**Bangabandhu Sheikh Mujibur Rahman Aviation and Aerospace University (BSMRAAU)**



**LAB REPORT**

**Experiment No: 01**

**Name of Experiment: MATLAB and Simulink Example**

**Problems and Solutions**

**Date of Experiment Submission: 19/09/2024**

**Student Name: Md. Mahdi Kamal Alif**

**Student ID: 22024019**

**Department: Department of Aerospace Engineering (Avionics)**

**Course Name: Modelling and Simulation Sessional  
Course Code: AVE 4510**

**Submitted To: Asst. Prof. Md Samin Rahman**

**Asst. Prof. Dr. Md. Sakir Hossain**

**Remarks:**

**Modeling and Simulation Sessional**

**AVE: 4510**

Experiment 1.1: Matrix Operations

Code:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  A = [3 4; 5 6];  B = [7 8; 9 10];  % 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) |

Output:

|  |
| --- |
| Matrix Addition:  10 12  14 16  Matrix Subtraction:  -4 -4  -4 -4  Matrix Multiplication:  57 64  89 100  Element-wise Multiplication:  21 32  45 60  Matrix Division:  3 -2  2 -1 |

Experiment 1.2: Solving Linear Equations:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  A = [1 2; 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) |

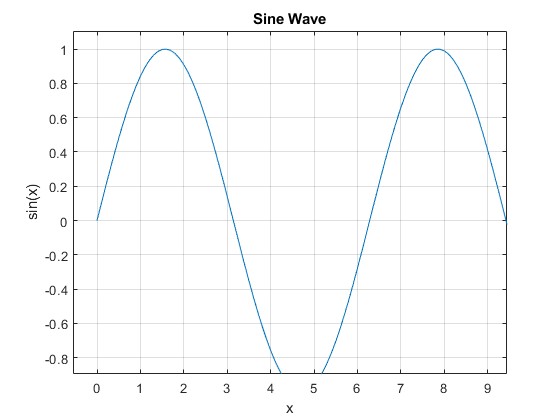
Output:

|  |
| --- |
| Solution using inverse:  -4.0000  4.5000  Solution using backslash operator:  -4.0000  4.5000 |

Experiment 2.1: Plotting Data

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  x = 0:0.1:10;  y = sin(x);  figure;  plot(x, y);  title('Sine Wave');  xlabel('x');  ylabel('sin(x)');  grid on; |

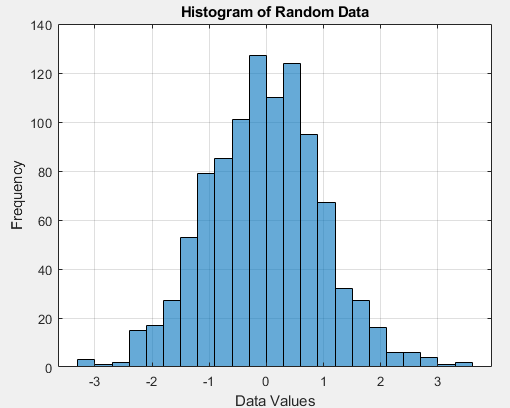
Output:



Experiment 2.2: Histogram

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  data = randn(1000, 1);  figure;  histogram(data);  title('Histogram of Random Data');  xlabel('Data Values');  ylabel('Frequency');  grid on; |

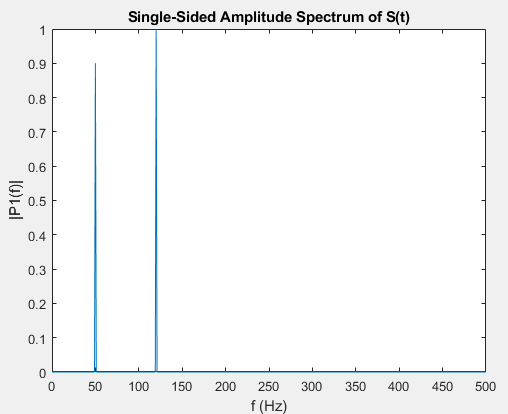
Output:



|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  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)|') |

Experiment 3.1: FFT of a Signal

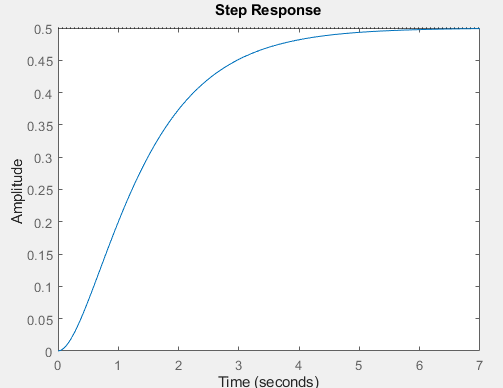
Output:



Experiment 4.1: Step Response of a Transfer Function

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  num = [1];  den = [1 3 2];  sys = tf(num, den);  figure;  step(sys);  title('Step Response'); |

Output:



Experiment 5.1: Minimizing a Function

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  fun = @(x) (x-4)^2 + 6;  x0 = 0; % Initial guess  x = fminsearch(fun, x0);    disp(‘The minimum value is at:’), disp(x) |

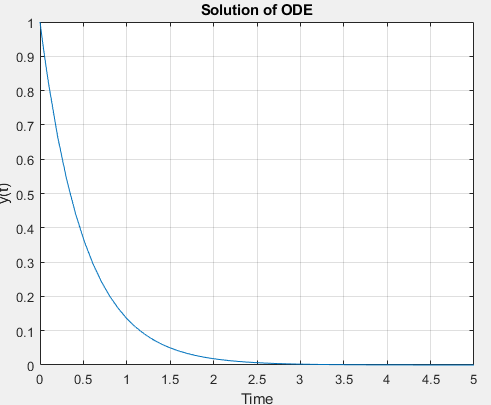
Output:

|  |
| --- |
| The minimum value is at:  4.0000 |

Experiment 6.1: Solving ODEs

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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; |

Output:



Experiment 7.1: Monte Carlo Simulation

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  %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) |

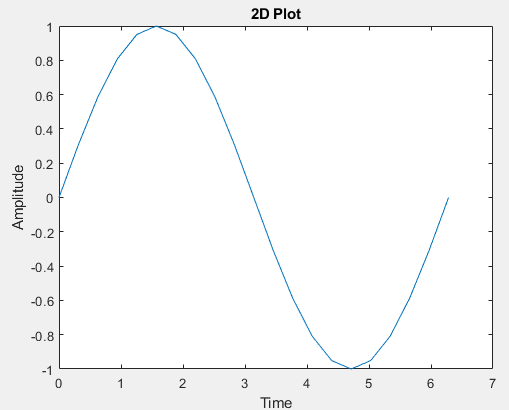
Output:

|  |
| --- |
| >> test1  Estimated value of pi:  3.1500  >> test2  Estimated value of pi:  3.1300  >> test3  Estimated value of pi:  3.1368 |

2D Plot:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  x=(0:pi/10:2\*pi)  y=sin(x)  plot(x,y)  title('2D Plot')  xlabel('Time')  ylabel('Amplitude') |

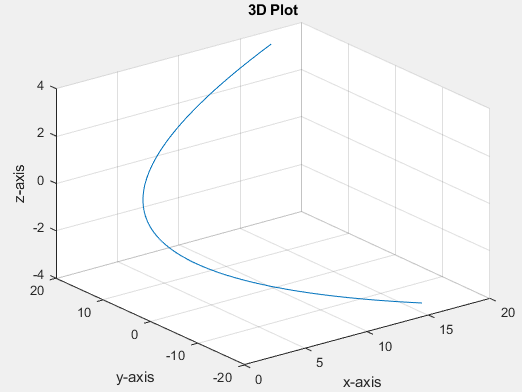
Output:



3D plot:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  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') |

Output:



Experiment 10.1: Reading and Writing CSV Files

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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) |

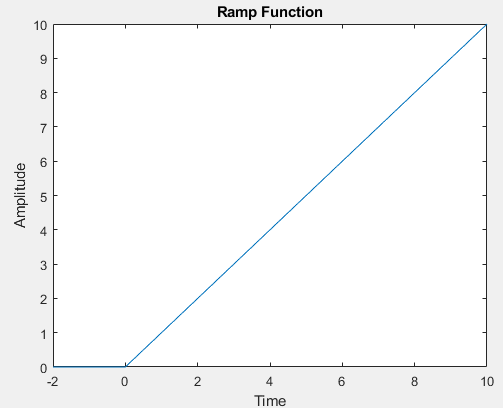
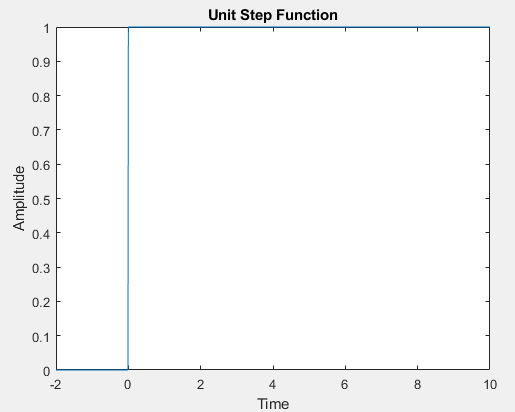
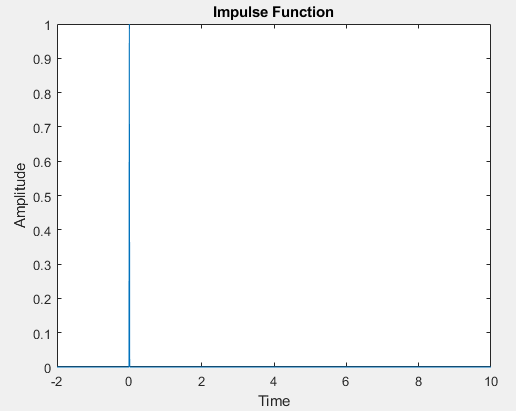
Output:

|  |
| --- |
| Data read from CSV:  0.4596 0.7866 0.0222 0.9014  0.6951 0.5821 0.2594 0.4888  0.3880 0.9930 0.2136 0.9997  0.8682 0.8606 0.3072 0.6387  0.3510 0.0358 0.5015 0.8051  0.7321 0.2902 0.7039 0.1264  0.0098 0.0452 0.8136 0.9392  0.4641 0.6310 0.7792 0.3837  0.9319 0.6587 0.1746 0.6526  0.7498 0.5415 0.6724 0.2099 |

Impulse Function, Step Function and Ramp Function:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  t = (-2:0.01:10);  impulse = t==0;  unitstep = t>=0;  ramp = t.\*unitstep;  % Plot impulse function  figure;  plot(t, impulse);  title('Impulse Function');  xlabel('Time');  ylabel('Amplitude');  % Plot unit step function  figure;  plot(t, unitstep);  title('Unit Step Function');  xlabel('Time');  ylabel('Amplitude');  % Plot ramp function  figure;  plot(t, ramp);  title('Ramp Function');  xlabel('Time');  ylabel('Amplitude'); |

Output:

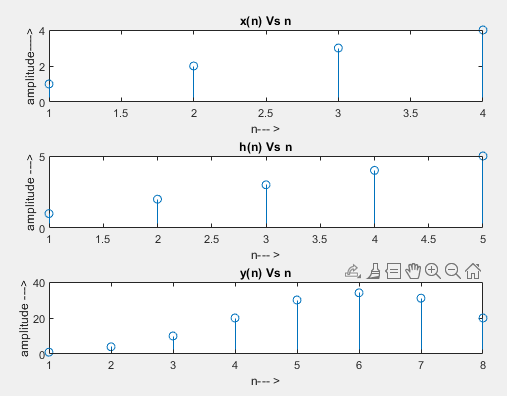


Linear Convolution:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  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'); |

Output:

|  |
| --- |
| enter the length of the first sequence m=  4  enter the first sequence x[m]=  1  2  3  4  enter the length of the second sequence n=  5  enter the second sequence h[n]=  1  2  3  4  5  linear convolution of x[m] and h[n] is y  >> |



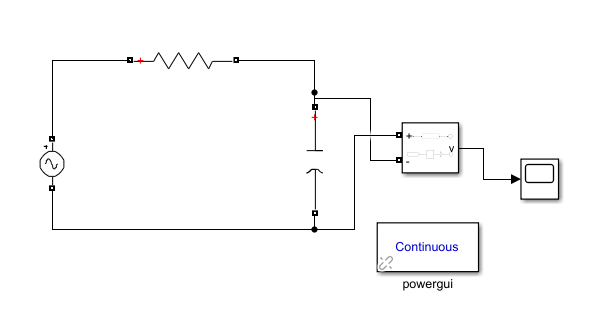
|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % Step 1: Define the resistance and current  R = 10; % resistance in ohms  I = 2; % current in amperes  % Step 2: Calculate the voltage  V = R \* I;  % Step 3: Display the result  disp(['Voltage (V) = ', num2str(V), ' V']); |

Ohm's Law Verification:

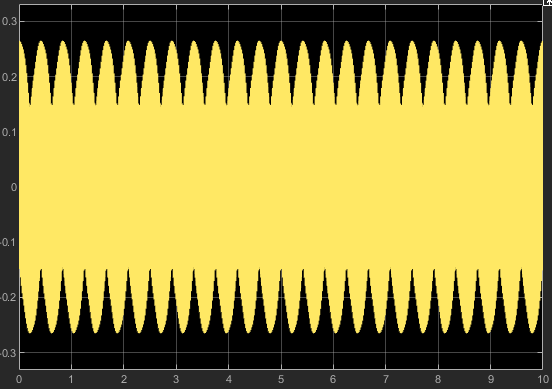
Output:

|  |
| --- |
| Voltage (V) = 20 V |

2. Series RC Circuit Response



Output:

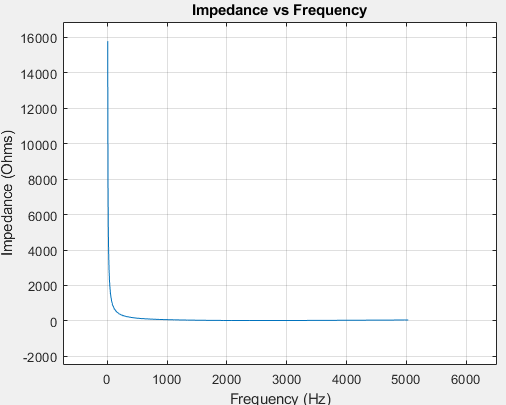


3. Parallel RLC Circuit

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % Step 1: Define the values of R, L, and C  R = 20; % resistance in ohms  L = 2e-3; % inductance in henries  C = 2e-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; |

Output:

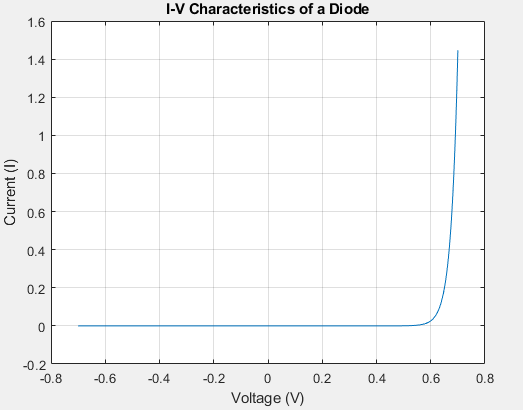
|  |
| --- |
| Resonance Frequency (Hz) = 2516.4606 |



4. Diode Characteristics

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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; |

Output:



5. Transistor as a Switch

A diagram of a circuit

Description automatically generated

Output:

A black screen with yellow lines

Description automatically generated

Full-wave Rectifier:

A diagram of a block diagram

Description automatically generated

Output:

A screenshot of a graph

Description automatically generated

8. RC Low-Pass Filter

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % Step 1: Define the values of R and C  R = 2e3; % resistance in ohms  C = 2e-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; |

Output:

A graph of a frequency response

Description automatically generated

9. RL High-Pass Filter

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % Step 1: Define the values of R and L  R = 2e3; % resistance in ohms  L = 2e-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; |

Output:

A graph of a function

Description automatically generated

10. BJT Amplifier

A diagram of a circuit

Description automatically generated

Output:

A screen shot of a graph

Description automatically generated

4. Fourier Transform:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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);  axis([-500 1500 0 1000]);  title('Fourier Transform');  xlabel('Frequency (Hz)');  ylabel('Magnitude'); |

Output:

A graph of a function

Description automatically generated

6. AC Voltage Analysis:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % AC Voltage Analysis  Fs = 1000; % Sampling frequency  t = 0:1/Fs:1-1/Fs;  f = 50; % Frequency of AC voltage  V\_peak = 20; % Peak voltage  V\_ac = V\_peak \* sin(2\*pi\*f\*t);  V\_rms = rms(V\_ac);  disp(['RMS Voltage: ', num2str(V\_rms), ' V']); |

Output:

|  |
| --- |
| RMS Voltage: 14.1421 V |

7. DC Motor Simulation:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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'); |

Output:

A graph with a line

Description automatically generated

8. LED Characteristic Curve:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % LED Characteristic Curve  V = 0:0.01:4; % 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)'); |

Output:

A graph of a voltage

Description automatically generated with medium confidence

9. BJT Common Emitter Amplifier:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % BJT Common Emitter Amplifier  beta = 100; % Current gain  R\_C = 1e3; % Collector resistor in ohms  V\_CC = 15; % 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']); |

Output:

|  |
| --- |
| Collector-Emitter Voltage: 10 V |

10. Diode Clipper Circuit:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % Diode Clipper Circuit  t = 0:0.001:1; % Time vector  Vi = 5\*sin(2\*pi\*20\*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'); |

Output:

A diagram of a diode clipper circuit

Description automatically generated

11. Transmission Line Impedance:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % Transmission Line Impedance  L = 1e-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']); |

Output:

|  |
| --- |
| Characteristic Impedance: 707.1068 Ohms |

12. Phase Shift in RC Circuit:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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']); |

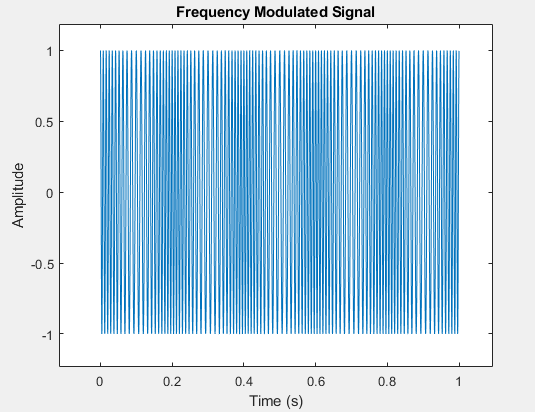
Output:

|  |
| --- |
| Phase Shift: 9.0431 degrees |

13. Frequency Modulation (FM):

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  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'); |

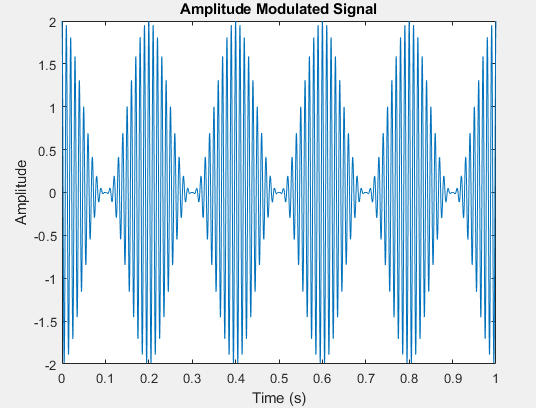
Output:



14. AM Modulation:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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'); |

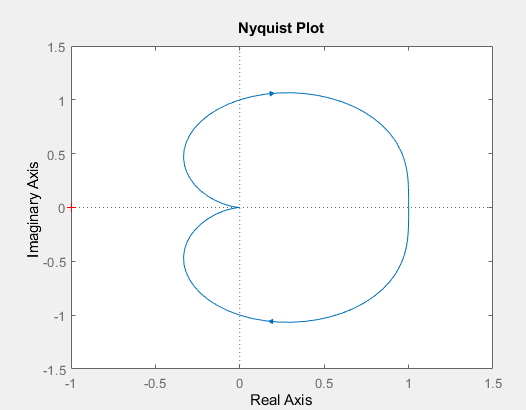
Output:



15. Nyquist Plot:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % Nyquist Plot  sys = tf([1], [1 1 1]);  nyquist(sys);  title('Nyquist Plot'); |

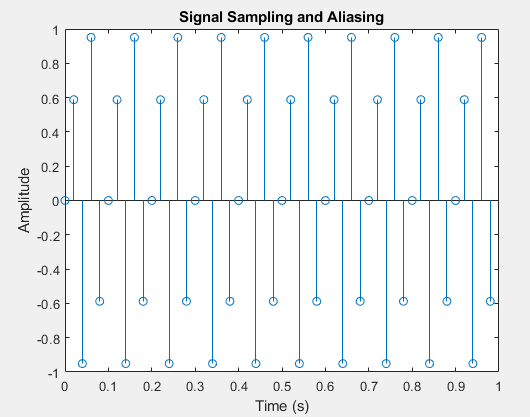
Output:



17. Signal Sampling and Aliasing:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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'); |

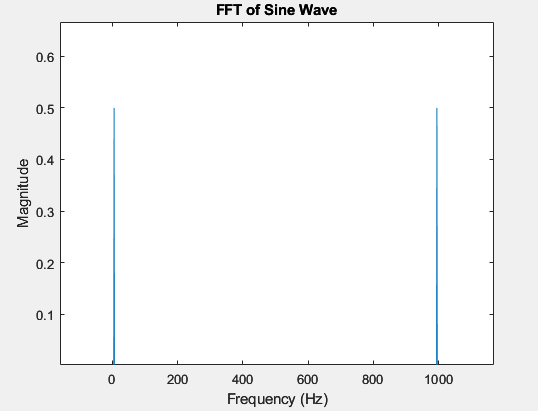
Output:



18. Fast Fourier Transform (FFT):

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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'); |

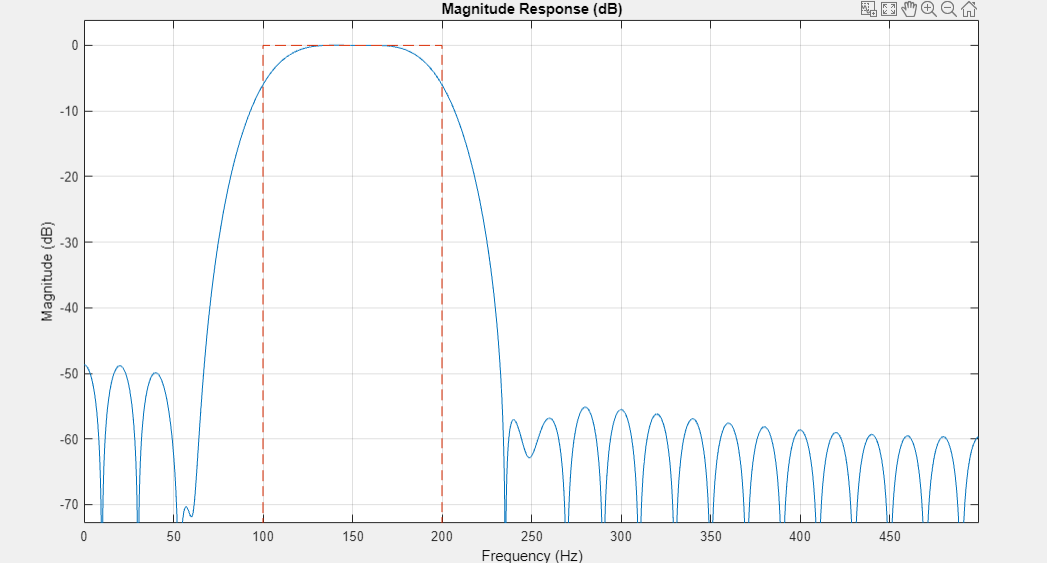
Output:



19. Bandpass Filter Design:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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'); |

Output:



20. Operational Amplifier (Op-Amp) Inverting Amplifier:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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']); |

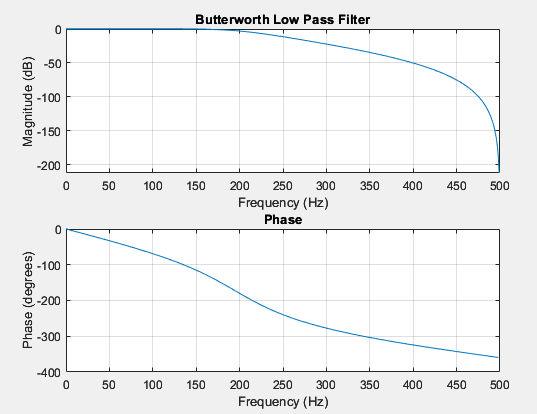
Output:

|  |
| --- |
| Output Voltage: -10 V |

21. Butterworth Low Pass Filter Design:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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'); |

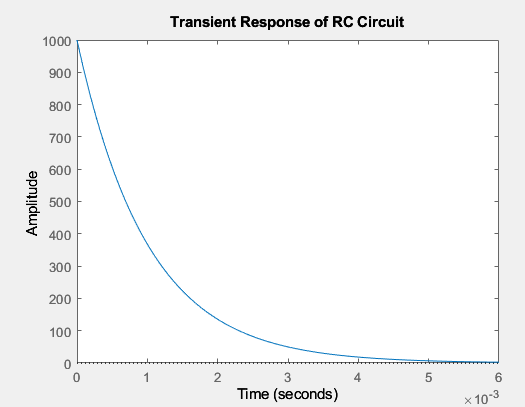
Output:



22. Transient Analysis of RC Circuit:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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'); |

Output:



23. Voltage Divider Rule:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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']); |

Output:

|  |
| --- |
| Output Voltage: 6.6667 V |

24. Digital to Analog Conversion (DAC):

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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'); |

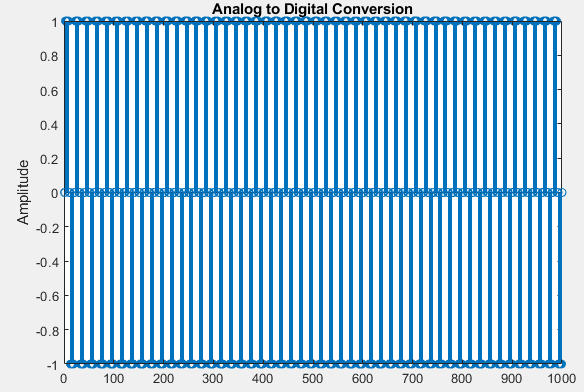
Output:

|  |
| --- |
| Output Voltage: 6.6667 V |

25. Analog to Digital Conversion:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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'); |

Output:



26. Impedance of RLC Circuit:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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']); |

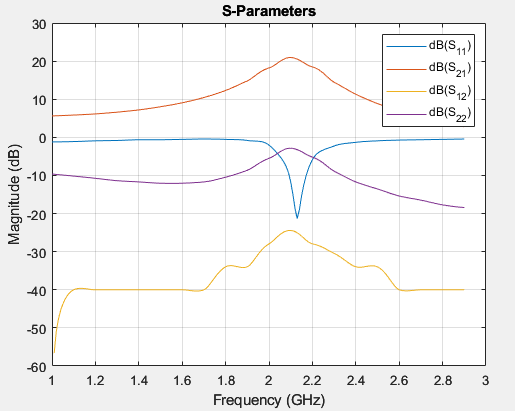
Output:

|  |
| --- |
| Impedance: 232.9875 Ohms |

27. S-Parameters in RF Circuit:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % S-Parameters in RF Circuit  s\_params = sparameters('default.s2p'); % Load S-parameter file  rfplot(s\_params);  title('S-Parameters'); |

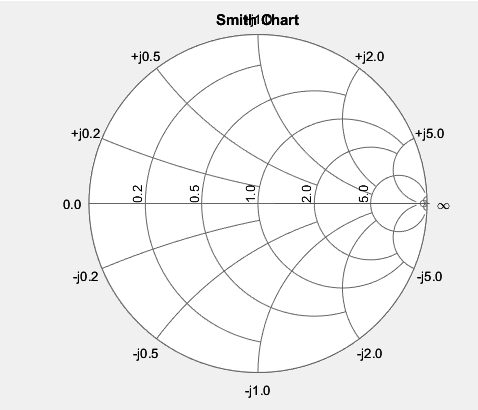
Output:



28. Smith Chart:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % Smith Chart  z = linspace(0.1, 10, 100); % Normalized impedance values  smithchart(z);  title('Smith Chart'); |

Output:



29. Resonance in RLC Circuit:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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']); |

Output:

|  |
| --- |
| Resonance Frequency: 15915.4943 Hz |

30. Stability Analysis Using Routh-Hurwitz Criterion:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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); |

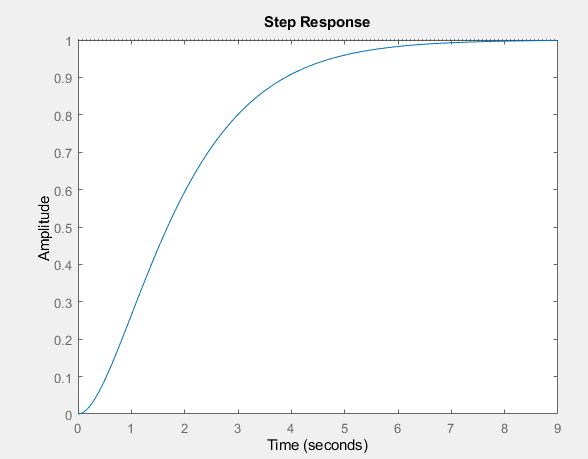
31. Power Spectral Density (PSD):

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % Step Response of a Transfer Function  sys = tf([1], [1 2 1]);  step(sys);  title('Step Response'); |

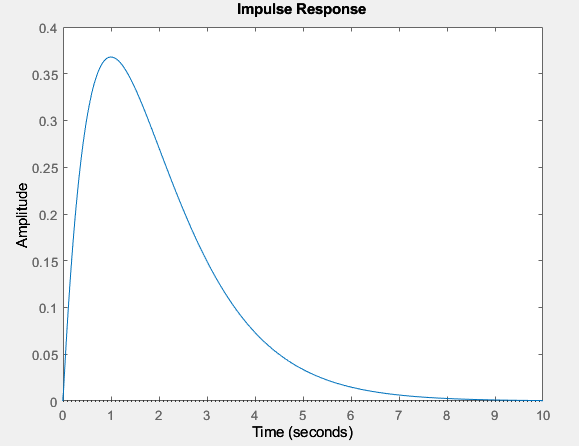
Output:



33. Impulse Response of a Transfer Function:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % Impulse Response of a Transfer Function  sys = tf([1], [1 2 1]);  impulse(sys);  title('Impulse Response'); |

Output:



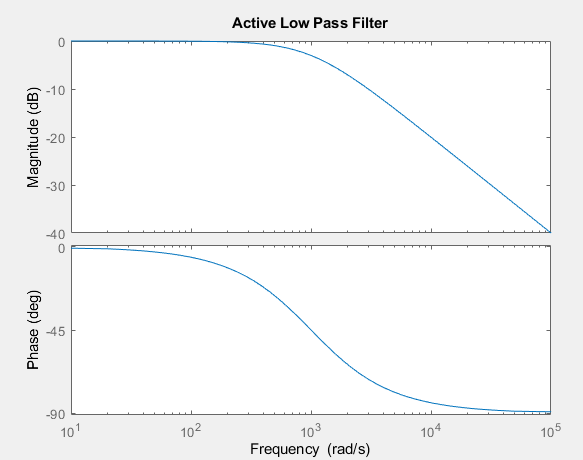
34. Phase Locked Loop (PLL) Simulation:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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'); |

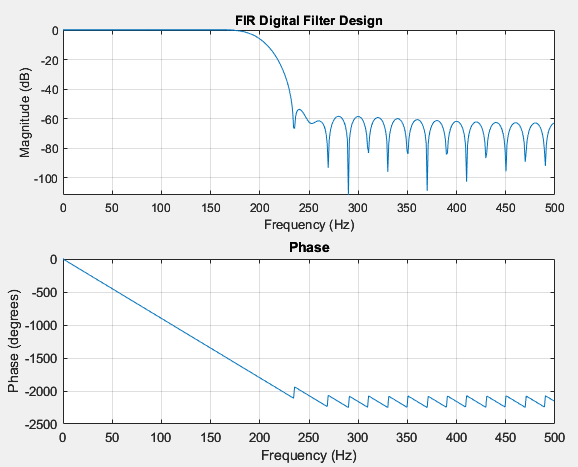
Output:



36. Digital Filter Design (FIR):

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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'); |

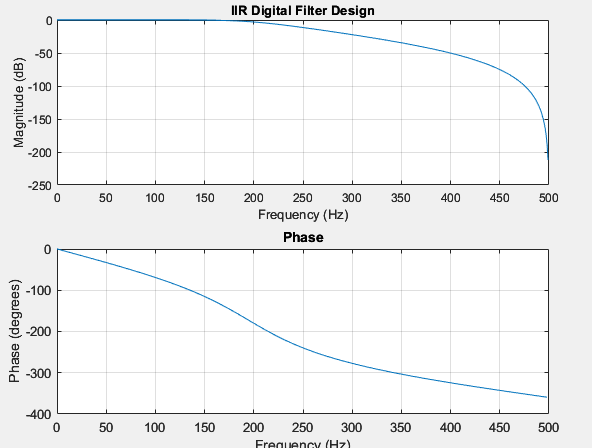
Output:



37. Digital Filter Design (IIR):

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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'); |

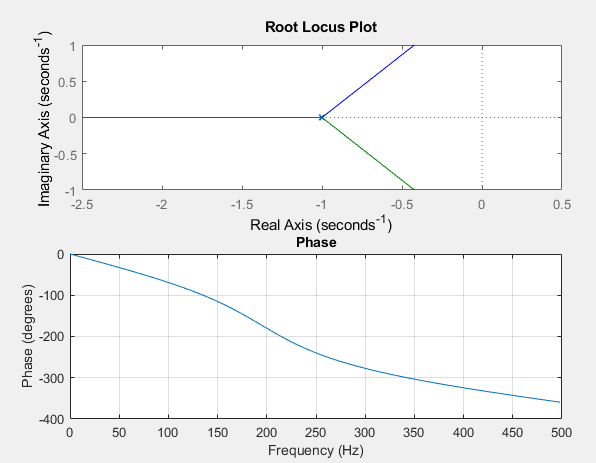
Output:



38. Root Locus Plot:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % Root Locus Plot  sys = tf([1], [1 3 3 1]);  rlocus(sys);  title('Root Locus Plot'); |

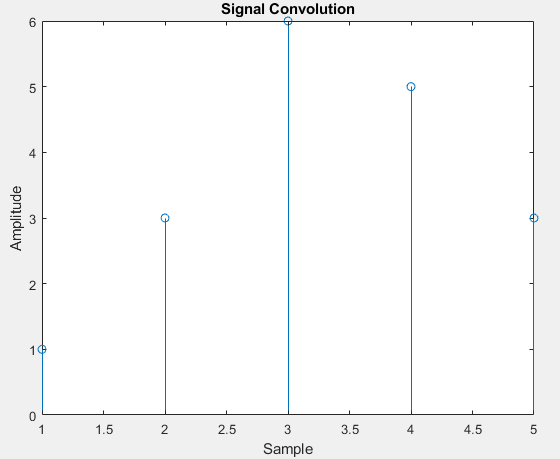
Output:



39. Signal Convolution:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % Signal Convolution  x = [1 2 3];  h = [1 1 1];  y = conv(x, h);  stem(y);  title('Signal Convolution');  xlabel('Sample');  ylabel('Amplitude'); |

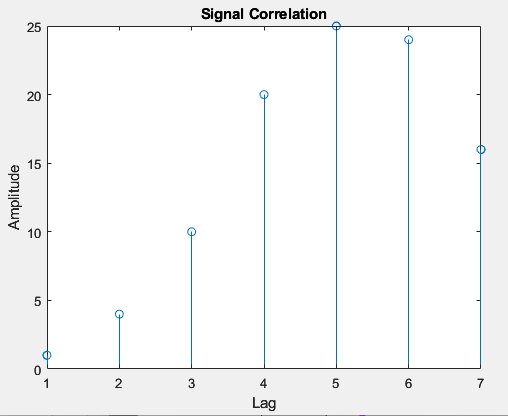
Output:



40. Signal Correlation:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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'); |

Output:



41. Power Factor Calculation:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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)]); |

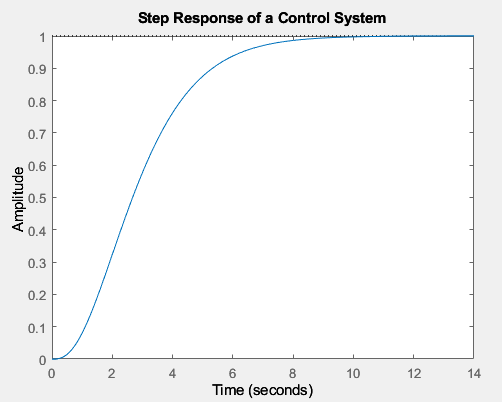
Output:

|  |
| --- |
| Power Factor: 0.8 |

42. Step Response of a Control System:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % Step Response of a Control System  sys = tf([1], [1 3 3 1]);  step(sys);  title('Step Response of a Control System'); |

Output:



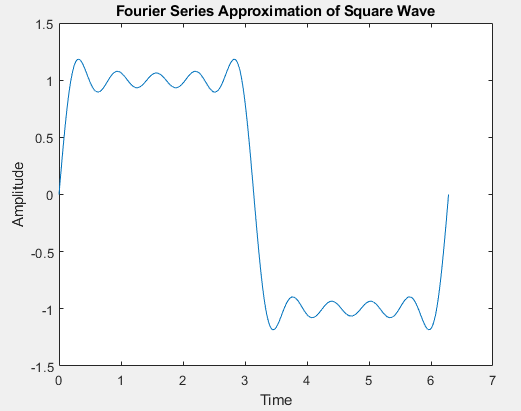
43. Nyquist Stability Criterion:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % Nyquist Stability Criterion  sys = tf([1], [1 3 3 1]);  nyquist(sys);  title('Nyquist Stability Criterion'); |

44. Fourier Series Approximation:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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'); |

Output:



45. Transmission Line Reflection Coefficient:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % Transmission Line Reflection Coefficient  Z0 = 50; % Characteristic impedance  ZL = 100; % Load impedance  reflection\_coefficient = (ZL - Z0) / (ZL + Z0);  disp(['Reflection Coefficient: ', num2str(reflection\_coefficient)]); |

Output:

|  |
| --- |
| Reflection Coefficient: 0.33333 |

46. State Space Representation:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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']); |

Output:

|  |
| --- |
| Electrostatic Force: 179.7548 N |

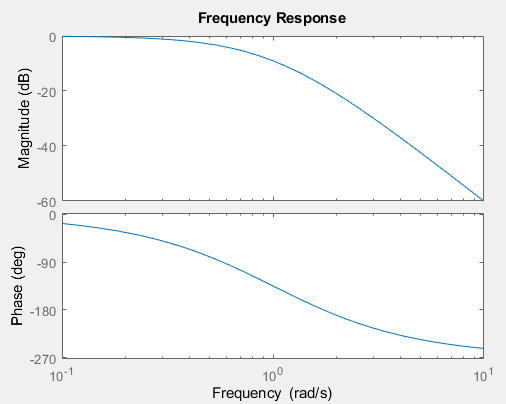
48. Electromagnetic Wave Propagation:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % Frequency Response of a System  sys = tf([1], [1 3 3 1]);  bode(sys);  title('Frequency Response'); |

Output:



50. Pulse Width Modulation (PWM):

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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)'); |

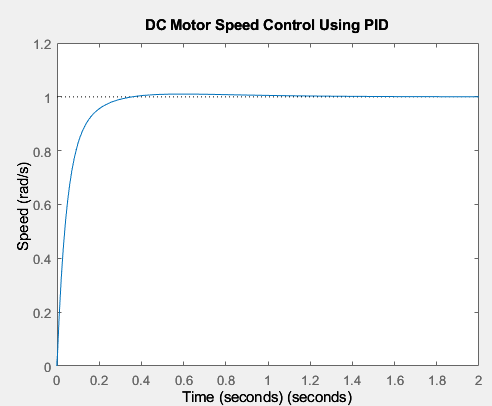
52. PID Controller Design:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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)'); |

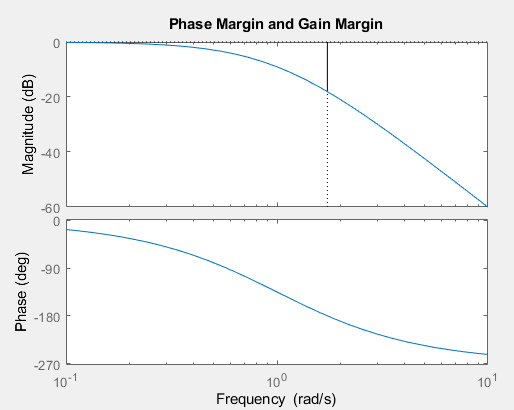
Output:



54. Phase Margin and Gain Margin:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % Phase Margin and Gain Margin  sys = tf([1], [1 3 3 1]);  margin(sys);  title('Phase Margin and Gain Margin'); |

Output:



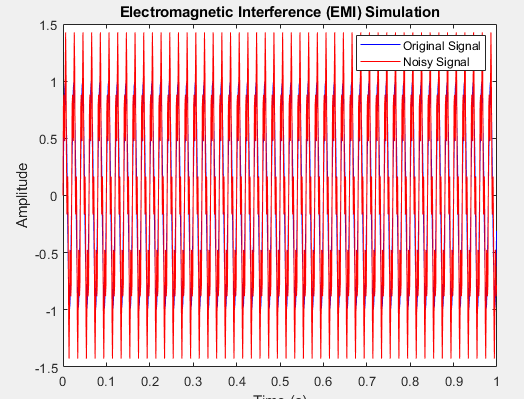
55. State Observer Design:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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'); |

Output:



57. Transmission Line Impedance Matching:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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']); |

Output:

|  |
| --- |
| Matching Impedance: 70.7107 Ohms |

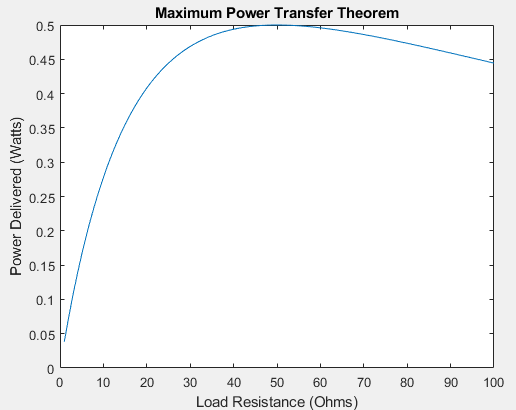
58. Switched-Mode Power Supply (SMPS) Simulation:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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)'); |

61. Maximum Power Transfer Theorem:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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)'); |

Output:



62. Mutual Inductance Calculation:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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']); |

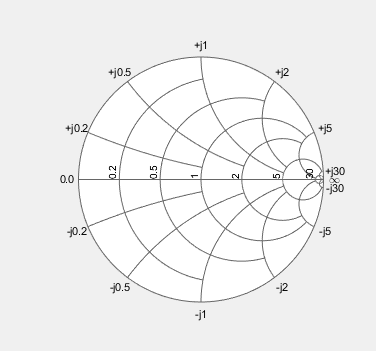
Output:

|  |
| --- |
| Mutual Inductance: 0.0070711 H |

63. Impedance Matching Using Smith Chart:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % Impedance Matching Using Smith Chart  z\_load = 2 + 3j; % Normalized load impedance  smithplot(z\_load);  title('Impedance Matching Using Smith Chart'); |

Output:



64. Sallen-Key Low Pass Filter:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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']); |

Output:

|  |
| --- |
| Thermistor Resistance: 10000 Ohms |

66. Phase Shifter Circuit:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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']); |

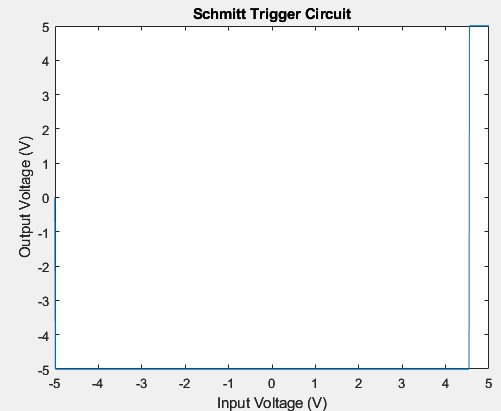
Output:

|  |
| --- |
| Phase Shift: 9.0431 degrees |

67. Schmitt Trigger Circuit:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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)'); |

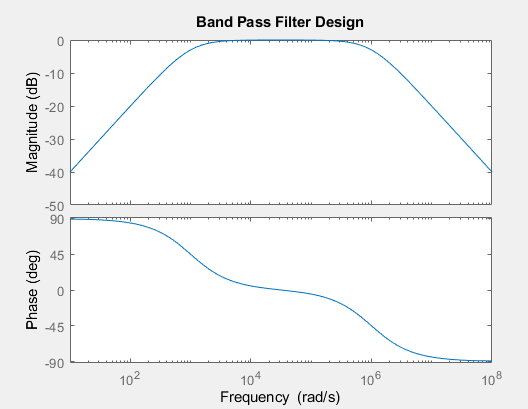
Output:



68. Band Pass Filter Design:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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'); |

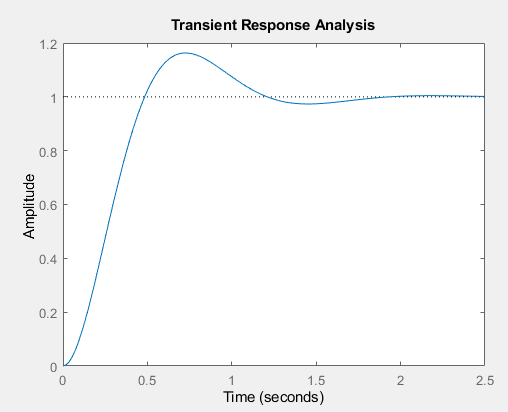
Output:



69. Transient Response Analysis:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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'); |

Output:



70. Delta-Wye Transformation:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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']); |

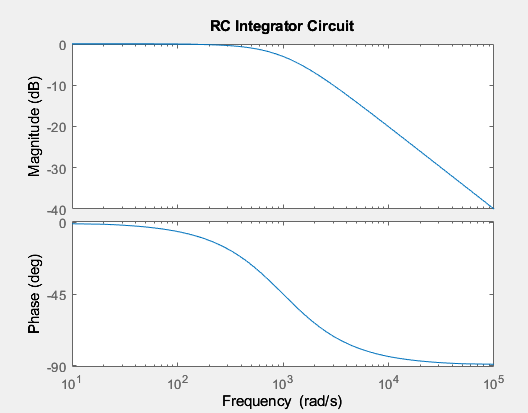
Output:

|  |
| --- |
| Z1: 16.6667 Ohms  Z2: 16.6667 Ohms  Z3: 16.6667 Ohms |

71. RC Integrator Circuit

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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'); |

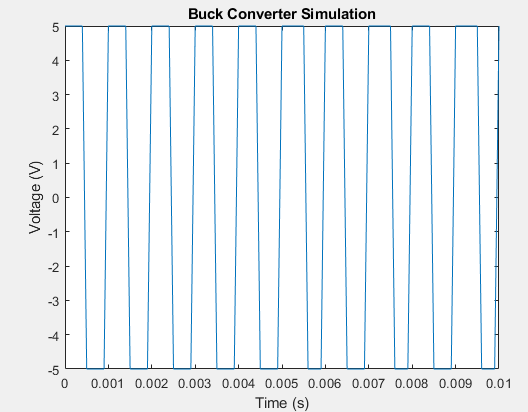
Output:



72. Buck Converter Simulation:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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)'); |

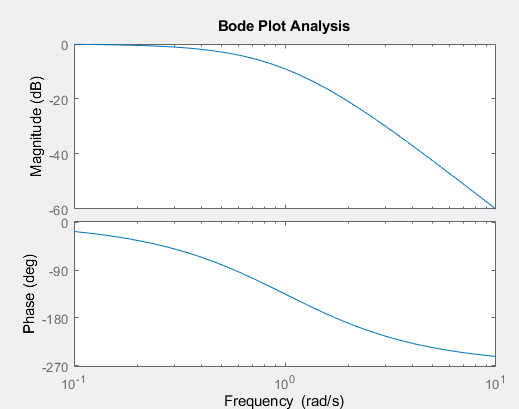
Output:



73. Bode Plot Analysis

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % Bode Plot Analysis  sys = tf([1], [1 3 3 1]);  bode(sys);  title('Bode Plot Analysis'); |

Output:



74. Root Mean Square (RMS) Calculation:

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % Root Mean Square (RMS) Calculation  t = linspace(0, 2\*pi, 1000);  x = sin(t);  RMS = sqrt(mean(x.^2));  disp(['RMS Value: ', num2str(RMS)]); |

Output:

|  |
| --- |
| RMS Value: 0.70675 |

75. AC Voltage Regulation

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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), '%']); |

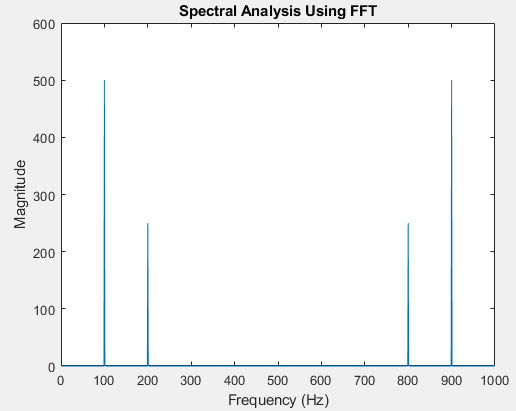
Output:

|  |
| --- |
| Voltage Regulation: 4.5455% |

76. Spectral Analysis Using FFT

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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'); |

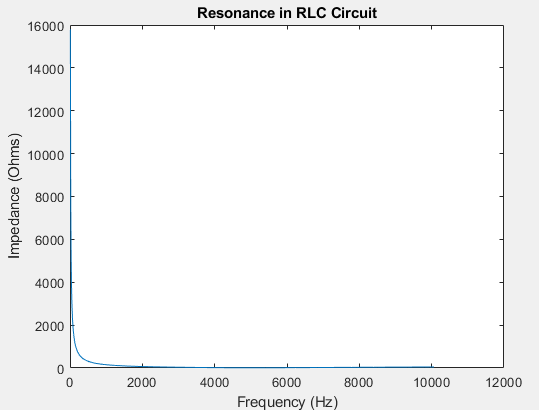
Output:



77. Resonance in RLC Circuit

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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)'); |

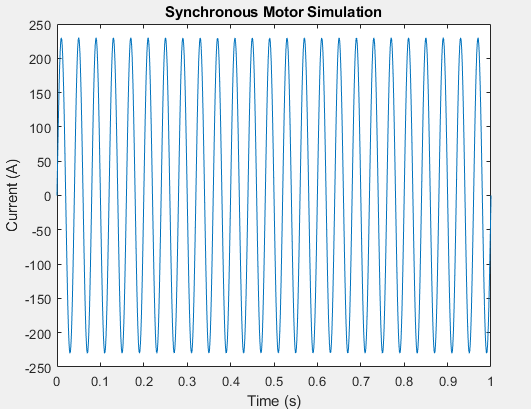
Output:



78. Synchronous Motor Simulation

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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)'); |

Output:



79. Power System Load Flow Analysis

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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)); |

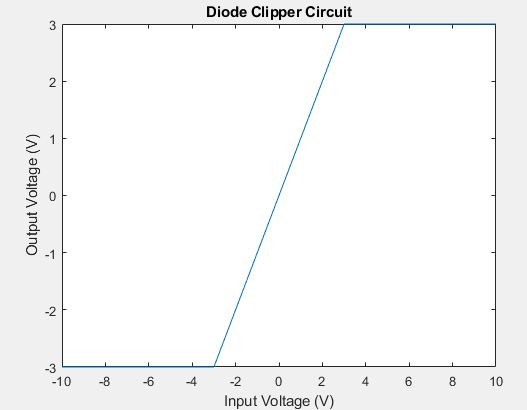
Output:

|  |
| --- |
| Voltage magnitudes (pu):  1.0e+14 \*  -1.3511  -1.3511  -1.3511  Voltage angles (degrees):  1.0e+15 \*  -3.8706  -3.8706  -3.8706 |

80. Diode Clipper Circuit

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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)'); |

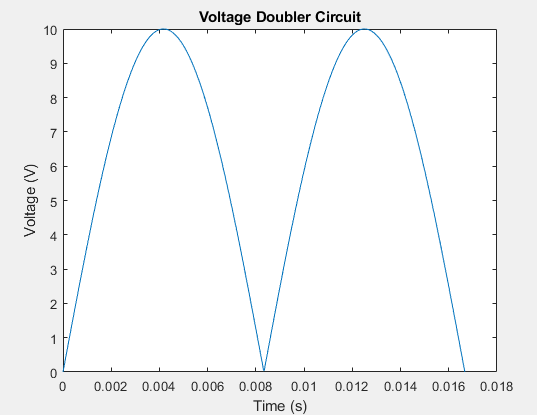
Output:



81. Voltage Doubler Circuit

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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)'); |

Output:



82. Three-Phase Power Calculation

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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']); |

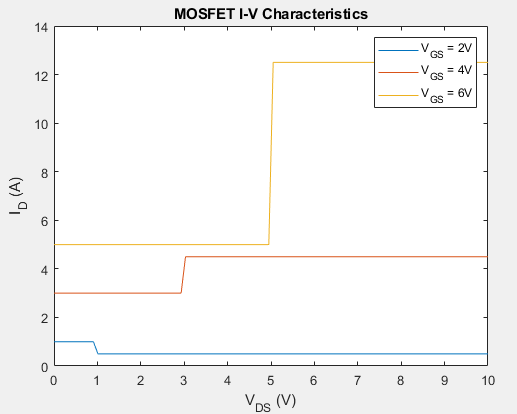
Output:

|  |
| --- |
| Total Real Power: 5520 W  Total Reactive Power: 4140 VAR |

83. MOSFET Characteristics Plot

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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'); |

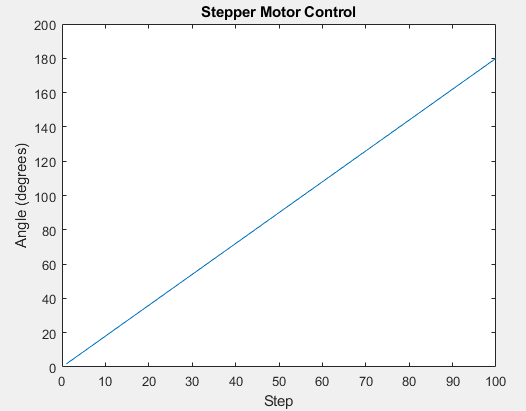
Output:



84. Stepper Motor Control

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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)'); |

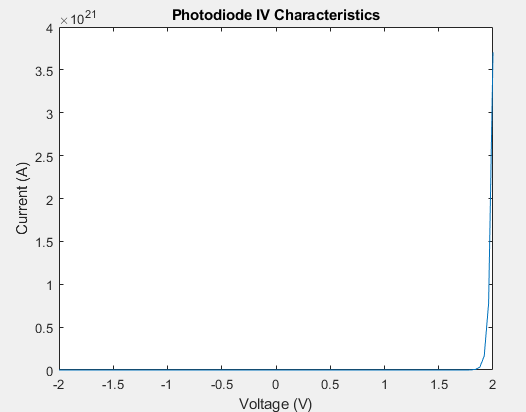
Output:



85. Photodiode IV Characteristics

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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)'); |

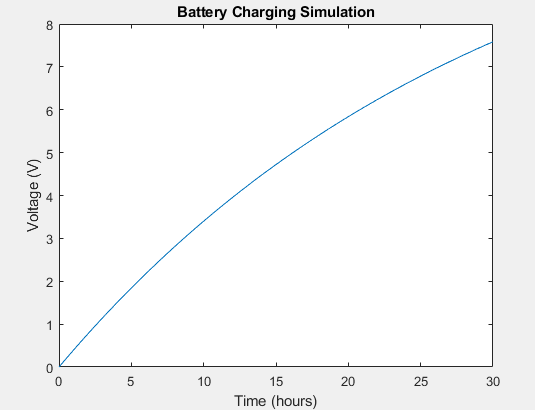
Output:



86. Battery Charging Simulation

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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)'); |

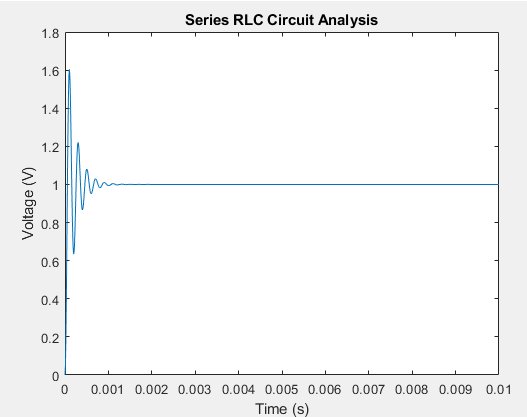
Output:



88. Series RLC Circuit Analysis

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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)'); |

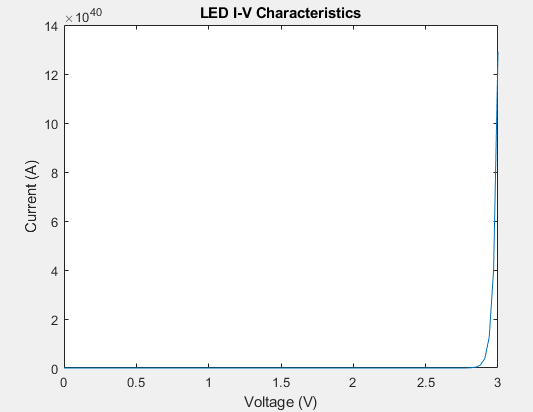
Output:



89. Light Emitting Diode (LED) Characteristics

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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)'); |

Output:



90. Full-Wave Rectifier Simulation

|  |
| --- |
| %Author: Md. Mahdi Kamal Alif  %ID: 22024019  % 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'); |

Output:

