Jurnal
Teknologi

Full Paper

INTELLIGENT SPEED CONTROL OF ELECTRIC VEHICLES USING BIO-INSPIRED ANT COLONY OPTIMIZATION ALGORITHMS

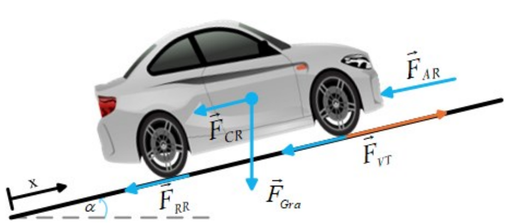
L. D. Hieu*

School of Engineering and Technology-Hue University, 43000, Hue city, Vietnam

Article history
Received
11 June 2025
Received in revised form
6 September 2025
Accepted
8 September 2025
Published Online
29 August 2026

*Corresponding author
ledinhhieu@hueuni.edu.vn

Graphical abstract



Abstract

This paper presents a speed control strategy for an interior permanent magnet synchronous motor (IPMSM) used in electric vehicles employing the ant colony optimization algorithm (ACO). The ACO algorithm is utilized to optimize the speed controller parameters, aiming to minimize overshoots, settling time, and tracking error speed of an electric vehicle (EV). Specifically, the EV model is developed in the MATLAB Simulink environment, utilizing an IPMSM as the primary drive system and powered by a 100 Ah lithium-ion battery. The proposed method is evaluated through simulation and field testing on the MicroAutoBox III dSPACE hardware-in-the-loop. Notably, ACO achieved the lowest threshold overshoot of 0% and the highest of 1.54%, outperforming PI by 24.15% and fuzzy PI by 10.56%, while maintaining comparable response time and stability under a wide range of conditions. In terms of energy efficiency, the ACO algorithm achieved a remaining battery capacity of 99.99702739%, outperforming the conventional PI controller at 99.99702723% and the fuzzy PI controller at 99.99701246%. This corresponds to improvements of approximately 1.6e−7% and 1.493e−5%, respectively. These results demonstrate the significant potential of ACO in improving energy efficiency for electric vehicles.

Keywords: Ant colony optimization, Artificial intelligence, Electric vehicle, Interior permanent magnet synchronous motor

© 2026 Penerbit UTM Press. All rights reserved

1.0 INTRODUCTION

With the depletion of fossil fuels and the increasing severity of climate change, mitigating CO_2 emissions and achieving carbon neutrality by 2050 have become urgent global priorities. Among the various sectors contributing to these emissions, the transportation sector remains one of the largest sources worldwide. As part of the global move toward a greener, more sustainable future, replacing gasoline-powered vehicles with electric vehicles (EVs) offers considerable environmental benefits. These vehicles, which generate zero emissions during operation, play a key role in mitigating pollution and safeguarding the environment [1]. According to a previous study [2], the global EV market is steadily expanding, with EV sales projected to reach US$802.81 billion by 2027 at a compound annual growth rate of 22.6%. Specifically, technological advancements in material manufacturing, power electronics, and control engineering have enabled the adoption of diverse electric motors in EV drivetrains, including synchronous reluctance motors (SynRMs), induction motors (IMs), and permanent magnet synchronous motors (PMSMs) [3, 4].

Among these, IMs offer low manufacturing costs, high rotor durability, and reliable performance, leveraging field-oriented control (FOC). However, elevated rotor heat losses under high-load conditions have driven efforts to explore copper rotor designs to enhance thermal efficiency [5]. Meanwhile, SynRMs feature robust structures and demonstrate high mechanical strength. Moreover, they are unaffected by back-electromotive force, rendering them suitable for high-speed applications. Their performance, however, is constrained by low torque density and high iron loss. To overcome these limitations, materials such as low-loss steel, grain-oriented silicon, and amorphous steel have been proposed [6]. PMSMs, which utilize rare earth magnets, are widely used in drive systems owing to their high torque density and performance advantages over IMs. Nevertheless, their applications are hindered by their high manufacturing costs and low mechanical strength, as the rotor magnets are susceptible to damage from centrifugal force at high operating speeds [7]. Currently, most commercial EVs employ internal-rotor PMSMs (IPMSMs) as their primary drive system [8].

For speed control, these motors typically rely on two primary methods: scalar control and rotor flux-oriented control. Among these, scalar control is commonly preferred in low-precision applications owing to its straightforward design and ease of implementation [9, 10]. To enhance control performance in this context, two vector control strategies are widely adopted: FOC [11, 12] and direct torque control [13, 14]. Previous studies have focused on regulating motor speed using classical proportional–integral (PI) and proportional–integral–derivative (PID) controllers owing to their structural simplicity and stable operation capability [15]. However, the nonlinear characteristics of the system and its complex operating conditions make experimental determination of PI and PID parameters difficult. To address this limitation, recent research has integrated advanced control methods such as fuzzy control [16], robust adaptive control [17], model predictive control [18], sliding mode control [19], and particle swarm optimization control [20], all of which have demonstrated effective performance. Building on this background, the current study introduces an ant colony optimization (ACO) algorithm that autonomously adjusts and optimizes the speed controller of EVs to minimize overshoot, settling time, and speed deviation of the vehicle. The speed control performance and energy consumption of an EV equipped with a 50 kW IPMSM are evaluated under the ECE-15 urban driving cycle. Moreover, the efficiency of the ACO speed controller, fuzzy PI controller, and conventional PI controller is analyzed and compared in the context of the vehicle's drive system. The results demonstrate that the proposed ACO speed controller offers improved performance in terms of speed tracking, overshoot, and fuel consumption. Overall, this paper is structured as follows: Part 1 presents the introduction. Part 2 presents the theoretical background and formulations. Part 3 presents the experimental results. Finally, part 4 presents the conclusion.

2.0 METHODOLOGY

Herein, a dynamic model of an EV is developed to facilitate speed control under a certain driving cycle. This model uses the speed profile derived from the driving cycle as input, sets it as the target speed, and applies motor torque to perform the control action. Figure 1 displays the resisting forces acting on the vehicle when climbing a hill. These forces are determined by the vehicle's design and dynamic properties.

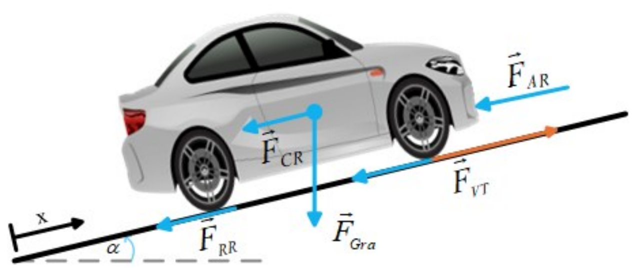


Figure 1 Forces acting on a vehicle when climbing a hill [21]

Based on the driving cycle and vehicle model, the reference speed is used as input to the vehicle's speed control strategy, while the torque required by the vehicle is treated as the load applied to the motor. This setup enables the analysis of system power, energy consumption, and performance.

2.1 Characteristics of the EV

The drag forces acting on the EV include rolling resistance, aerodynamic resistance, and climbing resistance, while the thrust forces comprise linear acceleration and rotational acceleration [22]. These forces are mathematically defined in equations (1)–(6):

$$F_{Gra} = M_{ev} g \quad (1)$$

$$F_{RR} = \mu_{rr} F_{Gra} \quad (2)$$

$$F_{AR} = \frac{1}{2} \rho A C_d v^2 \quad (3)$$

$$F_{CR} = M_{ev} g \sin\alpha \quad (4)$$

$$F_{LA} = M_{ev} \frac{dv}{dt} \quad (5)$$

$$F_{AA} = J \frac{G^2}{\eta_g r^2} \frac{dv}{dt} \quad (6)$$

where F_{Gra} denotes the gravitational force; F_{RR} represents the rolling resistance, and F_{AR} signifies the aerodynamic resistance. Further, F_{CR} denotes the climbing resistance, F_{LA} signifies the inertial force of translational masses, and F_{AA} represents the inertial force of rotational masses. Given these definitions, the total resistance acting on the vehicle is expressed as

$$F_{VT} = F_{RR} + F_{AR} + F_{CR} + F_{LA} + F_{AA} \tag{7}$$

The required torque and angular velocity at the axle and wheel are calculated as follows:

$$T_W = F_{VT} r \tag{8}$$

$$\omega_W = \frac{v}{r} \tag{9}$$

The power transferred from the IPMSM to the vehicle's wheels, considering the gear ratio, gear efficiency, and vehicle dynamics [23], is derived as

$$\omega_M = G\omega_W \tag{10}$$

$$T_M = \frac{1}{G\eta_g} T_W \tag{11}$$

The technical specifications of the IPMSM and vehicle's dynamic parameters are listed in Table 7 of the Appendix.

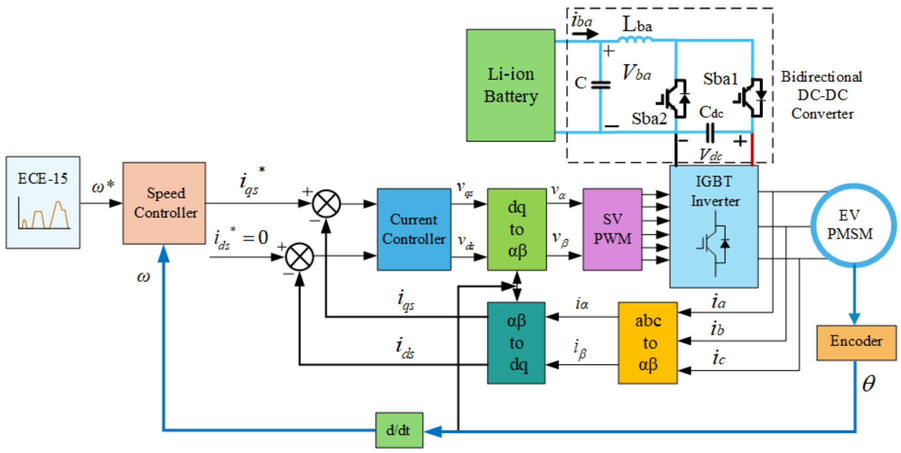


Figure 2 Structure of the speed regulation system based on field-oriented control for the PMSM

2.2 Characteristics of the IPMSM

In this analysis, iron core saturation, hysteresis losses, and eddy current losses in the PMSM are neglected. Additionally, three-phase stator currents are assumed to be symmetrical sinusoidal waves. The voltage equations in the dq coordinate system, as well as the expressions for torque and motor speed, are given as follows [24, 25]:

$$v_{ds} = R_s i_{ds} + \frac{d\psi_{ds}}{dt} - \omega_e \psi_{qs} \tag{12}$$

$$v_{qs} = R_s i_{qs} + \frac{d\psi_{qs}}{dt} + \omega_e \psi_{ds} \tag{13}$$

$$T_e = \frac{3}{2} p(\psi_r i_{qs} + (L_d - L_q) i_{ds} i_{qs}) \tag{14}$$

$$J\frac{d\omega_r}{dt} + f_d \omega_r = T_e - T_L \tag{15}$$

$$\omega_r = \int \frac{T_e - T_L}{J} dt \tag{16}$$

In the above expressions, v_{ds} and v_{qs} denote the dq-axis voltages, i_{ds} and i_{qs} represent the dq-axis currents, ψ_{ds} and ψ_{qs} represent the dq-axis magnetic fluxes, ω_e signifies the angular velocity of the electric field, f_d represents the viscous friction coefficient, ω_r represents the mechanical angular velocity of the rotor, and R_s denotes the effective resistance of the stator winding.

2.3 Construction of the Fuzzy Logic PI Controller and ACO Controller for the IPMSM

The structure of the field-oriented control speed control system for the PMSM used in electric vehicle applications, utilizing reference speed based on the standard ECE-15 driving cycle,is presented in Figure 2.

2.3.1 Fuzzy Logic PI Controller

This paper presents a fuzzy logic controller (FLC-PI) that dynamically adjusts the parameters Kp and Ki of a traditional PI controller to minimize speed control errors for the EV [26, 27], as illustrated in Figure 3.

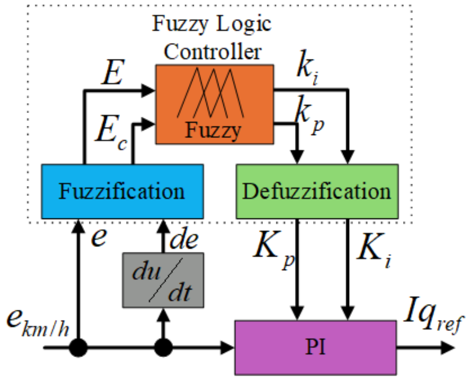


Figure 3 Construction of the fuzzy PI speed loop control algorithm for EV

The FLC-PI algorithm leverages expert knowledge to establish fuzzy rules and integrates them with the classical PI control strategy to optimize EV speed regulation. Notably, in the Takagi-Sugeno fuzzy controller, the input membership functions map the normalized error (E) and its rate of change (E_c) to membership values in the range $\left[-1.5, 1.5\right]$. Meanwhile, the output membership functions are also defined over the same range $K_p = \left[-1.5;\ 1.5\right]$ and $K_i = \left[-1.5;\ 1.5\right]$. Here, five linguistic variables are employed: medium positive (PMs), small positive (PSs), zero (ZOs), small negative (NSs), and medium negative (NMs).

Table 1 Fuzzy logic control rules of K_p, K_i.

E_c \ E	NMs	NSs	ZOs	PSs	PMs
NMs	NMs	NMs	NMs	NSs	ZOs
NSs	NMs	NSs	NSs	ZOs	PSs
Zs	NMs	NS	ZOs	PSs	PMs
PSs	NSs	ZOs	PSs	PMs	PMs
PMs	ZOs	PSs	PSs	PMs	PMs

This study adopts the integral of time-weighted absolute error (ITAE) as the performance evaluation metric for speed control. For calculations, let E(t) denote the error between the motor speed and reference speed over time. This error is calculated as follows:

$$E(t) = Speed_{set} - Speed_{fb} \tag{17}$$

$$ITAE = \int_0^\infty t\left|E(t)\right|dt \tag{18}$$

2.3.2 ACO Controller

The ACO algorithm is used to optimize the parameters of the PID controller and thus improve the system's control performance. At the outset, individual ants may choose suboptimal paths; however, through iterative cooperation, the colony progressively converges on more efficient solutions [28]. After each cycle, pheromone levels deposited on the travel routes are updated according to local pheromone update rules, as outlined in equation (19).

$$\tau_{ij}(k) = \tau(k-1)_{ij} + \frac{0.01\theta}{J} \tag{19}$$

where $\tau_{ij}(k)$ denotes the current pheromone concentration between two points i and j in the search space at iteration k. The parameter θ represents the pheromone concentration adjustment factor, while J denotes the cost function corresponding to the selected route. Pheromone concentrations for the most and least efficient paths are adjusted using equations (20) and (21), respectively:

$$\tau_{ij}(k)^{Best} = \tau(k)_{ij}^{Best} + \frac{\theta}{J_{Best}} \tag{20}$$

$$\tau_{ij}(k)^{Worst} = \tau(k)_{ij}^{Worst} + \frac{0.3\theta}{J_{Worst}} \tag{21}$$

When choosing a direction, each ant typically selects the path with the highest pheromone concentration (J_{Best}) while avoiding those with lower levels. This decision-making process is based on a transition probability influenced by pheromone intensity and, in less favorable scenarios, guided by the cost of the least efficient solution (J_{Worst}). Equation (22) defines the relationship between pheromone concentration and the evaporation constant [29]:

$$\tau_{ij}(k) = \tau(k)_{ij}^{\lambda} + (\tau(k)_{ij}^{Best} + \tau(k)_{ij}^{Worst}) \tag{22}$$

An essential component of the ACO algorithm is the selection of a cost function for evaluating the fitness of each solution. In this study, the ITAE is employed as the cost function and is mathematically expressed as in equation (23):

$$ITAE(E) = \sum_{i=1}^{n} i\left|E_i\right| \tag{23}$$

The parameters of the ACO algorithm used for PID tuning are as follows: $n\_iter = 100$, $NAB = 30$, $Evaporation = 0.9$, $alpha = 0.8$, $beta = 0.2$, $n\_node = 1000$, $n\_para = 3$.

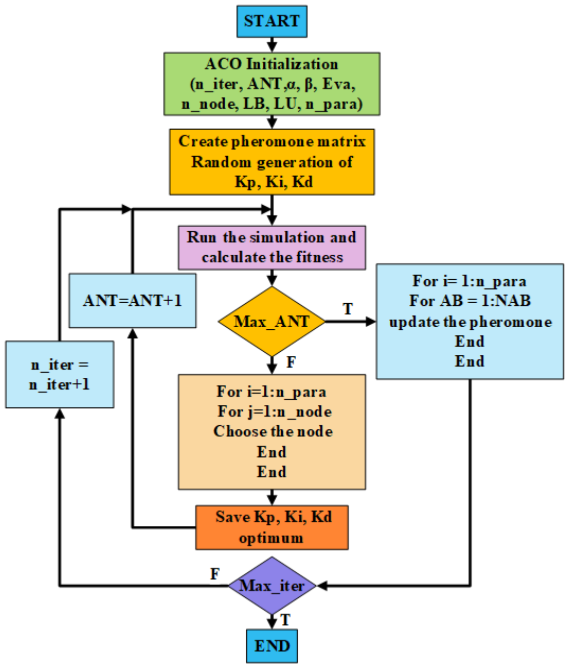


Figure 4 Flowchart of PID controller optimization algorithm using ACO

Figure 4 presents the flowchart of the ACO algorithm for optimizing the PID controller parameters Kp, Ki, and Kd. After the training process, the algorithm yields the optimal values for these parameters.

3.0 RESULTS AND DISCUSSION

3.1 Experimental Setup

The EV model was developed using MATLAB Simulink and implemented on MicroAutoBox III dSPACE hardware. A hardware-in-the-loop (HIL) method was employed to investigate the testing procedure and evaluate control quality, battery power consumption, torque, and response speed under various driving cycles, as illustrated in Figure 5.

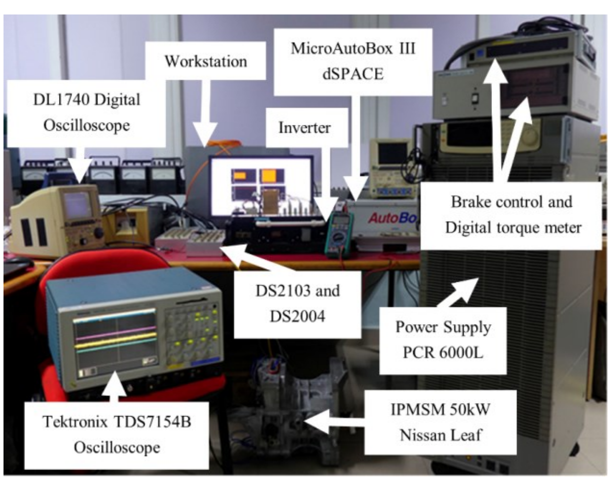


Figure 5 The EV drive system using Matlab-Simulink embedded HIL Micro AutoBox III dSPACE

The MicroAutoBox III dSPACE HIL system integrates a DS1005 real-time hardware core featuring a PowerPC 750GX processor running at 1 GHz with multi-threaded processing. The DS2103 I/O card includes 32 DAC channels with 14-bit resolution and a minimum response time of 10 μs. The DS2004 I/O card, designed for high-speed and high-precision analog-to-digital conversion, contains 16 independent ADC channels with 16-bit resolution and a conversion time of 800 ns per channel.

3.2 Results

The motor and EV parameters are listed in Table 7. The EV uses a lithium-ion battery system with a nominal capacity of 100 Ah [30] and a nominal voltage of 550 V; the initial state of charge (SOC) is set to 100%. The ACO algorithm determines the optimal PID parameters during training with the EV model in the MATLAB Simulink 2023b environment. To evaluate the real-time performance of the ACO-based speed controller in the EV drive system, experiments were conducted on the HIL MicroAutoBox III dSPACE platform at a sampling rate of 45 μs. Figure 6 illustrates the system's convergence behavior under the ACO algorithm, with the optimal cost achieved at 80282.32.

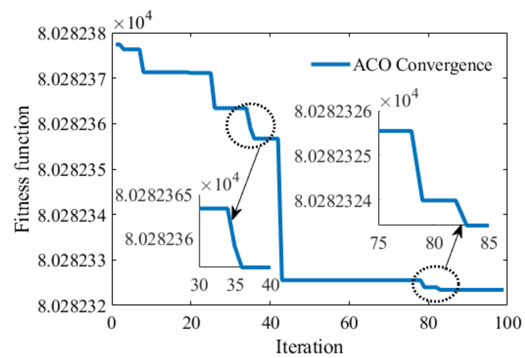


Figure 6 Convergence of the PID coefficient search using the ACO algorithm

A driving cycle based on the European ECE-15 standard was simulated for 200 s, with a maximum vehicle speed of 50 km/h [31, 32]. Notably, this cycle simulates urban driving conditions, characterized by frequent braking, acceleration, deceleration, and stopping, and is used to evaluate vehicle performance and stability. Figure 7 displays the real-time simulation results of the speed response of the ACO, fuzzy PI, and PI algorithms, along with battery SOC, load torque, and three-phase stator current under the driving cycle. The waveforms depict smooth transitions under varying load conditions, demonstrating the effectiveness of the controllers. In the driving cycle graph, the ACO algorithm exhibits low overshoot and fast speed response between 27 and 27.4 s. Between 86 and 86.5 s, it demonstrates superior tracking accuracy compared to the other algorithms. The charging behavior from 50 to 50.5 s illustrates stable battery discharge during acceleration, while the interval from 159 to 159.5 s reflects battery recovery due to regenerative braking. At this stage, the ACO algorithm achieves the highest efficiency in terms of energy optimization. The comparison of the remaining battery capacity shows that the ACO algorithm reaches 99.99702739%, which is higher than the traditional PI controller with 99.99702723% and the fuzzy PI controller with 99.99701246%. This proves that ACO has better energy-saving ability than the other two methods.

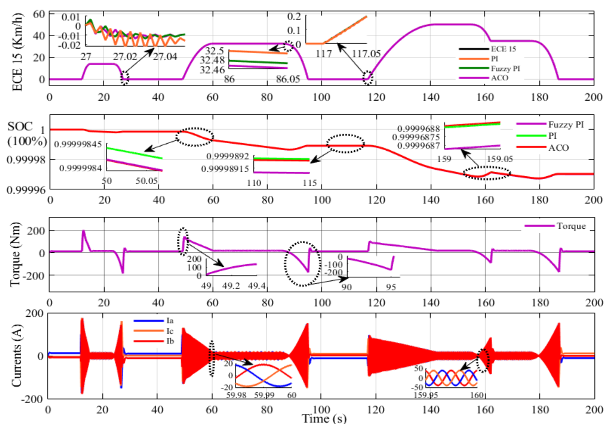


Figure 7 Real-time simulation results obtained from the ECE-15 driving cycle using the HIL system

To evaluate the responsiveness and optimization capabilities of the proposed algorithm, it was compared with the PI and fuzzy PI controllers. To ensure a fair comparison, real-time simulations of the speed response were conducted using the ACO, PI, and fuzzy PI controllers under various load conditions (−30 Nm, no load, and 30 Nm) and speed ranges, as presented in Figures 8–10.

Figure 8 and Tables 2, 5, and 6 illustrate the speed response of the algorithms under no-load conditions.

Figure 8a illustrates the acceleration process from 0 to 1200 rpm. The overshoot of the ACO algorithm is 0.15%, compared to 5.33% for the PI controller and 1.50% for the fuzzy PI controller. The response times are 31 ms for both PI and fuzzy PI, and 39 ms for ACO. The settling times are 48 ms for PI, 37 ms for fuzzy PI, and 64 ms for ACO. These results confirm that at 1200 rpm, the ACO algorithm exhibits the lowest overshoot, followed by the fuzzy PI and PI controllers. At this speed, the response times of the fuzzy PI and PI controllers are equal and approximately 8 ms shorter than that of the ACO controller. The settling time of the fuzzy PI controller is approximately 27 ms shorter than that of the ACO algorithm, while the PI controller's settling time is approximately 11 ms shorter.

Figure 8b illustrates the acceleration process from 0 to 800 rpm. The overshoot of the ACO algorithm is 0.016%, compared to 8.35% for the PI controller and 2.25% for the fuzzy PI controller. The response times are 23 ms for both PI and fuzzy PI, and 35 ms for ACO. The settling times are 37 ms for PI, 58 ms for fuzzy PI, and 56 ms for ACO. These results illustrate that at 800 rpm, the ACO algorithm exhibits the lowest overshoot compared to the fuzzy PI and PI controllers. At this speed, the response times of the fuzzy PI and PI controllers are equal and approximately 12 ms shorter than that of the ACO controller. The settling time of the ACO algorithm is approximately 2 ms shorter than that of the fuzzy PI controller but approximately 19 ms longer than that of the PI controller.

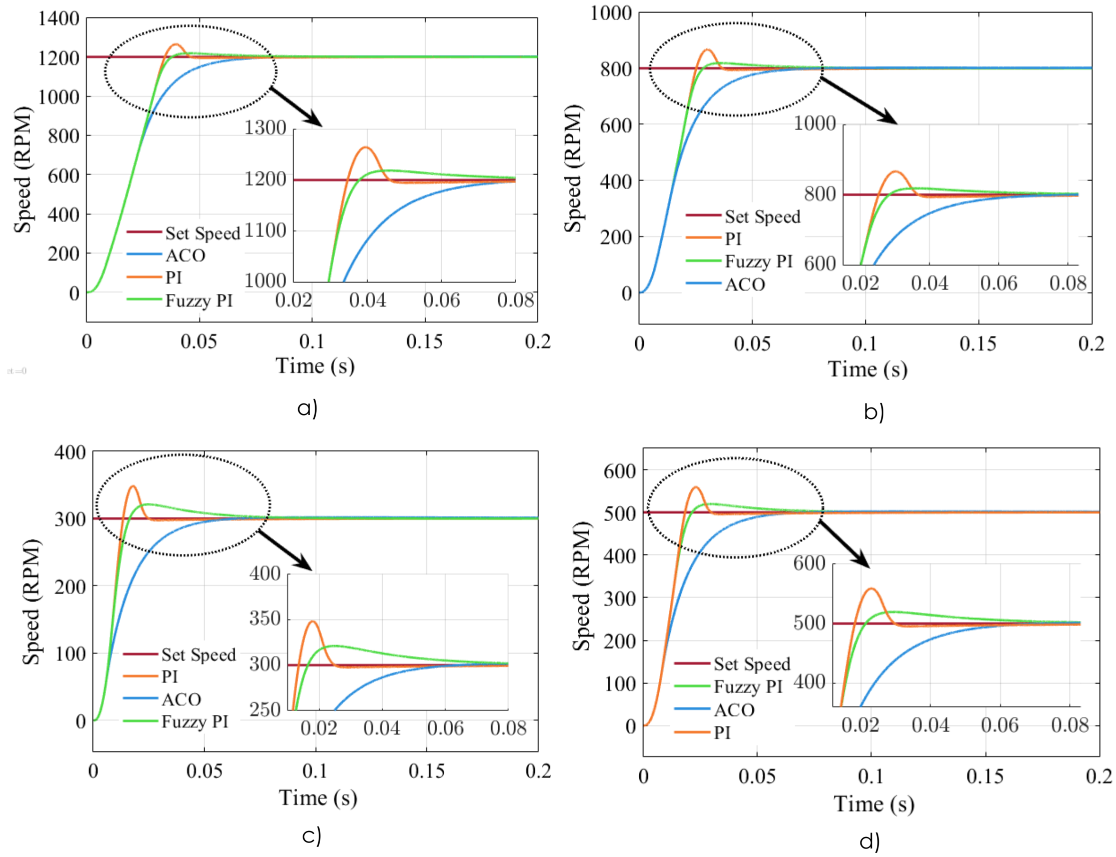


Figure 8 Speed control under no-load conditions

Figure 8c displays the acceleration process from 0 to 300 rpm. The overshoot of the ACO algorithm is 0.54%, compared to 16.08% for the PI controller and 7.02% for the fuzzy PI controller. The response times are 12 ms for PI, 13 ms for fuzzy PI, and 30 ms for ACO. The settling times are 25 ms for PI, 67 ms for fuzzy PI, and 56 ms for ACO. These results illustrate that at 300 rpm, the ACO algorithm exhibits the lowest overshoot compared to the fuzzy PI and PI controllers. At this speed, the response time of the fuzzy PI algorithm is slower than that of the PI controller but approximately 17 ms faster than that of the ACO controller. The settling time of the ACO algorithm is approximately 11 ms shorter than that of the fuzzy PI controller but approximately 31 ms longer than that of the PI controller.

Figure 8d displays the acceleration process from 0 to 500 rpm. The overshoot of the ACO algorithm is 0.21%, compared to 11.80% for the PI controller and 3.87% for the fuzzy PI controller. The response times are 17 ms for PI, 18 ms for fuzzy PI, and 32 ms for ACO. The settling times are 34 ms for PI, 69 ms for fuzzy PI, and 58 ms for ACO. These results indicate that at 500 rpm, the ACO algorithm achieves the lowest overshoot compared to the fuzzy PI and PI controllers. At this speed, the response time of the fuzzy PI algorithm is slower than that of the PI controller but approximately 14 ms faster than that of the ACO controller. The settling time of the ACO algorithm is approximately 11 ms shorter than that of the fuzzy PI controller but approximately 24 ms longer than that of the PI controller.

Therefore, under no-load conditions, the proposed ACO algorithm exhibits the lowest overshoot at all speed levels. However, its response time is slower than that of the PI and fuzzy PI controllers.

Figure 9 and Tables 2, 5, and 6 illustrate the speed response of the algorithms under a 30 Nm load condition.

Figure 9a illustrates the acceleration process from 0 to 1200 rpm under a 30 Nm load. The ACO algorithm has zero overshoot, compared to 3.32% for PI and 0.69% for fuzzy PI. The response times are 33 ms for PI, 34 ms for fuzzy PI, and 41 ms for ACO. The settling times are 48 ms for PI, 42 ms for fuzzy PI, and 72 ms for ACO. These results indicate that at 1200 rpm, the ACO algorithm achieves zero overshoot, followed by the fuzzy PI and PI controllers. At this speed, the response time of the PI controller is approximately 1 ms faster than that of the fuzzy PI controller, and approximately 8 ms faster than that of the ACO controller. The settling time of the fuzzy PI controller is approximately 30 ms shorter than that of the ACO algorithm, while the PI controller's settling time is approximately 6 ms shorter than that of ACO.

Figure 9b illustrates the acceleration process from 0 to 800 rpm. The ACO algorithm has zero overshoot, compared to 5.27% for PI and 1.08% for fuzzy PI. The response times are 25 ms for both PI and fuzzy PI, and

36 ms for ACO. The settling times are 36 ms for PI, 32 ms for fuzzy PI, and 63 ms for ACO. These results indicate that at 800 rpm, the ACO algorithm exhibits zero overshoot. At this speed, the response times of the fuzzy PI and PI controllers are equal and approximately 11 ms shorter than that of the ACO controller. The settling time of the ACO algorithm is approximately 31 ms longer than that of the fuzzy PI controller, and approximately 27 ms longer than that of the PI controller.

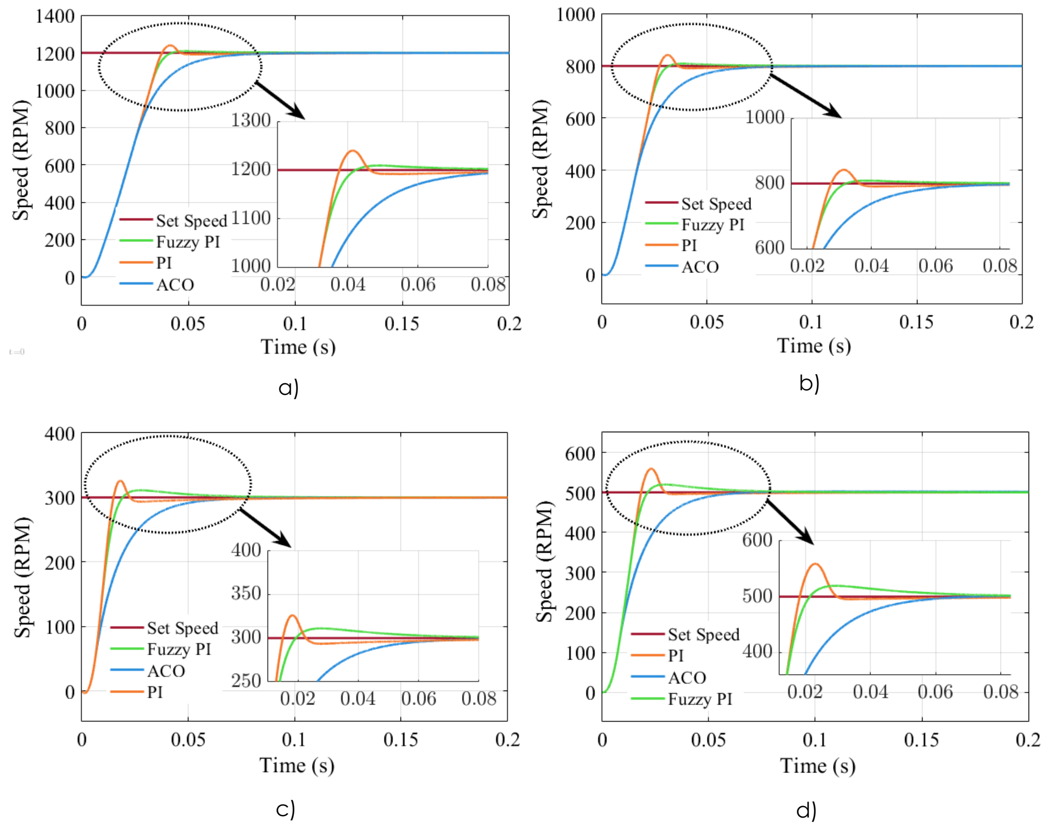


Figure 9 Speed control under load 30 Nm conditions

Figure 9c illustrates the acceleration process from 0 to 300 rpm. The ACO algorithm has zero overshoot, compared to 8.72% for PI and 3.65% for fuzzy PI. The response times are 13 ms for PI, 15 ms for fuzzy PI, and 33 ms for ACO. The settling times are 63 ms for PI, 57 ms for fuzzy PI, and 59 ms for ACO. These results indicate that at 300 rpm, the ACO algorithm exhibits zero overshoot. At this speed, the response time of the fuzzy PI controller is approximately 2 ms slower than that of the PI controller and approximately 18 ms faster than that of the ACO controller. The settling time of the ACO algorithm is approximately 4 ms shorter than that of the PI controller and approximately 2 ms longer than that of the fuzzy PI controller.

Figure 9d illustrates the acceleration process from 0 to 500 rpm. The ACO algorithm has zero overshoot, compared to 11.80% for PI and 3.88% for fuzzy PI. The response times are 18 ms for PI, 19 ms for fuzzy PI, and 34 ms for ACO. The settling times are 66 ms for PI, 43 ms for fuzzy PI, and 68 ms for ACO. These results indicate that at 500 rpm, the ACO algorithm exhibits zero overshoot. At this speed, the response time of the fuzzy PI controller is 1 ms slower than that of the PI controller and approximately 15 ms faster than that of the ACO controller. The settling time of the ACO algorithm is approximately 25 ms longer than that of the fuzzy PI controller and approximately 2 ms longer than that of the PI controller.

Therefore, under a 30 Nm load, the proposed ACO algorithm exhibits the lowest overshoot at all speed levels. However, its response time is slower than that of the PI and fuzzy PI controllers.

Figure 10 and Tables 2, 5, and 6 illustrate the speed response of the algorithms under a −30 Nm load condition.

Figure 10a illustrates the acceleration process from 0 to 1200 rpm under a −30 Nm load. The ACO algorithm has an overshoot of 0.17%, compared to 7.54% for PI and 2.30% for fuzzy PI. The response times are 29 ms for both PI and fuzzy PI, and 38 ms for ACO. The settling times are 45 ms for PI, 70 ms for fuzzy PI, and 71 ms for ACO. These results indicate that at 1200 rpm, the ACO algorithm exhibits the lowest overshoot compared to the fuzzy PI and PI controllers. At this speed, the response times of the PI and fuzzy PI controllers are equal and approximately 9 ms shorter than that of the ACO controller. The settling time of the fuzzy PI controller is approximately 1 ms shorter than that of the ACO algorithm but approximately 25 ms longer than that of the PI controller.

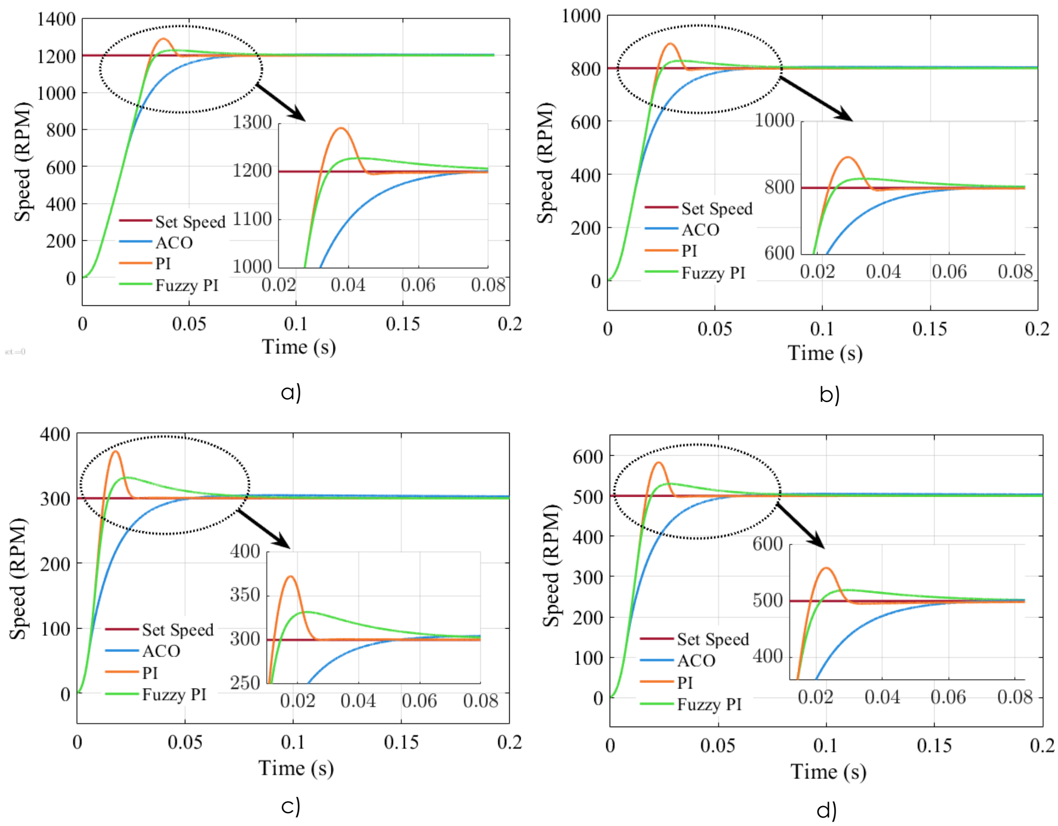


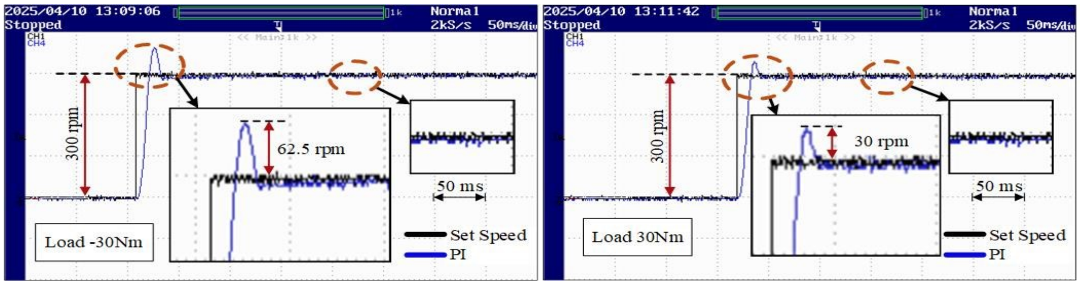
Figure 10 Speed control under load -30 Nm conditions

Figure 10b illustrates the acceleration process from 0 to 800 rpm. The ACO algorithm has an overshoot of 0.17%, compared to 7.54% for PI and 2.30% for fuzzy PI. The response times are 21 ms for PI, 22 ms for fuzzy PI, and 33 ms for ACO. The settling times are 36 ms for PI, 59 ms for fuzzy PI, and 63 ms for ACO. These results indicate that at 800 rpm, the ACO algorithm exhibits the lowest overshoot compared to the fuzzy PI and PI controllers. At this speed, the response time of the fuzzy PI controller is 1 ms slower than that of the PI controller and approximately 11 ms faster than that of the ACO controller. The settling time of the ACO algorithm is approximately 4 ms longer than that of the fuzzy PI controller and approximately 27 ms longer than that of the PI controller.

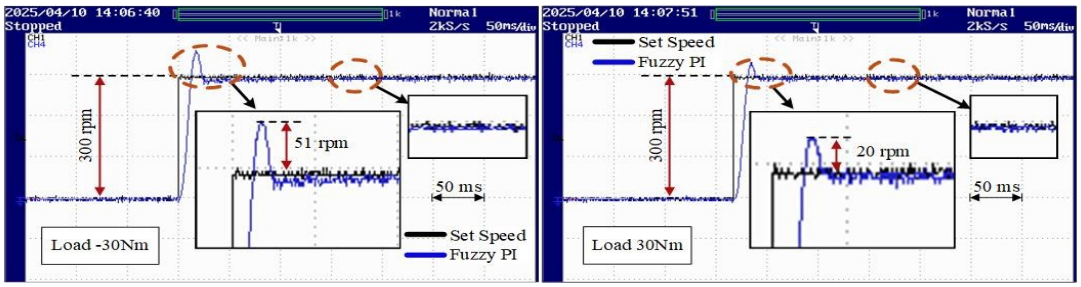
Figure 10c depicts the acceleration process from 0 to 300 rpm. The ACO algorithm has an overshoot of 1.45%, compared to 24.15% for PI and 10.56% for fuzzy PI. The response times are 11 ms for PI, 12 ms for fuzzy PI, and 29 ms for ACO. The settling times are 26 ms for PI, 63.6 ms for fuzzy PI, and 50 ms for ACO. These results indicate that at 300 rpm, the ACO algorithm exhibits the lowest overshoot compared to the fuzzy PI and PI controllers. At this speed, the response time of the fuzzy PI controller is 1 ms slower than that of the PI controller and approximately 17 ms faster than that of the ACO controller. The settling time of the ACO algorithm is approximately 13.6 ms shorter than that of the fuzzy PI controller and approximately 24 ms longer than that of the PI controller.

Figure 10d illustrates the acceleration process from 0 to 500 rpm. The ACO algorithm has an overshoot of 0.20%, compared to 11.80% for PI and 4.15% for fuzzy PI. The response times are 15 ms for PI, 16 ms for fuzzy PI, and 31 ms for ACO. The settling times are 31.2 ms for PI, 65 ms for fuzzy PI, and 86 ms for ACO. These results indicate that at 500 rpm, the ACO algorithm exhibits the lowest overshoot compared to the fuzzy PI and PI controllers. At this speed, the response time of the fuzzy PI controller is 1 ms slower than that of the PI controller and approximately 15 ms faster than that of the ACO controller. The settling time of the ACO algorithm is approximately 21 ms longer than that of the fuzzy PI controller and approximately 54.8 ms longer than that of the PI controller.

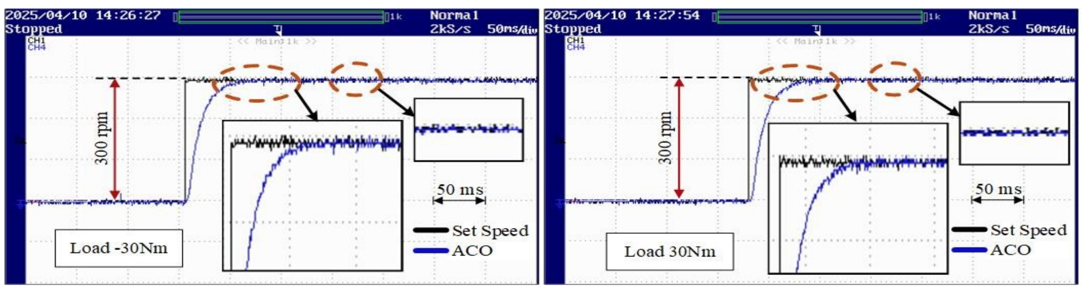
Tables 2, 5, and 6 summarize the overshoot, response time, and settling time metrics, showing that the ACO algorithm consistently achieves the lowest overshoot and delivers strong performance compared to the PI and fuzzy PI controllers.



a) PI algorithm speed response

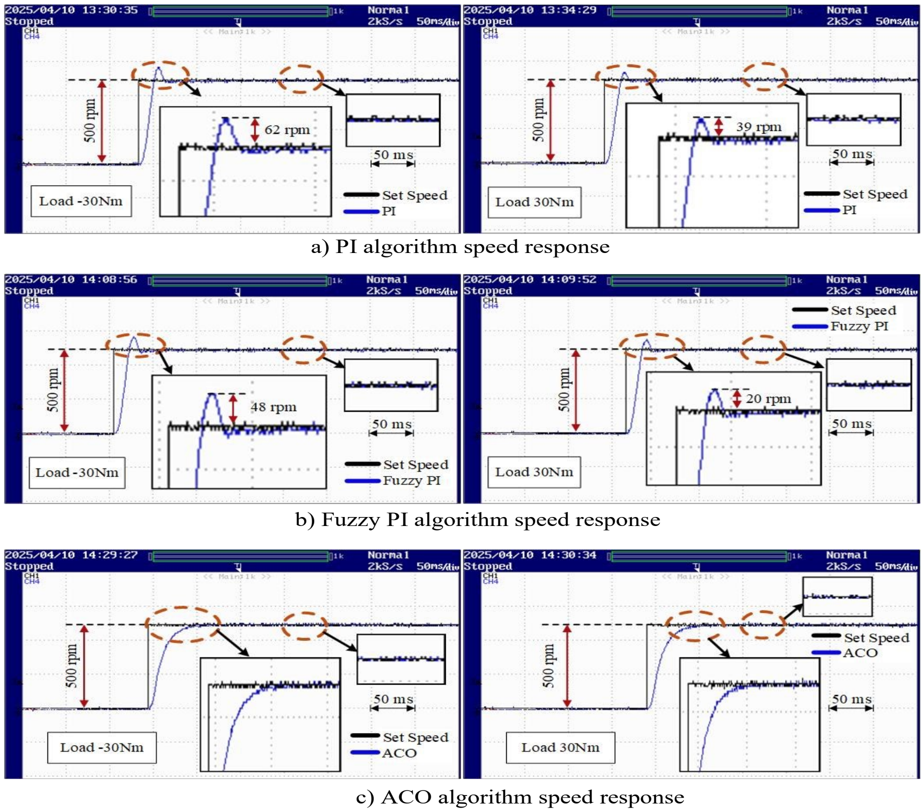


b) Fuzzy PI algorithm speed response



c) ACO algorithm speed response

Figure 11 Speed control at 300 rpm with different loads

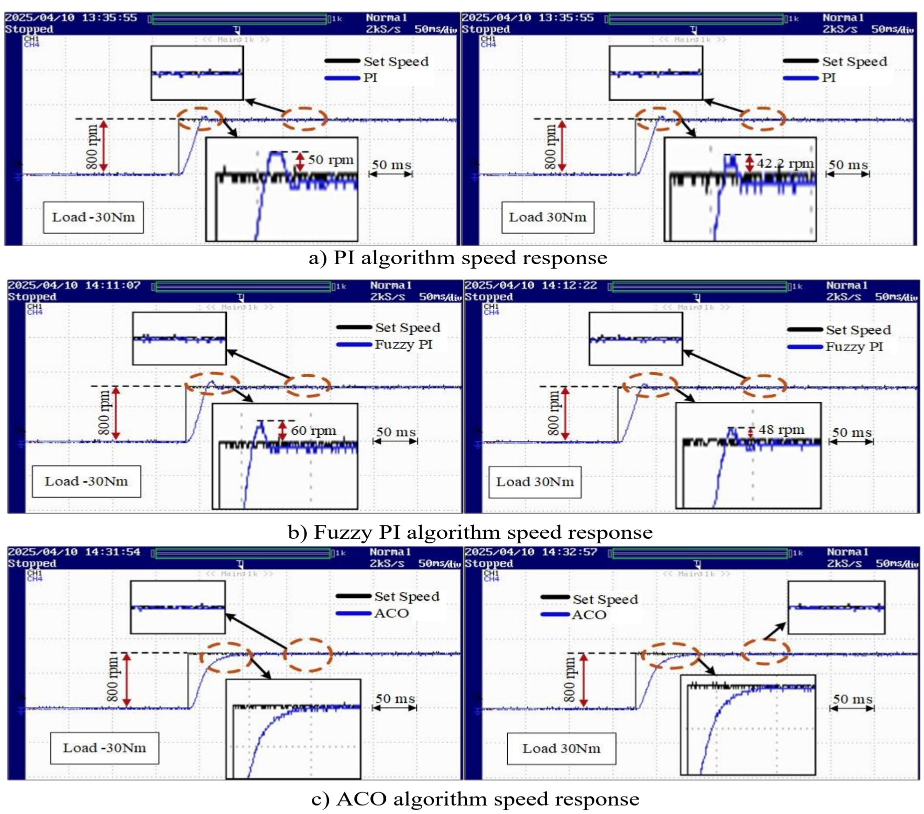


a) PI algorithm speed response

b) Fuzzy PI algorithm speed response

c) ACO algorithm speed response

Figure 12 Speed control at 500 rpm with different loads



a) PI algorithm speed response

b) Fuzzy PI algorithm speed response

c) ACO algorithm speed response

Figure 13 Speed control at 800 rpm with different loads

The Yokogawa DL1740 digital oscilloscope, with a real-time sampling rate of 1 GS/s and an equivalent-time sampling rate of 100 GS/s, was used to verify the experimental results. Experimental results for the ACO, PI, and fuzzy PI algorithms at motor speeds operating within an efficiency range of 93% or higher of 300 rpm, 500 rpm, and 800 rpm under different loads (−30 Nm and 30 Nm) are illustrated in Figures 11, 12, and 13. The settling time and response time of the ACO algorithm are longer than those of the PI and fuzzy PI controllers. However, the proposed algorithm produces considerably lower speed overshoot than the other two controllers, contributing to more stable system operation.

Table 2 displays the speed overshoot of the IPMSM motor under various load and speed conditions. The highest overshoot occurs at 300 rpm with a −30 Nm load, where the ACO algorithm reaches 1.54%, PI reaches 24.15%, and fuzzy PI reaches 10.56%. The lowest overshoot is recorded at 1200 rpm with a 30 Nm load: 0% for ACO, 3.32% for PI, and 0.69% for fuzzy PI.

Table 3 presents the transient load performance index, highlighting the superiority of the proposed algorithm in terms of speed response and performance optimization, based on comparisons of overshoot and settling time with the PI and fuzzy PI controllers.

The comparison results between the ACO algorithm and modern metaheuristic algorithms are summarized in Table 4.

Table 2 Summary of overshoot under different load conditions

Load torque (Nm)	Motor speed (rpm)	Overshoot (%) PI	fuzzy PI	ACO
-30 Nm	300	24.15	10.56	1.45
	500	11.80	4.15	0.20
	800	11.62	3.46	0.64
	1200	7.54	2.30	0.17
No-load	300	16.08	7.02	0.54
	500	11.80	3.87	0.21
	800	8.35	2.25	0.016
	1200	5.33	1.50	0.15
30 Nm	300	8.72	3.65	0
	500	11.80	3.88	0
	800	5.27	1.08	0
	1200	3.32	0.69	0

Table 3 Temporary load performance index

Load torque	Overshoot (%)			Settling time (ms)		
(Nm)	PI	fuzzy PI	ACO	PI	fuzzy PI	ACO
0 to 30	2.34	1.79	0.22	50.22	13.65	2.54
30 to 10	1.26	2.51	0.40	20.8	35	2.96
30 to 0	0.46	1.32	0.02	18.9	32.9	2.24
30 to -30	2.87	4.97	1.58	26.9	31.6	4.7

Table 4 Comparison with other published articles

Controller	Startup speed performance		Transient speed performance			
			Addition of load		Removal of load	
	Overshoot (%)	Settling time (ms)	Overshoot (%)	Settling time (s)	Overshoot (%)	Settling time (s)
PI	11.80	0.034	2.34	0.05	0.46	0.0189
fuzzy PI	3.87	0.069	1.79	0.0136	1.32	0.0329
FOSMC [33]	8.32	6	3.32	1.5	1.06	4
HHO-SMC [18]	0.7	0.046	0.17	0.007	0.18	0.007
FITSMC [34]	2.885	1.027	3	0.69	2.7	0.87
IPSO [35]	5.58	18.5	0.35	0.2016	-	-
ACO	0.21	0.058	0.22	0.0025	0.02	0.0047

Table 5 in the Appendix presents the speed response time of the IPMSM motor under different load and speed conditions. The fastest response times occur at 300 rpm with a −30 Nm load: 29 ms for ACO, 11 ms for PI, and 12 ms for fuzzy PI. The slowest response times are observed at 1200 rpm with a 30 Nm load: 41 ms for ACO, 33 ms for PI, and 34 ms for fuzzy PI.

Table 6 in the Appendix presents the settling times of the IPMSM motor under various load and speed conditions. The fastest settling time for ACO is 50 ms at 300 rpm with a −30 Nm load. For PI, it is 25 ms at

300 rpm with no load, and for fuzzy PI, it is 32 ms at 800 rpm with a 30 Nm load. The slowest settling time for ACO is 86 ms at 500 rpm with a 30 Nm load. For PI, it is 66 ms under the same condition, and for fuzzy PI, it is 70 ms at 1200 rpm with a −30 Nm load.

4.0 CONCLUSION

This paper evaluated the speed control efficiency, torque, SOC, and three-phase currents of the ACO method against conventional PI and fuzzy PI controllers when applied to an IPMSM-based EV. Real-time experiments on MicroAutoBox III dSPACE hardware demonstrate that the ACO effectively optimizes parameters and improves speed control performance when the motor operates within an efficiency range of 93% or higher compared to traditional PI and fuzzy PI. Through simulation and testing, HIL results indicate that the ACO controller provides a fast response and enhances system stability. Notably, ACO achieved the lowest overshoot of 0% and the highest overshoot of only 1.54%, while the PI and fuzzy PI controllers recorded 24.15% and 10.56%, respectively. In addition, ACO maintained shorter or equivalent response time and settling time under most different load and speed conditions. Furthermore, the ACO algorithm achieves the highest remaining battery capacity of 99.99702739%, improving 1.6e−7% and 1.493e−5% over PI and fuzzy PI, respectively. These advantages show that the proposed algorithm is a promising solution to improve motor control in EV applications.

Acknowledgement

This research was funded by Ministry of Education and Training under grant number B2024-DHH-11 and Hue University under the Core Research Program, Grant No. NCTB.DHH.2024.08.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

[1] Gnanavendan, S., S. K. Selvaraj, S. J. Dev, K. K. Mahato, R. S. Swathish, and G. Sundaramali. 2024. Challenges, Solutions, and Future Trends in EV Technology: A Review. *IEEE Access*. 12: 17242–60. https://doi.org/10.1109/ACCESS.2024.3353378.

[2] Qadir, S. A., F. Ahmad, A. M. A. Al-Wahedi, A. Iqbal, and A. Ali. 2024. Navigating the Complex Realities of Electric Vehicle Adoption: A Comprehensive Study of Government Strategies, Policies, and Incentives. *Energy Strategy Reviews*. 53: 101379. https://doi.org/10.1016/j.esr.2024.101379.

[3] Shewalkar, A. G., A. S. Dhoble, and V. P. Thawkar. 2024. Review on Cooling Techniques and Analysis Methods of an Electric Vehicle Motor. *Journal of Thermal Analysis and Calorimetry*. 149: 5919–47. https://doi.org/10.1007/s10973-024-13091-x.

[4] Nordin, M. I., J. M. Lazi, M. H. N. Talib, and Z. Ibrahim. 2022. Speed Sensorless Control for PMSM Drives Using Extended Kalman Filter. *Jurnal Teknologi (Sciences & Engineering)*. 84(1): 77–83. https://doi.org/10.11113/jurnalteknologi.v84.17270.

[5] Popescu, M., L. Di Leonardo, G. Fabri, G. Volpe, N. Riviere, and M. Villani. 2023. Design of Induction Motors with Flat Wires and Copper Rotor for E-Vehicles Traction System. *IEEE Transactions on Industry Applications*. 59: 3889–3900. https://doi.org/10.1109/TIA.2023.3256391.

[6] Zhang, S., C. Liu, Y. Wang, F. Niu, G. Lei, and J. Zhu. 2024. Shape Design Optimization and Comparative Analysis of a Novel Synchronous Reluctance Machine with Grain-Oriented Silicon Steel. *IEEE Transactions on Magnetics*. 60: 8101705. https://doi.org/10.1109/TMAG.2024.3408809.

[7] Sesham, H., J. Vijaya Kumar, and A. Venkatesh. 2025. Performance Evaluation of Various Electric Vehicle Drive Systems in Energy-Efficient Grid-Integrated Renewable Systems. *International Journal of Information Technology*. 1–18. https://doi.org/10.1007/s41870-025-02445-9.

[8] Wang, H., X. Liu, M. Kang, L. Guo, and X. Li. 2022. Oil Injection Cooling Design for the IPMSM Applied in Electric Vehicles. *IEEE Transactions on Transportation Electrification*. 8: 3427–40. https://doi.org/10.1109/TTE.2022.3161064.

[9] Cho, B.-G., C. Hong, J. Lee, and W.-J. Lee. 2021. Simple Position Sensorless V/f Scalar Control Method for Permanent-Magnet Synchronous Motor Drives. *Journal of Power Electronics*. 21: 1020–29. https://doi.org/10.1007/s43236-021-00249-1.

[10] Lee, K., and Y. Han. 2020. MTPA Control Strategy Based on Signal Injection for V/f Scalar-Controlled Surface Permanent Magnet Synchronous Machine Drives. *IEEE Access*. 8: 96036–44. https://doi.org/10.1109/ACCESS.2020.2997087.

[11] Candelo-Zuluaga, C., J.-R. Riba, and A. Garcia. 2021. PMSM Parameter Estimation for Sensorless FOC Based on Differential Power Factor. *IEEE Transactions on Instrumentation and Measurement*. 70: 1–12. https://doi.org/10.1109/TIM.2021.3096861.

[12] You, Z., Y. Bian, Y. Zhang, and L. Chen. 2025. An Intelligent Optimization Algorithm with Novel Fitness Function for High-Performance PMSM FOC. *Alexandria Engineering Journal*. 115: 286–96. https://doi.org/10.1016/j.aej.2024.12.029.

[13] Petkar, S. G., and V. K. Thippiripati. 2022. A Novel Duty-Controlled DTC of a Surface PMSM Drive with Reduced Torque and Flux Ripples. *IEEE Transactions on Industrial Electronics*. 70: 3373–83. https://doi.org/10.1109/TIE.2022.3181405.

[14] Veeresh, M. Y., V. B. Reddy, and R. Kiranmayi. 2022. Modeling and Analysis of Time Response Parameters of a PMSM-Based Electric Vehicle with PI and PID Controllers. *Engineering, Technology & Applied Science Research*. 12: 9737–41. https://doi.org/10.48084/etasr.5321.

[15] Shi, Q., M. Wang, Z. He, C. Yao, Y. Wei, and L. He. 2022. A Fuzzy-Based Sliding Mode Control Approach for Acceleration Slip Regulation of Battery Electric Vehicle. *Chinese Journal of Mechanical Engineering*. 35: 72. https://doi.org/10.1186/s10033-022-00729-w.

[16] Wang, Y., Y. Jiao, H. Chen, and L. Yao. 2025. Adaptive Constraint Control-Based Peak-Attenuated Disturbance Observer for the Electric Vehicle-Used Permanent Magnet Synchronous Motor. *International Journal of Electrical Power & Energy Systems*. 166: 110579. https://doi.org/10.1016/j.ijepes.2025.110579.

[17] Sangar, B., M. Singh, and M. Sreejeth. 2024. An Improved ANFIS Model Predictive Current Control Approach for Minimizing Torque and Current Ripples in PMSM-Driven

Electric Vehicle. *Electrical Engineering*. 106: 5897–5907. https://doi.org/10.1007/s00202-024-02346-3.

[18] Abraham, D. M., and A. Harish. 2024. Speed Regulation of PMSM Drive in Electric Vehicle Applications with Sliding Mode Controller Based on Harris Hawks Optimization. *e-Prime: Advances in Electrical Engineering, Electronics and Energy*. 9: 100643. https://doi.org/10.1016/j.prime.2024.100643.

[19] Abd Ali, J., M. A. Hannan, and A. Mohamed. 2016. Improved Indirect Field-Oriented Control of Induction Motor Drive Based on PSO Algorithm. *Jurnal Teknologi (Sciences & Engineering)*. 78(6–2). https://doi.org/10.11113/jt.v78.8888.

[20] Veeresh, M. Y., V. N. B. Reddy, and R. Kiranmayi. 2024. Performance of PMSM-Based Battery Electric Vehicle Using PI, PID, and FL Controllers. *Journal of Applied Science and Engineering*. 27: 2917–25. https://doi.org/10.6180/jase.202407_27(7).0015.

[21] Elhaj, A., M. Alzayed, and H. Chaoui. 2025. Direct Voltage MTPA Control of Interior Permanent Magnet Synchronous Motor-Driven Electric Vehicles. *Control Engineering Practice*. 156: 106197. https://doi.org/10.1016/j.conengprac.2024.106197.

[22] Muazzam, H., M. K. Ishak, A. Hanif, and A. I. Bhatti. 2022. Compensating Thermal-Derated Torque of IPMSM-Centric Electric Vehicles. *IEEE Access*. 10: 24468–80. https://doi.org/10.1109/ACCESS.2022.3155575.

[23] Ma, C., W. Song, J. Li, H. Yue, J. Rodriguez, and C. Garcia. 2024. A Direct Optimal Input Determination Data-Based Predictive Current Control for PMSM Drives without System Identification. *IEEE Journal of Emerging and Selected Topics in Power Electronics*. 12: 2707–17. https://doi.org/10.1109/JESTPE.2024.3378523.

[24] Liu, J., C. Gong, Z. Han, and H. Yu. 2018. IPMSM Model Predictive Control in Flux-Weakening Operation Using an Improved Algorithm. *IEEE Transactions on Industrial Electronics*. 65: 9378–87. https://doi.org/10.1109/TIE.2018.2818640.

[25] Khanh, P. Q., and H. P. H. Anh. 2023. Advanced PMSM Speed Control Using Fuzzy PI Method for Hybrid Power Control Technique. *Ain Shams Engineering Journal*. 14: 102222. https://doi.org/10.1016/j.asej.2023.102222.

[26] Li, T., A. Li, and L. Hou. 2025. Improved GOA-Based Fuzzy PI Speed Control of PMSM with Predictive Current Regulation. *PLOS ONE*. 20: e0318094. https://doi.org/10.1371/journal.pone.0318094.

[27] Zhou, X., H. Ma, J. Gu, H. Chen, and W. Deng. 2022. Parameter Adaptation-Based Ant Colony Optimization with Dynamic Hybrid Mechanism. *Engineering Applications of Artificial Intelligence*. 114: 105139. https://doi.org/10.1016/j.engappai.2022.105139.

[28] Skackauskas, J., T. Kalganova, I. Dear, and M. Janakiram. 2022. Dynamic Impact for Ant Colony Optimization Algorithm. *Swarm and Evolutionary Computation*. 69: 100993. https://doi.org/10.1016/j.swevo.2021.100993.

[29] Basir, M. S. S. M., A. R. Abdullah, and N. M. Saad. 2019. A Comparative Study on Spectrogram and S-Transform for Batteries Parameters Estimation. *Jurnal Teknologi (Sciences & Engineering)*. 81(2). https://doi.org/10.11113/jt.v81.12801.

[30] Miniguano, H., A. Barrado, and A. Lázaro. 2021. Li-Ion Battery and Supercapacitor Modeling for Electric Vehicles Based on Pulse-Pseudo Random Binary Sequence. *IEEE Transactions on Vehicular Technology*. 70: 11378–89. https://doi.org/10.1109/TVT.2021.3112284.

[31] Mehraban, A., E. Farjah, T. Ghanbari, and L. Garbuio. 2023. Integrated Optimal Energy Management and Sizing of Hybrid Battery/Flywheel Energy Storage for Electric Vehicles. *IEEE Transactions on Industrial Informatics*. 19: 10967–76. https://doi.org/10.1109/TII.2023.3242580.

[32] Valdez-Resendiz, J. E., J. C. Rosas-Caro, J. C. Mayo-Maldonado, A. Claudio-Sanchez, O. Ruiz-Martinez, and V. M. Sanchez. 2020. Improvement of Ultracapacitors-Energy Usage in Fuel Cell-Based Hybrid Electric Vehicle. *International Journal of Hydrogen Energy*. 45: 13746–56. https://doi.org/10.1016/j.ijhydene.2019.12.201.

[33] Zaihidee, F. M., S. Mekhilef, and M. Mubin. 2019. Application of Fractional-Order Sliding Mode Control for Speed Control of Permanent Magnet Synchronous Motor. *IEEE Access*. 7: 101765–74. https://doi.org/10.1109/ACCESS.2019.2931324.

[34] Yang, T., Y. Deng, H. Li, Z. Sun, H. Cao, and Z. Wei. 2023. Fast Integral Terminal Sliding Mode Control with a Novel Disturbance Observer Based on Iterative Learning for Speed Control of PMSM. *ISA Transactions*. 134: 460–71. https://doi.org/10.1016/j.isatra.2022.07.029.

[35] Wang, H., S. Xu, and H. Hu. 2023. PID Controller for PMSM Speed Control Based on Improved Quantum Genetic Algorithm Optimization. *IEEE Access*. 11: 61091–102. https://doi.org/10.1109/ACCESS.2023.3284971.

.

Appendix

Table 5 Summary of response times under different load conditions

Load torque (Nm)	Motor speed (rpm)	Response time (ms) PI	fuzzy PI	ACO
-30	300	11	12	29
	500	15	16	31
	800	21	22	33
	1200	29	29	38
No-load	300	12	13	30
	500	17	18	32
	800	23	23	35
	1200	31	31	39
30	300	13	15	33
	500	18	19	34
	800	25	25	36
	1200	33	34	41

Table 6 Summary of settling times under different load conditions

Load torque (Nm)	Motor speed (rpm)	Settling time (ms) PI	fuzzy PI	ACO
-30	300	26	63.6	50
	500	31.2	65	86
	800	36	59	63
	1200	45	70	71
No-load	300	25	67	56
	500	34	69	58
	800	37	58	56
	1200	48	37	64
30	300	63	57	59
	500	66	43	68
	800	36	32	63
	1200	48	42	72

Table 7 IPMSM 50kW and EV parameter

Parameter name	Symbol	Value	Unit
Rated power	P	50	kW
Maximum torque	M_{max}	400	Nm
Number of pole pairs	$2p$	4	-
Rotor moment of inertia	J_r	0.09	kg*m²
Stator resistance	R_s	0.05	Ω
d-axis inductance	L_{ds}	1.59	mH
q-axis inductance	L_{qs}	2.05	mH
Magnetic flux	ψ_m	0.175	Wb
Gear ratio	G	-	2
Front bumper area	A	m²	2.2
Drive efficiency	η_g	-	0.9
Aerodynamic drag coefficient	C_d	-	0.26