voltage Equation

$$V_{RMS} = V_{pk}\frac{1}{\sqrt{2}} = V_{pk} \times 0.7071$$

Then the RMS voltage (V_{RMS}) of a sinusoidal waveform is determined by multiplying the peak voltage value by **0.7071**, which is the same as one divided by the square root of two (1/√2).

The RMS voltage, which can also be referred to as the effective value, depends on the magnitude of the waveform and is not a function of either the waveforms frequency nor its phase angle.

From the graphical example above, the peak voltage (V_{pk}) of the waveform was given as 20 Volts. By using the analytical method just defined we can calculate the RMS voltage as being:

$V_{RMS} = V_{pk} * 0.7071 = 20 \times 0.7071 = 14.14V$

Note that this value of 14.14 volts is the same value as for the previous graphical method. Then we can use either the graphical method of mid-ordinates, or the analytical method of calculation to find the RMS voltage or current values of a sinusoidal waveform.

Note that multiplying the peak or maximum value by the constant 0.7071, **ONLY** applies to sinusoidal waveforms. For non-sinusoidal waveforms the graphical method must be used.

But as well as using the peak or maximum value of the sinusoid, we can also use the peak-to-peak (V_{P-P}) value or the average (V_{AVG}) value to find the sinusoids equivalent root mean squared value as shown:

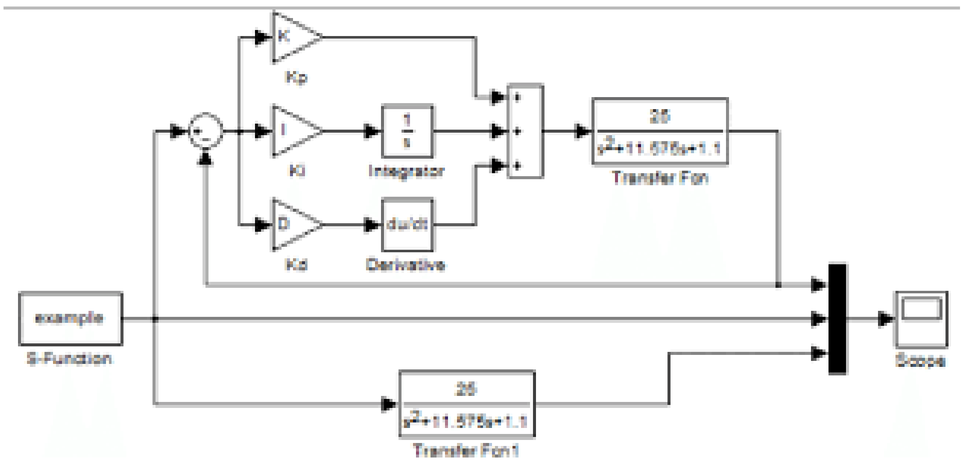
MATLAB Code:

```
% AC Voltage Analysis
Fs = 1000; % Sampling frequency
t = 0:1/Fs:1-1/Fs;
f = 50; % Frequency of AC voltage
V_peak = 10; % Peak voltage
V_ac = V_peak * sin(2*pi*f*t);
V_rms = rms(V_ac);
```

```
disp(['RMS Voltage: ', num2str(V_rms), ' V']);
```

7. DC Motor Simulation

Theory: Model and simulate the behavior of a DC motor.



MATLAB Code:

```
% DC Motor Simulation
J = 0.01; % Moment of inertia of the rotor
b = 0.1; % Damping ratio of the mechanical system
K = 0.01; % Electromotive force constant
R = 1; % Electric resistance
L = 0.5; % Electric inductance

A = [0 1 0; 0 -b/J K/J; 0 -K/L -R/L];
B = [0; 0; 1/L];
C = [1 0 0];
D = 0;
sys = ss(A,B,C,D);
step(sys);
title('DC Motor Step Response');
```

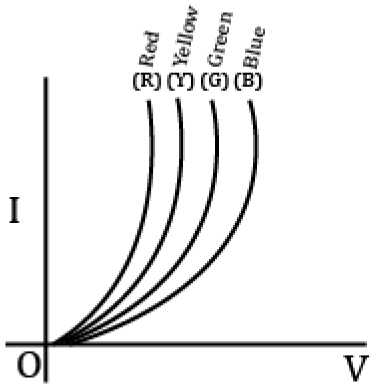
8. LED Characteristic Curve

Theory: Plot the I-V characteristic curve of an LED.

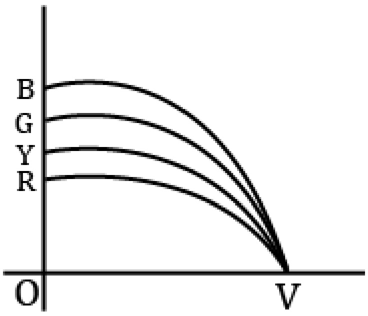
The characteristic curve of an LED is strongly non-linear. An LED is non-conductive if no external voltage is applied. The LED starts to conduct when the applied forward voltage UF is at least as high than the conducting voltage UD and the band gap is overcome by the electrons. The forward current is not proportional to the applied forward voltage. A small change in voltage can cause a large change in current. A small voltage change can lead to a strong change in light emission due to the proportionality of luminous flux and current intensity. This means that LEDs must generally be operated with a current limiter of some form or other, otherwise even slight fluctuations in the applied voltage can destroy the LED.

The I-V characteristic curve of an LED is

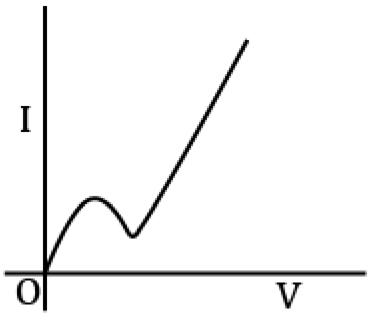
A



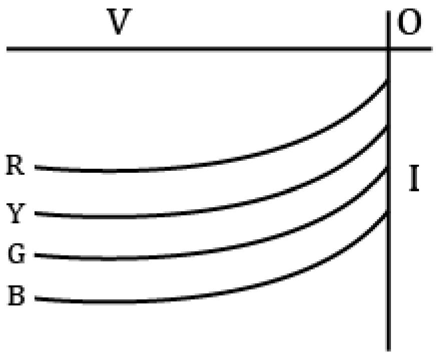
B



C



D

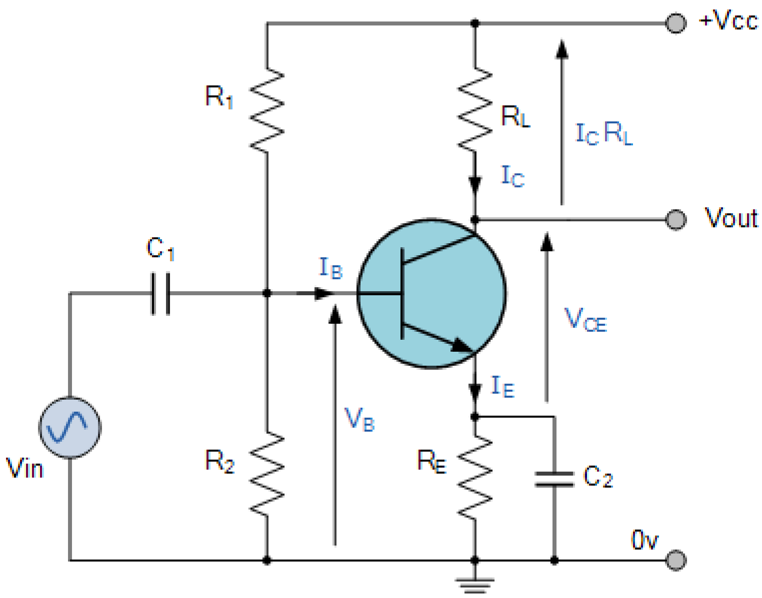


MATLAB Code:

```
% LED Characteristic Curve
V = 0:0.01:2; % Voltage range
I = exp(V); % Simplified exponential I-V relationship
plot(V, I);
title('LED I-V Characteristic Curve');
xlabel('Voltage (V)');
ylabel('Current (A)');
```

9. BJT Common Emitter Amplifier

Theory: Simulate the behavior of a BJT common emitter amplifier.

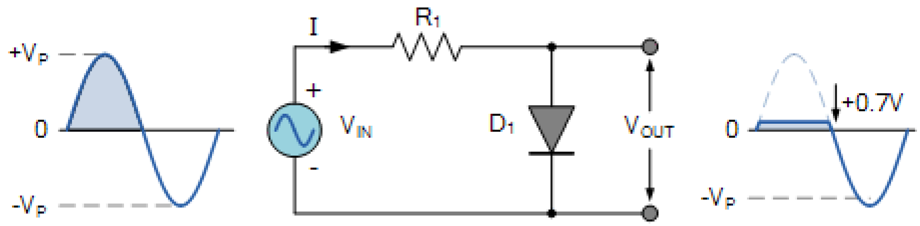


MATLAB Code:

```
% BJT Common Emitter Amplifier
beta = 100; % Current gain
R_C = 1e3; % Collector resistor in ohms
V_CC = 10; % Supply voltage in volts
I_B = 50e-6; % Base current in amperes
I_C = beta * I_B; % Collector current in amperes
V_CE = V_CC - I_C * R_C; % Collector-emitter voltage
disp(['Collector-Emitter Voltage: ', num2str(V_CE), ' V']);
```

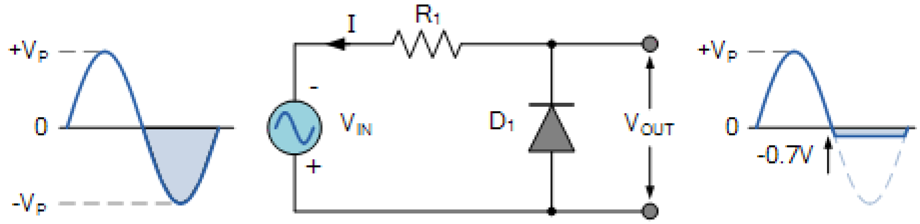
10. Diode Clipper Circuit

This diode clipping of the input signal produces an output waveform that resembles a flattened version of the input. For example, the half-wave rectifier is a clipper circuit, since all voltages below zero are eliminated.
But **Diode Clipping Circuits** can be used a variety of applications to modify an input waveform using signal and Schottky diodes or to provide over-voltage protection using zener diodes to ensure that the output voltage never exceeds a certain level protecting the circuit from high voltage spikes. Then diode clipping circuits can be used in voltage limiting applications.
We saw in the *Signal Diodes* tutorial that when a diode is forward biased it allows current to pass through itself clamping the voltage. When the diode is reverse biased, no current flows through it and the voltage across its terminals is unaffected, and this is the basic operation of the diode clipping circuit.
Although the input voltage to diode clipping circuits can have any waveform shape, we will assume here that the input voltage is sinusoidal. Consider the circuits below.
Positive Diode Clipping Circuits



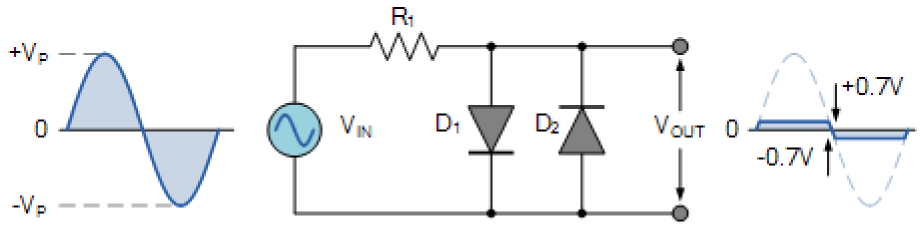
In this diode clipping circuit, the diode is forward biased (anode more positive than cathode) during the positive half cycle of the sinusoidal input waveform. For the diode to become forward biased, it must have the input voltage magnitude greater than +0.7 volts (0.3 volts for a germanium diode).
When this happens the diodes begins to conduct and holds the voltage across itself constant at 0.7V until the sinusoidal waveform falls below this value. Thus the output voltage which is taken across the diode can never exceed 0.7 volts during the positive half cycle.
During the negative half cycle, the diode is reverse biased (cathode more positive than anode) blocking current flow through itself and as a result has no effect on the negative half of the sinusoidal voltage which passes to the load unaltered. Thus the diode limits the positive half of the input waveform and is known as a positive clipper circuit.

Negative Diode Clipping Circuit



Here the reverse is true. The diode is forward biased during the negative half cycle of the sinusoidal waveform and limits or clips it to –0.7 volts while allowing the positive half cycle to pass unaltered when reverse biased. As the diode limits the negative half cycle of the input voltage it is therefore called a negative clipper circuit.

Clipping of Both Half Cycles



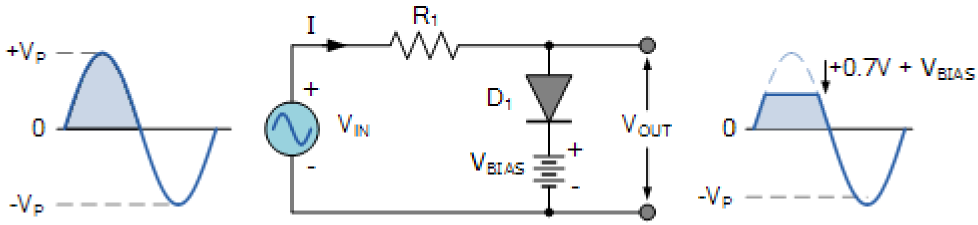
If we connected two diodes in inverse parallel as shown, then both the positive and negative half cycles would be clipped as diode D_1 clips the positive half cycle of the sinusoidal input waveform while diode D_2 clips the negative half cycle. Then diode clipping circuits can be used to clip the positive half cycle, the negative half cycle or both.

For ideal diodes the output waveform above would be zero. However, due to the forward bias voltage drop across the diodes the actual clipping point occurs at +0.7 volts and –0.7 volts respectively. But we can increase this ±0.7V threshold to any value we want up to the maximum value, (V_{PEAK}) of the sinusoidal waveform either by connecting together more diodes in series creating multiples of 0.7 volts, or by adding a voltage bias to the diodes.

Biased Diode Clipping Circuits

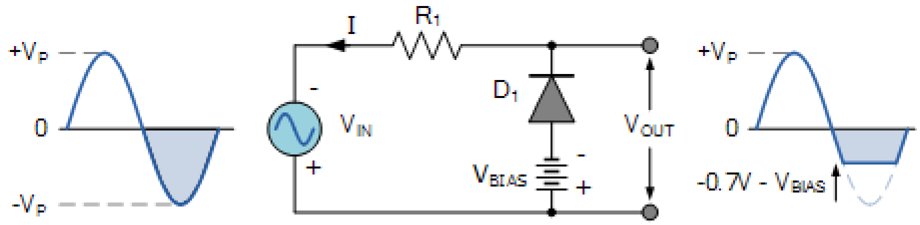
To produce diode clipping circuits for voltage waveforms at different levels, a bias voltage, V_{BIAS} is added in series with the diode to produce a combination clipper as shown. The voltage across the series combination must be greater than V_{BIAS} + 0.7V before the diode becomes sufficiently forward biased to conduct. For example, if the V_{BIAS} level is set at 4.0 volts, then the sinusoidal voltage at the diode's anode terminal must be greater than **4.0 + 0.7 = 4.7 volts** for it to become forward biased. Any anode voltage levels above this bias point are clipped off.

Positive Bias Diode



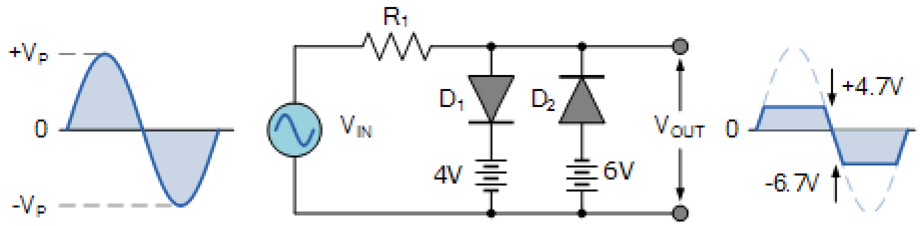
Likewise, by reversing the diode and the battery bias voltage, when a diode conducts the negative half cycle of the output waveform is held to a level $-V_{BIAS}$ – 0.7V as shown.

Negative Bias Diode



A variable diode clipping or diode limiting level can be achieved by varying the bias voltage of the diodes. If both the positive and the negative half cycles are to be clipped, then two biased clipping diodes are used. But for both positive and negative diode clipping, the bias voltage need not be the same. The positive bias voltage could be at one level, for example 4 volts, and the negative bias voltage at another, for example 6 volts as shown.

Diode Clipping of Different Bias levels



When the voltage of the positive half cycle reaches +4.7 V, diode D_1 conducts and limits the waveform at +4.7 V. Diode D_2 does not conduct until the voltage reaches –6.7 V. Therefore, all positive voltages above +4.7 V and negative voltages below –6.7 V are automatically clipped.

The advantage of biased diode clipping circuits is that it prevents the output signal from exceeding preset voltage limits for both half cycles of the input waveform, which could be an input from a noisy sensor or the positive and negative supply rails of a power supply.

If the diode clipping levels are set too low or the input waveform is too great then the elimination of both waveform peaks could end up with a square-wave shaped waveform.

Zener Diode Clipping Circuits

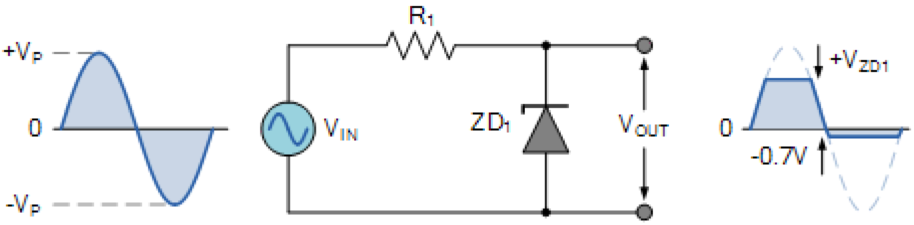
The use of a bias voltage means that the amount of the voltage waveform that is clipped off can be accurately controlled. But one of the main disadvantages of using voltage biased diode clipping circuits, is that they need an additional emf battery source which may or may not be a problem.

One easy way of creating biased diode clipping circuits without the need for an additional emf supply is to use Zener Diodes.

As we know, the zener diode is a another type of diode that has been specially manufactured to operate in its reverse biased breakdown region and as such can be used for voltage regulation or zener diode clipping applications. In the forward region, the zener acts just like an ordinary silicon diode with a forward voltage drop of 0.7V (700mV) when conducting, the same as above.

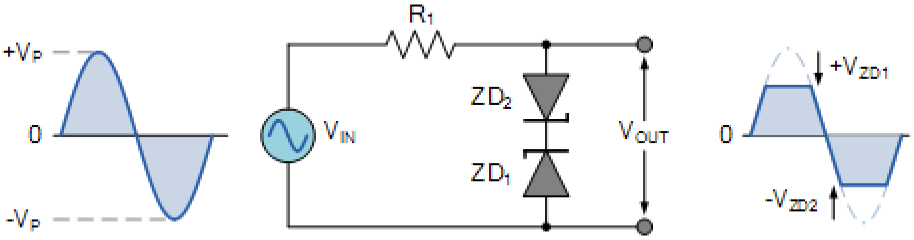
However, in the reverse bias region, the voltage is blocked until the zener diodes breakdown voltage is reached. At this point, the reverse current through the zener increases sharply but the zener voltage, V_Z across the device remains constant even if the zener current, I_Z varies. Then we can put this zener action to good effect by using them for clipping a waveform as shown.

Zener Diode Clipping



The zener diode is acting like a biased diode clipping circuit with the bias voltage being equal to the zener breakdown voltage. In this circuit during the positive half of the waveform the zener diode is reverse biased so the waveform is clipped at the zener voltage, V_{ZD1}. During the negative half cycle the zener acts like a normal diode with its usual 0.7V junction value.
We can develop this idea further by using the zener diodes reverse-voltage characteristics to clip both halves of a waveform using series connected back-to-back zener diodes as shown.

Full-wave Zener Diode Clipping



The output waveform from full wave zener diode clipping circuits resembles that of the previous voltage biased diode clipping circuit. The output waveform will be clipped at the zener voltage plus the 0.7V forward volt drop of the other diode. So for example, the positive half cycle will be clipped at the sum of zener diode, ZD_1 plus 0.7V from ZD_2 and vice versa for the negative half cycle.
Zener diodes are manufactured with a wide range of voltages and can be used to give different voltage references on each half cycle, the same as above. Zener diodes are available with zener breakdown voltages, V_Z ranging from 2.4 to 33 volts, with a typical tolerance of 1 or 5%. Note that once conducting in the reverse breakdown region, full current will flow through the zener diode so a suitable current limiting resistor, R_1 must be chosen.

Theory: Simulate a diode clipper circuit that limits the output voltage.
MATLAB Code:

```
% Diode Clipper Circuit
t = 0:0.001:1; % Time vector
Vi = 5*sin(2*pi*50*t); % Input sine wave
Vd = 0.7; % Diode forward voltage
Vo = Vi;
Vo(Vi > Vd) = Vd;
Vo(Vi < -Vd) = -Vd;
plot(t, Vi, t, Vo);
title('Diode Clipper Circuit');
xlabel('Time (s)');
ylabel('Voltage (V)');
legend('Input Voltage', 'Output Voltage');
```

11. Transmission Line Impedance

Theory:

Transmission Line Impedance

Transmission line impedance matching is a critical part of any layout. Whenever you are routing traces, there are several important points to check in order to ensure signal integrity throughout your board. Let's take a look at which transmission line impedance you need to consider for termination.

Everything You Need to Know About Transmission Line Impedance

Before getting into the topic of determining transmission line impedance, you should read this article, which shows the different impedances used to describe real transmission lines in a PCB. Just to summarize, we have some important values in a transmission line, some of which have simple formulas that can be used analytically:

- **Characteristic impedance:** This is the impedance of an isolated transmission line. In other words, this is the transmission line impedance when it is not coupled in any way to any other nearby transmission lines, such as in a differential pair.
- **Differential impedance:** This is the impedance of a pair of transmission lines. It is only equal to double the characteristic impedance in certain cases. In general, it is double the odd-mode impedance, which is the value we care about for differential signaling, as it is used in high-speed PCB design.

The two most common impedances that are used in PCB design are the characteristic impedance and the differential impedance. The formulas for these values in terms of basic circuit theory are shown below:

Single-ended impedance	**Odd-mode impedance**	**Differential impedance**
$Z_0 = \sqrt{\frac{L}{C}}$	$Z_d = 2\sqrt{\frac{L - L_m}{C + 2C_m}}$	$Z_{odd} = \sqrt{\frac{L - L_m}{C + 2C_m}}$

The equations shown above are simplified in that they do not include losses along the transmission line, which must be included in a real transmission line. The subscript "m" values are mutual values, meaning they are the mutual capacitance and mutual inductance. This is a parasitic effect and is unavoidable, even on the most carefully designed boards. This coupling produces the even and odd mode impedance values for a transmission line, depending on how both lines are driven. It will also affect which impedance value you use for termination (see the next section).

When designing an interconnect to have a specific transmission line impedance, we worry more about the characteristic impedance, but the odd-mode impedance is generally more important in high-speed layout and routing. Most design guides will only talk about the characteristic impedance. In reality, to properly route and terminate transmission lines, we have to understand the even and odd mode impedance, and the differential impedance when dealing with high-speed interfaces.

Termination and Impedance Matching

Here is something the PCB community won't tell you about transmission lines and their critical length—all interconnects that do not run at DC will behave as transmission lines. The issue is whether the effects of a transmission line impedance mismatch are noticeable at different frequencies or different signal rise times. This brings up the question of when it is worthwhile to match the transmission line impedance to a load. However, which transmission line impedance should be used?

As a result, designers have defined various values for a critical length below which you do not need impedance matching. As a general rule, if you are working with razor thin noise margins, then you should always match impedances between a driver, load, and source, even in electrically short transmission lines. With high-speed signals and interfaces, you should also always match transmission line impedance to the load. The correct way to do this is to look at the input impedance of the transmission line, not just the characteristic impedance.

This so-called critical length is actually quite important beyond just determining when to impedance match a transmission line to the source and load. Here is how this is quantified. If you calculate the voltage V and current I along a transmission line with length ℓ, you'll find that the impedance seen by a signal (analog or digital) that reflects off a mismatched load depends on the length of the transmission line and its capacitive and inductive characteristics. This is shown in the equation below.

$$Z_{\mathrm{in}}(\ell) = \frac{V(\ell)}{I(\ell)} = Z_0 \frac{1 + \Gamma_{\mathrm{L}} e^{-2\gamma\ell}}{1 - \Gamma_{\mathrm{L}} e^{-2\gamma\ell}}$$

$$Z_{\mathrm{in}}(\ell) = Z_0 \frac{Z_{\mathrm{L}} + Z_0 \tanh(\gamma\ell)}{Z_0 + Z_{\mathrm{L}} \tanh(\gamma\ell)}$$

γ = propagation constant

Z_{in} = impedance seen by a signal at the input

ℓ = transmission line length

Z_0 = characteristic impedance of the line

Z_{L} = load impedance

$$\Gamma_{\mathrm{L}} = \frac{Z_{\mathrm{L}} - Z_0}{Z_{\mathrm{L}} + Z_0}$$

The final equation defines the lossy transmission line input impedance seen by a signal that is input to the line

If the propagation constant is known, then the input impedance can be determined for any frequency. However, as we see above, the input impedance depends on the length of the line, not just the impedances.

Long or Short Lines

The author, being something of a mathematical purist, is a proponent of taking this approach in all situations. First, simply calculate the value of γ and the characteristic impedance. Next, plug in the length and γ into the equation shown above. The impedance value you calculate is the transmission line impedance the signal sees as it reflects off the mismatched load and travels on the line.

In the limit of a very long transmission line (such as when the line length is many multiples of the wavelength), then the tanh function eventually converges to 1. In this case, the input impedance is just the transmission line's characteristic impedance:

$$Z_{in} = Z_0 \text{ for long lines}$$

In contrast, when the transmission line is very small compared to the wavelength (i.e., at low enough frequency), the impedance seen by a traveling signal will reduce to the load impedance because tanh(0) = 0. Note that this applies to both lossy and lossless transmission lines:

$$Z_{in} = Z_L \text{ for short lines}$$

This explains why we have a critical length: when the transmission line is short enough that $\tanh(\gamma\ell) \sim 0$ (or $\tan(\gamma\ell) \sim 0$ for a lossless line), then the input signal only sees the load impedance. The source and load impedance should be matched to ensure maximum power transfer into the load and prevent signal reflection.

Special Cases for a Lossless Transmission Line

For transmission lines with sufficiently low losses (i.e., $\mathrm{Re}(\gamma) = 0$), the tanh(x) function above must be replaced with the function jtan(x), where j is the imaginary constant. You will have certain cases where $\mathrm{Im}(\gamma)\ell = m\pi/2$, where m is an integer. In this case, you will be evaluating $\tan(m\pi/2)$ in the above equation. The result reduces to:

Even m:

$$Z_{in}(l) = Z_L$$

Odd m:

$$Z_{in}(l) = \frac{Z_0^2}{Z_L}$$

These are the impedances that a signal sees as it reflects back along the transmission line. If the source, load, and transmission line are all mismatched, then there are repeated reflections along the length of the line, which leads to a stair-step response seen in digital signals or standing waves with analog signals.

Matching the transmission line's characteristic impedance and the load prevents reflection at the load end, and the input impedance will just be the characteristic impedance. In this case, there are no reflections at the load, but you do not have maximum power transfer down the line if the source is unmatched. If you go a step further and match the source to the characteristic impedance, you now ensure maximum power transfer across the line.

Analog vs. Digital Transmission Lines

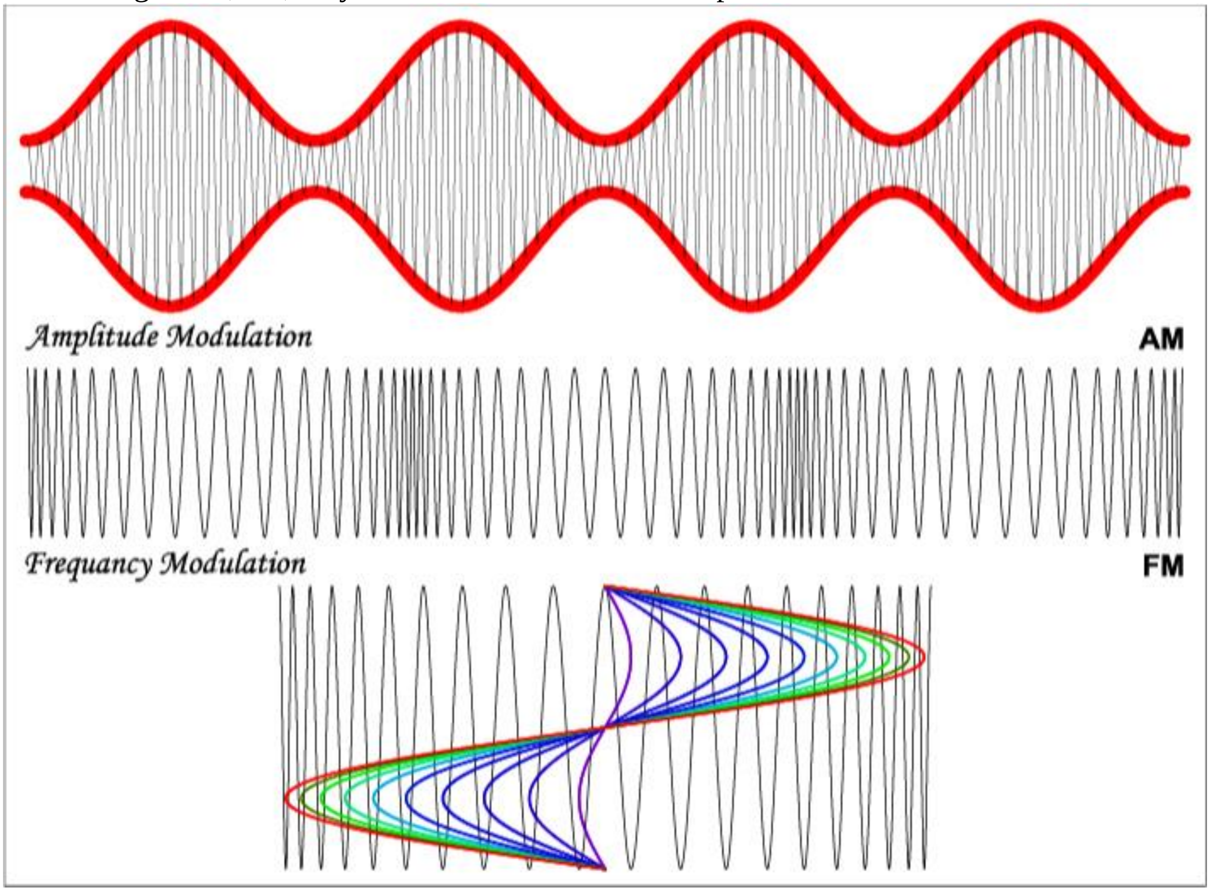
We should make a distinction here regarding the value of γ. In distinguishing the effects of transmission for different types of signals, think of transmission lines (either isolated or coupled) as filters with some transfer function. With an analog signal oscillating at a single frequency, γ is just the complex wavenumber multiplied by the effective dielectric constant (pi divided by half the wavelength inside the trace) plus the attenuation per unit length along the line. However, you might remember from discussions of dispersion, that the dielectric constant and the characteristic impedance depend on frequency.

This affects signals in different ways. For analog and digital transmission lines, we need to analyze transmission lines in the following ways:

- **Analog signals:** Generally, we only care about one specific frequency. For modulated signals, we typically only design to the carrier frequency.
- **Digital signals:** With digital signals, we have to design so that the characteristic impedance (or differential impedance) and termination are determined at a very high

limiting frequency. The limited frequency of interest is usually the Nyquist frequency for the receiver or some limit determined from the rise time.

For frequency-modulated analog signals, the characteristic impedance of a transmission line has a constant value throughout the signal's frequency spectrum as long as the relevant frequency range is high enough. At lower frequencies and with amplitude modulated signals, this may not be the case, and the other relevant impedance values will depend on frequency and driving mode, i.e., they will have some associated spectrum.



Transmission line impedance with modulated signals only concerns the carrier frequency

With digital signals, one must remember that the source and load impedances are not consistent at all frequencies. The relevant bandwidth to consider for transmission line impedance matching is the range extending from the pulse repetition rate up to some very high frequency. For high-speed receivers, this is usually the Nyquist frequency. If you're worried about oversampling the digital signal or measuring it, then you need to use at least 0.5/(rise time). As long as the component's bandwidth is flat within this range of frequencies, then you can consider a single value for your PCB routing design rules and transmission line impedance matching.

MATLAB Code:

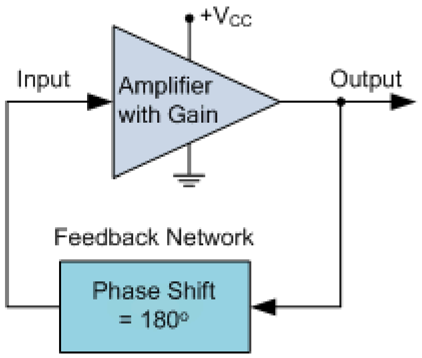
```
% Transmission Line Impedance
L = 0.5e-6; % Inductance per unit length (H/m)
C = 2e-12; % Capacitance per unit length (F/m)
```

```
Z0 = sqrt(L/C); % Characteristic impedance
disp(['Characteristic Impedance: ', num2str(Z0), ' Ohms']);
```

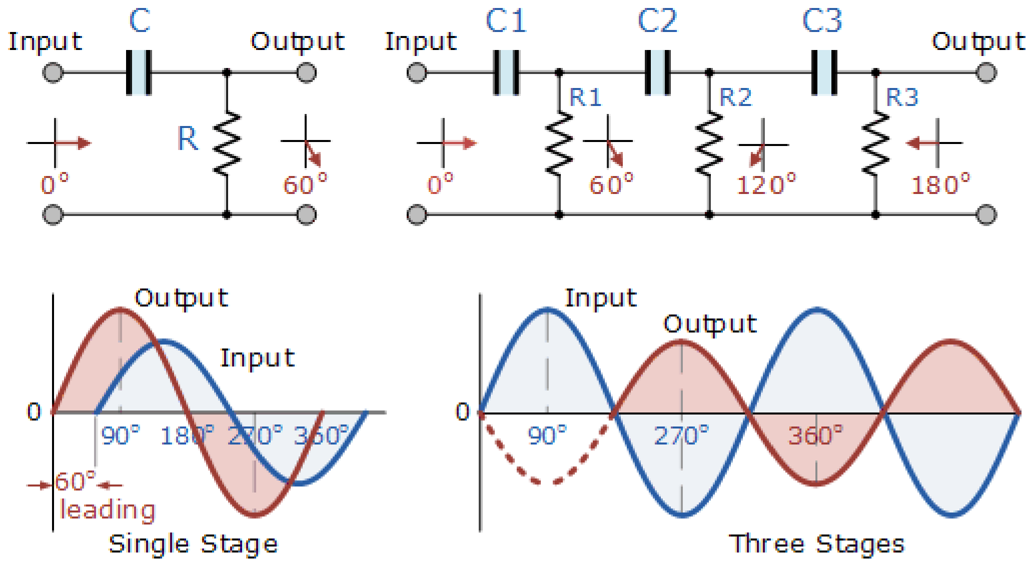
12. Phase Shift in RC Circuit

Theory: Calculate and plot the phase shift introduced by an RC circuit.

A single stage transistor amplifier can produce 180^o of phase shift between its output and input signals when connected as a common-emitter type amplifier and we can use this configuration to produce an RC oscillator circuit.
But we can configure transistor stages to operate as oscillators by placing resistor-capacitor (RC) networks around the transistor to provide the required regenerative feedback without the need for a tank circuit. Frequency selective RC coupled amplifier circuits are easy to build and can be made to oscillate at any desired frequency by selecting the appropriate values of resistance and capacitance.
For an RC oscillator to sustain its oscillations indefinitely, sufficient feedback of the correct phase, that is positive (in-phase) Feedback must be provided along with the voltage gain of the single transistor amplifier being used to inject adequate loop gain into the closed-loop circuit in order to maintain oscillations allowing it to oscillates continuously at the selected frequency.



In an **RC Oscillator** circuit the input is shifted 180^o through the feedback circuit returning the signal out-of-phase and 180^o again through an inverting amplifier stage to produces the required positive feedback. This then gives us "180^o + 180^o = 360^o" of phase shift which is effectively the same as 0^o, thereby giving us the required positive feedback. In other words, the total phase shift of the feedback loop should be "0" or any multiple of 360^o to obtain the same effect.
In a **Resistance-Capacitance Oscillator** or simply known as an **RC Oscillator**, we can make use of the fact that a phase shift occurs between the input to a RC network and the output from the same network by using interconnected RC elements in the feedback branch, for example.
RC Phase-Shift Network



The circuit on the left shows a single resistor-capacitor network whose output voltage "leads" the input voltage by some angle less than 90°. In a pure or ideal single-pole RC network. it would produce a maximum phase shift of exactly 90°, and because 180° of phase shift is required for oscillation, at least two single-poles networks must be used within an *RC oscillator* design.

However in reality it is difficult to obtain exactly 90° of phase shift for each RC stage so we must therefore use more RC stages cascaded together to obtain the required value at the oscillation frequency. The amount of actual phase shift in the circuit depends upon the values of the resistor (R) and the capacitor (C), at the chosen frequency of oscillations with the phase angle (φ) being given as:

RC Phase Angle

$$X_C = \frac{1}{2\pi f C} \qquad R = R,$$

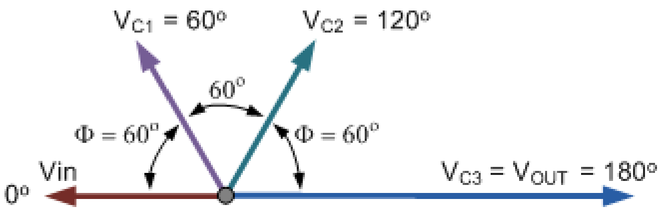
$$Z = \sqrt{R^2 + (X_C)^2}$$

$$\therefore \quad \phi = \tan^{-1}\frac{X_C}{R}$$

Where: X_C is the Capacitive Reactance of the capacitor, R is the Resistance of the resistor, and f is the Frequency.

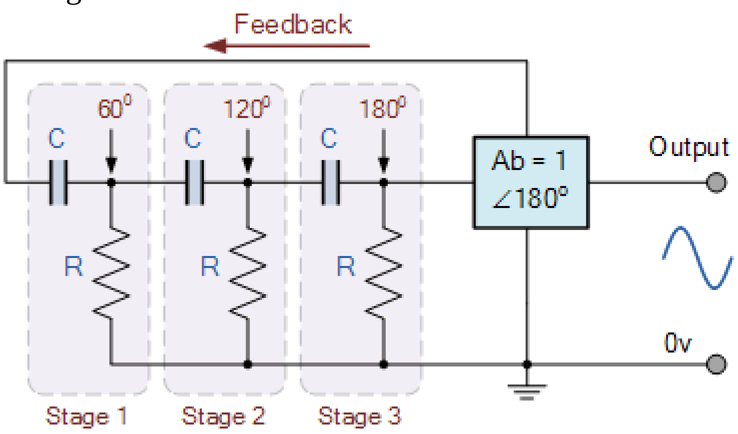
In our simple example above, the values of R and C have been chosen so that at the required frequency the output voltage leads the input voltage by an angle of about 60°. Then the phase angle between each successive RC section increases by another 60° giving a phase difference between the input and output of 180° (3 x 60°) as shown by the following vector diagram.

Vector Diagram



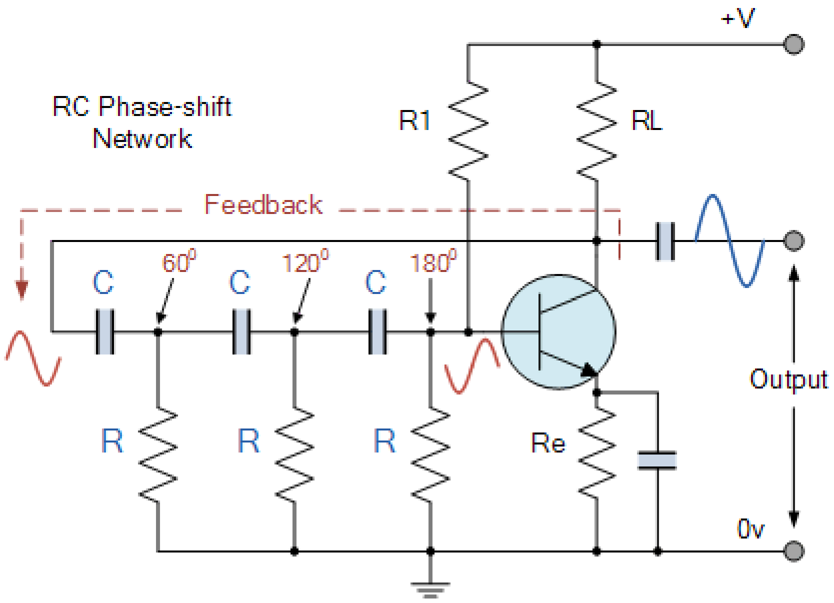
So by cascading together three such RC networks in series we can produce a total phase shift in the circuit of 180° at the chosen frequency and this forms the bases of a "RC Oscillator" otherwise known as a **Phase Shift Oscillator** as the phase angle is shifted by an amount through each stage of the circuit. Then the phase shift occurs in the phase difference between the individual RC stages. Conveniently op-amp circuits are available in quad IC packages. For example, the LM124, or the LM324, etc. so four RC stages could also be used to produce the required 180° of phase shift at the required oscillation frequency.

We know that in an amplifier circuit either using a Bipolar Transistor or an Inverting Operational Amplifier configuration, it will produce a phase-shift of 180° between its input and output. If a three-stage RC phase-shift network is connected as a feedback network between the output and input of an amplifier circuit, then the total phase shift created to produce the required regenerative feedback is: 3 x 60° + 180° = 360° = 0° as shown.



The three RC stages are cascaded together to obtain the required slope for a stable oscillation frequency. The feedback loop phase shift is -180° when the phase shift of each stage is -60°. This occurs when jω = 2pi$f$ = 1/1.732RC as (tan 60° = 1.732). Then to achieve the required phase shift in an RC oscillator circuit is to use multiple RC phase-shifting networks such as the circuit below.

Basic RC Oscillator Circuit



The basic **RC Oscillator** which is also known as a **Phase-shift Oscillator**, produces a sine wave output signal using regenerative feedback obtained from the resistor-capacitor (RC) ladder network. This regenerative feedback from the RC network is due to the ability of the capacitor to store an electric charge, (similar to the LC tank circuit).

This resistor-capacitor feedback network can be connected as shown above to produce a leading phase shift (phase advance network) or interchanged to produce a lagging phase shift (phase retard network) the outcome is still the same as the sine wave oscillations only occur at the frequency at which the overall phase-shift is 360^o.

By varying one or more of the resistors or capacitors in the phase-shift network, the frequency can be varied and generally this is done by keeping the resistors the same and using a 3-ganged variable capacitor because capacitive reactance (X_C) changes with a change in frequency as capacitors are frequency-sensitive components. However, it may be required to re-adjust the voltage gain of the amplifier for the new frequency.

If the three resistors, R are equal in value, that is $R_1 = R_2 = R_3$, and the capacitors, C in the phase shift network are also equal in value, $C_1 = C_2 = C_3$, then the frequency of oscillations produced by the RC oscillator is simply given as:

$$f_r = \frac{1}{2\pi RC\sqrt{2N}}$$

- Where:
- f_r is the oscillators output frequency in Hertz
- R is the feedback resistance in Ohms
- C is the feedback capacitance in Farads
- N is the number of RC feedback stages.

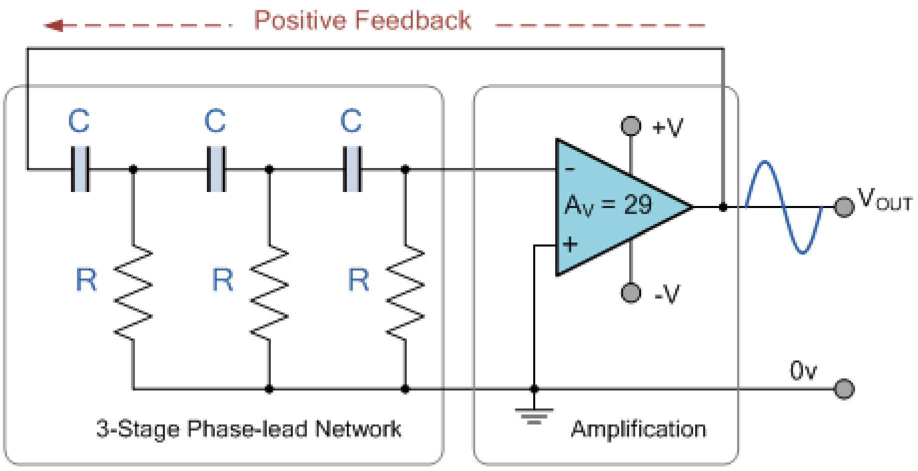
This is the frequency at which the phase shift circuit oscillates. In our simple example above, the number of stages is given as three, so N = 3 ($\sqrt{2*3} = \sqrt{6}$). For a four stage RC network, N = 4 ($\sqrt{2*4} = \sqrt{8}$), etc.

Since the resistor-capacitor combination in the **RC Oscillator** ladder network also acts as an attenuator, that is the signal reduces by some amount as it passes through each passive stage. It could be assumed that the three phase shift sections are independent of each other but this is not the case as the total accumulative feedback attenuation becomes -1/29th (Vo/Vi = β = -1/29) across all three stages.
Thus the voltage gain of the amplifier must be sufficiently high enough to overcome these passive RC losses. Clearly then in order to produce a total loop gain of -1, in our three stage RC network above, the amplifier gain must be equal too, or greater than, 29 to compensate for the attenuation of the RC network.
The loading effect of the amplifier on the feedback network has an effect on the frequency of oscillations and can cause the oscillator frequency to be up to 25% higher than calculated. Then the feedback network should be driven from a high impedance output source and fed into a low impedance load such as a common emitter transistor amplifier but better still is to use an Operational Amplifier as it satisfies these conditions perfectly.

The Op-amp RC Oscillator

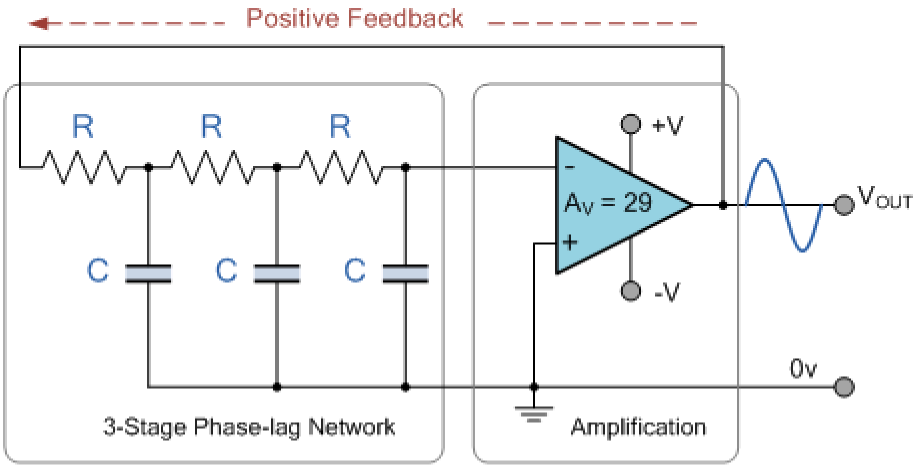
When used as RC oscillators, **Operational Amplifier RC Oscillators** are more common than their bipolar transistors counterparts. The oscillator circuit consists of a negative-gain operational amplifier and a three section RC network that produces the 180° phase shift. The phase shift network is connected from the op-amps output back to its "inverting" input as shown below.

Op-amp Phase-lead RC Oscillator Circuit



As the feedback is connected to the inverting input, the operational amplifier is therefore connected in its "inverting amplifier" configuration which produces the required 180° phase shift while the RC network produces the other 180° phase shift at the required frequency (180° + 180°). This type of feedback connection with the capacitors in series and the resistors connected to ground (0V) potential is known as a *phase-lead* configuration. In other words, the output voltage leads the input voltage producing a positive phase angle.
But we can also create a *phase-lag* configuration by simply changing the positions of the RC components so that the resistors are connected in series and the capacitors are connected to ground (0V) potential as shown. This means that the output voltage lags the input voltage producing a negative phase angle.

Op-amp Phase-lag RC Oscillator Circuit



However, due to the reversal of the feedback components, the orginal equation for the frequency output of the phase-lead RC oscillator is modified to:

$$f_r = \frac{\sqrt{2N}}{2\pi RC}$$

Although it is possible to cascade together only two single-pole RC stages to provide the required 180° of phase shift (90° + 90°), the stability of the oscillator at low frequencies is generally poor.

One of the most important features of an **RC Oscillator** is its frequency stability which is its ability to provide a constant frequency sine wave output under varying load conditions. By cascading three or even four RC stages together (4 x 45°), the stability of the oscillator can be greatly improved.

RC Oscillators with four stages are generally used because commonly available operational amplifiers come in quad IC packages so designing a 4-stage oscillator with 45° of phase shift relative to each other is relatively easy.

RC Oscillators are stable and provide a well-shaped sine wave output with the frequency being proportional to 1/RC and therefore, a wider frequency range is possible when using a variable capacitor. However, RC Oscillators are restricted to frequency applications because of their bandwidth limitations to produce the desired phase shift at high frequencies.

RC Oscillator Example No1

An operational amplifier based *3-stage RC Phase Shift Oscillator* is required to produce a sinusoidal output frequency of 4kHz. If 2.4nF capacitors are used in the feedback circuit, calculate the value of the frequency determining resistors and the value of the feedback resistor required to sustain oscillations. Also draw the circuit.

The standard equation given for the phase shift RC Oscillator is:

$$f_r = \frac{1}{2\pi RC\sqrt{2N}}$$

The circuit is to be a 3-stage RC oscillator which will therefore consist of equal resistors and three equal 2.4nF capacitors. As the frequency of oscillation is given as 4.0kHz, the value of the resistors are calculated as:

$$f = 4.0\text{kHz} = \frac{1}{2\pi\sqrt{(2\times 3)}\times R\times 2.4\text{nF}}$$

$$R = \frac{1}{2\pi\sqrt{(6)}\times 4000\times 2.4\times 10^{-9}}$$

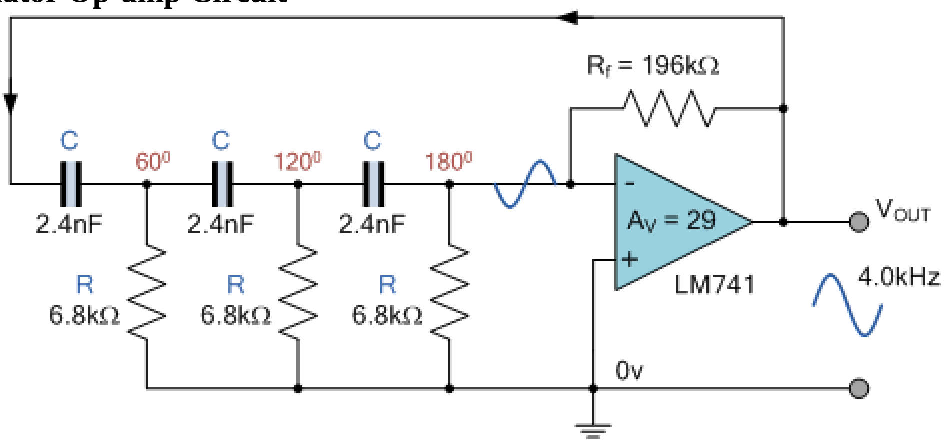
$$\therefore R = 6770\Omega \text{ or } 6.8\text{k}\Omega$$

The operational amplifiers gain must be equal to 29 in order to sustain oscillations. The resistive value of the oscillation resistors are 6.8kΩ, therefore the value of the op-amps feedback resistor R_f is calculated as:

$$A_V = \frac{R_F}{R} = 29$$

$$\therefore R_F = A_V \times R = 29 \times 6.8\text{k}\Omega = 197.2\text{k}\Omega$$

RC Oscillator Op-amp Circuit



In the next tutorial about Oscillators, we will look at another type of **RC Oscillator** called a Wien Bridge Oscillators which uses resistors and capacitors as its tank circuit to produce a low frequency sinusoidal waveform.

MATLAB Code:

```
% Phase Shift in RC Circuit
R = 1e3; % Resistance in ohms
C = 1e-6; % Capacitance in farads
f = 1000; % Frequency in Hz
omega = 2*pi*f; % Angular frequency
phi = atan(1/(omega*R*C)); % Phase shift
disp(['Phase Shift: ', num2str(rad2deg(phi)), ' degrees']);
```

13. Frequency Modulation (FM)

Frequency modulation, commonly referred to as FM, is a common term that we hear in our daily lives. Today, Frequency modulation is used widely in radio communication and broadcasting. But have we wondered what it actually is or what is the technology and mechanism behind FM? We will understand what frequency modulation is in this lesson and also learn its mechanism as well as its applications.

What Is Frequency Modulation?

Frequency modulation is a technique or a process of encoding information on a particular signal (analogue or digital) by varying the carrier wave frequency in accordance with the frequency of the modulating signal. As we know, a modulating signal is nothing but information or message that has to be transmitted after being converted into an electronic signal.

Much like amplitude modulation, frequency modulation also has a similar approach, where a carrier signal is modulated by the input signal. However, in the case of FM, the amplitude of the modulated signal is kept, or it remains constant.

The frequency modulation index is mostly over 1, and it usually requires a high bandwidth at a range of 200 kHz. FM operates in a very high-frequency range, normally between 88 to 108 Megahertz. There are complex circuits with an infinite number of sidebands that help in receiving high-quality signals with high sound quality.

Broadcast stations in the VHF portion of the frequency spectrum between 88.5 and 108 MHz often use large values of deviation (±75 kHz). This is known as wide-band FM (WBFM). Even though these signals support high-quality transmissions, they do occupy a large amount of bandwidth. Normally, 200 kHz is allowed for each wide-band FM transmission. On the other hand, communications use very little bandwidth. Whereas narrowband FM (NBFM) often uses deviation figures of around ±3 kHz. Besides, narrow-band FM is mostly used for two-way radio communication applications.

Applications of Frequency Modulation

If we talk about the applications of frequency modulation, it is mostly used in radio broadcasting. It offers a great advantage in radio transmission as it has a larger signal-to-noise ratio, which means that it results in low radio frequency interference. This is the main reason that many radio stations use FM to broadcast music over the radio.

Additionally, some of its uses are also found in radar, telemetry, seismic prospecting, and in EEG, different radio systems, music synthesis as well as in video-transmission instruments.

In radio transmission, frequency modulation has a good advantage over other modulation. It has a larger signal-to-noise ratio, meaning it will reject radio frequency interferences much better than an equal power amplitude modulation (AM) signal. Due to this major reason, most music is broadcasted over FM radio.

FM Modulators

There are several methods that can be used to generate either direct or indirect frequency-modulated signals.

- **A voltage-controlled oscillator or varactor diode oscillator:** A voltage-controlled oscillator can be used to form direct FM modulation by directly feeding the message into the input of the oscillator. In the case of the varactor diode, we place this device within the tuned circuit of an oscillator circuit.
- **Crystal oscillator circuit:** A varactor diode can also be used within a crystal oscillator circuit wherein the signal needs to be multiplied in frequency, and only narrowband FM is attained.
- **Phase-locked loop:** This is an excellent method to generate frequency modulation signals. However, the constraints within the loop should be checked carefully, and once everything is stable, it offers an excellent solution.

Frequency Modulation Equations

Frequency modulation equations mainly consist of a sinusoidal expression with the integral of the baseband signal that can be either a sine or cosine function.

It can be represented mathematically as;

$m(t) = A_m \cos(\omega_m t + \Theta)$ ……………… 1

$m(t) \rightarrow$ modulating signal

Where,

$A_m \rightarrow$ Amplitude of the modulating signal.

$\omega_m \rightarrow$ Angular frequency of the modulating signal.

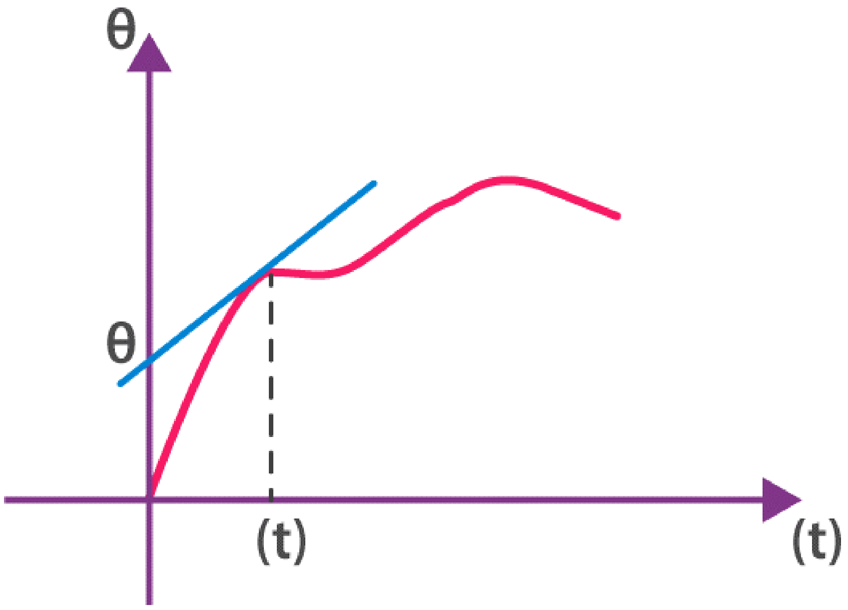
$\Theta \rightarrow$ It is the phase of the modulating signal.

Same as amplitude modulation, when we try to modulate an input signal (information), we need a carrier wave, and we will experience

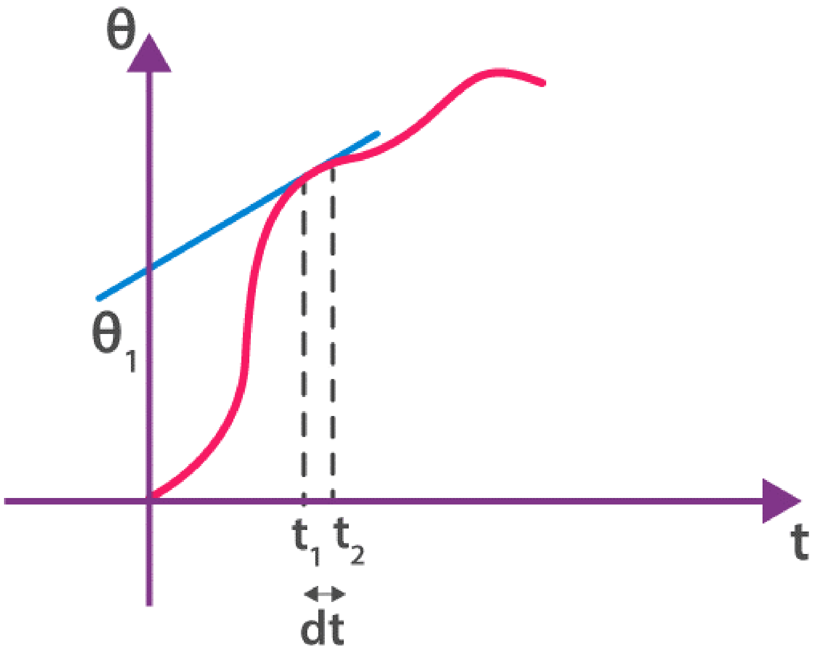
$C(t) = A_c \cos(\omega_c t + \Theta)$ ………….. 2

Angular modulation, which means ω_c (or) Θ of the carrier wave, starts varying linearly with respect to the modulating signal, like amplitude modulation.

The modulating signal at any instant of time.



$C(t) = A_c \cos(\omega_c t + \Theta)$
For any particular instant $(\omega_c t + \Theta)$ is not varying with respect to time the Θ becomes so it Θ_0 = constant, then
ω_c = also constant
If we draw a tangent for the given signal at any instant of time, the slope of the tangent gives ω_c, and the tangent, when it cuts the Θ-axis, gives Θ_0 value.
Case II: For a Small Interval of Time
Now, let us consider a small interval of time $\Delta t = t_2 - t_1$
Time interval, $t_1 < t < t_2$, we will look this into two particular instant of t_1 and t_2



Let us consider an instant at (t_1); if we draw a tangent to a given signal at (t_1), the slope of the curve instantaneous frequency (w_i) at that particular instant.
Similarly, the intercept of the tangent with Θ – axis gives the instantaneous phase (Θ_i). Likewise, we can get w_i for any instant of the given curve. From this, one thing is clear, Θ – is the phase at the instant
We write this as a function of time (t), and instantaneous frequency is
Equation 3 and 4 gives the fundamental understanding of phase and frequency. If we try to modulate this signal, let us see what's happening at any instant of time the signal phase is [this is known as phase modulation]
Where,
k – constant
$\omega_c \rightarrow$ frequency of the carrier wave
$m(t) \rightarrow$ modulating signal
The insert signal in this phase becomes,
$A \cos \Theta (t) = A_c \cos [\omega_c t + k\, m(t)]$
If we need the frequency of the wave,
$\omega_c + km(t)$ ……….6 [this is known as frequency modulation]
Where,
As we know, in the idea of frequency modulation, the frequency of the carrier wave must vary linearly with respect to a particular signal, as we can see in Equation 5. From this, we get,
$\omega(t) = \omega_c + km(t)$
If we do phase modulation, it is nothing but frequency modulation. When we do frequency modulation, we are differentiating the particular modulating signal, which then automatically depicts it as phase modulation.

Expression for Frequency Modulated Wave
As we know, from amplitude modulation, we need two sine (or) cosine waves for modulation.
$m(t) = A_m \cos (\omega_m t)$ and
$c(t) = A_c \cos (\omega_c t)$
or
$m(t) = A_m \cos (2\pi f_m t)$
$c(t) = A_c \cos (2\pi f_c t)$
Then frequency modulated wave will be:
$f_m(t) = f_c + k A_m \cos (2\pi f_m t)$
$f_m (t) = f_c + k\, m(t)$
Where,
fm(t) = is frequency modulated wave
$f_c \rightarrow$ frequency of the carrier wave
$m(t) \rightarrow$ modulating signal
$k \rightarrow$ proportionality constant

Frequencies in Frequency Modulation
In FM, variation (or) deviation in frequency, the maximum deviation is Δf_{max}
$\Delta f_{max} = | f_m(t) - f_c |$

$= | KA_m \cos(2\pi f_m t) |$

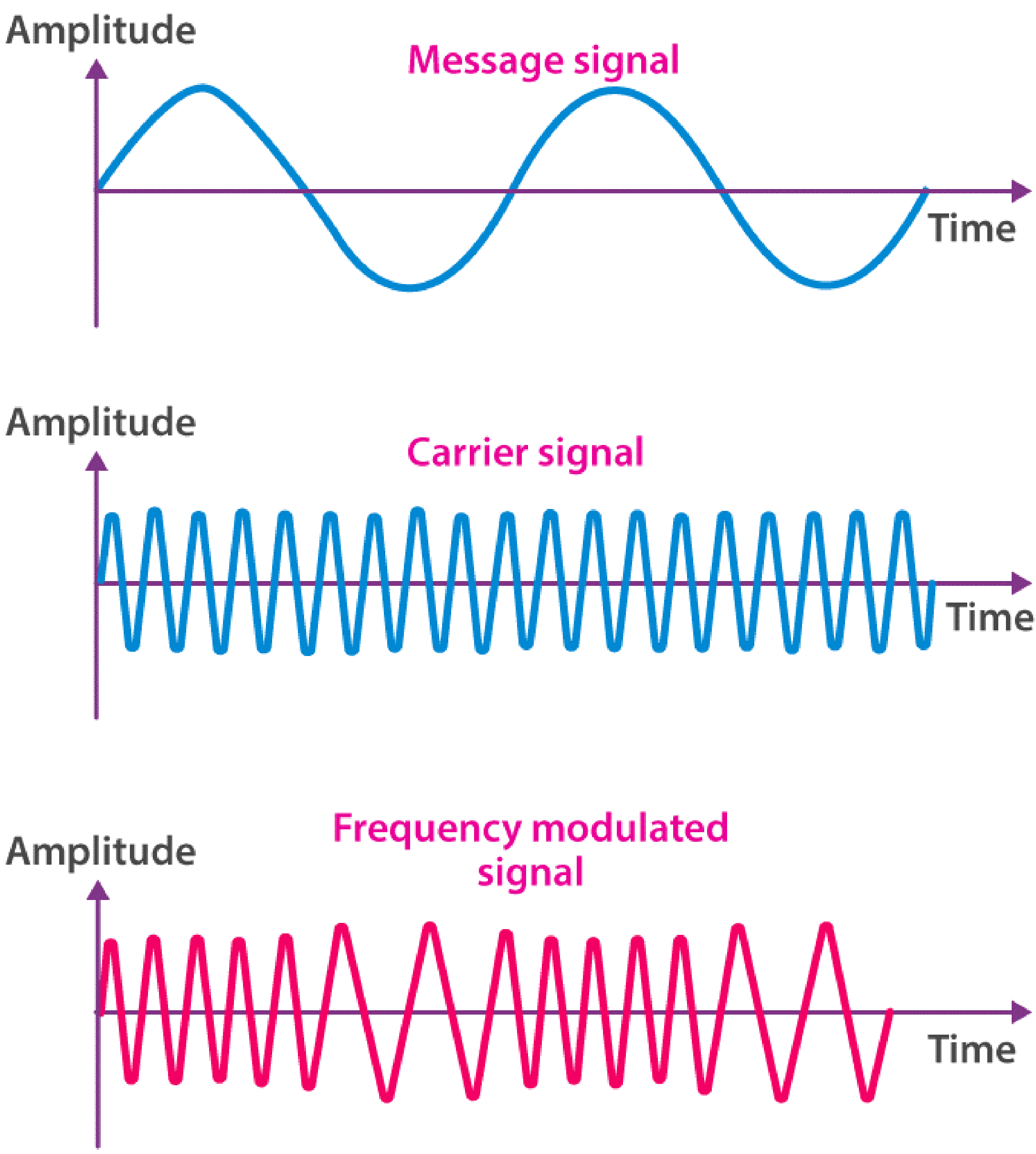
The maximum deviation in frequency is $K A_m$

Generally, frequency deviation is defined as the measure of the change in a carrier frequency produced by the amplitude of the input-modulating signal.

Modulation Index (μ)

The modulation index is the ratio of maximum deviation in frequency of the modulating signal.

Graphical Representation of Frequency Modulated Wave



If we observe the graph, we can notice that the frequency of a carrier increases when the amplitude of the input signal is increased. Here, the carrier frequency is maximum when the input signal is at its highest. Also, the frequency of a carrier decreases if the amplitude of the modulating signal goes down. What it means is that the carrier frequency is minimum when the input signal is at its lowest.

Frequency Demodulation

When there is modulation, usually, we need to successfully demodulate it and, at the same time, recover the original signal. In such cases, an FM demodulator, also known as an FM discriminator or FM detector, is used. While there are several types of FM demodulators, the main functionality of these devices is to convert the frequency variations of the input signal into amplitude variations of the output signal. The demodulators are used along with an audio amplifier or possibly a digital interface.

Advantages and Disadvantages of Frequency Modulation

Advantages	**Disadvantages**
Less interference and noise.	Equipment cost is higher. Has a large bandwidth.
Power consumption is less as compared to AM.	More complicated receiver and transmitter
Adjacent FM channels are separated by guard bands.	The antennas for FM systems should be kept close for better communication.

Amplitude Modulation vs Frequency Modulation

While FM and AM function in a similar manner, the main difference lies in how their carrier waves are modulated. In the case of AM, the signal strength varies in order to incorporate sound information, whereas for FM, the frequency at which the current changes direction per second for the carrier signal is varied. We will look at some of the differences between FM and AM below.

Amplitude Modulation (AM)	**Frequency Modulation (FM)**
Frequency and phase remain the same	Amplitude and phase remain the same
It can be transmitted over a long distance but has poor sound quality	Better sound quality with higher bandwidth.
The frequency range varies between 535 and 1705 kHz	For FM, it is from 88 to 108 MHz, mainly in the higher spectrum
Signal distortion can occur in AM	Less instances of signal distortion
It consists of two sidebands	An infinite number of sidebands
Circuit design is simple and less expensive	Circuit design is intricate and more expensive
Easily susceptible to noise	Less susceptible to noise

Theory: Simulate frequency modulation of a carrier signal.
MATLAB Code:

```
% Frequency Modulation
```

```
Fs = 10000; % Sampling frequency
t = 0:1/Fs:1-1/Fs;
Fc = 100; % Carrier frequency
Fm = 5; % Modulating frequency
Am = 1; % Modulating amplitude
beta = 5; % Modulation index
carrier = cos(2*pi*Fc*t);
modulator = cos(2*pi*Fm*t);
fm_signal = cos(2*pi*Fc*t + beta*sin(2*pi*Fm*t));
plot(t, fm_signal);
title('Frequency Modulated Signal');
xlabel('Time (s)');
ylabel('Amplitude');
```

14. AM Modulation

Theory: Simulate amplitude modulation of a carrier signal.
.
Amplitude modulation is a process by which the wave signal is transmitted by modulating the amplitude of the signal. It is often called AM and is commonly used in transmitting a piece of information through a radio carrier wave. Amplitude modulation is mostly used in the form of electronic communication.
Download Complete Chapter Notes of Communication System
Currently, this technique is used in many areas of communication, such as in portable two-way radios, citizens band radios, VHF aircraft radios and in modems for computers. Amplitude modulation is also used to refer to mediumwave AM radio broadcasting.

What Is Amplitude Modulation?
Amplitude modulation, or just AM, is one of the earliest modulation methods that is used in transmitting information over the radio. This technique was devised in the 20th century at a time when Landell de Moura and Reginald Fessenden were conducting experiments using a radiotelephone in the 1900s. After successful attempts, the modulation technique was established and used in electronic communication.
In general, amplitude modulation definition is given as a type of modulation where the amplitude of the carrier wave varies in some proportion with respect to the modulating data or the signal.
As for the mechanism, when amplitude modulation is used, there is a variation in the amplitude of the carrier. Here, the voltage or the power level of the information signal changes the amplitude of the carrier. In AM, the carrier does not vary in amplitude. However, the modulating data is in the form of signal components consisting of frequencies either higher or lower than that of the carrier. The signal components are known as sidebands, and the sideband power is responsible for the variations in the overall amplitude of the signal.
The AM technique is totally different from frequency modulation and phase modulation, where the frequency of the carrier signal is varied in the first case and in the second one, the phase is varied.

Types of Amplitude Modulation

There are three main types of amplitude modulation. They are

- Double Sideband-suppressed Carrier Modulation (DSB-SC)
- Single Sideband Modulation (SSB)
- Vestigial Sideband Modulation (VSB)

Designations by ITU

The International Telecommunication Union (ITU) designated different types of amplitude modulation in 1982. They are as follows:

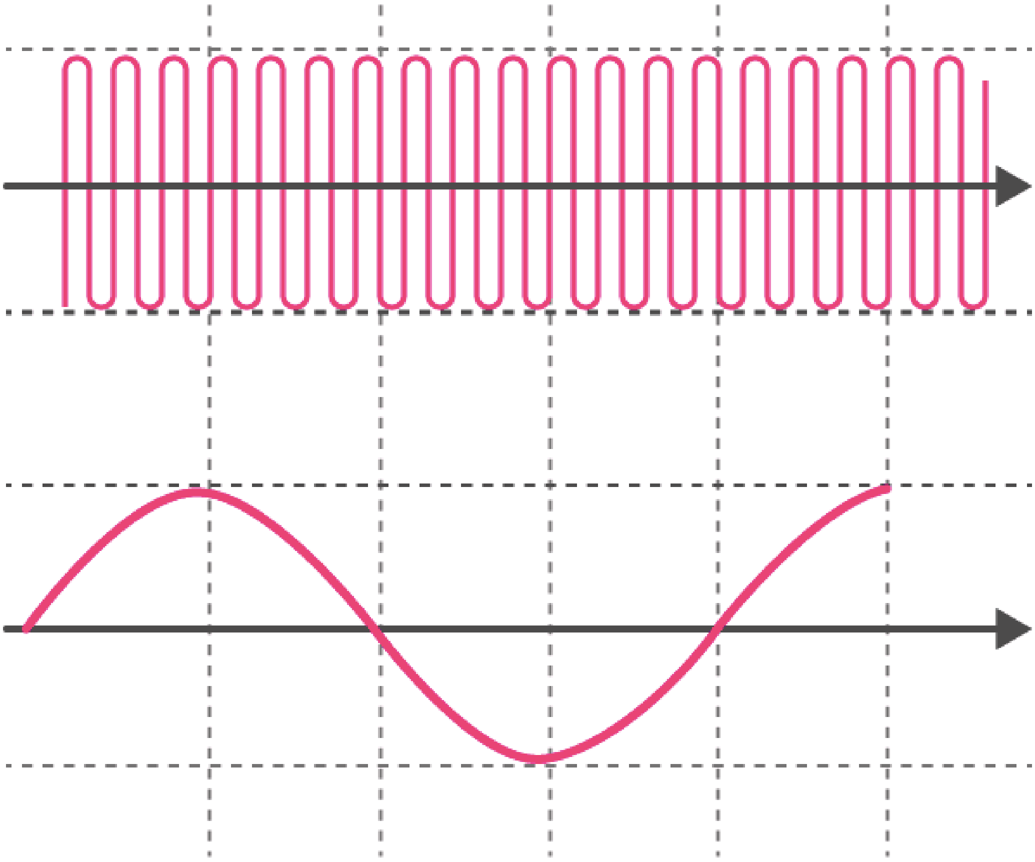
Designation	**Description**
A3E	Double-sideband a full-carrier
R3E	Single-sideband reduced-carrier
H3E	Single-sideband full-carrier
J3E	Single-sideband suppressed-carrier
B8E	Independent-sideband emission
C3F	Vestigial-sideband
Lincompex	Linked compressor and expander

Communication Systems and Modulation

We are studying modulation under communication systems. They are used to transmit and receive messages (information) from one place to another place in the form of electronic signals, and they are carried out in two different ways.
(i) Analog signal transmission
(ii) Digital signal transmission
So, we can represent an analogue electronic signal (information) as follows:
.
We can represent the analogue electronic signal either as a sine (or) cosine wave. Every wave will have an amplitude and phase.
Where m(t) = Modulating signal (input signal) or baseband signal
A_m = Amplitude of the modulating signal
$(\omega_m t + \Theta)$ = Phase of the signal phase contains both frequency $(\omega_m t)$ ad angle (Θ) term

What Is Modulation?

Basically, it is a process in a communication system. For communication, we need some fundamental elements. One is the high-frequency carrier wave, and the other is the information that has to be transmitted (modulating signal) or input signal. These are essential for communication which is done using a device from one place to another. All in all, we need the help of the communication system.
An electronic communication system converts our message (information) into an electronic signal, and the electronic signal is carried out by carrier waves to the destination.



Message (information)
or
Modulating signal

The superposition of modulating signal onto a carrier wave is known as modulation.
Modulation is defined as,
Varying any one of the fundamental parameters of a carrier wave in accordance with the modulating signal. A carrier wave can be represented as a sine or cosine.
$C(t) = A_c \sin(\omega_c t + \Theta)$

Amplitude Phase
If we vary the amplitude of the carrier wave in accordance with the modulating signal (input signal), it is known as amplitude modulation.
Similarly, it can be frequency modulation and phase modulation, too. In other words, modulation is the phenomenon of "superimposition of the modulating signal (input signal) into the carrier wave".

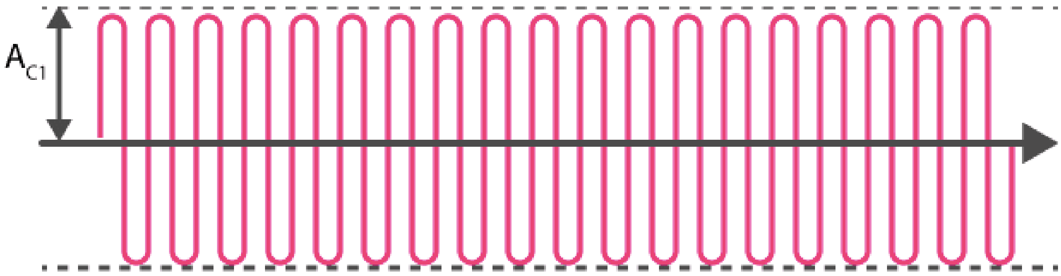
Why Do We Need Modulation?
Practically speaking, modulation is required for

- High range transmission
- Quality of transmission
- To avoid the overlapping of signals

High Range Transmission: (Effective Length of Antenna)

For effective communication, the length of the antenna should be λ/4 times the modulating signal.
$H_{min} = \lambda/4$
λ – Wavelength of the modulating signal or transmitting signal H> λ/4
For example, if I need to transmit a signal of a frequency of f = 20 kHz
As we know, c = f λ
$3 \times 10^8 = 20 \times 10^3 (\lambda)$
So
H_{min} = 3750 m
H_{min} = 3750 m is practically impossible; for that, we can transmit our modulating signal onto a carrier wave of frequency 1MHz. What we did here is we raised our transmission frequency from 20kHz to 1mHz.
Now, let us find out what the H_{min} is needed for good transmission.
$c = f\lambda$
$3 \times 10^8 = 1\times10^6 (\lambda)$
If we increase the transmitting frequency, wavelengths
will decreases so, H_{min} also decreases.
$3 \times 10^2 = \lambda$
λ = 300 m
This is practically possible, so we need modulation to increase the transmission frequency to transmit a low-frequency signal.
Quality of Transmission: (Power of Transmission by Antenna)
Since, from the Q-factor, we know sharpness or quality is maximum when power is maximum.
Sharpness or quality α power
Power radiated by a linear antenna is
Where l = Length of the antenna
λ = Wavelength of the transmitting signal
.
Avoiding the Overlapping of Signals
Two different transmitting stations transmit signals of the same frequency. They will get mixed up or overlap one another. To avoid this, we need to modulate these signals by different carrier waves.
When we talk about amplitude modulation, it is a technique that is used to vary the amplitude of the high-frequency carrier wave in accordance with the amplitude of the modulating signal. But the frequency of the carrier wave remains constant. Now, let us see what carrier waves and modulating signals are.

Common Terms
Carrier Wave (High Frequency)

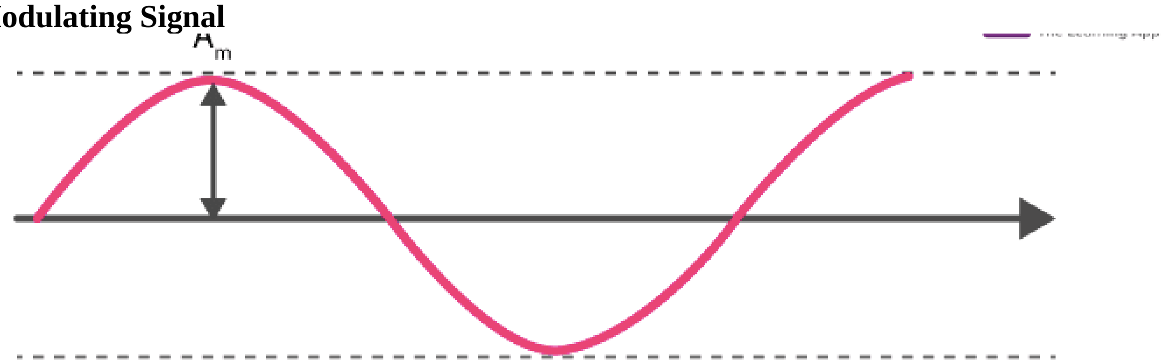


Carrier wave

The amplitude and frequency of a carrier wave remain constant. Generally, it will be high frequency, and it will be a sine or cosine wave of electronic signal; it can be represented as

$C(t) = A_c \sin w_c t$ ……………. 1

Modulating Signal



Modulating signal

The modulating signal is nothing but the input signal (electronic signal), which has to be transmitted. It is also a sine or cosine wave; it can be represented as

$m(t) = A_m \sin w_m t$

Where

A_c and A_m = Amplitude of the carrier wave and the modulating signal

$\sin w_c t$ = Phase of the carrier wave

$\sin w_m t$ = Phase of the modulating signal

Expression for Amplitude Modulated Wave

We have carrier wave and modulating signals,

.

m(t) = Modulating signal

c(t) = Carrier wave

A_m and Ac are the amplitude of modulating signal and carrier wave, respectively, in amplitude modulation. We are superimposing modulating signal into a carrier wave and also varying the amplitude of the carrier wave in accordance with the amplitude of the modulating signal, and the amplitude-modulated wave $C_m(t)$ will be

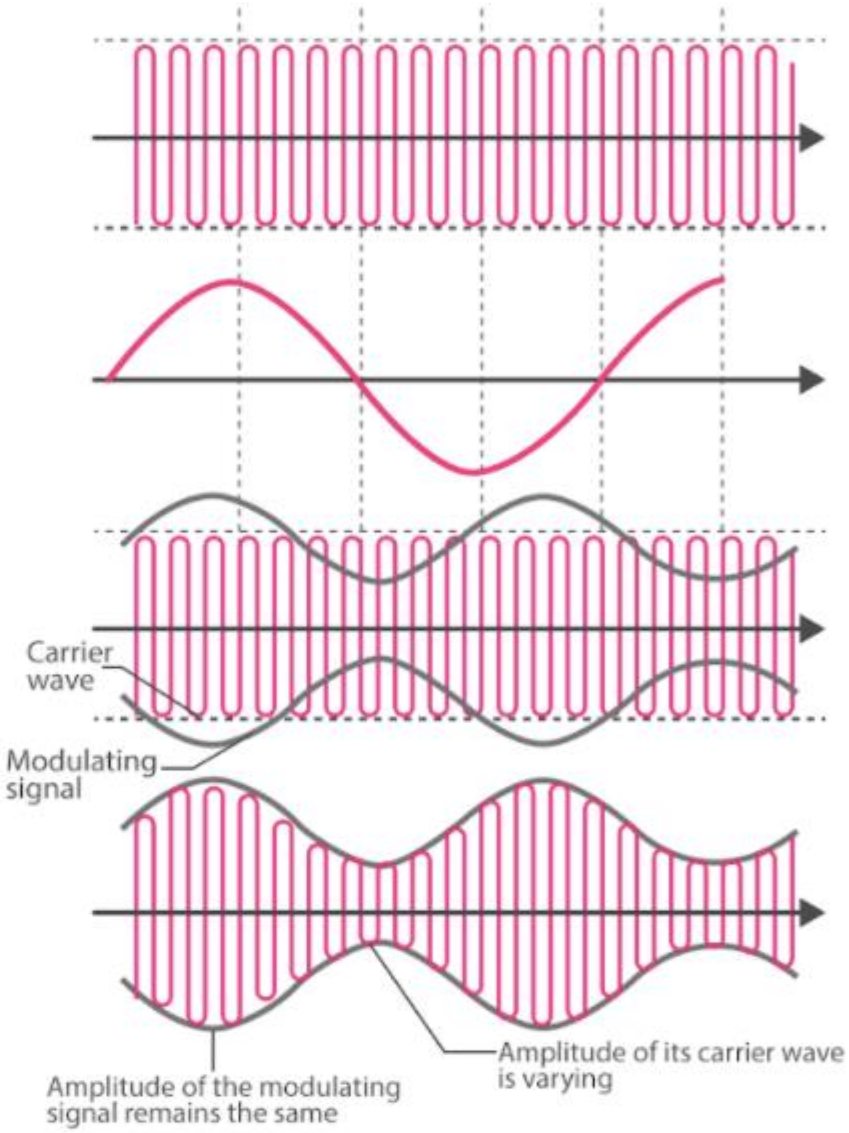
$C_m(t) = (A_c + A_m \sin \omega_m t) \sin \omega_c t$ ……………….. 2

This is the general form of an amplitude-modulated wave.

$C_m(t)$ is the amplitude-modulated wave

Where,
$A = A_c + A_m \sin \omega_m t$ = Amplitude of the modulated wave
$\sin w_c t$ = Phase of modulated wave
Where,
$C_m (t) = A_c \sin \omega_c t + A_c \mu \sin \omega_m t \sin \omega_c t$
We can rewrite the above equation as
From equation 3, we can see amplitude modulated wave is the sum of three sine or cosine waves.

Frequencies of Amplitude Modulated Wave
There are three frequencies in amplitude modulated wave – f_1, f_2 and f_3 – corresponding to ω_c, $\omega_c + \omega_m$ and $\omega_c - \omega_m$, respectively.
$\omega_1 = \omega_c \rightarrow$ it is corresponding $f_1 = f_c$
$\omega_2 = \omega_c + \omega_m \rightarrow$ it is corresponding $f_2 = f_c + f_m$
$\omega_3 = \omega_c - \omega_m \rightarrow$ it is corresponding $f_3 = f_c - f_m$
Where $f_c \rightarrow$ Carrier wave frequency
$f_c + f_m \rightarrow$ Upper side band frequency
$f_c - f_m \rightarrow$ Lower side band frequency
$f_m \rightarrow$ Modulating signal frequency
In general, $f_c >> f_m$
Bandwidth: (BW) It is the difference between the highest and lowest frequencies of the signal.
BW = Upper sideband frequency – Lower sideband frequency $(f_c - f_m)$
Or
$BW = f_{max} - f_{min}$
$BW = f_c + f_m - f_c + f_m = 2 f_m$
$BW = 2f_m$ = Twice the frequency of the modulating signal
Modulation Index
It is the ratio of the amplitude of the modulating signal to the amplitude of the carrier wave.
Amplitude Modulated Waveform
The waveform representation of amplitude modulated wave is given below.



1. Carrier wave
2. Modulating signal
3. Superposition of the carrier wave and modulating signal
4. Amplitude modulated wave

Summary:

Carrier wave, $c(t) = A_c \sin w_c t$

Modulating single $m(t) = A_m \sin w_m t$

Amplitude modulate wave $(m(t) = (A_c + A_m \sin \omega_m t) \sin \omega_c t$

Frequencies of modulated wave $\rightarrow f_c, f_c + f_m, f_c - f_m$

Bandwidth $(w) = f_c + f_m - (f_c - f_m) = 2f_m$

Advantages and Disadvantages of Amplitude Modulation

Advantages	**Disadvantages**
Amplitude modulation is easier to implement.	When it comes to power usage, it is not efficient.

Demodulation can be done using a few components and a circuit.	It requires a very high bandwidth that is equivalent to that of the highest audio frequency.
The receiver used for AM is very cheap.	Noise interference is highly noticeable.

Applications of Amplitude Modulation

While amplitude modulation use has decreased over the years, it is still present and has several applications in certain transmission areas. We will look at them below.

- ***Broadcast Transmissions:*** AM is used in broadcasting transmission over the short, medium and long wavebands. Since AM is easy to demodulate, radio receivers for amplitude modulation are, therefore, easier and cheaper to manufacture.
- ***Air-band Radio:*** AM is used in VHF transmissions for many airborne applications, such as ground-to-air radio communications or two-way radio links, for ground staff personnel.
- ***Single Sideband:*** Amplitude modulation in this form is used for HF radio links or point-to-point HF links. AM uses a lower bandwidth and provides more effective use of the transmitted power.
- ***Quadrature Amplitude Modulation:*** AM is used extensively in transmitting data in several ways, including short-range wireless links, such as Wi-Fi to cellular telecommunications and others.

These are some of the important applications of amplitude modulation.

Demodulation Methods

The most simple AM demodulator is made up of a diode that acts as an envelope detector. The product detector, which is another type of demodulator, can offer better-quality demodulation but with a complex additional circuit.

Solved Problems

Question 1:

A carrier wave of frequency f = 1mHz with a pack voltage of 20V is used to modulate a signal of frequency 1kHz with a pack voltage of 10v. Find out the following:

(i) μ

(ii) Frequencies of the modulated wave

(iii) Bandwidth

Solution:

(i)

(ii) Frequencies of modulated wave

$f \rightarrow f_c$, $f_c + f_m$ and $f_c - f_m$

f_c = 1mHz, f_m = 1kHz

$f_c + f_m = 1\times10^6 + 1\times10^3 = 1001 \times10^3 = 1001$ kHz

$f_c - f_m = 1\times10^6 - 1\times10^3 = 999 \times 10^3 = 999$ kHz

(iii) Bandwidth: (W)

(W) = Upper side band frequency – Lower side band frequency

$= f_c + f_m - (fc - fm)$

$= 2f_m$ = 1001 kHz – 999 kHz = 2 kHz

Question 2:

$y = 10 \cos (1800 \pi t) + 20 \cos 2000 \pi t + 10 \cos 2200 \pi t$. Find the modulation index (μ) of the given wave.

Solution:

As we know, the expression for amplitude modulated wave is

$C_m(t) = (A_c + A_m \cos \omega_m t) \cos \omega_c t$ ……………… 1
So, we have to bring the given wave equation into the known form
$y = 10 [\cos(1800 \pi t) + \cos (2200\pi t)] + 20 \cos 2000 \pi t$
We can rewrite the above equation as
$\cos(2000 \pi t) + \cos (1800 \pi t) = 2 \cos 2000 \pi t \cos 200 \pi t$
Compare equations 1 and 2.
$A_c = 20$
$A_m = 20$
Then, the modulation index (μ)
We can also find the frequencies of the modulated wave and B and width
(ii) Frequencies off the modulated wave
We know frequencies are fc, fc + fm and fc – fm from the modulated wave expression.
$y = (A_c + A_m \cos \omega_m t) \cos \omega_c t$ ……………….. 4
Comparing equations 3 and 4, we get
$\cos \omega_m t = \cos (200 \pi t)$
$\omega_m = 200 \pi$
$2 \pi f_m = 200 \pi$
$f_m = 100$ Hz
Similarly,
$\cos \omega_c t = \cos 2000 \pi t$
$\omega_c = 2000 \pi$
$2\pi f_c = 2000 \pi$
$f_c = 1000$ Hz
f_c, $f_c + f_m$ and $f_c - f_m$, respectively, 1000 Hz, 1100 Hz and 900 Hz.
(iii) Bandwidth (W)
$W = f_c + f_m - (f_c - f_m) = 2f_m$
$W = 200$ Hz

MATLAB Code:

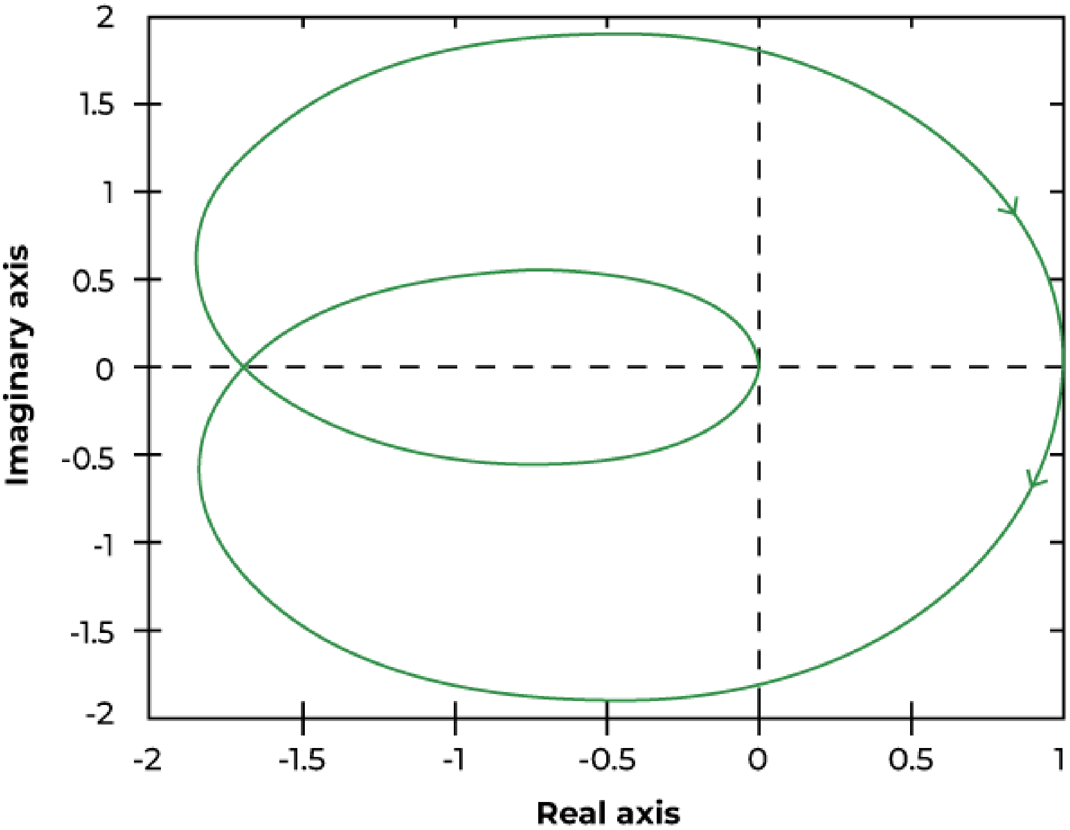
```
% Amplitude Modulation
Fs = 10000; % Sampling frequency
t = 0:1/Fs:1-1/Fs;
Fc = 100; % Carrier frequency
Fm = 5; % Modulating frequency
Am = 1; % Modulating amplitude
carrier = cos(2*pi*Fc*t);
modulator = cos(2*pi*Fm*t);
am_signal = (1 + Am*modulator) .* carrier;
plot(t, am_signal);
title('Amplitude Modulated Signal');
xlabel('Time (s)');
ylabel('Amplitude');
```

15. Nyquist Plot

A Nyquist plot is a graphical representation used in control engineering. It is used to analyze the stability and frequency response of a system. The plot represents the complex transfer function of a system in a complex plane. The x-axis represents the real part of the complex numbers and the y-axis represents the imaginary part. Each point on the Nyquist plot reflects the complex value of the transfer function at that frequency.

Nyquist Stability Criteria

It is used to determine the stability of a control system. This criterion works on the principle of argument. It is useful for feedback control system analysis and is expressed in terms of frequency domain plot. It is applicable for minimum and non-minimum phase systems.



Nyquist Plot

According to the Nyquist Stability Criterion the number of encirclement of the point (-1, 0) is equal to the P-Z times of the closed loop transfer function. The equation for stability analysis is given below:

N = Z – P —— (equation 1)

Where,

P = open loop pole of the system on right hand side (RHP)

Z = close loop zero of the system on right hand side (RHP)

N = number of encirclement around (-1,0)

Note: ‘N’ is negative for anticlockwise encirclement around (-1,0) and positive for clockwise encirclement around (-1,0).

Stability Cases:

N (encirclements)	Condition	Stability
0 (no encirclement)	Z = P = 0 $ Z = P	Marginally stable
N > 0 (clockwise encirclement)	P = 0, Z ≠0 & Z > P	Unstable
N < 0 (anti-clockwise encirclements)	Z = 0, P ≠0 & P > Z	Stable

Principle of Argument in Nyquist Plot

It states that if in a transfer function, there are ‘P’ poles and ‘Z’ zeros enclosed in s-plane contour, then the corresponding GH plane contour will encircle the origin ‘Z-P’ times as seen in equation 1.

- If the contour encircles more zeros than poles (Z > P), there is increase in the encirclement number which leads to a positive value of N. This corresponds to a clockwise encirclement of the critical point, indicating instability.
- If the contour encircles more poles than zeros (P > Z), there is decrease in encirclement number which leads to a negative value of N. This corresponds to a anti-clockwise encirclement of the critical point, indicating stability.
- If the number of poles and zeros enclosed by the contour are equal (Z = P), the encirclement number is zero, indicating a marginally stable system. The contour neither contributes to a net clockwise nor counter-clockwise encirclement.

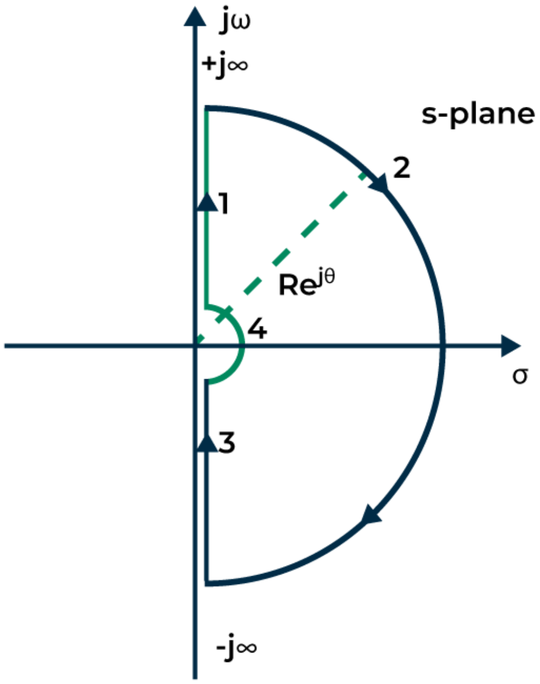
Important Terminologies of Nyquist Plot

Important terminologies of Nyquist Plot are mentioned below:

- Nyquist Path or Contour
- Nyquist Encirclement
- Nyquist Mapping
- Gain Crossover Frequency
- Phase cross over frequency
- Phase Margin
- Gain Margin

1. Nyquist Path or Contour

The Nyquist Path or Contour is a closed contour on the right side of the s-plane. To enclose the entire RHS of the plane, a large semicircle lane is drawn with a diameter along the ‘j’ axis and center at the source. The semicircle radius is simply treated as Nyquist Encirclement. The Nyquist Contour is shown in the figure below:



Nyquist Contour

In the above plot, marked points are denoting:

Point 1 is denoting $s = j\omega$

Point 2 is denoting s = lim R→ooRejθlimR→ooRejθ where, $\theta = +90^o$ to -90^o

Point 3 is denoting $s = -j\omega$

Point 2 is denoting s = lim r→0rejθlimr→0rejθ where, $\theta = -90^o$ to $+90^o$

2. Nyquist Encirclement

It is a point, if it is found in the curve, is known to be encircled by a line.

3. Nyquist Mapping

Mapping is the process of transforming a point in the s-plane into a point in the F(s) plane, and F(s) is the result of mapping.

4. Gain Crossover Frequency

It is the frequency at which the Nyquist plot has unity magnitude. It is denoted by 'ω_{gc}'.

5. Phase cross over frequency

It is the frequency at which point the Nyquist plot crosses the negative real axis is called the phase cross-over frequency and it is denoted with 'ω_{pc}'.

6. Phase Margin

The phase margin indicates how much more phase shift we may put in the open loop transfer function before our system becomes unstable. It can be calculated from the phase at the gain cross-over frequency.

7. Gain Margin

The gain margin is the amount of open loop gain that can be increased before our system becomes unstable. It can be calculated from the gain at the phase cross-over frequency.

How to Draw Nyquist Plot?

Procedure to draw Nyquist Plot:

1. Draw Polar Plot

2. Draw Inverse polar Plot
3. Draw and map using Nyquist Contour

Rules of Nyquist Plot

1. Determine the transfer function of the system.
2. Calculate the complex transfer function value by putting s= jω, where 'ω' is angular frequency and 'j' is the imaginary unit.
3. Determine the magnitude and phase of the complex transfer function.
4. Plot the points in the polar coordinate for each frequency point.
5. Connect all the points to create a smooth Nyquist curve. The curve represents the change in magnitude and phase with frequency.
6. Calculate the number of encirclements around the point (-1,0). Each encirclement represents the value of 'N'. The stability can be calculated using Nyquist Stability Criteria as discussed above.

Stability Analysis Using Nyquist Plot

The nyquist plot helps in analysis of the stability of a control system. It tells whether the system is stable, unstable or marginally stable. The stability is analyzed using various parameters which are as follows:

- Gain cross-over frequency
- Phase cross-over frequency
- Gain margin
- Phase margin

Phase Cross Over Frequency

It is the frequency at which point the Nyquist plot crosses the negative real axis is called the phase cross-over frequency and it is denoted with 'ω_{pc}'.

The stability of the control system based on the relationships between the two frequencies i.e., phase cross-over and gain cross-over is given below:

ωpc > ωgc ->System is stable

ωpc < ωgc ->System is unstable

ωpc = ωgc ->System is marginally stable

Phase Margin

The phase margin indicates how much more phase shift we may put in the open loop transfer function before our system becomes unstable. It can be calculated from the phase at the gain cross-over frequency.

Phase Margin (PM) = $180^\circ + \angle G(j\omega)H(j\omega)|_{\omega=\omega gc}$

The stability of the control system based on the relationships between the two margins i.e., gain margin and phase margin is given below:

GM > 1 and PM is positive -> System is stable

GM < 1 and PM is negative ->System is unstable

GM = 1 and PM is 0° ->System is marginally stable

Gain Crossover Frequency

It is the frequency at which the Nyquist plot has unity magnitude. It is denoted by 'ω_{gc}'.

Gain Margin

The gain margin is the amount of open loop gain that can be increased before our system becomes unstable. It can be calculated from the gain at the phase cross-over frequency.

Gain Margin (GM): 1|G(jw)H(jw)|w=wpc|G(jw)H(jw)|1w=wpc

Let us understand the above concept with the help of an example.

Question: Find the gain crossover frequency and phase margin of the given transfer function

G(s)=1s(s+1)G(s)=s(s+1)1

Solution

PM = *180°+∠G(jω)H(jω)* $/_{\omega=\omega gc}$

ω_{gc} = Frequency at which magnitude of G(s)H(s) is equal to 1.

G(jω)H(jω) = 1jω(jω+1)jω(jω+1)1

|G(jω)H(jω)| = 1

1ω1+ω2ω1+ω2

1 = 1

On solving the quadratic equation we will get:

ω_{gc} = −1±522−1±5

(we will neglect the negative term inside the root)

ω_{gc} will now become = −1+522−1+5

ω_{gc} = 0.786 rad/sec

Now finding the phase at gain crossover frequency

∠G(jω)H(jω) $/_{\omega=\omega gc}$ = −Π2−2Π – $\tan^{-1}$ (ω_{gc})

∠G(jω)H(jω) $/_{\omega=\omega gc}$ = −Π2−2Π – $\tan^{-1}$ (0.786)

∠G(jω)H(jω) $/_{\omega=\omega gc}$ = -90° – 38.16°

∠G(jω)H(jω) $/_{\omega=\omega gc}$ = -128.16

PM = 180 – 128.16

PM = 51.84°

Solved Example of Nyquist Plot

1. Draw the Nyquist plot for the system whose open loop transfer function is given by G(s)H(s) = ks(s+2)(s+10)s(s+2)(s+10)k

Also determine the value of 'k' for which the system is stable.

Solution

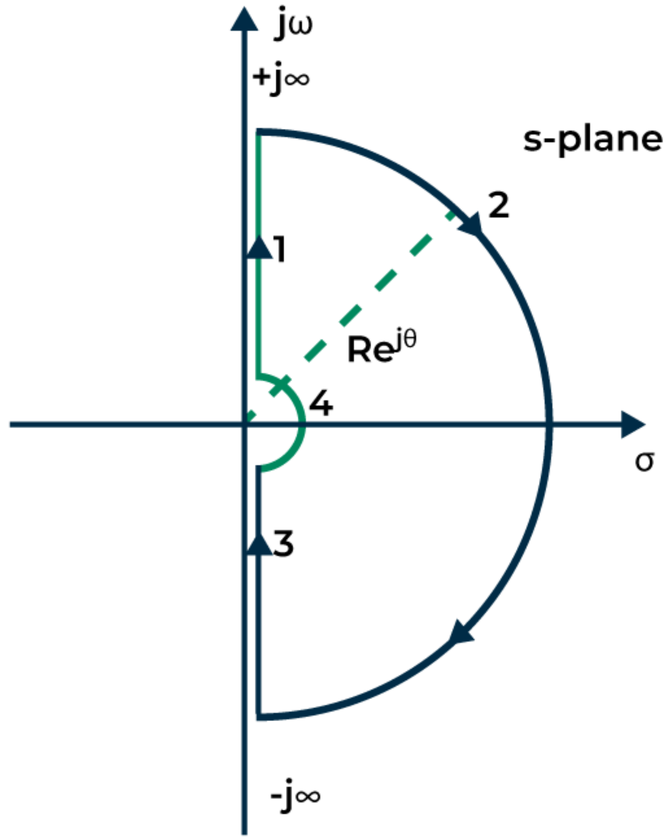
Step 1: Identify the poles and zeros

There are 3 poles and no zeros

Poles: 0, -2, -10

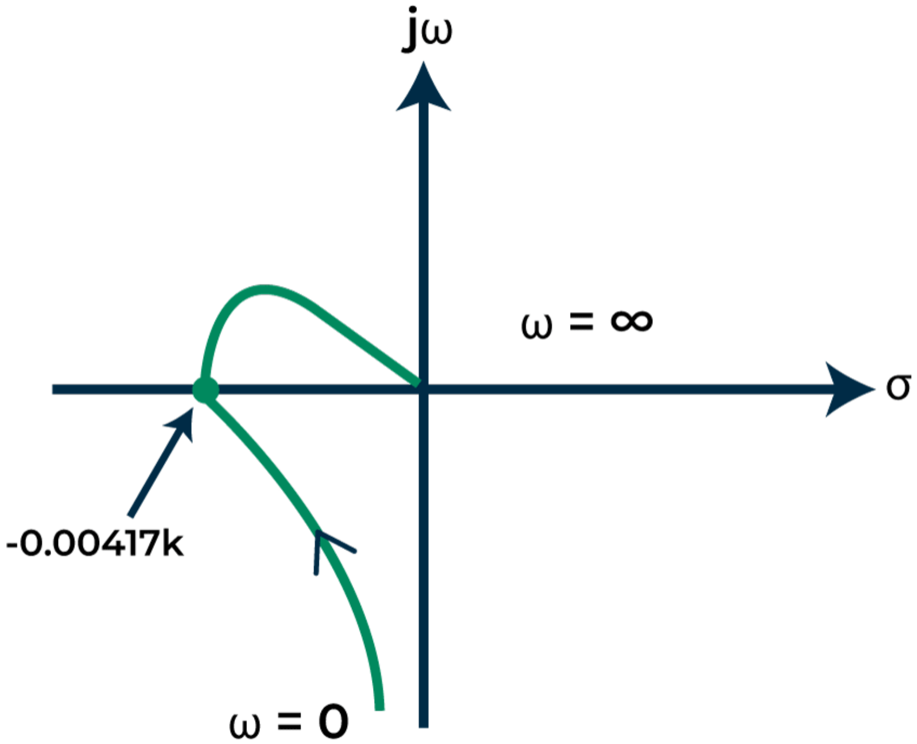
G(s)H(s) = 0.05ks(1+0.5s)(1+0.1s)s(1+0.5s)(1+0.1s)0.05k

Step 2: Mapping of all the 4 section as seen in the given image



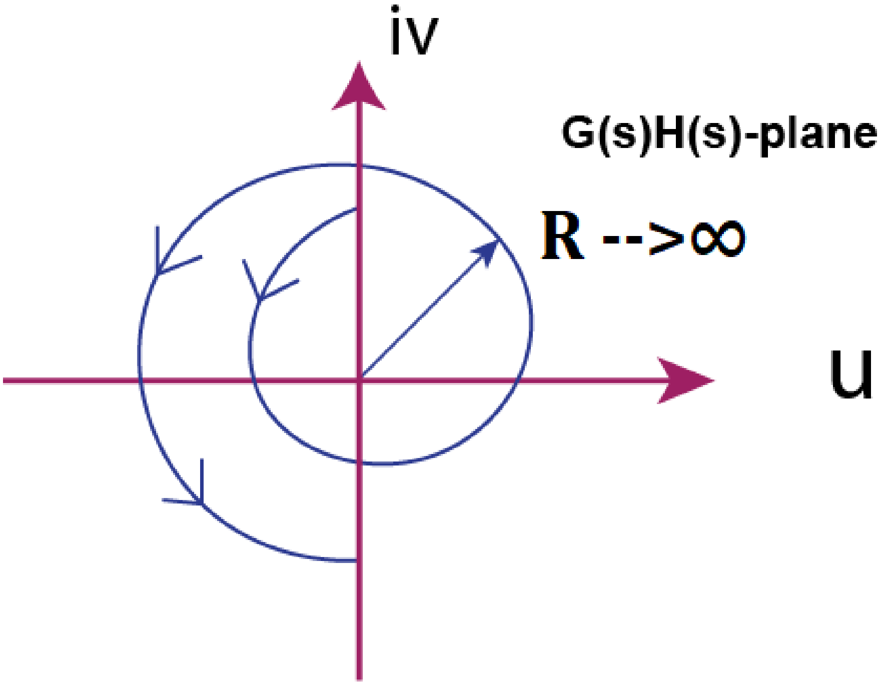
Nyquist Contour

The value of **ω** in the region (1) is s = jω. Replacing the transfer function with s = jω
G(jω)H(jω) = 0.05kjω(1+0.5jω)(1+0.1jω)jω(1+0.5jω)(1+0.1jω)0.05k
G(jω)H(jω) = 0.05kjω(1+0.6jω–0.05ω2)jω(1+0.6jω–0.05ω2)0.05k
G(jω)H(jω) = 0.05k−0.6ω2 +jω(1–0.05ω2)−0.6ω2 +jω(1–0.05ω2)0.05k —— equation 1
Now keeping the imaginary term 0 to find the phase crossover frequency.
$\omega(1 - 0.05\omega^2) = 0$
$1 - 0.05\omega^2 = 0$
ω = 4.472 radsecsecrad = phase crossover frequency
Putting the value of frequency in the real part of the equation 1.
G(jω)H(jω) = 0.05k−0.6ω2−0.6ω20.05k
G(jω)H(jω) = 0.05k−0.6∗4.4722−0.6∗4.47220.05k
G(jω)H(jω) = -0.00417K
The polar plot will start at -90 degrees and crosses the x axis at point -0.00417K as given below:



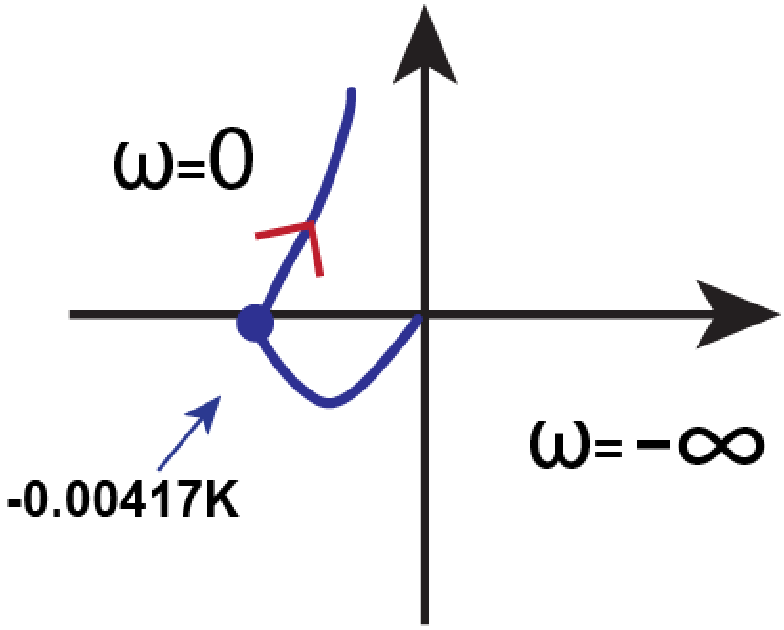
Polar Plot

Step 3: Mapping the region 2 whose equation is given by: s = lim R→ooRejθlimR→oo Rejθ where, θ = +90º to -90º
If the transfer function is in the form G(s)H(s) =
0.05ks(1+0.5s)(1+0.1s)s(1+0.5s)(1+0.1s)0.05k, then assume that (1 + sT) is equal to sT. It is given by:
G(s)H(s) = 0.05ks∗0.5s∗0.1ss∗0.5s∗0.1s0.05k
G(s)H(s) = 0.05k0.05s30.05s30.05k
G(s)H(s) = ks3s3k
Let s = lim R→ooRejθlimR→ooRejθ
G(s)H(s) = 0e−j3θ0e−j3θ
At θ=Π2θ=2Π, G(s)H(s) = 0e−j3Π20e−j23Π
At θ=−Π2θ=2−Π, G(s)H(s) = 0ej3Π20ej23Π
Thus, the region 2 varies from -270 degrees to 270 degrees. It is represented below:



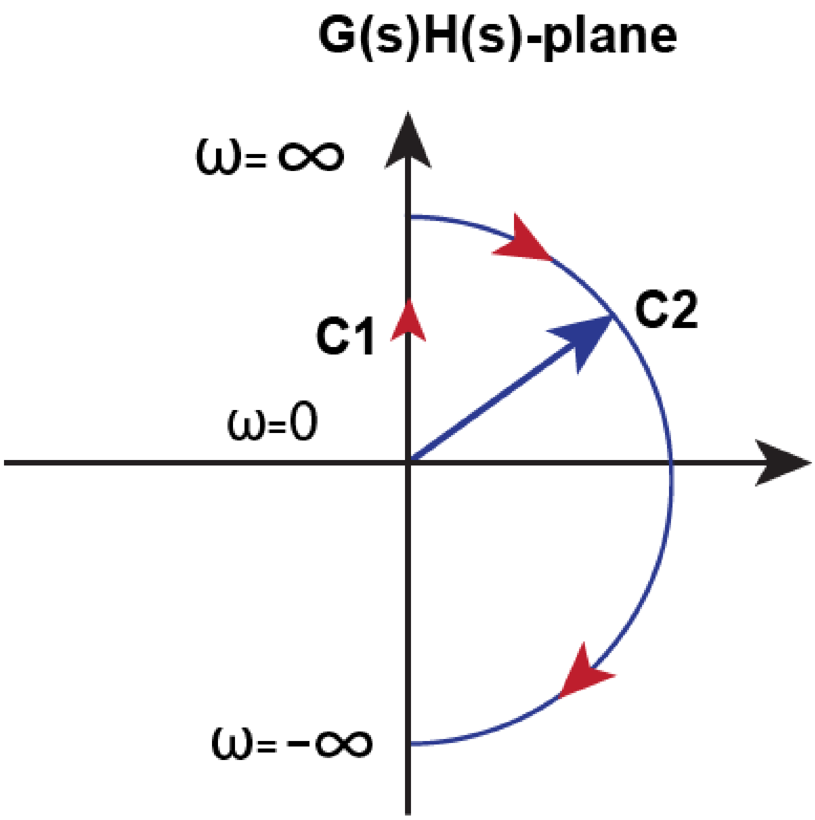
Graph for Region 2 Mapping

Step 4: Now map the region 3 of the nyquist contour. The region 3 is simple inverse of region 1. Thus the graph will mirror image of region 1 graph which is shown below:



Mapping of Region 3 Graph

Step 5: Mapping the region 4 whose equation is given by: s = lim r→0rejθlimr→0rejθ ,where, θ = -90º to +90º
If the transfer function is in the form G(s)H(s) = 0.05ks(1+0.5s)(1+0.1s)s(1+0.5s)(1+0.1s)0.05k, then assume that (1 + sT) is equal to 1. It is given by:
G(s)H(s) = 0.05ks∗1∗1s∗1∗10.05k
G(s)H(s) = 0.05kss0.05k
Let s = s = lim r→0rejθlimr→0rejθ
G(s)H(s) = ∞e−jθe−jθ
At θ=Π2θ=2Π, G(s)H(s) = ∞e−jΠ2ye−j2Πy
At θ=−Π2θ=2−Π, G(s)H(s) = ∞ejΠ2yej2Πy
So the graph will be circular arc of infinite radius as seen in the given image.



-

Region 4 Mapping Graph

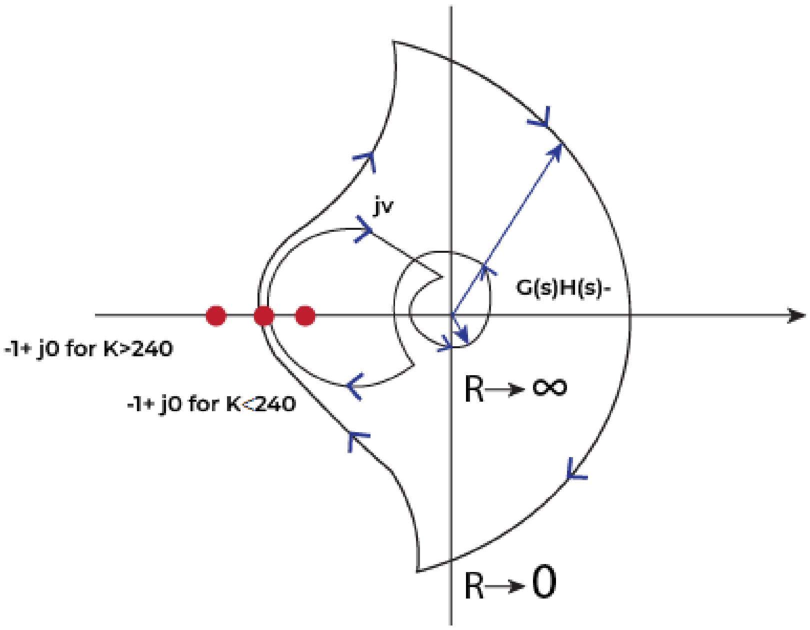
Step 6: Now we will perform the stability analysis. To find the value of ‘k’, we will check when our contour will pass through the point (-1+j0)

$-0.00417K = -1$

$K = 1/0.00417$

$K = 240$

Step 7: Final Nyquist Plot is shown below. It as been plotted by combining the mapping of all the four regions.



Nyquist plot of $G(s)\,H(s) = \dfrac{K}{s(s+2)(s+10)}$

Final Nyquist Plot

Now we will check foe which value of ‘k’, the system is stable.

Case 1: If k< 240

The point -1+j0 is not encircled. This means that there are no poles on the right half of the plane. This means the system is stable for k less than 240.

Case 2: k>240

The point -1+j0 is encircled two times in the clockwise direction. This means that Z>P and hence the system is unstable.

Stability condition: $0 < K < 240$

Advantages of Nyquist Plot

1. It provides a graphical representation of the system that how a system responds to different frequencies.
2. It simplifies the calculation of the stability of a system without calculating the poles explicitly.
3. It provides the number of unstable poles in the closed loop system by just calculating the encirclements around critical points.
4. It helps in adjusting the parameters like gain and phase margins to achieve stability.
5. It is useful to analyze the stability and behavior of the feedback control system.

Disadvantages of Nyquist Plot

1. It is very time consuming to draw it by hand and more prone to errors with complex transfer function.
2. It is applicable to LTI system. It cannot be applied to non-linear or time varying systems.

3. It rely on open-loop transfer function and can lead to inaccurate representation of the behavior of closed-loop system.
4. It does not provide the information about transient response.

Application of Nyquist Plot

- **Stability analysis:** It is used in stability analysis of a control system by determining the number of encirclements around a specific point.
- **Control system design:** It is used in adjusting the controller parameters like gain margins and phase margins to achieve the desired stability of a control system.
- **Phase margin evaluation:** It represents the system's robustness and stability margin. It can be directly read from the plot and can be adjusted accordingly to achieve the stability.
- **Filter designing:** It is used in the design of filters and equalizers to optimize frequency response and phase characteristics for desired signal processing applications.
- **Feedback loop analysis:** It is used in the analysis of the feedback loops which affect the stability and performance of a control system. It shows that how change in controller or system parameters influence the overall behavior.

Theory: Plot the Nyquist plot of a transfer function.
MATLAB Code:

```
% Nyquist Plot
sys = tf([1], [1 1 1]);
nyquist(sys);
title('Nyquist Plot');
```

Theory: Plot the Bode plot of a transfer function.
MATLAB Code:

```
% Bode Plot
sys = tf([1], [1 1 1]);
bode(sys);
title('Bode Plot');
```

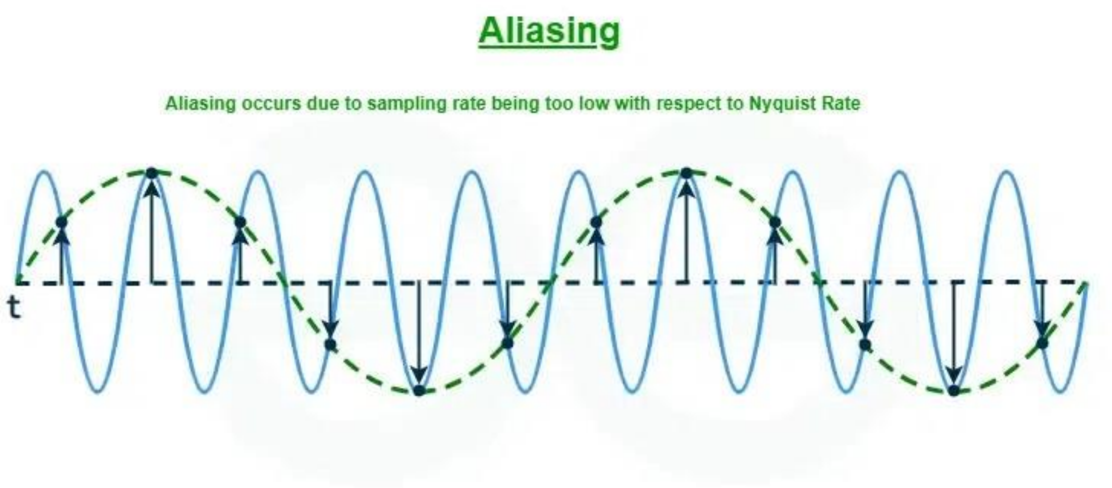
17. Signal Sampling and Aliasing

The aliasing effect, also known as aliasing distortion or simply aliasing, is a phenomenon that occurs in signal processing, particularly in digital signal processing (DSP), when a continuous signal is sampled at a frequency that is too low to accurately represent the original signal.

In digital communication, aliasing occurs due to a measurement error in the signal because of an incorrect sampling rate, if the sampling rate is too low aliasing may occur. In simple terms, the Nyquist rate, also known as the Nyquist sampling theorem, is the minimum sampling rate required to accurately sample and reconstruct a signal without distortion. It is defined as twice the maximum frequency present in the signal. Sampling at a rate below the Nyquist rate can lead to aliasing

Points to remember:

- Aliasing is an effect which occurs when the input frequency is half the sampling frequency
- It causes distortion in reconstructed signals
- Anti-aliasing filters are used to prevent aliasing
- Aliasing mainly occurs in digital audio and digital images



Aliasing occurs due to sampling rate being too low with respect to Nyquist Rate

Cause of Aliasing

Aliasing is mainly caused by poor sampling, if the sampling rate is too low aliasing may occur, aliasing can also occur when signals are not sampled fast enough but mainly aliasing can occur when the sample rate is too low for accurately reconstructing the original signal or when high frequencies in the signal exceed half the sample rate.

Well, how can we determine the correct cause of Aliasing? The answer is, to check if the sample rate is sufficient for accurately capturing the original signal. If the sample rate is too low, aliasing can occur due to incorrect sampling. Nyquist derived the Nyquist theorem, also known as the sampling theorem, which is fundamental concept to accurately reproduce any analog signals into its correct digital form.

Nyquist states that to reconstruct a signal, the sampling rate must be at least twice the frequency of the signal being sampled.

Applications of Aliasing

Aliasing mainly has a negative impact on signals, but sometimes it can have positive impact also but it is quite rare. Here are some few applications of Aliasing:

- **To lower frequency of any signal:** Aliasing can be used to convert any high frequency signal into low frequency signal.
- **Medical image processing:** Aliasing is used in medical fields to process signals in their correct form.
- **Facial and text image analysis:** Aliasing has been found useful in analyzing faces and text quite efficiently by using pure sine wave signals.
- **Biometric Recognition:** Aliasing can also recognize biometrics such as fingerprints quite reliably.

Effects of Aliasing

There are a few effects of aliasing and why it should be prevented:

- **Signal Accuracy:** Aliasing makes signal become distorted which can cause unwanted problems in any signal. This can be a major problem in Audio, which can cause audio instruments to sound distorted and also in Video, which can cause sharp/pixelated or jagged edges in pictures.
- **Reconstruction of Signal:** Due to aliasing, it may become impossible to perfectly reconstruct orignal signal from its sample because of data loss. Hence aliasing can make reconstruction of signals hard.
- **Signal processing:** Aliasing can make signal processing complicated by producing unwanted noise.
- **Poor signal quality:** Aliasing can negatively impact any signals quality and cause distortion and corruption of the signal.
- **Jagged or blocky appearance:** Aliasing in imagery can also cause jagged edges, pixelated or blocky patterns in the image.

Advantages of Aliasing

Some of the Advantages of Aliasing are :

- **Bandwidth Reduction**: In certain applications where bandwidth is limited, such as in telecommunications or data compression, controlled aliasing techniques can be used to reduce the amount of data that needs to be transmitted or stored.
- **Simulation of Analog Systems**: In certain contexts, aliasing can mimic the behavior of analog systems.
- **Data Analysis**: In signal processing and data analysis, aliasing can sometimes provide useful insights or simplify analysis.

Disadvantages of Aliasing

- It leads to noise, which can disrupt a signal.
- It leads to distortion and pixelation of of any signal.
- It disrupts data signal transmission.
- It can lead to degrade the quality of the signal which can lead to loss of data.
- It can interfere with the accurate detection of signals, leading to missed or false detections.
- It can cause misinterpretation of the signal

Prevention of Aliasing

Aliasing should be prevented to stop distortion in any signal Although aliasing can be prevented just by correctly sampling the signal, there are mainly two methods to prevent aliasing:

Method 1: Using Anti-aliasing Filters

Anti-aliasing filters: These are special filters which can prevent or block any frequencies higher than a specific limit. It simply filters out the unwanted frequencies in any signal.

Advantages of Anti-aliasing Filters

- **Effectively prevents aliasing:** Anti-aliasing filters are a sure-fire or reliable way of prevent aliasing by blocking the unwanted frequencies.
- **Quality Enhancement:** Anti-Aliasing filters can enhance the quality of images by preventing pixelated or distorted parts.
- **Boost Visual Appeal:** Anti-aliasing filters can improve the visual representation of any object

Disadvantages of Anti-Aliasing Filters

- **Time effect:** This refers to the undesirable effects because of the long time responses that occur in digital audio in video, In simple term it means sampling can cause a lot of time to achieve.
- **Phase distortion:** This method cannot prevent aliasing in phase distortions.
- **Unwanted frequency modification:** Sometimes this method will cause the unwanted modification of frequency which can cause another noise to be produced in a signal.

Method 2: Using Oversampling

Oversampling: Oversampling is a technique which is used to measure the signal at a much higher rate than actually needed, which pushes the Nyquist frequency higher. These higher frequencies are then removed without affecting the desired signal. It is a popular method of reducing Aliasing in digital audio and video.

Advantages of Oversampling

- **Helps preserve the data integrity:** Oversampling does not cause any damage to the original signal or file, which helps promote data integrity and prevents data loss.
- **Reliable:** Oversampling is a reliable way to prevent aliasing in any digital signal.
- **Performance:** Oversampling can make the signal perform better by preventing distortion and also giving it a much higher frequency.

Disadvantages of Oversampling

- **Sampling error:** Oversampling may lead to sampling error since the measured signal is at a higher frequency than the original.
- **Time consuming:** Sampling is a time consuming process overall, Oversampling is even more time consuming because it requires you to measure any signal in its highest frequency
- **Cost and Complexity:** Implementing oversampling techniques may require additional computational power and cause an overall more complex signal to be formed.

Theory: Demonstrate the effects of sampling and aliasing on a signal.
MATLAB Code:

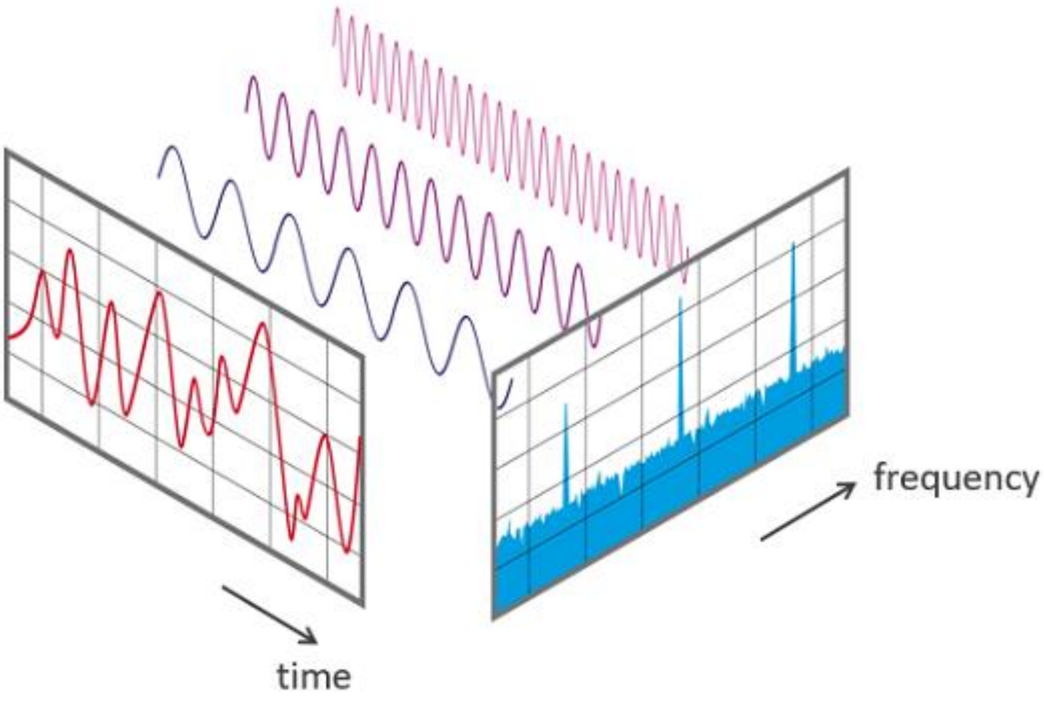
```
% Signal Sampling and Aliasing
Fs = 50; % Sampling frequency
t = 0:1/Fs:1-1/Fs;
f_signal = 20; % Signal frequency
x = sin(2*pi*f_signal*t);
stem(t, x);
title('Signal Sampling and Aliasing');
```

```
xlabel('Time (s)');
ylabel('Amplitude');
```

18. Fast Fourier Transform (FFT)

The "Fast Fourier Transform" (FFT) is an important measurement method in the science of audio and acoustics measurement. It converts a signal into individual spectral components and thereby provides frequency information about the signal. FFTs are used for fault analysis, quality control, and condition monitoring of machines or systems. This article explains how an FFT works, the relevant parameters and their effects on the measurement result.

Strictly speaking, the FFT is an optimized algorithm for the implementation of the "Discrete Fourier Transformation" (DFT). A signal is sampled over a period of time and divided into its frequency components. These components are single sinusoidal oscillations at distinct frequencies each with their own amplitude and phase. This transformation is illustrated in the following diagram. Over the time period measured, the signal contains 3 distinct dominant frequencies.



View of a signal in the time and frequency domain

Step by step

In the first step, a section of the signal is scanned and stored in the memory for further processing. Two parameters are relevant:

1. The sampling rate or sampling frequency fs of the measuring system (e.g. 48 kHz). This is the average number of samples obtained in one second (samples per second).
2. The selected number of samples; the blocklength BL. This is always an integer power to the base 2 in the FFT (e.g., 2^10 = 1024 samples)

From the two basic parameters fs and BL, further parameters of the measurement can be determined.

Bandwidth fn (= Nyquist frequency). This value indicates the theoretical maximum frequency that can be determined by the FFT.

$fn = fs / 2$

For example at a sampling rate of 48 kHz, frequency components up to 24 kHz can be theoretically determined. In the case of an analog system, the practically achievable value is usually somewhat below this, due to analog filters - e.g. at 20 kHz.

Measurement duration D. The measurement duration is given by the sampling rate fs and the blocklength BL.

$D = BL / fs.$

At fs = 48 kHz and BL = 1024, this yields 1024/48000 Hz = 21.33 ms

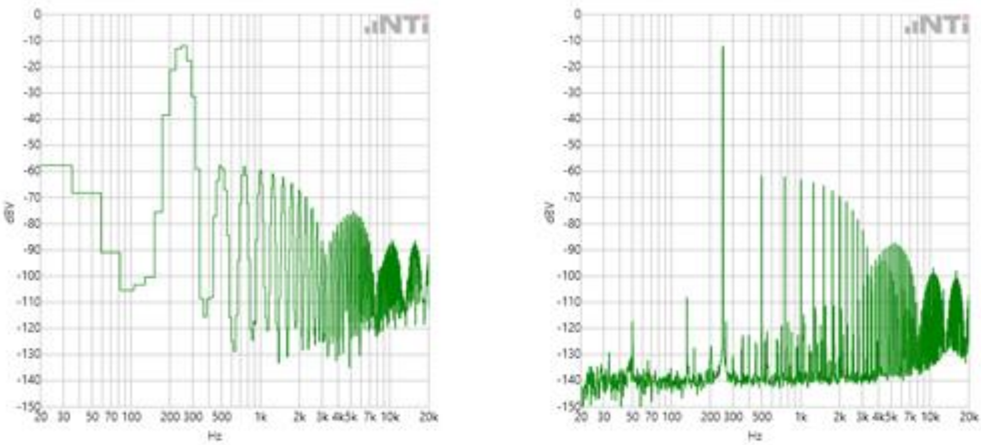
Frequency resolution df. The frequency resolution indicates the frequency spacing between two measurement results.

$df = fs / BL$

At fs = 48 kHz and BL = 1024, this gives a df of 48000 Hz / 1024 = 46.88 Hz.

In practice, the sampling frequency fs is usually a variable given by the system. However, by selecting the blocklength BL, the measurement duration and frequency resolution can be defined. The following applies:

- A small blocklength results in fast measurement repetitions with a coarse frequency resolution.
- A large blocklength results in slower measuring repetitions with fine frequency resolution.



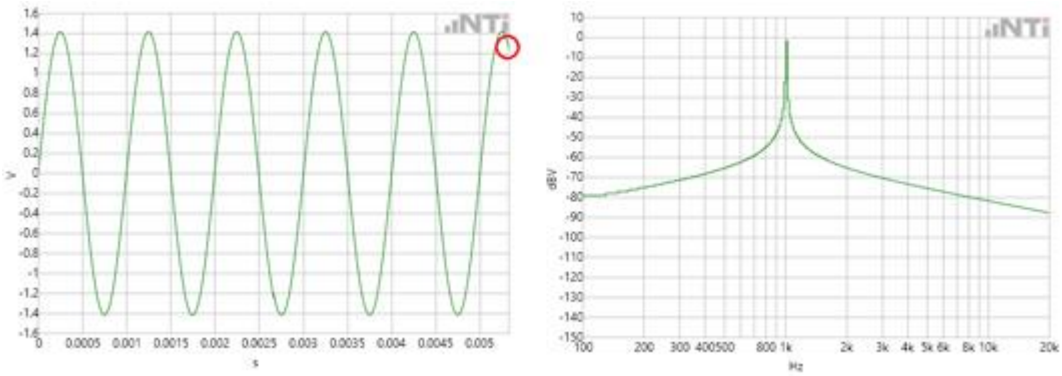
Representation of the FFT of a signal with small and large blocklength

To Infinity…

In the Fourier transformation, the assumption is that the sampled signal segment is repeated periodically for an infinite period of time. This brings two conclusions:

1. The FFT is only suitable for periodic signals.
2. The sampled signal segment must contain a whole number of periods.

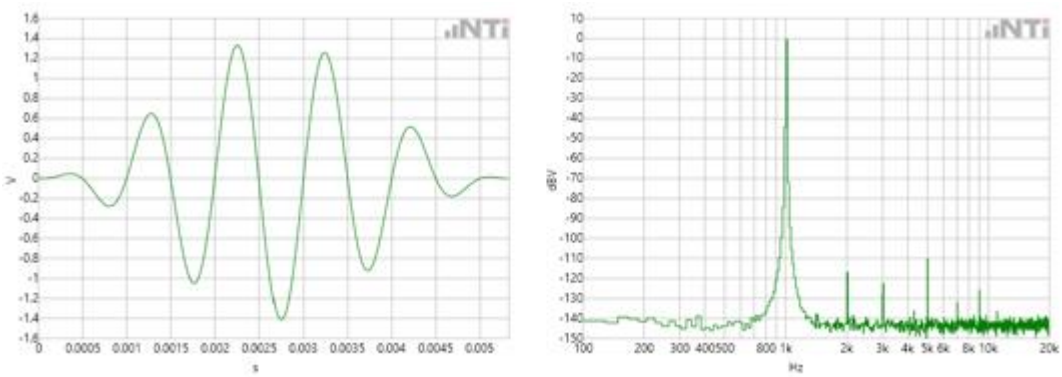
It can be seen that condition 2. would apply only to very few signals. The sampling of a signal whose frequencies are not an integer multiple of df would begin and end within a block of 2^n samples with different values. This results in a jump in the time signal, and a "smeared" FFT spectrum. (aka Leakage)



Un-windowed time signal with smeared spectrum

Windowing

In order to prevent this smearing, in practice "windowing" is applied to the signal sample. Using a weighting function, the signal sample is more or less gently turned on and off. The result is that the sampled and subsequent "windowed" signal begins and ends at amplitude zero. The sample can now be repeated periodically without a hard transition.



Windowed time signal with spectrum

A practical example

A classic example of the signal theory is the spectral composition of a square-wave signal. This consists of the sum of all weighted odd multiples of the fundamental frequency.

$$f(t) = \frac{4h}{\pi}\left[\sin(\omega t) + \frac{1}{3}\sin(3\omega t) + \frac{1}{5}\sin(5\omega t) + \ldots\right]$$

This video shows the creation of a 500 Hz square wave as a time signal and spectrum.

How to measure?

The portable audio and acoustic analyzer XL2 is ideally suited for fast and simple FFT analysis up to 20 kHz. For multi-channel and more detailed analysis or calculations, a more powerful system with large bandwidth and fast signal processors such as the FLEXUS FX100 Audio Analyzer is required. In conjunction with the FX-Control PC software, the FFT can be easily and quickly adapted and visualized according to the requirements of the measurement. The larger internal memory of the FLEXUS FX100 allows significantly longer blocklengths to be processed, resulting in a much finer frequency resolution.

FFT Further Considerations

This second part of this article deals with specific aspects that are helpful in the practical application of FFT measurements. FFT measurements are used in numerous applications. The results are usually presented as graphs and are easy to interpret. For accurate FFT measurements, there are some things to look out for. This article provides valuable tips.

As explained in the first part, the sampling rate fs of the measuring system and the block length BL are the two central parameters of an FFT. The sampling rate indicates how often the analog signal to be analyzed is scanned. When recording wav files via a commercially-available PC sound card, for example, the audio signal is usually sampled 44,100 times per second.

Nyquist Theorem

Harry Nyquist was the discoverer of a fundamental rule in the sampling of analog signals: the sampling frequency must be at least double the highest frequency of the signal. If, for example, a signal containing frequencies up to 24 kHz is to be sampled, a sampling rate of at least 48 kHz is required for this purpose. Half the sampling rate, in this example 24 kHz, is called the "Nyquist frequency".

But what happens if signals above the Nyquist frequency are fed in to the system?

Aliasing

For the most, a signal is sampled with a more-than-sufficient number of samples. With a 48 kHz sampling rate, for example, the 6 kHz frequency is sampled 8 times per cycle, while the 12 kHz frequency is only sampled 4 times per cycle. At the Nyquist frequency, only 2 samples are available per cycle.

With 2 samples or more it is still possible to reconstruct the signal without loss. If, however, less than 2 samples are available, artifacts which do not occur in the sampled (original) signal are generated.

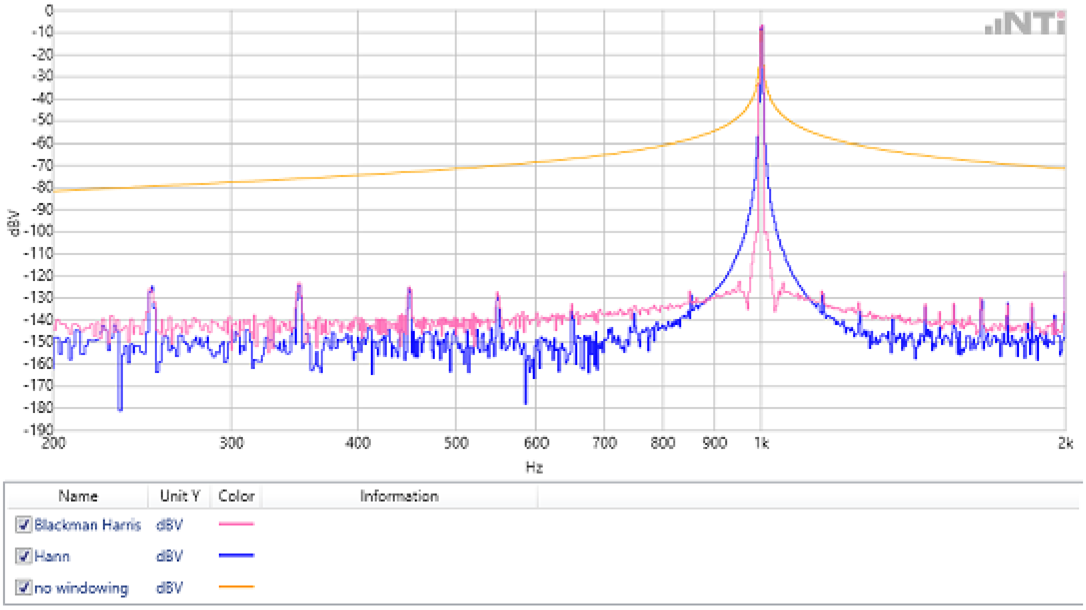
Mirror frequencies

In the FFT, these artifacts appear as mirror frequencies. If the Nyquist frequency is exceeded, the signal is reflected at this imaginary limit and falls back into the useful frequency band. The following video shows an FFT system with 44.1 kHz sampling rate. A sweep signal of 15 kHz to 25 kHz is fed in to this system.

These unwanted mirror frequencies are counteracted with an analog low-pass filter (anti-aliasing filter) before the scanning. The filter ensures that frequencies above the Nyquist frequency are suppressed.

Time window

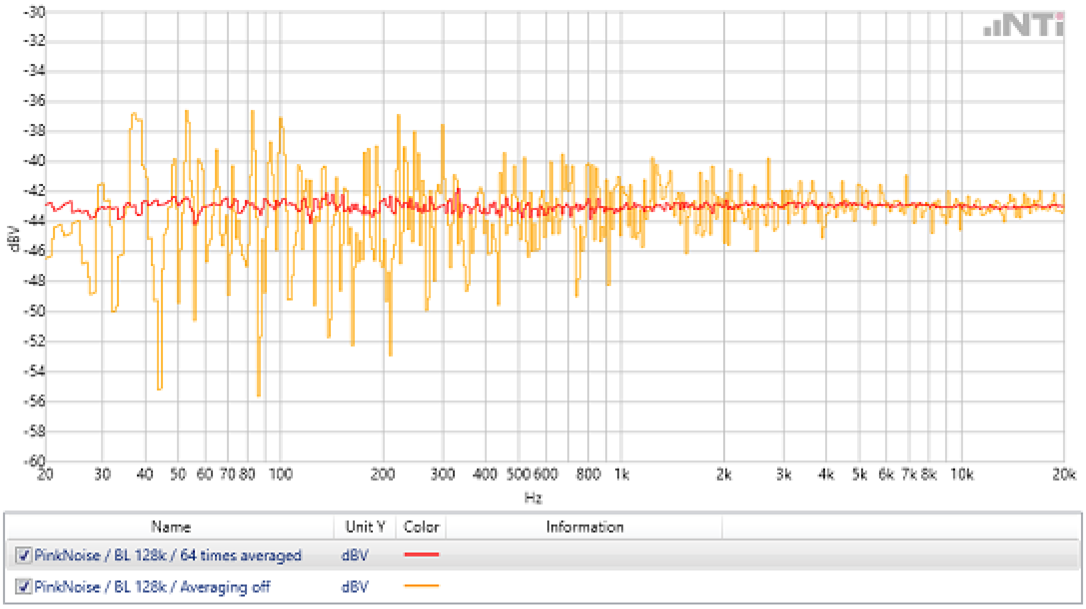
In the case of periodically-continuous signals, the time windowing serves to smooth the undesired transitional jumps at the end of the scanning (see part 1). This prevents smearing in the spectrum. There are numerous types of windows, some of which differ only slightly. When selecting the time window, the following rule applies: Each window requires a compromise between frequency selectivity and amplitude accuracy.



Averaging of Spectra

In the analysis of non-periodic signals, e.g. noise or music, it is often advantageous to capture multiple FFT blocks and determine mean values therefrom. There are two possible approaches:

1. The classical mean: A number of FFTs are measured. Each result is considered in equal parts in the averaged final result. This method is suitable for measurements with a defined duration.
2. The exponential mean: FFTs are continuously measured. Here, too, a fixed number of results of the continuous measurements are considered. However, the weighting is inversely proportional to the 'age' of the result. The oldest of the measurements is taken the least into account, the most recent measurement contributes most effectively to the averaged result. This exponential average is used when the spectrum is continuously monitored over a long period of time.



Power vs. Peak detector

Modern high-resolution FFT analyzers offer the possibility to decouple the number of measurement results from the FFT block length. This results in an increase in measurement performance time, especially for high-resolution FFTs. Thus, for example, with a 2MB block length it is no longer necessary to measure and represent more than 1 Million points (bins), but only the number necessary for the display, e.g. 1024.

The value chosen for each FFT bin can be defined in two ways:

1. "MaxPeak": Here the maximum value of the FFT results is used. This type is well suited for the visual representation of FFTs
2. "Power": Here the FFT results are summed up and averaged energetically. This is necessary when the FFT is used for calculations.

Calculations with FFT results

FFTs are mainly used to visualize signals. However, there are also applications where FFT results are used in calculations. For example, very simple levels of defined frequency bands can be calculated by adding them via an RSS (Root Sum Square) algorithm.

Another application is the comparison of spectra. The example below shows an acoustic measurement of a cordless screwdriver. The measured spectrum is subtracted from a defined reference spectrum. This difference is compared against an upper and lower tolerance. The upper spectrum shows a functional cordless screwdriver. In the lower, the acoustic spectrum suggests that the test specimen is defective.

Theory: Compute and plot the FFT of a signal.
MATLAB Code:

```
% Fast Fourier Transform (FFT)
Fs = 1000; % Sampling frequency
t = 0:1/Fs:1-1/Fs;
f = 5; % Frequency of sine wave
x = sin(2*pi*f*t);
X = fft(x);
n = length(x);
f = (0:n-1)*(Fs/n); % Frequency range
magnitude = abs(X)/n;
plot(f, magnitude);
title('FFT of Sine Wave');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
```

19. Bandpass Filter Design

Passive Band Pass Filter

Passive Band Pass Filters can be made by connecting together a low pass filter with a high pass filter

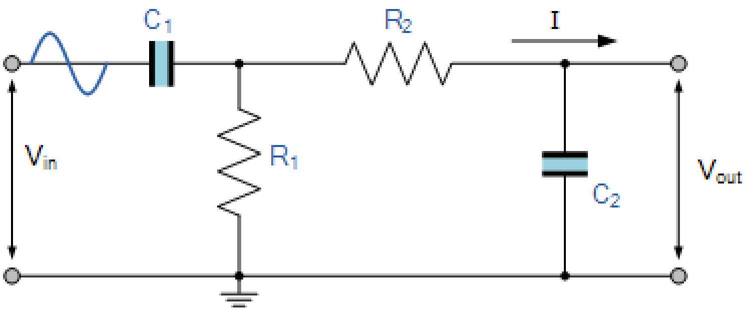
The **Passive Band Pass Filter** can be used to isolate or filter out certain frequencies that lie within a particular band or range of frequencies. The cut-off frequency or fc point in a simple RC passive filter can be accurately controlled using just a single resistor in series with a non-polarized capacitor, and depending upon which way around they are connected, we have seen that either a Low Pass or a High Pass filter is obtained.

One simple use for these types of passive filters is in audio amplifier applications or circuits such as in loudspeaker crossover filters or pre-amplifier tone controls. Sometimes it is necessary to only pass a certain range of frequencies that do not begin at 0Hz, (DC) or end at some upper high frequency point but are within a certain range or band of frequencies, either narrow or wide.

By connecting or "cascading" together a single *Low Pass Filter* circuit with a *High Pass Filter* circuit, we can produce another type of passive RC filter that passes a selected range or "band" of frequencies that can be either narrow or wide while attenuating all those outside of this range.

This new type of passive filter arrangement produces a frequency selective filter known commonly as a **Band Pass Filter** or **BPF** for short.

Typical Band Pass Filter Circuit



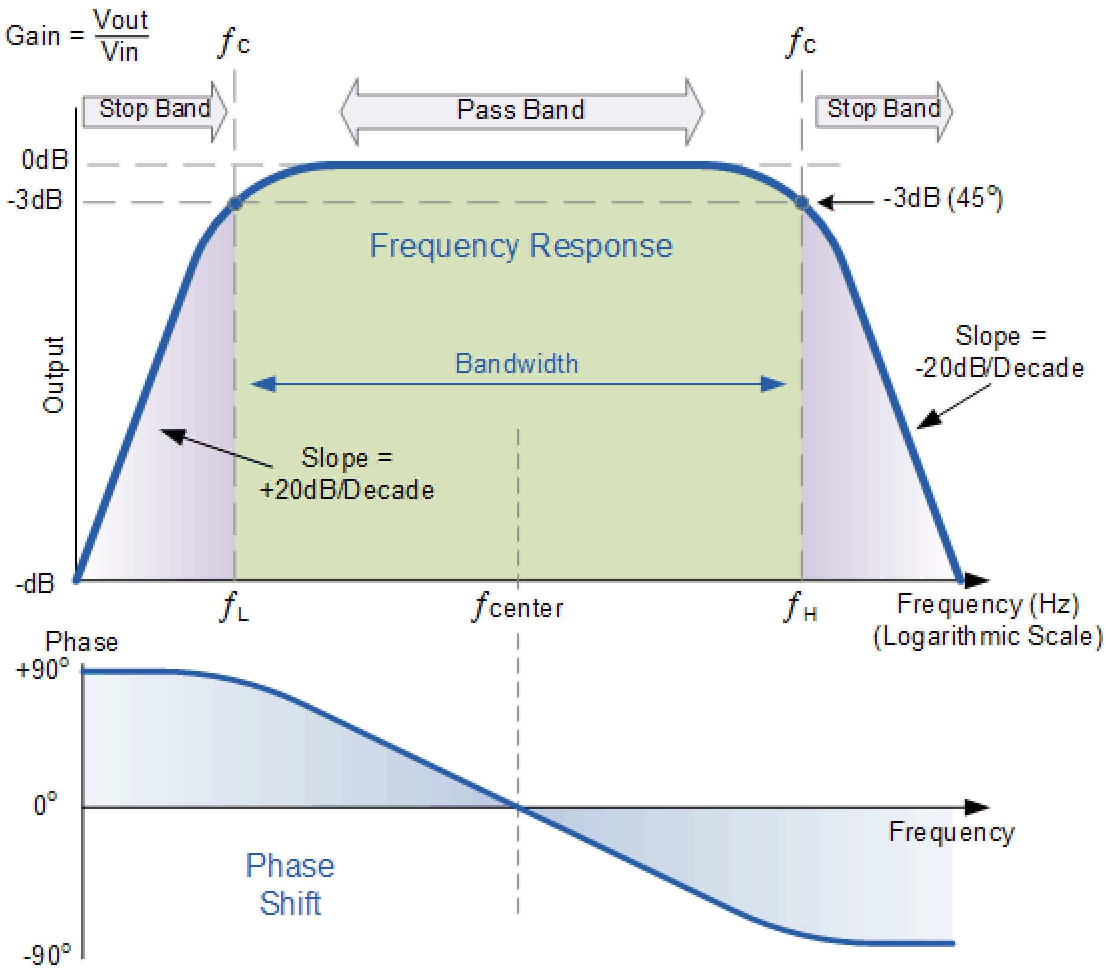
Unlike the low pass filter which only pass signals of a low frequency range or the high pass filter which pass signals of a higher frequency range, a **Band Pass Filters** passes signals within a certain "band" or "spread" of frequencies without distorting the input signal or introducing extra noise. This band of frequencies can be any width and is commonly known as the filters **Bandwidth**.

Bandwidth is commonly defined as the frequency range that exists between two specified frequency cut-off points (fc), that are 3dB below the maximum centre or resonant peak while attenuating or weakening the others outside of these two points.

Then for widely spread frequencies, we can simply define the term "bandwidth", BW as being the difference between the lower cut-off frequency (fc_{LOWER}) and the higher cut-off frequency (fc_{HIGHER}) points. In other words, $BW = f_H - f_L$. Clearly for a pass band filter to function correctly, the cut-off frequency of the low pass filter must be higher than the cut-off frequency for the high pass filter.

The "ideal" **Band Pass Filter** can also be used to isolate or filter out certain frequencies that lie within a particular band of frequencies, for example, noise cancellation. Band pass filters are known generally as second-order filters, (two-pole) because they have "two" reactive component, the capacitors, within their circuit design. One capacitor in the low pass circuit and another capacitor in the high pass circuit.

Frequency Response of a 2nd Order Band Pass Filter



The **Bode Plot** or frequency response curve above shows the characteristics of the band pass filter. Here the signal is attenuated at low frequencies with the output increasing at a slope of +20dB/Decade (6dB/Octave) until the frequency reaches the "lower cut-off" point f_L. At this frequency the output voltage is again $1/\sqrt{2}$ = 70.7% of the input signal value or **-3dB** ($20*\log(V_{OUT}/V_{IN})$) of the input.

The output continues at maximum gain until it reaches the "upper cut-off" point f_H where the output decreases at a rate of -20dB/Decade (6dB/Octave) attenuating any high frequency signals. The point of maximum output gain is generally the geometric mean of the two -3dB value between the lower and upper cut-off points and is called the "Centre Frequency" or "Resonant Peak" value fr. This geometric mean value is calculated as being $fr^2 = f_{(UPPER)} \times f_{(LOWER)}$.

A band pass filter is regarded as a second-order (two-pole) type filter because it has "two" reactive components within its circuit structure, then the phase angle will be twice that of the previously seen first-order filters, ie, **180°**.

The phase angle of the output signal **LEADS** that of the input by **+90°** up to the centre or resonant frequency, fr point were it becomes "zero" degrees (0°) or "in-phase" and then changes to **LAG** the input by **-90°** as the output frequency increases.

The upper and lower cut-off frequency points for a band pass filter can be found using the same formula as that for both the low and high pass filters, For example.

$$fc = \frac{1}{2\pi RC}\ Hz$$

Then clearly, the width of the pass band of the filter can be controlled by the positioning of the two cut-off frequency points of the two filters.

Band Pass Filter Example No1.

A second-order **band pass filter** is to be constructed using RC components that will only allow a range of frequencies to pass above 1kHz (1,000Hz) and below 30kHz (30,000Hz). Assuming that both the resistors have values of 10kΩ, calculate the values of the two capacitors required.

The High Pass Filter Stage

The value of the capacitor C1 required to give a cut-off frequency f_L of 1kHz with a resistor value of 10kΩ is calculated as:

$$C_1 = \frac{1}{2\pi f_L R} = \frac{1}{2\pi x 1{,}000 x 10{,}000} = 15.9\,nF$$

Then, the values of R1 and C1 required for the high pass stage to give a cut-off frequency of 1.0kHz are: R1 = 10kΩ and to the nearest preferred value, C1 = 15nF.

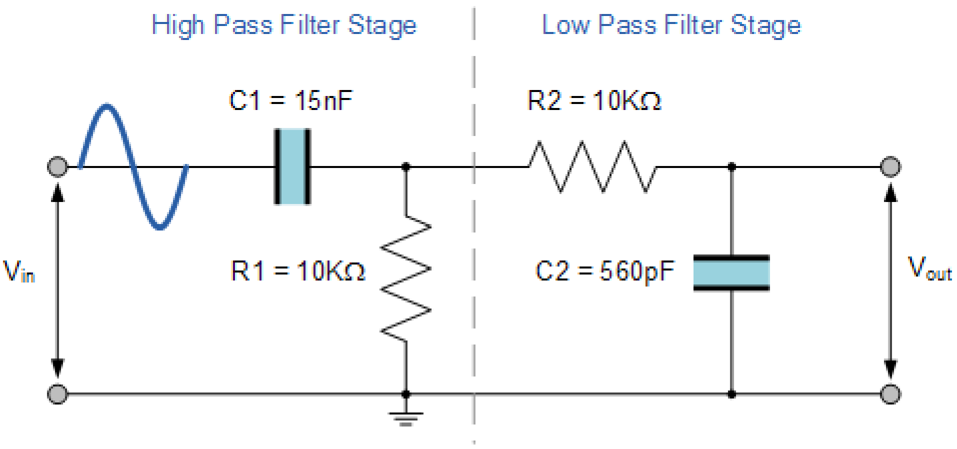
The Low Pass Filter Stage

The value of the capacitor C2 required to give a cut-off frequency f_H of 30kHz with a resistor value of 10kΩ is calculated as:

$$C_2 = \frac{1}{2\pi f_H R} = \frac{1}{2\pi x 30{,}000 x 10{,}000} = 530\,pF$$

Then, the values of R2 and C2 required for the low pass stage to give a cut-off frequency of 30kHz are, R = 10kΩ and C = 530pF. However, the nearest preferred value of the calculated capacitor value of 530pF is 560pF, so this is used instead.
With the values of both the resistances R1 and R2 given as 10kΩ, and the two values of the capacitors C1 and C2 found for both the high pass and low pass filters as 15nF and 560pF respectively, then the circuit for our simple passive **Band Pass Filter** is given as.

Completed Band Pass Filter Circuit



The Filters Resonant Frequency

We can also calculate the "Resonant" or "Centre Frequency" (fr) point of the band pass filter were the output gain is at its maximum or peak value. This peak value is not the arithmetic average of the upper and lower -3dB cut-off points as you might expect but is in fact the "geometric" or mean value. This geometric mean value is calculated as being $fr^2 = fc_{(UPPER)} \times fc_{(LOWER)}$ for example:

Centre Frequency Equation

$$fr = \sqrt{f_L \ x \ f_H}$$

- Where, f_r is the resonant or centre frequency
- f_L is the lower -3dB cut-off frequency point
- f_H is the upper -3db cut-off frequency point

and in our simple example above, the calculated cut-off frequencies were found to be f_L = 1,060 Hz and f_H = 28,420 Hz using the filter values.
Then by substituting these values into the above equation gives a central resonant frequency of:

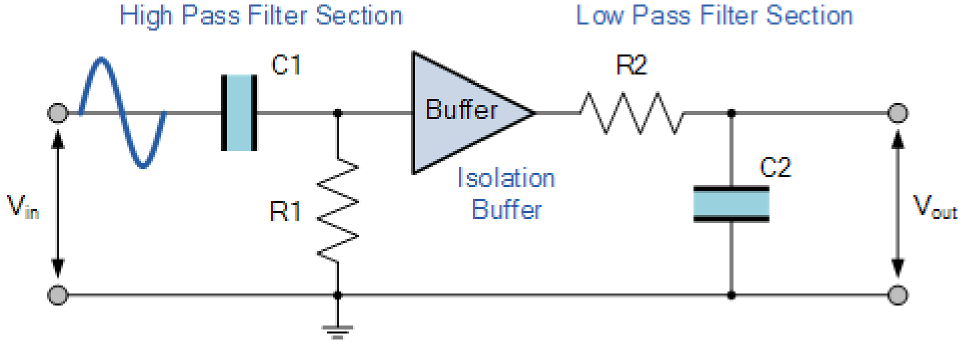
$$fr = \sqrt{f_L \ x \ f_H} = \sqrt{1{,}060 \ x \ 28{,}420} = 5{,}48 \, kHz$$

Band Pass Summary

A simple passive **Band Pass Filter** can be made by cascading together a single **Low Pass Filter** with a **High Pass Filter**. The frequency range, in Hertz, between the lower and upper -3dB cut-off points of the RC combination is know as the filters "Bandwidth".
The width or frequency range of the filters bandwidth can be very small and selective, or very wide and non-selective depending upon the values of R and C used.

The centre or resonant frequency point is the geometric mean of the lower and upper cut-off points. At this centre frequency the output signal is at its maximum and the phase shift of the output signal is the same as the input signal.
The amplitude of the output signal from a band pass filter or any passive RC filter for that matter, will always be less than that of the input signal. In other words a passive filter is also an attenuator giving a voltage gain of less than 1 (Unity). To provide an output signal with a voltage gain greater than unity, some form of amplification is required within the design of the circuit.
A **Passive Band Pass Filter** is classed as a second-order type filter because it has two reactive components within its design, the capacitors. It is made up from two single RC filter circuits that are each first-order filters themselves.
If more filters are cascaded together the resulting circuit will be known as an "n^{th}-order" filter where the "n" stands for the number of individual reactive components and therefore poles within the filter circuit. For example, filters can be a 2^{nd}-order, 4^{th}-order, 10^{th}-order, etc.
The higher the filters order the steeper will be the slope at n times -20dB/decade. However, a single capacitor value made by combining together two or more individual capacitors is still one capacitor.
Our example above shows the output frequency response curve for an "ideal" band pass filter with constant gain in the pass band and zero gain in the stop bands. In practice the frequency response of this Band Pass Filter circuit would not be the same as the input reactance of the high pass circuit would affect the frequency response of the low pass circuit (components connected in series or parallel) and vice versa. One way of overcoming this would be to provide some form of electrical isolation between the two filter circuits as shown below.

Buffering Individual Filter Stages



One way of combining amplification and filtering into the same circuit would be to use an Operational Amplifier or Op-amp, and examples of these are given in the Operational Amplifier section.
In the next tutorial we will look at filter circuits which use an operational amplifier within their design to not only to introduce gain but provide isolation between stages. These types of filter arrangements are generally known as **Active Filters**.

Theory: Design and simulate a bandpass filter.
MATLAB Code:

```
% Bandpass Filter Design
```

```
Fs = 1000; % Sampling frequency
Fpass1 = 100; % First passband frequency
Fpass2 = 200; % Second passband frequency
N = 50; % Filter order
bpFilt = designfilt('bandpassfir', 'FilterOrder', N, ...
   'CutoffFrequency1', Fpass1, 'CutoffFrequency2', Fpass2, ...
   'SampleRate', Fs);
fvtool(bpFilt);
title('Bandpass Filter Design');
```

20. Operational Amplifier (Op-Amp) Inverting Amplifier

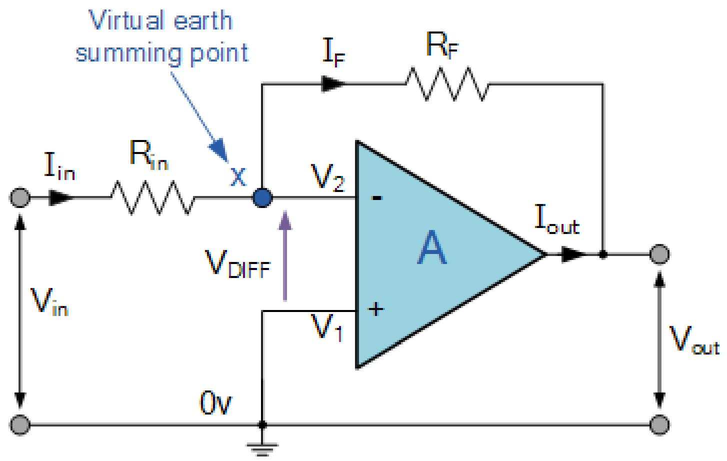
Theory: Analyze the behavior of an inverting amplifier using an operational amplifier.

Inverting Operational Amplifier

The Inverting Operational Amplifier configuration is one of the simplest and most commonly used op-amp topologies
The inverting operational amplifier is basically a constant or fixed-gain amplifier producing a negative output voltage as its gain is always negative.
We saw in the last tutorial that the **Open Loop Gain**, (A_{VO}) of an operational amplifier can be very high, as much as 1,000,000 (120dB) or more.
However, this very high gain is of no real use to us as it makes the amplifier both unstable and hard to control as the smallest of input signals, just a few micro-volts, (μV) would be enough to cause the output voltage to saturate and swing towards one or the other of the voltage supply rails losing complete control of the output.
As the open loop DC gain of an operational amplifier is extremely high we can therefore afford to lose some of this high gain by connecting a suitable resistor across the amplifier from the output terminal back to the inverting input terminal to both reduce and control the overall gain of the amplifier. This then produces and effect known commonly as Negative Feedback, and thus produces a very stable Operational Amplifier based system.
Negative Feedback is the process of "feeding back" a fraction of the output signal back to the input, but to make the feedback negative, we must feed it back to the negative or "inverting input" terminal of the op-amp using an external **Feedback Resistor** called R_f.
This feedback connection between the output and the inverting input terminal forces the differential input voltage towards zero.
This effect produces a closed loop circuit to the amplifier resulting in the gain of the amplifier now being called its **Closed-loop Gain**. Then a closed-loop inverting amplifier uses negative feedback to accurately control the overall gain of the amplifier, but at a cost in the reduction of the amplifiers gain.
This negative feedback results in the inverting input terminal having a different signal on it than the actual input voltage as it will be the sum of the input voltage plus the negative feedback voltage giving it the label or term of a *Summing Point*. We must therefore separate the real input signal from the inverting input by using an **Input Resistor**, Rin.
As we are not using the positive non-inverting input this is connected to a common ground or zero voltage terminal as shown below, but the effect of this closed loop feedback circuit results in the voltage potential at the inverting input being equal to that at the non-inverting

input producing a *Virtual Earth* summing point because it will be at the same potential as the grounded reference input. In other words, the op-amp becomes a "differential amplifier".

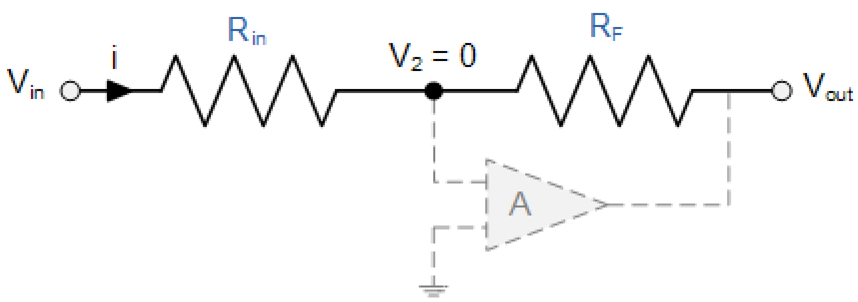
Inverting Operational Amplifier Configuration



In this **Inverting Amplifier** circuit the operational amplifier is connected with feedback to produce a closed loop operation. When dealing with operational amplifiers there are two very important rules to remember about inverting amplifiers, these are: "No current flows into the input terminal" and that "V1 always equals V2". However, in real world op-amp circuits both of these rules are slightly broken.
This is because the junction of the input and feedback signal (X) is at the same potential as the positive (+) input which is at zero volts or ground then, the junction is a **"Virtual Earth"**. Because of this virtual earth node the input resistance of the amplifier is equal to the value of the input resistor, Rin and the closed loop gain of the inverting amplifier can be set by the ratio of the two external resistors.
We said above that there are two very important rules to remember about **Inverting Amplifiers** or any operational amplifier for that matter and these are.

- No Current Flows into the Input Terminals
- The Differential Input Voltage is Zero as V1 = V2 = 0 (Virtual Earth)

Then by using these two rules we can derive the equation for calculating the closed-loop gain of an inverting amplifier, using first principles.
Current (i) flows through the resistor network as shown.

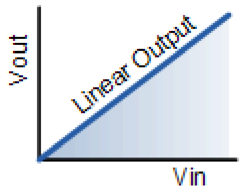


Then, the **Closed-Loop Voltage Gain** of an Inverting Amplifier is given as.

$$\text{Gain}\,(Av) = \frac{V_{out}}{V_{in}} = -\frac{R_f}{R_{in}}$$

and this can be transposed to give Vout as:

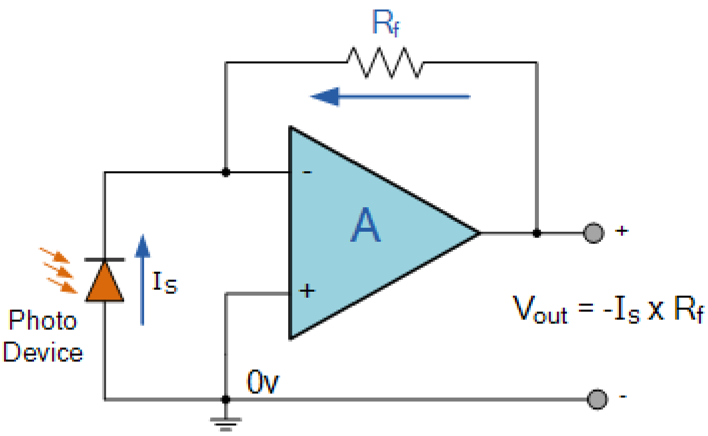
$$Vout = -\frac{Rf}{Rin} \times Vin$$



Linear Output

The negative sign in the equation indicates an inversion of the output signal with respect to the input as it is 180° out of phase. This is due to the feedback being negative in value.
The equation for the output voltage Vout also shows that the circuit is linear in nature for a fixed amplifier gain as Vout = Vin x Gain. This property can be very useful for converting a smaller sensor signal to a much larger voltage.
Another useful application of an inverting amplifier is that of a "transresistance amplifier" circuit. A **Transresistance Amplifier** also known as a "transimpedance amplifier", is basically a current-to-voltage converter (Current "in" and Voltage "out"). They can be used in low-power applications to convert a very small current generated by a photo-diode or photo-detecting device etc, into a usable output voltage which is proportional to the input current as shown.

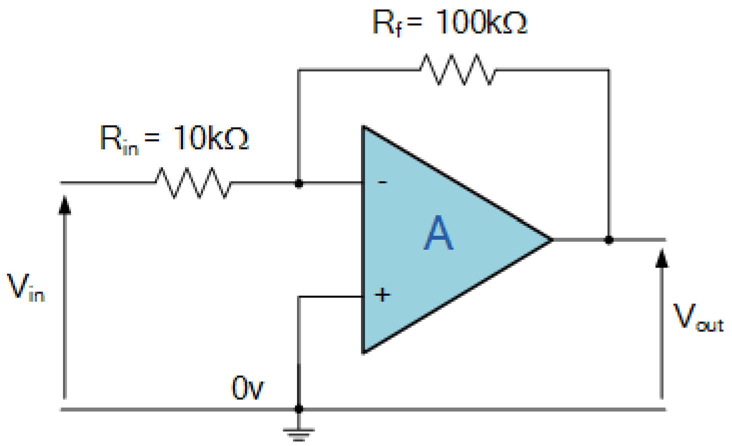
Transresistance Amplifier Circuit



The simple light-activated circuit above, converts a current generated by the photo-diode into a voltage. The feedback resistor Rf sets the operating voltage point at the inverting input and controls the amount of output. The output voltage is given as Vout = I_s x Rf. Therefore, the output voltage is proportional to the amount of input current generated by the photo-diode.

Inverting Operational Amplifier Example No1

Find the closed loop gain of the following inverting amplifier circuit.



Using the previously found formula for the gain of the circuit

$$\text{Gain (Av)} = \frac{V_{out}}{V_{in}} = -\frac{R_f}{R_{in}}$$

we can now substitute the values of the resistors in the circuit as follows,

Rin = 10kΩ and Rf = 100kΩ

and the gain of the circuit is calculated as: -Rf/Rin = 100k/10k = -10

Therefore, the closed loop gain of the inverting amplifier circuit above is given **-10** or **20dB** (20log(10)).

Inverting Operational Amplifier Example No2

The gain of the original circuit is to be increased to **40** (32dB), find the new values of the resistors required.
Assuming that the input resistor is to remain at the same value of 10KΩ, then by re-arranging the closed loop voltage gain formula we can find the new value required for the feedback resistor Rf.

Gain = Rf/Rin

therefore,   Rf = Gain x Rin

Rf = 40 x 10,000

Rf = 400,000 or 400KΩ

The new values of resistors required for the circuit to have a gain of **40** would be:

Rin = 10KΩ and Rf = 400KΩ

The formula could also be rearranged to give a new value of Rin, keeping the same value of Rf.
One final point to note about the **Inverting Amplifier** configuration for an operational amplifier, if the two resistors are of equal value, Rin = Rf then the gain of the amplifier will be **-1** producing a complementary form of the input voltage at its output as Vout = -Vin. This type of inverting amplifier configuration is generally called a **Unity Gain Inverter** of simply an *Inverting Buffer*.
In the next tutorial about Operational Amplifiers, we will analyse the complement of the **Inverting Amplifier** operational amplifier circuit called the Non-inverting Amplifier that produces an output signal which is "in-phase" with the input.

MATLAB Code:

```
% Op-Amp Inverting Amplifier
Rin = 10e3; % Input resistor in ohms
Rf = 100e3; % Feedback resistor in ohms
Av = -Rf/Rin; % Voltage gain
Vin = 1; % Input voltage in volts
Vout = Av * Vin; % Output voltage
disp(['Output Voltage: ', num2str(Vout), ' V']);
```

21. Butterworth Low Pass Filter Design

Butterworth Filter Design

In the previous filter tutorials we looked at simple first-order type low and high pass filters that contain only one single resistor and a single reactive component (a capacitor) within their RC filter circuit design.
The Butterworth filter is an analogue filter design which produces the best output response with no ripple in the pass band or the stop band resulting in a maximally flat filter response but at the expense of a relatively wide transition band.
In applications that use filters to shape the frequency spectrum of a signal such as in communications or control systems, the shape or width of the roll-off also called the "transition band". For simple first-order filters this transition band maybe too long or too wide, so active filters designed with more than one "order" are required. These types of filters are commonly known as "High-order" or "nth-order" filters.
The complexity or filter type is defined by the filters "order", and which is dependant upon the number of reactive components such as capacitors or inductors within its design. We also

know that the rate of roll-off and therefore the width of the transition band, depends upon the order number of the filter and that for a simple first-order filter it has a standard roll-off rate of 20dB/decade or 6dB/octave.

Then, for a filter that has an n^{th} number order, it will have a subsequent roll-off rate of 20n dB/decade or 6n dB/octave. So a first-order filter has a roll-off rate of 20dB/decade (6dB/octave), a second-order filter has a roll-off rate of 40dB/decade (12dB/octave), and a fourth-order filter has a roll-off rate of 80dB/decade (24dB/octave), etc, etc.

High-order filters, such as third, fourth, and fifth-order are usually formed by cascading together single first-order and second-order filters.

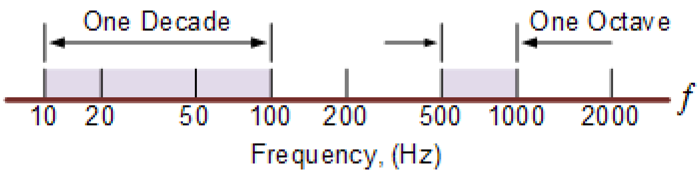
For example, two second-order low pass filters can be cascaded together to produce a fourth-order low pass filter, and so on. Although there is no limit to the order of the filter that can be formed, as the order increases so does its size and cost, also its accuracy declines.

Decades and Octaves

One final comment about *Decades* and *Octaves*. On the frequency scale, a **Decade** is a tenfold increase (multiply by 10) or tenfold decrease (divide by 10). For example, 2 to 20Hz represents one decade, whereas 50 to 5000Hz represents two decades (50 to 500Hz and then 500 to 5000Hz).

An **Octave** is a doubling (multiply by 2) or halving (divide by 2) of the frequency scale. For example, 10 to 20Hz represents one octave, while 2 to 16Hz is three octaves (2 to 4, 4 to 8 and finally 8 to 16Hz) doubling the frequency each time. Either way, *Logarithmic* scales are used extensively in the frequency domain to denote a frequency value when working with amplifiers and filters so it is important to understand them.

Logarithmic Frequency Scale



Since the frequency determining resistors are all equal, and as are the frequency determining capacitors, the cut-off or corner frequency (f_C) for either a first, second, third or even a fourth-order filter must also be equal and is found by using our now old familiar equation:

$$f_C = \frac{1}{2\pi RC}\ Hz$$

As with the first and second-order filters, the third and fourth-order high pass filters are formed by simply interchanging the positions of the frequency determining components (resistors and capacitors) in the equivalent low pass filter. High-order filters can be designed by following the procedures we saw previously in the Low Pass filter and High Pass filter tutorials. However, the overall gain of high-order filters is **fixed** because all the frequency determining components are equal.

Filter Approximations

So far we have looked at a low and high pass first-order filter circuits, their resultant frequency and phase responses. An ideal filter would give us specifications of maximum pass band gain and flatness, minimum stop band attenuation and also a very steep pass band to stop band roll-off (the transition band) and it is therefore apparent that a large number of network responses would satisfy these requirements.

Not surprisingly then that there are a number of “approximation functions” in linear analogue filter design that use a mathematical approach to best approximate the transfer function we require for the filters design.

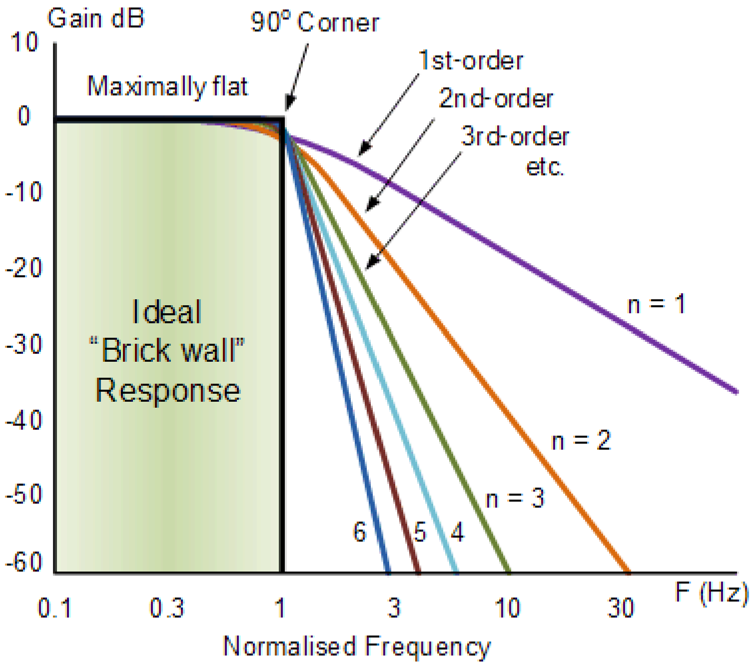
Such designs are known as **Elliptical**, **Butterworth**, **Chebyshev**, **Bessel**, **Cauer** as well as many others. Of these five “classic” linear analogue filter approximation functions only the **Butterworth Filter** and especially the *low pass Butterworth filter* design will be considered here as its the most commonly used function.

Low Pass Butterworth Filter Design

The frequency response of the **Butterworth Filter** approximation function is also often referred to as “maximally flat” (no ripples) response because the pass band is designed to have a frequency response which is as flat as mathematically possible from 0Hz (DC) until the cut-off frequency at -3dB with no ripples. Higher frequencies beyond the cut-off point rolls-off down to zero in the stop band at 20dB/decade or 6dB/octave. This is because it has a “quality factor”, “Q” of just 0.707.

However, one main disadvantage of the Butterworth filter is that it achieves this pass band flatness at the expense of a wide transition band as the filter changes from the pass band to the stop band. It also has poor phase characteristics as well. The ideal frequency response, referred to as a “brick wall” filter, and the standard Butterworth approximations, for different filter orders are given below.

Ideal Frequency Response for a Butterworth Filter



Note that the higher the Butterworth filter order, the higher the number of cascaded stages there are within the filter design, and the closer the filter becomes to the ideal "brick wall" response.
In practice however, Butterworth's ideal frequency response is unattainable as it produces excessive passband ripple.

Where the generalised equation representing a "nth" Order Butterworth filter, the frequency response is given as:

$$H_{(j\omega)} = \frac{1}{\sqrt{1+\varepsilon^2\left(\frac{\omega}{\omega_p}\right)^{2n}}}$$

Where: n represents the filter order, Omega ω is equal to $2\pi f$ and Epsilon ε is the maximum pass band gain, (A_{max}). If A_{max} is defined at a frequency equal to the cut-off -3dB corner point (fc), ε will then be equal to one and therefore ε^2 will also be one. However, if you now wish to define A_{max} at a different voltage gain value, for example 1dB, or 1.1220 (1dB = 20*logA_{max}) then the new value of epsilon, ε is found by:

$$H_1 = \frac{H_0}{\sqrt{1+\varepsilon^2}}$$

- Where:
- H_0 = the Maximum Pass band Gain, A_{max}.
- H_1 = the Minimum Pass band Gain.

Transpose the equation to give:

$$\frac{H_0}{H_1} = 1.1220 = \sqrt{1+\varepsilon^2} \quad gives \quad \varepsilon = 0.5088$$

The **Frequency Response** of a filter can be defined mathematically by its **Transfer Function** with the standard Voltage Transfer Function H(jω) written as:

$$H(j\omega) = \left[\frac{Vout(j\omega)}{Vin(j\omega)}\right]$$

- Where:
- Vout = the output signal voltage.
- Vin = the input signal voltage.
- j = to the square root of -1 (√-1)
- ω = the radian frequency ($2\pi f$)

Note: (jω) can also be written as (s) to denote the **S-domain.** and the resultant transfer function for a second-order low pass filter is given as:

$$H(s) = \frac{Vout}{Vin} = \frac{1}{S^2+S+1}$$

Normalised Low Pass Butterworth Filter Polynomials

To help in the design of his low pass filters, Butterworth produced standard tables of normalised second-order low pass polynomials given the values of coefficient that correspond to a cut-off corner frequency of 1 radian/sec.

n	**Normalised Denominator Polynomials in Factored Form**
1	$(1+s)$
2	$(1+1.414s+s^2)$
3	$(1+s)(1+s+s^2)$
4	$(1+0.765s+s^2)(1+1.848s+s^2)$
5	$(1+s)(1+0.618s+s^2)(1+1.618s+s^2)$
6	$(1+0.518s+s^2)(1+1.414s+s^2)(1+1.932s+s^2)$
7	$(1+s)(1+0.445s+s^2)(1+1.247s+s^2)(1+1.802s+s^2)$
8	$(1+0.390s+s^2)(1+1.111s+s^2)(1+1.663s+s^2)(1+1.962s+s^2)$
9	$(1+s)(1+0.347s+s^2)(1+s+s^2)(1+1.532s+s^2)(1+1.879s+s^2)$
10	$(1+0.313s+s^2)(1+0.908s+s^2)(1+1.414s+s^2)(1+1.782s+s^2)(1+1.975s+s^2)$

Filter Design – Butterworth Low Pass

Find the order of an active low pass Butterworth filter whose specifications are given as: A_{max} = 0.5dB at a pass band frequency (ωp) of 200 radian/sec (31.8Hz), and A_{min} = -20dB at a stop band frequency (ωs) of 800 radian/sec. Also design a suitable Butterworth filter circuit to match these requirements.

Firstly, the maximum pass band gain A_{max} = 0.5dB which is equal to a gain of **1.0593**, remember that: 0.5dB = 20*log(A) at a frequency (ωp) of 200 rads/s, so the value of epsilon ε is found by:

$$1.0593 = \sqrt{1 + \varepsilon^2}$$

$$\therefore \ \varepsilon = 0.3495 \ \ \textit{and} \ \ \varepsilon^2 = 0.1221$$

Secondly, the minimum stop band gain A_{min} = -20dB which is equal to a gain of **10** (-20dB = 20*log(A)) at a stop band frequency (ωs) of 800 rads/s or 127.3Hz.

Substituting the values into the general equation for a Butterworth filters frequency response gives us the following:

$$H(j\omega) = \frac{H_0}{\sqrt{1 + \varepsilon^2 \left(\frac{\omega_S}{\omega_P}\right)^{2n}}}$$

$$\frac{1}{10} = \frac{1}{\sqrt{1 + 0.1221 \left(\frac{800}{200}\right)^{2n}}}$$

$$(10)^2 = 1 + 0.1221 \; x \; 4^{2n}$$

$$\therefore 100 - 1 = 0.1221 \; x \; 4^{2n}$$

$$4^{2n} = \frac{99}{0.1221} = 810.811$$

$$4^n = \sqrt{810.811} = 28.475$$

$$\therefore n = \frac{\log 28.475}{\log 4} = 2.42$$

Since n must always be an integer (whole number) then the next highest value to 2.42 is n = 3, therefore a **"a third-order filter is required"** and to produce a third-order **Butterworth filter**, a second-order filter stage cascaded together with a first-order filter stage is required. From the normalised low pass Butterworth Polynomials table above, the coefficient for a third-order filter is given as $(1+s)(1+s+s^2)$ and this gives us a gain of 3-A = 1, or A = 2. As A = 1 + (Rf/R1), choosing a value for both the feedback resistor Rf and resistor R1 gives us values of 1kΩ and 1kΩ respectively as: (1kΩ/1kΩ) + 1 = 2.
We know that the cut-off corner frequency, the -3dB point (ω_o) can be found using the formula 1/CR, but we need to find ω_o from the pass band frequency ω_p then,

$$H(j\omega) = \frac{H_0}{\sqrt{1 + \varepsilon^2 \left(\frac{\omega_o}{\omega_p}\right)^{2n}}}$$

$$3dB = 1.414 \ at \ \omega = \omega_o$$

$$\frac{1}{1.414} = \frac{1}{\sqrt{1 + \varepsilon^2 \left(\frac{\omega_o}{\omega_p}\right)^{2n}}}$$

$$2 = 1 + \varepsilon^2 \left(\frac{\omega_o}{\omega_p}\right)^{2n}$$

$$\therefore 1 = \varepsilon \left(\frac{\omega_o}{\omega_p}\right)^{n}$$

$$\omega_o{}^n = \frac{\omega_p{}^n}{\varepsilon}$$

$$\omega_o{}^3 = \frac{200^3}{0.3495}$$

$$\omega_o{}^3 = 22.889 \ x \ 10^6$$

$$\therefore \omega_o = 283.93 \simeq 284 \, rads/s$$

So, the cut-off corner frequency is given as 284 rads/s or 45.2Hz, (284/2π) and using the familiar formula 1/CR we can find the values of the resistors and capacitors for our third-order circuit.

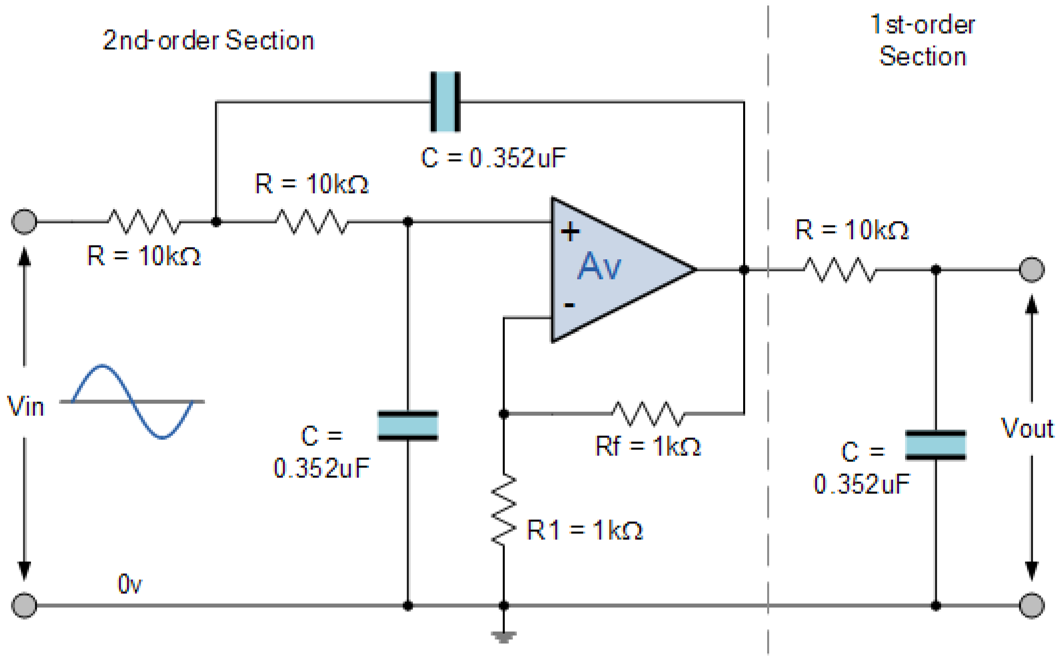
$$284 rads/s = \frac{1}{CR} \quad use\, a\, value\, of\, R = 10k\Omega$$

$$\therefore\ Capacitor\ C = \frac{1}{284 \times 10{,}000} = 0.352uF$$

Note that the nearest preferred value to 0.352uF would be 0.36uF, or 360nF.

Third-order Butterworth Low Pass Filter

and finally our circuit of the third-order low pass **Butterworth Filter** with a cut-off corner frequency of 284 rads/s or 45.2Hz, a maximum pass band gain of 0.5dB and a minimum stop band gain of 20dB is constructed as follows.



So for our 3rd-order Butterworth Low Pass Filter with a corner frequency of 45.2Hz, C = 360nF and R = 10kΩ

Theory: Design and simulate a Butterworth low pass filter.
MATLAB Code:

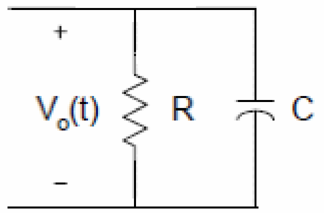
```
% Butterworth Low Pass Filter Design
Fs = 1000; % Sampling frequency
Fc = 200; % Cutoff frequency
[b, a] = butter(4, Fc/(Fs/2)); % 4th order Butterworth filter
freqz(b, a, [], Fs);
```

title('Butterworth Low Pass Filter');

22. Transient Analysis of RC Circuit

Theory: Analyze the transient response of an RC circuit.

Considering the RC Circuit (also called RC network) shown in this figure



we can use the Kirchhoff's current law (KCL) to write the following equation

$$C\frac{dv_0(t)}{dt}+\frac{v_0(t)}{R}=0$$

and we can rearrange into the equation

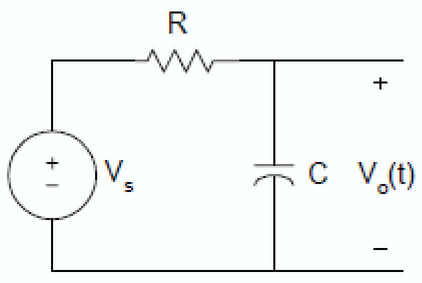
$$\frac{dv_0(t)}{dt}+\frac{v_0(t)}{RC}=0$$

The solution to the equation above is

$$v_0(t)=V_m e^{\frac{-t}{RC}}$$

where
V_m is the initial voltage across the capacitor
RC is the time constant
This solution represents the voltage across a discharging capacitor.

Now, to obtain the voltage across a charging capacitor, let us consider this figure that includes a voltage source



Again, using KCL, the equation describing the charging RC circuit is

$$C\frac{dv_0(t)}{dt}+\frac{v_0(t)-V_s}{R}=0$$

If the capacitor is not charged initially, that is $v_0(t) = 0$ when $t = 0$, then the solution to the equation above is given by

$$v_0(t) = V_s\left(1 - e^{\frac{-t}{RC}}\right)$$

The following examples illustrate the use of Matlab for solving problems related to RC circuits.

MATLAB Code:

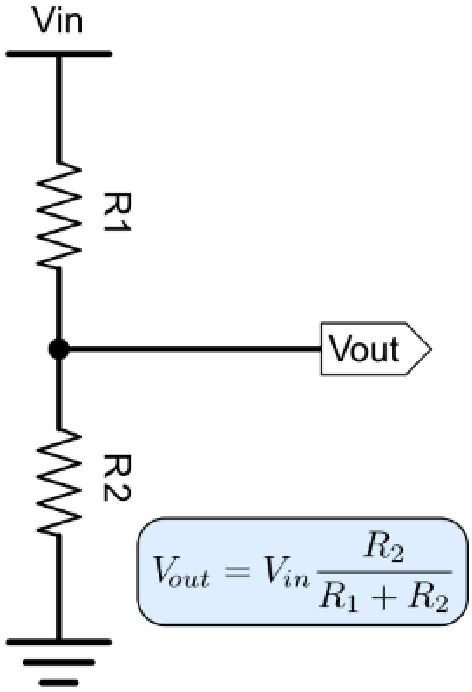
```
% Transient Analysis of RC Circuit
R = 1e3; % Resistance in ohms
C = 1e-6; % Capacitance in farads
sys = tf([1], [R*C 1]);
impulse(sys);
title('Transient Response of RC Circuit');
```

23. Voltage Divider Rule

Theory: Calculate the output voltage in a voltage divider circuit.

Voltage Division Rule

A series circuit acts as a **voltage divider** as it divides the total supply voltage into different voltages across the circuit elements. Figure2 shows a voltage divider circuit in which the total supply voltage *V* has been divided into voltages V_1 and V_2 across two resistances R_1 and R_2. Although, the current through both resistances is same, i.e., *I*.



MATLAB Code:

```
% Voltage Divider Rule
R1 = 10e3; % Resistor 1 in ohms
R2 = 20e3; % Resistor 2 in ohms
Vin = 10; % Input voltage in volts
Vout = (R2 / (R1 + R2)) * Vin; % Output voltage
disp(['Output Voltage: ', num2str(Vout), ' V']);
```

24. Digital to Analog Conversion (DAC)

Theory: Simulate a digital to analog conversion process.
MATLAB Code:

```
% Digital to Analog Conversion (DAC)
digital_signal = [0 1 1 0 1 0 0 1];
analog_signal = filter(1, [1 -0.9], digital_signal); % Simple DAC model
stem(analog_signal);
title('Digital to Analog Conversion');
xlabel('Sample');
ylabel('Amplitude');
```

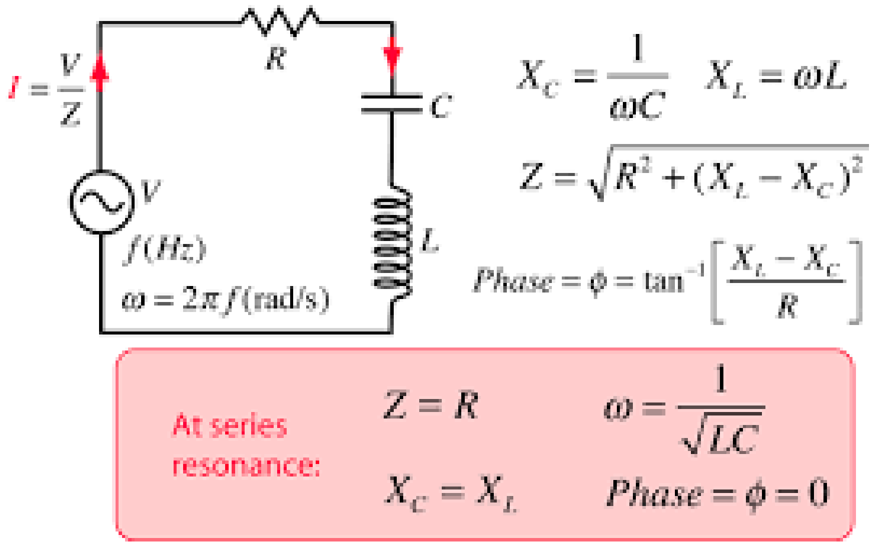
25. Analog to Digital Conversion (ADC)

Theory: Simulate an analog to digital conversion process.
MATLAB Code:

```
% Analog to Digital Conversion (ADC)
Fs = 1000; % Sampling frequency
t = 0:1/Fs:1-1/Fs;
analog_signal = sin(2*pi*50*t); % Analog signal
digital_signal = round(analog_signal); % Simple ADC model
stem(digital_signal);
title('Analog to Digital Conversion');
xlabel('Sample');
ylabel('Amplitude');
```

26. Impedance of RLC Circuit

Theory: Calculate the impedance of a series RLC circuit at a given frequency.

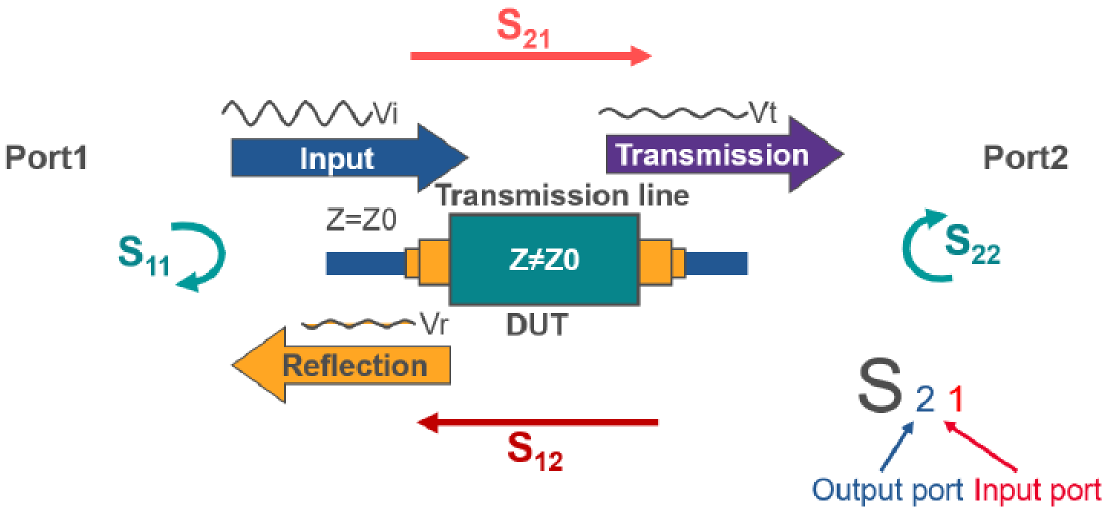


MATLAB Code:

```
% Impedance of RLC Circuit
R = 50; % Resistance in ohms
L = 100e-3; % Inductance in henrys
C = 10e-6; % Capacitance in farads
f = 60; % Frequency in Hz
omega = 2*pi*f; % Angular frequency
Z = R + 1i*(omega*L - 1/(omega*C)); % Impedance
disp(['Impedance: ', num2str(abs(Z)), ' Ohms']);
```

27. S-Parameters in RF Circuit

Theory: Compute and plot the S-parameters for a two-port network.



Reflection/Input = Reflection coefficient → S_{11},S_{22}

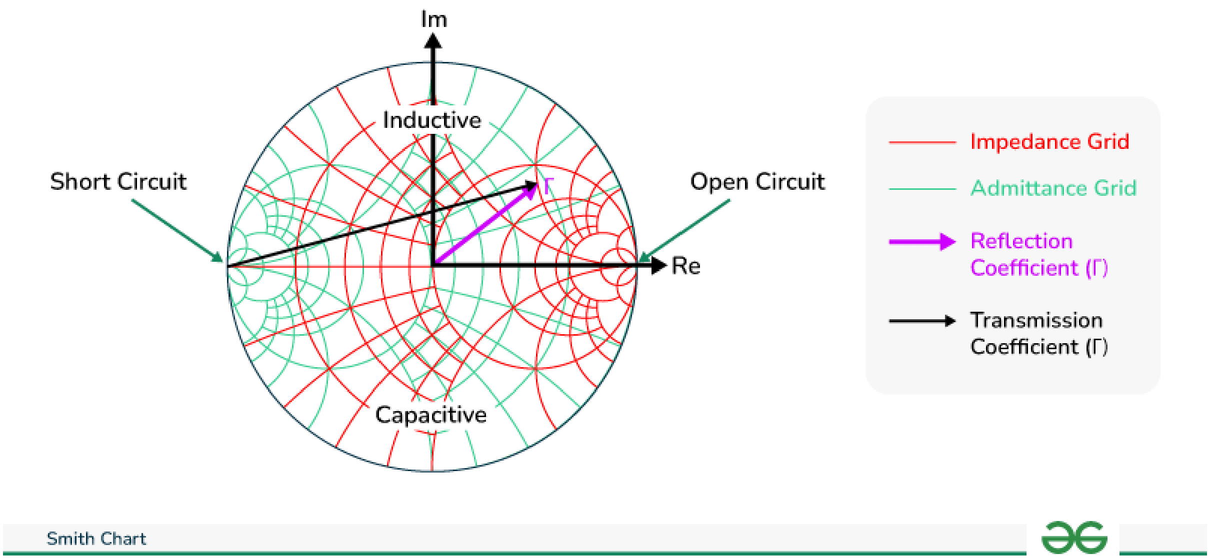
Transmission/Input = Transmission coefficient → S_{21},S_{12}

MATLAB Code:

```
% S-Parameters in RF Circuit
s_params = sparameters('default.s2p'); % Load S-parameter file
rfplot(s_params);
title('S-Parameters');
```

28. Smith Chart

The Smith Chart is a graphical tool that is used in RF transmission line design and electrical engineering. It helps in analyzing and designing transmission lines and impedance-matching networks. It represents complex impedance values on a polar plot, which allows experts to visualize and manipulate impedance changes.



Smith Chart

By using the Smith Chart, engineers can improve signal transmission and minimize reflections, which is particularly important in applications like antenna design, microwave circuits, and radio frequency systems. The Smith Chart simplifies complex calculations involved in designing and analyzing impedance-matching circuits and transmission lines, making it an important tool in RF engineering and microwave design.

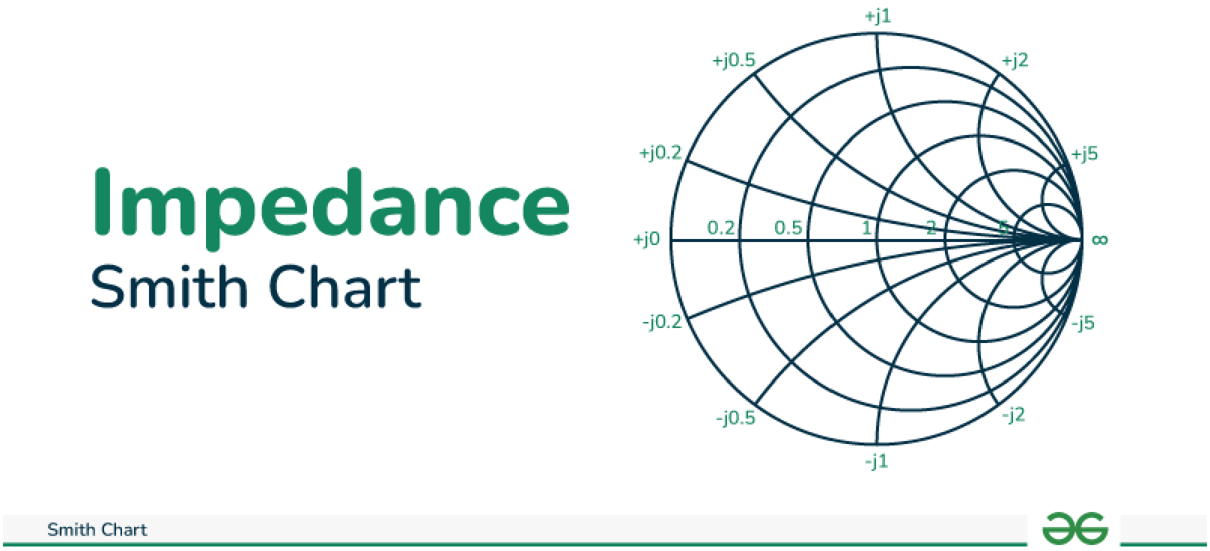
Types of Smith Chart

There are various types of Smith Charts depend on the various parameters:

- Impedance Smith Chart
- Admittance Smith Chart
- Immittance Smith Chart

Impedance Smith Chart

Impedance Smith Charts also known as Z charts are the polar graphs that show the normalized line impedance in the complex reflection coefficient plane. It is made of circles that represent different values. These graphs are used for visualizing the impedance at any point on the transmission line or any input in the systems of the antenna. The Smith Charts are generally termed as the usual type of Smith Charts as they correspond to impedance. It is useful while working with series components of the circuit. The impedance of these is the main type where other types are considered as their derivatives.



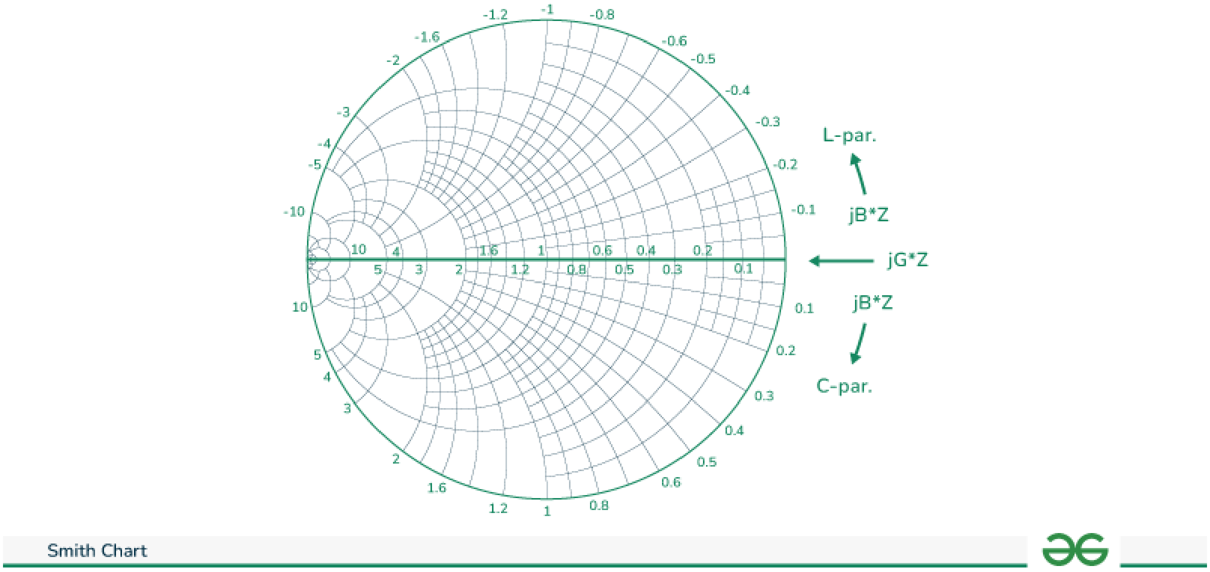
Impedance Smith Chart

The Smith Charts are made of many circles and segments of circles which are arranged for plotting impedance in the form of R ± jX. The horizontal line that passes through the center of the circle as seen in the above figure represents the resistance of R=0 on the very left side of the line and has infinite resistance on the very right side. The value R=1 passes from the center of the circle.

The curves which are above the horizontal line represent the inductive reactance and the curves which are below the line are capacitive reactance. In most of the cases, the RF impedance is 50Ω and the value of the center of the chart is R=1, so the center point becomes 50Ω.

Admittance Smith Chart

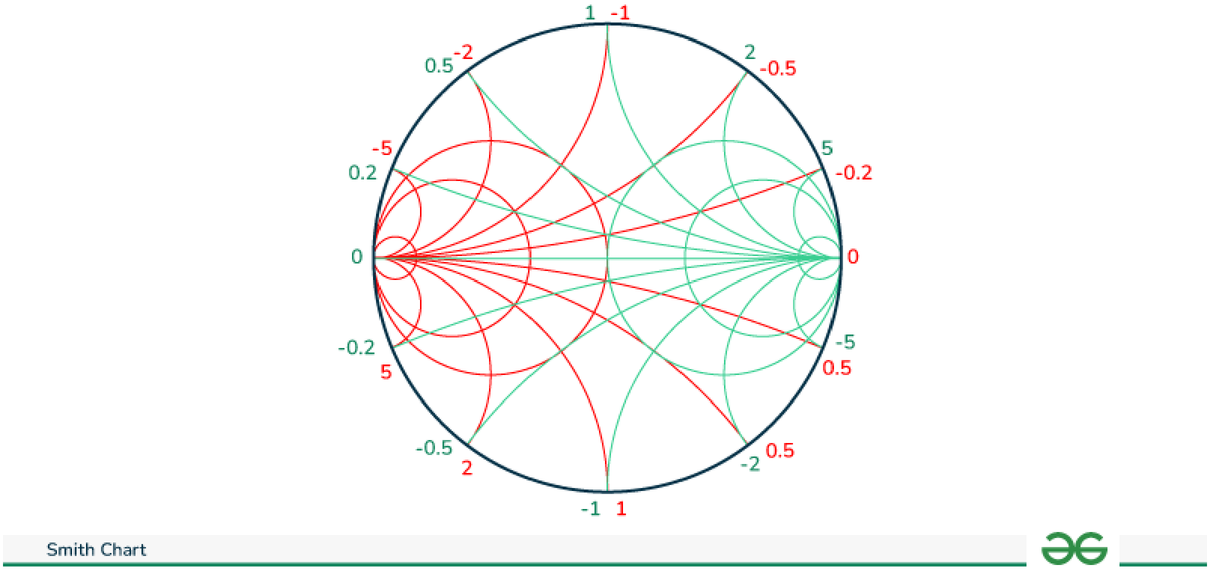
The Admittance Smith Chart also known as the Y chart is the normalized admittance (Y=C+iS) in the Γ-plane where (C,S) represents the conductance and susceptance of Y. The admittance chart can be obtained by rotating the impedance chart by 180°. The upper half of the chart represents negative values of S (or negative susceptance). The image given below represents the Admittance Smith Chart.



Admittance Smith Chart

It is useful while working with parallel components of the circuit. The equation which establish the relationship between the admittance and impedance is:
YL=1ZL=C+iSYL=ZL1=C+iS
Where,
Y_L : Admittance of the load
Z_L : Impedance
C: Conductance
S: Susceptance

Immittance Smith Chart
An Immittance Smith chart also known as a YZ chart is used when both series and parallel components are present in the circuit. It superimposes the impedance and admittance charts on each other. It is helpful in situations like transmission lines and matching impedances where both admittance and impedance are used simultaneously. The image given below represents the Immittance Smith Chart.



Immittance Smith Chart
According to the figure given above, when we move along the constant conduction circles (red color), the inductors and transmission lines will affect the load impedance. When we move along the constant resistance circles (green color), the series capacitor and inductors will affect the load impedance.

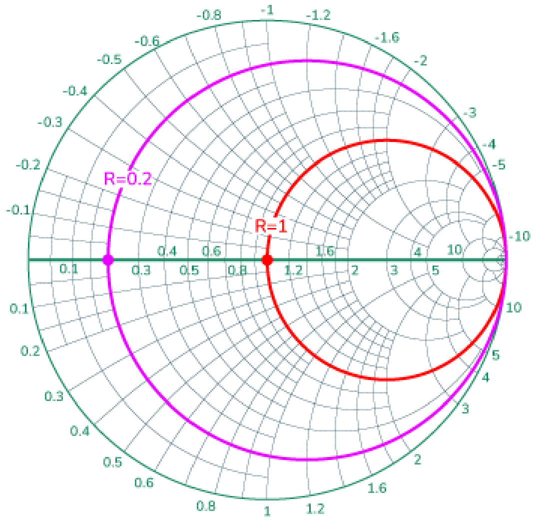
Basics of Smith Chart
The Smith Chart exhibits complex reflection coefficients in polar form for particular load impedances.We All Know Impedance is the Sum of Reactance and Resistance ,Similarly reflection coefficient, also a complex number, are represented by load impedance ZL and reference impedance Z0 respectively.
It can Represented as
ZL−Z0ZL+Z0=ZL−1ZL+1ZL+Z0ZL−Z0=ZL+1ZL−1
Where
ZL Corresponds to Load Impedance
Z0 Corresponds to Transmitters Impedance
It primarily serves as a graphical representation illustrating the impedance of an antenna across frequencies, which may encompass single or multiple ranges of points.

Components of Smith Chart
While understanding the Smith chart, we need to understand its components. There are various components depending on the type of Smith Chart which is as follows:

Smith Chart	**Components**
Impedance Smith Chart	1. Constant R circle 2. Constant X circle
Admittance Smith Chart	1. Constant C circle 2. Constant S circle

Constant R Circles
The figure given below represents the constant resistance circle. The horizontal line represents the resistance axis. It is used to represent the complex impedances of the resistive

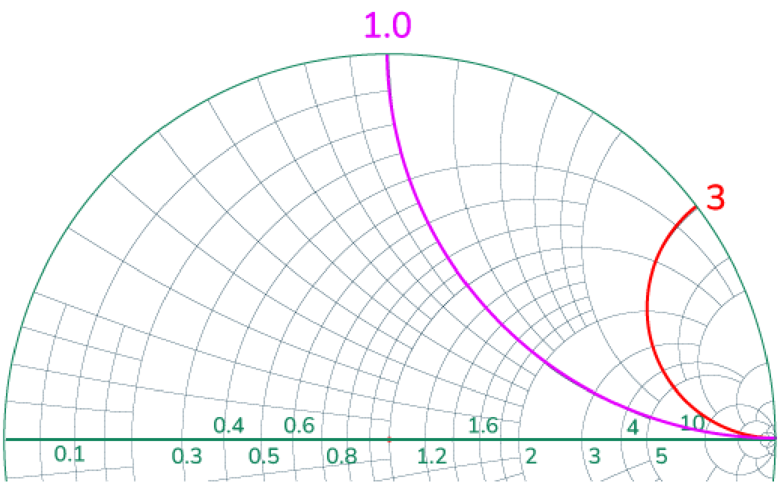
part of circuit.



Smith Chart

Constant R Circle
Let’s start from the center having the normalized resistance R=1. A circle (red color) tangent to the right side of the chart which passes through the prime center represents the constant normalized resistance circle with the constant resistance of 1. A similar circle (pink color) which passes through the resistance axis at R=0.2 represents the normalized resistance of 0.2 at every point on that circle.

Constant X Circles
It is known as the constant reactance circle. The reactance axis lies across the the circumference of the Smith Chart. The figure given below represents the constant reactance circle.



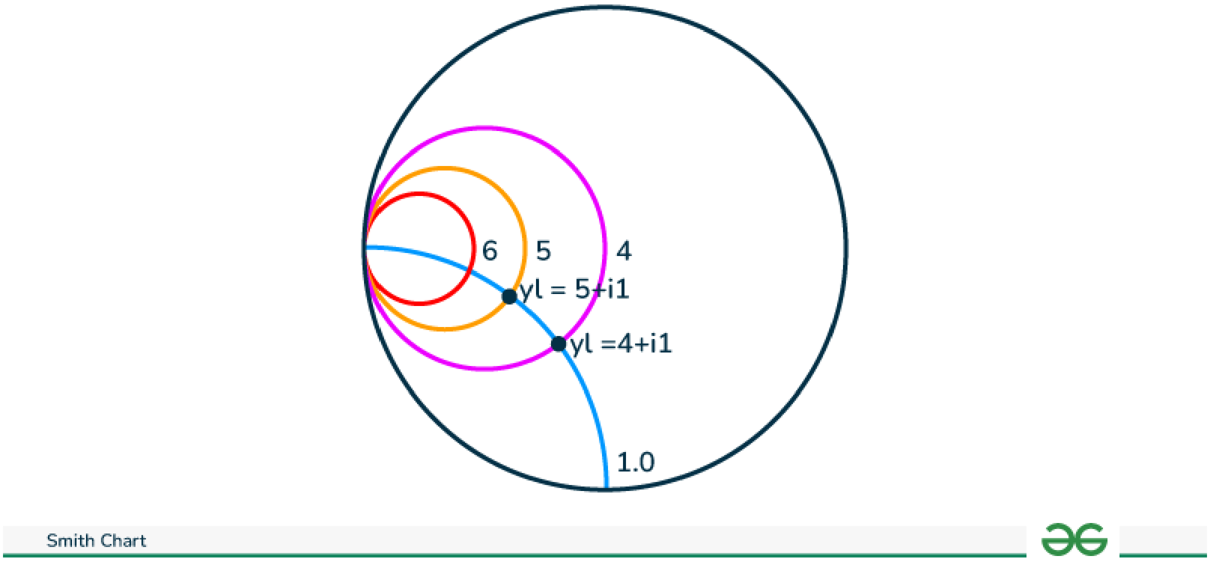
Smith Chart

Constant Reactance Circle
Every point along the curve (either pink or red) has the same value of reactance or imaginary

part. The points lying on the pink curve have the normalized reactance of 1.0 while the points lying on the red curve have the normalized reactance of 3.0. The upper half of the Smith Chart have the positive reactance value (inductive) while the lower half of the Smith chart have the negative reactance value (capacitive).

Constant C and S Circle

The admittance chart is just the reverse of the impedance chart. In the admittance Smith chart, instead of having a constant R circle, we have a constant C (conductance) circle, and instead of a constant X circle, we have a constant S (susceptance) circle. The concept is the same as mentioned above for the circles, but the graph will be opposite. The graph of constant C and S circles are given below.



Constant C and S Circles

The blue line shows the susceptance of value 1.0. The circles represents the constant conductance. Let’s take an example of pink circle. Its value of conductance is 4.

How to Plot Impedance and Admittance Smith Chart?

To plot the Smith Chart follow the given steps:

Impedance Smith Chart

For plotting the impedance chart, we should keep two major points in mind:

1. **The constant R circles:** This circle is generated when the resistance is constant while the reactance is variable. The outermost and innermost R constant circles represent the boundaries of the Smith Chart. The outermost circle represents zero and the innermost circle represents infinite resistance.
2. **The constant X circles:** It is the part of the circle that represents the reactance. The curves which are above the horizontal line passing through the center represent the inductive reactance and the curves which are below the horizontal line represent capacitive reactance which is already shown in the figure above. The circles with the constant R circles plot the impedance values in the form of (R ±jX) on the chart.

The equation used for construction of Smith Chart is:

Z= R+jX

z=R+jXR0+jX0z=R0+jX0R+jX

where,

- Z: Complex impedance
- z: Normalized impedance
- R: Real number of impedance
- Z: Imaginary part of the impedance
- R_0: Reference impedance
- X_0: Reference impedance in angular form

Admittance Smith Chart

The frequency of series RC, RL, and RLC can be easily analyzed using an impedance chart. When parallel connection of RLC components we use the concept of admittance (Y) which simplifies the calculation. We can draw plot admittance contours in the Γ-plane for analyzing the parallel connection.

For this, we will use concepts of impedance Smith Chart for deriving the admittance Smith Chart. The impedance chart is a plot of function for some specific values of z in Γ-plane.

Γ=z−1z+1Γ=z+1z−1 ——— (equation 1)

Where, z =r+jx is normalized impedance

As we know the parameter z in equation 1 represents the impedance of a circuit. If we look from a mathematical perspective z is a simple complex number on Γ-plane. The relation between Γ and admittance Y in the equation is:

Γ=z−z0z+z0Γ=z+z0z−z0

Substituting the values of Z= 1/Y and Z_0=1/Y_0 where Y_0 is reference admittance.

Γ=Y0−YY0+YΓ=Y0+YY0−Y

Now we divide both the numerator and denominator by Y_0 and defining the normalized admittance y = Y/Y_0

The final equation is:

Γ=−y−1y+1Γ=−y+1y−1 ——— (equation 2)

Here y is a complex number where (y=C+jS) which is similar to equation 1 with additional negative sign or multiplied by (-1). We can observe that admittance contours in the Γ-plane which is obtained by rotating the impedance chart by 180°.

Advantages and Disadvantages of Smith Chart

There are some list of Advantages and Disadvantages of Smith Chart given below :

Advantages of Smith Chart

- Smith chart helps find the complex impedance and reflection coefficients. It makes the analysis of RF circuits easier.
- It helps in finding the matching impedance of the network which helps in the maximum transfer of the power.
- The reflection coefficients can be easily found with the help of Smith Charts. It helps in analyzing and visualizing the impedance mismatches. This helps prevent the signal reflections.
- With the help of the Smith Chart, we can find the admittance of the circuit easily. It provides additional information about the circuit which enhances the flexibility in the circuit design.

Disadvantages of Smith Chart

- It is not applicable in the analysis of the DC circuits. It is only applicable in the RF and microwave applications.

- The Smith Chart represents the complex impedance on the 2-D chart. It does not accurately represent the three-dimensional impedances.
- Smith Chart is a bit complex. To understand the concept, it requires a certain level of expertise.
- It does not provide information about the absolute frequency. To determine the same, additional tools are required.

Applications of Smith Charts

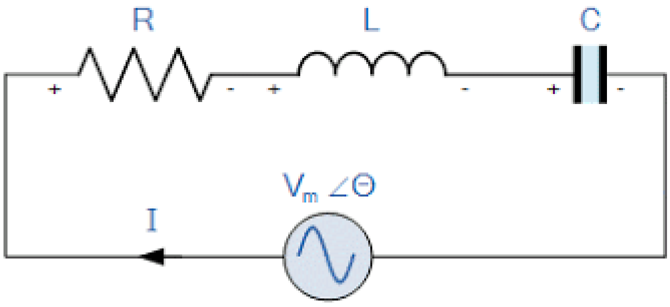
- **Transmission Line Analysis:** Smith charts help in understanding and correcting issues in transmission lines, such as impedance mismatches and signal reflections, critical in high-frequency applications.
- **Antenna Design:** Engineers use Smith charts to design and tune antennas for optimal performance by matching the antenna's impedance to the transmission line's impedance.
- **Filter Design:** In the field of microwave and RF filter design Smith Charts play a role in attaining desired frequency response characteristics by manipulating component values and impedance transformations.
- **Amplifier Design:** Engineers utilize Smith charts to optimize input output matching networks of amplifiers in order to maximize gain while minimizing noise levels and distortion.
- **S-parameter Analysis:** These charts find application in vector network analyzers where they display S parameters providing information on how electrical signals propagate through a system.

Theory: Plot the reflection coefficient on a Smith chart.
MATLAB Code:

```
% Smith Chart
z = linspace(0.1, 10, 100); % Normalized impedance values
smithchart(z);
title('Smith Chart');
```

29. Resonance in RLC Circuit

Theory: Analyze the resonance frequency and behavior of a series RLC circuit.



MATLAB Code:

```
% Resonance in RLC Circuit
L = 1e-3; % Inductance in henrys
C = 100e-9; % Capacitance in farads
f_resonance = 1/(2*pi*sqrt(L*C)); % Resonance frequency
disp(['Resonance Frequency: ', num2str(f_resonance), ' Hz']);
```

30. Stability Analysis Using Routh-Hurwitz Criterion

Theory: Determine the stability of a system using the Routh-Hurwitz criterion.

Routh-Hurwitz Stability Criterion
Routh-Hurwitz stability criterion is having one necessary condition and one sufficient condition for stability. If any control system doesn’t satisfy the necessary condition, then we can say that the control system is unstable. But, if the control system satisfies the necessary condition, then it may or may not be stable. So, the sufficient condition is helpful for knowing whether the control system is stable or not.

Necessary Condition for Routh-Hurwitz Stability
The necessary condition is that the coefficients of the characteristic polynomial should be positive. This implies that all the roots of the characteristic equation should have negative real parts.
Consider the characteristic equation of the order ‘n’ is -
a0sn+a1sn−1+a2sn−2+...+an−1s1+ans0=0
Note that, there should not be any term missing in the $\mathbf{n^{th}}$ order characteristic equation. This means that the $\mathbf{n^{th}}$ order characteristic equation should not have any coefficient that is of zero value.

Sufficient Condition for Routh-Hurwitz Stability
The sufficient condition is that all the elements of the first column of the Routh array should have the same sign. This means that all the elements of the first column of the Routh array should be either positive or negative.

Routh Array Method
If all the roots of the characteristic equation exist to the left half of the ‘s’ plane, then the control system is stable. If at least one root of the characteristic equation exists to the right half of the ‘s’ plane, then the control system is unstable. So, we have to find the roots of the characteristic equation to know whether the control system is stable or unstable. But, it is

difficult to find the roots of the characteristic equation as order increases.
So, to overcome this problem there we have the **Routh array method**. In this method, there is no need to calculate the roots of the characteristic equation. First formulate the Routh table and find the number of the sign changes in the first column of the Routh table. The number of sign changes in the first column of the Routh table gives the number of roots of characteristic equation that exist in the right half of the 's' plane and the control system is unstable.
Follow this procedure for forming the Routh table.

- Fill the first two rows of the Routh array with the coefficients of the characteristic polynomial as mentioned in the table below. Start with the coefficient of sn and continue up to the coefficient of s0.
- Fill the remaining rows of the Routh array with the elements as mentioned in the table below. Continue this process till you get the first column element of **row s0** is an. Here, an is the coefficient of s0 in the characteristic polynomial.

Note − If any row elements of the Routh table have some common factor, then you can divide the row elements with that factor for the simplification will be easy.
The following table shows the Routh array of the nth order characteristic polynomial.
a0sn+a1sn−1+a2sn−2+...+an−1s1+ans0

sn	a0	a2	a4	a6	...	...
sn−1	a1	a3	a5	a7	...	...
sn−2	b1=a1a2−a3a0a1	b2=a1a4−a5a0a1	b3=a1a6−a7a0a1	...	...	...
sn−3	c1=b1a3−b2a1b1	c2=b1a55−b3a1b1	⋮			
⋮	⋮	⋮	⋮			
s1	⋮	⋮				
s0	an					

Example
Let us find the stability of the control system having characteristic equation,
s4+3s3+3s2+2s+1=0
Step 1 − Verify the necessary condition for the Routh-Hurwitz stability.
All the coefficients of the characteristic polynomial, s4+3s3+3s2+2s+1 are positive. So, the control system satisfies the necessary condition.
Step 2 − Form the Routh array for the given characteristic polynomial.

s4	1	3	1
s3	3	2	
s2	(3×3)−(2×1)3=73	(3×1)−(0×1)3=33=1	
s1	(73×2)−(1×3)73=57		
s0	1		

Step 3 − Verify the sufficient condition for the Routh-Hurwitz stability.
All the elements of the first column of the Routh array are positive. There is no sign change in the first column of the Routh array. So, the control system is stable.

Special Cases of Routh Array
We may come across two types of situations, while forming the Routh table. It is difficult to complete the Routh table from these two situations.
The two special cases are −

- The first element of any row of the Routh array is zero.
- All the elements of any row of the Routh array are zero.

Let us now discuss how to overcome the difficulty in these two cases, one by one.

First Element of any row of the Routh array is zero

If any row of the Routh array contains only the first element as zero and at least one of the remaining elements have non-zero value, then replace the first element with a small positive integer, ϵ. And then continue the process of completing the Routh table. Now, find the number of sign changes in the first column of the Routh table by substituting ϵ tends to zero.

Example

Let us find the stability of the control system having characteristic equation,
s4+2s3+s2+2s+1=0

Step 1 − Verify the necessary condition for the Routh-Hurwitz stability.

All the coefficients of the characteristic polynomial, s4+2s3+s2+2s+1 are positive. So, the control system satisfied the necessary condition.

Step 2 − Form the Routh array for the given characteristic polynomial.

s4	1	1	1
s3	2 1	2 1	
s2	(1×1)−(1×1)1=0	(1×1)−(0×1)1=1	
s1			
s0			

The row s3 elements have 2 as the common factor. So, all these elements are divided by 2.

Special case (i) − Only the first element of row s2 is zero. So, replace it by ϵ and continue the process of completing the Routh table.

s4	1	1	1
s3	1	1	
s2	ϵ	1	
s1	(ϵ×1)−(1×1)ϵ=ϵ−1ϵ		
s0	1		

Step 3 − Verify the sufficient condition for the Routh-Hurwitz stability.

As ϵ tends to zero, the Routh table becomes like this.

s4	1	1	1
s3	1	1	
s2	0	1	
s1	-∞		
s0	1		

There are two sign changes in the first column of Routh table. Hence, the control system is unstable.

All the Elements of any row of the Routh array are zero

In this case, follow these two steps −

- Write the auxilary equation, A(s) of the row, which is just above the row of zeros.
- Differentiate the auxiliary equation, A(s) with respect to s. Fill the row of zeros with these coefficients.

Example

Let us find the stability of the control system having characteristic equation,
s5+3s4+s3+3s2+s+3=0

Step 1 − Verify the necessary condition for the Routh-Hurwitz stability.

All the coefficients of the given characteristic polynomial are positive. So, the control system satisfied the necessary condition.

Step 2 − Form the Routh array for the given characteristic polynomial.

s5	1	1	1
s4	3 1	3 1	3 1
s3	(1×1)−(1×1)1=0	(1×1)−(1×1)1=0	
s2			
s1			
s0			

The row s4 elements have the common factor of 3. So, all these elements are divided by 3.

Special case (ii) − All the elements of row s3 are zero. So, write the auxiliary equation, A(s) of the row s4.

A(s)=s4+s2+1

Differentiate the above equation with respect to s.

dA(s)ds=4s3+2s

Place these coefficients in row s3.

s5	1	1	1
s4	1	1	1
s3	4 2	2 1	
s2	(2×1)−(1×1)2=0.5	(2×1)−(0×1)2=1	
s1	(0.5×1)−(1×2)0.5=−1.50.5=−3		
s0	1		

Step 3 − Verify the sufficient condition for the Routh-Hurwitz stability.

There are two sign changes in the first column of Routh table. Hence, the control system is unstable.

In the Routh-Hurwitz stability criterion, we can know whether the closed loop poles are in on left half of the 's' plane or on the right half of the 's' plane or on an imaginary axis. So, we can't find the nature of the control system. To overcome this limitation, there is a technique known as the root locus.

MATLAB Code:

```
% Stability Analysis Using Routh-Hurwitz Criterion
coefficients = [1 3 3 1]; % Coefficients of the characteristic equation
Routh_table = routh(coefficients); % Routh array
disp('Routh Table:');
disp(Routh_table);
```

31. Power Spectral Density (PSD)

Theory: Calculate and plot the power spectral density of a signal.

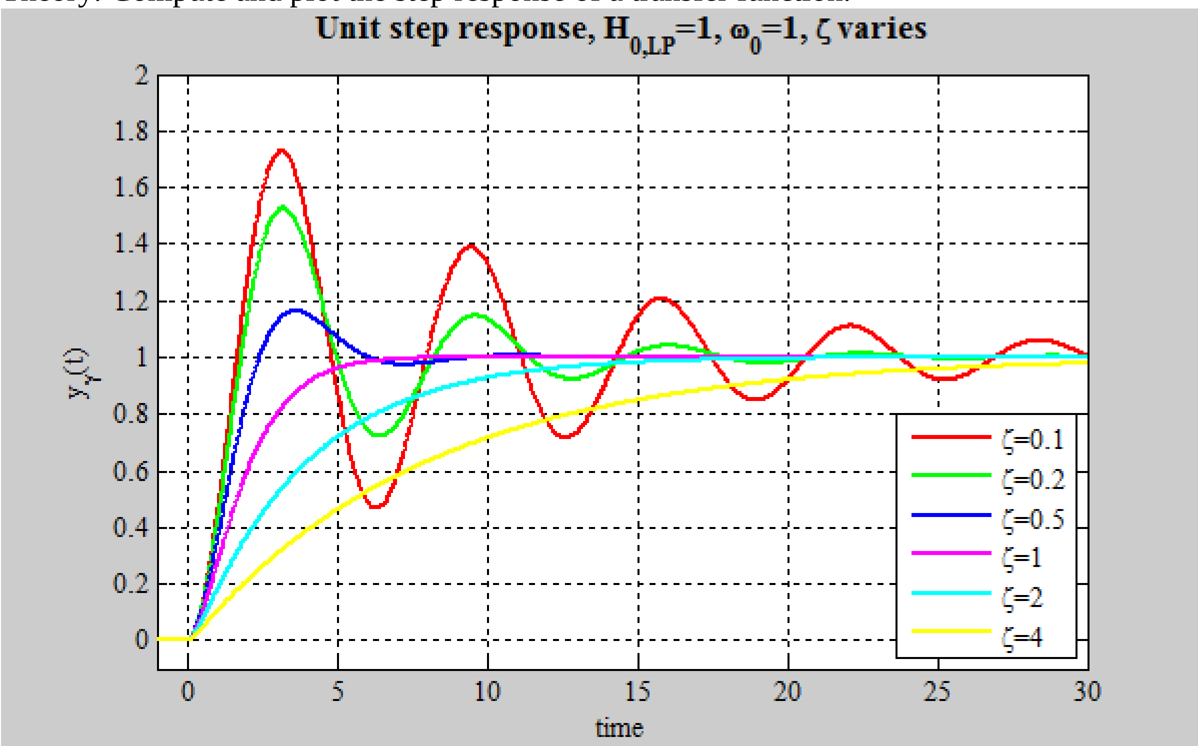
What is Power Spectral Density (PSD)?
Power Spectral Density also known as PSD is a fundamental concept used in signal processing to measure how the average power or the strength of the signal is distributed across different frequency components. The Average Power referred to here is known as the mean amount of the energy transferred or distributed throughout a given time range. Mathematically, Power Spectral Density (PSD) sometimes also known as Power Density (PD) denoted here as S(ω)S(ω) for a signal x(t)x(t) can be expressed as below:
S(ω)=lim⁡τ→∞|X(ω)|2τS(ω)=limτ→∞τ|X(ω)|2

MATLAB Code:

```
% Power Spectral Density (PSD)
Fs = 1000; % Sampling frequency
t = 0:1/Fs:1-1/Fs;
x = cos(2*pi*100*t) + randn(size(t)); % Signal with noise
[Pxx, f] = pwelch(x, [], [], [], Fs);
plot(f, 10*log10(Pxx));
title('Power Spectral Density');
xlabel('Frequency (Hz)');
ylabel('Power/Frequency (dB/Hz)');
```

32. Step Response of a Transfer Function

Theory: Compute and plot the step response of a transfer function.

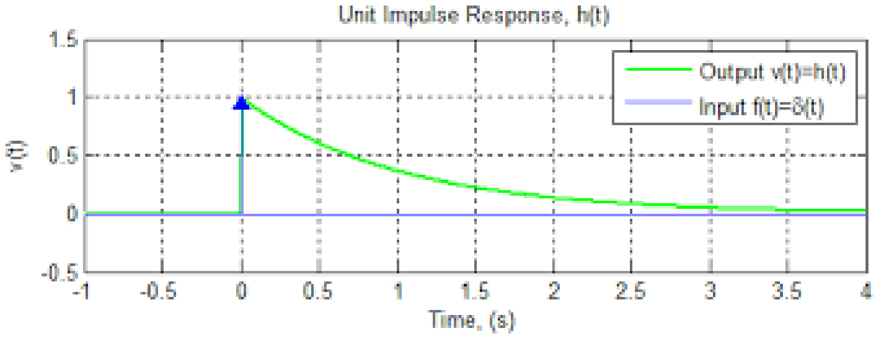


MATLAB Code:

```
% Step Response of a Transfer Function
sys = tf([1], [1 2 1]);
step(sys);
title('Step Response');
```

33. Impulse Response of a Transfer Function

Theory: Compute and plot the impulse response of a transfer function.

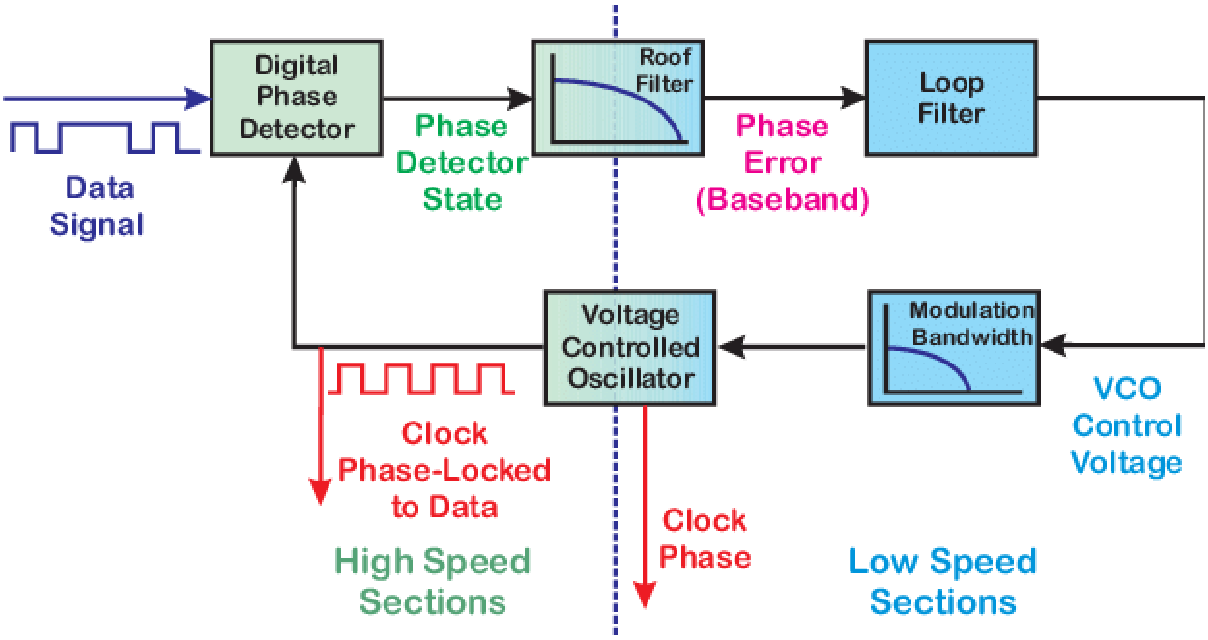


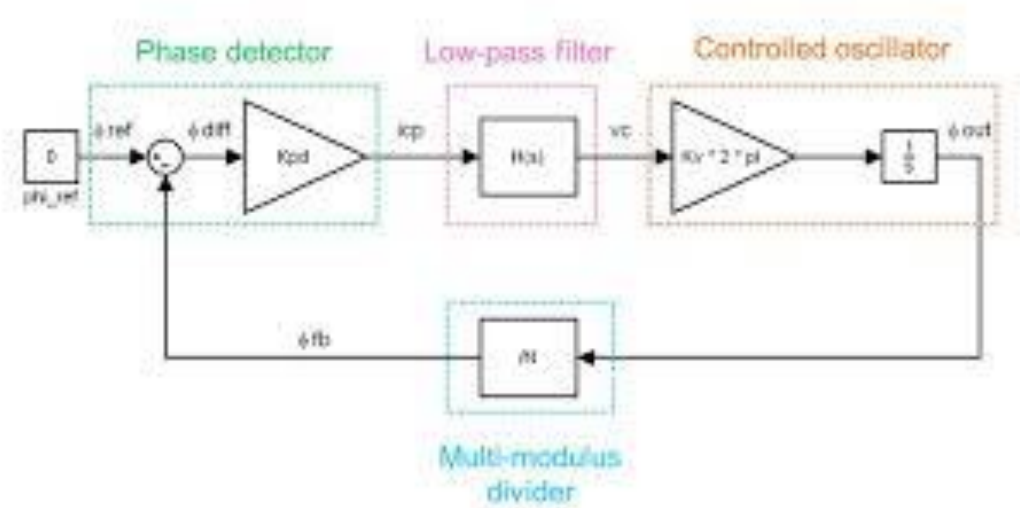
MATLAB Code:

```
% Impulse Response of a Transfer Function
sys = tf([1], [1 2 1]);
impulse(sys);
title('Impulse Response');
```

34. Phase Locked Loop (PLL) Simulation

Theory: Simulate a phase locked loop system.





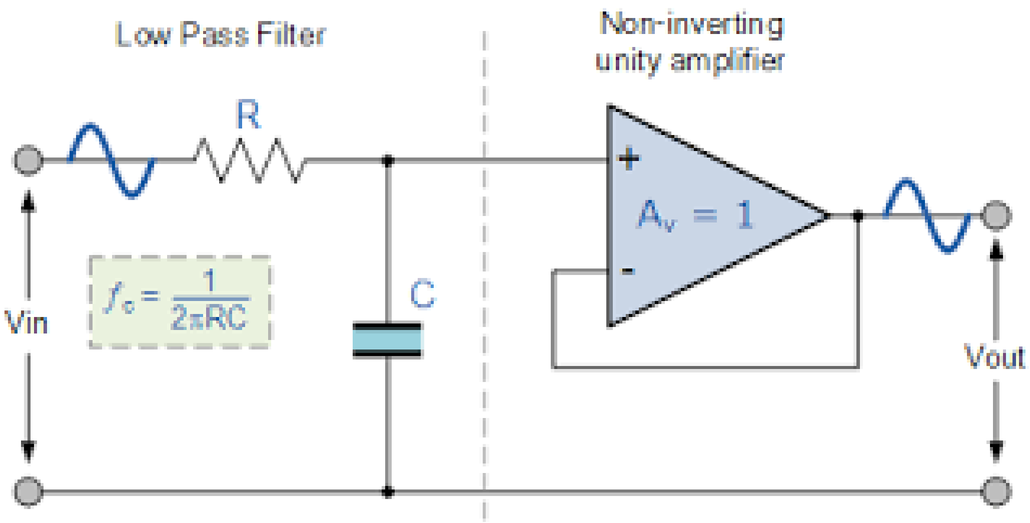
MATLAB Code:

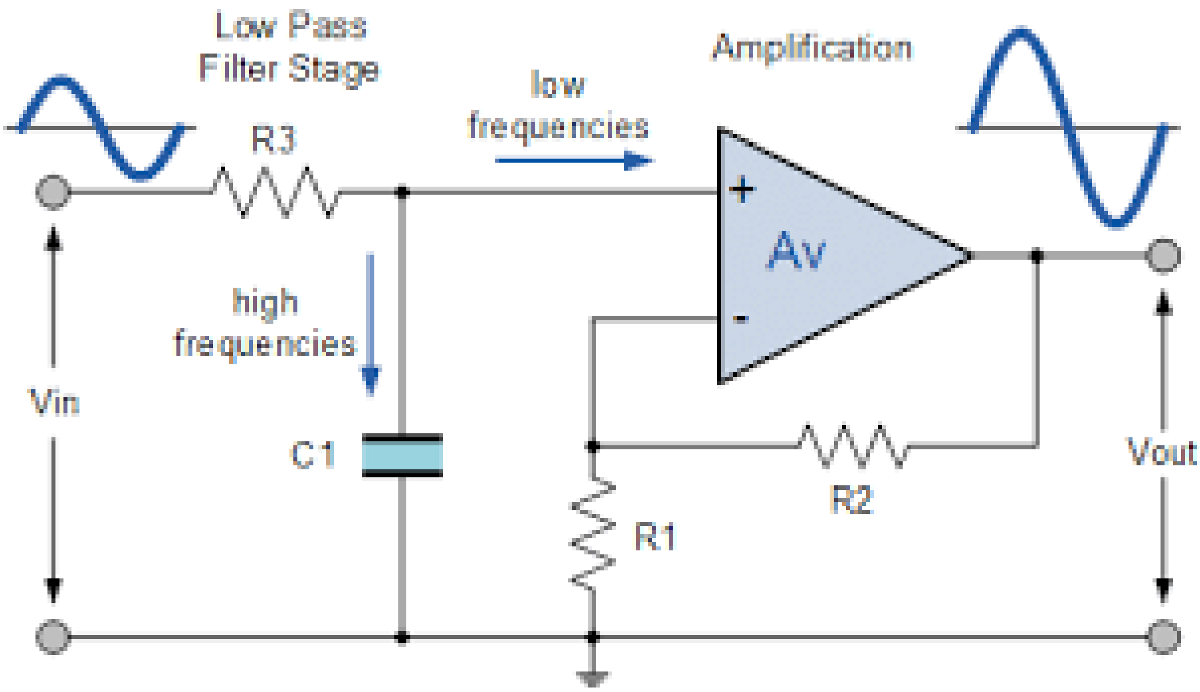
```
% Phase Locked Loop (PLL) Simulation
Fs = 1000; % Sampling frequency
t = 0:1/Fs:1-1/Fs;
F0 = 50; % Initial frequency
input_signal = cos(2*pi*F0*t + pi/4); % Input signal
[~, pll_output] = pll(input_signal, Fs); % Using a PLL function
plot(t, input_signal, t, pll_output);
title('Phase Locked Loop (PLL) Simulation');
xlabel('Time (s)');
```

```
ylabel('Amplitude');
legend('Input Signal', 'PLL Output');
```

35. Active Low Pass Filter

Theory: Design and simulate an active low pass filter using an operational amplifier.



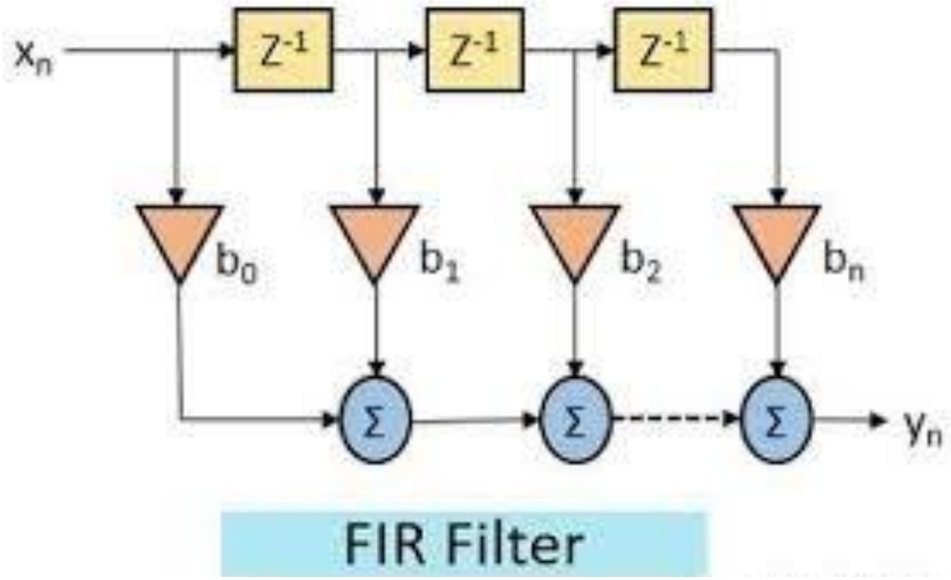


MATLAB Code:

```
% Active Low Pass Filter
R = 1e3; % Resistance in ohms
C = 1e-6; % Capacitance in farads
sys = tf([1], [R*C 1]);
bode(sys);
title('Active Low Pass Filter');
```

36. Digital Filter Design (FIR)

Theory: Design and simulate a finite impulse response (FIR) digital filter.

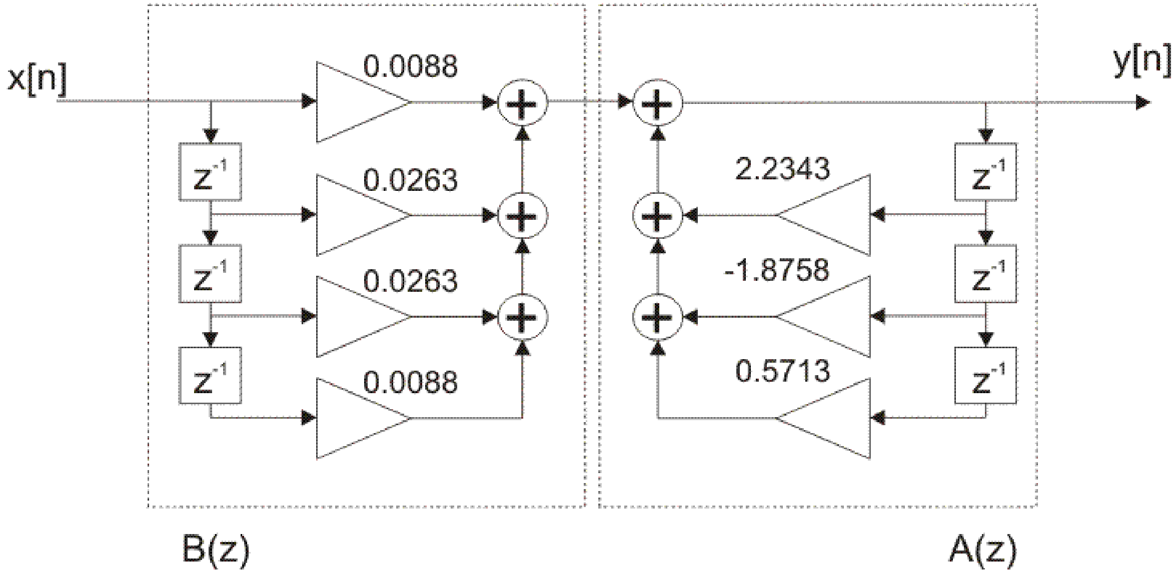


MATLAB Code:

```
% Digital Filter Design (FIR)
Fs = 1000; % Sampling frequency
N = 50; % Filter order
Fc = 200; % Cutoff frequency
fir_coeff = fir1(N, Fc/(Fs/2));
freqz(fir_coeff, 1, [], Fs);
title('FIR Digital Filter Design');
```

37. Digital Filter Design (IIR)

Theory: Design and simulate an infinite impulse response (IIR) digital filter.

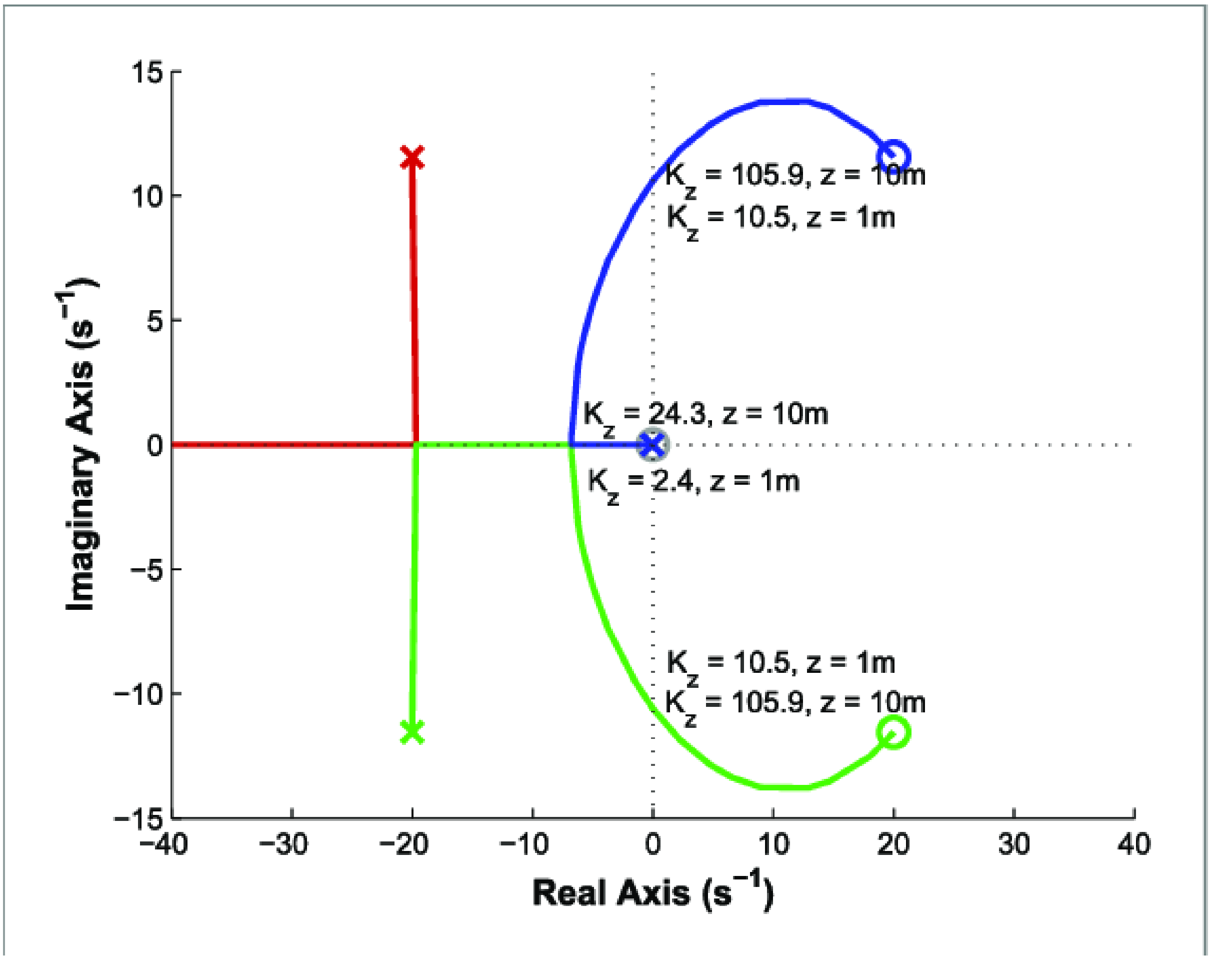


MATLAB Code:

```
% Digital Filter Design (IIR)
Fs = 1000; % Sampling frequency
Fc = 200; % Cutoff frequency
[b, a] = butter(4, Fc/(Fs/2));
freqz(b, a, [], Fs);
title('IIR Digital Filter Design');
```

38. Root Locus Plot

Theory: Plot the root locus of a transfer function.

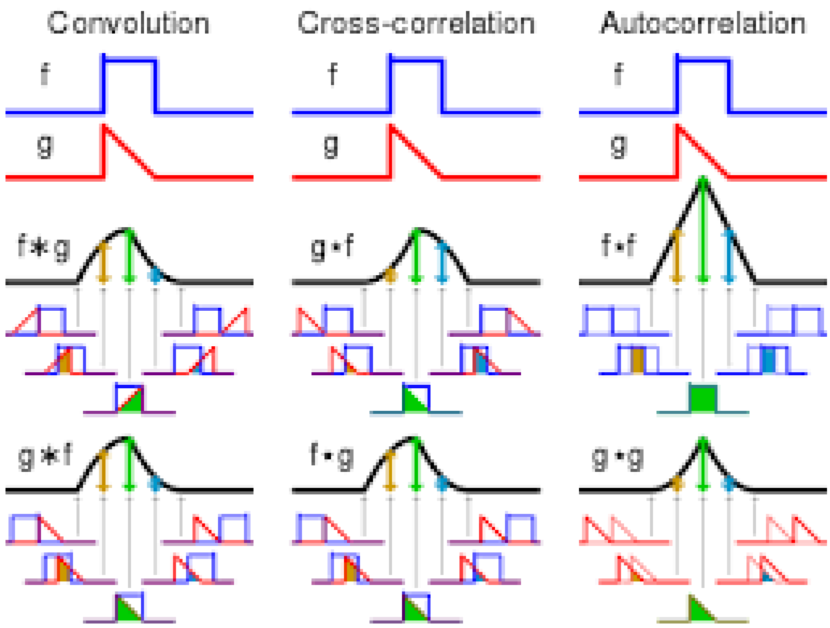


MATLAB Code:

```
% Root Locus Plot
sys = tf([1], [1 3 3 1]);
rlocus(sys);
title('Root Locus Plot');
```

39. Signal Convolution

Theory: Perform convolution of two signals.

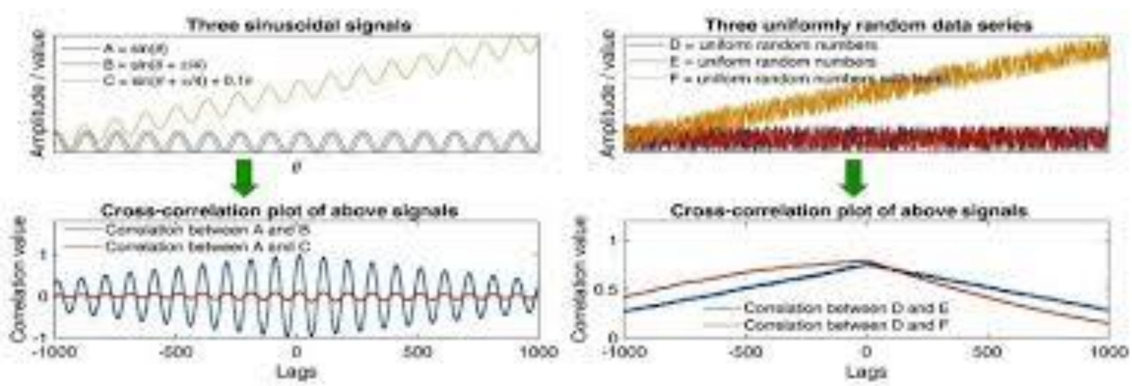


MATLAB Code:

```
% Signal Convolution
x = [1 2 3];
h = [1 1 1];
y = conv(x, h);
stem(y);
title('Signal Convolution');
xlabel('Sample');
ylabel('Amplitude');
```

40. Signal Correlation

Theory: Calculate the correlation between two signals.



MATLAB Code:

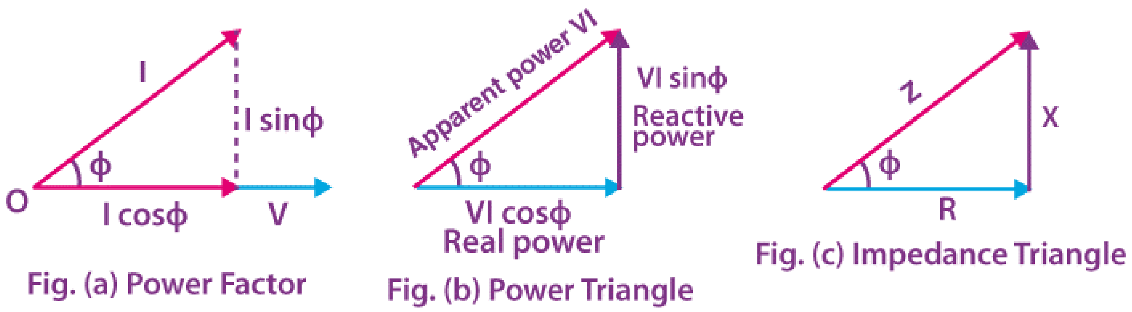
```
% Signal Correlation
x = [1 2 3 4];
y = [4 3 2 1];
correlation = xcorr(x, y);
stem(correlation);
title('Signal Correlation');
xlabel('Lag');
ylabel('Amplitude');
```

41. Power Factor Calculation

Theory: Calculate the power factor of an AC circuit.

Power factor is defined as the cosine of the angle between voltage and current. Ideally, in AC circuits, the phase difference between voltage and current is zero. But, practically there exists some phase difference between the two. The cosine of the phase difference between the two is defined as the power factor.

Power Factor Calculation



In the figure, angle ϕ is the phase difference between the voltage and current phasor. Therefore, the power factor is:

Power Factor = cosϕ

In the figure of the power triangle,

VI sinϕ = Reactive power (in VAR)

VI cosϕ = Active power (in Watts)

VI = Apparent power (in VA)

PF = cosϕ = Active Power (W) / Apparent Power (VA)

<u>Knowledge Review</u>

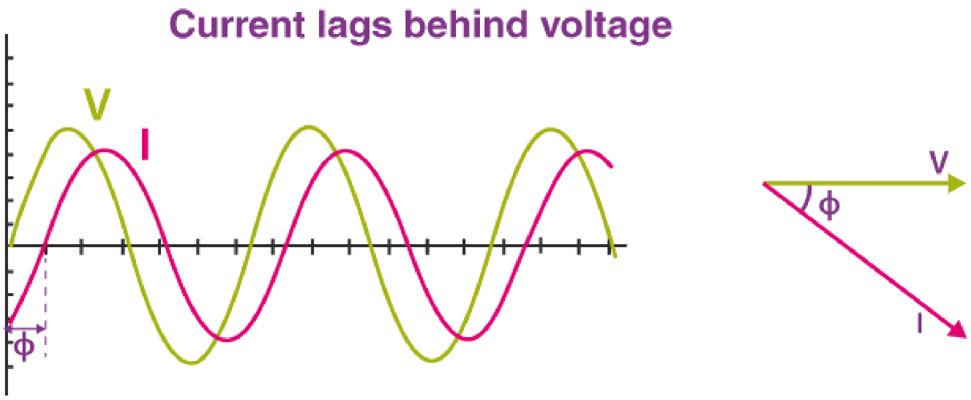
Real Power – The power supplied to the equipment that performs useful and productive work.

Reactive Power – The power required by equipment such as transformers and motors to produce magnetic fields enabling actual work to be done.

Apparent Power – The vector sum of real power and reactive power

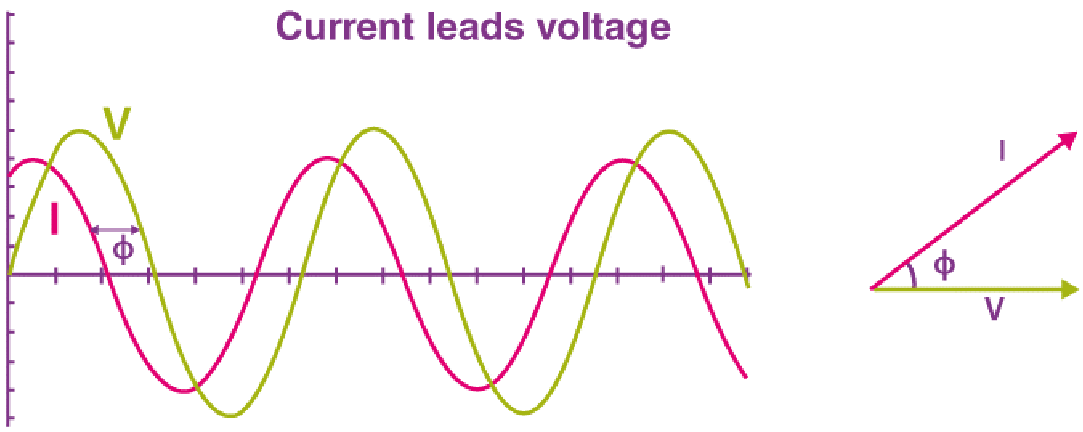
Power Factor can be Lagging, Leading or Unity

Lagging Power Factor



In a circuit, when the current lags behind the voltage, then the power factor of the circuit is known as a lagging power factor. The power factor lags when the circuit is inductive. Loads such as coils, motors and lamps are inductive and have lagging pf.

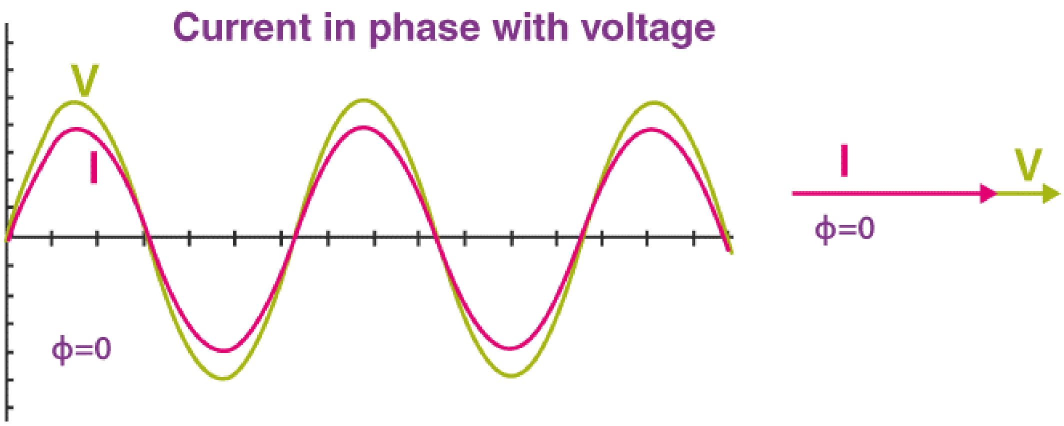
Leading Power Factor



When the current in the circuit leads the voltage, then the power factor of the circuit is said to be leading. A capacitive circuit has a leading power factor. Capacitor banks and Synchronous condensers are capacitive loads that have a leading power factor.

Unity Power Factor

The power factor is unity for ideal circuits. The power factor is unity when the current and voltage are in phase.



Similar Reading:

MATLAB Code:

```
% Power Factor Calculation
P = 100; % Real power in watts
Q = 75; % Reactive power in VAR
S = sqrt(P^2 + Q^2); % Apparent power in VA
power_factor = P / S;
disp(['Power Factor: ', num2str(power_factor)]);
```

42. Step Response of a Control System

Theory: Analyze the step response of a control system.

The **step response** of a system in a given initial state consists of the time evolution of its outputs when its control inputs are Heaviside step functions. In electronic engineering and control theory, step response is the time behaviour of the outputs of a general system when its inputs change from zero to one in a very short time. The concept can be extended to the abstract mathematical notion of a dynamical system using an evolution parameter.
From a practical standpoint, knowing how the system responds to a sudden input is important because large and possibly fast deviations from the long term steady state may have extreme effects on the component itself and on other portions of the overall system dependent on this component. In addition, the overall system cannot act until the component's output settles down to some vicinity of its final state, delaying the overall system response. Formally, knowing the step response of a dynamical system gives information on the stability of such a system, and on its ability to reach one stationary state when starting from another.

MATLAB Code:

```
% Step Response of a Control System
sys = tf([1], [1 3 3 1]);
step(sys);
title('Step Response of a Control System');
```

43. Nyquist Stability Criterion

Theory: Analyze the stability of a system using the Nyquist stability criterion.

Nyquist Stability Criterion
The Nyquist stability criterion works on the **principle of argument**. It states that if there are P poles and Z zeros are enclosed by the 's' plane closed path, then the corresponding
$G(s)H(s)$
plane must encircle the origin $P-Z$
times. So, we can write the number of encirclements N as,
$N=P-Z$

- If the enclosed 's' plane closed path contains only poles, then the direction of the encirclement in the $G(s)H(s)$

 plane will be opposite to the direction of the enclosed closed path in the 's' plane.
 If the enclosed 's' plane closed path contains only zeros, then the direction of the encirclement in the $G(s)H(s)$

- plane will be in the same direction as that of the enclosed closed path in the 's' plane.

Let us now apply the principle of argument to the entire right half of the 's' plane by selecting it as a closed path. This selected path is called the **Nyquist** contour.
We know that the closed loop control system is stable if all the poles of the closed loop transfer function are in the left half of the 's' plane. So, the poles of the closed loop transfer function are nothing but the roots of the characteristic equation. As the order of the characteristic equation increases, it is difficult to find the roots. So, let us correlate these roots of the characteristic equation as follows.

- The Poles of the characteristic equation are same as that of the poles of the open loop transfer function.
- The zeros of the characteristic equation are same as that of the poles of the closed loop transfer function.

We know that the open loop control system is stable if there is no open loop pole in the the right half of the 's' plane.
i.e.,$P=0 \Rightarrow N=-Z$
We know that the closed loop control system is stable if there is no closed loop pole in the right half of the 's' plane.
i.e.,$Z=0 \Rightarrow N=P$
Nyquist stability criterion states the number of encirclements about the critical point (1+j0) must be equal to the poles of characteristic equation, which is nothing but the poles of the open loop transfer function in the right half of the 's' plane. The shift in origin to (1+j0) gives the characteristic equation plane.

MATLAB Code:

```
% Nyquist Stability Criterion
sys = tf([1], [1 3 3 1]);
nyquist(sys);
title('Nyquist Stability Criterion');
```

44. Fourier Series Approximation

Theory: Approximate a periodic function using Fourier series.

A Fourier series is an expansion of a periodic function f(x) in terms of an infinite sum of sines and cosines. Fourier Series makes use of the orthogonality relationships of the sine and cosine functions.

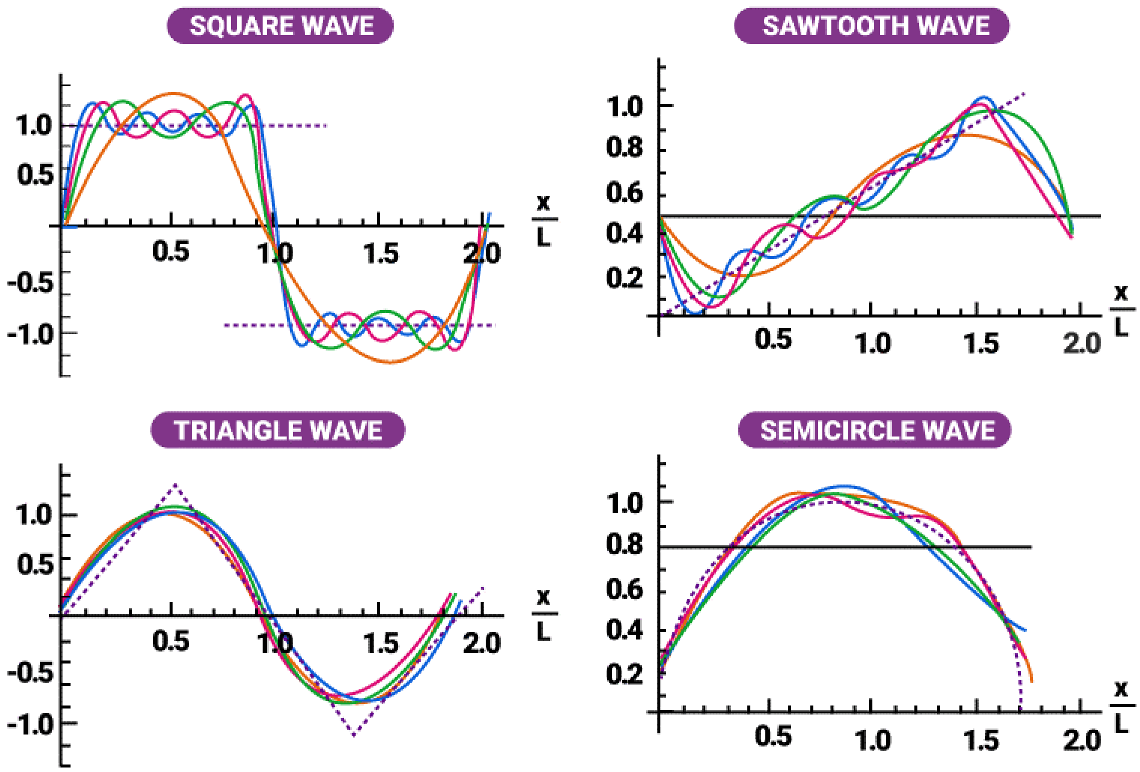
Laurent Series yield Fourier Series

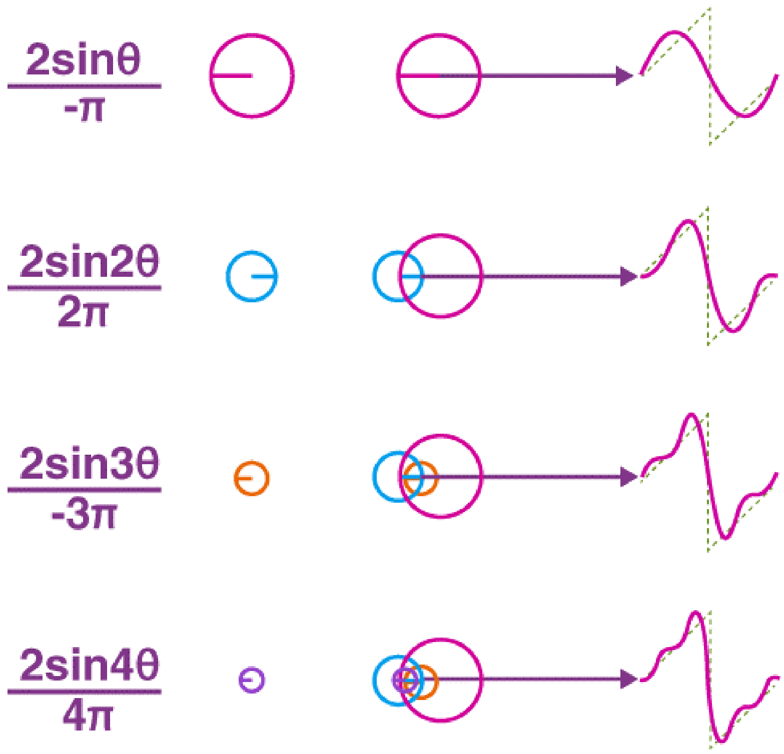
A difficult thing to understand and/or motivate is the fact that arbitrary periodic functions have Fourier series representations. In this section, we prove that periodic analytic functions have such a representation using Laurent expansions.

Fourier Analysis for Periodic Functions

The Fourier series representation of analytic functions is derived from Laurent expansions. The elementary complex analysis is used to derive additional fundamental results in the harmonic analysis including the representation of $C\infty$ periodic functions by Fourier series, the representation of rapidly decreasing functions by Fourier integrals, and Shannon's sampling theorem. The ideas are classical and of transcendent beauty.

Bangabandhu Sheikh Mujibur Rahman Aviation and Aerospace University (BSMRAAU)
Modelling and Simulation Sessional





A function is periodic of period L if f(x+L) = f(x) for all x in the domain of f. The smallest positive value of L is called the fundamental period.
The trigonometric functions sin x and cos x are examples of periodic functions with fundamental period 2π and tan x is periodic with fundamental period π. A constant function is a periodic function with arbitrary period L.
It is easy to verify that if the functions f1, . . . , fn are periodic of period L, then any linear combination
is also periodic. Furthermore, if the infinite series
consisting of 2L-periodic functions converges for all x, then the function to which it converges will be periodic of period 2L. There are two symmetry properties of functions that will be useful in the study of Fourier series.

Even and Odd Function

A function f(x) is said to be even if f(−x) = f(x).
The function f(x) is said to be odd if f(−x) = −f(x).
Graphically, even functions have symmetry about the y-axis, whereas odd functions have symmetry around the origin.

MATLAB Code:

```
% Fourier Series Approximation
T = 2*pi; % Period
t = linspace(0, T, 1000); % Time vector
n_terms = 10; % Number of terms in the series
square_wave = 0;
for n = 1:2:n_terms
    square_wave = square_wave + (4/pi)*(sin(n*t)/n);
end
plot(t, square_wave);
title('Fourier Series Approximation of Square Wave');
xlabel('Time');
ylabel('Amplitude');
```

45. Transmission Line Reflection Coefficient

Theory: Calculate the reflection coefficient of a transmission line.
MATLAB Code:

```
% Transmission Line Reflection Coefficient
Z0 = 50; % Characteristic impedance
ZL = 100; % Load impedance
reflection_coefficient = (ZL - Z0) / (ZL + Z0);
disp(['Reflection Coefficient: ', num2str(reflection_coefficient)]);
```

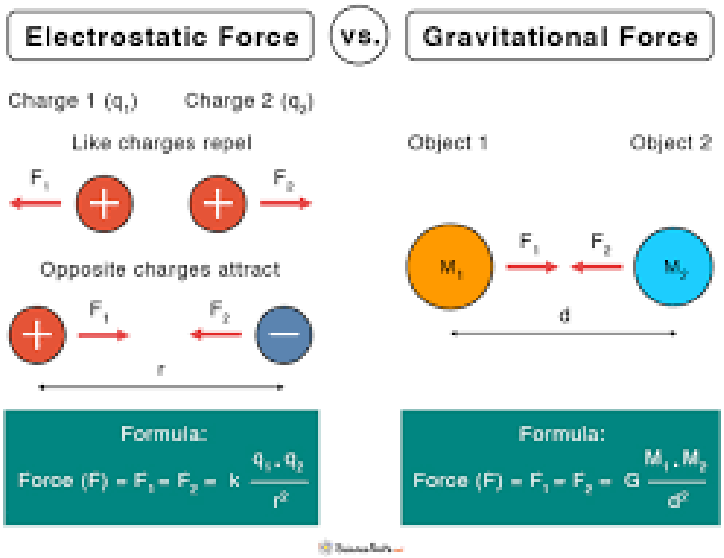
46. State Space Representation

Theory: Represent a system using state space equations.
MATLAB Code:

```
% State Space Representation
A = [0 1; -1 -1];
B = [0; 1];
C = [1 0];
D = 0;
sys = ss(A, B, C, D);
step(sys);
title('State Space Representation');
```

47. Electrostatic Force Calculation

Theory: Calculate the electrostatic force between two charges.

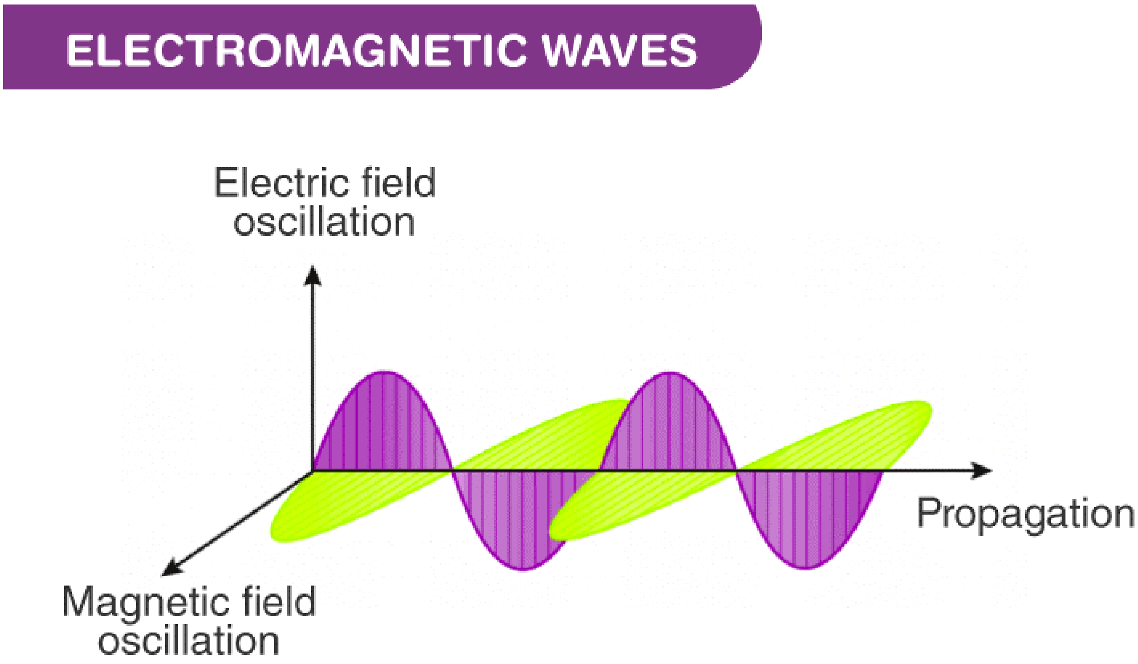


MATLAB Code:

```
% Electrostatic Force Calculation
q1 = 1e-6; % Charge 1 in coulombs
q2 = 2e-6; % Charge 2 in coulombs
r = 0.01; % Distance between charges in meters
epsilon_0 = 8.854e-12; % Permittivity of free space
F = (q1 * q2) / (4 * pi * epsilon_0 * r^2); % Force in newtons
disp(['Electrostatic Force: ', num2str(F), ' N']);
```

48. Electromagnetic Wave Propagation

Theory: Simulate the propagation of an electromagnetic wave.



MATLAB Code:

```
% Electromagnetic Wave Propagation
c = 3e8; % Speed of light in m/s
f = 1e9; % Frequency in Hz
lambda = c / f; % Wavelength in meters
x = linspace(0, lambda, 1000); % Space vector
E = sin(2 * pi * x / lambda); % Electric field
plot(x, E);
title('Electromagnetic Wave Propagation');
xlabel('Distance (m)');
ylabel('Electric Field (V/m)');
```

49. Frequency Response of a System

Theory: Analyze the frequency response of a system.

A frequency response describes the steady-state response of a system to sinusoidal inputs of varying frequencies and lets control engineers analyze and design control systems in the frequency domain.
To understand why the frequency domain is important consider an acoustic guitar. If we place a microphone close to its soundboard and pluck one of the strings (Fig. 1. left), the vibrating string will resonate in the guitar cavity and produce a sound wave that is captured

by the microphone. Looking at the time trace of the captured signal (Fig. 1, right) it is difficult to quickly extract information about what is going on.

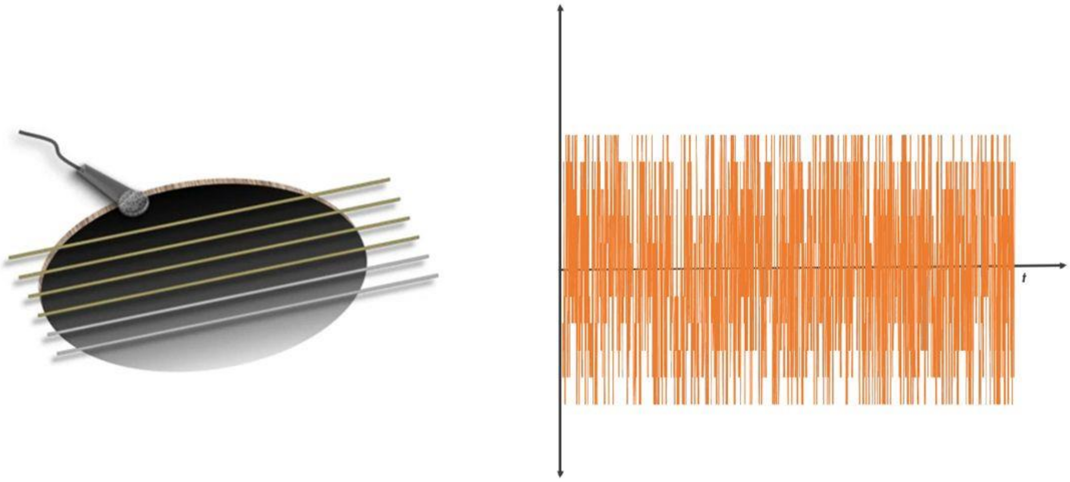


Figure 1: The vibration resonates in the guitar cavity and produces a sound wave (left). Time trace of signal in the time domain (right).

When we look at that same signal in the frequency domain on a spectrum analyzer or by taking a fast Fourier transform (FFT) of the time domain signal, we see an amplitude peak at some frequency (Fig. 2, left). This peak frequency is the underlying tone that forms the note we just played. When we adjust the tuner knob or press the string to the neck of the guitar, we change the preload or the effective length of that string. This will shift the frequency at which the string resonates up or down, and we produce a different note (Fig. 2, right). With this simple analysis in the frequency domain, we can see how the guitar (the system) responds to plucking (the system input).

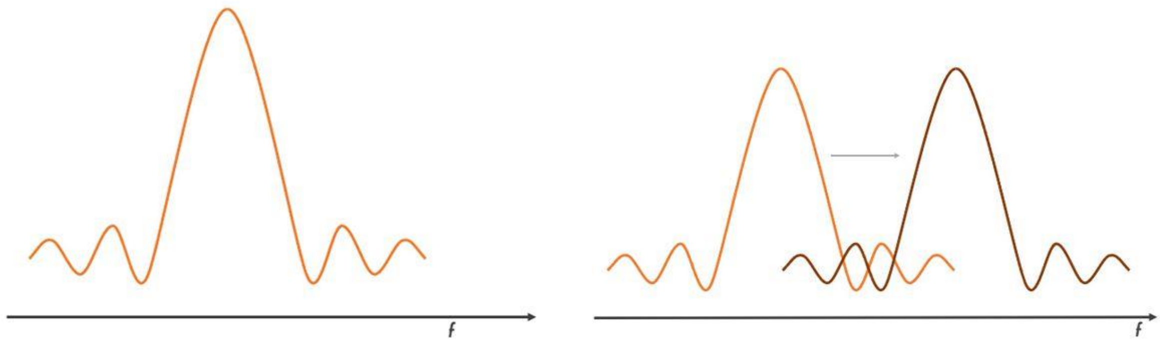


Figure 2: The same signal is shown in the frequency domain (left). Shifting the resonant frequency of the string by preloading it (right).

This analogy can be carried over to other systems where we are interested in the system's response to inputs or stimuli from the environment. We can get insights into the system dynamics such as frequency of a resonant peak, DC gain, bandwidth, phase delay, and phase and gain margins for a closed-loop system.

Getting the Frequency Response of a System

The chart below helps identify an approach (shown in gray) to obtain the frequency response of a system using MATLAB® and Simulink®.

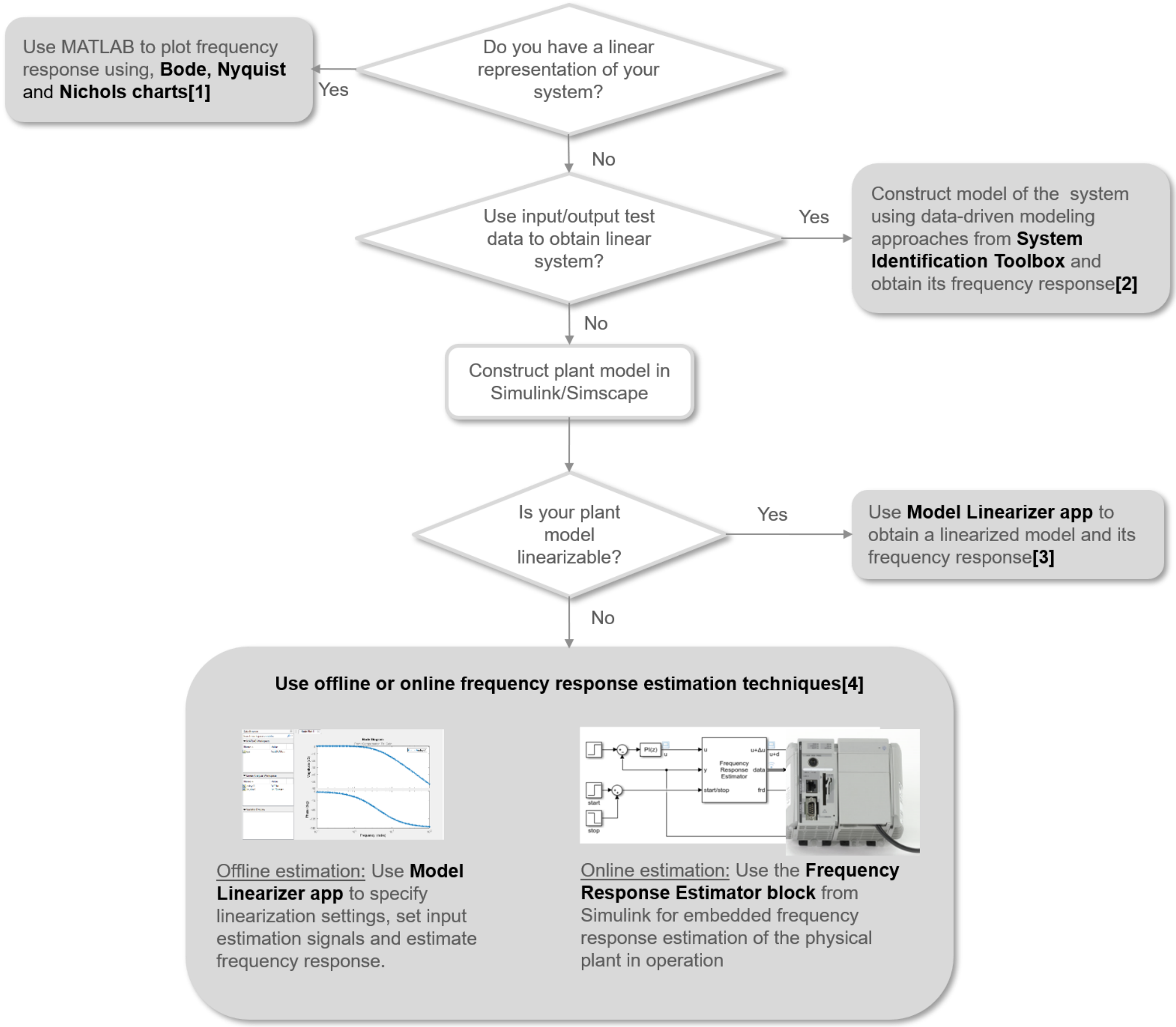


Figure 3: Getting a frequency response for your system using MATLAB and Simulink.

1. If you have a linear representation of the system in the form of a transfer function or state-space model, you can plot the frequency response using one of the three plots: a Bode plot, Nyquist plot, or a Nichols chart. The Bode plot displays magnitude and phase as functions of the frequency of the excitation signal (Fig. 4).

For example, given the transfer function representation of a system (*H*)

,

$H(s)=s2+0.1s+7.5s4+0.12s3+9s2$.

you can plot its frequency response in MATLAB using the following commands:

H=*tf*([10.17.5],[10.12900]);

bode(*H*)

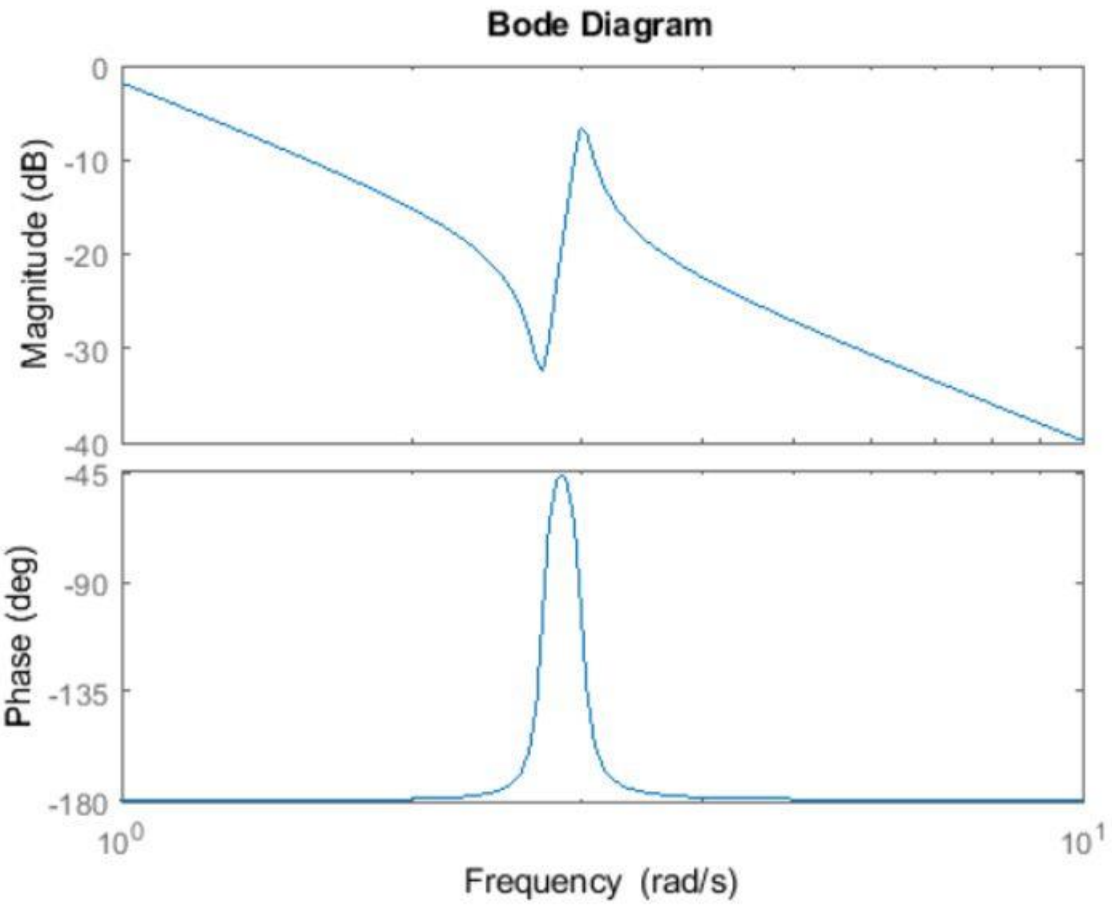


Figure 4: Bode plot.

In some situations, a linear representation of the system might not be available.

2. In that case, if you have access to input-output test data from the physical system, you can use data-driven modeling approaches with System Identification Toolbox™ to identify transfer function, state-space representations, and frequency response models of the system.
3. If you use Simulink to model the system dynamics, you can use the Model Linearizer app in Simulink Control Design™ to linearize your model to create a linear state-space approximation of your Simulink model and plot the frequency response.
4. In case the Simulink models cannot linearize due to discontinuities, you can use frequency response estimation to directly estimate a frequency response model.

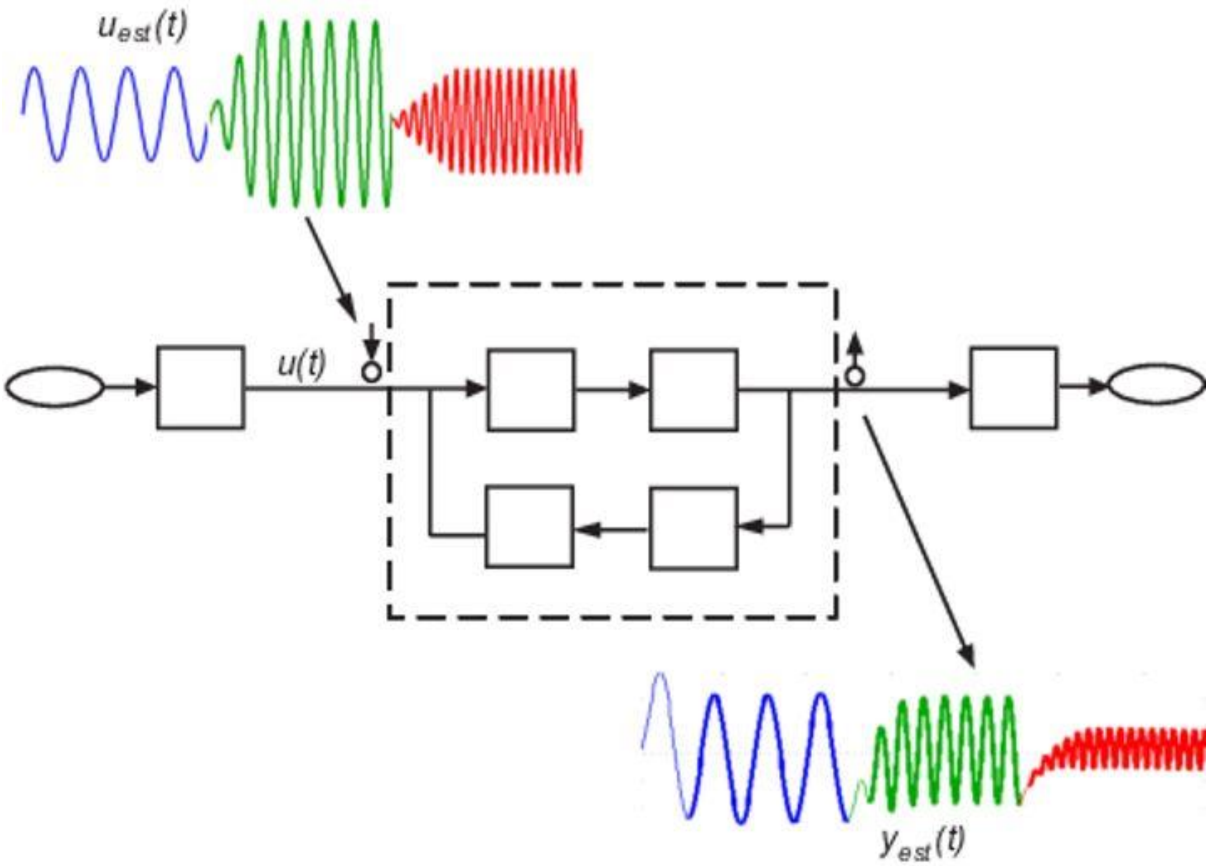


Figure 5: Frequency response estimation in Simulink.

MATLAB Code:

```
% Frequency Response of a System
sys = tf([1], [1 3 3 1]);
bode(sys);
title('Frequency Response');
```

50. Pulse Width Modulation (PWM)

Theory: Simulate pulse width modulation.

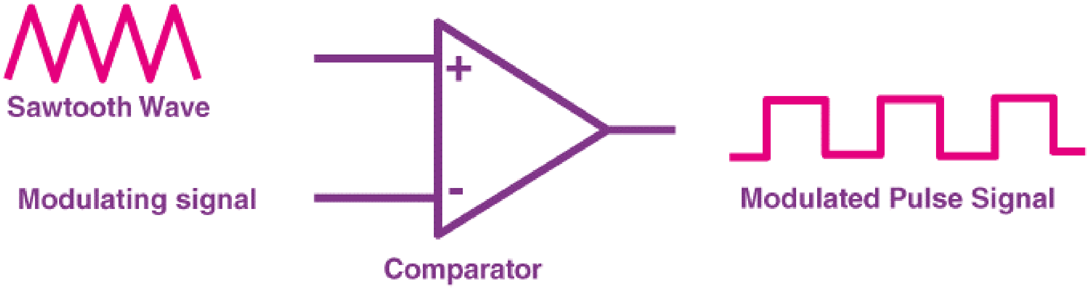
In power electronics, pulse width modulation is a proven effective technique that is used to control semiconductor devices. Pulse width modulation or PWM is a commonly used control technique that generates analog signals from digital devices such as microcontrollers. The signal thus produced will have a train of pulses, and these pulses will be in the form of square waves. Thus, at any given time, the wave will either be high or low. Let us learn more about pulse width modulation in this article.

What is Pulse Width Modulation?

Pulse width modulation reduces the average power delivered by an electrical signal by converting the signal into discrete parts. In the PWM technique, the signal's energy is distributed through a series of pulses rather than a continuously varying (analogue) signal.

How is a Pulse Width Modulation Signal generated?

A pulse width modulating signal is generated using a comparator. The modulating signal forms one part of the input to the comparator, while the non-sinusoidal wave or sawtooth wave forms the other part of the input. The comparator compares two signals and generates a PWM signal as its output waveform.



If the sawtooth signal is more than the modulating signal, then the output signal is in a "High" state. The value of the magnitude determines the comparator output which defines the width of the pulse generated at the output.

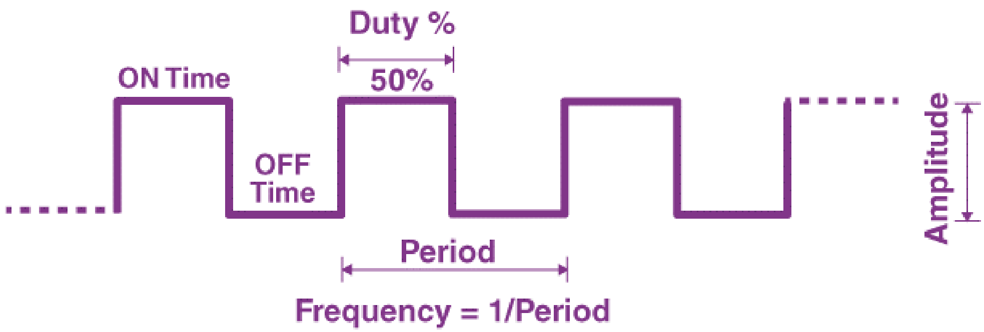
Important Parameters associated with PWM signal

Duty Cycle of PWM

As we know, a PWM signal stays "*ON*" for a given time and stays "*OFF*" for a certain time. The percentage of time for which the signal remains "*ON*" is known as the duty cycle. If the signal is always "*ON*," then the signal must have a 100 % duty cycle. The formula to calculate the duty cycle is given as follows:

The average value of the voltage depends on the duty cycle. As a result, the average value can be varied by controlling the width of the "ON" of a pulse.

Frequency of PWM



The frequency of PWM determines how fast a PWM completes a period. The frequency of a pulse is shown in the figure above.

The frequency of PWM can be calculated as follows:

Frequency = 1/Time Period
Time Period = On Time + OFF time

Output Voltage of PWM signal

The output voltage of the PWM signal will be the percentage of the duty cycle. For example, for a 100% duty cycle, if the operating voltage is 5 V then the output voltage will also be 5 V. If the duty cycle is 50%, then the output voltage will be 2.5 V.

Types of Pulse Width Modulation Technique

There are three conventional types of pulse width modulation technique and they are named as follows:

- **Trail Edge Modulation –** In this technique, the signal's lead edge is modulated, and the trailing edge is kept fixed.
- **Lead Edge Modulation –** In this technique, the signal's lead edge is fixed, and the trailing edge is modulated.
- **Pulse Center Two Edge Modulation –** In this technique, the pulse centre is fixed and both edges of the pulse are modulated.

Applications of Pulse Width Modulation

Due to the high efficiency, low power loss, and the PWM technique's ability to precisely control the power, the technique is used in a variety of power applications. Some of the applications of PWM are as follows:

- The pulse width modulation technique is used in telecommunication for encoding purposes.
- The PWM helps in voltage regulation and therefore is used to control the speed of motors.
- The PWM technique controls the fan inside a CPU of the computer, thereby successfully dissipating the heat.
- PWM is used in Audio/Video Amplifiers.

Advantages and Disadvantages of Pulse Width Modulation

First, let us talk about the advantages of the pulse width modulation technique before discussing its disadvantages:

- PWM technique prevents overheating of LED while maintaining its brightness.
- PWM technique is accurate and has a fast response time.
- PWM technique provides a high input power factor.
- PWM technique helps motors generate maximum torque even when they run at lower speeds.

The above were some of the advantages of the Pulse width modulation technique, now let us look at some of the disadvantages:

- As the PWM frequency is high, switching losses are considerably high.
- It induces Radio Frequency Interference (RFI).

MATLAB Code:

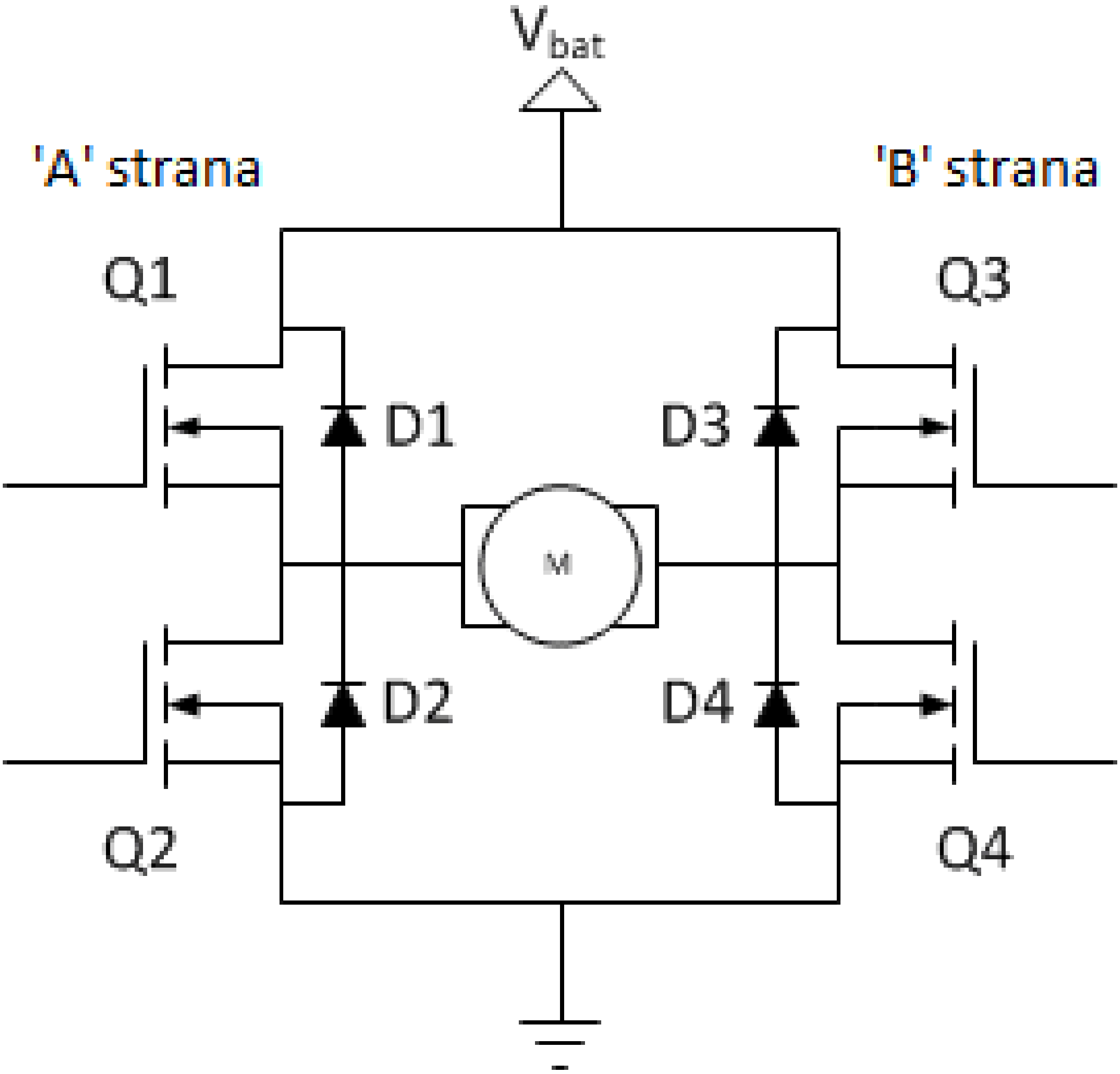
```
% Pulse Width Modulation (PWM)
Fs = 1000; % Sampling frequency
t = 0:1/Fs:1-1/Fs;
duty_cycle = 0.5; % Duty cycle
```

```
pwm_signal = square(2*pi*10*t, duty_cycle*100); % PWM signal
plot(t, pwm_signal);
title('Pulse Width Modulation (PWM)');
xlabel('Time (s)');
ylabel('Amplitude');
```

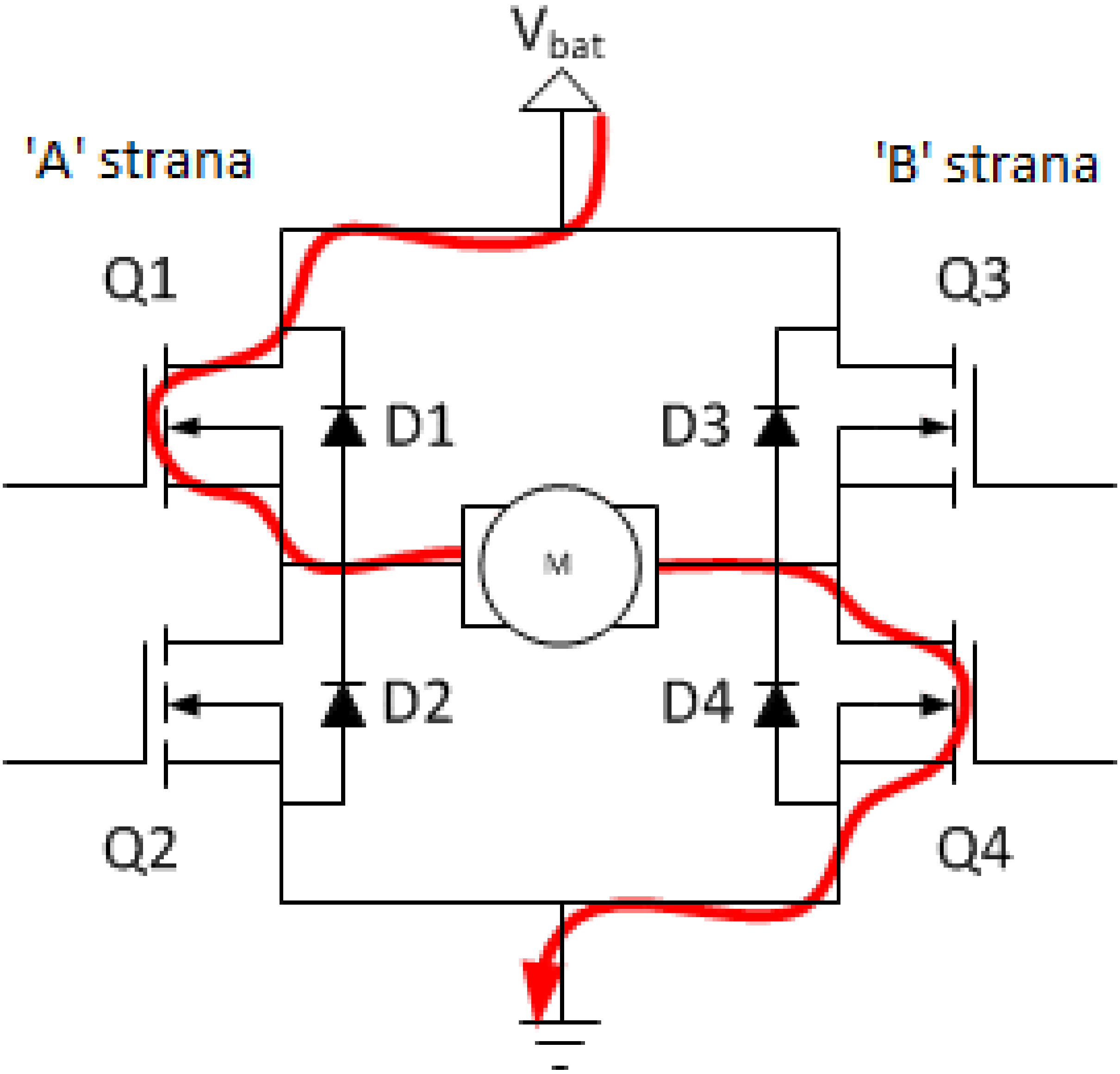
51. H-Bridge Circuit Simulation

Theory: Simulate an H-Bridge circuit for DC motor control.

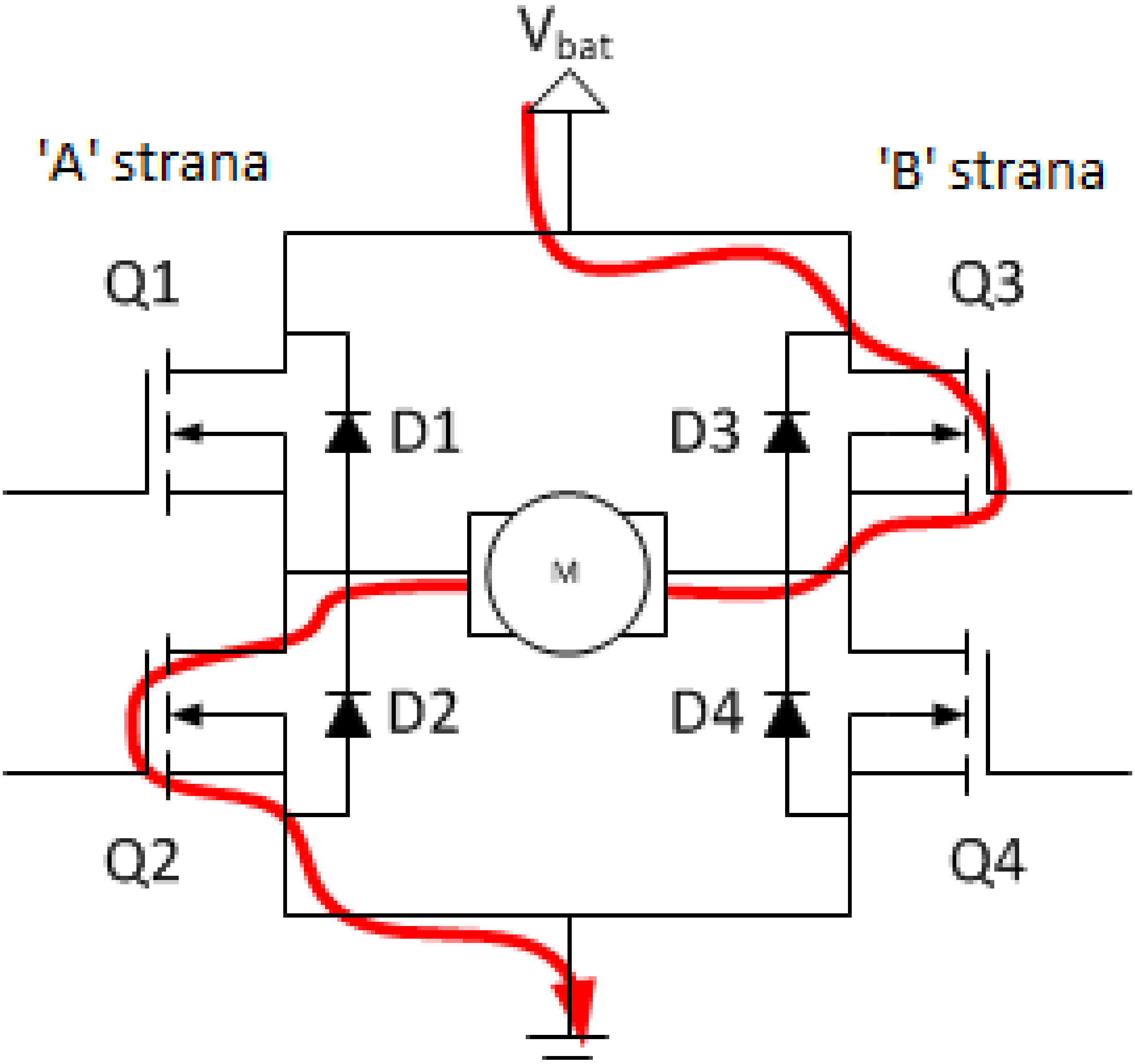
H-bridge is a circuit that, in theory, consists of four switches connected to some load (e.g. DC motor). Various configurations of these switches can allow us to control current flow in the circuit. Using the H-bridge, it is very simple to switch the polarity on the load. Even though the load can theoretically be whatever you want, by far, the most widely used H-bridge application is for DC and stepper motors. It is most commonly used to control and replace the DC motor rotation direction. H-bridge is also used in many other applications such as DC/AC, AC/AC or DC/DC converters. The H-bridge circuit reminds of the letter H, and that is how it has gotten its name.



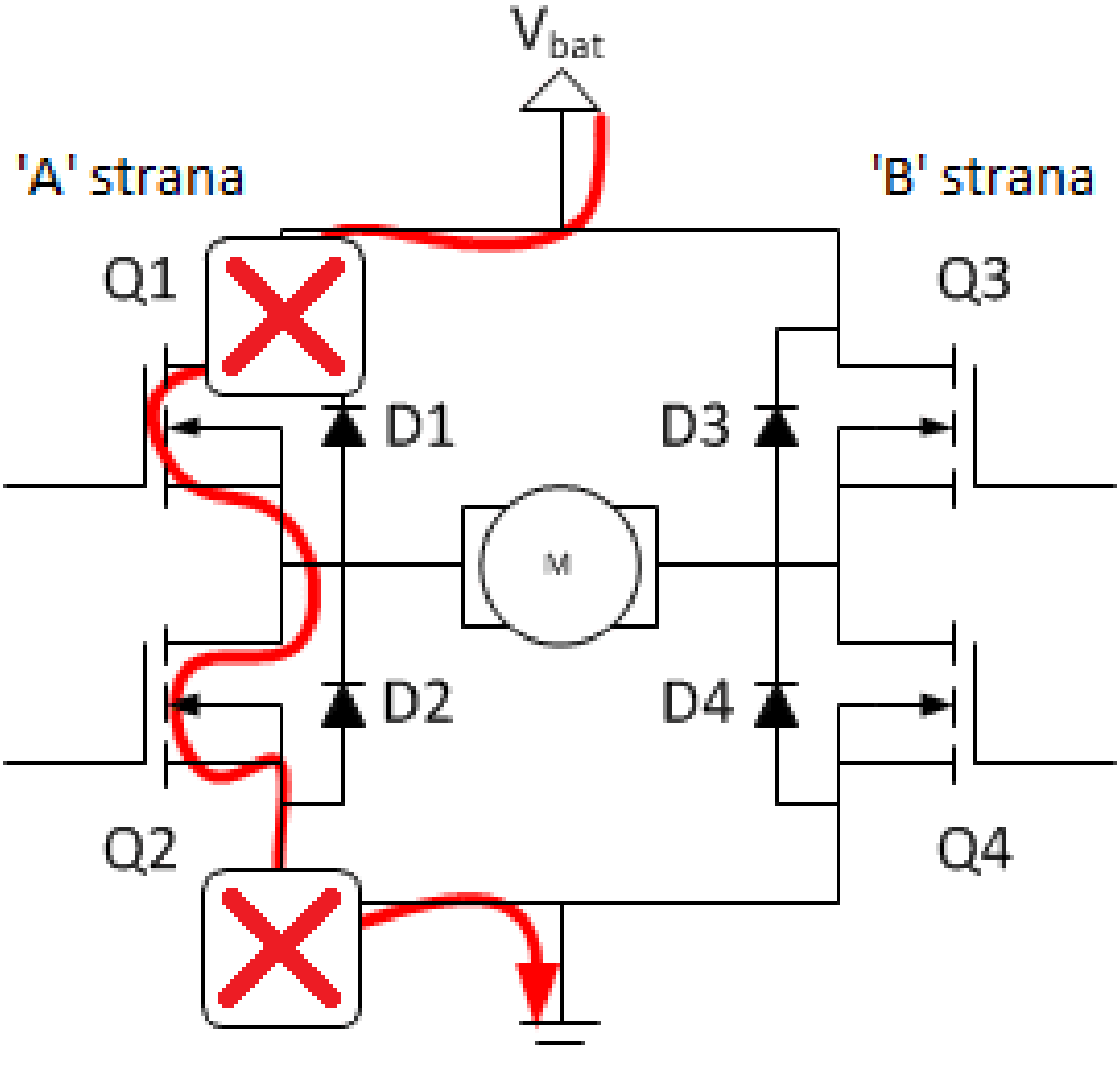
The image above represents a scheme display of the H-bridge. We can see that the H-bridge consists of four MOSFETs connected in a certain way. Because of the inability to simultaneously switch between the two possible states of the bridge, diodes were added to the circuit (mostly Schottky's diodes) in order for it to ensure simultaneous switch of states and limit the current's flow power during these short periods of imbedding, without the load voltage being too high. The upper end of the bridge is connected to a power source (e.g. battery), while the bottom end is grounded. The basic work principle of the H-bridge is very simple: if **Q1** and **Q4** are turned on, the left cable of the motor will be connected to the power supply, and the right to the ground. The current flows through the motor (so to say) in the forward direction, and the engine shaft starts rotating.



If **Q2** and **Q3** are on, what will happen is quite the opposite, the motor is powered in the reversed direction, and the shaft will rotate in the opposite direction.



You must never close **Q1** and **Q2** (or **Q3** and **Q4**) at the same time! If that happens, a direct path between the power supply voltage and ground which is of very low resistance is created, which ultimately results in a short circuit. A short circuit can lead to destruction of the H-bridge, or some other component in the circuit.



The table below shows every possible combination of switch states and their outcome.

Q1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Q2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
Q3	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
Q4	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
Rezultat	Nema promjene	Nema promjene	Nema promjene	Kratki spoj	Nema promjene	Režim kočenja	Motor se okreće u lijevu stranu	Kratki spoj	Nema promjene	Motor se okreće u desnu stranu	Režim kočenja	Kratki spoj	Kratki spoj	Kratki spoj	Kratki spoj	Kratki spoj

MATLAB Code:

```
% H-Bridge Circuit Simulation
V_dc = 12; % DC supply voltage in volts
R_load = 10; % Load resistance in ohms
t = 0:0.001:1; % Time vector
```

```
V_out = V_dc * square(2*pi*10*t); % Output voltage with PWM control
plot(t, V_out);
title('H-Bridge Circuit Simulation');
xlabel('Time (s)');
ylabel('Voltage (V)');
```

52. PID Controller Design

Theory: Design and simulate a PID controller.

Proportional-Integral-Derivative (PID) control is the most common control algorithm used in industry and has been universally accepted in industrial control. The popularity of PID controllers can be attributed partly to their robust performance in a wide range of operating conditions and partly to their functional simplicity, which allows engineers to operate them in a simple, straightforward manner.

As the name suggests, PID algorithm consists of three basic coefficients; proportional, integral and derivative which are varied to get optimal response. Closed loop systems, the theory of classical PID and the effects of tuning a closed loop control system are discussed in this paper. The PID toolset in LabVIEW and the ease of use of these VIs is also discussed.

Contents

Control System

The basic idea behind a PID controller is to read a sensor, then compute the desired actuator output by calculating proportional, integral, and derivative responses and summing those three components to compute the output. Before we start to define the parameters of a PID controller, we shall see what a closed loop system is and some of the terminologies associated with it.

Closed Loop System

In a typical control system, the *process variable* is the system parameter that needs to be controlled, such as temperature (ºC), pressure (psi), or flow rate (liters/minute). A sensor is used to measure the process variable and provide feedback to the control system. The *set point* is the desired or command value for the process variable, such as 100 degrees Celsius in the case of a temperature control system. At any given moment, the difference between the process variable and the set point is used by the control system algorithm *(compensator)*, to determine the desired actuator output to drive the system (plant). For instance, if the measured temperature process variable is 100 ºC and the desired temperature set point is 120 ºC, then the *actuator output* specified by the control algorithm might be to drive a heater. Driving an actuator to turn on a heater causes the system to become warmer, and results in an increase in the temperature process variable. This is called a closed loop control system, because the process of reading sensors to provide constant feedback and calculating the desired actuator output is repeated continuously and at a fixed loop rate as illustrated in figure 1.

In many cases, the actuator output is not the only signal that has an effect on the system. For instance, in a temperature chamber there might be a source of cool air that sometimes blows into the chamber and disturbs the temperature.Such a term is referred to as *disturbance*. We usually try to design the control system to minimize the effect of disturbances on the process variable.

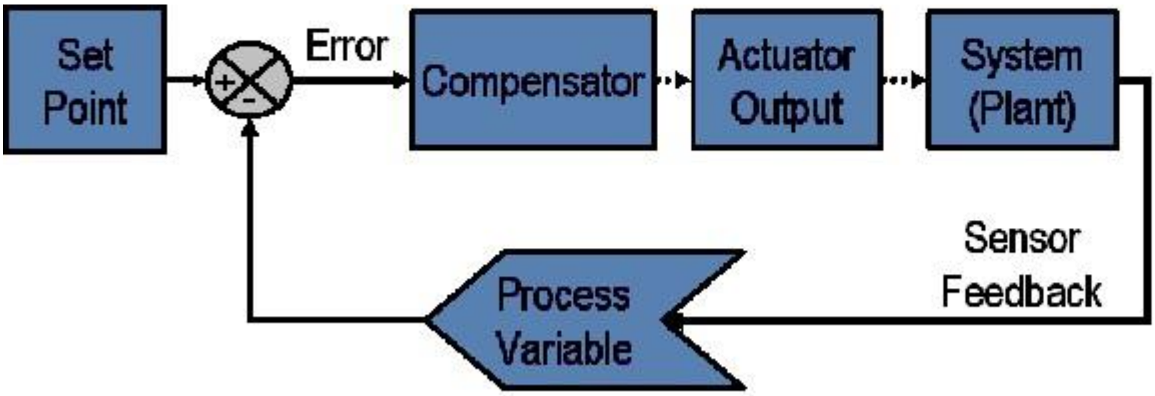
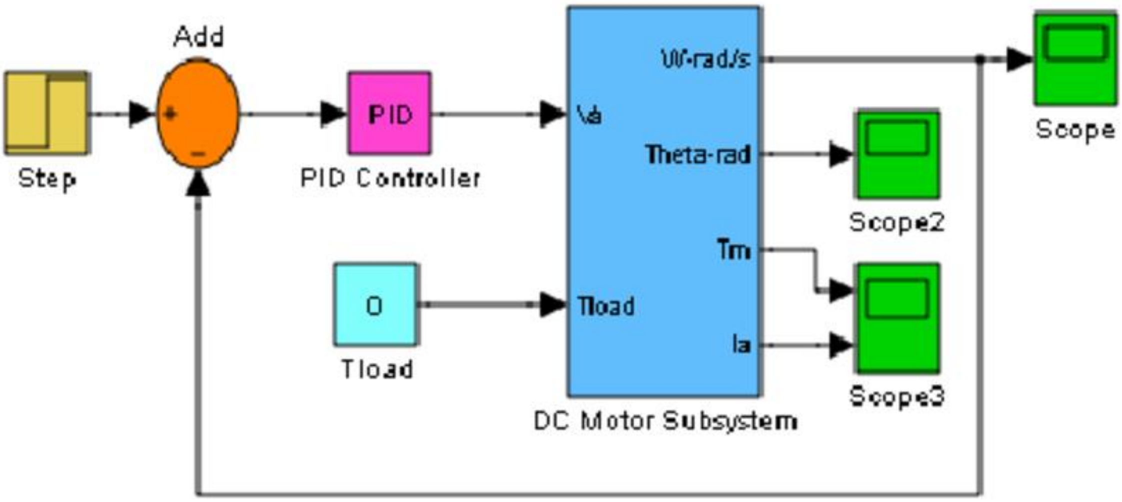


Figure 1: Block diagram of a typical closed loop system.

MATLAB Code:

```
% PID Controller Design
Kp = 1; % Proportional gain
Ki = 1; % Integral gain
Kd = 0.1; % Derivative gain
sys = tf([1], [1 3 3 1]); % Plant transfer function
pid_controller = pid(Kp, Ki, Kd);
closed_loop_system = feedback(pid_controller * sys, 1);
step(closed_loop_system);
title('PID Controller Step Response');
```

53. Motor Speed Control Using PID
Theory: Simulate the speed control of a DC motor using a PID controller.



MATLAB Code:

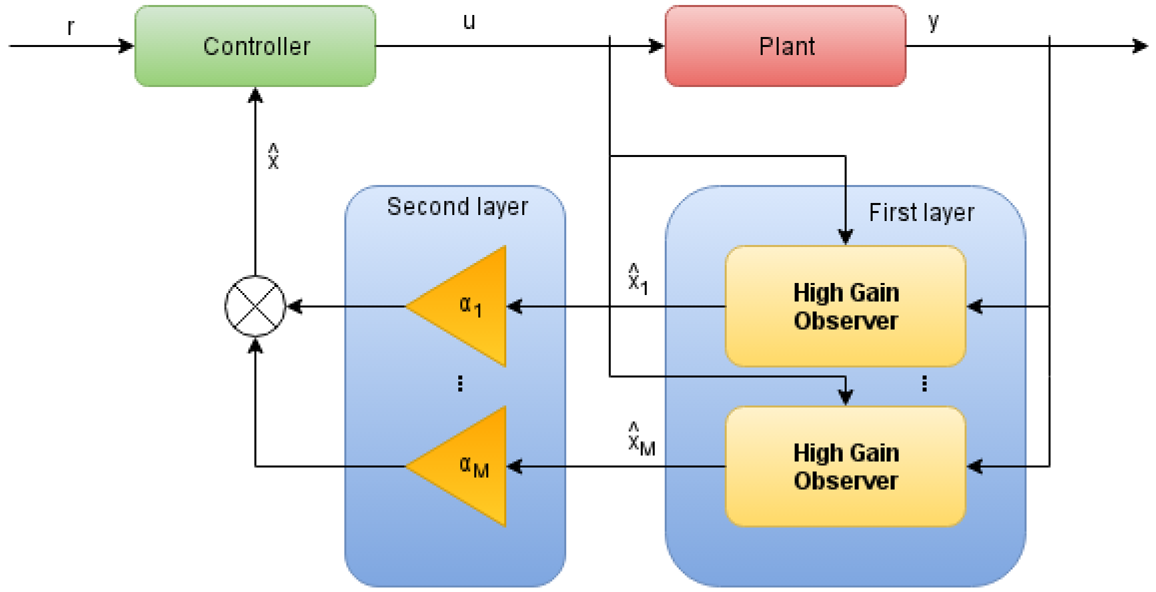
```
% Motor Speed Control Using PID
J = 0.01; % Moment of inertia of the rotor
b = 0.1; % Damping ratio
K = 0.01; % Motor constant
R = 1; % Electrical resistance
L = 0.5; % Electrical inductance

% Transfer function of the DC motor
s = tf('s');
P_motor = K / (J*L*s^2 + (J*R + L*b)*s + (b*R + K^2));

% PID controller design
Kp = 100;
Ki = 200;
Kd = 10;
C = pid(Kp, Ki, Kd);

% Closed-loop transfer function
sys_cl = feedback(C*P_motor, 1);

% Simulation
t = 0:0.01:2;
step(sys_cl, t);
title('DC Motor Speed Control Using PID');
xlabel('Time (seconds)');
ylabel('Speed (rad/s)');
```

54. Phase Margin and Gain Margin

Theory: Determine the phase margin and gain margin of a control system.
MATLAB Code:

```
% Phase Margin and Gain Margin
sys = tf([1], [1 3 3 1]);
margin(sys);
title('Phase Margin and Gain Margin');
```

55. State Observer Design



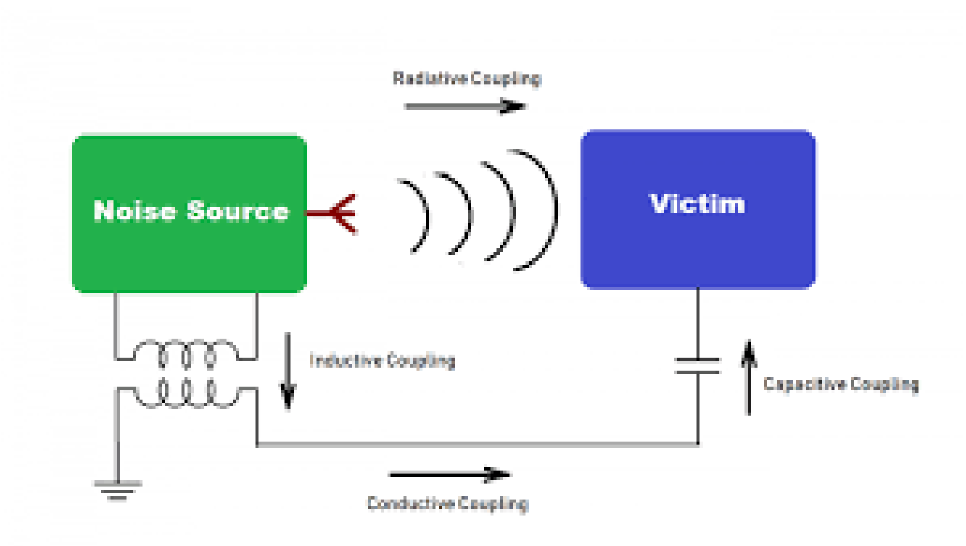
Theory: Design a state observer for a given system.
MATLAB Code:

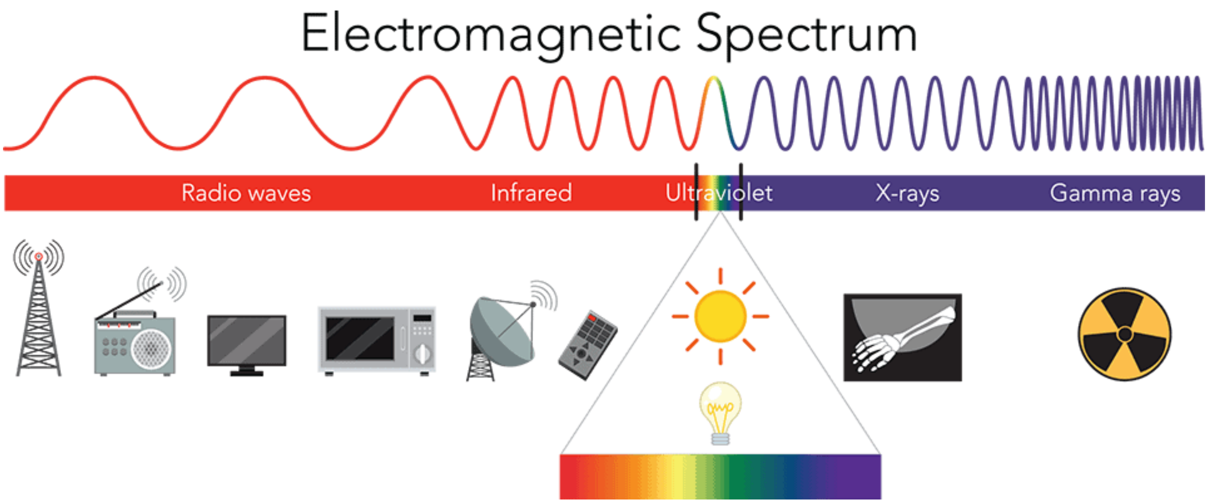
```
% State Observer Design
A = [0 1; -2 -3];
B = [0; 1];
C = [1 0];
D = 0;
sys = ss(A, B, C, D);

% Desired poles for the observer
poles = [-5 -6];
L = place(A', C', poles)';

% Observer simulation
t = 0:0.01:5;
u = ones(size(t));
x0 = [0; 0];
[y, ~, x] = lsim(sys, u, t, x0);

% State estimation
x_hat = zeros(size(x));
for i = 2:length(t)
    x_hat(:, i) = x_hat(:, i-1) + 0.01 * (A*x_hat(:, i-1) + B*u(i) + L*(y(i-1) - C*x_hat(:, i-1)));
end

plot(t, x(1, :), 'b', t, x_hat(1, :), 'r--');
legend('True State', 'Estimated State');
```

```
title('State Observer Design');
xlabel('Time (s)');
ylabel('State');
```

56. Electromagnetic Interference (EMI) Simulation

Theory: Simulate the effects of electromagnetic interference on a signal.





MATLAB Code:

```
% Electromagnetic Interference (EMI) Simulation
Fs = 1000; % Sampling frequency
t = 0:1/Fs:1-1/Fs;
signal = sin(2*pi*50*t); % Original signal
EMI = 0.5 * sin(2*pi*200*t); % Interference signal
```

```
noisy_signal = signal + EMI;
plot(t, signal, 'b', t, noisy_signal, 'r');
legend('Original Signal', 'Noisy Signal');
title('Electromagnetic Interference (EMI) Simulation');
xlabel('Time (s)');
ylabel('Amplitude');
```

57. Transmission Line Impedance Matching

Theory: Design an impedance matching network for a transmission line.

Impedance is defined as the total resistance of a given electric component or circuit to an alternating current originating from the reactance and resistance of the given system. Unlike resistance, which has magnitude only, impedance has both phase and magnitude in alternating current circuits. For direct current, there is no difference between resistance and impedance as the impedance has zero phase angle, just like resistance has no phase angle. We can define impedance matching as the process where the input impedance and the output impedance of a given electrical load are designed to reduce signal reflection and maximize the power transferred to the electric load. For a better understanding of impedance matching, let’s look at the maximum power transfer theorem.

Maximum Power Transfer Theorem

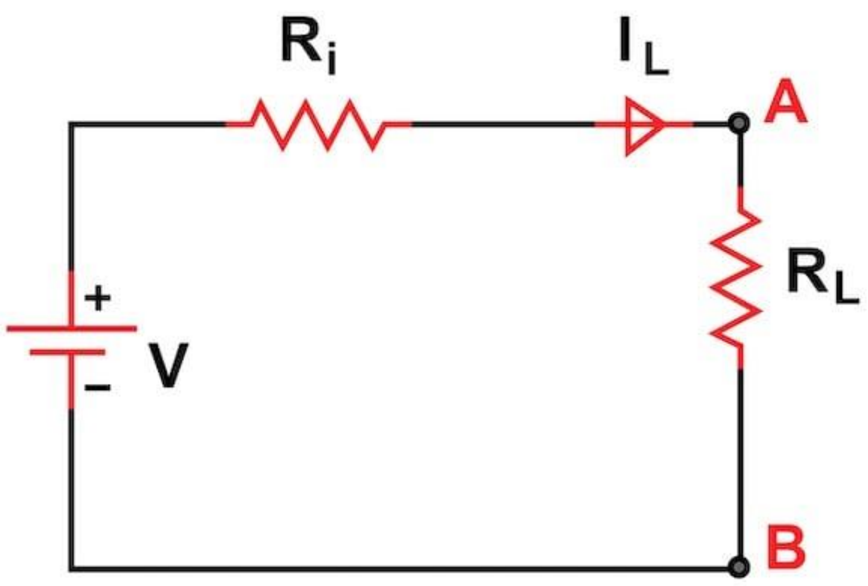


Figure 1. Maximum Power Transfer Theorem Circuit. Image used courtesy of Simon Mugo

Suppose we have a system with a voltage source V of internal resistance R_i and powering an electric load of resistance R_L. The maximum power theorem will be used to determine the value of the load resistance $R_{L,\ allowing}$ maximum power to be transferred from the source to the load. The maximum power transferred to the load depends on the size of the load

resistance.
From the circuit above, power transferred to the load resistance is
P=I2LRL=V2RL(Ri+RL)2
Figure 2. Equation of Power Transferred to the Load.

For maximum power, we differentiate the above equation to the load resistance R_L and equate the outcome to zero. We shall have
dPdRL=V2(Ri+RL)2−2RL(Ri+RL)V2(Ri+RL)4=0⇒RL=Ri
Figure 3. Differentiated Equation and Equated to Zero.

Note that maximum power can only be transferred from the source to the load when the internal resistance of the voltage source is equal to the resistance of the load. Impedance matching ensures that the source resistance is equal to the load resistance. Another thing to note is that the load reactance should also be equal to the negative of the source reactance for maximum power to be reflected at the electric load side. This means the load power can only be at maximum when the load impedance is equal to the source impedance complex conjugate.

Impedance Matching Formulas and Circuits

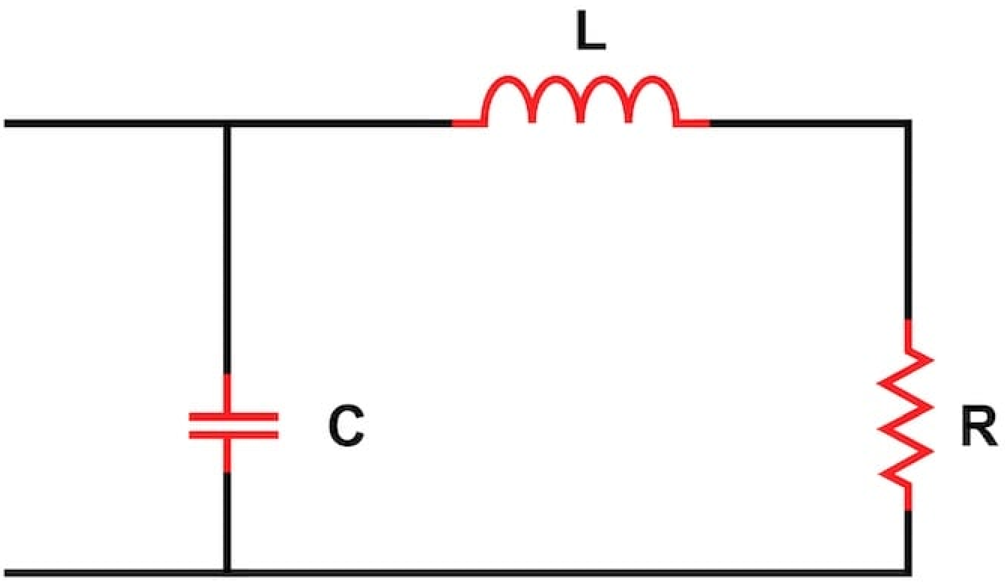


Figure 4. Impedance Matching Circuit. Image used courtesy of Simon Mugo

For an available resistance R, we shall find the circuit that will match the resistance R’ at a certain frequency w_0 and develop a design of the L matching circuit displayed in Figure 4 above.
Let us start by finding the admittance Y_{in} of our circuit above.
From the figure, we can note that the resistance R and the inductor L are in series, and the combination of the two is in parallel with capacitor C.
The impedance will be given by
Z=(R+jωL)//1jωC
Z=[(R+jωL)×1jωC](R+jωL)+1jωC

$Z=R+j\omega L(R+j\omega L)(j\omega C)+1$
$Y_{in}=1Z=[(R+j\omega L)(j\omega C)+1](R+j\omega L)$
$Y_{in}=j\omega C+1(R+j\omega L)$
Let us use complex conjugates to separate the real and imaginary parts of the above equation.
$Y_{in}=j\omega C+1(R+j\omega L)\times(R-j\omega L)(R-j\omega L)$
$Y_{in}=j\omega C+R(R2+(\omega L)2)-j\omega L(R2(\omega L)2)$
Reorganizing the above equation, we get
$Y_{in}=R(R2+(\omega L)2)+j[\omega C-\omega L(R2+(\omega L)2)]$
Finally, we have
$Y=RR2+(\omega L)2$
(1)
And
$\omega 0=\sqrt{1}LC-(RL)2$
(2)
At the frequency of ω= 0, the resistance of Yin should be set to R'.
$R'=1Y=R2+(\omega 0L)2R$
$R'=R+\omega 02L2R$
Separate R from the equation to get
$R'=R[1+(\omega 0LR)2]$
Let $\omega 0LR$
be the Q-factor for L and R networks and our equation becomes
$R'=R[1+Q2]$
(3)
From the above equations, it is easy to solve the impedance matching problems in any electrical circuit.

Why Impedance Matching Is Needed

Impedance matching has great use in high-frequency and high-speed devices. When designing printed circuit boards that require such characteristics, ensure that the impedance at the source matches the impedance at the load.
When designing applications of ultra-high frequencies, impedance matching becomes a difficult operation for designers. The challenge is also reflected while designing microwaves and radio frequency circuits. When you get a wrong impedance matching, expect distorted pulses and high signal reflections.
An increase in frequency decreases the window of errors. The electrical circuit works the best when we have a perfectly matched impedance. If the impedance matching is not done, expect the system to work abnormally because of the effects such as the signal reflections. The reflected waves cause data delays and distortion of the phase and minimize the ratio of signal to noise.

Application of Impedance Matching

The goal of any electrical and electronic design engineer is to ensure that there is maximum power delivery from the system source to the electric load. In almost all electrical and electronic applications, impedance matching is a necessity.
Let us have a look at a few applications of impedance matching below.

Transformer Impedance Matching

Transformers are one of the components used to match the impedance of the source to load. The power input of the transformer is similar to the power output by it. The transformer changes the electrical energy c\voltage level and does not affect the power level of the system.
To match the impedance of both sides, you have to set the turn ratio accordingly. Low voltage has fewer turns, and this tells us that the impedance of the low voltage winding is lower when compared to that of the high voltage winding.
Therefore, to ensure that we have matched our impedance, a transformer with proper winding turns is connected between the source and the load. The transformer is known as a matching transformer.
We can define the matching transformer turn ratio as the square root of the ratio of the resistance of the source to the resistance of the load.
TurnsRatio=√SourceResistanceLoadResistance
Figure 5. Transformer Turn Ratio Calculation Formula.

Impedance Matching Transmission Line
Transmission of electrical energy from the source to the load is done using a transmission line. While transferring this energy, it is important to zero or minimize energy losses that occur. For this to be possible, we should match the source and load impedances to the transmission line being used.
The characteristic impedance is defined as the voltage and current wave ratio at any given point along the transmission line. If the transmission line in discussion is long, then we expect to have a different characteristic impedance at different distances along this transmission line. If we fail to do the impedance matching, the signs reaching the load will be reflected in the source of the origin, giving rise to a standing wave. The amount of power reflected is measured using the coefficient of reflection, which is calculated using the equation below:
Γ=ZL−Z0ZL+Z0
Figure 6. Transmission Line Reflection Coefficient Calculating Formula

With Z_L being Line Impedance and Z_0 Characteristic Impedance.
An ideal system has a load impedance similar to the characteristic impedance. The system is said to be ideal when the reflection coefficient is zero. In practice, it is difficult to achieve a zero-reflection coefficient; hence, in the transmission line, the reflective coefficient is kept close to zero.

Antenna Impedance Matching
The antenna needs to be coupled with a television. For this to be achieved, impedance matching is employed. Here, the antenna becomes our source as it has to give signals to the television, whereas the TV becomes the load because it gets signals from the antenna.
Let us take an example where the antenna and its cable have a resistance of 150 ohms, and the resistance of the TV is 600 ohms. These conditions indicate a different impedance between the source and the load, and hence it becomes impossible to have maximum power transmitted; therefore, poor signals are received by the TV.

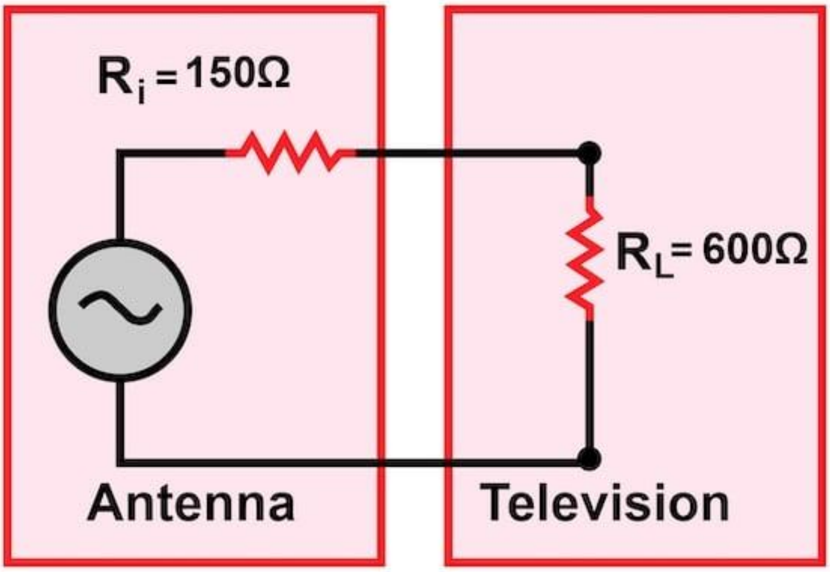


Figure 7. Antenna Impedance Matching. Image used courtesy of Simon Mugo

To avoid the above condition, we have to use a transformer to achieve the antenna and television impedance matching. We had already introduced the formula for the turn ratio of the transformer calculation in this article, and will input the values accordingly, as shown below:

n=√RLRin

n=√600Ω150Ω

n=2

From the calculations above, our turn ratio is 1:2, and we have to connect our transformer as in the circuit below:

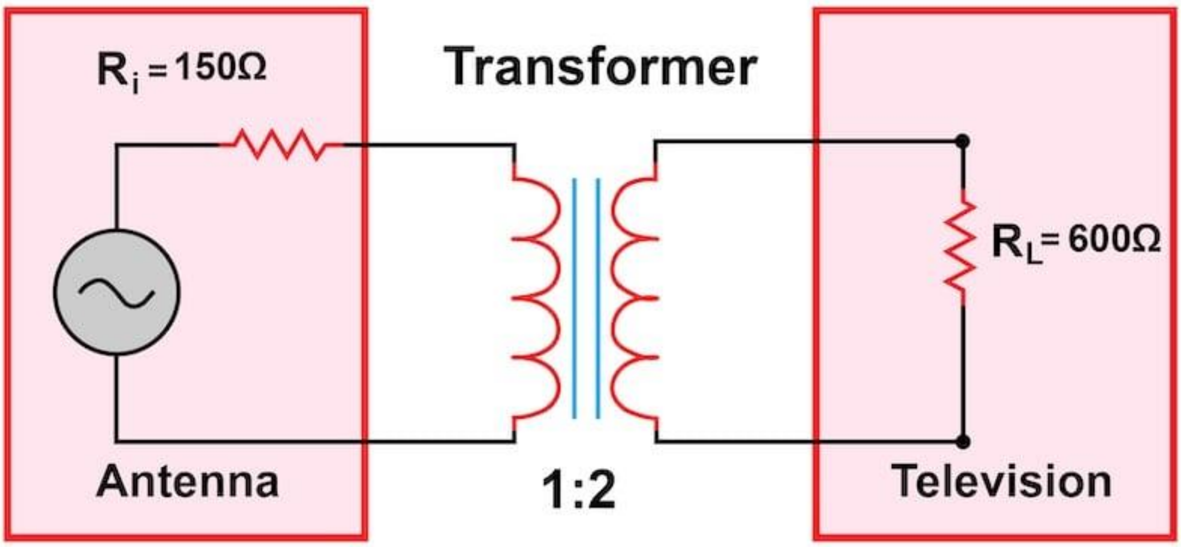


Figure 8. Impedance Matching of Antenna with Transformer. Image used courtesy of Simon Mugo

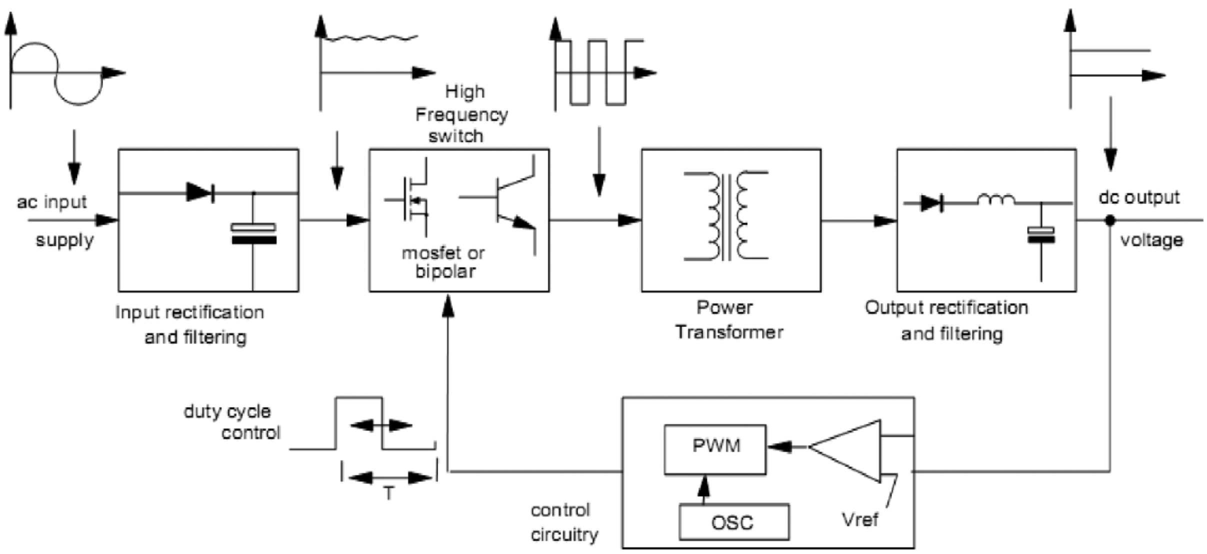
Headphone Impedance Matching
In the case of the headphone, the signal source is the device where the headphone is plugged. The headphone is the load. For the system to attain quality audio output, the source, and the load impedances must be matched. By matching the impedances, we make sure that there is maximum power transfer from the source of the audio to the headphone.

MATLAB Code:

```
% Transmission Line Impedance Matching
Z0 = 50; % Characteristic impedance
ZL = 100; % Load impedance
matching_impedance = sqrt(Z0 * ZL); % Matching impedance
disp(['Matching Impedance: ', num2str(matching_impedance), ' Ohms']);
```

58. Switched-Mode Power Supply (SMPS) Simulation

Theory: Simulate a basic switched-mode power supply.



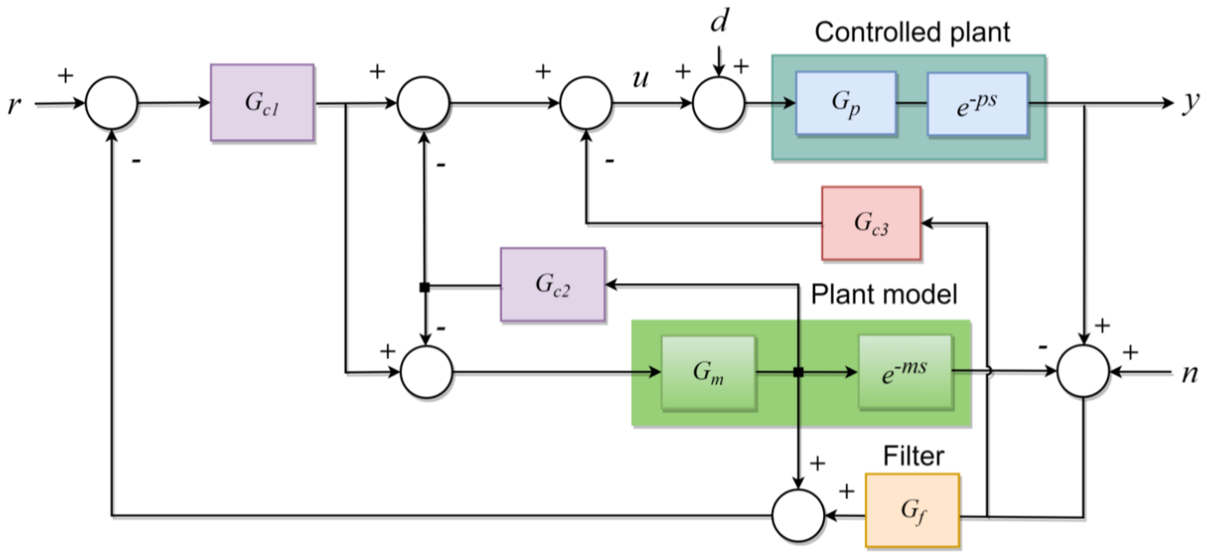
MATLAB Code:

```
% Switched-Mode Power Supply (SMPS) Simulation
V_in = 12; % Input voltage in volts
D = 0.5; % Duty cycle
L = 1e-3; % Inductance in henries
C = 1e-6; % Capacitance in farads
R = 10; % Load resistance in ohms
```

```
t = linspace(0, 0.01, 1000); % Time vector
V_out = V_in * D * (1 - exp(-t / (L / R))); % Output voltage

plot(t, V_out);
title('Switched-Mode Power Supply (SMPS) Simulation');
xlabel('Time (s)');
ylabel('Voltage (V)');
```

59. Smith Predictor Design

Theory: Design and simulate a Smith predictor for a time-delay system.



MATLAB Code:

```
% Smith Predictor Design
G = tf([1], [1 3 3 1], 'InputDelay', 1); % Plant with time delay
G_nom = tf([1], [1 3 3 1]); % Nominal plant
Kp = 1; % Proportional gain
C = pid(Kp); % Controller
smith_predictor = feedback(C*G_nom, 1); % Smith predictor
sys_cl = feedback(smith_predictor, G);

step(sys_cl);
title('Smith Predictor Design');
xlabel('Time (s)');
ylabel('Output');
```

60. Kalman Filter Implementation

Theory: Implement a Kalman filter for state estimation.
A Kalman filter is an optimal estimator - ie infers parameters of interest from indirect, inaccurate and uncertain observations. It is recursive so that new measurements can be processed as they arrive. (cf batch processing where all data must be present).

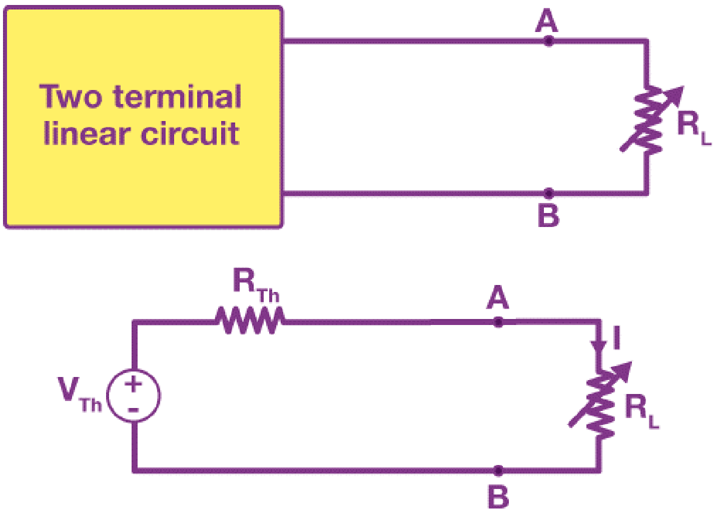
MATLAB Code:

```
% Kalman Filter Implementation
A = [1 1; 0 1]; % State transition matrix
B = [0.5; 1]; % Control input matrix
C = [1 0]; % Measurement matrix
Q = [1 0; 0 1]; % Process noise covariance
R = 1; % Measurement noise covariance
x = [0; 0]; % Initial state
P = eye(2); % Initial state covariance
x_est = x; % Initial state estimate

% Simulation
t = 0:0.1:10;
u = sin(t);
y = zeros(size(t));
x_true = zeros(2, length(t));
x_estimated = zeros(2, length(t));

for k = 1:length(t)
   % True state and measurement
   x = A*x + B*u(k) + sqrt(Q)*randn(2, 1);
   y(k) = C*x + sqrt(R)*randn;
   x_true(:, k) = x;

   % Kalman filter update
   x_pred = A*x_est + B*u(k);
   P_pred = A*P*A' + Q;
   K = P_pred*C' / (C*P_pred*C' + R);
   x_est = x_pred + K * (y(k) - C*x_pred);
   P = (eye(2) - K*C) * P_pred;
   x_estimated(:, k) = x_est;
end

plot(t, x_true(1, :), 'b', t, x_estimated(1, :), 'r--');
legend('True State', 'Estimated State');
title('Kalman Filter Implementation');
xlabel('Time (s)');
ylabel('State');
```

61. Maximum Power Transfer Theorem

Theory: Verify the maximum power transfer theorem in an electrical circuit.
Maximum Power Transfer Theorem explains that to generate maximum external power through a finite internal resistance (DC network), the resistance of the given load must be equal to the resistance of the available source.
In other words, the resistance of the load must be the same as Thevenin's equivalent resistance.
In the case of AC voltage sources, maximum power is produced only if the load impedance's value is equivalent to the complex conjugate of the source impedance.

Maximum Power Transfer Formula
As shown in the figure, a dc source network is connected with variable resistance R_L.



The fundamental Maximum Power Transfer Formula is

MATLAB Code:

```
% Maximum Power Transfer Theorem
R_source = 50; % Source resistance in ohms
R_load = linspace(1, 100, 100); % Load resistance in ohms
V_source = 10; % Source voltage in volts

P_load = (V_source^2) * (R_load ./ (R_source + R_load).^2); % Power delivered to the load

plot(R_load, P_load);
title('Maximum Power Transfer Theorem');
xlabel('Load Resistance (Ohms)');
```

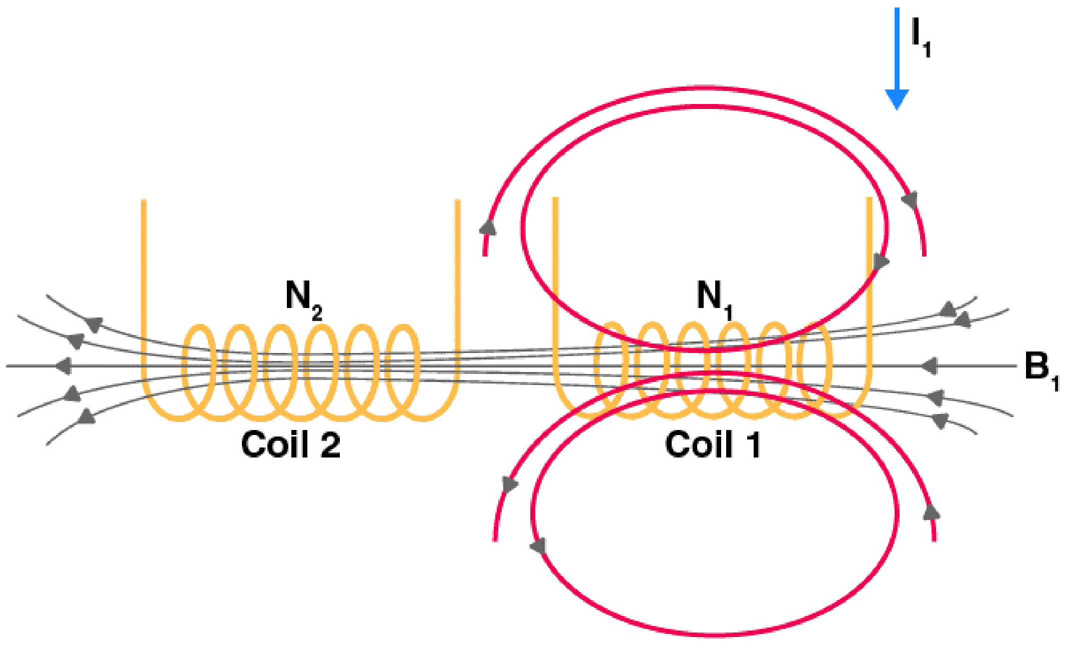
```
ylabel('Power Delivered (Watts)');
```

62. Mutual Inductance Calculation

Theory: Calculate the mutual inductance between two inductors.

Mutual inductance is the main operating principle of generators, motors and transformers. Any electrical device having components that tend to interact with another magnetic field also follows the same principle. The interaction is usually brought about by a mutual induction where the current flowing in one coil generates a voltage in a secondary coil.
Download Complete Chapter Notes of Electromagnetic Induction

What Is Mutual Inductance?
When two coils are brought in proximity to each other, the magnetic field in one of the coils tends to link with the other. This further leads to the generation of voltage in the second coil. This property of a coil which affects or changes the current and voltage in a secondary coil is called mutual inductance.

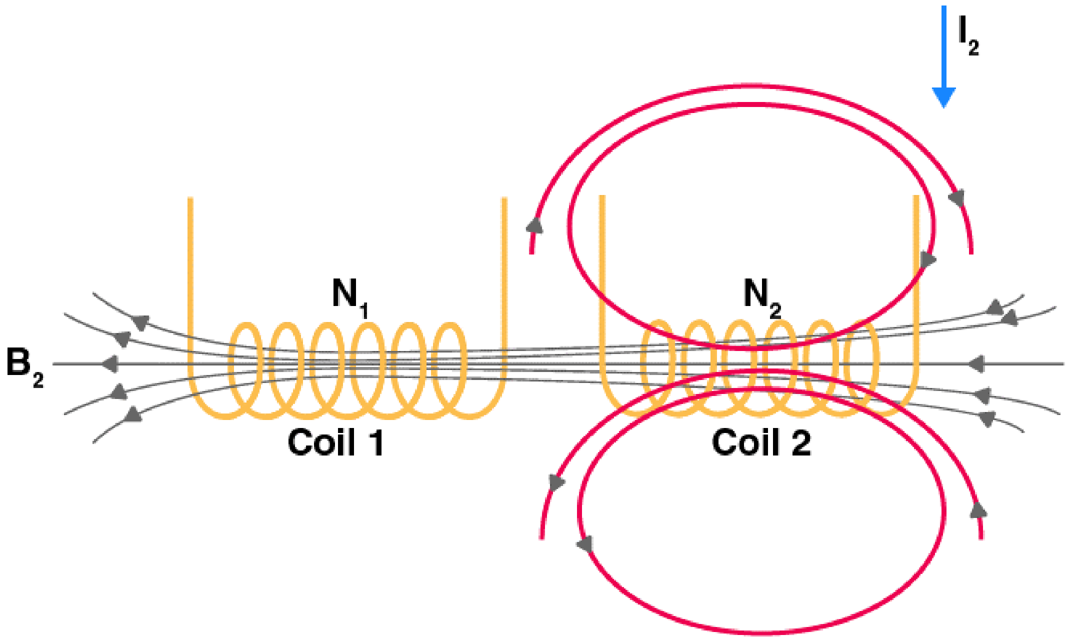


Changing I_1 produces changing magnetic flux in coil 2.
In the first coil of N_1 turns, when a current I_1 passes through it, magnetic field B is produced.
As the two coils are closer to each other, a few magnetic field lines will also pass through coil 2.
If we vary the current with respect to time, then there will be an induced emf in coil 2.
[According to Faraday's law]

The induced emf in coil 2 is directly proportional to the current that passes through coil 1.

The constant of proportionality is called mutual inductance. It can be written as
The SI unit of inductance is henry (H)
In a similar manner, the current in coil 2, I_2, can produce an induced emf in coil 1 when I_2 varies with respect to time. Then,

This constant of proportionality is another mutual inductance.



Changing I_2 produces changing magnetic flux in coil 1.

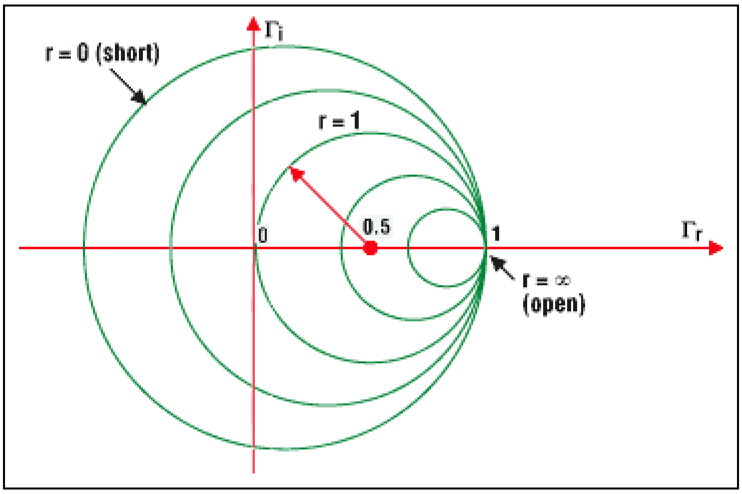
MATLAB Code:

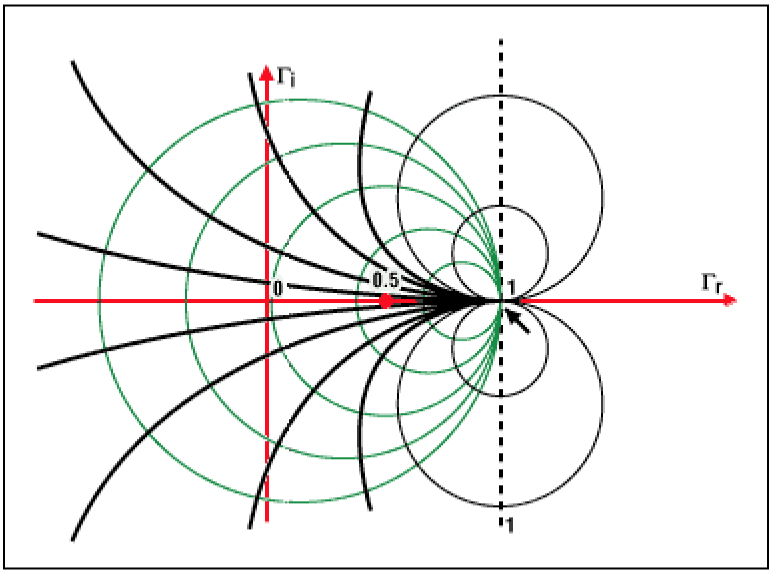
```
% Mutual Inductance Calculation
L1 = 10e-3; % Inductance of coil 1 in henrys
L2 = 20e-3; % Inductance of coil 2 in henrys
k = 0.5; % Coupling coefficient

M = k * sqrt(L1 * L2); % Mutual inductance
disp(['Mutual Inductance: ', num2str(M), ' H']);
```

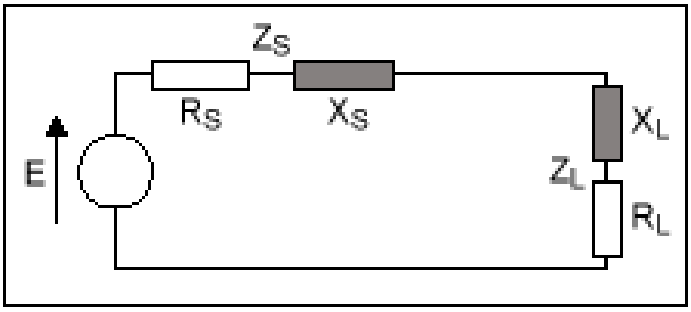
63. Impedance Matching Using Smith Chart

Theory: Design an impedance matching network using a Smith chart.

Bangabandhu Sheikh Mujibur Rahman Aviation and Aerospace University (BSMRAAU)
Modelling and Simulation Sessional





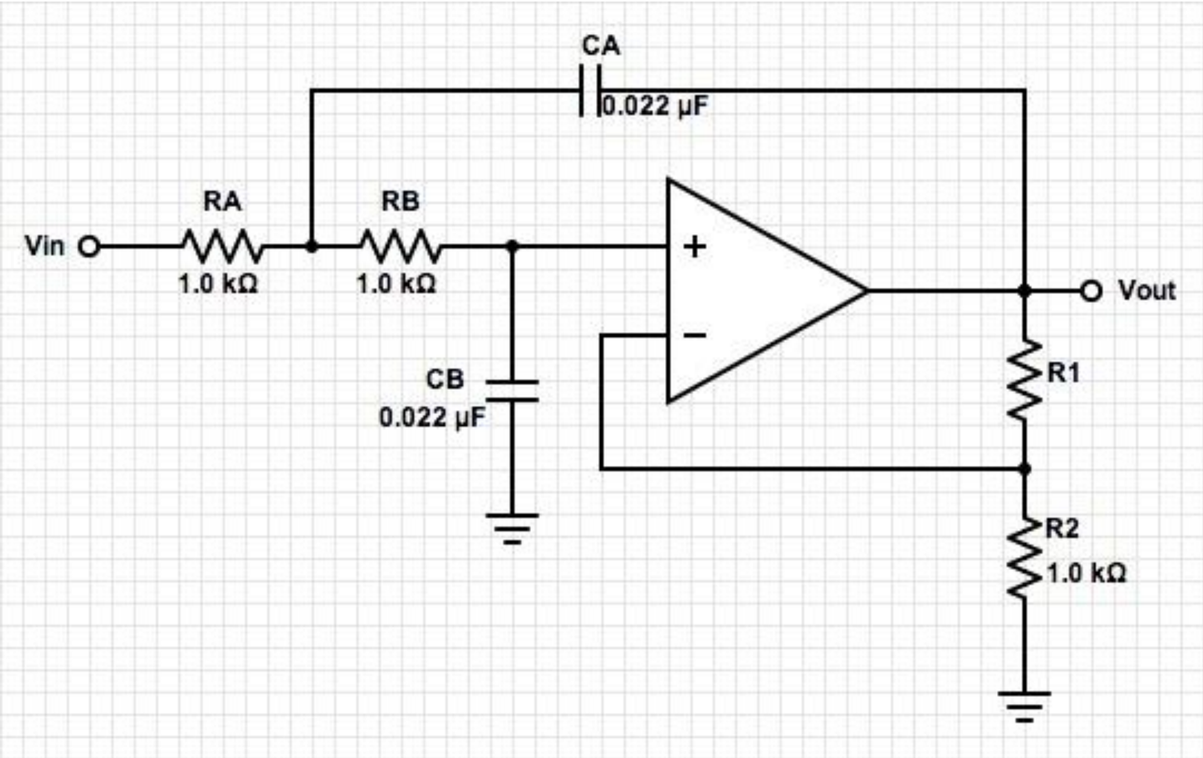


MATLAB Code:

```
% Impedance Matching Using Smith Chart
z_load = 2 + 3j; % Normalized load impedance
smithchart(z_load);
title('Impedance Matching Using Smith Chart');
```

64. Sallen-Key Low Pass Filter

Theory: Design and simulate a Sallen-Key low pass filter.



MATLAB Code:

```
% Sallen-Key Low Pass Filter
R = 1e3; % Resistance in ohms
C = 1e-6; % Capacitance in farads
sys = tf([1], [R^2*C^2 3*R*C 1]);
bode(sys);
title('Sallen-Key Low Pass Filter');
```

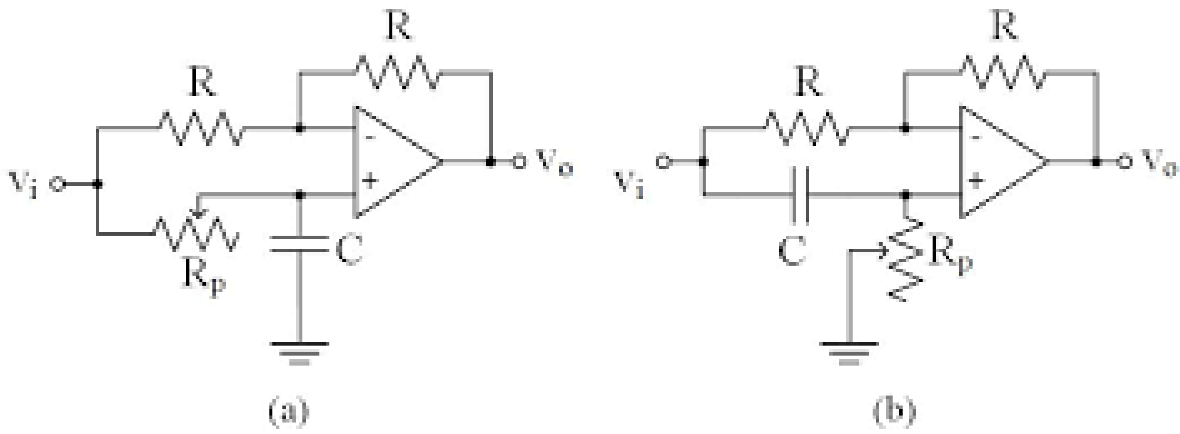
65. Thermistor Resistance Calculation

Theory: Calculate the resistance of a thermistor at a given temperature.
MATLAB Code:

```
% Thermistor Resistance Calculation
T = 25; % Temperature in Celsius
T0 = 25; % Reference temperature in Celsius
R0 = 10e3; % Resistance at T0 in ohms
B = 3950; % Beta coefficient

R = R0 * exp(B * ((1 / (T + 273.15)) - (1 / (T0 + 273.15))));
disp(['Thermistor Resistance: ', num2str(R), ' Ohms']);
```

66. Phase Shifter Circuit

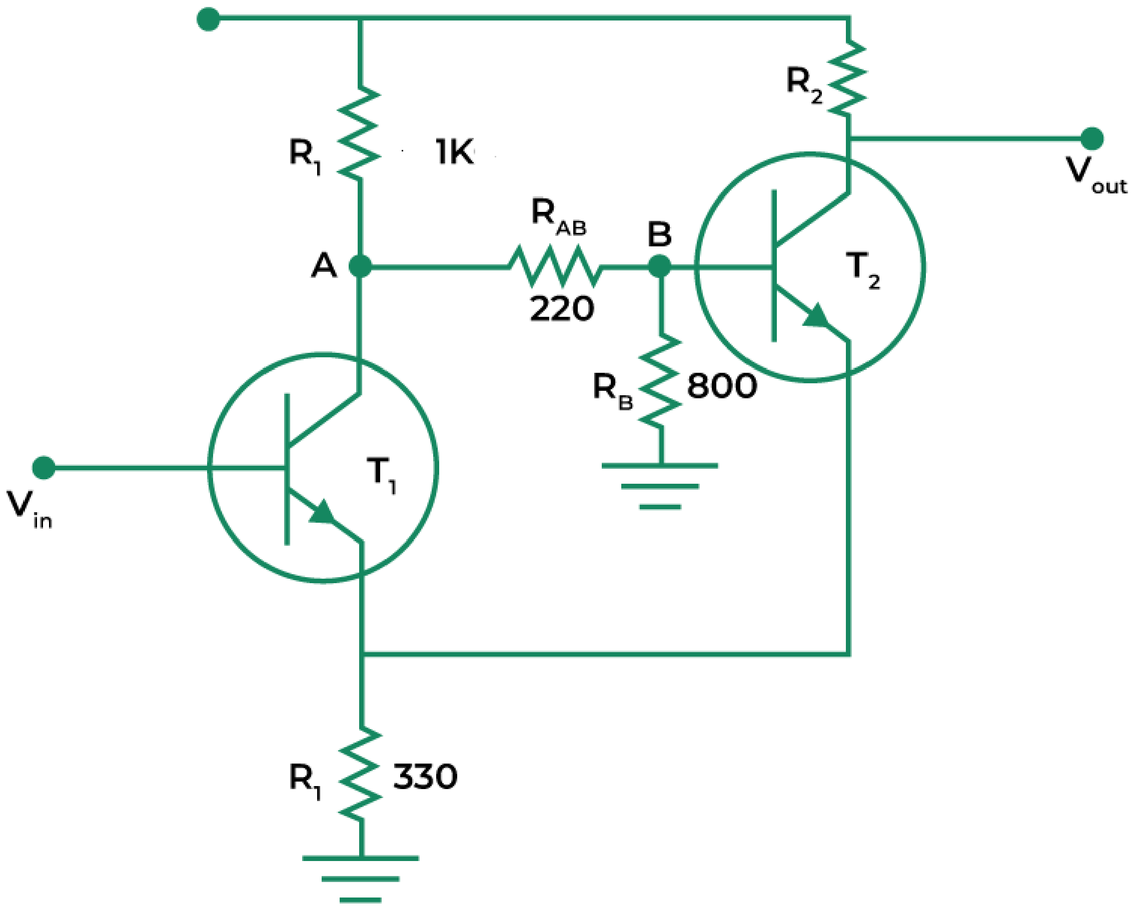
Theory: Design and simulate a phase shifter circuit.



MATLAB Code:

```
% Phase Shifter Circuit
f = 1000; % Frequency in Hz
R = 1e3; % Resistance in ohms
C = 1e-6; % Capacitance in farads
omega = 2 * pi * f;
phi = atan(1 / (omega * R * C)); % Phase shift in radians
disp(['Phase Shift: ', num2str(phi * (180/pi)), ' degrees']);
```

67. Schmitt Trigger Circuit
Theory: Design and simulate a Schmitt trigger circuit using an operational amplifier.



MATLAB Code:

```
% Schmitt Trigger Circuit
Vcc = 5; % Supply voltage in volts
R1 = 1e3; % Resistance R1 in ohms
R2 = 10e3; % Resistance R2 in ohms
Vin = linspace(-Vcc, Vcc, 1000); % Input voltage sweep

Vout = zeros(size(Vin));
Vh = Vcc * (R2 / (R1 + R2)); % High threshold voltage
Vl = -Vcc * (R2 / (R1 + R2)); % Low threshold voltage

for i = 2:length(Vin)
   if Vin(i) > Vh
      Vout(i) = Vcc;
   elseif Vin(i) < Vl
      Vout(i) = -Vcc;
   else
```

```
        Vout(i) = Vout(i-1);
    end
end

plot(Vin, Vout);
title('Schmitt Trigger Circuit');
xlabel('Input Voltage (V)');
ylabel('Output Voltage (V)');
```

68. Band Pass Filter Design

Theory: Design and simulate a band pass filter using RLC components.

Passive Band Pass Filter

Passive Band Pass Filters can be made by connecting together a low pass filter with a high pass filter

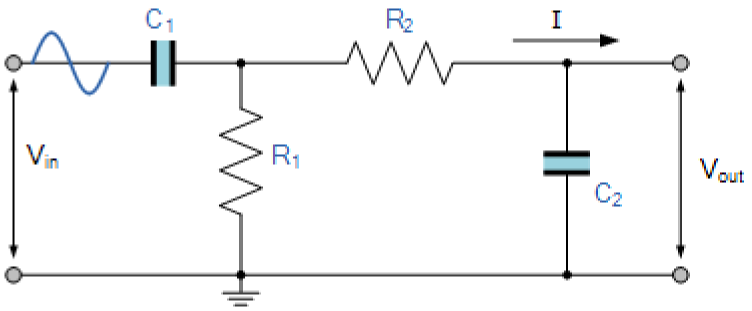
The **Passive Band Pass Filter** can be used to isolate or filter out certain frequencies that lie within a particular band or range of frequencies. The cut-off frequency or *ƒc* point in a simple RC passive filter can be accurately controlled using just a single resistor in series with a non-polarized capacitor, and depending upon which way around they are connected, we have seen that either a Low Pass or a High Pass filter is obtained.

One simple use for these types of passive filters is in audio amplifier applications or circuits such as in loudspeaker crossover filters or pre-amplifier tone controls. Sometimes it is necessary to only pass a certain range of frequencies that do not begin at 0Hz, (DC) or end at some upper high frequency point but are within a certain range or band of frequencies, either narrow or wide.

By connecting or "cascading" together a single *Low Pass Filter* circuit with a *High Pass Filter* circuit, we can produce another type of passive RC filter that passes a selected range or "band" of frequencies that can be either narrow or wide while attenuating all those outside of this range.

This new type of passive filter arrangement produces a frequency selective filter known commonly as a **Band Pass Filter** or **BPF** for short.

Typical Band Pass Filter Circuit



Unlike the low pass filter which only pass signals of a low frequency range or the high pass filter which pass signals of a higher frequency range, a **Band Pass Filters** passes signals within a certain "band" or "spread" of frequencies without distorting the input signal or introducing extra noise. This band of frequencies can be any width and is commonly known as the filters **Bandwidth**.

MATLAB Code:

```
% Band Pass Filter Design
R = 1e3; % Resistance in ohms
L = 1e-3; % Inductance in henrys
C = 1e-6; % Capacitance in farads
sys = tf([R*C 0], [L*C R*C 1]);
bode(sys);
title('Band Pass Filter Design');
```

69. Transient Response Analysis

Theory: Analyze the transient response of a second-order system.
Transient Response in Control System
Transient Response is an important concept in the analysis of dynamic systems. In engineering and physics, dynamic systems are systems that change over time, often in response to an external stimulus or disturbance. Understanding how a system responds to such stimuli is critical in many fields, including control systems, signal processing, and electronics.
Transient Response refers to the behavior of a system during the time it takes to reach a stable state after a disturbance. In other words, it's the system's initial response to a sudden change in its inputs. On the other hand, steady-state response refers to the behavior of a system when it has reached a stable state and is no longer changing in response to the original disturbance. In this article, we will explore these concepts in more detail, including the factors that affect the transient and steady-state response, and how they are used in practice.

MATLAB Code:

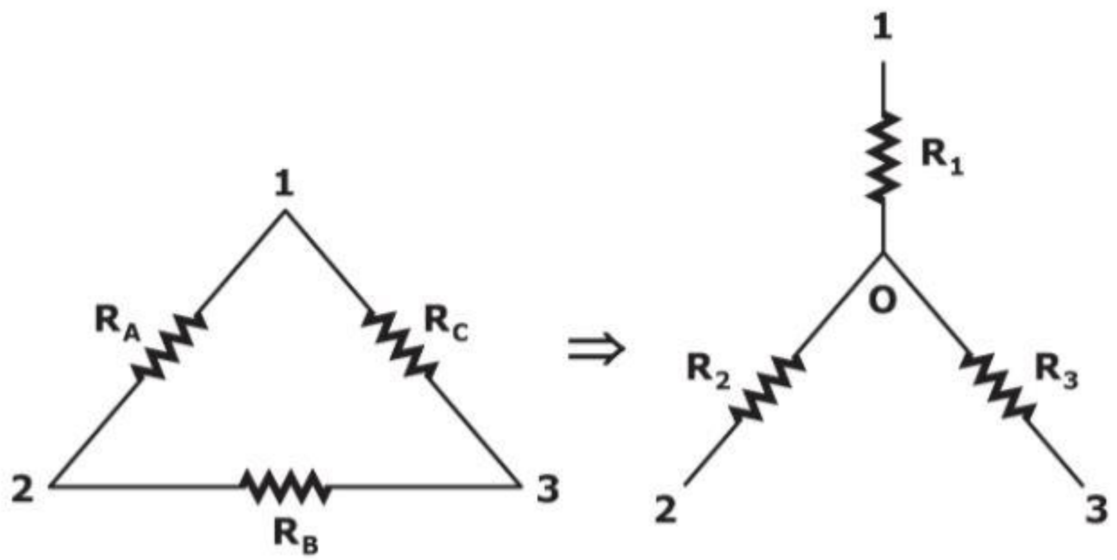
```
% Transient Response Analysis
wn = 5; % Natural frequency
zeta = 0.5; % Damping ratio
sys = tf([wn^2], [1 2*zeta*wn wn^2]);
step(sys);
title('Transient Response Analysis');
```

70. Delta-Wye Transformation

Theory: Perform Delta-Wye transformation for a three-phase network.
Delta–Wye Transformation Formula
Depending upon the conversion of the network different delta-wye transformation formula is produced. Let us see the formula for finding the resistance for various cases of conversion.

Conversion of Delta–Wye or Conversion of Delta – Star

If the resistors are connected in the form of a Delta or triangle (Δ), then it is known as the Delta network. If we invert this delta symbol, then we will get the symbol of the del operator (). This del form, we can modify and represent in the form of the pi symbol (Π). Then it is known as pi-network or Π-network. Hence, these three networks (Δ, ∇, Π) are the same.
If the resistors are connected in the form of the letter Y, then it is known as the Wye network. If we invert the letter Y, then it is known as an inverted Y or star (⅄) network. The letter Y, can be modified into the letter T, then it is known as T-network. Hence, these three networks (Y, ⅄, T) are the same.

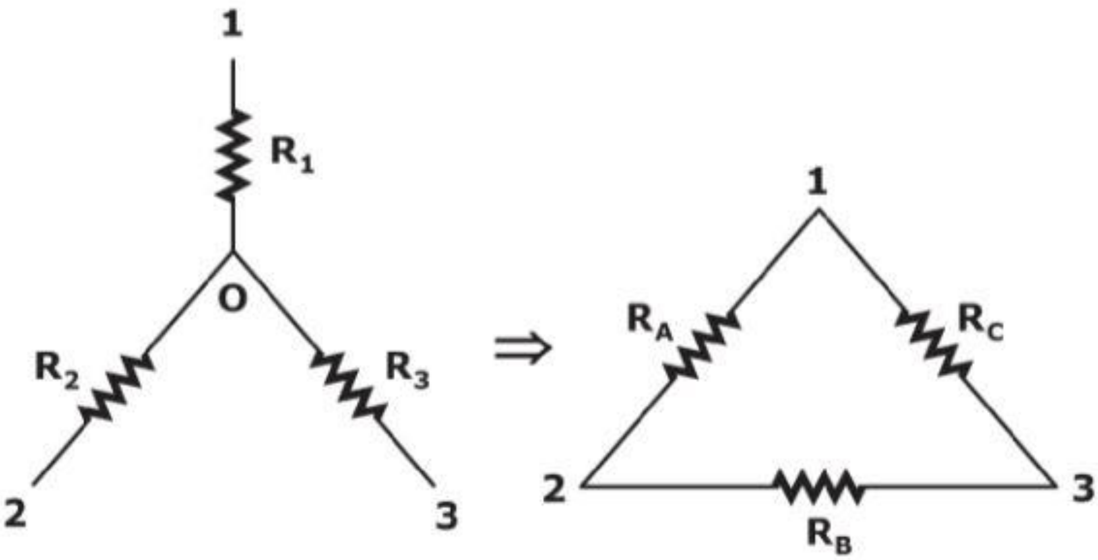


The above figure represents the conversion of the Delta network into a star network. Here, Delta networks consist of three resistors RA, RB, and RC. The resistors in the corresponding Wye network are R1, R_2,and R3. We can calculate the values of R_1, R_2, and R_3 by using the following formulae.

- $R_1=(R_AR_C)/(R_A+R_B+R_C)$.
- $R_2=(R_AR_B)/(R_A+R_B+R_C)$
- $R_3=(R_BR_C)//(R_A+R_B+R_C)$

Conversion of Wye-Delta or Conversion of Star-Delta

Previously, we converted the Delta network into the Star network. Now, let’s see the conversion of Wye-Delta or Star–Delta. The following figure represents the conversion of the Star network into the Delta network.



Here, Star networks consist of three resistors R_1, R_2, and R_3. The resistors in the corresponding Delta network are R_A, R_B, and R_C. We can calculate the values of R_A, R_B, and RC by using the following formulae.

- $R_A=R_1+R_2+(R_1R_2)/R_3$.
- $R_B=R_2+R_3+(R_2R_3)/R_1$.
- $R_C=R_1+R_3+(R_1R_3)R_2$.

MATLAB Code:

```
% Delta-Wye Transformation
Zab = 50; % Impedance between points A and B
Zbc = 50; % Impedance between points B and C
Zca = 50; % Impedance between points C and A

Z1 = Zab * Zbc / (Zab + Zbc + Zca);
Z2 = Zbc * Zca / (Zab + Zbc + Zca);
Z3 = Zca * Zab / (Zab + Zbc + Zca);

disp(['Z1: ', num2str(Z1), ' Ohms']);
disp(['Z2: ', num2str(Z2), ' Ohms']);
disp(['Z3: ', num2str(Z3), ' Ohms']);
```

71. RC Integrator Circuit

Theory: Design and simulate an RC integrator circuit.

For a passive RC integrator circuit, the input is connected to a resistance while the output voltage is taken from across a capacitor being the exact opposite to the *RC Differentiator*

Circuit. The capacitor charges up when the input is high and discharges when the input is low.

In Electronics, the basic series connected resistor-capacitor (RC) circuit has many uses and applications from basic charging/discharging circuits to high-order filter circuits. This two component passive RC circuit may look simple enough, but depending on the type and frequency of the applied input signal, the behaviour and response of this basic RC circuit can be very different.

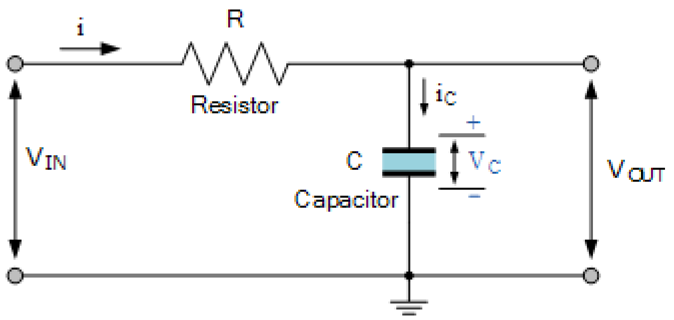
A passive RC network is nothing more than a resistor in series with a capacitor, that is a fixed resistance in series with a capacitor that has a frequency dependant reactance which decreases as the frequency across its plates increases. Thus at low frequencies the reactance, Xc of the capacitor is high while at high frequencies its reactance is low due to the standard capacitive reactance formula of $Xc = 1/(2\pi f C)$, and we saw this effect in our tutorial about *Passive Low Pass Filters*.

If the input signal is a sine wave, an **rc integrator** will simply act as a simple low pass filter (LPF) above its cut-off point with the cut-off or corner frequency corresponding to the RC time constant (tau, τ) of the series network. Thus when fed with a pure sine wave, an RC integrator acts as a passive low pass filter reducing its output above the cut-off frequency point.

As we have seen previously, the RC time constant reflects the relationship between the resistance and the capacitance with respect to time with the amount of time, given in seconds, being directly proportional to resistance, **R** and capacitance, **C**.

Thus the rate of charging or discharging depends on the RC time constant, $\tau = RC$. Consider the circuit below.

RC Integrator



For an RC integrator circuit, the input signal is applied to the resistance with the output taken across the capacitor, then V_{OUT} equals V_C. As the capacitor is a frequency dependant element, the amount of charge that is established across the plates is equal to the time domain integral of the current. That is it takes a certain amount of time for the capacitor to fully charge as the capacitor can not charge instantaneously only charge exponentially.

Therefore the capacitor charging current can be written as:

$$i_{C(t)} = C\frac{dV_{C(t)}}{dt}$$

This basic equation above of $i_C = C(dVc/dt)$ can also be expressed as the instantaneous rate of change of charge, Q with respect to time giving us the following standard equation of:

$i_C = dQ/dt$ where the charge $Q = C \times V_C$, that is capacitance times voltage.

The rate at which the capacitor charges (or discharges) is directly proportional to the amount of the resistance and capacitance giving the time constant of the circuit. Thus the time constant of a RC integrator circuit is the time interval that equals the product of R and C. Since capacitance is equal to Q/Vc where electrical charge, Q is the flow of a current (i) over time (t), that is the product of i x t in coulombs, and from Ohms law we know that voltage (V) is equal to i x R, substituting these into the equation for the RC time constant gives:

RC Time Constant

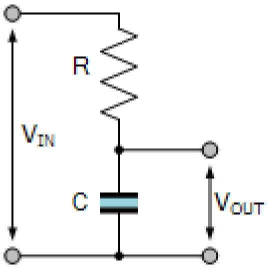
$$RC = R\frac{Q}{V} = R\frac{i\times T}{i\times R} = \cancel{R}\frac{\cancel{i}\times T}{\cancel{i}\times \cancel{R}} = T$$

$$\therefore T = RC$$

Then we can see that as both i and R cancel out, only T remains indicating that the time constant of an RC integrator circuit has the dimension of time in seconds, being given the Greek letter tau, τ.

Note that this time constant reflects the time (in seconds) required for the capacitor to charge up to 63.2% of the maximum voltage or discharge down to 36.8% of maximum voltage.

Capacitor Voltage



We said previously that for the RC integrator, the output is equal to the voltage across the capacitor, that is: V_{OUT} equals V_C. This voltage is proportional to the charge, Q being stored on the capacitor given by: Q = V x C.

The result is that the output voltage is the integral of the input voltage with the amount of integration dependent upon the values of R and C and therefore the time constant of the network.

We saw above that the capacitors current can be expressed as the rate of change of charge, Q with respect to time. Therefore, from a basic rule of differential calculus, the derivative of Q with respect to time is dQ/dt and as i = dQ/dt we get the following relationship of:

$Q = \int i\,dt$ (the charge Q on the capacitor at any instant in time)

Since the input is connected to the resistor, the same current, i must pass through both the resistor and the capacitor ($i_R = i_C$) producing a V_R voltage drop across the resistor so the current, (i) flowing through this series RC network is given as:

$$i(t) = \frac{V_{IN}}{R} = \frac{V_R}{R} = C\frac{dV}{dt}$$

therefore:

$$V_{OUT} = V_C = \frac{Q}{C} = \frac{\int i dt}{C} = \frac{1}{C}\int i(t) dt$$

As i = V_{IN}/R, substituting and rearranging to solve for V_{OUT} as a function of time gives:

$$V_{OUT} = \frac{1}{C}\int\left(\frac{V_{IN}}{R}\right)dt = \frac{1}{RC}\int V_{IN} dt$$

So in other words, the output from an RC integrator circuit, which is the voltage across the capacitor is equal to the time Integral of the input voltage, V_{IN} weighted by a constant of 1/RC. Where RC represents the time constant, τ.
Then assuming the initial charge on the capacitor is zero, that is V_{OUT} = 0, and the input voltage V_{IN} is constant, the output voltage, V_{OUT} is expressed in the time domain as:

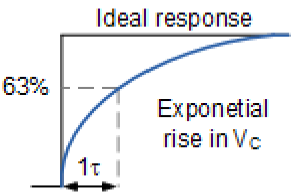
RC Integrator Formula

$$V_{OUT} = \frac{1}{RC}\int_0^t V_{IN(t)} dt$$

Thus an RC integrator circuit is one in which the output voltage, V_{OUT} is proportional to the integral of the input voltage, and with this in mind, lets see what happens when we apply a single positive pulse in the form of a step voltage to the RC integrator circuit.

Single Pulse RC Integrator

When a single step voltage pulse is applied to the input of an RC integrator, the capacitor charges up via the resistor in response to the pulse. However, the output is not instant as the voltage across the capacitor cannot change instantaneously but increases exponentially as the capacitor charges at a rate determined by the RC time constant, τ = RC.



We now know that the rate at which the capacitor either charges or discharges is determined by the RC time constant of the circuit. If an ideal step voltage pulse is applied, that is with the leading edge and trailing edge considered as being instantaneous, the voltage across the capacitor will increase for charging and decrease for discharging, exponentially over time at a rate determined by:

Capacitor Charging:

$$V_{C(t)} = V\left(1 - e^{-\left(\frac{t}{RC}\right)}\right)$$

Capacitor Discharging:

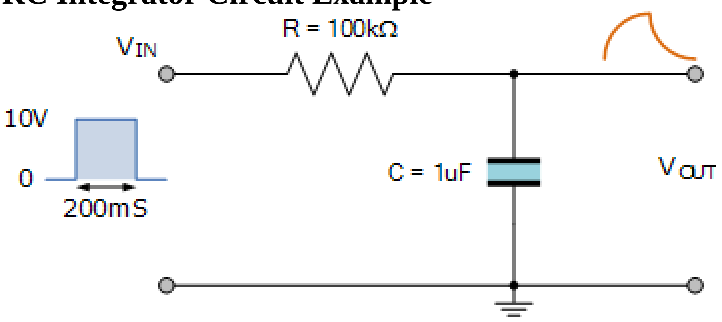
$$V_{C(t)} = V\left(e^{-\left(\frac{t}{RC}\right)}\right)$$

So if we assume a capacitor voltage of one volt (1V), we can plot the percentage of charge or discharge of the capacitor for each individual R time constant as shown in the following table.

Time Constant	Capacitor Charging	Capacitor Discharging
τ	% Charged	% Discharged
0.5	39.4%	60.6%
0.7	50%	50%
1	63.2%	36.7%
2	86.4%	13.5%
3	95.0%	4.9%
4	98.1%	1.8%
5	99.3%	0.67%

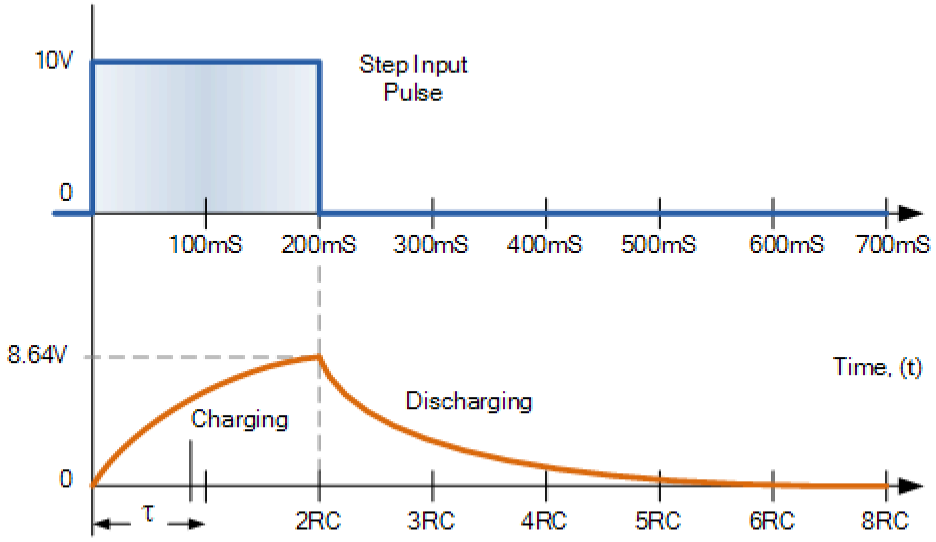
Note that at 5 time constants or above, the capacitor is considered to be 100 percent fully charged or fully discharged.
Now lets assume we have an RC integrator circuit consisting of a 100kΩ resistor and a 1uF capacitor as shown.

RC Integrator Circuit Example



The time constant, τ of the RC integrator circuit is therefore given as:
RC = 100kΩ x 1uF = 100ms.
If we apply a step voltage pulse to the input with a duration of say, two time constants (200mS), then from the table above we can see that the capacitor will charge to 86.4% of its fully charged value.
If this pulse has an amplitude of 10 volts, then this equates to 8.64 volts before the capacitor discharges again back through the resistor to the source as the input pulse returns to zero.
Assuming that the capacitor is allowed to fully discharge in a time of 5 time constants, or 500mS before the arrival of the next input pulse, then the graph of the charging and discharging curves would look something like this:

RC Integrator Charging/Discharging Curves



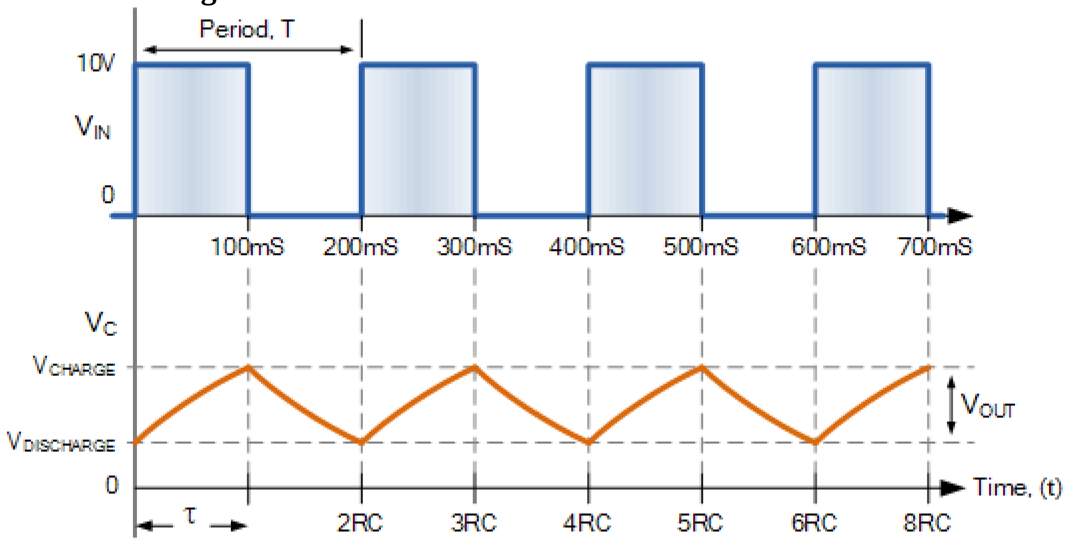
Note that the capacitor is discharging from an initial value of 8.64 volts (2 time constants) and not from the 10 volts input.

Then we can see that as the RC time constant is fixed, any variation to the input pulse width will affect the output of the RC integrator circuit. If the pulse width is increased and is equal too or greater than 5RC, then the shape of the output pulse will be similar to that of the input as the output voltage reaches the same value as the input.

If however the pulse width is decreased below 5RC, the capacitor will only partially charge and not reach the maximum input voltage resulting in a smaller output voltage because the capacitor cannot charge as much resulting in an output voltage that is proportional to the integral of the input voltage.

So if we assume an input pulse equal to one time constant, that is 1RC, the capacitor will charge and discharge not between 0 volts and 10 volts but between 63.2% and 38.7% of the voltage across the capacitor at the time of change. Note that these values are determined by the RC time constant.

Fixed RC Integrator Time Constant

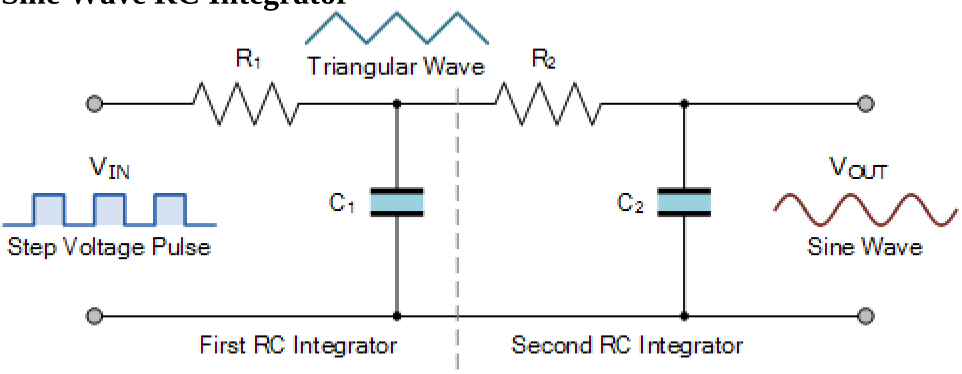


So for a continuous pulse input, the correct relationship between the periodic time of the input and the RC time constant of the circuit, integration of the input will take place producing a sort of ramp up, and then a ramp down output.
But for the circuit to function correctly as an integrator, the value of the RC time constant has to be large compared to the inputs periodic time. That is RC ≫ T, usually 10 times greater. This means that the magnitude of the output voltage (which was proportional to 1/RC) will be very small between its high and low voltages severely attenuating the output voltage. This is because the capacitor has much less time to charge and discharge between pulses but the average output DC voltage will increase towards one half magnitude of the input and in our pulse example above, this will be 5 volts (10/2).

RC Integrator as a Sine Wave Generator

We have seen above that an *RC integrator* circuit can perform the operation of integration by applying a pulse input resulting in a ramp-up and ramp-down triangular wave output due to the charging and discharging characteristics of the capacitor. But what would happen if we reversed the process and applied a triangular waveform to the input, would we get a pulse or square wave output?
When the input signal to an RC integrator circuit is a pulse shaped input, the output is a triangular wave. But when we apply a triangular wave, the output becomes a sine wave due to the integration over time of the ramp signal.
There are many ways to produce a sinusoidal waveform, but one simple and cheap way to electronically produce a sine waves type waveform is to use a pair of passive RC integrator circuits connected together in series as shown.

Sine Wave RC Integrator



Here the first RC integrator converts the original pulse shaped input into a ramp-up and ramp-down triangular waveform which becomes the input of the second RC integrator. This second RC integrator circuit rounds off the points of the triangular waveform converting it into a sine wave as it is effectively performing a double integration on the original input signal with the RC time constant affecting the degree of integration.
As the integration of a ramp produces a sine function, (basically a round-off triangular waveform) its periodic frequency in Hertz will be equal to the period T of the original pulse. Note also that if we reverse this signal and the input signal is a sine wave, the circuit does not act as an integrator, but as a simple low pass filter (LPF) with the sine wave, being a pure waveform does not change shape, only its amplitude is affected.

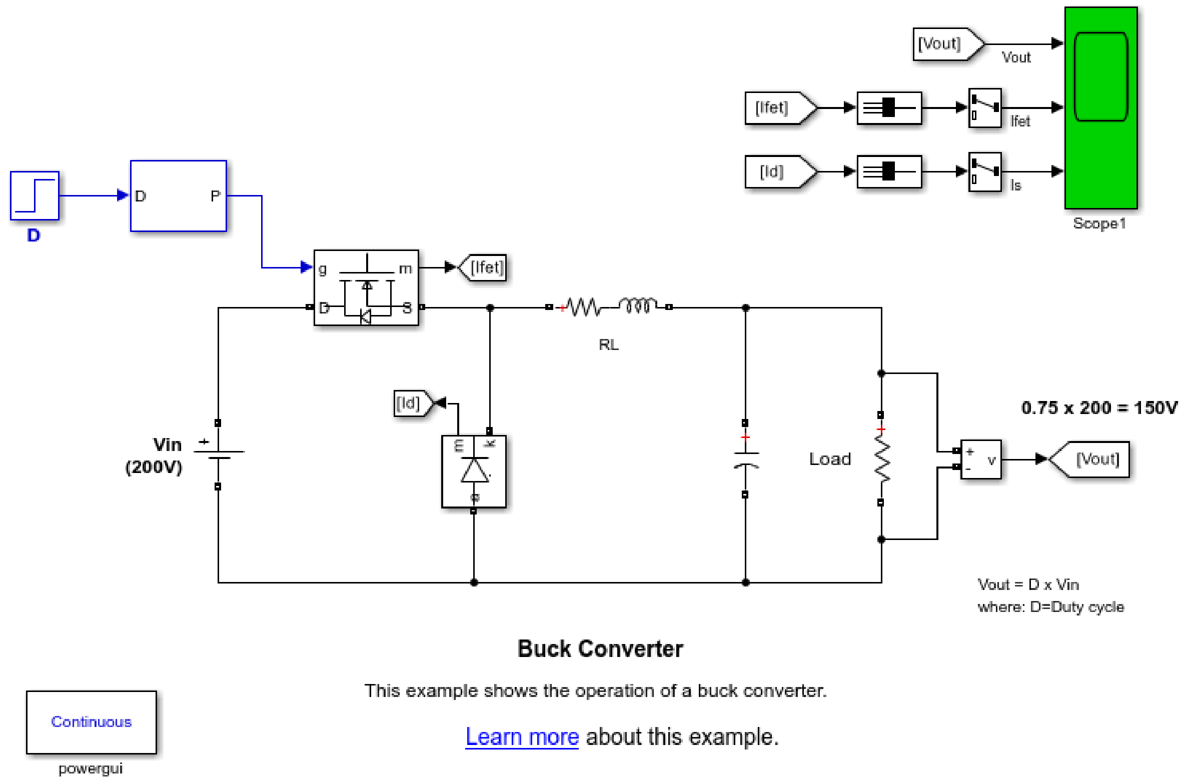
MATLAB Code:

```
% RC Integrator Circuit
R = 1e3; % Resistance in ohms
C = 1e-6; % Capacitance in farads
sys = tf([1], [R*C 1]);
bode(sys);
title('RC Integrator Circuit');
```

72. Buck Converter Simulation

Theory: Simulate a buck converter circuit.

Buck Converter
This example shows the operation of a buck converter.



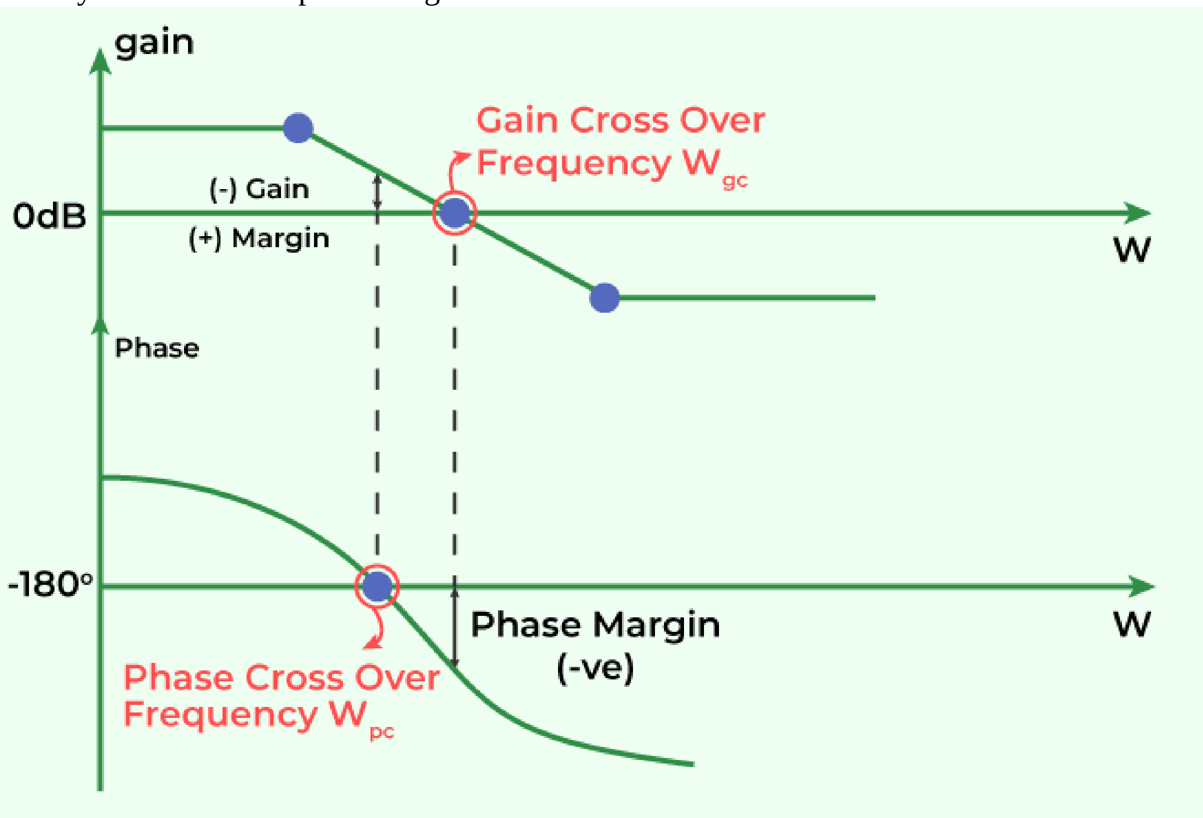
Description
A buck converter is a DC/DC power converter which steps down voltage from its input (source) to its output (load). In continuous conduction mode (current through the inductor never falls to zero), the theoretical transfer function of the buck converter is:
$Vout/Vin = D$
where D is the duty cycle.
In this example, the converter is feeding an RC load from a 200 V source and the PWM frequency is set to 10 kHz.

MATLAB Code:

```
% Buck Converter Simulation
Vin = 12; % Input voltage in volts
Vout = 5; % Desired output voltage in volts
D = Vout / Vin; % Duty cycle

t = 0:0.0001:0.01; % Time vector
V = Vin * D * square(2*pi*1e3*t); % Output voltage waveform

plot(t, V);
title('Buck Converter Simulation');
xlabel('Time (s)');
ylabel('Voltage (V)');
```

73. Bode Plot Analysis

Theory: Create a Bode plot for a given transfer function.



MATLAB Code:

```
% Bode Plot Analysis
sys = tf([1], [1 3 3 1]);
bode(sys);
title('Bode Plot Analysis');
```

74. Root Mean Square (RMS) Calculation

Theory: Calculate the RMS value of a periodic signal.
What is Root Mean Square (RMS)?
Statistically, the **root mean square (RMS)** is the square root of the mean square, which is the arithmetic mean of the squares of a group of values. RMS is also called a quadratic mean and is a special case of the generalized mean whose exponent is 2. Root mean square is also defined as a varying function based on an integral of the squares of the values which are instantaneous in a cycle.
In other words, the RMS of a group of numbers is the square of the arithmetic mean or the function's square which defines the continuous waveform.
Root Mean Square Formula
The **formula for Root Mean Square** is given below to get the RMS value of a set of data values.
For a group of n values involving {x_1, x_2, x_3,…. X_n}, the RMS is given by:

-

The formula for a continuous function f(t), defined for the interval $\mathbf{T_1 \leq t \leq T_2}$ is given by:

-

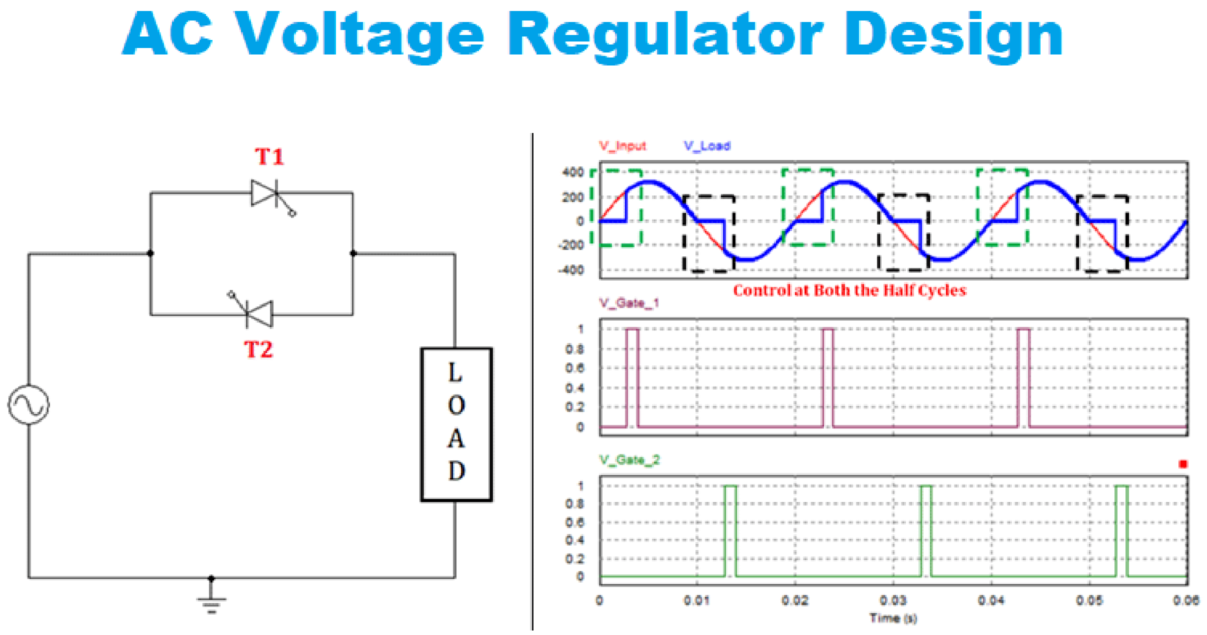
The RMS of a periodic function is always equivalent to the RMS of a function's single period. The continuous function's RMS value can be considered approximately by taking the RMS of a sequence of evenly spaced entities. Also, the RMS value of different waveforms can also be calculated without calculus.
How to Calculate the Root Mean Square
Steps to Find the Root mean square for a given set of values are given below:
Step 1: Get the squares of all the values
Step 2: Calculate the average of the obtained squares
Step 3: Finally, take the square root of the average
Solved Example
Question:
Calculate the root mean square (RMS) of the data set: 1, 3, 5, 7, 9
Solution:
Given set of data values:
1, 3, 5, 7, 9
Step 1: Squares of these values
$1^2, 3^2, 5^2, 7^2, 9^2$
Or
1, 9, 25, 49, 81
Step 2: Average of the squares
(1 + 9 + 25 + 49 + 81)/5
= 165/5
= 33
Step 3: Take the square root of the average.
RMS = $\sqrt{33}$ = 5.745 (approx)

MATLAB Code:

```
% Root Mean Square (RMS) Calculation
t = linspace(0, 2*pi, 1000);
x = sin(t);
RMS = sqrt(mean(x.^2));
disp(['RMS Value: ', num2str(RMS)]);
```

75. AC Voltage Regulation

Theory: Calculate the voltage regulation of a transformer.

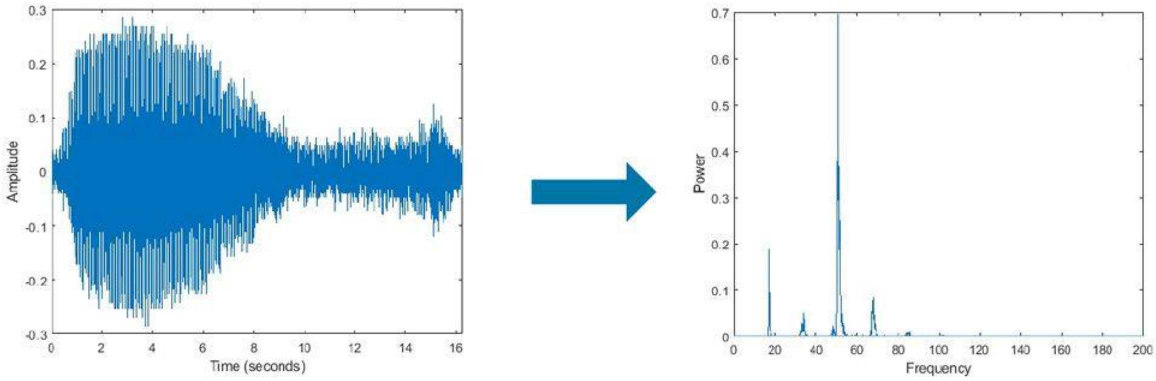


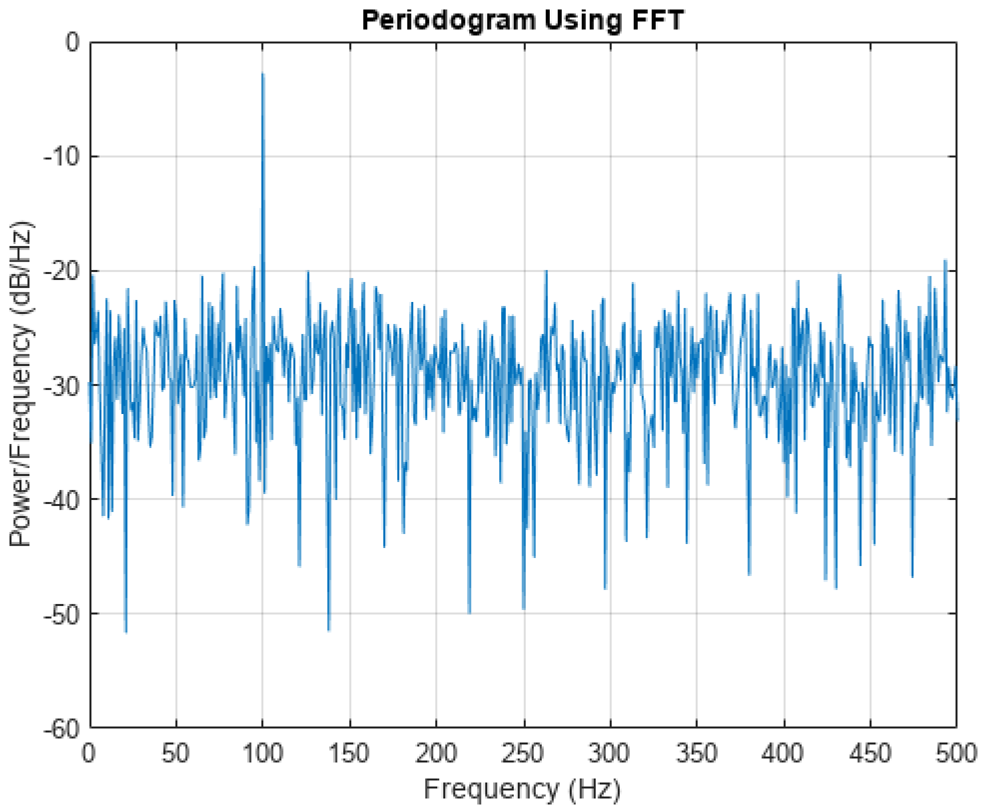
MATLAB Code:

```
% AC Voltage Regulation
V_no_load = 230; % No-load voltage in volts
V_full_load = 220; % Full-load voltage in volts
regulation = ((V_no_load - V_full_load) / V_full_load) * 100;
disp(['Voltage Regulation: ', num2str(regulation), '%']);
```

76. Spectral Analysis Using FFT

Theory: Perform spectral analysis of a signal using the Fast Fourier Transform (FFT).





MATLAB Code:

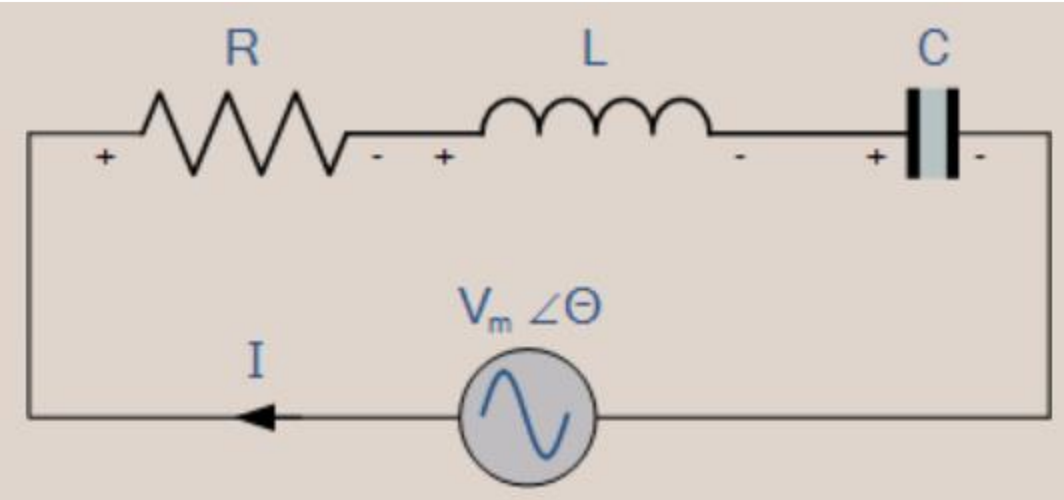
```
% Spectral Analysis Using FFT
Fs = 1000; % Sampling frequency
t = 0:1/Fs:1-1/Fs;
x = cos(2*pi*100*t) + 0.5*cos(2*pi*200*t); % Signal with two frequencies

X = fft(x);
f = (0:length(X)-1)*Fs/length(X); % Frequency vector

plot(f, abs(X));
title('Spectral Analysis Using FFT');
xlabel('Frequency (Hz)');
```

ylabel('Magnitude');

77. Resonance in RLC Circuit

Theory: Analyze the resonance behavior in an RLC circuit.

In the series RLC circuit at resonance, we will connect the AC voltage source, Resistor (R), Inductor (L), and Capacitor (C), all in series. This circuit diagram is shown in the figure below.



The current is the same in series, but the supply voltage (AC) gets divided among the passive elements.

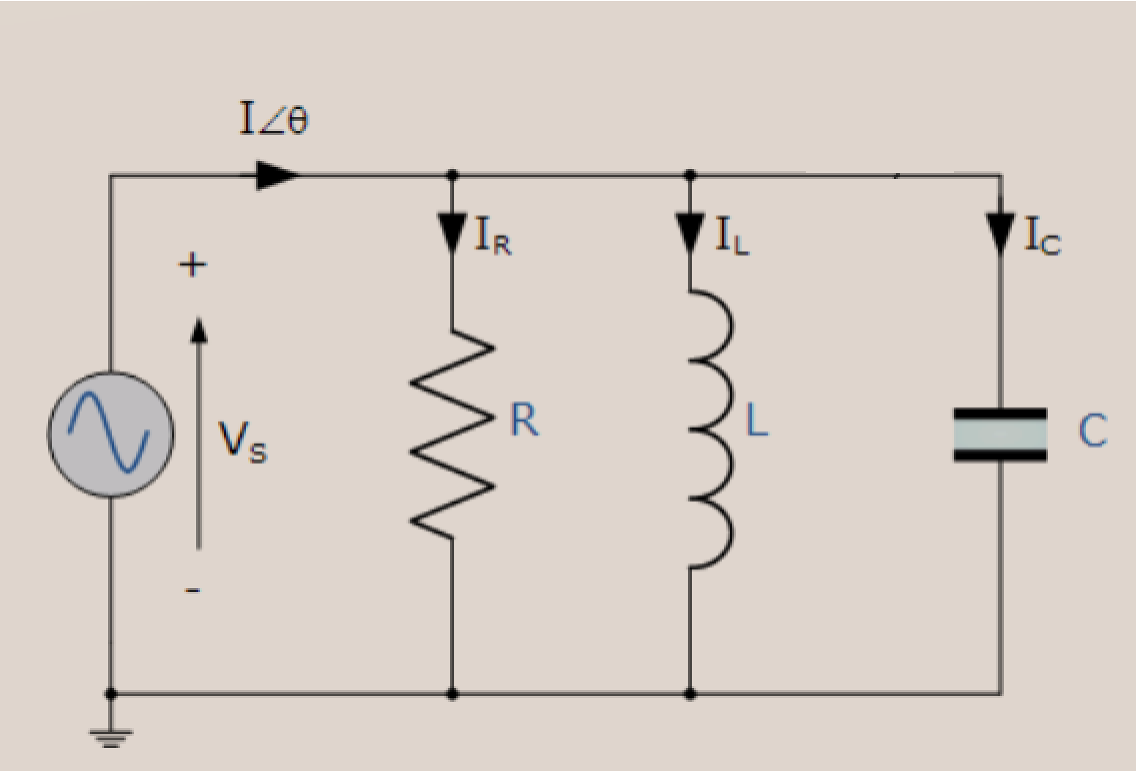
- Since R, L, and C are connected in series, the equivalent impedance will be $Z=R+j(\omega L-1/\omega C)$.
- The impedance, Z, will be real, equal to R when the imaginary part of impedance becomes zero at $\omega=0$.
- At $\omega=\omega_0$, the inductor and capacitor reactance is the same.

$\omega_0 L=1/\omega_0 C$
$=>\omega_0 2=1/LC$
$=>\omega_0=1/\sqrt{}LC$
$=>f_0=1/2\pi\sqrt{}LC$

- f0 is the series resonant frequency, equal to $1/2\pi\sqrt{}LC$.
- At $\omega=\omega_0$, V=IZ=IR. That is, voltage and current are in phase at a resonant frequency. i.e., $\emptyset=0$.
- At the series resonant frequency, the power factor cos $\emptyset$ will equal 1. Hence, it is called a unity power factor.

Resonance in Parallel RLC Circuit

In a parallel RLC circuit, we will connect the AC source, Resistor (R), Inductor (L), and Capacitor (C) all in parallel. This circuit diagram is shown in the figure below.



Voltage is the same, but the supply current (AC) gets divided among the passive elements.

- Since R, L and C are connected in parallel, and the equivalent admittance will be $Y=1/R+j(\omega C-1/\omega L)$.
- The admittance, Y will be real, and it is equal to 1R when the imaginary part of admittance becomes zero at $\omega=\omega_0$.
- At $\omega=\omega_0$, the inductor and capacitor susceptance are the same.

$$\omega_0 C=1/\omega_0 L$$
$$\Rightarrow \omega_0 2=1/LC$$
$$\Rightarrow \omega_0=1/\sqrt{LC}$$
$$\Rightarrow f_0=1/2\pi\sqrt{LC}$$

- f_0 is the parallel resonant frequency, equal to $1/2\pi\sqrt{LC}$.
- At $\omega=\omega_0$, $I=VY=V/R$. That is, current and voltage are in phase at a resonant frequency. i.e., $\emptyset=0$.
- The power factor at resonance in RLC parallel circuit is cos $\emptyset$, which will equal 1. Hence, it is called a unity power factor.

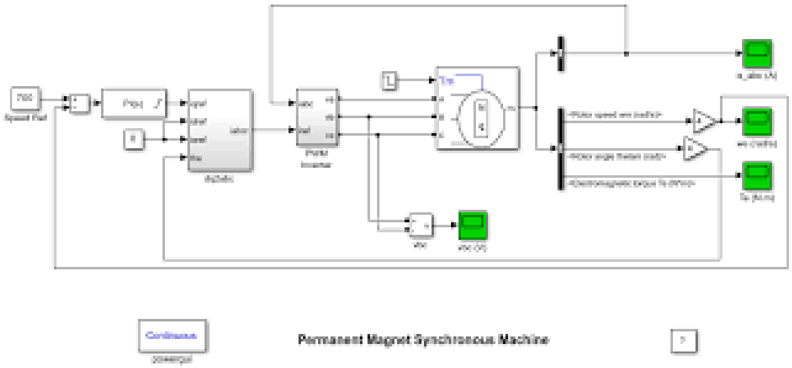
MATLAB Code:

```
% Resonance in RLC Circuit
R = 10; % Resistance in ohms
L = 1e-3; % Inductance in henrys
C = 1e-6; % Capacitance in farads

f_resonance = 1 / (2*pi*sqrt(L*C)); % Resonant frequency
```

```
disp(['Resonant Frequency: ', num2str(f_resonance), ' Hz']);

f = linspace(0, 2*f_resonance, 1000);
Z = sqrt(R^2 + (2*pi*f*L - 1./(2*pi*f*C)).^2); % Impedance

plot(f, Z);
title('Resonance in RLC Circuit');
xlabel('Frequency (Hz)');
ylabel('Impedance (Ohms)');
```

78. Synchronous Motor Simulation

Theory: Simulate the operation of a synchronous motor.



MATLAB Code:

```
% Synchronous Motor Simulation
V = 230; % Supply voltage in volts
f = 50; % Supply frequency in Hz
P = 4; % Number of poles
N_sync = 120*f/P; % Synchronous speed in RPM

t = linspace(0, 1, 1000);
theta = 2*pi*N_sync/60 * t; % Rotor angle in radians

I = V * sin(theta); % Induced current

plot(t, I);
title('Synchronous Motor Simulation');
xlabel('Time (s)');
ylabel('Current (A)');
```

79. Power System Load Flow Analysis

Theory: Perform load flow analysis for a power system using Newton-Raphson method.

Load flow analysis, also known as power flow analysis, is a steady-state analysis technique used to determine the voltage magnitudes, voltage angles, and power flows in an electrical network. Its primary objective is to calculate the steady-state operating conditions of the system, considering the active power (real power), reactive power, and power factor requirements. Load flow analysis helps in determining the loading conditions of transmission lines, transformers, and generators, and ensures that voltage levels are within acceptable limits.

Short circuit analysis, on the other hand, is performed to assess the behavior of a power system under fault conditions. A fault occurs when an abnormal condition, such as a short circuit or a ground fault, leads to an unintended flow of current in the power system. Short circuit analysis aims to calculate the fault levels and the impact of faults on the system's protective devices. This analysis is vital for determining the adequacy of protection schemes and the selection of appropriate equipment.

Both load flow and short circuit analysis are essential components of power system modeling. They complement each other by providing a comprehensive understanding of the power system's operating conditions and its response to faults. This analysis aids in system planning, expansion, and operation, enabling facilities to optimize their resources, enhance system reliability, and ensure the uninterrupted supply of electricity to consumers.

MATLAB Code:

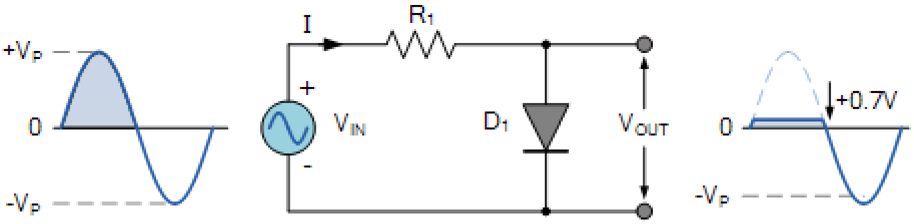
```
% Power System Load Flow Analysis using Newton-Raphson Method
% Define system parameters
Ybus = [
    10-20j -5+10j -5+10j;
    -5+10j  8-16j -3+6j;
    -5+10j -3+6j  8-16j
]; % Y-bus matrix

P = [-1.2; 1.0; 0.5]; % Active power demand (per unit)
Q = [-0.5; 0.3; 0.2]; % Reactive power demand (per unit)

V = [1; 1; 1]; % Initial guess for voltage magnitudes
theta = [0; 0; 0]; % Initial guess for voltage angles

% Newton-Raphson iteration
max_iter = 10;
tol = 1e-6;
for iter = 1:max_iter
    % Calculate power mismatches
    P_calc = real(V .* (Ybus * (V .* exp(1j*theta))));
    Q_calc = imag(V .* (Ybus * (V .* exp(1j*theta))));
    dP = P - P_calc;
    dQ = Q - Q_calc;
    mismatch = [dP; dQ];

    % Check for convergence
```

```
    if norm(mismatch) < tol
        break;
    end

    % Jacobian matrix calculation
    J11 = diag(V) * real(Ybus * diag(V .* exp(1j*theta)));
    J12 = diag(V) * imag(Ybus * diag(V .* exp(1j*theta)));
    J21 = diag(V) * imag(Ybus * diag(V .* exp(1j*theta)));
    J22 = -diag(V) * real(Ybus * diag(V .* exp(1j*theta)));
    J = [J11 J12; J21 J22];

    % Update voltage magnitudes and angles
    correction = inv(J) * mismatch;
    dtheta = correction(1:length(P));
    dV = correction(length(P)+1:end);
    theta = theta + dtheta;
    V = V + dV;
end

disp('Voltage magnitudes (pu):');
disp(V);
disp('Voltage angles (degrees):');
disp(rad2deg(theta));
```

80. Diode Clipper Circuit

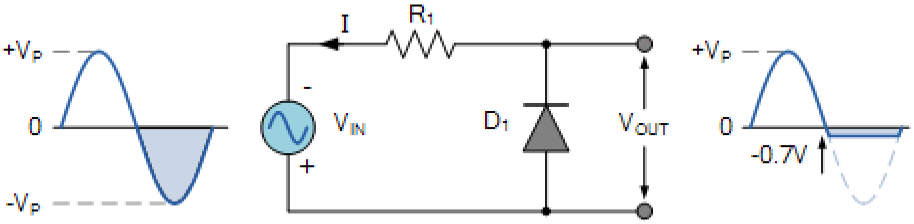
Theory: Design and simulate a diode clipper circuit.

Positive Diode Clipping Circuits



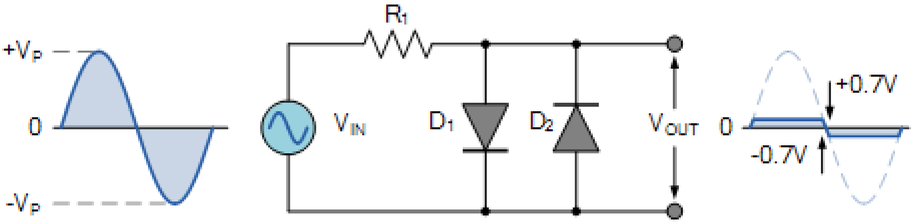
In this diode clipping circuit, the diode is forward biased (anode more positive than cathode) during the positive half cycle of the sinusoidal input waveform. For the diode to become forward biased, it must have the input voltage magnitude greater than +0.7 volts (0.3 volts for a germanium diode).
When this happens the diodes begins to conduct and holds the voltage across itself constant at 0.7V until the sinusoidal waveform falls below this value. Thus the output voltage which is taken across the diode can never exceed 0.7 volts during the positive half cycle.
During the negative half cycle, the diode is reverse biased (cathode more positive than anode) blocking current flow through itself and as a result has no effect on the negative half of the sinusoidal voltage which passes to the load unaltered. Thus the diode limits the positive half of the input waveform and is known as a positive clipper circuit.

Negative Diode Clipping Circuit



Here the reverse is true. The diode is forward biased during the negative half cycle of the sinusoidal waveform and limits or clips it to –0.7 volts while allowing the positive half cycle to pass unaltered when reverse biased. As the diode limits the negative half cycle of the input voltage it is therefore called a negative clipper circuit.

Clipping of Both Half Cycles



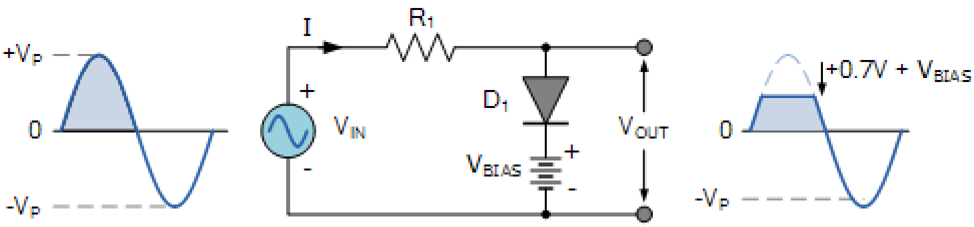
If we connected two diodes in inverse parallel as shown, then both the positive and negative half cycles would be clipped as diode D_1 clips the positive half cycle of the sinusoidal input waveform while diode D_2 clips the negative half cycle. Then diode clipping circuits can be used to clip the positive half cycle, the negative half cycle or both.

For ideal diodes the output waveform above would be zero. However, due to the forward bias voltage drop across the diodes the actual clipping point occurs at +0.7 volts and –0.7 volts respectively. But we can increase this ±0.7V threshold to any value we want up to the maximum value, (V_{PEAK}) of the sinusoidal waveform either by connecting together more diodes in series creating multiples of 0.7 volts, or by adding a voltage bias to the diodes.

Biased Diode Clipping Circuits

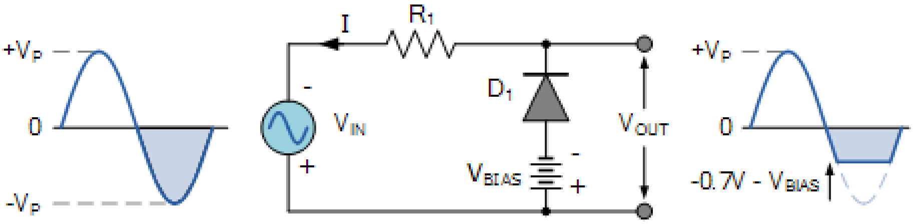
To produce diode clipping circuits for voltage waveforms at different levels, a bias voltage, V_{BIAS} is added in series with the diode to produce a combination clipper as shown. The voltage across the series combination must be greater than V_{BIAS} + 0.7V before the diode becomes sufficiently forward biased to conduct. For example, if the V_{BIAS} level is set at 4.0 volts, then the sinusoidal voltage at the diode's anode terminal must be greater than **4.0 + 0.7 = 4.7 volts** for it to become forward biased. Any anode voltage levels above this bias point are clipped off.

Positive Bias Diode



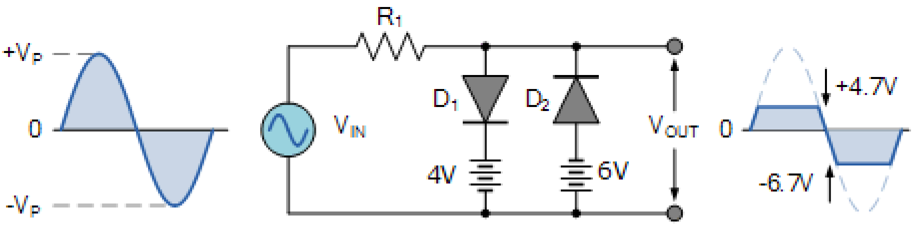
Likewise, by reversing the diode and the battery bias voltage, when a diode conducts the negative half cycle of the output waveform is held to a level $-V_{BIAS}$ – 0.7V as shown.

Negative Bias Diode



A variable diode clipping or diode limiting level can be achieved by varying the bias voltage of the diodes. If both the positive and the negative half cycles are to be clipped, then two biased clipping diodes are used. But for both positive and negative diode clipping, the bias voltage need not be the same. The positive bias voltage could be at one level, for example 4 volts, and the negative bias voltage at another, for example 6 volts as shown.

Diode Clipping of Different Bias levels



When the voltage of the positive half cycle reaches +4.7 V, diode D_1 conducts and limits the waveform at +4.7 V. Diode D_2 does not conduct until the voltage reaches –6.7 V. Therefore, all positive voltages above +4.7 V and negative voltages below –6.7 V are automatically clipped.

The advantage of biased diode clipping circuits is that it prevents the output signal from exceeding preset voltage limits for both half cycles of the input waveform, which could be an input from a noisy sensor or the positive and negative supply rails of a power supply.

If the diode clipping levels are set too low or the input waveform is too great then the elimination of both waveform peaks could end up with a square-wave shaped waveform.

Zener Diode Clipping Circuits

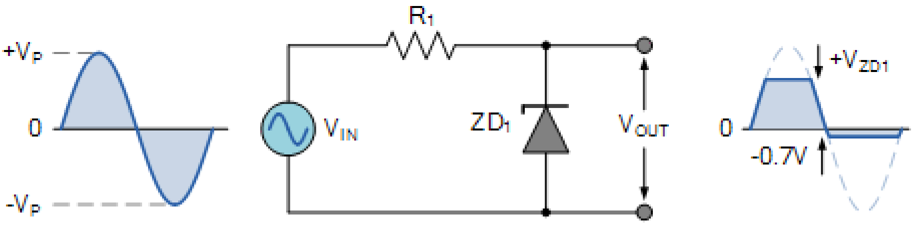
The use of a bias voltage means that the amount of the voltage waveform that is clipped off can be accurately controlled. But one of the main disadvantages of using voltage biased diode clipping circuits, is that they need an additional emf battery source which may or may not be a problem.

One easy way of creating biased diode clipping circuits without the need for an additional emf supply is to use Zener Diodes.

As we know, the zener diode is a another type of diode that has been specially manufactured to operate in its reverse biased breakdown region and as such can be used for voltage regulation or zener diode clipping applications. In the forward region, the zener acts just like an ordinary silicon diode with a forward voltage drop of 0.7V (700mV) when conducting, the same as above.

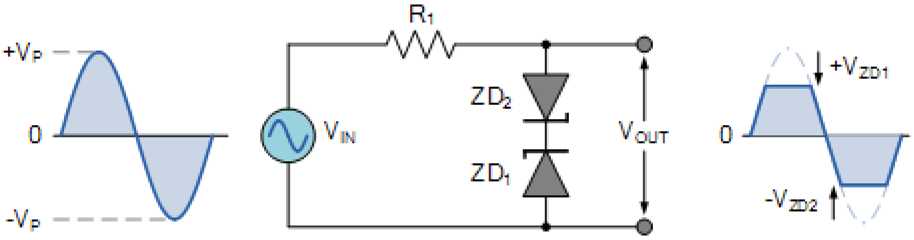
However, in the reverse bias region, the voltage is blocked until the zener diodes breakdown voltage is reached. At this point, the reverse current through the zener increases sharply but the zener voltage, V_Z across the device remains constant even if the zener current, I_Z varies. Then we can put this zener action to good effect by using them for clipping a waveform as shown.

Zener Diode Clipping



The zener diode is acting like a biased diode clipping circuit with the bias voltage being equal to the zener breakdown voltage. In this circuit during the positive half of the waveform the zener diode is reverse biased so the waveform is clipped at the zener voltage, V_{ZD1}. During the negative half cycle the zener acts like a normal diode with its usual 0.7V junction value.
We can develop this idea further by using the zener diodes reverse-voltage characteristics to clip both halves of a waveform using series connected back-to-back zener diodes as shown.

Full-wave Zener Diode Clipping



The output waveform from full wave zener diode clipping circuits resembles that of the previous voltage biased diode clipping circuit. The output waveform will be clipped at the zener voltage plus the 0.7V forward volt drop of the other diode. So for example, the positive half cycle will be clipped at the sum of zener diode, ZD_1 plus 0.7V from ZD_2 and vice versa for the negative half cycle.
Zener diodes are manufactured with a wide range of voltages and can be used to give different voltage references on each half cycle, the same as above. Zener diodes are available with zener breakdown voltages, V_Z ranging from 2.4 to 33 volts, with a typical tolerance of 1 or 5%. Note that once conducting in the reverse breakdown region, full current will flow through the zener diode so a suitable current limiting resistor, R_1 must be chosen.

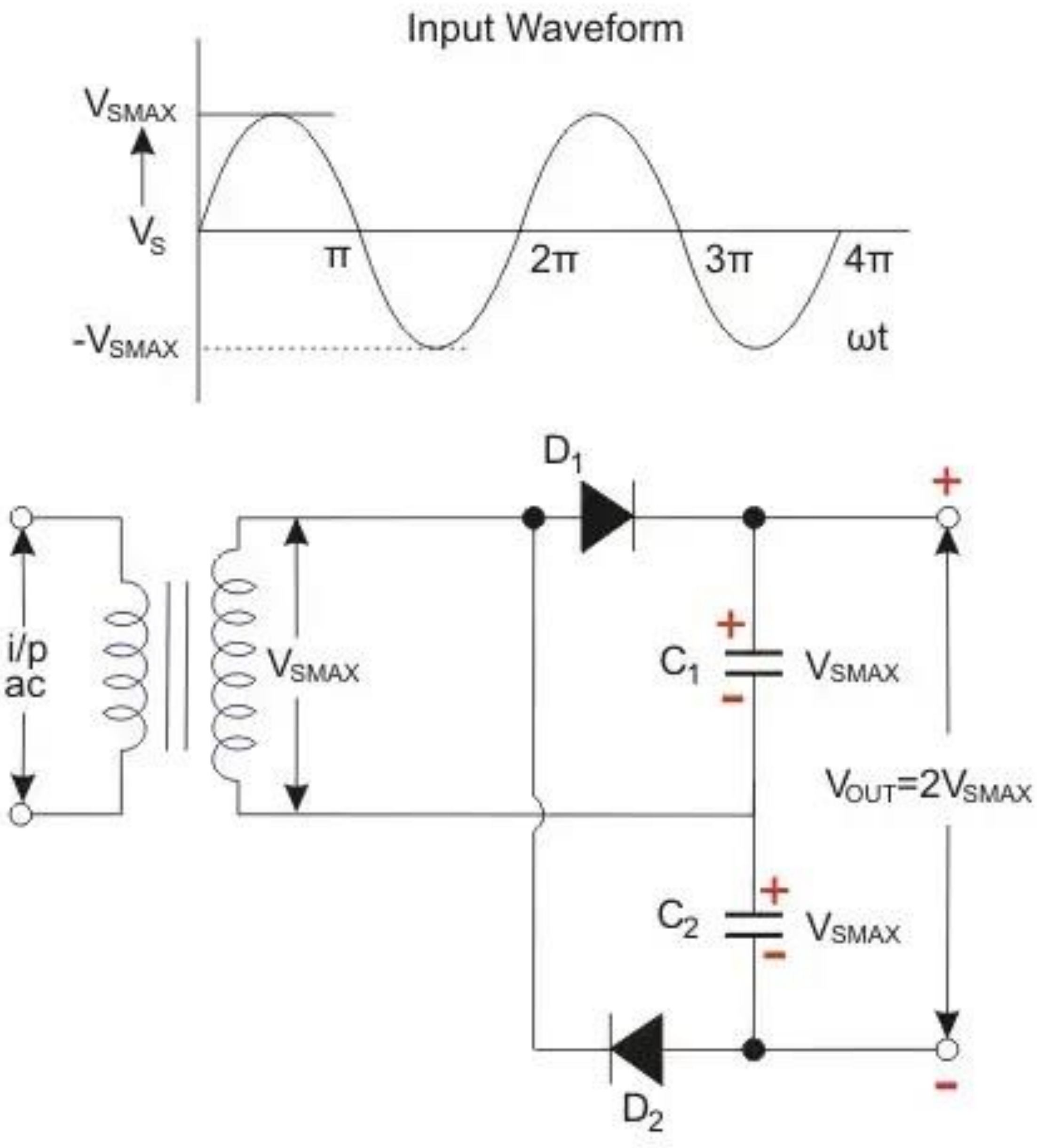
MATLAB Code:

```
% Diode Clipper Circuit
V_in = linspace(-10, 10, 1000); % Input voltage range
V_clip = 3; % Clipping voltage
V_out = zeros(size(V_in));

for i = 1:length(V_in)
   if V_in(i) > V_clip
      V_out(i) = V_clip;
   elseif V_in(i) < -V_clip
      V_out(i) = -V_clip;
   else
      V_out(i) = V_in(i);
   end
end
```

```
plot(V_in, V_out);
title('Diode Clipper Circuit');
xlabel('Input Voltage (V)');
ylabel('Output Voltage (V)');
```

81. Voltage Doubler Circuit

Theory: Design and simulate a voltage doubler circuit.



MATLAB Code:

```
% Voltage Doubler Circuit
V_in = 5; % Input AC voltage peak
```

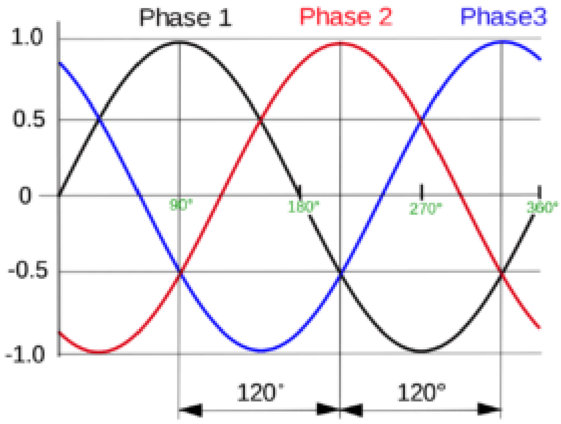
```
f = 60; % Frequency in Hz
t = linspace(0, 1/f, 1000); % Time vector

V_out = 2 * V_in * abs(sin(2*pi*f*t)); % Output voltage waveform

plot(t, V_out);
title('Voltage Doubler Circuit');
xlabel('Time (s)');
ylabel('Voltage (V)');
```

82. Three-Phase Power Calculation

Theory: Calculate the power in a balanced three-phase system.

Three Phase Power and Current

The power was taken by a circuit (single or three-phase) is measured in watts W (or kW). The product of the voltage and current is the apparent power and measured in VA (or kVA). The relationship between kVA and kW is the power factor (Pf):



kW = kVA x pf

which can also be expressed as:

kVA = kW/pf

Single phase system

this is the easiest to deal with. Given the kW and power factor, the kVA can be easily worked out. The current is simply the kVA divided by the voltage. As an example, consider a load consuming 23 kW of power at 230 V and a power factor of 0.86:

kVA = kW / power factor = 23/0.86 = 26.7 kVA (26700VA)

Current = VA / voltage = 26700 / 230 = 116 A

Note: you can do these equations in either VA, V, and A or kVA, kV, and KA depending on the parameters you are dealing with. To convert from VA to kVA just divide by 1000.

Three phase system

The main difference between a three-phase system and a single-phase system is the voltage. In a three-phase system, we have the line to line voltage (V_{LL}) and the phase voltage (V_{LN}), related by:

$V_{LL} = \sqrt{3} \times V_{LN}$

or alternatively as:

$V_{LN} = V_{LL} / \sqrt{3}$

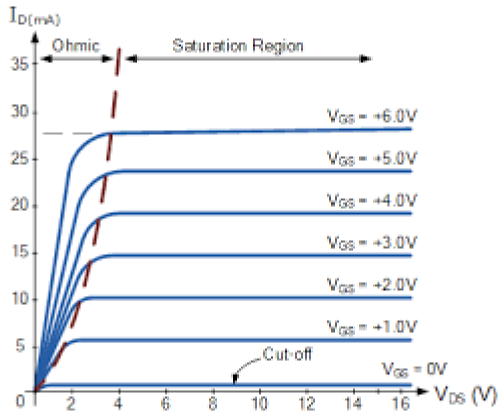
the easiest way to solve three-phase problems is to convert them to a single-phase problem. Take a three-phase motor (with three windings, each identical) consuming a given kW. The kW per winding (single-phase) has to be the total divided by 3. Similarly, a transformer (with three windings, each identical) supplying a given kVA will have each winding supplying a third of the total power. To convert a three-phase problem to a single phase problem take the total kW (or kVA) and divide it by three.
As an example, consider a balanced three phase load consuming 36 kW at a power factor of 0.86 and line to line voltage of 400 V (V_{LL}) :
note: the line to neutral (phase) voltage **V_{LN} = 400/ √3 = 230 V**
three phase power is 36 kW, single phase power = 36/3 = 12 kW
now simply follow the above single phase method
kVA = kW / power factor = 12 / 0.86 = 13.9 kVA (13900 VA)
Current = VA / voltage = 13900 / 230 = 60 A

MATLAB Code:

```
% Three-Phase Power Calculation
V_phase = 230; % Phase voltage in volts
I_phase = 10; % Phase current in amperes
pf = 0.8; % Power factor

P = 3 * V_phase * I_phase * pf; % Total real power
Q = 3 * V_phase * I_phase * sqrt(1 - pf^2); % Total reactive power

disp(['Total Real Power: ', num2str(P), ' W']);
disp(['Total Reactive Power: ', num2str(Q), ' VAR']);
```

83. MOSFET Characteristics Plot

Theory: Plot the I-V characteristics of a MOSFET.



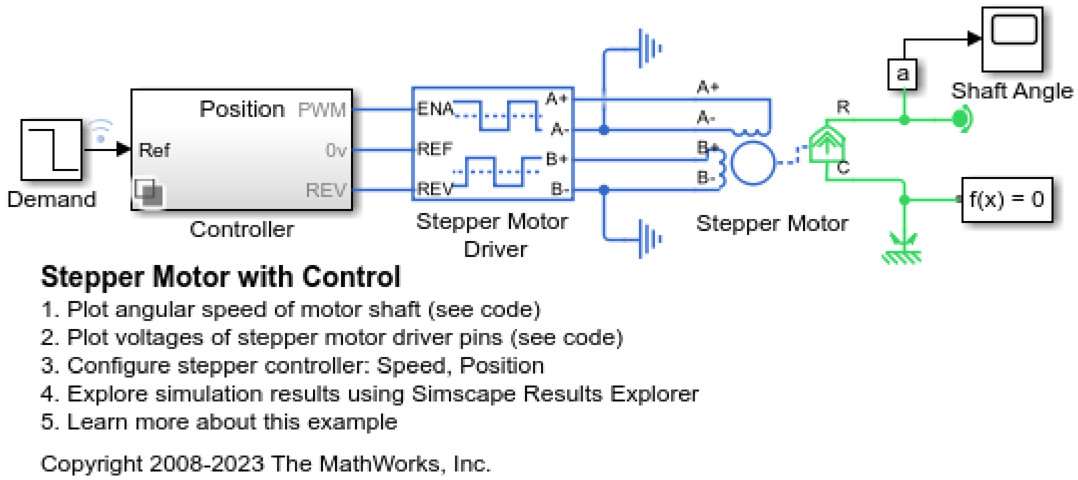
MATLAB Code:

```
% MOSFET Characteristics Plot
Vds = linspace(0, 10, 100); % Drain-source voltage range
```

```
Vgs = [2, 4, 6]; % Gate-source voltages

for Vg = Vgs
    Id = (Vg > 1) .* ((Vg - 1) .* (Vds < Vg - 1) + (Vg - 1)^2/2 .* (Vds >= Vg - 1));
    plot(Vds, Id);
    hold on;
end

hold off;
title('MOSFET I-V Characteristics');
xlabel('V_{DS} (V)');
ylabel('I_{D} (A)');
legend('V_{GS} = 2V', 'V_{GS} = 4V', 'V_{GS} = 6V');
```

84. Stepper Motor Control

Theory: Simulate the control of a stepper motor using pulse signals.



MATLAB Code:

```
% Stepper Motor Control
steps = 100; % Number of steps
step_angle = 1.8; % Step angle in degrees
angle = 0; % Initial angle

angles = zeros(1, steps);

for i = 1:steps
    angle = angle + step_angle;
    angles(i) = angle;
end

plot(1:steps, angles);
title('Stepper Motor Control');
```

```
xlabel('Step');
ylabel('Angle (degrees)');
```

85. Photodiode IV Characteristics

Theory: Plot the I-V characteristics of a photodiode.

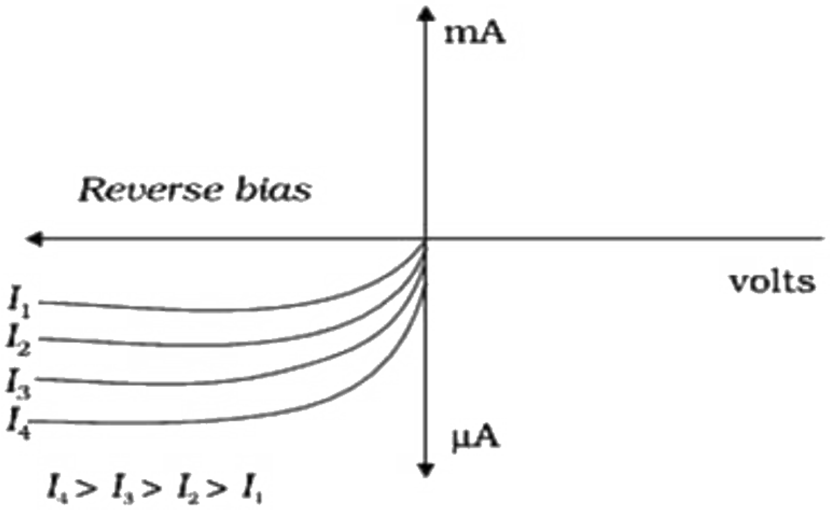
Photodiode is reverse biased p-n junction diode fabricated with a transparent window to allow light to fall on the diode.

When the photodiode is illuminated with light (photons) with energy greater than the energy gap (E_g) of the semiconductor, then electron-hole pairs are generated due to the absorption of photons , in or near the depletion region. Due to electric field of the junction, electrons and holes are separated before they recombine , electrons reach n-side and holes reach p-side.

Electrons are collected on n-side and holes are collected on p-side giving rise to an emf .When connected to an external load is, current flows , whose magnitude depends on the intensity of incident light.

The photodiode can be used as a photo detector to detect optical signals.

The I-V characteristics of a photodiode is shown below

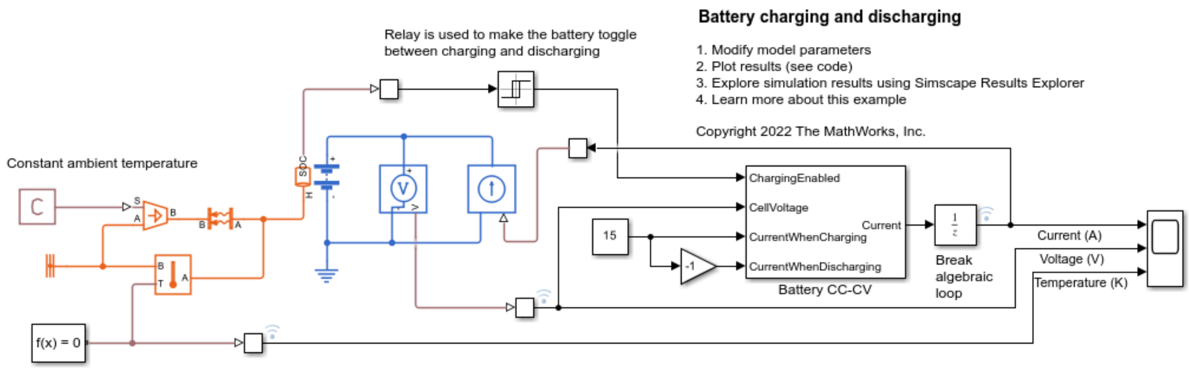


MATLAB Code:

```
% Photodiode IV Characteristics
V = linspace(-2, 2, 100); % Voltage range
I_dark = 1e-9; % Dark current in amperes
n = 1; % Ideality factor
Is = 1e-12; % Saturation current in amperes
k = 1.38e-23; % Boltzmann constant
T = 300; % Temperature in Kelvin
q = 1.6e-19; % Electron charge

I = Is * (exp((q*V)/(n*k*T)) - 1) + I_dark; % Current

plot(V, I);
title('Photodiode IV Characteristics');
xlabel('Voltage (V)');
ylabel('Current (A)');
```

86. Battery Charging Simulation

Theory: Simulate the charging process of a battery.

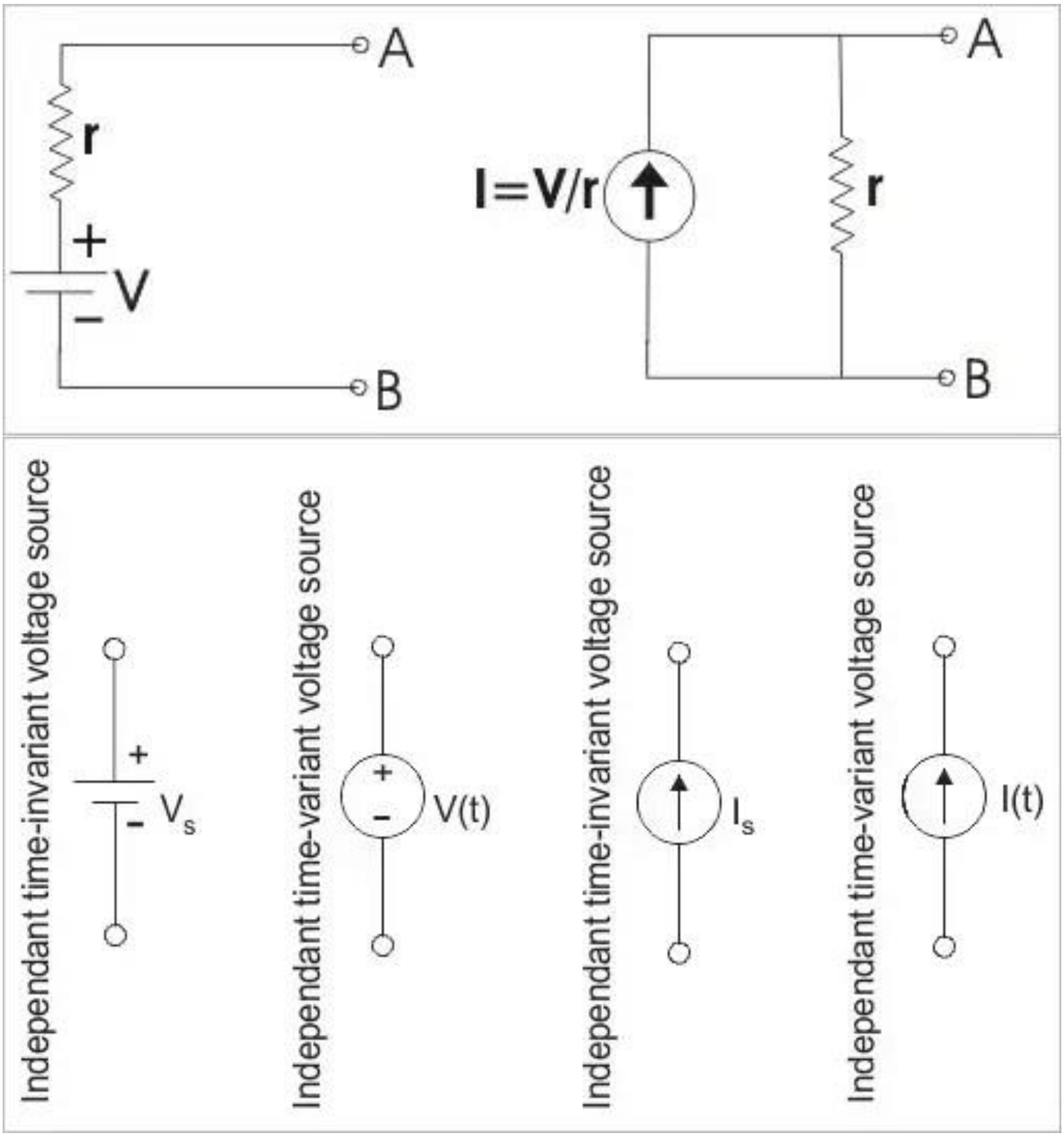


MATLAB Code:

```
% Battery Charging Simulation
V_batt = 12; % Battery voltage in volts
I_charge = 2; % Charging current in amperes
C = 60; % Battery capacity in Ah

t = linspace(0, C/I_charge, 1000); % Time vector
V = V_batt * (1 - exp(-t / (C/I_charge))); % Voltage during charging

plot(t, V);
title('Battery Charging Simulation');
xlabel('Time (hours)');
```

```
ylabel('Voltage (V)');
```

87.



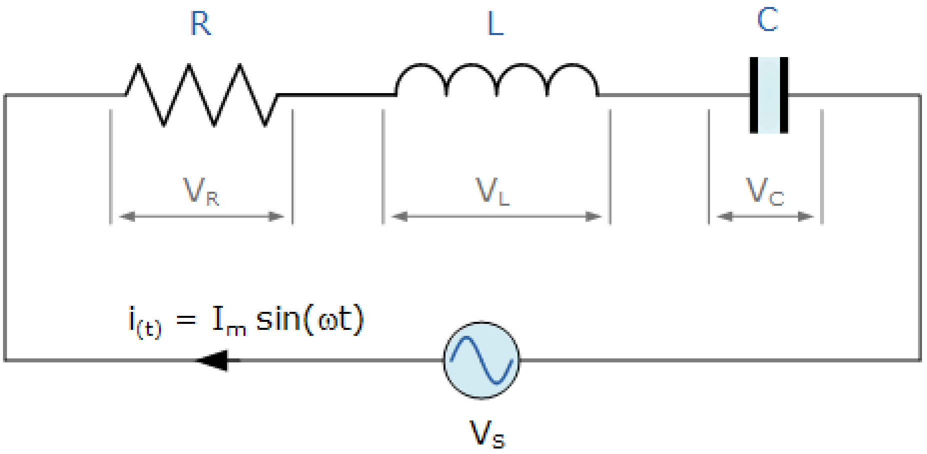
Theory: Plot the characteristics of an ideal current source.
MATLAB Code:

```
% Current Source Characteristics
I_source = 10; % Current source value in amperes
V = linspace(-10, 10, 1000); % Voltage range

I = I_source * ones(size(V)); % Current is constant
```

```
plot(V, I);
title('Current Source Characteristics');
xlabel('Voltage (V)');
ylabel('Current (A)');
```

88. Series RLC Circuit Analysis

Theory: Analyze a series RLC circuit in the time domain.

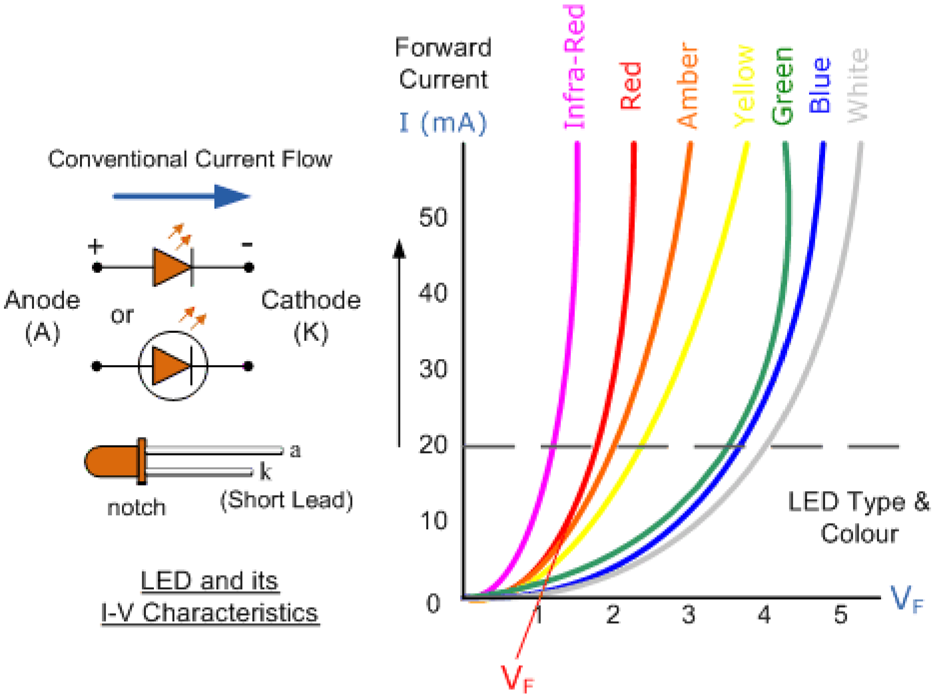


MATLAB Code:

```
% Series RLC Circuit Analysis
R = 10; % Resistance in ohms
L = 1e-3; % Inductance in henrys
C = 1e-6; % Capacitance in farads
V = 1; % Voltage step input in volts

sys = tf([1], [L*C R*C 1]);
t = linspace(0, 0.01, 1000);
step_response = step(V*sys, t);

plot(t, step_response);
title('Series RLC Circuit Analysis');
xlabel('Time (s)');
ylabel('Voltage (V)');
```

89. Light Emitting Diode (LED) Characteristics

Theory: Plot the I-V characteristics of an LED.



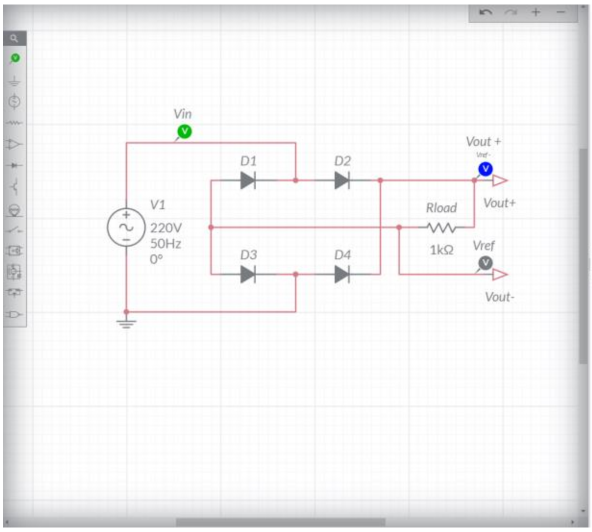
MATLAB Code:

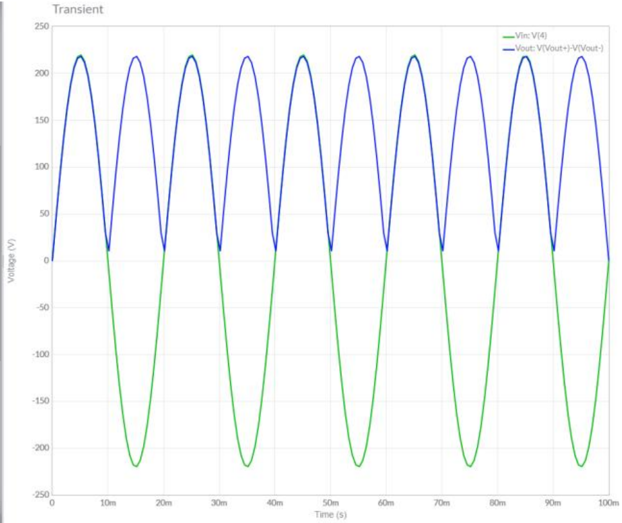
```
% LED Characteristics
V = linspace(0, 3, 100); % Voltage range
I = 1e-9 * (exp(V / 0.026) - 1); % Current

plot(V, I);
title('LED I-V Characteristics');
xlabel('Voltage (V)');
ylabel('Current (A)');
```

90. Full-Wave Rectifier Simulation

Theory: Simulate the operation of a full-wave rectifier circuit.

Bangabandhu Sheikh Mujibur Rahman Aviation and Aerospace University (BSMRAAU)
Modelling and Simulation Sessional





MATLAB Code:

```
% Full-Wave Rectifier Simulation
t = linspace(0, 0.1, 1000); % Time vector
V_in = sin(2 * pi * 50 * t); % Input AC voltage
V_out = abs(V_in); % Full-wave rectified voltage

plot(t, V_in, t, V_out);
title('Full-Wave Rectifier Simulation');
xlabel('Time (s)');
ylabel('Voltage (V)');
legend('Input Voltage', 'Output Voltage');

)]);
```