Implementation of a Basic Elevator Drive System

1st Erdem Canaz
Electrical and Electronics Engineering Dept.
Middle East Technical University
Ankara, Turkey
erdemcanaz@gmail.com
2374676

2nd Onur Toprak
Electrical and Electronics Engineering Dept.
Middle East Technical University
Ankara, Turkey
onur.toprak@metu.edu.tr
2375897

***Abstract*—This paper presents the design and simulation of a control system for an elevator drive system driven by a permanent magnet excited brushed DC machine. The system is designed considering various load conditions and evaluated using a python based simulation. Key performance metrics such as speed-time characteristics, acceleration, cabin speed and reference speed tracking performances are analyzed. Current and Speed controller with a unipolar DC-DC full-bridge converter is fully developed.**

***Index Terms*—Elevator drive system, brushed DC machine, control system, unipolar full bridge DC-DC converter.**

I. Introduction

DRIVE systems are crucial for efficient and reliable vertical transportation in modern buildings, directly impacting user convenience and safety. This paper focuses on designing and simulating a control system for an elevator powered by a permanent magnet excited brushed DC machine.

The main goal is to create a robust controller that ensures optimal speed, acceleration, and smooth rides, even under varying load conditions and sudden load changes. This research includes a comprehensive analysis of both electrical and mechanical aspects, along with fine-tuning the PI controllers.

The control system will use PI regulators for both outer speed and inner current controls. These will be validated through simulations conducted in Python under various conditions.

The final model will integrate a full-bridge DC-DC converter, a control algorithm, and PI speed and current controllers. This high-fidelity simulation is intended to lay the groundwork for future experimental validation and real-world implementation.

II. System Description

The considered elevator system consists of a permanent magnet excited brushed DC machine, counterweight, cabin, and pulley system as shown in Fig. 1. The system characteristics are given as follows:

- gravitational acceleration (g): 9.80 m/s²
- Cabin weight (M_c): 1000 kg
- Maximum load ($M_{\text{load-max}}$): 500 kg
- Counterweight (M_{cw}): 1250 kg
- Elevation: 10 m
- Maximum speed ($v_{\text{cabin-max}}$): 2 m/s
- Acceleration/deceleration ($|a_{\text{cabin-max}}|$): 1 m/s²
- Gear box ratio ($\alpha_g = \omega_{drum}/\omega_{em}$): 1/5
- Drum diameter (R_{drum}): 0.4 m
- DC machine inertia (J_{em}): 0.2 kg·m²
- Viscous friction coefficient (b_{em}): 0.05 Nms/rad
- Rated armature voltage (V_a): 200 V
- Rated armature current (I_a): 75 A
- Armature resistance (R_a): 0.1 Ω
- Armature inductance (L_a): 2 mH
- Torque constant (K_t): 1.75 Nm/A
- Back EMF constant (K_b): 1.75 Vs/rad

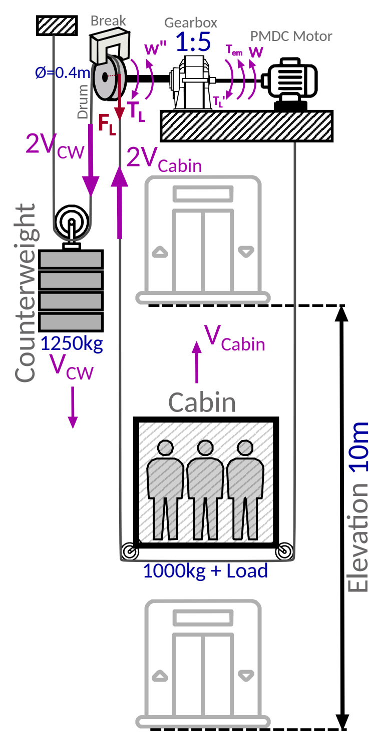

Fig. 1. Block Diagram of the Elevator System

III. Generic Equations

In this section, the generically applicable equations will be derived. The primary objective is to simplify the electro-mechanical system by referring all loads, frictions, and other forces to the motor side.

A. Equivalent Inertia Calculation

Total Kinetic Energy of the system can be written as

$$K.E. = \frac{1}{2} \cdot J_{eq} \cdot \omega^2 = \frac{1}{2}(\cdot J_{em} \cdot \omega^2 + M_{cw} \cdot V_{cw}^2 + (M_c + M_{load}) \cdot V_c^2) \tag{1}$$

The velocities of the cabin and counterweight can be expressed in terms of ω.

$$V_c = V_{cw} = \frac{1}{4}\alpha_g \cdot R_{drum} \cdot \omega \tag{2}$$

$$V_c = V_{cw} = (0.02) \cdot \omega \tag{3}$$

Then the total kinetic energy in terms of ω is;

$$K.E. = \frac{1}{2} \cdot J_{eq} \cdot \omega^2 = \frac{1}{2} \cdot (J_{em} + (0.02)^2 \cdot (M_{cw} + M_c + M_{load})) \cdot \omega^2 \tag{4}$$

$$J_{eq} = J_{em} + (0.02)^2 \cdot (M_{cw} + M_c + M_{load}) \tag{5}$$

It can be assumed that the average mass of the load is half of the rated load. Therefore, the equivalent inertia j_{eq} is approximately calculated as follows;

$$J_{eq} = 1.2\ kg^2 \tag{6}$$

B. $T_L^{'}$ Calculation

The $T_L^{'}$ is defined on Fig. 1. It can be calculated by the following formula;

$$T_L^{'} = \frac{\alpha_g \cdot R_{drum} \cdot g}{4}(M_c + M_{load} - M_{cw}) \tag{7}$$

$$T_L^{'} = (0.196)(M_c + M_{load} - M_{cw}) \tag{8}$$

IV. Inner and Outer Controllers Tuning

In this section, both the inner (current) controller and the outer (speed) controller will be designed.

A. Inner PI Controller (Current)

The inner PI controller is responsible for maintaining the armature current at the desired value by adjusting the armature voltage. This controller's rise time is at least five times faster than that of the outer controller, as the system primarily comprises electrical components. The differential equation governing the electrical system is given by:

$$V_a = L_a \frac{d}{dt} I_a + R_a I_a + K_b \omega \tag{9}$$

After applying Laplace transform;

$$V_a = (sL_a + R_a) I_a + K_b \omega \tag{10}$$

When small perturbations method is used on the Eq. 10, the voltage-to-current transfer function can be found as;

$$\frac{\Delta I_a}{\Delta V_a} = \frac{1}{(sL_a + R_a)} \tag{11}$$

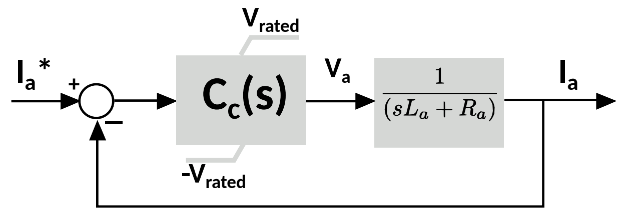

Fig. 2. Block Diagram of the Inner Loop

The proposed controller diagram can be found in Fig. 2. where the PI output is limited to rated PMDC armature voltage.

We want the closed loop transfer function to be of the form;

$$T_{closed}(s) = \frac{\alpha}{\alpha + s} \tag{12}$$

so that the rise time can be expressed as;

$$t_{rise} = \frac{\ln(9)}{\alpha} \tag{13}$$

In other words, below equation should hold

$$C_c(s) \cdot \frac{1}{sL_a + R_a} = \frac{\alpha_{in}}{s}$$

By setting K_p and K_i parameters of $C_c(s)$ as;

$$Kp, in = \alpha_{in} L_a$$

$$Ki, in = \alpha_{in} R_a$$

Such a closed-loop transfer function can be achieved. For $t_{rise,in} = 10ms$, $\alpha_{in} = 220 rad/s$ the controller parameters can be found as;

$$Kp, in = 0.44$$

$$Ki, in = 22$$

The simulated current tracking performance is shown in Fig 3. where the rotor is halted. The results shows that the parameter tuning for the inner loop is fine.

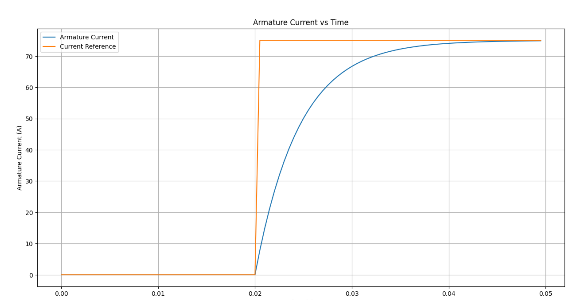

Fig. 3. Inner Loop Tracking Performance Simulation Results

B. Outer PI Controller (Speed)

The outer PI controller is responsible for maintaining the rotor speed at the desired value by adjusting the armature current (i.e., PMDC torque). Given its slower rise time compared to the inner controller, the IMC method can be applied, where the inner current control block functions as a unity block, as illustrated in Fig. 4. One should note that the output of the outer controller is limited by the rated PMDC armature current.

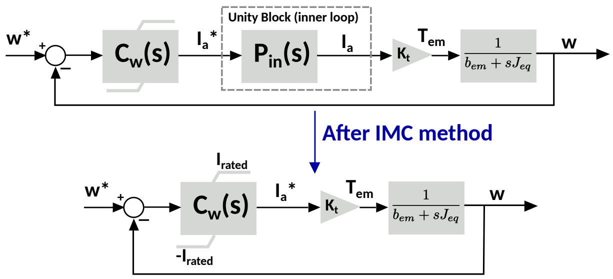

Fig. 4. Block Diagram of the Outer Loop with IMC Method Applied

The differential equation governing the speed-torque relation can be given as;

$$T_{em} = T_L^{'} + b_{em} \cdot \omega + J_{eq}\frac{d}{dt}\omega$$

After applying the laplace transform;

$$T_{em} = T_L^{'} + (b_{em} + sJ_{eq}) \cdot \omega$$

After applying the small perturbations method, the torque-to-speed transfer function can be found as;

$$\frac{\Delta\omega}{\Delta T_{em}} = \frac{1}{(b_{em} + sJ_{eq})}$$

Similar to inner loop control, below equation should be satisfied;

$$C_w(s) \cdot \frac{K_t}{b_{em} + sJ_{eq}} = \frac{\alpha_{out}}{s}$$

By setting K_p and K_i parameters of $C_w(s)$ as;

$$Kp,out = \frac{\alpha_{out}J_{eq}}{K_t}$$

$$Ki,out = \frac{\alpha_{out}b_{em}}{K_t}$$

The condition is satisfied. Then for $t_{rise,out} = 75ms$, $\alpha_{out} = 29.3rad/s$; the outer controller parameters can be found as;

$$Kp,out = 20.1$$

$$Ki,out = 0.83$$

The simulated speed tracking performance is shown in Fig. 5. where the armature current is not limited. The results show that the parameter tuning for the inner loop is fine. However one should note that after applying the current limits to protect the motor and the supply, the tracking performance is limited as shown in Fig. 6.

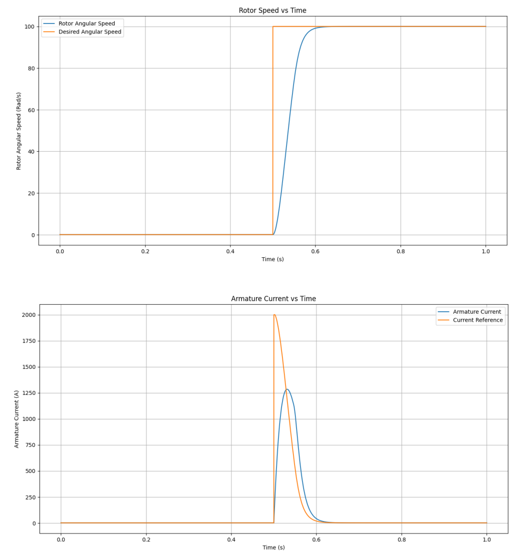

Fig. 5. Outer Loop Tracking Performance without Current Limit

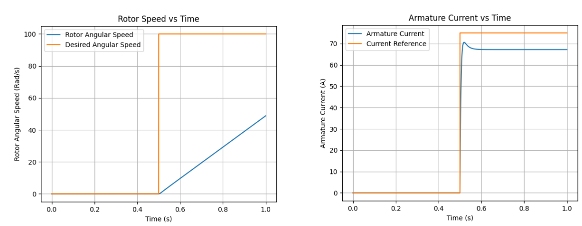

Fig. 6. Outer Loop Tracking Performance with Rated Armature Current Limit

V. DC-DC UNIPOLAR FULL BRIDGE CONVERTER

To apply varying armature voltages to control the armature current, an active DC-DC converter is required. In this project, a full-bridge DC-DC converter has been chosen for this purpose, with control implemented using unipolar switching as shown in Fig. 7.

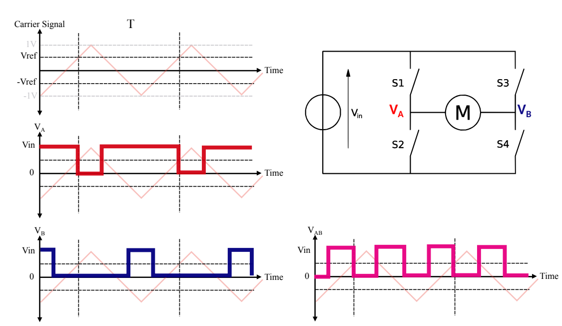

Fig. 7. H-bridge Unipolar Switching

The model is implemented in Python, focusing on the analytical aspects and ignoring losses, switching times, and voltage drops. The output voltage of the converter is expressed in terms of the input voltage and the current switching state. The HBridge_Unipolar class models a unipolar H-Bridge DC-DC converter used to control armature voltage. The class initializes with the input DC voltage, carrier signal magnitude, frequency, and reference voltage. It generates a triangular carrier signal that alternates between negative and positive values over time. This carrier signal is crucial for comparing with the reference voltage. To achieve the desired output voltage, the reference voltage (v_{ref}) is updated dynamically by the inner controller, but only when the carrier signal is below a specific threshold. This timing ensures accuracy and prevents errors. The output voltage (V_{ab}) is then calculated by comparing the carrier signal with the DC reference voltage. Based on this comparison, the appropriate voltages are assigned to Van and Vbn, and their difference gives the final output voltage. The parameters of the carrier wave are chosen as

- $f_{carrier} = 7500Hz$
- $|V_{carrier}| = 1V$

VI. Main Control Algorithm

The main control algorithm generates a speed reference and applies brakes at the appropriate times. It assumes that the elevator's speed tracking performance is nearly perfect. For example, consider the elevator moving upward. First, the algorithm releases the brakes and then gradually increases the elevator's reference speed to ensure a smooth and comfortable acceleration. Once the desired speed is reached, the speed reference is kept constant. As the elevator approaches the desired location, the algorithm begins to decrease the reference speed with a similar deceleration, ensuring a smooth stop. The controller ensures that the elevator speed is almost zero (i.e., zero crossing) at the desired location, minimizing stress on the mechanical brake

VII. Simulation Results

In this section, four different cases concerning movement direction and load will be presented.

A. Full Load Going Up

The results are given in Fig. 8

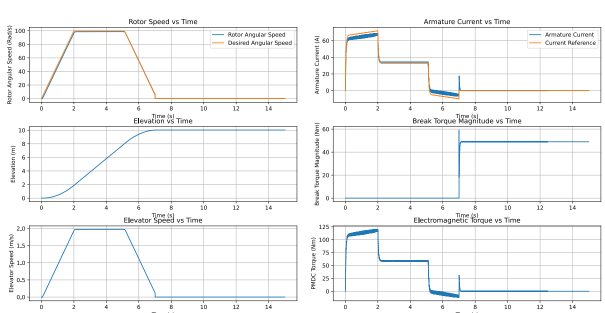

Fig. 8. Full Load Going Up Simulation Results

B. No Load Going Up

The results are given in Fig. 9

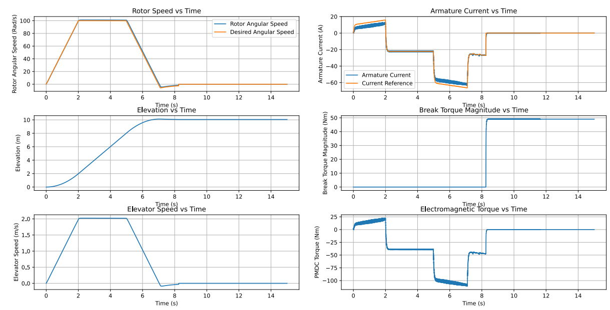

Fig. 9. No Load Going Up Simulation Results

C. Full Load Going Down

The results are given in Fig. 10

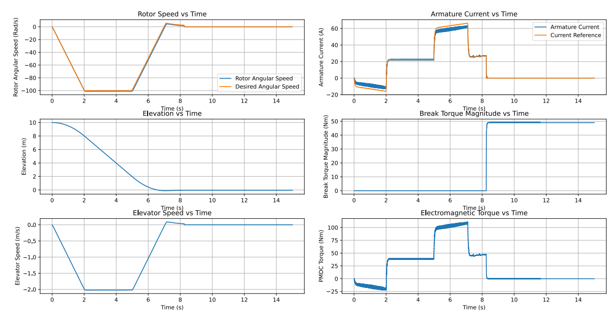

Fig. 10. Full Load Going Down Simulation Results

D. No Load Going Down

The results are given in Fig. 11

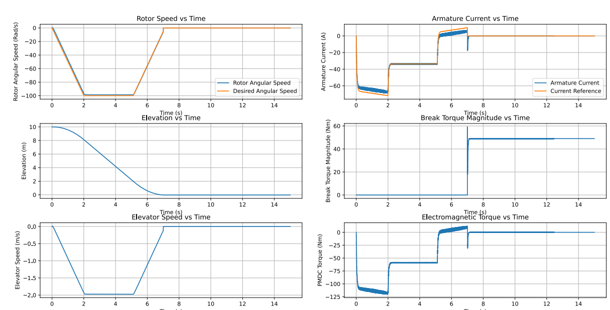

Fig. 11. No Load Going Down Simulation Results

VIII. Source Code

The source code can be found at the following GitHub repository: EE462-PROJECT

IX. Conclusion

In this project, a comprehensive control system for an elevator drive driven by a permanent magnet excited brushed DC machine was designed and simulated. The system's performance was analyzed under various load conditions using a Python-based simulation. Key aspects such as speed-time characteristics, acceleration, and reference speed tracking were meticulously examined.

The control system employs both inner and outer PI controllers. The inner PI controller maintains the armature current by adjusting the armature voltage, ensuring a fast response due to the predominantly electrical nature of the system. The outer PI controller manages the rotor speed by modulating the armature current, employing an IMC method to achieve a slower, yet smooth, response.

A unipolar full-bridge DC-DC converter was implemented to control the varying armature voltages. The converter's operation was modeled analytically, focusing on the essential dynamics while ignoring losses and switching times.

The main control algorithm was designed to provide a smooth and comfortable ride by generating speed references and applying brakes at appropriate times. The algorithm gradually increases the speed reference, maintains a constant speed, and then decelerates smoothly as the elevator approaches its destination, ensuring minimal stress on the mechanical brakes.

Simulation results demonstrated the system's robustness and efficiency under different load conditions and movement directions. These simulations validate the effectiveness of the designed control system and the tuning of the PI controllers.