IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Journal Européen des Systèmes Automatisés
Vol. 59, No. 4, April, 2026, pp. 965-979
Journal homepage: http://iieta.org/journals/jesa

Real-Time Hardware-in-the-Loop Sensorless FOC of BLDC Motors in Electric Vehicles Using Slime Mould Optimization for Online PI Tuning and Back-EMF Estimation

Zakaria Alili[1*], Ismail Ghadbane[1], Riad Bouzidi[1], Ahmed Marouane Ghodbane[2], Abdelmoumen Ghilani[3], Djallal Eddine Zabia[4]

[1] LGE Laboratory, University of Msila, Msila 28000, Algeria
[2] LGEB Laboratory, University of Biskra, Biskra 07000, Algeria
[3] LMSE Laboratory, University of Biskra, Biskra 07000, Algeria
[4] LI3CUB Laboratory, University of Biskra, Biskra 07000, Algeria

Corresponding Author Email: zakaria.alili@univ-msila.dz

https://doi.org/10.18280/jesa.590409

Received: 11 February 2026
Revised: 2 April 2026
Accepted: 9 April 2026
Available online: 30 April 2026

Keywords:
Brushless DC, Slime Mould Optimization, sensorless, field-oriented control, back-Electro-Motive-Force, electric vehicle

ABSTRACT

This paper presents a sensorless field-oriented control (FOC) strategy for Brushless DC (BLDC) motors in electric vehicle (EV) applications, enhanced through online tuning of Proportional Integral (PI) controllers using the Slime Mould Optimization (SMO) algorithm. To remove the need for mechanical sensors and reduce hardware complexity, a back-Electro-Motive-Force (EMF) observer is designed to estimate rotor position and speed based on measured stator currents and voltages. The performance of the speed control loop is further improved through real-time optimization of PI gains, where SMO continuously adjusts controller parameters according to instantaneous speed error. A full electric-vehicle load model—including aerodynamic drag, rolling resistance, slope force, and inertial dynamics—is integrated to evaluate system behavior under realistic driving conditions. The proposed sensorless FOC-SMO approach is validated through MATLAB/Simulink simulations and Hardware-in-the-Loop (HIL) testing on a dual dSPACE DS1104 platform. Results demonstrate fast dynamic response, robust tracking of variable speed profiles, improved disturbance rejection, and enhanced adaptability under nonlinear load variations. The method achieves accurate back-EMF estimation and stable operation in both motoring and regenerative braking modes, confirming its suitability for real-time EV propulsion systems.

1. INTRODUCTION

Electric vehicles (EVs) are using high-performance electrical drives more and more. These drives are efficient, reliable, and can control torque very accurately in a wide range of operating conditions [1]. Brushless DC (BLDC) motors are now very popular because they have a high power density, a wide speed range, and need less maintenance [2]. To control the speed and torque of BLDC motors well, you usually need to know exactly where the rotor is. This information is often obtained from Hall sensors or encoders [3]. But mechanical sensors add to the cost, make the system less reliable, and make installation more difficult. These problems can be avoided by using sensorless control strategies.

Field-oriented control (FOC) is a strong method for BLDC motor drives because it lets you quickly change the speed and torque of the motor without affecting the flux [4]. Additionally, the mathematical model of vehicle dynamics employed in this study was derived from the framework delineated in the study [5]. When there are no mechanical sensors, FOC has to use strong estimation methods, like back-Electro-Motive-Force (EMF) observers, to figure out the rotor's position and speed from electrical measurements, as shown in the study [6]. These methods make hardware less complicated, but they need Proportional Integral (PI) regulators that are well-tuned in both the current and speed loops to keep things stable and respond quickly. In electric vehicle (EV) applications, the load conditions are often very nonlinear and change over time, which makes it hard for traditional fixed-gain PI controllers to work well.

This work proposes a sensorless FOC drive improved by online parameter adaptation using the Slime Mould Optimization (SMO) algorithm, following the methodology in [7], to solve these problems. SMO changes the controller gains in real time based on speed error, which makes the system much more robust against load disturbances, back-EMF nonlinearity, and changes in vehicle dynamics [8]. To make driving conditions more realistic, a full EV mechanical model that includes aerodynamic drag, rolling resistance, slope resistance, and inertial forces is used. The suggested control algorithm is tested for real-time feasibility and safe experimental verification using dSPACE DS1104 platforms in simulation and Hardware-in-the-Loop (HIL) testing [9].

The primary contributions of this paper are encapsulated as follows:

Creating a sensorless back-EMF observer to get the rotor's

position and speed right.

Integration of a real-time SMO-based PI tuning strategy for better speed tracking and getting rid of disturbances.

Building a full EV load model for dynamic testing that is realistic.

HIL implementation that checks the control system in situations that are almost like real driving [10].

The results show that the suggested FOC-SMO method works well in EV applications by giving high tracking accuracy, a quick response to changes, and stability across a wide range of operating conditions.

1.1 System modeling

In a directly driven EV, the BLDC is mechanically linked to the Car wheels. The stator windings are linked to the inverter by a back-to-back voltage source converter interconnected through a DC link bus, as shown in Figure 1.

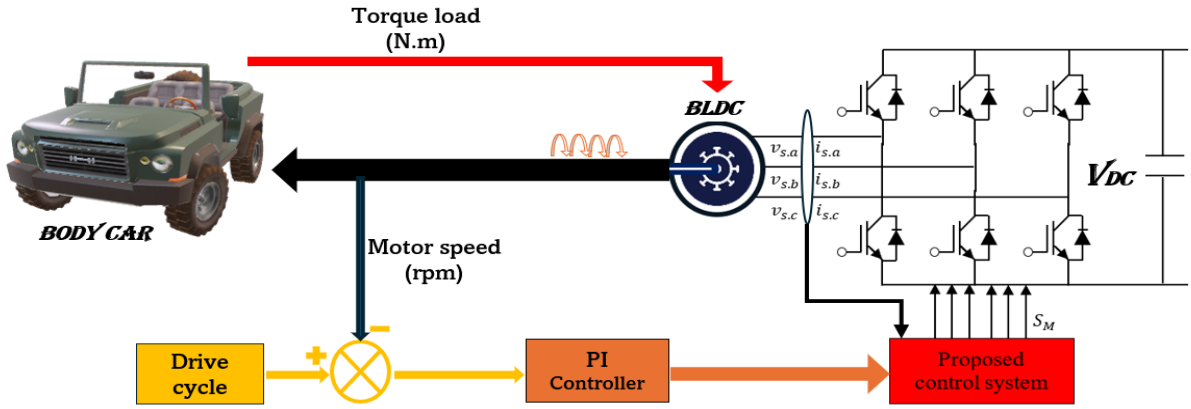


Figure 1. Configuration of an electric vehicle (EV) system with a Brushless DC (BLDC) motor

1.1.1 Brushless DC model

Compared to machines that are sinusoidally controlled, the BLDC motor is generally modeled in its natural three-phase form. This method is a way to model the behavior of a motor with typical six-step commutation, typically motivated by trapezoidal back-EMF waveforms and logic-based switching. The BLDC motor is made up of a three-phase stator and a permanent magnet rotor. Position sensors or sensorless algorithms are commonly used to implement the control scheme to calculate the correct sequence of commutation. In this part, the dynamic model in the natural frame aka the frame of the abc is formed.

- Electrical Model in the abc Frame

The electrical dynamics of each phase winding are described by [11]:

$$\begin{aligned} V_a &= \mathrm{R}\,.\,i_a + L\,.\frac{di_a}{dt} + e_a \\ V_b &= \mathrm{R}\,.\,i_b + L\,.\frac{di_b}{dt} + e_b \\ V_c &= \mathrm{R}\,.\,i_c + L\,.\frac{di_c}{dt} + e_c \end{aligned} \tag{1}$$

where,

- V_a, V_b, V_c are the phase voltages.
- i_a, i_b, i_c are the phase currents.
- R is the stator phase resistance.
- L is the self-inductance (assumed equal for all phases, and mutual inductance neglected).
- e_a, e_b, e_c are the back-EMF voltages in each phase, which are a function of rotor position.

The back-EMF of each phase is assumed to be a trapezoidal function of rotor electrical position θ_e, and its amplitude is proportional to the rotor speed ω_m:

$$e_k = K_e.\, f_k(\theta_e).\, \omega_m \text{ for } k \in \{a\,,b\,,c\} \tag{2}$$

where,

- k_e is the back-EMF constant.
- $f_k(\theta_e)$ is the normalized trapezoidal waveform for each phase (ranging from–1 to +1).
- ω_m is the mechanical angular speed of the rotor. These waveforms f_a, f_b, f_c are typically shifted by 120° from each other.

- Mechanical Dynamics

The mechanical dynamics of the BLDC motor are described by:

$$B\omega_m + J\,\frac{d\omega_m}{dt} = T_e\, - T_L \tag{3}$$

where,

- J is the total inertia of the rotor and load.
- B is the viscous friction coefficient.
- ω_m is the mechanical angular speed (rad/s).
- T_L is the external load torque (e.g., from vehicle body dynamics).
- T_e is the electromagnetic torque generated by the motor.

- Electromagnetic Torque

The electromagnetic torque generated by a BLDC motor with trapezoidal back EMF is given by [12]:

$$T_e = \frac{1}{\omega}(e_a\,.\,i_a + e_b\,.\,i_b + e_c\,.\,i_c) \tag{4}$$

This form highlights the interaction between the back-EMF and phase currents in generating torque.

1.1.2 Electric vehicle load dynamics

In EV applications, the mechanical load dynamics must be carefully defined in order to accurately model and assess the behavior of the motor control system. The motor must overcome a number of opposing resistive forces acting on the

body of the vehicle, as well as the inertial force needed to speed up. The total mechanical load torque that is sent to the motor shaft is determined by the cumulative effect of these longitudinal forces, which is what [13] says. This is what determines the exact amount of tractive effort needed to move the vehicle, as in Figure 2.

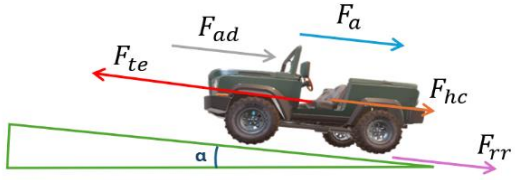


Figure 2. Different forces acting on the car body

- Aerodynamic Drag Force

The aerodynamic drag force Fad is the resistance exerted by the air against the moving vehicle and is given by [14]:

$$F_{ad} = \frac{1}{2}\rho C_d A V^2 \tag{5}$$

where,

- ρ is the air density (kg/m^3).
- C_d is the aerodynamic drag coefficient.
- A is the frontal area of the vehicle (m^2).
- V is the vehicle speed (m/s).

- Rolling Resistance Force

The rolling resistance force F_{rr} results from the deformation of tires at the road contact and is defined as [15]:

$$F_{rr} = K_r mg \tag{6}$$

- K_r is the rolling resistance coefficient.
- m is the total vehicle mass (kg).
- g is the gravitational acceleration (9.81m/s^2).

- Hill-Climbing Resistance Force

The resistance due to slope (uphill or downhill) is expressed by [16]:

$$F_{hc} = \pm mg \sin\alpha \tag{7}$$

where, α is the road slope angle in radians.

- Acceleration Force

The total force needed for vehicle acceleration includes both linear and rotational components [17]:

$$F_a = F_{la} + F_{wa} \tag{8}$$

$$F_{la} = ma \tag{9}$$

$$F_{wa} = \frac{J_m}{G^2 \eta_G R_w^2} a \tag{10}$$

where,

- a is the linear acceleration of the vehicle (m/s^2).
- J_m is the motor rotor moment of inertia (kg·m^2).
- G is the gear ratio.
- η_G is the gear efficiency.
- R_ω is the wheel radius (m).

2. PROPOSED CONTROL SYSTEM SLIME MOULD OPTIMIZATION-BASED ADJUSTMENT

In this discussion, we will discuss a proposed sensorless FOC system for EVs as it is in the study [18]. The system has online adjustment of the PI regulator parameters, which is optimized by the SMO. The approach seeks to improve the work of the BLDC motor in electric cars. The system enhances efficiency, and the complexity of the hardware is reduced by using sensorless control. We will discuss the principles of work and the advantages of such control strategy in application to EVs as in Figure 3.

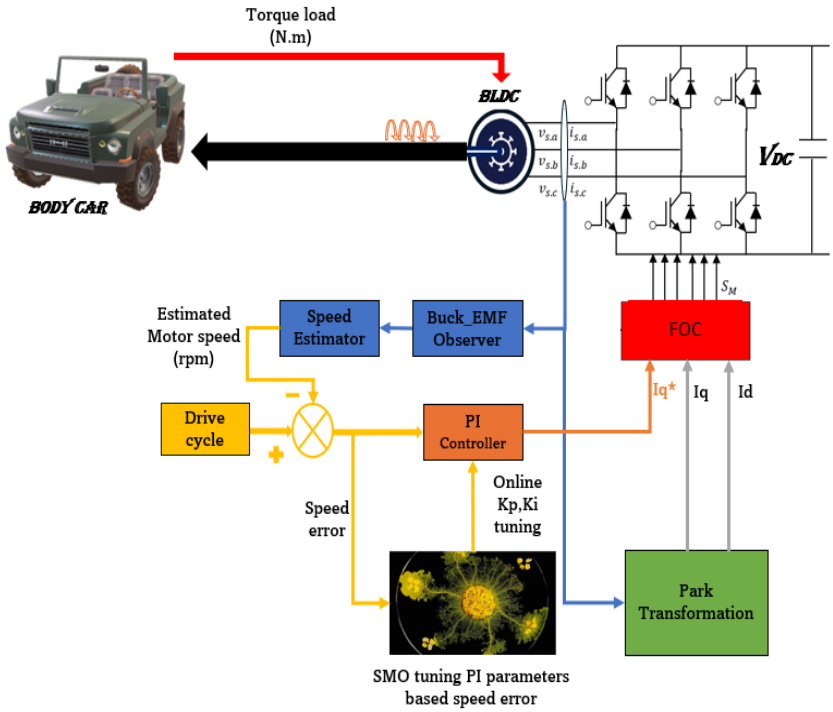


Figure 3. Proposed sensorless field-oriented control (FOC) control system

2.1 Speed estimation method

To estimate the speed, we must estimate the back-EMF first using an observer based on measured values (current, voltage). We will approximate it using the following method:

- The estimation of the back-EMF

To accurately detect the trapezoidal back-electromotive force (back-EMF) of the BLDC motor, an unknown input observer (UIO) is utilized to approximate the back-EMF signals, based on the approach proposed in the study [19].

$$i_{ab} = -\frac{2R}{2L} i_{ab} + \frac{1}{2L} v_{ab} - \frac{1}{2L} e_{ab} \tag{11}$$

Because v_{ab} and i_{ab} are known variables in Eq. (11) that can be measured, but the back-EMF cannot, we can formulate it as follows [20]:

$$\dot{x} = Ax + Bu \tag{12}$$

$$y = Cx \tag{13}$$

And we have:

$$A = \begin{bmatrix} -\frac{2R}{2L} & -\frac{1}{2L} \\ 0 & 0 \end{bmatrix}, x = \begin{bmatrix} i_{ab} \\ v_{ab} \end{bmatrix}, B = \begin{bmatrix} \frac{1}{2L} \\ 0 \end{bmatrix},$$
$$u = [v_{ab}], y = [i_{ab}], C = [1\ 0]$$

So, based on Eqs. (12) and (13), we could construct the following observer:

$$\hat{\dot{x}} = A\,\hat{x} + B\,u + K\,(y - \hat{y}) \tag{14}$$

where, K is the gain matrix [18].

The equation for the observer of one phase (it's the same for the other phases) is as follows [21]:

$$\frac{d}{dt}\begin{bmatrix} \hat{\imath}_{ab} \\ \hat{e}_{ab} \end{bmatrix} = \begin{bmatrix} -\frac{2R}{2L} & \frac{1}{2L} \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \hat{\imath}_{ab} \\ \hat{e}_{ab} \end{bmatrix} + \begin{bmatrix} \frac{1}{2L} \\ 0 \end{bmatrix} v_{ab} + \begin{bmatrix} K_1 \\ K_2 \end{bmatrix}[i_{ab} - \hat{\imath}_{ab}] \tag{15}$$

Finally, in Figure 4, we present the block diagram of the observer [22].

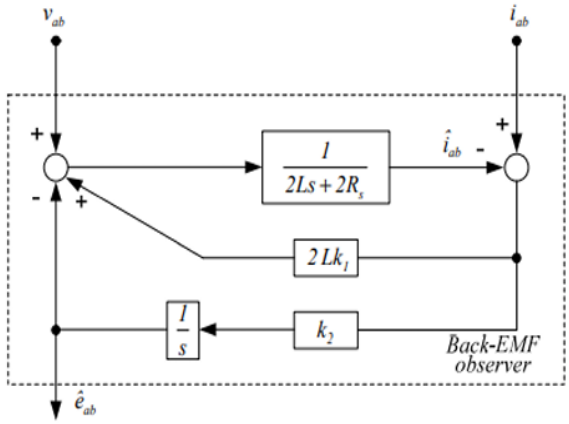


Figure 4. Diagram of the back-Electro-Motive-Force (EMF) observer

- The estimation of speed

Calculating the rotor position and speed is a matter of elementary arithmetic. We have in Eq. (16) [23]:

$$E = K_e \omega_e \tag{16}$$

where,

E is back-EMF amplitude.

K_e is buck-EMF constant.

ω_e is electrical angular velocity.

So:

$$\omega_e = \frac{E}{K_e} \tag{17}$$

Then we have this relation:

$$\omega_m = \frac{2}{p}\omega_e \tag{18}$$

where, P the number of pole pairs.

So the estimated mechanical angular speed $\hat{\omega}_m$ following Eq. (19) as in Figures 5 and 6:

$$\omega_m = \frac{2}{pK_e} E \tag{19}$$

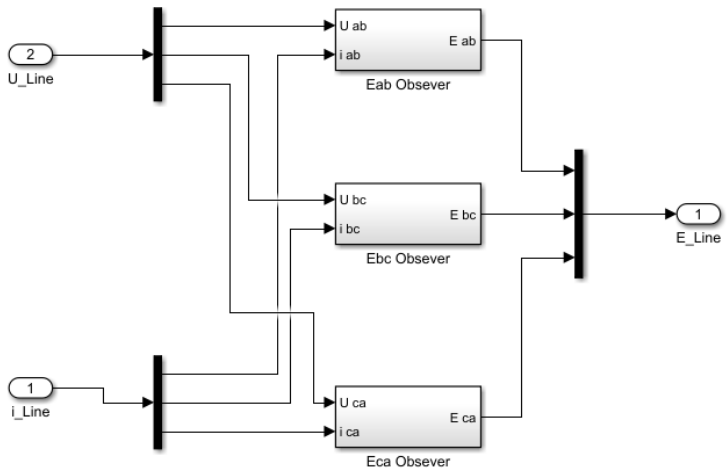


Figure 5. Diagram of the back-Electro-Motive-Force (EMF) amplitude E_line to line (E_{ab}, E_{bc}, E_{ca})

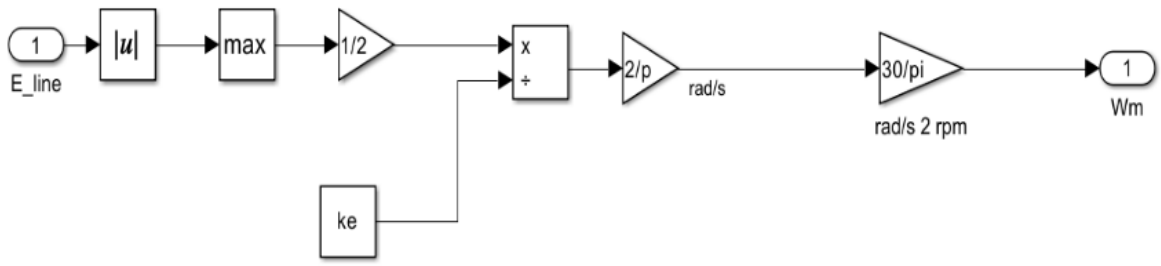


Figure 6. Diagram of the speed estimator

2.2 Derivation of Proportional Integral controller gains K_p and K_i

To design the PI controller for the speed regulation loop, we begin by modeling the motor dynamics and deriving the closed-loop transfer function [24] as in Figure 7.

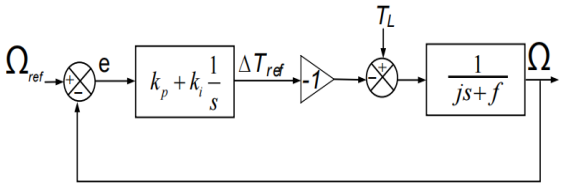


Figure 7. Diagram of the DC motor, including a speed controller

The closed-loop transfer function is given by Eq. (20)

$$\frac{\Omega_{ref}}{\Omega} = \frac{\frac{1}{J}(K_p s + K_i)}{s^2 + \frac{(K_p + f)}{J} s + \frac{K_i}{j}} \tag{20}$$

The standard form of a second-order system is:

$$\frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \tag{21}$$

Matching the coefficients of the denominators [25]:

$$\begin{cases} s^2 + \frac{f + K_p}{J} s + \frac{K_i}{J} = 0 \\ s^2 + 2\xi\omega_n s + \omega_n^2 = 0 \end{cases} \tag{22}$$

$$\begin{cases} K_P = 2J\xi\omega_n - f \\ K_i = J\omega_n^2 \end{cases}$$

where, ω_n is the desired natural frequency (rad/s). ξ is the damping ratio.

2.3 Online tuning of Proportional Integral controller gains using Slime Mould Optimization algorithm

The SMO tune PID functional acts as an online adaptation mechanism of the K_p and K_i obtained by PI controller, are founded on SMO algorithm. In this way the controller is able to dynamically adjust itself in real time, with the measured error in control used as feedback [26].

Principle of Operation

The tuning algorithm takes the speed error $e(t) = \omega* - \omega$ as a fitness function to guide the search process. The goal is to reduce this error by discovering the maximum values of the PI controller as reported in the study [27]. The purpose functions in the following way [28]:

A small set of candidate solutions (usually 3) are generated randomly. Every candidate corresponds to two gains [K_p, K_i] [29]. One candidate is deployed at a time and its performance is measured based on the resulting error. However, the candidate with the most performance (i.e. the one that gives the least output is the best error) is saved as the world optimum [30].

SMO update rule is applied in generating new candidates on the basis of a combination of exploration (diversity) and exploitation (convergence toward the best). The new gains [K_p, K_i] are fed into the speed controller in real-time.

- Fitness Evaluation and Update

Each candidate solution is evaluated using the current system error. If a candidate produces a lower error than its previous best, it is updated [31]:

$$\text{If } e(t) < p_i \Rightarrow p_i = e(t), \text{ and } p_{best,i} = [K_p, K_i]$$

where, p_i is the stored error (fitness) of candidate i, and $p_{best,i}$ is its best gain pair.

- SMO Update Equation

The SMO algorithm updates each candidate using the following form [32]:

$$new_j = g_{best,j} + v_b \cdot [w_i \cdot C_{eq,j} - D_{2,j}] \tag{23}$$

where,

new_j: new value for dimension j ∈ {1,2} corresponding to K_p or K_i.

$g_{best,j}$: best solution found so far.

$C_{eq,j}, D_{2,j}$: two randomly chosen candidate solutions.

v_b: adaptive scaling factor based on iteration.

w_i: weight derived from the candidate's relative fitness.

The adaptive term v_b is computed as [33]:

$$v_b = (2a).rand - a \text{ with } a = \tanh^{-1}(-\frac{iter}{\max\_iter} + 1) \tag{24}$$

2.4 Online adaptation and evaluation mechanism

To ensure that the proposed SMO-PI controller is truly adaptive, two important real-time execution mechanisms were added to the algorithm: the Candidate Evaluation Delay and the Dynamic Re-initialization trigger.

To begin with, a stabilization counter is used to make sure that the measured tracking error accurately shows how the BLDC motor physically reacts to the new PI gains. For 0.0001

seconds (or 10 simulation sample times at Ts = 10 μs), each candidate solution (K_p, K_i) stays the same. This short evaluation window lets the electrical and mechanical transients move through the system before the fitness function checks for errors.

Second, a pulse-triggered Re-initialization controls how well the system can adapt to sudden changes in motor characteristics or outside disturbances. A pulse signal is sent out every time there is a change in state, such as a step change in reference speed or the addition of an external load. As soon as it sees this signal, the SMO algorithm resets its convergence memory (setting the previous best errors to infinity), resets the iteration counter, and creates a new, random initial population in the search space. This mechanism makes the algorithm throw away old control gains and actively look for a new set of optimal parameters that are better suited to the motor's current operating condition. As we can see in Table 1 and Figure 8.

Table 1. Comprehensive setup parameters for the online Adaptive Slime Mould Optimization (SMO) algorithm

Parameter	Value / Description	Purpose / Mechanism
Population Size (N)	3 candidate solutions	Minimizes computational burden for real-time execution.
Maximum Iterations (T_{max})	30 iterations per cycle	Limits the search time to maintain rapid dynamic response.
Candidate Evaluation Time	0.0001 s sample steps)	Allows motor dynamics to settle before evaluating the error.
Adaptive Trigger (pulse)	Active upon state changes	Triggers algorithm re-initialization during speed/load changes.
Reinitialization Action	Resets memory & randomizes	Discards old gains and forces search for new optimal parameters.
Initial Search Space K_p	[0, 10]	Starting range for proportional gain after a reset.
Initial Search Space K_i	[0, 5]	Starting range for integral gain after a reset.
Absolute Boundaries K_p, K_i	[0, 300]	Hard constraints to prevent system instability (saturation).
Fitness Function	Absolute tracking error e	The objective is to minimize the difference between actual and reference speed.

- Advantages

This SMO-based tuning allows the PI controller to adapt in real-time to varying operating conditions such as:

Changes in load torque from the vehicle body.

Nonlinearities or disturbances in motor dynamics [34].

Variations in the drive cycle reference speed.

By continuously minimizing the error through intelligent candidate updates, the controller becomes more robust, adaptive, and efficient for EV applications.

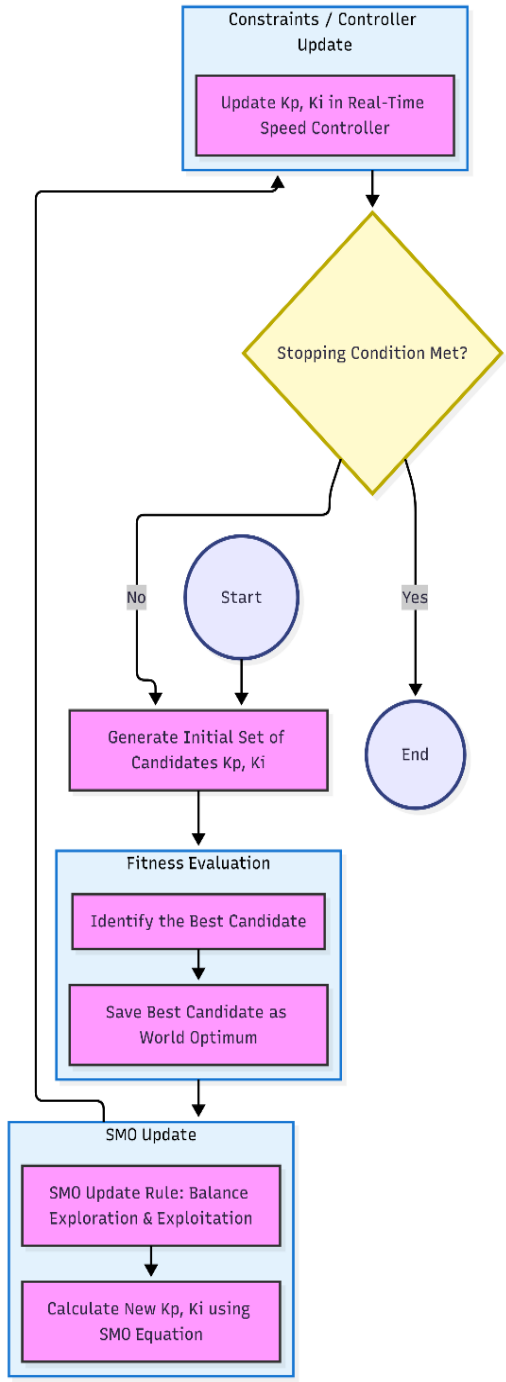


Figure 8. Flowchart of the Slime Mould Optimization (SMO) algorithm

Note: The adaptability of the slime mold model contributes to a dynamic search process, where the algorithm adjusts to environmental conditions for optimal exploration and exploitation. The algorithm's efficiency is influenced by parameters such as the branching factor, which controls the expansion of the search space.

- FOC Control Scheme for BLDC Motor

The three-phase BLDC is converted to FOC scheme. transform the motor control problem into a two-axis equivalent which can be controlled independently. of torque and flux [21]. The schematic is simple and has the following main steps.

Measurement and transformation of three-phase stator currents (i_a, i_b, i_c) transform the dq rotating reference frame into Clarke and Park transformations.

The reference i_q^* is produced by a speed controller (usually a PI controller) based on the speed error. The i_d^* is typically assumed to be zero in order to maximize efficiency.

A current controller compares the i_q^* , i_d^* references containing the actual i_q, i_d calculates the control voltages v_q, v_d.

These voltages are once again converted to three-phase values by the inverse Clarke transformations and inverse Park transformations [35]. The three-phase voltage signals are then output to a PWM inverter that drives the BLDC motor.

This is a closed-loop design to provide rapid dynamic response, linear torque efficiently, and accurate control of speed, and therefore very suitable for EV applications.

3. RESULTS OF THE SIMULATION

Table 2. Brushless DC (BLDC) parameters [36]

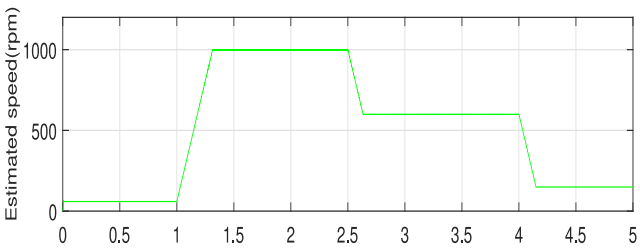
Parameter	Value
Rated speed	2,500 rpm
Rated power	35 kW
Rated voltage	500 V
Rated torque	135 N.m
Moment of Inertia J	45e-4 kg/m²
Friction factor	1e-7
Poles number	2
Torque constant	2.62 N.m/A
Efficiency	94.3%

Table 3. Vehicle body parameters [36]

Parameter	Value
Mass of vehicle M (kg)	800
Vehicle frontal area A_f (m²)	2.5
Transmission ratio I	3.5
Air density ρ (kg/m³)	1.28
Aerodynamic drag coefficient C_d	0.32
Wheel radius R (m)	0.33
Tire rolling resistance coefficient μ	0.0015
Earth gravity g (m/s²)	9.81

3.1 Speed protocol for testing the speed estimator

This protocol is a test of how well a Speed Estimator works over time as in Figure 9. The test profile is used to see how well the algorithm works and how accurate it is in different ranges of operation. The sequence starts off very slowly (60 RPM) to see how well it works with low Back-EMF levels. Then it goes to a high-speed stage (1000 RPM). After that, the speed is set to medium (600 RPM) and low (200 RPM). The main goal is to make sure that the estimator can follow the reference signal with as little error as possible during both steady-state and transient phases. where we could find electric motor parameters in Table 2 and the vehicle body parameters in Table 3.



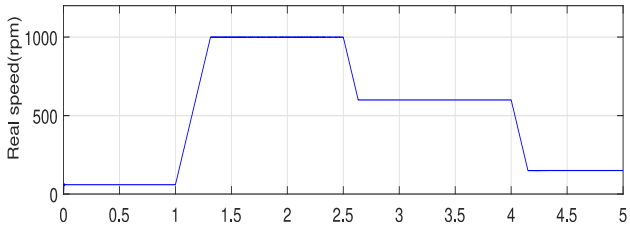


Figure 9. Speed (rpm) Comparison between the speed estimator and speed real speed

3.2 Speed protocol for the comparison between Proportional Integral and Adaptive SMO-PI

The goal of this simulation is to test and compare how well a Conventional PI Controller and an Adaptive PI Controller (optimized using Slime Mold Optimization - SMO) work in real time. The tests are done on a BLDC motor model to see how well it tracks and how strong it is when the conditions change.

The system is subjected to a step-variable speed reference to test its ability to follow setpoint changes every 3 seconds Figure 10:

- **Phase 1 (0 s \to 3 s):** A starting reference speed of 300 RPM.
- **Phase 2 (3 s \to 6 s):** An upward step change to 600 RPM to test the rise time and overshoot.
- **Phase 3 (6 s \to 9 s):** A downward step change to 400 RPM to observe the settling time and recovery.

Exactly at t = 5 s add a torque load T_L.

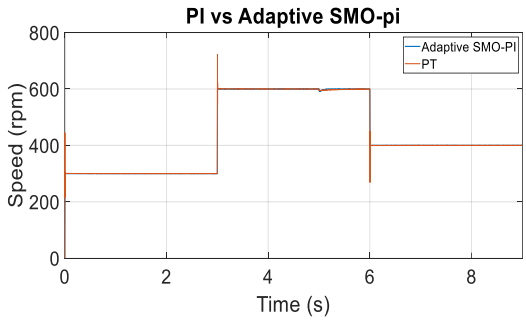


Figure 10. Brushless DC (BLDC) speed (rpm) comparison between using PI and Adaptive SMO-PI
Note: PI = Proportional Integral; SMO-PI: Slime Mould Optimization-Proportional Integral

3.3 Comparative analysis: Conventional Proportional Integral vs. Adaptive SMO-PI

The BLDC motor's dynamic performance was tested with two different ways to control it. The simulation results show that the Adaptive SMO-PI controller is much better than the Conventional PI controller when it comes to transient response, steady-state accuracy, and rejecting disturbances as in Figure 4.

- **Transient Response and Tracking (at t = 3 s)**

The Conventional PI controller went way too far when the reference speed was changed from 300 rpm to 600 rpm (Figure 11). It reached about 725 rpm. It also oscillated for a long time before it settled down. The Adaptive SMO-PI controller, on the other hand, got to the target speed without going over it and settled down much faster. This made it easier for the motor

to switch between tasks and put less strain on it.

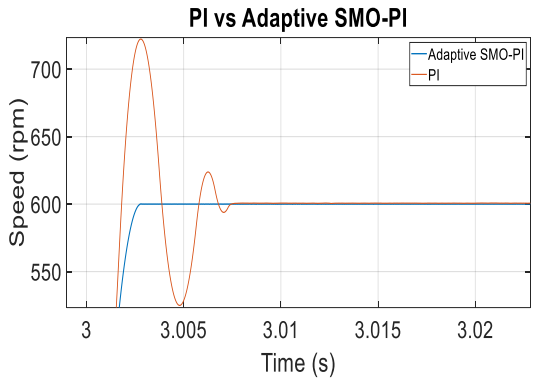


Figure 11. Over shoot and rise time comparison between PI and Adaptive SMO-PI
Note: PI = Proportional Integral; SMO-PI: Slime Mould Optimization-Proportional Integral

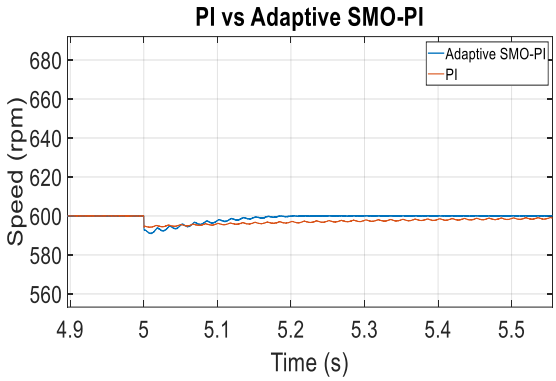


Figure 12. Load recovery comparison between PI and Adaptive SMO-PI
Note: PI = Proportional Integral; SMO-PI: Slime Mould Optimization-Proportional Integral

- **Disturbance Rejection and Robustness (at t = 5 s)**

To test the system's strength, a sudden load torque of 20 Nm was added at 5 seconds as in Figure 12. The Conventional PI controller caused the speed to drop a lot, and it took a long time for it to get back to normal. The ripples were high-frequency (Table 4). The Adaptive SMO-PI controller, on the other hand, did a good job of stopping the disturbance, with only a small change in speed and a quick return to the steady-state speed. This means that the algorithm can change the values of K_p and K_i while it is running.

3.3.1 SMO < K_p, K_i > Response

The dynamic behavior of the control gains (K_P, K_i) shows how well the suggested method works (Figure 13). The SMO changes the values of K_p and K_i all the time, while the regular PI keeps them the same: When the speed changes, K_p quickly rises to give the necessary drive. The gains settle at their best values when the system reaches steady state or a load is put on it. This keeps things stable and gets rid of errors.

3.3.2 Torque (N.m) Response

To see how stable and accurate the control signal was at different speeds and loads, we looked at the electromagnetic torque T_e response. The comparison shows that the Adaptive SMO-PI gives a more "refined" torque response with fewer spikes and better control (Figure 14).

When the reference speed changes, the torque has to change quickly so that the motor can reach the new setpoint.

Conventional PI: During these transitions, it shows very high and sharp torque spikes (overshoot). These spikes can cause high peak currents, which can put stress on the motor windings.

Adaptive SMO-PI: Controls the transition by keeping torque peaks in check. Changing the K_p and K_i values makes the acceleration and deceleration smoother, which lessens the mechanical "kick" on the shaft.

When the load of 20 (N.m) is put on: The torque shows high-frequency ripples and a slightly more chaotic behavior before settling at the required 20 (N.m) to match the load.

Adaptive SMO-PI: There is a very clear and strong increase in torque. The ripple content (Torque Ripples) is much lower, which is important for lowering vibration and noise in BLDC motors.

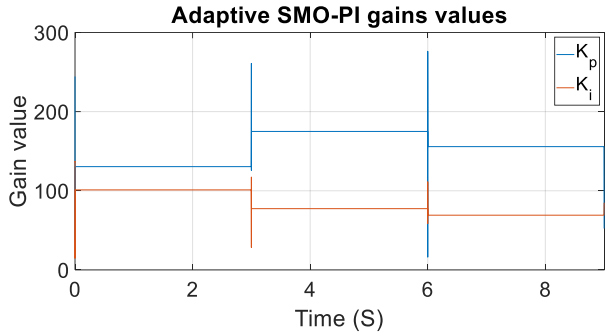
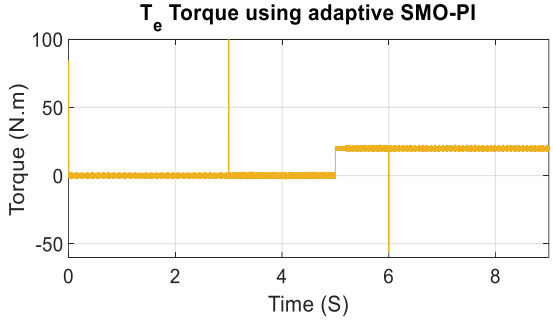


Figure 13. Evolution of K_p and K_i with time

Table 4. Comparative analysis

Performance Metric	Conventional PI Controller	Adaptive SMO-PI Controller	Best Performer
Overshoot	High (approx. 20.8%)	Negligible / Zero (0%)	Adaptive SMO-PI
Settling Time t (s)	Slow with oscillations (t = 0.008 s)	Ultra-fast / Near-instant (t = 0.003 s)	Adaptive SMO-PI
Load Recovery	Significant drop & slow recovery (t = 0.6 s)	Minimal drop & rapid recovery (t = 0.2 s)	Adaptive SMO-PI
Steady-State Error	Moderate ripples under load	Highly precise & smooth	Adaptive SMO-PI
System Stability	Vulnerable to step changes	Robust and highly stable	Adaptive SMO-PI

Note: PI = Proportional Integral; SMO-PI: Slime Mould Optimization-Proportional Integral



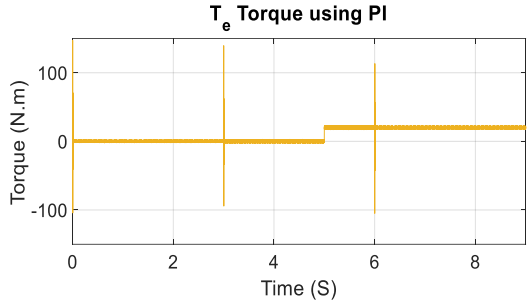


Figure 14. Brushless DC (BLDC) torque (N.m) using PI and Adaptive SMO-PI
Note: PI = Proportional Integral; SMO-PI: Slime Mould Optimization-Proportional Integral

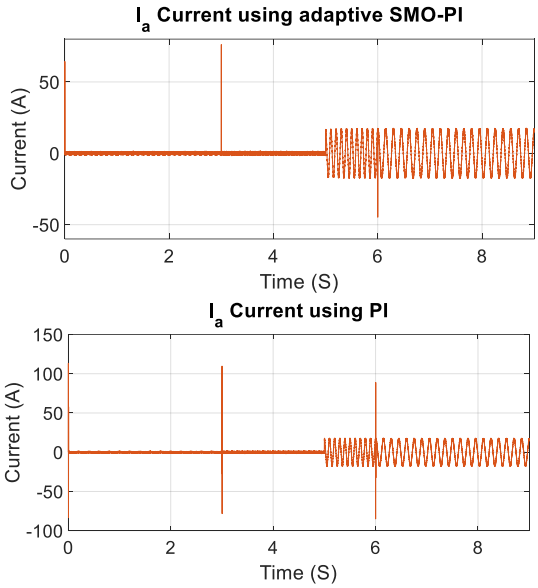


Figure 15. Brushless DC (BLDC) current (A) using Proportional Integral (PI) and Adaptive SMO-PI

3.3.3 Current (A) waveform: Stator phase current analysis Ia: Conventional PI vs. Adaptive SMO-PI

The stator phase current I_a was watched to see how much electrical stress the BLDC motor drive system was under during speed changes and load changes. The comparison shows that the Adaptive SMO-PI controller can limit overcurrent better than the Conventional PI (Figure 15).

Transient current spikes (at t = 3 s and t = 6 s). The most important thing to notice is how big the current spikes are when the car speeds up and slows down:

Traditional PI: The current shows a huge spike of about 110 A at t = 3 s (when the speed increases to 600 rpm). At t = 6 s (deceleration), there is also a sharp negative spike that goes below -90 A. These high currents that aren't limited can cause saturation, overheating, and possibly damage to the power electronic switches.

Adaptive SMO-PI: The adaptive tuning really does help with these transients. At t = 3 s, the peak current is limited to about 80 A, and at t = 6 s, the negative spike is cut down to about -45 A. This decrease shows that the adaptive gain adjustment stops too much control effort.

Performance under load (at 5 seconds). When the load of 20 Nm is put on: To create the needed electromagnetic torque, both controllers raise the current amplitude.

The Conventional: PI current waveform, on the other hand, shows some small irregularities and more harmonics right after the load impact.

The Adaptive SMO-PI keeps a more stable envelope that looks like a sine wave, which means that the torque generation is smoother (as shown in the Torque analysis section).

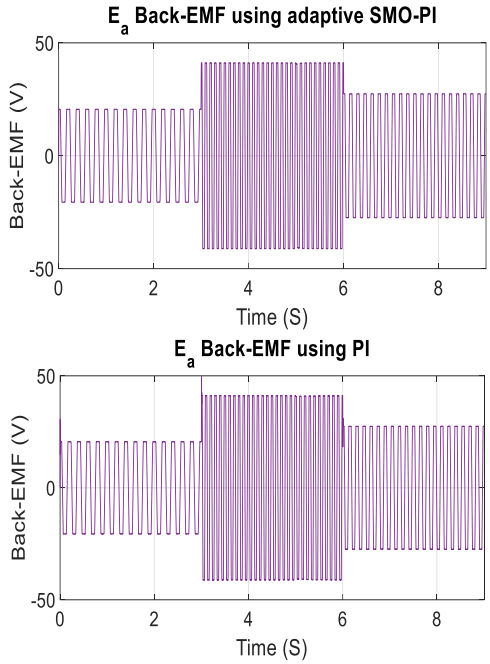


Figure 16. Brushless DC (BLDC) back-Electro-Motive-Force (EMF) (V) using Proportional Integral (PI) and Adaptive SMO-PI

3.3.4 Back-EMF (V) response: Back-EMF Ea Comparison: Standard PI vs. Adaptive SMO-PI

The Back-Electromotive Force (Back-EMF) in a BLDC motor is directly related to the speed at which the rotor turns. So, looking at the E_a waveform directly shows how well the control system is tracking speed and how stable it is overall (Figure 16).

Transient Response and Voltage Overshoot (at t = 3 s and t = 6 s): As seen in the PI Back-EMF waveform, the sudden speed reference step at t = 3 s causes a noticeable voltage overshoot in the amplitude of E_a. The Back-EMF amplitude briefly went above its steady-state boundary because the PI controller let the speed go up to 725 rpm. This puts extra voltage stress on the motor insulation and the inverter parts.

Adaptive SMO-PI: The adaptive controller shows a smooth transition. The Back-EMF amplitude instantly and smoothly rises to the new required level because it tracks speed without overshooting. There are no transient voltage spikes. Stability in steady state and effect of load.

The small speed ripples and slow load recovery at t = 5 s in Conventional PI cause small changes in the frequency and amplitude of the Back-EMF signal.

Adaptive SMO-PI: The waveform keeps a very uniform and symmetrical trapezoidal shape in all phases. This perfect regulation makes sure that the phase currents interact with each other in the best way possible, which is what caused the smooth, ripple-free torque generation we saw earlier.

3.4 Complete assessment in a realistic electric vehicle drive cycle

To confirm the real-world usefulness of the suggested Adaptive SMO-PI controller, the system was tested with a

realistic EV drive cycle that used real EV parameters like inertia, aerodynamic drag, and changing load conditions.

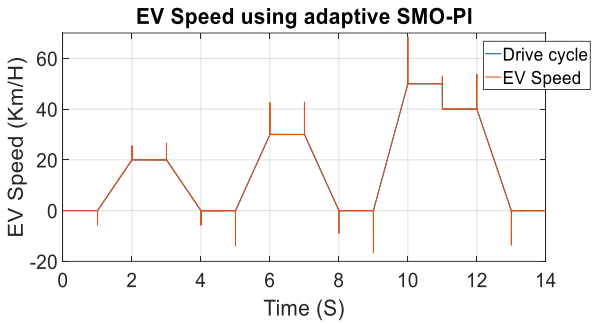


Figure 17. Electric vehicle (EV) speed (Km/H)

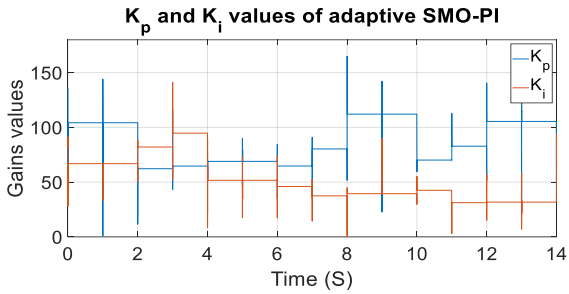


Figure 18. K_p and K_i values using Adaptive SMO-PI

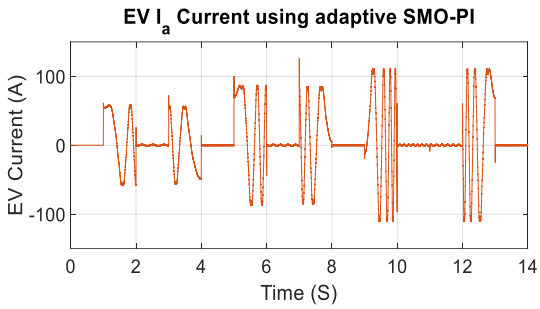


Figure 19. Electric vehicle (EV) current (A)

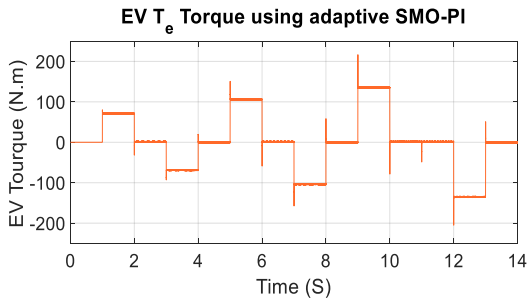


Figure 20. Electric vehicle (EV) torque (N.m)

3.4.1 Perfect speed tracking of trajectories

The EV speed fits perfectly with the reference drive cycle profile. The controller smoothly handles acceleration ramps, constant cruising speeds (up to 50 km/h), and deceleration phases with almost no steady-state error (Figure 17).

3.4.2 Dynamic control effort (adaptive gains)

The Kp and Ki parameters (Figure 18) show that the system is always changing in real time. The SMO algorithm quickly recalculates the best gains at each transition (for example, when the car goes from accelerating to cruising or from cruising to braking) to make up for the car's weight and changing mechanical loads (Figure 18).

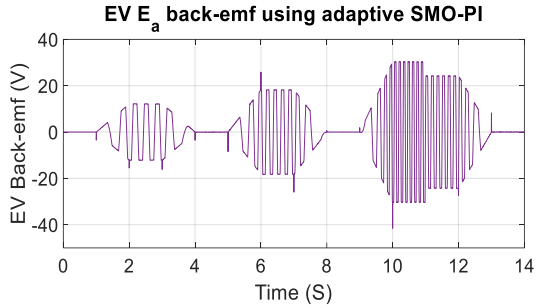


Figure 21. Electric vehicle (EV) back-Electro-Motive-Force (EMF) (V)

The overall performance of the system shows that it is very strong and can adapt to changes quickly.

3.4.3 Good torque and current management

The electromagnetic torque $T_{electromagnetic}$ (Figure 19) and phase currents I_a (Figure 18) show how real EV physics works. During acceleration, the controller needs stable positive torque. During cruising, it needs to keep the torque as low as possible to overcome friction. During deceleration, it needs to effectively apply negative torque, which is what efficient Regenerative Braking is all about (Figure 20).

3.4.4 Stable back-the back-Electro-Motive-Force Ea

waveform scales perfectly symmetrically with the EV's speed changes, keeping a clean trapezoidal shape without any dangerous voltage spikes. This protects the motor insulation and inverter (Figure 21).

3.5 Hardware-in-the-Loop validation setup

To justify the performance and in-situ flexibility of the suggested FOC strategy with SMO-optimized PI tuning, a HIL system was implemented. Such an arrangement allows for the safe, accurate, and rigorous testing of the control algorithm under realistic dynamic conditions without the immediate necessity of a full physical prototype [37]. As illustrated in Figure 21, the HIL architecture is sub-divided into two main hardware subsystems communicating in real-time [38]:

3.5.1 Plant system (virtual EV & motor dynamics):

This subsystem represents the physical dynamics of the three-phase BLDC motor, the mechanical drivetrain, and the EV load. It is executed on a dedicated dSPACE real-time processing node. The model is programmed using MATLAB/Simulink and incorporates both electrical equations and mechanical subsystems, applying variable torque loads that accurately represent the EV's aerodynamic and inertial behaviors [39].

3.5.2 Control system (adaptive FOC algorithm)

The second dSPACE unit acts as the Electronic Control Unit (ECU) of the vehicle. It is dedicated solely to executing the control algorithms, which include the sensorless FOC, the PI speed controller, and the online SMO tuning mechanism. It processes the feedback signals from the virtual plant and

generates the necessary Voltage Source Inverter (VSI) reference signals via PWM logic [40].

3.5.3 Hardware-in-the-Loop timing, sampling, and communication delay

To ensure strict real-time execution and accurately emulate real-world hardware constraints, the sampling frequencies and communication interfaces between the two dSPACE units were rigorously configured.

- **Execution Rates:** Both the Plant subsystem and the Control subsystem operate at a fundamental base sample time of $T_s = 1^{-4}$ s. This high-frequency execution is critical to capture the fast electrical transients of the BLDC phase currents and to ensure precise generation of the 10 kHz PWM switching signals.
- **Hardware Communication Loop:** As depicted in Figure 21, the two units are physically hardwired. The Plant outputs continuous states (phase currents i_a, i_b, i_c, and rotor speed ω via Digital-to-Analog Converters (DAC). These analog signals are continuously sampled by the Control unit's Analog-to-Digital Converters (ADC). Conversely, the calculated gating signals from the Control unit are sent back to the Plant via high-speed Digital Outputs to Digital Inputs (1 . . . 6 PWM channels).
- **Hardware Loop Delay:** The physical hardwiring, combined with the DAC/ADC conversion times and the 1^{-4} s step size, introduces a highly realistic, bounded communication delay of approximately 1 → 2 sample periods 1^{-4} s to 2^{-4} s. This inherent hardware loop latency perfectly emulates the actual signal delays caused by real-world current transducers, Hall-effect sensors, and inverter dead-times, thereby validating the robustness of the adaptive SMO-PI controller against realistic sensor delays and processing latencies [41].

3.5.4 Interface, monitoring, and programming environment

The plant and control subsystems are modeled and compiled using MATLAB/Simulink and the dSPACE Real-Time Interface (RTI), allowing seamless C-code generation and deployment to the hardware processors. The overall setup is monitored via dSPACE ControlDesk, a comprehensive graphical interface used for:

- Monitoring internal real-time signals (currents, voltages, speed, torque).
- Adjusting adaptive boundaries and reference EV drive cycles on the fly.
- Triggering hardware test scenarios, such as sudden torque disturbances.

This dual-dSPACE HIL setup provides a flexible, safe, and powerful environment for verifying the true real-time execution capabilities of the proposed control method under near-real EV driving conditions [42].

3.6 Hardware-in-the-Loop real-time simulation results analysis of speed variable speed reference with constant torque

The results of the HIL real-time simulation presented demonstrate the operation of a FOC sensorless of a BLDC motor in an EV use (Figure 22). The controller uses PI regulator in the current and speed loops, and parameters adjusted to give optimum transient, stability and constant reaction and accuracy to a variable speed reference profile (have step changes which imitate acceleration, cruise and deceleration phases). A Buck EMF observer is used to estimate the Rotor position and speed, which calculates EMF and angle of measured voltages and currents of sensorless operation. A steady torque load (= 6 Nm or so) is applied, represented by constant resistive forces such as rolling resistance or a slight gradient in EV scenarios.

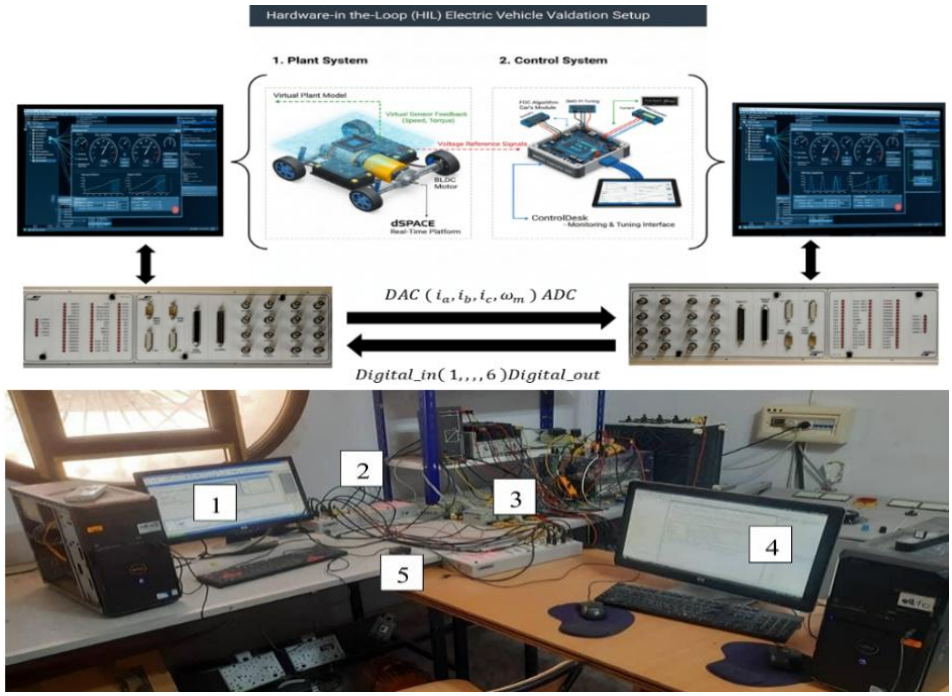


Figure 22. Hardware in the loop of the electric vehicle (EV) system
[1- Computer 1, 2- Dspace DS1104 (of the plant system), 3- Computer 1, 4- Dspace DS1104 (of the control system), 5- Connection cables]

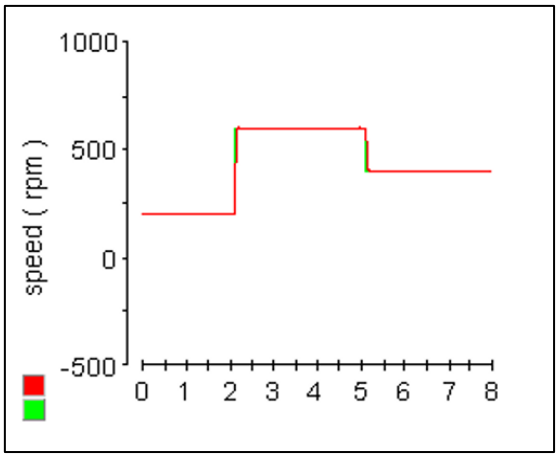


Figure 23. Brushless DC (BLDC) speed (rpm)

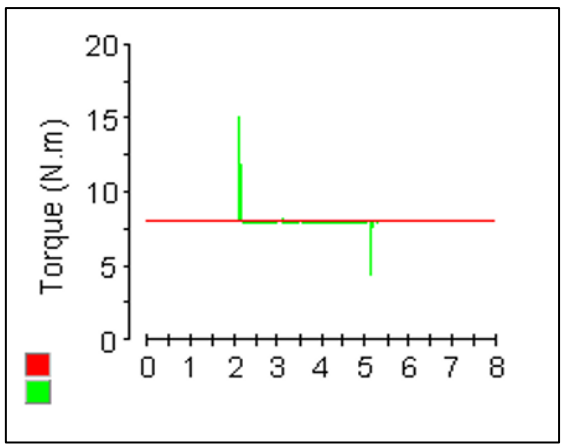


Figure 24. Brushless DC (BLDC) torque (N.m)

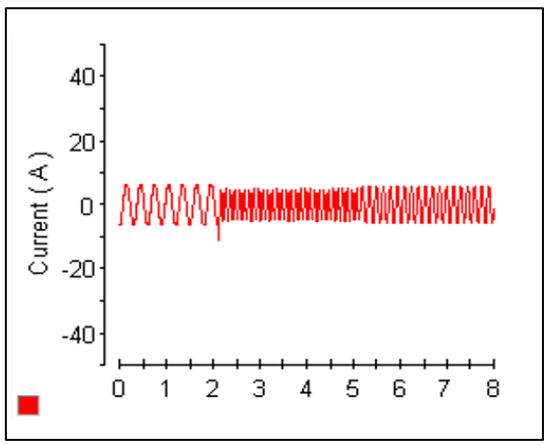


Figure 25. Brushless DC (BLDC) current (A)

- **Speed Response**:

The motor speed is based on a variable reference profile, with a step increase of rpm beginning at 0 rpm to 200 rpm (approximately) (t = 0 s), a plateau to t = 2 s and then a step increase to approximately 600 rpm (t = 2-5 s), and deceleration a step down to 400 rpm (t = 5-8 s). Rise time = 0.1 s with insignificant overshoot (less than 2 percent) and settling times smaller than 0.3 s which implies optimized PI gains in the outer speed loop - proportional term of rapid response as well as the integral of zero steady-state error. Minimal wave (Effects) When plateaus occur, the ripples are minimal (around 5-10 rpm). effective disturbance rejection of the constant load, SMO supplying accurate speed feedback. The step profile loads bandwidth and verifies the suitability of system to EV variable-speed requirements such as traffic or route variations (Figure 23).

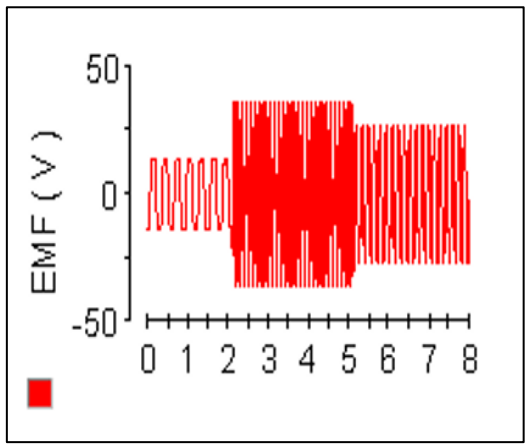


Figure 26. Brushless DC (BLDC) back-Electro-Motive-Force (EMF) (V)

- **Torque Response:**

Torque does not change at 6 Nm all along with small positive surges (+7 Nm) when stepping up in speed simulated at (t = 2 s) step-downs and negative dips (-2 Nm) at (t = 5 s) step-downs quickly damped within 0.1 s. This constancy under FOC (with id = 0 for efficiency) represents decoupled control, with the inner loop works with load without influencing the speed tracking. PI tuning probably in order of priority large integral gain to counter load torque, which makes no sustained deviation, which is vital to EV propulsion regularity (Figure 24).

- **Current Waveform:**

Phase Current sinewaved up and down between -8 A and +8 A, frequency proportional to speed (denser cycles with increasing rpm). Envelope peaks match also torque requirement during transients (10 A at acceleration) and constant amplitude (about 8 A) location (steady-state amplitude) accuracy and convergence (Figure 25).

- **Back EMF:**

EMF estimated between -30 V and +30 V, with trapezoidal waves with a decreasing frequency (matching speed drops). The amplitude is proportional to the rpm maximum 37V at 600 rpm and 25V at 400 rpm,12V at 200 rpm, confirming observer accuracy (Figure 26).

3.2 Hardware-in-the-Loop real-time simulation results and analysis of drive cycle speed reference with real torque load

The given results of HIL real-time simulation demonstrate the work of a. FOC now sensorless strategy of a BLDC motor in an EV setting. The setup employs SMO-PI inner current trollers and outer speed loop trollers, parameters set to provide stability and responsiveness when driving in a reference (e.g., urban such as speed profile consisting of accelerations, cruises and decelerations). Rotor position speed estimation and estimation of the buck EMF is done through a buck EMF Observer that reconstructs back EMF and angle on voltage/current measurements, which makes sensorsless FOC. The load is not constant torque as is the case in constant torque situations created based on a realistic car body model with car

parameters real world road loads such as propulsion loads, regenerative braking, and resistive forces.

- **Speed (Km/H) Tracking:**

The motor speed adheres to a drive cycle reference, initiating from 0 rpm with a rapid ramp to 50 Km/H (t = 0–1 s), followed by a plateau and controlled deceleration to 30 Km/H (t = 2–4 s), near-zero (t = 4–6 s), a brief surge to 20 Km/H (t = 6–8 s), another dip, and final decay to standstill (t = 10–14 s). Overshoots are limited (< 10% at ramps), with settling times 0.3–0.5 s, demonstrating well-tuned PI gains in the speed loop—likely higher proportional for agility and moderate integral to minimize steady-state error under varying loads. Ripples (20–30 rpm) during transitions reflect torque fluctuations from the vehicle model (e.g., inertia during accel/decel) (Figure 27).

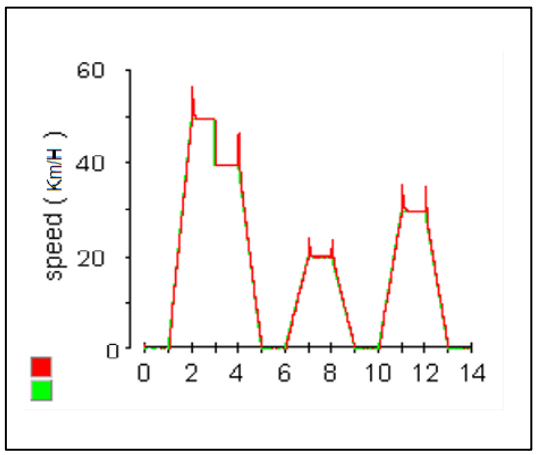


Figure 27. Electric vehicle (EV) speed (Km/H)

- **Torque (N.m) Response:**

Torque varies dynamically per the car body model, starting near 0 Nm, peaking at 150 Nm during initial acceleration (t = 0–2 s) to overcome inertia/aerodynamics, dropping to ~ 0 Nm during coasting (t = 2–4 s), then exhibiting negative values (-100 Nm at t = 4–6 s) indicative of regenerative braking, followed by positive pulses (100 Nm at t = 6–8 s) for re-acceleration, and further negatives for deceleration. This profile mirrors real EV dynamics, with peaks aligning to speed changes. SMO-PI tuning ensures torque follows demands without excessive oscillation, though minor undershoots (10–20 Nm) suggest potential for adaptive gains to better handle nonlinear loads like varying drag (proportional to v 2) (Figure 28).

- **Current (A) Waveform:**

phase current oscillates between -100 A and +100 A, with dense bands from PWM switching (>10 kHz) and FOC transformations. Positive envelopes correspond to motoring torque (e.g., t = 0–2 s, 80 A peak), while negatives indicate regeneration (e.g., t = 4–6 s, -70 A). The waveform's frequency scales with speed, showing SMO chattering effects (high-frequency ripples), but the PI current loop (tuned for high bandwidth, e.g., crossover > 1 kHz) regulates effectively, limiting distortion. Tuning likely used root locus or frequency response methods to achieve damping ratio $\zeta \approx 0.7$, preventing instability from inductance/resistance mismatches in the BLDC model (Figure 29).

- **Back EMF (V) Response:**

SMO-estimated EMF varies from -40 V to +40 V, with trapezoidal waves typical for BLDC under FOC, frequency aligned to speed (e.g., denser cycles early on). Amplitude scales with rpm, peaking ~ 40 V at max speed. The observer robustly handles load-induced disturbances, with minimal phase errors, though low-speed regions (Figure 30).

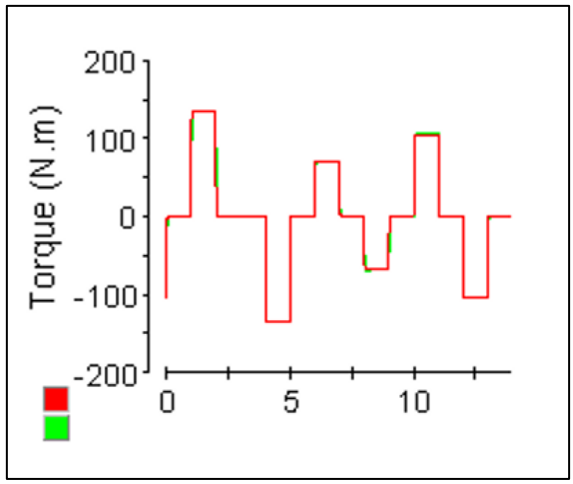


Figure 28. Electric vehicle (EV) torque (N.m)

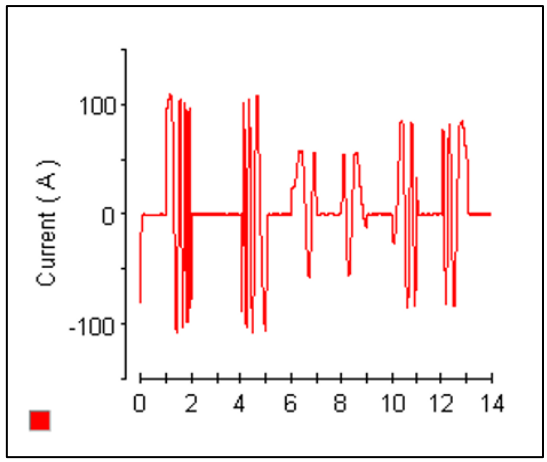


Figure 29. Electric vehicle (EV) current (A)

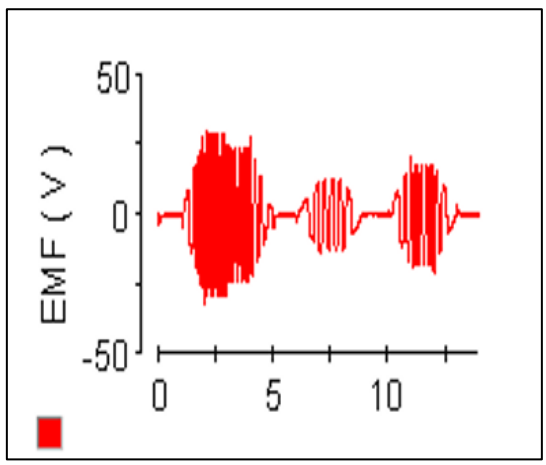


Figure 30. Electric vehicle (EV) back-Electro-Motive-Force (EMF) (V)

4. CONCLUSION

This work presented a sensorless FOC strategy for BLDC

motors in EV applications, supported by an online PI tuning mechanism based on the SMO algorithm. The back-EMF observer successfully replaced mechanical sensors by accurately estimating rotor speed and position from measured electrical signals, thereby reducing hardware cost and improving system reliability. The adaptive SMO-based tuning enabled the PI regulators to adjust dynamically according to real-time operating conditions, ensuring fast convergence, minimal overshoot, and robustness against nonlinear disturbances arising from vehicle dynamics. Both simulation and HIL experiments confirmed the effectiveness of the proposed control approach. Under variable speed references, constant torque disturbance, and realistic drive-cycle profiles, the system exhibited excellent speed tracking capability, smooth torque production, reduced current ripple, and accurate back-EMF reconstruction. The methodology demonstrated stable operation in both motoring and regenerative braking modes, highlighting its suitability for real-time EV propulsion systems. Overall, the integration of sensorless FOC, intelligent PI tuning via SMO, and EV load modeling provides a comprehensive and practical solution for improving the performance and adaptability of BLDC motor drives. Future work may investigate the use of multi-objective optimization, advanced observers for low-speed operation, and experimental validation on a full physical EV drivetrain.

ACKNOWLEDGEMENTS

Sincere gratitude to the LGEB for the invaluable support during this research; the equipment in the laboratory was crucial for HIL validation.

REFERENCES

[1] Rau, D., Rodina, J., Palkovič, L., Hubinský, P. (2015). Sensorless field oriented control of BLDC motors for MAVS. Transactions on Electrical Engineering, 4(4): 91-96. https://doi.org/10.2478/jee-2023-0027

[2] Akrami, M., Jamshidpour, E., Nahid-Mobarakeh, B., Pierfederici, S., Frick, V. (2024). Sensorless control methods for BLDC motor drives: A review. IEEE Transactions on Transportation Electrification, 11(1): 135-152. https://doi.org/10.1109/TTE.2024.3387371

[3] Ranjith, S., Abeykoon, C., Maithripala, D. (2022). Design of a sensorless field oriented control drive for brushless DC motors. In Proceedings of the 9th International Conference of Control, Dynamic Systems, and Robotics (CDSR'22), Niagara Falls, Canada, pp. 2-4. https://doi.org/10.11159/cdsr22.180

[4] Muthamizhan, T., Saravanan, P., Maharana, R. (2021). Sensorless control of z source inverter fed BLDC motor drive by FOC-DTC hybrid control strategy using fuzzy logic controller. In 2021 7th International Conference on Electrical Energy Systems (ICEES), Chennai, India, pp. 358-363.

[5] Sharma, P.K., Sindekar, A.S. (2016). Performance analysis and comparison of BLDC motor drive using PI and FOC. In 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC), Jalgaon, India, pp. 485-492. https://doi.org/10.1109/ICGTSPICC.2016.7955350

[6] Zhang, Z. (2022). Sensorless back EMF based control of synchronous PM and reluctance motor drives—A review. IEEE Transactions on Power Electronics, 37(9): 10290-10305. https://doi.org/10.1109/TPEL.2022.3162963

[7] Gharehchopogh, F.S., Ucan, A., Ibrikci, T., Arasteh, B., Isik, G. (2023). Slime mould algorithm: A comprehensive survey of its variants and applications. Archives of Computational Methods in Engineering, 30(4): 2683-2723. https://doi.org/10.1007/s11831-023-09883-3

[8] Houssein, E.H., Mahdy, M.A., Shebl, D., Manzoor, A., Sarkar, R., Mohamed, W.M. (2022). An efficient slime mould algorithm for solving multi-objective optimization problems. Expert Systems with Applications, 187: 115870. https://doi.org/10.1016/j.eswa.2021.115870

[9] Nabi, S., Balike, M., Allen, J., Rzemien, K. (2004). An overview of hardware-in-the-loop testing systems at Visteon. SAE Technical Paper 2004-01-1240. https://doi.org/10.4271/2004-01-1240

[10] Soltani, A., Assadian, F. (2016). A hardware-in-the-loop facility for integrated vehicle dynamics control system design and validation. Ifac-Papersonline, 49(21): 32-38. https://doi.org/10.1016/j.ifacol.2016.10.507

[11] Huang, C.L., Chen, G.R., Yang, S.C., Hsu, Y.L. (2019). Comparison of high speed permanent magnet machine sensorless drive using trapezoidal BLDC and sinusoidal FOC under insufficient PWM frequency. In 2019 IEEE Energy Conversion Congress and Exposition (ECCE), Baltimore, MD, USA, pp. 321-325. https://doi.org/10.1109/ECCE.2019.8912495

[12] Arif, A.H., Muslim, M.A., Yudaingtyas, E. (2024). Sensorless field-oriented control (FOC) using sliding mode observer for BLDC motor. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 9(2): 139-148. https://doi.org/10.22219/kinetik.v9i2.1937

[13] Jigang, H., Hui, F., Jie, W. (2019). A PI controller optimized with modified differential evolution algorithm for speed control of BLDC motor. Automatika: Časopis za Automatiku, Mjerenje, Elektroniku, Računarstvo i Komunikacije, 60(2): 135-148. https://doi.org/10.1080/00051144.2019.1596014

[14] Tian, H., Tzelepis, D., Papadopoulos, P.N. (2021). Electric vehicle charger static and dynamic modelling for power system studies. Energies, 14(7): 1801. https://doi.org/10.3390/en14071801

[15] Huaman-Rivera, A., Calloquispe-Huallpa, R., Luna Hernandez, A.C., Irizarry-Rivera, A. (2024). An overview of electric vehicle load modeling strategies for grid integration studies. Electronics, 13(12): 2259. https://doi.org/10.3390/electronics13122259

[16] Kroeze, R.C., Krein, P.T. (2008). Electrical battery model for use in dynamic electric vehicle simulations. In 2008 IEEE Power Electronics Specialists Conference, Rhodes, Greece, pp. 1336-1342. https://doi.org/10.1109/PESC.2008.4592119

[17] Wang, Y., Fujimoto, H., Hara, S. (2016). Driving force distribution and control for EV with four in-wheel motors: A case study of acceleration on split-friction surfaces. IEEE Transactions on Industrial Electronics, 64(4): 3380-3388. https://doi.org/10.1109/TIE.2016.2613838

[18] Awaar, V.K., Nagilla, A., Bannuru, S., Veeramaneni, S., Veerla, P., Abbas, A.H. (2023). Position sensorless field-oriented control of BLDC motor for EV applications.

E3S Web of Conferences, 391: 01176. https://doi.org/10.1051/e3sconf/202339101176
[19] Alex, S.S., Daniel, A.E. (2020). Optimal gain selection strategy in back EMF observer for position sensorless operation of BLDC motors. Arabian Journal for Science and Engineering, 45(3): 1345-1356. https://doi.org/10.1007/s13369-019-03908-5
[20] Soni, U.K., Tripathi, R.K. (2017). Novel estimated back EMF ZDP based sensorless controlled BLDCM using unknown input observer. In 2017 International Seminar on Intelligent Technology and Its Applications (ISITIA), Surabaya, Indonesia, pp. 205-210. https://doi.org/10.1109/ISITIA.2017.8124081
[21] Ramesh, P., Ranjeev, A., Santhakumar, C., Vinoth, J., Bharatiraja, C. (2022). Sensor-less field orientation control for brushless direct current motor controller for electric vehicles. Materials Today: Proceedings, 65: 277-284. https://doi.org/10.1016/j.matpr.2022.06.168
[22] Örnek, B.N., Aydemir, S.B., Düzenli, T., Özak, B. (2022). A novel version of slime mould algorithm for global optimization and real world engineering problems: Enhanced slime mould algorithm. Mathematics and Computers in Simulation, 198: 253-288. https://doi.org/10.1016/j.matcom.2022.02.030
[23] Çabuk, A.S. (2021). Sensorless control of outer rotor brushless DC motor with back-EMF observer for drone. Balkan Journal of Electrical and Computer Engineering, 9(4): 379-385. https://doi.org/10.17694/bajece.958760
[24] Zebiri, F., Benheniche, A., Abd Elhakim, D., Kessal, A. (2024). Energy management strategy with regenerative-breaking recovery of mixed storage systems for electric vehicles. Power Electronics and Drives, 9: 463-481. https://doi.org/10.2478/pead-2024-0029
[25] Ovalle, A., Fernandez, J., Hably, A., Bacha, S. (2016). An electric vehicle load management application of the mixed strategist dynamics and the maximum entropy principle. IEEE Transactions on Industrial Electronics, 63(5): 3060-3071. https://doi.org/10.1109/TIE.2016.2516975
[26] Tang, A.D., Tang, S.Q., Han, T., Zhou, H., Xie, L. (2021). A modified slime mould algorithm for global optimization. Computational Intelligence and Neuroscience, 2021(1): 2298215. https://doi.org/10.1155/2021/2298215
[27] Altay, O. (2022). Chaotic slime mould optimization algorithm for global optimization. Artificial Intelligence Review, 55(5): 3979-4040. https://doi.org/10.1007/s10462-021-10100-5
[28] Precup, R.E., David, R.C., Roman, R.C., Petriu, E.M., Szedlak-Stinean, A.I. (2021). Slime mould algorithm-based tuning of cost-effective fuzzy controllers for servo systems. International Journal of Computational Intelligence Systems, 14(1): 1042-1052. https://doi.org/10.2991/ijcis.d.210309.001
[29] Abid, S., El-Rifaie, A.M., Elshahed, M., Ginidi, A.R., Shaheen, A.M., Moustafa, G., Tolba, M.A. (2023). Development of slime mold optimizer with application for tuning cascaded PD-PI controller to enhance frequency stability in power systems. Mathematics, 11(8): 1796. https://doi.org/10.3390/math11081796
[30] [Behera, S., Choudhury, N.B.D. (2022). Modelling and simulations of modified slime mould algorithm based on fuzzy PID to design an optimal battery management system in microgrid. Cleaner Energy Systems, 3: 100029. https://doi.org/10.1016/j.cles.2022.100029
[31] Liu, X., Zhang, Z., Peng, Z., Yang, J. (2024). Optimization of fuzzy-PID temperature controller based on improved slime mould algorithm. Journal of Physics, 2862: 012017. https://doi.org/10.1088/1742-6596/2862/1/012017
[32] Precup, R.E., David, R.C., Roman, R.C., Szedlak-Stinean, A.I., Petriu, E.M. (2023). Optimal tuning of interval type-2 fuzzy controllers for nonlinear servo systems using slime mould algorithm. International Journal of Systems Science, 54(15): 2941-2956. https://doi.org/10.1080/00207721.2021.1927236
[33] İzci, D., Ekinci, S. (2021). Comparative performance analysis of slime mould algorithm for efficient design of proportional-integral-derivative controller. Electrica, 21(1): 151-159. http://doi.org/10.5152/electrica.2021.20077
[34] Tulpule, P., Rezaeian, A., Karumanchi, A., Midlam-Mohler, S. (2017). Model based design (MBD) and hardware in the loop (HIL) validation: Curriculum development. In 2017 American Control Conference (ACC), Seattle, WA, USA, pp. 5361-5366. https://doi.org/10.23919/ACC.2017.7963788
[35] Zeghlache, A., Djerioui, A., Mekki, H., Zeghlache, S., Benkhoris, M.F. (2025). Robust sensorless pmsm control with improved back-EMF observer and adaptive parameter estimation. Electronics, 14(7): 1238. https://doi.org/10.3390/electronics14071238
[36] Bouzidi, R., Ghadbane, I., Boukhari, M., Bendib, A., Kherbachi, A. (2022). Experimental implementation of low-cost and robust sensorless control based on SOGI-FLL estimator for electric vehicles. International Journal of Power Electronics and Drive Systems, 13(4): 1993. https://doi.org/10.11591/ijpeds.v13.i4.pp1993-2004
[37] Mihalič, F., Truntič, M., Hren, A. (2022). Hardware-in-the-loop simulations: A historical overview of engineering challenges. Electronics, 11(15): 2462. https://doi.org/10.3390/electronics11152462
[38] Yan, Q.Z., Oueslati, F., Bielenda, J., Hirshey, J. (2005). Hardware in the loop for a dynamic driving system controller testing and validation. SAE Technical Paper 2005-01-1667. https://doi.org/10.4271/2005-01-1667
[39] Toman, J., Ančík, Z., Singule, V. (2011). Hardware-in-the-loop testing of control algorithms for brushless DC motor. In Mechatronics: Recent Technological and Scientific Advances, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 165-173. https://doi.org/10.1007/978-3-642-23244-2_20
[40] Xie, B., Wang, S., Wu, X., Wen, C., Zhang, S., Zhao, X. (2022). Design and hardware-in-the-loop test of a coupled drive system for electric tractor. Biosystems Engineering, 216: 165-185. https://doi.org/10.1016/j.biosystemseng.2022.02.014
[41] Schulte, T., Bracker, J. (2008). Real-time simulation of BLDC motors for hardware-in-the-loop applications incorporating sensorless control. In 2008 IEEE International Symposium on Industrial Electronics, Cambridge, UK, pp. 2195-2200. https://doi.org/10.1109/ISIE.2008.4677088
[42] Khanke, P.K., Jain, S.D. (2015). Comparative analysis of speed control of BLDC motor using PI, simple FLC and Fuzzy-PI controller. In 2015 International Conference on Energy Systems and Applications, Pune, India, pp. 296-301. https://doi.org/10.1109/ICESA.2015.7503359