ECE 811 – SOFTWARE ENGINEERING

MATLAB FOR ELECTRICAL ENGINEERS - STUDY GUIDE

---

1. INTRODUCTION TO MATLAB FOR EE

1. Why MATLAB For Electrical Engineering?

MATLAB is a crucial tool for electrical engineers due to its powerful capabilities in simulation, analysis, and algorithm development.

MATLAB enables electrical engineers to model and simulate complex electrical systems, design control systems, and analyse signals and systems, making it essential for various applications like power systems, electronics, and communication systems.

MATLAB is the industry-standard for electrical engineering simulations, data analysis, and hardware integration (Arduino, FPGAs).

2. Key MATLAB Toolboxes:

MATLAB toolboxes are collections of pre-built functions, classes, apps, and other resources that extend MATLAB's core functionality for specific domains or tasks.

Key MATLAB toolboxes for electrical engineers are:

1. Simulink
2. Control System Toolbox
3. Signal Processing Toolbox
4. Simscape Electrical (circuit simulation)
5. Power System Toolbox

2. CORE MATLAB SYNTAX

```
% Variables & Operations
R = 100; C = 1e-6;        % Resistance (Ω), Capacitance (F)
tau = R * C;              % Time constant


% Complex Numbers (Phasors)
V = 10 * exp(1j*pi/4);    % 10∠45° V


% Matrices (for mesh analysis)
A = [2, -1; -1, 3];       % Impedance matrix
```

```
B = [5; 0];              % Voltage vector
I = A\B;                 % Solve for currents
```

3. CIRCUIT ANALYSIS

- **Transient RC Circuit:**

```
t = 0:1e-5:0.1;          % Time vector (0 to 100ms)
Vc = 10*(1-exp(-t/tau)); % Capacitor voltage
plot(t, Vc); grid on;
xlabel('Time (s)');
ylabel('V_c (V)');
```

- **AC Analysis:**

```
f = 50;
w = 2*pi*f;    % Frequency (Hz), angular freq (rad/s)
Zc = 1/(1j*w*C);       % Capacitive impedance
```

4. SIGNALS & SYSTEMS

- **FFT for Harmonics:**

```
Fs = 1000; t = 0:1/Fs:1;
x = 5*sin(2*pi*50*t) + 1*sin(2*pi*150*t); % Signal + harmonic
Y = fft(x); P2 = abs(Y/length(x)); P1 = P2(1:length(x)/2+1);
f = Fs*(0:length(x)/2)/length(x);
plot(f, P1); title('Harmonic Spectrum');
```

- **Convolution:**

```
h = [1, 0.5, 0.25];     % Impulse response
y = conv(x, h);         % System output
```

5. CONTROL SYSTEMS

```
s = tf('s');
sys = 1/(s^2 + 2*s + 5);  % Transfer function
step(sys);                % Step response
bode(sys); grid on;       % Bode plot
margin(sys);              % Stability margins
```

6. POWER SYSTEMS

1. **3-Phase Voltage**

```
t = 0:0.0001:0.04;      % 50Hz period
Va = 220*sin(2*pi*50*t);
Vb = 220*sin(2*pi*50*t - 2*pi/3);
Vc = 220*sin(2*pi*50*t + 2*pi/3);
plot(t, Va, t, Vb, t, Vc);
```

2. **Load Flow:**

```
Use power_loadflow (Power System Toolbox).
```

7. ELECTRONICS

1. **Diode I-V Curve**

```
Vd = -0.7:0.01:0.7;
Id = 1e-12*(exp(Vd/0.026) - 1); % Shockley diode equation
plot(Vd, Id);
```

2. **Active Filter Design**

```
[b, a] = butter(4, 100/(Fs/2), 'low'); % 4th-order Butterworth LPF
freqz(b, a);                           % Frequency response
```

8. SIMULINK FOR ELECTRICAL ENGINEERS

1. **Key Blocks**
    1. Simscape → Electrical → Resistor, Capacitor, Op-Amp
    2. Sources → AC Voltage, PWM Generator
    3. Sinks → Scope, Spectrum Analyzer
2. **Use Cases**
    1. Motor control systems
    2. Grid-tied inverters
    3. Digital communication systems

9. HARDWARE INTEGRATION

1. **Arduino**

```
a = arduino('COM3', 'Uno');
writePWMVoltage(a, 'D9', 3.3); % Set PWM pin
```

2. **FPGA**

```
Use HDL Coder to generate VHDL/Verilog.
```

10. PROJECTS

1. Design a PID controller for a DC motor
2. Simulate a solar inverter with MPPT
3. Analyze power quality in a microgrid