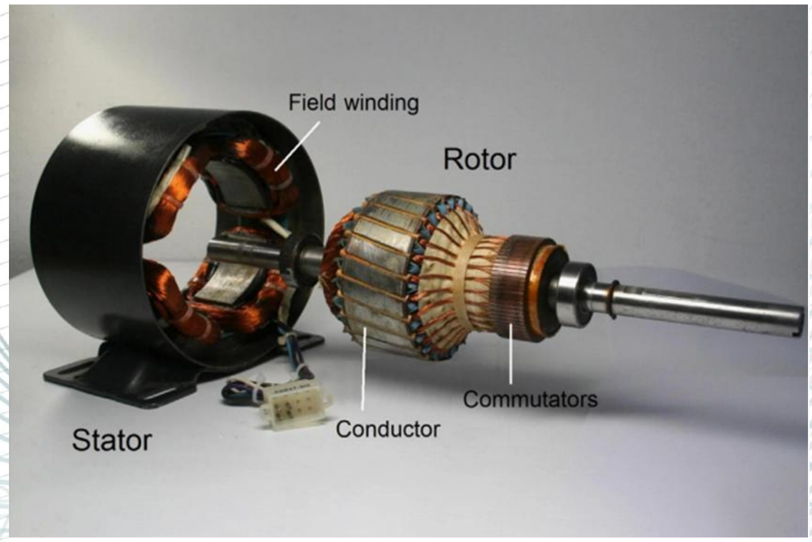
DC motor modeling

Students Names

Motasem Faous	0225254
Zaid Al-Qaisi	0227640
Mohammed Fino	0224640

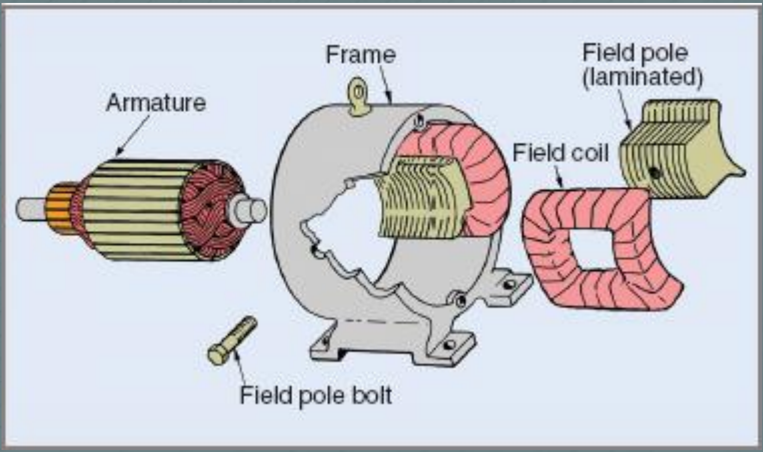
Electrical Machine

Prof:
Mohammed Masadah



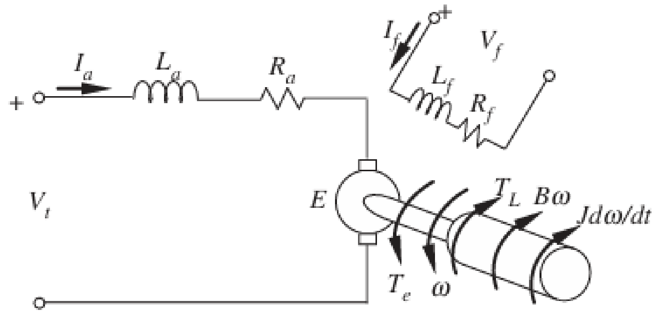
INTRODUCTION

Mechatronic systems recently became the most essential parts of automatic applications in several industries such as robotics, defense, automotive, etc. An electromechanical actuation unit which involves a DC motor, transmission and electronic controller is a kind of mechatronic system.
This project aims to develop a graphical user interface (GUI) for real-time analysis of the dynamic performance of DC motors. By integrating a robust mathematical model of DC motor dynamics, the system allows users to simulate and observe motor behavior under various operating conditions. The GUI is designed to be user-friendly, enabling parameter input, real-time visualizatio of key performance metrics such as speed, current, and torque, and exporting simulation data for further analysis.



Mathematical Model

Schematics and Equations:



Electrical part:

$$V - IR_a - L_a\dot{I} = e_{emf}$$
$$e_{emf} = K_e\,\omega$$

Mechanical part:

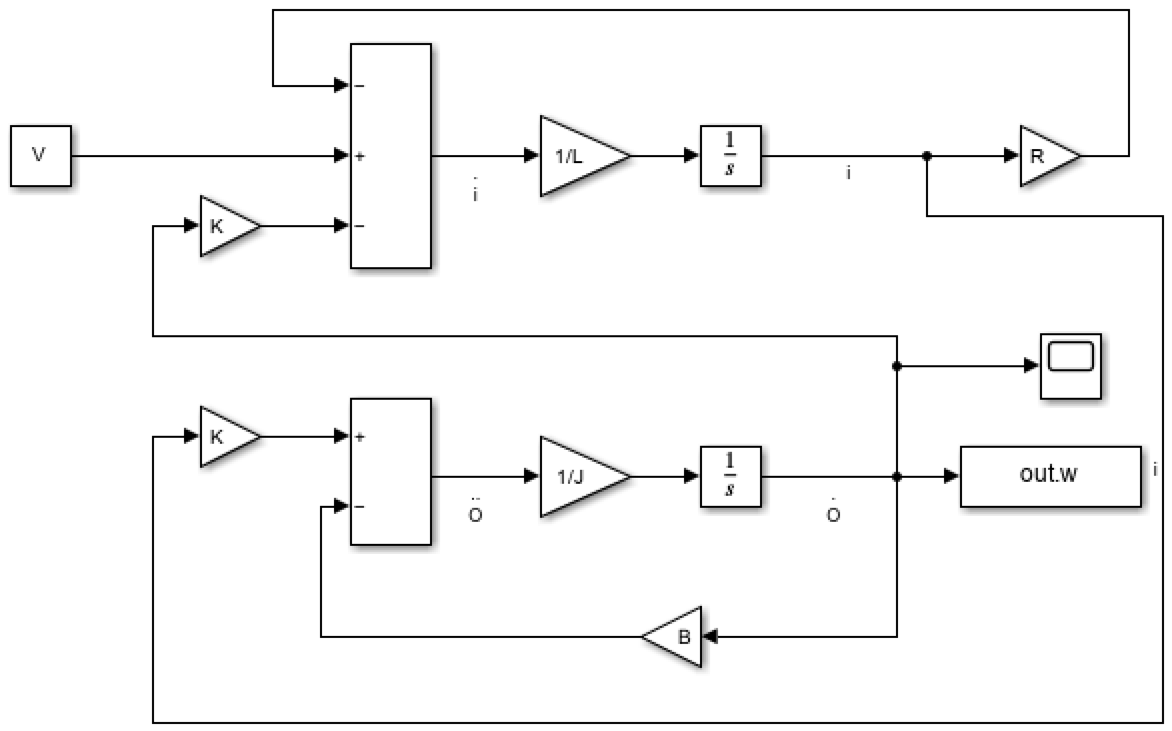
$$\sum T = J\ddot{\theta}$$
$$T - b\dot{\theta} = J\,\ddot{\theta}$$
$$T = K_t i$$
$$-i\,R = K_e\theta \dot{+} L_a\dot{I} - V$$
$$i = \frac{K_e\dot{\theta} + L_a\dot{I} - V}{-R}$$

The diagram illustrates the dynamics of a DC motor, separating the electrical and mechanical components. In the electrical part, the applied voltage V drives the armature circuit, which consists of the armature resistance R_a, inductance L_a, and the back EMF (e_{emf}=$K_e\,\omega$). The back EMF, generated by the rotor's motion, opposes the applied voltage and is proportional to the angular velocity ω. This interaction governs the current i_a, which flows through the circuit. In the mechanical part, the electromagnetic torque $T = K_t i$ drives the rotor, overcoming the load torque T_L, rotational friction b ω and the rotor's inertia J$\frac{d\omega}{dt}$ The coupling between the electrical and mechanical domains is achieved through the constants K_t and K_e , representing the motor's ability to convert electrical energy into mechanical motion. Together, these equations model the complete electromechanical behavior of the DC motor, enabling it to respond dynamically to variations in input voltage, load torque, and speed

In the model, the torque constant (K_t) and back EMF constant (K_e) connect the electrical and mechanical domains. K_t ($\frac{Nm}{A}$)relates the armature current to the generated torque, while K_e ($\frac{V\,S}{rad}$) links the rotor's angular velocity to the induced back EMF. These constants ensure energy conservation and seamless interaction, allowing the electrical input to drive mechanical motion and the mechanical feedback to influence electrical behavior.

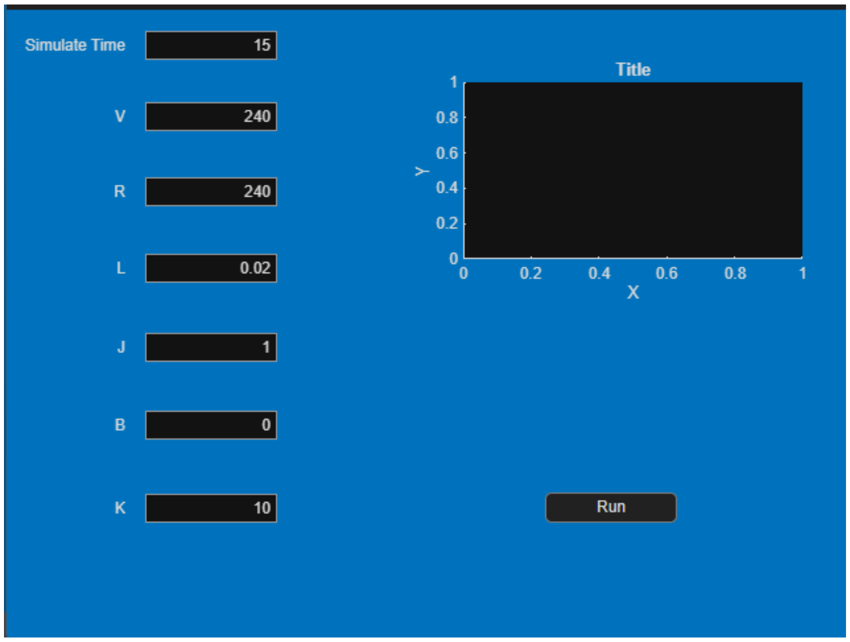
MATLAB/SIMULINK Block Diagram

The Simulink model shown below represents the system dynamics based on the derived equations. It illustrates the input-output relationships, including the response of the motor Speed under the given conditions.



We have included a dedicated block to extract the angular velocity (ω\omegaω) from the system. This value is dynamically sent to the GUI, where it is displayed on a graph to provide a visual representation of the motor's performance in real-time. The GUI also allows for interactive adjustment of key parameters to observe their effect on the motor's behavior

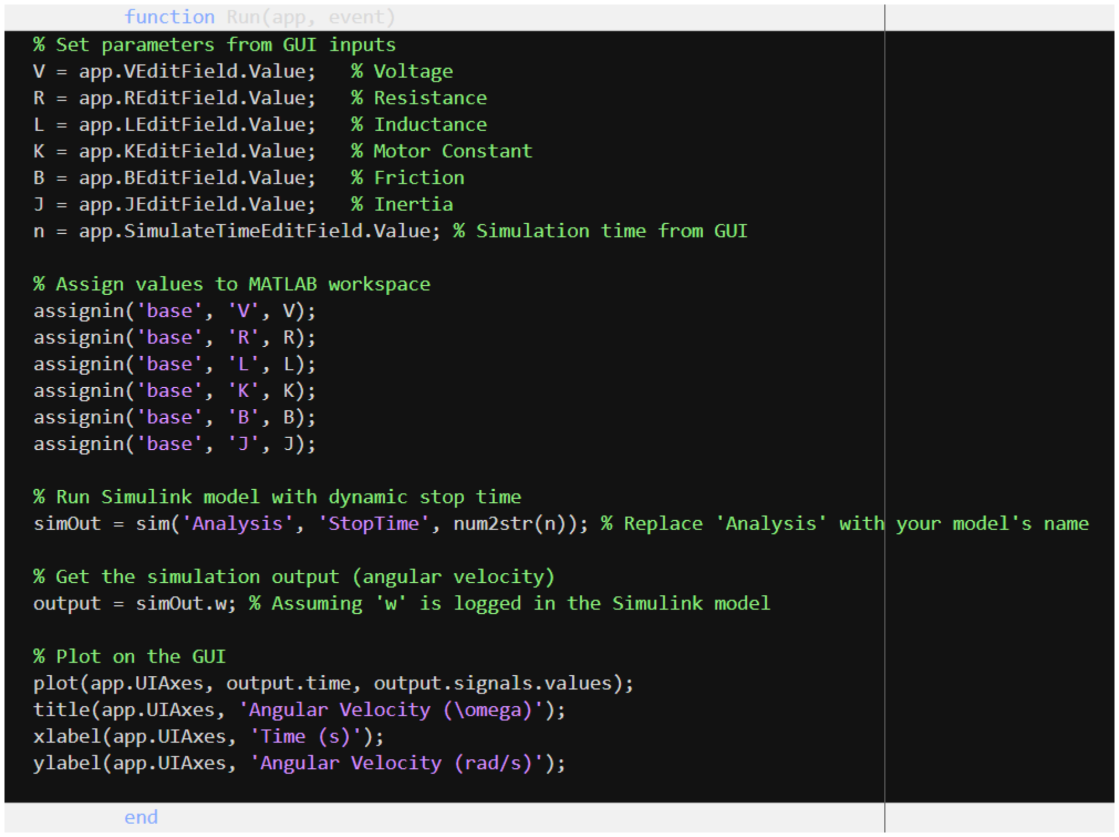
Gui and Code



Description of the GUI Interface:

This GUI interface is designed for simulating and analyzing the behavior of a DC motor. It includes the following **key elements:**

1. **Graphical Display (Axes):**
   - Displays the angular velocity (ω) of the motor over time. The simulation results are dynamically plotted here to provide a visual representation of the motor's performance.
2. **Adjustable Parameters:**
   - V (Voltage): The input voltage applied to the motor.
   - R (Resistance): The resistance in the motor's electrical circuit.
   - L (Inductance): The inductance of the motor, representing the effect of the coil on the current flow.
   - J (Inertia): The moment of inertia of the motor's rotor, which affects how quickly the motor accelerates or decelerates.
   - B (Friction): The friction coefficient representing the resistance to motion due to mechanical components.
   - K (Motor Constant): A proportionality constant relating to the motor's torque and speed.
3. **Simulation Duration:**
   - You can specify the total simulation time, allowing you to observe the motor's behavior over a custom period.



```
        function Run(app, event)
    % Set parameters from GUI inputs
    V = app.VEditField.Value;   % Voltage
    R = app.REditField.Value;   % Resistance
    L = app.LEditField.Value;   % Inductance
    K = app.KEditField.Value;   % Motor Constant
    B = app.BEditField.Value;   % Friction
    J = app.JEditField.Value;   % Inertia
    n = app.SimulateTimeEditField.Value; % Simulation time from GUI

    % Assign values to MATLAB workspace
    assignin('base', 'V', V);
    assignin('base', 'R', R);
    assignin('base', 'L', L);
    assignin('base', 'K', K);
    assignin('base', 'B', B);
    assignin('base', 'J', J);

    % Run Simulink model with dynamic stop time
    simOut = sim('Analysis', 'StopTime', num2str(n)); % Replace 'Analysis' with your model's name

    % Get the simulation output (angular velocity)
    output = simOut.w; % Assuming 'w' is logged in the Simulink model

    % Plot on the GUI
    plot(app.UIAxes, output.time, output.signals.values);
    title(app.UIAxes, 'Angular Velocity (\omega)');
    xlabel(app.UIAxes, 'Time (s)');
    ylabel(app.UIAxes, 'Angular Velocity (rad/s)');

        end
```

Overview:

This code defines the Run function that is triggered when the user interacts with the GUI, which collects inputs and runs the simulation for a DC motor system in Simulink and displays the results on a graph.

Code Breakdown:

1. **Get Parameters from GUI Inputs:**
   - V, R, L, K, B, J, n: The values for are collected from the user input fields in the GUI.
2. **Assign Values to MATLAB Workspace:**
   - The parameters are assigned to the MATLAB base workspace using assigning. This makes them accessible to the Simulink model for simulation.
3. **Run the Simulink Model:**
   - The sim function is used to run the Simulink model (named Analysis). The simulation time is dynamically set using the value of n from the GUI. The Stop Time parameter is specified to control how long the simulation runs.
4. **Get Simulation Output:**
   - The simulation output (ω) is retrieved from the simulation results, which correspond to the angular velocity of the motor. It assumes that (ω)is logged in the Simulink model.
5. **Plot the Output on the GUI:**
   - The plot function is used to plot the simulation results (angular velocity vs. time) on the UIAxes of the GUI.
   - Labels and a title are added to the plot for clarity, showing time on the x-axis and angular velocity on the y-axis.

Conclusion:

In this project, a GUI-based application was developed to simulate the behavior of a DC motor, incorporating key motor parameters such as voltage, resistance, inductance, motor constant, friction, and inertia. The application allows users to dynamically adjust these parameters and visualize the resulting angular velocity over time through an interactive graph.
By integrating MATLAB and Simulink, the system enables seamless communication between the GUI and the motor model. The user-friendly interface provides a straightforward way to experiment with different motor configurations, giving valuable insights into the motor's performance under various conditions.
Through this project, we have demonstrated the ability to model and simulate the behavior of a DC motor using MATLAB and Simulink, while offering a flexible and intuitive platform for analysis. Future enhancements could include adding more advanced motor control features, improving the real-time simulation capabilities, and optimizing the user interface for better experience.
Overall, this project showcases the powerful combination of simulation and GUI in understanding complex electrical systems, providing a useful tool for both educational and engineering purposes.