Lab 03: Machine Learning

Mathematical Theory of the DC Motor Modeling and Control Method

🎯 Lab Objectives

By the end of this lab, students will be able to:

- Model a **DC motor using experimental data**
- Use **multiple linear regression** (two input features)
- Interpret regression coefficients physically
- Predict motor speed under different operating conditions
- Prepare a model suitable for **control design**

1. Theoretical Part

1.1 DC Motor Static Modeling

In the considered operating range, the DC motor is assumed to operate in **steady state** and **linear conditions.**

The motor speed ω depends mainly on:

- the applied armature voltage V,
- the load torque T_L.

Thus, the motor is modeled as a **static multi-input system**:

$$\omega = f(V, T_L)$$

1.2 Linear Regression Model

1.2.1 Model Structure

The system is approximated by a **linear affine model**:

$$\boxed{\omega = \theta_0 + \theta_1 V + \theta_2 T_L}$$

Where:

- θ_0: constant term (friction and losses),
- θ_1: voltage-to-speed gain,
- θ_2: load torque influence (usually negative).

1.2.2 Vector–Matrix Formulation

For N experimental measurements:

$$\mathbf{y} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_N \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} 1 & V_1 & T_{L1} \\ 1 & V_2 & T_{L2} \\ \vdots & \vdots & \vdots \\ 1 & V_N & T_{LN} \end{bmatrix} \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix}$$

The model becomes:

$$\boxed{\mathbf{y} = \mathbf{X}\theta}$$

1.3 Least Squares Parameter Identification

1.3.1 Error Definition

The prediction error vector is:

$$\mathbf{e} = \mathbf{y} - \mathbf{X}\theta$$

1.3.2 Cost Function

The Least Squares method minimizes the quadratic cost:

$$J(\theta) = \|\mathbf{e}\|^2 = (\mathbf{y} - \mathbf{X}\theta)^T(\mathbf{y} - \mathbf{X}\theta)$$

1.3.3 Optimal Solution

Setting the gradient to zero:

$$\frac{\partial J}{\partial \theta} = 0$$

Gives the **normal equation**:

$$\mathbf{X}^T\mathbf{X}\hat{\theta} = \mathbf{X}^T\mathbf{y}$$

If $\mathbf{X}^T\mathbf{X}$ is invertible:

$$\boxed{\hat{\theta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}}$$

1.4 Model Validation: Mean Squared Error (MSE)

To evaluate model accuracy, the **Mean Squared Error** is used:

$$\boxed{\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\omega_i - \hat{\omega}_i)^2}$$

Where:

- ω_i: measured speed,
- $\hat{\omega}_i$: predicted speed.

Interpretation

- Small MSE → accurate model
- Large MSE → poor identification or missing nonlinearities

1.5 Proportional Speed Control

1.5.1 Control Objective

Force the motor speed ω to track a desired reference ω_{ref}.

1.5.2 Control Law

The proportional controller is defined as:

$$\boxed{V = K_p(\omega_{ref} - \omega)}$$

Where:

- $e = \omega_{ref} - \omega$: speed error,
- K_p: proportional gain.

1.6 Closed-Loop Mathematical Model

Substituting the control law into the motor model:

$$\omega = \theta_0 + \theta_1 K_p(\omega_{ref} - \omega) + \theta_2 T_L$$

Rearranging:

$$(1 + \theta_1 K_p)\omega = \theta_1 K_p \omega_{ref} + \theta_0 + \theta_2 T_L$$

Thus, the closed-loop steady-state speed is:

$$\boxed{\omega = \frac{\theta_1 K_p}{1 + \theta_1 K_p}\omega_{ref} + \frac{\theta_0 + \theta_2 T_L}{1 + \theta_1 K_p}}$$

2. Practical Part, Identification, Validation, and Classification of a DC Motor

Context

A DC motor is tested experimentally under different operating conditions.

The objectives of this exercise are to:

1. **Identify** a mathematical model of the motor speed
2. **Validate** the model using a train/test strategy
3. **Classify** the motor operating condition using K-Nearest Neighbors (KNN)
4. Apply the identified model in a **simple closed-loop speed control**

Given Experimental Data

The motor speed ω depends on:

- Applied voltage V (V)
- Load torque T_L (N·m)

Voltage (V)	Load Torque (N·m)	Speed (rad/s)
10	0.5	92
12	0.5	110
14	0.5	128
10	1.0	75
12	1.0	94
14	1.0	112
10	1.5	60
12	1.5	78
14	1.5	95

```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
```

Part A – Linear Regression Model Identification

1. Assume a linear model of the form:

$$\omega = \theta_0 + \theta_1 V + \theta_2 T_L$$

2. Construct the input matrix X and output vector y.
3. Identify the model parameters $\theta_0, \theta_1, \theta_2$ using linear regression.
4. Write the identified DC motor model.
5. Compute the **Mean Squared Error (MSE)** using all experimental data.
6. Plot the **measured speed vs predicted speed** and comment on the result.

Using the identified model, predict the motor speed for: $V = 37\ \mathrm{V}, \quad T_L = 1.0$ N·m

Draw three curves corresponding to:

- $T_L = 0.5$ N·m
- $T_L = 1.0$ N·m
- $T_L = 1.5$ N·m

On the same graph, indicate the predicted operating point:

- $V = 37$ V
- $T_L = 1.0$ N·m

Part B – Model Validation Using Train/Test Split

7. Explain the importance of separating data into **training** and **testing** sets.
8. Split the dataset into:
   - 70% training data
   - 30% testing data
9. Train the linear regression model using only the training set.
10. Predict the motor speed for the test set.
11. Compute the **test MSE** and compare it with the MSE obtained using all data.
12. Plot **true speed vs predicted speed** for the test set and comment on the model generalization.

Part C – Prediction and Closed-Loop Speed Control

14. A proportional controller is defined as: $V(k) = K_p(\omega_{ref} - \omega(k))$

where:

- $K_p = 0.05$
- $\omega_{ref} = 150\,\mathrm{rad/s}$

15. Simulate the closed-loop speed response of the motor.
16. Plot:
    - Motor speed vs time
    - Control voltage vs time
17. Comment on the stability and tracking performance.

Part D – Motor Condition Classification Using KNN

To monitor motor health, the operating condition is classified as:

- **Class 0** → Normal operation
- **Class 1** → Overloaded motor

Based on:

- Motor current I (A)
- Motor temperature T (°C)

Sample	I (A)	T (°C)	Class
1	4.2	55	0
2	4.6	58	0
3	5.8	72	1
4	6.0	75	1
5	4.9	60	0
6	6.1	78	1

18. Explain the principle of the **K-Nearest Neighbors (KNN)** algorithm.
19. For $K = 3$, classify the following motor condition:
    - $I = 5.3\,\mathrm{A}$
    - $T = 65^\circ\mathrm{C}$
20. Plot the classification results and indicate the position of the new sample.
21. Discuss the effect of choosing a **small** or **large** value of K.

3. DC Motor Identification Using NumPy Lib (Homework)

Given the experimental measurements of a DC motor (Voltage V, Load Torque T_L, and Speed ω):

1. Assume the following linear static model:

$$\omega = \theta_0 + \theta_1 V + \theta_2 T_L$$

2. Rewrite the model in vector–matrix form:

$$Y = X\theta$$

3. Using **NumPy only (without using sklearn)**:
    - Construct the regression matrix X.
    - Construct the output vector Y.
    - Compute the parameter vector θ using the Least Squares formula:

$$\theta = (X^T X)^{-1} X^T Y$$

4. Print the identified model in the form:

$$\omega = \theta_0 + \theta_1 V + \theta_2 T_L$$

5. Compute the Mean Squared Error (MSE) manually using NumPy.
6. Compare the obtained results with those obtained using the `LinearRegression` function from sklearn.