# Lab 01: NumPy in Electrical & Electromechanical Engineering

## 1. Library of NumPy:

# What is NumPy?

NumPy is a Python library used for working with arrays.

It also has functions for working in domain of linear algebra, fourier transform, and matrices.

NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.

NumPy stands for Numerical Python.

# Why Use NumPy?

In Python we have lists that serve the purpose of arrays, but they are slow to process.

NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.

The array object in NumPy is called `ndarray`, it provides a lot of supporting functions that make working with `ndarray` very easy.

Arrays are very frequently used in data science, where speed and resources are very important.

## ◆ Exercise 1: RLC Circuit – Impedance Matrix

An RLC network is modeled by the impedance matrix (Ω):

$$Z = \begin{bmatrix} 5 & -2 & 0 \\ -2 & 8 & -3 \\ 0 & -3 & 6 \end{bmatrix}$$

### Tasks

1. Create the matrix
2. Compute the transpose
3. Compute the determinant
4. Check if the matrix is invertible
5. Compute the inverse (if possible)

## 🔷 Exercise 2: Mesh Currents – Linear System

Currents satisfy:

$$\begin{cases} 4I_1 - I_2 = 12 \\ -2I_1 + 5I_2 = 3 \end{cases}$$

### Tasks

1. Write matrices **A** and **B**
2. Solve the system
3. Verify the solution

## 🔷 Exercise 3: DC Motor – State Space Analysis

State matrix:

$$A = \begin{bmatrix} 0 & 1 \\ -4 & -3 \end{bmatrix}$$

### Tasks

1. Compute eigenvalues
2. Check stability
3. Compute the trace and determinant

## 🔷 Exercise 4: Power Grid – Load Flow Matrix

Load flow coefficient matrix:

$$L = \begin{bmatrix} 12 & -4 & -2 \\ -4 & 10 & -6 \\ -2 & -6 & 14 \end{bmatrix}$$

### Tasks

1. Compute row-wise sum
2. Compute column-wise sum
3. Find the bus with highest total load impact

## 🔷 Exercise 5: Renewable Energy – Solar Panel Power

Voltage (V) and Current (A) over 4 hours:

```
text

V = [220, 225, 230, 235]
I = [5.2, 5.5, 5.8, 6.0]
```

### Tasks

1. Compute instantaneous power
2. Compute average power
3. Reshape power into a 2×2 matrix

## 🔷 Exercise 6: Wind Farm – Speed Correlation

Wind speed matrix (m/s):

$$W = \begin{bmatrix} 6 & 7 & 8 \\ 5 & 6 & 7 \\ 7 & 8 & 9 \\ 6 & 7 & 8 \end{bmatrix}$$

### Tasks

1. Compute mean speed per turbine
2. Compute standard deviation
3. Compute covariance matrix

## 🔷 Exercise 7: Electromechanical System – Torque Matrix

Torque data (Nm):

$$T = \begin{bmatrix} 10 & 12 & 14 \\ 11 & 13 & 15 \\ 9 & 11 & 13 \end{bmatrix}$$

### Tasks

1. Normalize the data
2. Find max torque per column
3. Extract the second column

## 🔷 Exercise 8 (Mini Challenge): Energy Storage Matrix

Battery charge levels (%):

1. Create a **5×4** random matrix
2. Find batteries below **30%**
3. Replace values <30 with **30**

## 2. Homework NumPy in Electrical & Electromechanical Engineering

### System Description

An **electric DC motor** is supplied by a voltage source.

You are given measurements of voltage, current, speed, and torque over time.

```python
import numpy as np

t = np.linspace(0, 5, 6)            # time (s)
V = np.array([12, 12, 12, 12, 12, 12])  # voltage (V)
I = np.array([1.2, 1.8, 2.5, 3.0, 3.4, 3.8])  # current (A)
RPM = np.array([500, 900, 1300, 1600, 1800, 2000])  # speed (rpm)
T = np.array([0.8, 1.1, 1.4, 1.6, 1.7, 1.8])  # torque (Nm)
```

### 2. 1- Electrical Analysis

**Q1** Using NumPy, compute the **electrical power input**: $P_{in} = V \times I$

**Q2** Calculate the **mean**, **max**, and **min** values of the current.

**Q3** Determine the time instants where the **current exceeds 3 A**.

### 2. 2- Mechanical Analysis

**Q4** Convert the motor speed from **RPM to rad/s**: $\omega = \frac{2\pi \cdot RPM}{60}$

**Q5** Compute the **mechanical output power**: $P_{out} = T \times \omega$

**Q6** Find the time instant where the **maximum torque** occurs.

### 2. 3- Efficiency & Performance

**Q7** Using NumPy, compute the **efficiency** of the motor: $\eta = \frac{P_{out}}{P_{in}}$

**Q8** Find the **average efficiency** of the motor.

**Q9** Detect the operating points where the **efficiency is greater than 80%**.

## 2. 4- Signal & Array Processing

**Q10** Normalize the speed vector RPM between **0 and 1**.

**Q11** Using NumPy indexing, extract:

- First 3 values of current
- Last 2 values of torque

**Q12** Compute the **numerical derivative of speed** with respect to time using NumPy.

## 2. 5- Engineering Decision

**Q13** Assume:

- Maximum allowed current = **3.5 A**
- Minimum required efficiency = **75%**

Using NumPy logical operations, determine:

- Safe operating points
- Unsafe operating points

**Q14** Count how many safe operating points exist.

**Q15 (Conceptual)** Explain **why vectorized NumPy operations are preferred** over loops in engineering simulations.