scientific reports

Check for updates

OPEN

Hybrid adaptive ant lion optimization with traditional controllers for driving and controlling switched reluctance motors to enhance performance

Mostafa Jabari[1], Davut Izci[2,3,4], Serdar Ekinci[5], Mohit Bajaj[6,7,8]✉, Vojtech Blazek[9] & Lukas Prokop[9]

Switched reluctance motors (SRMs) are favored in industrial applications for their durability, efficiency, and cost-effectiveness, yet face challenges such as torque ripple and nonlinear magnetic behavior that limit their precision in control tasks. To address these issues, this work introduces a novel hybrid adaptive ant lion optimization (HAALO) algorithm, combined with PI and FOPID controllers, to improve SRM performance. The HAALO algorithm enhances traditional ant lion optimization by integrating adaptive mutation and elite preservation techniques for dynamic real-time control, optimizing both torque ripple and speed regulation. Simulation results demonstrate the superiority of the HAALO-optimized controllers over conventional methods, showing faster convergence and enhanced control accuracy. This study provides a new hybrid optimization method that significantly advances SRM control, offering efficient solutions for high-performance applications.

Keywords Switched reluctance motors, Hybrid adaptive optimization, HAALO algorithm, PI controller, FOPID controller, Torque ripple minimization

Abbreviations

SRM	Switched reluctance motor
FTSMC	Fast terminal sliding mode control
TSF	Torque sharing functions
FLC	Fuzzy logic control
Hb	Hysteresis band
ITAE	Integral of time-weighted absolute error
ITSE	Integral of time-weighted square error
HAALO	Hybrid adaptive ant lion optimization
FPA	Flower pollination algorithm
PI	Proportional integral
PID	Proportional integral derivative
PID*n*	Proportional integral derivative with derivative filter
MSACO	Multiple stage ant colony optimization
FOPI	Fractional order proportional integral
r.p.m	Revolutions per minute
FOPID	Fractional order proportional integral derivative

[1]Faculty of Electrical Engineering, Sahand University of Technology, Tabriz, Iran. [2]Department of Electrical and Electronics Engineering, Bursa Uludag University, 16059 Bursa, Turkey. [3]Applied Science Research Center, Applied Science Private University, Amman 11931, Jordan. [4]Jadara University Research Center, Jadara University, Irbid, Jordan. [5]Department of Computer Engineering, Batman University, Batman, Turkey. [6]Department of Electrical Engineering, Graphic Era (Deemed to Be University), Dehradun 248002, India. [7]College of Engineering, University of Business and Technology, 21448 Jeddah, Saudi Arabia. [8]Graphic Era Hill University, Dehradun 248002, India. [9]ENET Centre, CEET, VSB-Technical University of Ostrava, 708 00 Ostrava, Czech Republic. ✉email: mohitbajaj.ee@geu.ac.in

IAE	Integral of absolute error
ISE	Integral of square error
DITC	Direct torque control
LUS	Local unimodal sampling
ACO	Ant colony optimization
SHO	Spotted hyena optimization
PSO	Particle swarm optimization
FOPI(1 + PIDn)	Fractional order proportional integral one plus proportional integral derivative with derivative filter
PSO-TVAC	Particle swarm optimization with time-varying acceleration coefficients
OF	Objective function
OS	Overshoot

List of symbols

V_{in}	Input voltage
R_s	Resistance of per-phase stator
ω_m	Motor's angular velocity
T_L	Load torque
K_{PC}, K_{PS}	Proportional gains of cascaded controller
K_{DC}, K_{DS}	Derivative gains of cascaded controller
Hb_{up}	Upper hysteresis band
I_{ref}	Reference current
I_{Ph}	Per-phase current
T_r	Rise time
T_s	Settling time
T_p	Peak time
β_s, β_r	Rotor and stator pole arcs
θ_r	Start angle of positive inductance region
θ_s	Angle at which inductance is maximum
L_{max}, L_{min}	Minimum and maximum inductance
x	Simulation time in second
i_s	Current in the stator winding
θ	Rotor position
J	Moment of inertia
F_ω	Coefficient of friction
ΔS	Speed error
ΔC	Current error
t_{sim}	Simulation time
e(t)	Error signal between speed and current
K_{IC}, K_{IS}	Integral gains of cascaded controller
λ, μ	Integral and derivative order
Hb_{down}	Lower hysteresis band
θ_{on}, θ_{off}	Turn on and turn off angles
ω_{ref}	Reference value for angular velocity
T_e	Motor electric torque
$\Delta\omega$	Speed deviation
P_{mut}	Mutation probability
θ_t	End angle of negative inductance region
θ_0	Zero slope angle

Switched Reluctance Motors (SRMs) have become increasingly popular in industrial applications due to their robust design, simplicity, and high efficiency. However, they face challenges such as torque ripple and acoustic noise, which arise from their inherent nonlinear magnetic properties and the nature of their switching operations[1,2]. These issues hinder the use of SRMs in applications that require smooth torque output and precise speed control, especially under varying loads[3,4].

To mitigate the issues of torque ripple and improve control accuracy, researchers have explored various control strategies for SRMs. Traditional proportional-integral-derivative (PID) controllers are widely used for their simplicity and effectiveness in maintaining stable speed control[5]. However, advanced controllers, such as fractional order PID (FOPID)[6], fuzzy logic control (FLC)[7], and fast terminal sliding mode control (FTSMC) [8], have also been implemented to handle SRM's nonlinear behavior and provide better performance in high-torque and high-speed applications. Additionally, torque sharing functions (TSF) specifically address torque ripple by distributing torque smoothly across phases[9]. The versatility of these controllers, from the cascaded PID and FOPID to fuzzy-based configurations, highlights the diversity of approaches aimed at optimizing SRM control for enhanced stability and reduced ripple[10,11].

Optimization algorithms play a vital role in tuning controller parameters to achieve optimal performance in SRM systems. Traditional optimization methods such as particle swarm optimization (PSO)[12] and genetic algorithms (GA)[6] have been successfully applied to SRM control, specifically for improving speed regulation. More recent algorithms, including ant colony optimization (ACO)[13] and multiple stage ant colony optimization (MSACO)[14], have focused on minimizing torque ripple and enhancing control precision. The dung beetle optimizer (DBO)[12] and ant lion optimizer (ALO)[15] have shown significant improvements in convergence

speed and computational efficiency. Additionally, bio-inspired algorithms like the flower pollination algorithm (FPA)[16] and the spotted hyena optimizer (SHO)[6], as well as techniques like local unimodal sampling (LUS)[6], have been utilized to tune cascaded PID controllers, demonstrating the evolution of intelligent algorithms in SRM optimization.

Several studies have sought to address the control and performance challenges of SRMs by integrating advanced optimization techniques and control strategies. Al-Amyal et al.[14] utilized multiple stage ant colony optimization (MSACO) to optimize switching angles in direct instantaneous torque control, which reduced torque ripple and enhanced motor efficiency. Their method included a current detector to minimize negative torque, achieving a smoother torque profile. da Cunha Reis et al.[17] explored different control techniques for SRMs by modulating excitation voltage and switching angles. They demonstrated that dynamic control parameters improved efficiency and response time while maintaining low computational costs, particularly in speed control applications. Divandari et al.[18] proposed a fuzzy logic-based FTSMC, combining robustness and reduced chattering effects. Their results showed improved speed stability and faster response times compared to conventional PI controllers, making FTSMC suitable for applications with parameter uncertainties. Gengaraj et al.[19] employed the FPA with a TSF to minimize torque ripple. By optimizing commutation angles, their method effectively distributed torque across motor phases, making it particularly beneficial for applications that demand smooth torque output.

Kotb et al.[6] implemented LUS and the SHO to tune cascaded PID controllers for SRMs. Their results showed that the SHO-based cascaded PID controller reduced torque ripple more effectively than the FOPID controller, providing enhanced speed response across varying load conditions. Rajendran and Karthik[20] compared PI and FLCs, finding that while PI controllers ensured reliable speed control, FLCs were better suited for handling SRM's nonlinear dynamics and torque control. Jabari and Rad[12] applied the DBO and ALO to optimize PID and FOPID controllers, which led to reduced torque ripple and improved speed control. Their study highlighted the benefits of bio-inspired optimization methods in refining SRM performance. The literature review summary is provided in Table 1.

To effectively controls the SRM, an optimized method for identifying the switching angles is essential for minimizing torque ripple and ensuring smooth operation[22]. In this study, we use a combination of proportional-integral (PI) and fractional-order proportional-integral-derivative (FOPID) controllers, integrated with the hybrid adaptive ant lion optimization (HAALO) algorithm. The HAALO algorithm dynamically adjusts the switching angles in real-time, optimizing their values based on the motor's operating conditions to achieve better performance. The switching angles determine the timing for current switching between the motor's phases, and their optimization is crucial for reducing torque ripple and enhancing stability. The SRM is powered by a full-bridge converter, which regulates the current delivered to the motor's phases and ensures the appropriate voltage levels are maintained. This converter type was selected for its precision in controlling phase currents and ensuring stable operation across varying speeds and loads, which complements the performance of the proposed control method.

This paper presents the HAALO algorithm integrated with PI and FOPID controllers to improve SRM performance. The main contributions are as follows:

1. *Fastest and optimized parameters with HAALO*: HAALO continuously adjusts optimization parameters, enhancing SRM control by adapting to diverse load conditions and achieving more effective torque ripple reduction than conventional methods.

Article	Controller	Optimization Method	Advantages	Disadvantages	Key Findings
Al-Amyal et al.[14]	DITC	Multistage ant colony optimization (MSACO)	Effective torque ripple reduction; improved efficiency	Complex implementation of MSACO	MSACO improved torque ripple and performance by optimizing switching angles
da Cunha Reis et al.[17]	PID with dynamic parameters	Modulation of excitation voltage and switching angles	Enhanced efficiency and reduced computational cost	Requires precise control parameter tuning	Dynamic control parameters improved response time for speed control applications
Divandari et al.[18]	Fuzzy logic-based fast terminal sliding mode control (FTSMC)	None	Improved speed stability; reduced chattering	Limited to specific disturbances and conditions	FTSMC provided robust control under uncertainty, reducing chattering effects
Gengaraj et al.[19]	Torque sharing function (TSF)	Flower pollination algorithm (FPA)	Smooth torque distribution; reduced vibration	Dependent on proper TSF selection for varied speeds	FPA with TSF effectively minimized torque ripple across a range of speeds
Kotb et al.[6]	Cascaded PID	Local unimodal sampling (LUS) and spotted hyena optimizer (SHO)	Better speed response; lower torque ripple	Higher computational complexity than basic PID	SHO-based cascaded PID showed superior speed control and torque ripple reduction
Rajendran and Karthik[20]	PI and fuzzy logic	None	Reliable speed control; adaptable to nonlinear dynamics	Limited adaptability compared to newer methods	PI was reliable, but fuzzy logic offered better adaptability to SRM's nonlinear nature
Jabari and Rad[12]	PID and FOPID	Dung beetle optimizer (DBO) and ant lion optimizer (ALO)	Reduced torque ripple; faster response	Requires bio-inspired algorithm tuning	DBO and ALO significantly enhanced PID and FOPID performance in SRMs
Mostafa Jabari[21]	Multistage FOPI(1 + PIDn)	PSO-TVAC algorithm	Reduces torque ripple, enhances response speed, and improves tracking accuracy in a short time	Requires multiple controller parameters to achieve a good response	Proposed a multistage FOPI(1 + PIDn) controller for better performance in SRMs

Table 1. Overview of utilizing controller and optimization method for SRM in the recent papers.

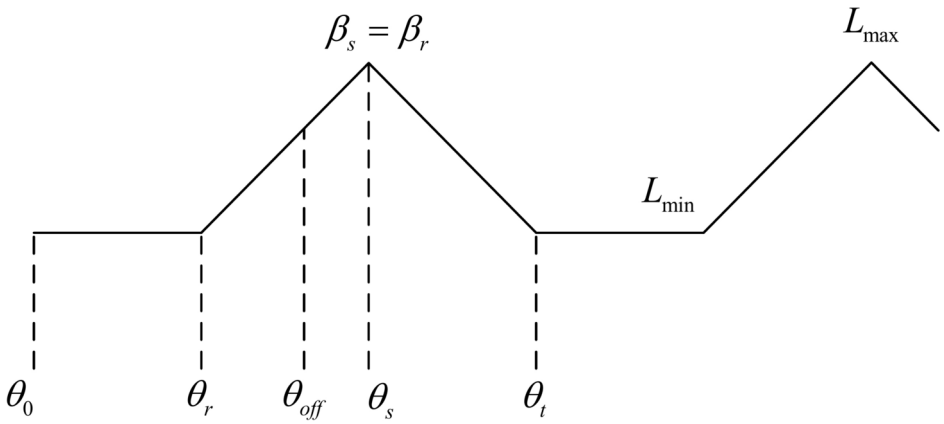


Fig. 1. Per-phase inductance (linear) profile.

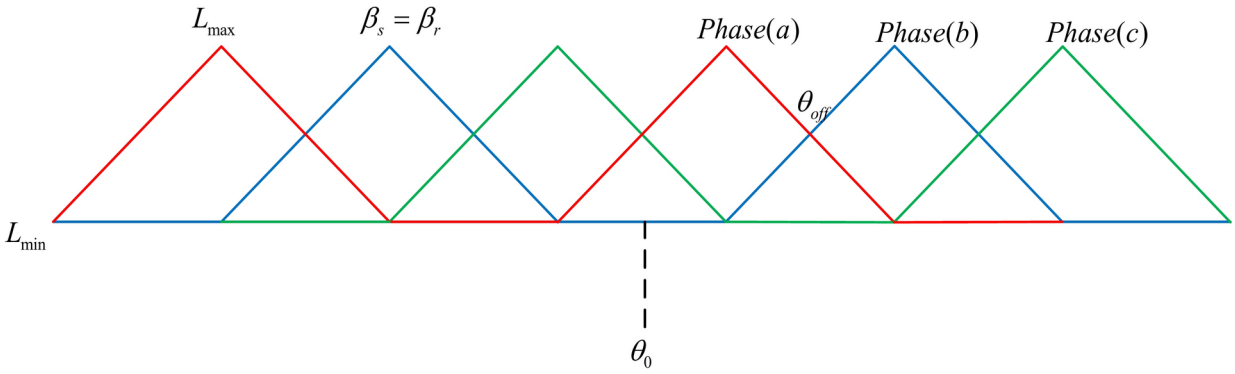


Fig. 2. Inductance (linear) profile of all three phase.

SRM parameters	Input voltage	SRM power	Reference speed	Turn off angle	Turn on angle	Stator resistance of SRM	Inertia of SRM	Friction of SRM
Values	240 V_{dc}	60 kW	2000 r.p.m	75°	45°	0.05 Ω	0.05 kg m^2	0.02 N.m. s

Table 2. Parameters used in case study.

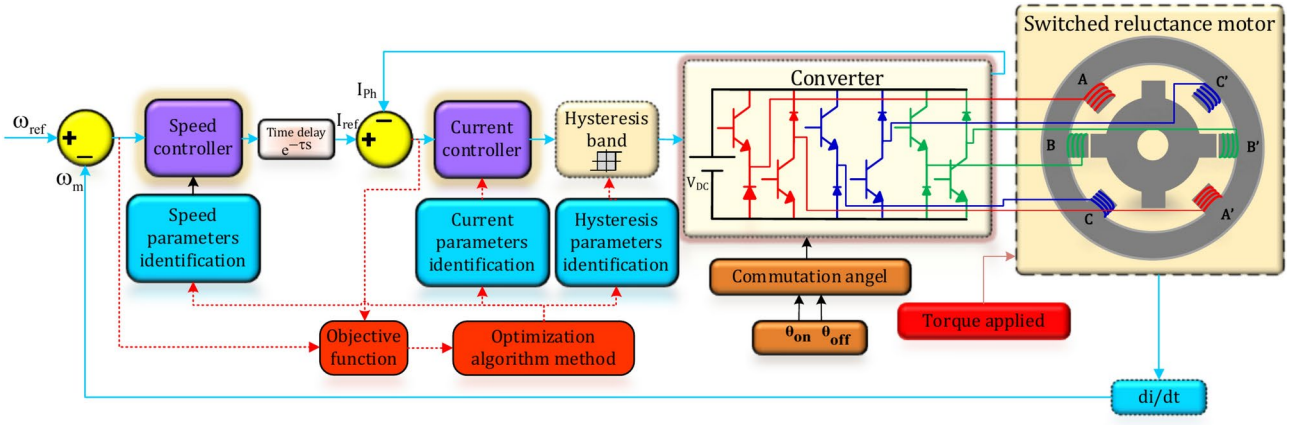


Fig. 3. General schematic of proposed controller and SRM system.

2. *Enhanced PI and FOPID controllers*: By leveraging HAALO, the PI and FOPID controllers achieve superior speed regulation and torque stability, accommodating SRM's nonlinear characteristics and outperforming standard PI and FOPID setups.
3. *Improved torque ripple and speed response*: Simulation results show that HAALO-optimized PI and FOPID controllers provide significant torque ripple reduction and enhanced speed stability, surpassing algorithms such as PSO, PPSO, GEO, GA, ALO, and RIME-optimized PID and FOPID controllers.
4. *Robust performance across operating conditions*: The adaptive PI and FOPID controllers optimized with HAALO maintain efficient SRM control under various speeds and loads, making them suitable for dynamic industrial applications.

The paper is organized as follows: "Mathematical model of the switched reluctance motor" section addresses the modeling of the SRM, with a thorough explanation of its nonlinear characteristics. In "Optimization" section, we introduce the design and development of both the ALO and the HAALO methods. "Results of the simulation and analysis" section goes into detail on the HAALO method, emphasizing its effectiveness for parameter tuning. "Comparison in different scenarios" section presents simulation results, comparing the performance of HAALO with other commonly used controllers and meta-heuristic optimization techniques. Lastly, "Conclusions" section concludes with a summary of key insights and suggestions for future research directions.

Mathematical model of the switched reluctance motor

The mathematical model of the SRM describes the complex interaction between electrical inputs, magnetic characteristics, and mechanical outputs, enabling us to understand and control the motor's operation. At the core of SRM modeling are three main relationships: the voltage applied across the windings, the flux linkage in response to the rotor's position, and the resulting torque generated by electromagnetic forces.

The voltage equation for each SRM phase describes how applied voltage is divided between resistive losses and the time-dependent changes in the flux linkage. This equation can be expressed as:

$$Vin = Rs \times i + \frac{d\lambda\left(\theta, i\right)}{dt} \tag{1}$$

where Vin is the phase voltage, Rs is the winding resistance, i is the phase current, and $\lambda\left(\theta, i\right)$ represents the flux linkage. This flux linkage depends on both the rotor's angular position, θ, and the phase current, i, due to the nonlinearity arising from magnetic saturation. To express the rate of change in flux linkage, the derivative can be expanded to account for both rotor position and current change over time, applying the chain rule as follows:

$$\frac{d\lambda\left(\theta, i\right)}{dt} = \frac{\delta\lambda}{\delta\theta} \times \frac{d\theta}{dt} + \frac{\delta\lambda}{\delta i} \times \frac{di}{dt} \tag{2}$$

Here, $\frac{d\theta}{dt} = \omega$ represents the rotor speed, connecting flux changes directly to the motion of the rotor. This expanded form is critical for dynamic simulations and SRM control, as it reflects the intertwined effects of electrical input and mechanical motion. The electromagnetic torque generated by the SRM arises from the tendency of the rotor to align with the highest magnetic field, a torque production mechanism distinct from conventional motors. This torque can be computed as $T_e = i \times \frac{\delta\lambda(\theta,i)}{\delta\theta}$, linking it to the co-energy of the system.

The electromagnetic torque depends on both the current and the rate of change in inductance relative to rotor position, which varies as the rotor rotates through alignment with the stator poles. The mechanical dynamics of the SRM's motion can be described by Newton's second law for rotation:

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

Here, J represents the rotor's moment of inertia, B is a friction coefficient, ω denotes rotor speed, and T_L represents the load torque. This equation expresses how the net torque, a combination of electromagnetic and load torque, affects the rotor's acceleration and speed. Together, this set of equations represents a highly nonlinear system. Another important aspect of SRM modeling is the inductance profile of each phase winding. The inductance varies as the rotor position changes, introducing nonlinearity. The flux linkage can thus be modeled as $d\lambda\left(\theta, i\right) = L\left(\theta\right) \times i$, or by using a look-up table to capture magnetic saturation effects accurately. This approach helps account for the unique inductance values encountered at different rotor positions.

Figure 1 shows a linear profile for the per-phase inductance in an SRM motor. Figure 2 shows Inductance (linear) profile of all three phase. Table 2 presents the parameters used for SRM modeling in this study, while Fig. 3 provides an overview of the SRM and the controllers analyzed[12,21].

SRM structure and implementation controller

Converter structure

As showed in Fig. 3 proposed system uses a three-phase asymmetric bridge converter to drive the SRM. This converter consists of two switches and two diodes per phase. These two switches can be IGBTs or MOSFETs. When a phase needs to be energized, the two switches turn on, allowing current to flow from the DC power source into the motor winding. This creates a magnetic field that pulls the rotor towards the aligned position. When the switches turn off, the energy stored in the winding is released through the diodes, ensuring smooth current decay and reducing voltage spikes. This process is repeated for each phase, ensuring continuous rotation of the motor.

Identification of switching angles

The switching angles are determined based on rotor position feedback. The system uses a position sensor to detect the rotor's current position and decide the best moment to energize and de-energize each phase. The HAALO optimization method is used to fine-tune input current in converter for improving efficiency and reducing torque ripple. Inner and outer controllers help the motor achieve better performance and smooth current regulation. This smooth current flow ensures more stable torque generation, reduces sudden variations

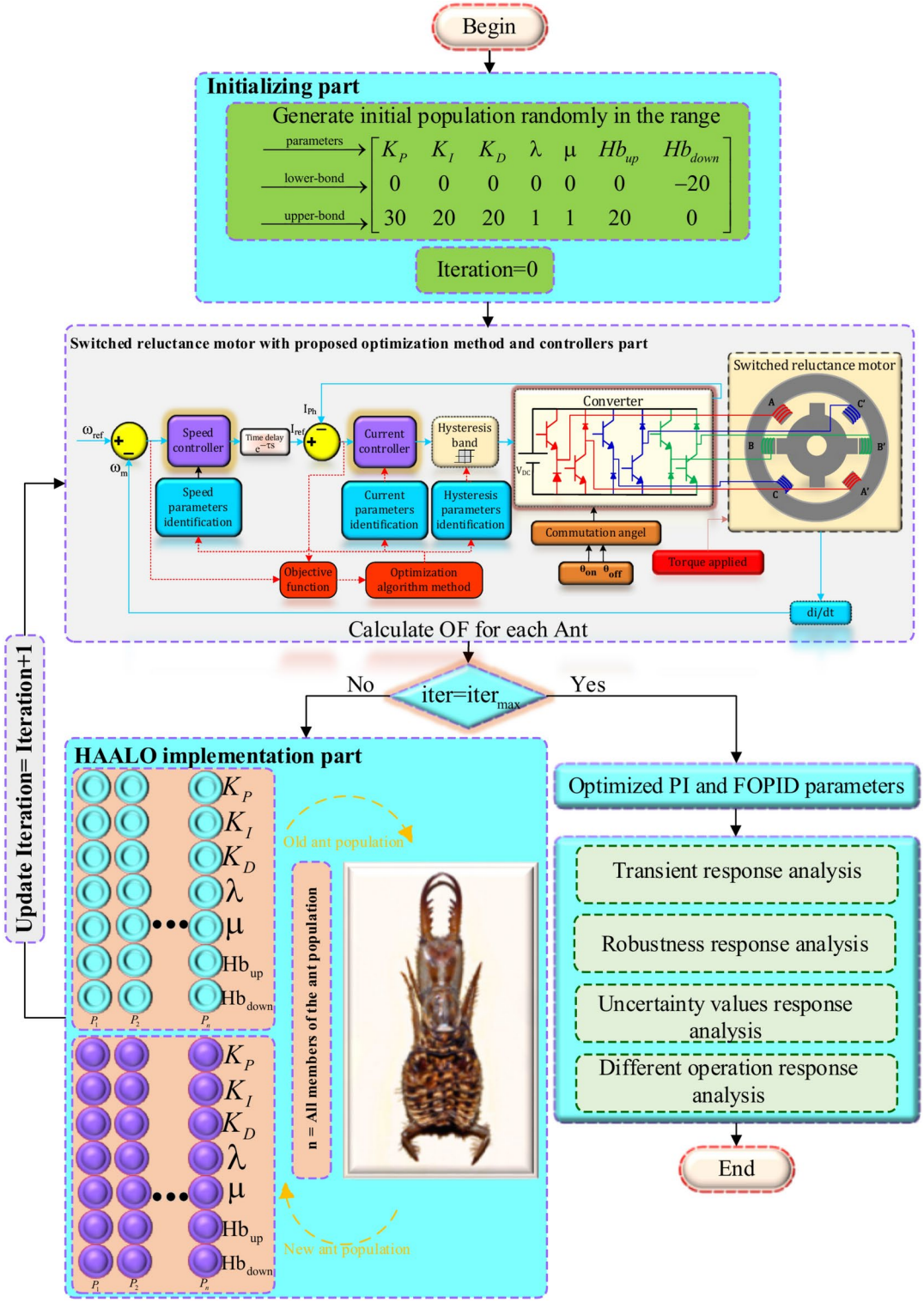


Fig. 4. Flowchart of HAALO Implementation for the SRM System.

in motor operation. Additionally, maintaining a well-regulated current improves switching transitions, reducing stress on power electronics components and enhancing the overall reliability of the system.

Control method
The system uses a cascaded control structure with two loops:

- *Outer loop*: This controller compares the desired speed ω_{ref} with the actual motor speed ω_m and generates a reference current. The HAALO optimization method helps adjust the speed controller parameters for better performance.
- *Inner loop*: The inner loop ensures that the motor phase currents follow the reference current generated by the speed controller. The HAALO optimization method helps adjust the current controller and hysteresis band parameters for better performance.

This cascaded control structure ensures precise speed regulation and reduced torque ripple, making the system more efficient and stable.

Optimization

Ant-lion optimization (ALO)

The ant lion optimization (ALO) algorithm is a bio-inspired metaheuristic introduced by[23]. It mimics the predatory behavior of antlions in nature, where antlions capture ants by trapping them in funnel-shaped sand pits. In the ALO algorithm, ants represent potential solutions, and antlions represent elite solutions guiding the search process. The algorithm alternates between exploration and exploitation, making it a powerful tool for solving complex optimization problems.

Exploration
Random walk of ants In ALO, the ants' movements are modeled as random walks across the search space, representing the exploration phase. Each ant performs a stochastic exploration defined by the equation:

$$x_i\left(t+1\right) = x_i\left(t\right) + r\left(t\right) \tag{4}$$

where $x_i\left(t\right)$ is the position of the ith ant at iteration t, and $r(t)$ is a random walk vector. Each element of $r(t)$ is generated using a stochastic process defined by:

$$r_j\left(t\right) = \text{cumulative sum of random steps}\left(2 \times \text{rand}\left(1\right) - 1\right) \tag{5}$$

where, rand (1) generates random numbers uniformly distributed in the range [0,1]. The search space is bounded by lower and upper limits l_b and u_b, ensuring that ants remain within feasible regions:

$$x_i\left(t\right) \in \left[l_b, u_b\right] \tag{6}$$

The random walk of ants provides the exploration capability, allowing the algorithm to search diverse regions of the solution space.

Exploitation
Trap building by antlions As the ants explore the search space, they are influenced by antlions, which represent elite solutions guiding the ants towards better regions. During each iteration, antlions create traps, which metaphorically pull the ants towards optimal solutions. The process of trapping is modeled by narrowing the boundaries of the search space around the ants as they approach antlions, mimicking the natural hunting behavior of antlions. The elite antlions influence the ants' movements by dynamically adjusting their search regions, encouraging exploitation of promising areas in the solution space. The algorithm incorporates a roulette wheel selection mechanism to favor fitter antlions, which further ensures that the best solutions are more likely to guide the search process.

Balance between exploration and exploitation
ALO is designed to strike a balance between exploration and exploitation. In the early stages of the algorithm, the ants are allowed to explore the search space broadly to identify potential solutions. As the iterations progress, the focus shifts toward exploitation, refining the best solutions found so far. This balance is crucial in preventing premature convergence to local optima while ensuring that the algorithm converges to the global optimum over time. The ALO algorithm continues this process until a stopping criterion, such as a maximum number of iterations, is met. The best solution found by the antlions is then reported as the global optimum.

Hybrid adaptive ant lion optimization (HAALO)

To further enhance the performance of the standard ALO algorithm, a hybrid adaptive ant lion optimization (HAALO) approach has been developed. HAALO introduces several adaptive and hybrid techniques to address limitations such as slow convergence or premature stagnation in local optima. These enhancements make the algorithm more robust, especially when dealing with highly complex and multimodal optimization problems. Figure 4 depicts flowchart of HAALO Implementation for the SRM system.

1. Initialize parameters of ants:
a. Define objective function
b. Set Dim (Number of dimension)
c. set L_b and U_b (Lower and upper bounds for each dimension)
d. Define algorithm parameters:
i. nAntlions= Number of antlions
ii. maxIter= Maximum number of iterations
iii. mutationProb= Mutation probability
iv. Mutation probability= Fraction of elite antlions preserved
2. Initialize population:
a. Generate random positions within bounds for each antlion
b. Set fitness of each antlion to infinity
3. Evaluate initial population:
a. For each antlion *x* from 1 to *nAntlions*:
i. Compute fitness
4. Store best initial solution:
a. Find minimum fitness and corresponding index
b. Set *bestPos* to the position of the antlion with the best fitness
5. Optimization loop:
a. Calculate population diversity:
i. If diversity > threshold (*Th* = 0.5)
- Set *alpha* for exploitation (α):
alpha = 2 × (1 - iter / maxIter) ^{2}
- Reduce inertia weight (ω):
ω *= 0.5 + (iter / maxIter) × 0.4*
b. Else (diversity ≤ threshold):
- Set α for exploration:
alpha = 2 × rand ()
- Increase inertia weight (ω):
ω *= 0.9 - (iter / maxIter) × 0.5*
c. Preserve elite antlions:
i. Identify indices of elite antlions
ii. Retain their positions for next generation
d. Mutation and position update:
i. For each antlion *x*:
- Update position based on adaptive weights (α and ω) and diversity
- If mutation condition (*random* < *mutationProb*) is met, apply mutation to position
e. Evaluate updated positions:
i. Compute fitness for each new position
ii. Update *bestFitness* and *bestPos* if a new position improves the fitness
6. Output Results
a. Return *bestPos* and *bestFitness* after all iterations

Table 3. Pseudocode for proposed optimization method (HAALO).

Mutation-based exploration
To increase diversity and prevent the algorithm from getting trapped in local optima, HAALO introduces a mutation operator. Mutation introduces random perturbations in the ants' positions with a probability P_{mut}:

$$x_i(t+1) = \begin{cases} x_i(t) + \delta \times rand(n), & if rand(1) < P_{mut} \\ x_i(t) & otherwise \end{cases} \quad (7)$$

where δ is a small perturbation factor, and P_{mut} is the mutation probability. The mutation ensures that the search process continues exploring new regions of the solution space, especially in later iterations when exploitation becomes dominant.

Elite preservation strategy
HAALO incorporates an elite preservation mechanism, ensuring that the best-performing solutions are retained throughout the optimization process. A fraction α of the population, representing elite solutions, is preserved at each iteration:

$$n_{\text{elite}} = [\alpha \times n_{population}] \quad (8)$$

The elite solutions guide the ants during exploitation, preventing the loss of high-quality solutions due to randomness or mutation. Elite solutions are selected based on their fitness and used to refine the search in subsequent iterations.

Function	Metric	HAALO	PSO	PPSO	GA	ALO	RIME	GEO	MSACO	DBO	FPA	SHO	LUS
F22-01	Minimum	604.88	734.92	676.92	744.46	718.72	663.18	696.42	695.3	690.96	714.75	716.92	650.79
	Mean	618.81	753.19	706.19	760.22	745.52	684.98	711.17	724.93	704.36	726.47	731.08	673.18
	Std Dev	9.18	10.66	11.19	8.96	6.43	7.98	7.15	6.04	8.4	9.76	6.89	7.74
	Rank	1.0	11.0	4.0	12.0	10.0	3.0	7.0	6.0	5.0	8.0	9.0	2.0
F22-02	Minimum	645.48	650.53	746.85	744.78	717.87	745.62	727.57	739.22	728.3	729.55	711.02	691.74
	Mean	668.79	665.4	772.78	754.97	731.34	757.09	742.55	762.03	739.08	750.3	722.44	710.05
	Std Dev	11.24	13.68	5.27	14.94	14.1	5.89	14.86	12.66	13.33	9.09	10.43	6.76
	Rank	1.0	2.0	12.0	10.0	5.0	11.0	6.0	9.0	7.0	8.0	4.0	3.0
F22-03	Minimum	635.87	707.4	666.23	709.56	742.86	679.62	671.72	747.08	706.25	671.05	704.64	653.83
	Mean	649.02	723.42	685.04	720.92	756.57	705.44	697.18	757.63	722.19	695.99	718.65	676.78
	Std Dev	9.3	8.91	9.3	7.66	10.58	13.84	13.82	6.64	6.03	5.75	7.96	11.89
	Rank	1.0	9.0	3.0	10.0	11.0	6.0	5.0	12.0	8.0	4.0	7.0	2.0
F22-04	Minimum	609.07	688.84	686.42	691.78	701.92	741.24	699.87	704.05	667.36	727.39	711.78	722.57
	Mean	628.85	717.8	715.83	717.44	728.87	763.53	721.11	726.86	680.85	744.58	739.34	735.2
	Std Dev	6.07	7.31	7.2	13.38	7.16	8.9	11.0	7.48	13.65	11.53	9.5	8.78
	Rank	1.0	4.0	3.0	5.0	7.0	12.0	6.0	8.0	2.0	11.0	9.0	10.0
F22-05	Minimum	642.35	704.88	692.24	689.49	740.97	672.92	654.64	735.47	722.24	656.56	732.43	704.97
	Mean	657.64	726.86	718.78	713.41	768.79	698.14	671.69	753.21	737.46	686.43	751.71	715.16
	Std Dev	10.12	12.98	11.51	7.79	8.97	9.67	14.48	14.33	11.8	7.44	7.12	7.88
	Rank	1.0	7.0	6.0	5.0	12.0	4.0	2.0	11.0	9.0	3.0	10.0	8.0
F22-06	Minimum	628.52	676.2	748.91	745.71	661.83	686.99	742.57	709.83	691.66	714.07	730.58	656.64
	Mean	647.37	703.45	768.1	761.91	678.95	709.86	755.07	720.67	719.07	732.0	748.95	678.6
	Std Dev	8.28	12.18	5.0	10.72	14.6	13.34	10.71	6.89	12.59	11.08	7.56	5.96
	Rank	1.0	4.0	12.0	11.0	3.0	5.0	10.0	7.0	6.0	8.0	9.0	2.0
F22-07	Minimum	601.81	665.43	747.95	658.23	746.79	673.53	732.43	701.01	672.44	687.32	726.65	671.75
	Mean	628.03	679.31	764.72	683.9	761.39	685.02	745.7	726.14	698.08	700.42	739.42	682.33
	Std Dev	10.34	12.41	6.82	6.02	9.52	8.41	9.72	12.83	14.34	14.58	14.46	11.43
	Rank	1.0	3.0	12.0	2.0	11.0	6.0	10.0	8.0	5.0	7.0	9.0	4.0
F22-08	Minimum	602.73	667.89	713.55	742.83	719.62	735.54	724.49	736.75	712.66	731.57	749.71	720.53
	Mean	614.2	691.11	734.75	753.05	747.16	759.04	751.27	755.44	740.88	756.09	779.03	741.55
	Std Dev	6.22	11.37	12.73	14.44	5.94	8.09	11.27	7.75	10.0	12.72	7.39	13.68
	Rank	1.0	2.0	4.0	11.0	5.0	9.0	7.0	10.0	3.0	8.0	12.0	6.0
F22-09	Minimum	631.25	683.73	681.41	731.92	696.4	705.25	660.72	717.33	697.38	651.27	665.31	720.57
	Mean	643.71	701.92	698.25	758.63	719.78	715.63	685.64	747.29	724.41	672.13	675.56	738.93
	Std Dev	9.6	9.55	5.59	9.84	9.61	13.35	12.54	9.94	6.79	6.54	7.31	13.6
	Rank	1.0	6.0	5.0	12.0	7.0	9.0	3.0	10.0	8.0	2.0	4.0	11.0
F22-10	Minimum	627.92	740.11	737.08	662.52	735.79	710.34	705.96	736.27	663.82	721.1	727.21	720.97
	Mean	648.24	762.99	753.94	691.76	761.54	732.6	734.82	747.81	675.39	745.93	753.15	734.92
	Std Dev	8.82	8.39	6.78	14.75	8.86	11.28	10.15	5.88	13.77	8.3	5.09	14.63
	Rank	1.0	12.0	11.0	2.0	9.0	5.0	4.0	10.0	3.0	7.0	8.0	6.0
F22-11	Minimum	619.43	732.17	674.5	718.27	737.34	677.4	743.82	746.33	652.9	707.75	709.67	715.57
	Mean	638.25	760.79	691.49	736.68	765.42	693.03	759.69	763.94	677.13	732.77	722.36	732.2
	Std Dev	11.92	9.73	5.17	7.05	9.38	11.3	14.34	12.74	10.69	6.22	6.97	8.23
	Rank	1.0	9.0	3.0	8.0	10.0	4.0	11.0	12.0	2.0	5.0	6.0	7.0
F22-12	Minimum	634.69	665.83	718.33	717.03	712.47	721.98	673.03	656.69	658.29	710.04	678.06	727.55
	Mean	658.65	695.26	730.49	746.09	722.6	732.14	698.32	669.43	688.14	736.96	691.46	743.66
	Std Dev	11.18	6.03	8.62	10.08	14.36	5.24	8.2	8.97	14.77	10.58	8.31	14.58
	Rank	1.0	4.0	10.0	9.0	8.0	11.0	5.0	2.0	3.0	7.0	6.0	12.0
Friedman rank		12.0	73.0	85.0	97.0	98.0	85.0	76.0	105.0	61.0	78.0	93.0	73.0
Mean rank		1.0	6.08	7.08	8.08	8.17	7.08	6.33	8.75	5.08	6.5	7.75	6.08

Table 4. Comparative statistical results on CEC-2022 test suite (Dimension = 10).

Function	Metric	HAALO	PSO	PPSO	GA	ALO	RIME	GEO	MSACO	DBO	FPA	SHO	LUS
F22-01	Minimum	634.09	673.85	650.86	660.19	681.5	738.33	705.11	707.47	738.65	687.94	673.72	749.66
	Mean	657.53	692.49	669.88	671.13	700.56	768.28	732.82	720.42	766.14	715.83	687.2	764.34
	Std Dev	10.24	14.74	13.01	8.51	11.44	8.73	14.92	9.63	11.07	5.88	5.76	5.33
	Rank	1.0	5.0	2.0	3.0	6.0	10.0	8.0	9.0	11.0	7.0	4.0	12.0
F22-02	Minimum	606.38	702.42	714.1	704.34	690.76	694.06	707.46	733.17	749.01	697.24	708.28	697.49
	Mean	635.92	729.82	739.86	728.05	715.44	706.24	727.32	763.02	769.57	716.38	735.51	725.49
	Std Dev	7.98	7.1	8.44	13.99	12.03	11.08	13.27	11.87	8.75	7.31	5.49	9.51
	Rank	1.0	6.0	10.0	7.0	2.0	3.0	8.0	11.0	12.0	4.0	9.0	5.0
F22-03	Minimum	610.75	686.69	707.57	724.91	731.45	656.91	736.39	666.35	719.64	678.85	719.17	735.96
	Mean	637.82	707.63	717.75	743.62	761.3	667.83	750.35	678.62	738.37	697.64	732.27	758.57
	Std Dev	12.3	7.19	14.66	10.89	7.32	10.38	11.81	13.7	14.9	9.82	7.94	5.77
	Rank	1.0	5.0	6.0	9.0	10.0	2.0	12.0	3.0	8.0	4.0	7.0	11.0
F22-04	Minimum	638.4	729.41	720.95	692.96	694.37	738.71	670.06	714.08	747.91	688.26	677.21	743.5
	Mean	667.87	757.63	731.46	711.1	715.85	765.82	681.51	743.84	777.3	713.57	699.97	769.06
	Std Dev	8.59	5.69	13.13	9.54	11.51	5.25	11.54	9.97	10.32	11.52	5.04	11.5
	Rank	1.0	9.0	8.0	5.0	6.0	10.0	2.0	7.0	12.0	4.0	3.0	11.0
F22-05	Minimum	635.21	680.9	668.87	735.0	702.41	678.95	716.66	657.83	670.79	723.74	702.2	685.52
	Mean	661.61	707.11	694.59	748.43	712.46	702.16	743.38	669.12	682.56	735.81	722.01	697.86
	Std Dev	10.83	5.92	8.79	11.73	7.4	7.7	8.28	12.1	13.15	5.61	10.37	14.0
	Rank	1.0	6.0	3.0	12.0	9.0	5.0	10.0	2.0	4.0	11.0	8.0	7.0
F22-06	Minimum	636.72	732.13	737.91	658.83	743.95	689.22	712.67	699.58	714.05	726.89	698.67	737.57
	Mean	663.12	760.47	755.98	681.54	767.14	712.07	727.37	719.18	735.65	753.0	727.49	750.34
	Std Dev	6.5	10.19	9.96	5.85	13.24	14.29	6.07	14.77	9.7	10.58	13.77	11.83
	Rank	1.0	9.0	11.0	2.0	12.0	3.0	6.0	5.0	7.0	8.0	4.0	10.0
F22-07	Minimum	649.72	736.32	650.95	746.79	713.96	736.25	667.64	683.12	748.61	732.69	656.02	706.52
	Mean	674.85	749.77	677.68	762.8	728.61	763.1	683.89	706.21	773.15	758.24	676.83	720.76
	Std Dev	8.91	11.79	6.8	10.8	15.0	6.45	13.62	14.75	14.13	9.08	14.07	10.39
	Rank	1.0	10.0	2.0	11.0	7.0	9.0	4.0	5.0	12.0	8.0	3.0	6.0
F22-08	Minimum	610.0	700.4	668.33	748.4	702.76	736.43	729.69	656.69	707.99	672.69	727.28	679.85
	Mean	625.07	719.8	686.85	766.32	728.82	752.65	741.95	684.01	723.57	684.59	755.04	709.41
	Std Dev	14.18	14.57	14.02	14.2	13.49	12.85	8.58	14.74	9.53	11.59	5.22	13.21
	Rank	1.0	6.0	3.0	12.0	7.0	11.0	10.0	2.0	8.0	4.0	9.0	5.0
F22-09	Minimum	645.86	676.37	726.35	704.03	662.85	697.05	689.99	651.51	694.11	714.62	706.76	743.73
	Mean	666.88	695.05	742.93	731.47	686.57	719.36	718.03	676.81	718.52	730.28	728.9	771.42
	Std Dev	10.33	8.51	13.11	14.79	8.23	9.91	13.11	14.06	7.21	10.81	11.03	10.17
	Rank	1.0	4.0	11.0	8.0	3.0	7.0	5.0	2.0	6.0	10.0	9.0	12.0
F22-10	Minimum	649.55	707.28	738.53	655.56	669.07	708.21	685.19	692.77	671.03	694.9	654.96	657.64
	Mean	670.43	734.6	759.11	680.91	686.65	736.31	701.81	711.69	682.19	705.22	668.49	672.08
	Std Dev	10.96	11.66	8.71	6.19	11.15	13.18	14.04	12.62	12.87	14.77	7.74	5.8
	Rank	1.0	10.0	12.0	3.0	5.0	11.0	7.0	8.0	6.0	9.0	2.0	4.0
F22-11	Minimum	607.63	697.55	747.22	699.96	692.97	694.04	705.95	704.95	696.05	740.48	679.2	746.57
	Mean	624.65	717.46	768.64	725.5	715.17	718.57	728.3	726.49	711.69	769.89	704.17	771.2
	Std Dev	8.25	14.56	5.0	7.96	6.43	8.91	11.71	6.8	6.25	5.25	6.61	5.18
	Rank	1.0	6.0	12.0	7.0	3.0	4.0	9.0	8.0	5.0	10.0	2.0	11.0
F22-12	Minimum	637.82	650.01	749.09	706.85	711.74	661.29	662.22	684.12	652.37	668.8	718.42	717.81
	Mean	650.46	679.17	759.47	723.7	733.51	675.76	682.96	698.67	662.8	684.74	730.17	741.55
	Std Dev	6.87	5.74	8.58	10.12	6.5	9.91	14.81	14.41	12.28	12.47	13.71	10.15
	Rank	1.0	2.0	12.0	8.0	9.0	4.0	5.0	7.0	3.0	6.0	11.0	10.0
Friedman rank		12.0	78.0	92.0	87.0	79.0	79.0	86.0	69.0	94.0	85.0	71.0	104.0
Mean rank		1.0	6.5	7.67	7.25	6.58	6.58	7.17	5.75	7.83	7.08	5.92	8.67

Table 5. Comparative statistical results on CEC-2022 test suite (Dimension = 20).

Function	HAALO vs PSO	HAALO vs PPSO	HAALO vs GA	HAALO vs ALO	HAALO vs RIME	HAALO vs GEO	HAALO vs MSACO	HAALO vs DBO	HAALO vs FPA	HAALO vs SHO	HAALO vs LUS
F22-01	1.5E−09	9.3E−09	4.9E−06	2.6E−06	4.6E−06	9.8E−06	4.9E−03	3.3E−03	6.3E−03	2.4E−03	7.7E−04
F22-02	1.3E−09	1.3E−09	1.5E−06	1.4E−06	6.4E−06	1.8E−06	3.5E−03	9.0E−03	4.7E−03	6.7E−03	1.7E−03
F22-03	1.9E−09	4.1E−10	1.7E−06	2.8E−06	1.8E−06	9.0E−07	1.2E−03	4.6E−03	2.1E−03	3.6E−03	5.0E−03
F22-04	6.9E−09	3.9E−10	8.0E−06	6.3E−06	8.3E−07	8.7E−06	9.2E−03	6.2E−04	2.8E−03	8.1E−03	7.5E−03
F22-05	1.8E−09	2.1E−09	3.7E−06	4.9E−06	6.2E−06	3.7E−06	4.6E−03	7.5E−03	3.8E−04	2.5E−03	7.1E−03
F22-06	9.0E−09	5.1E−09	5.3E−06	1.1E−06	4.5E−06	5.3E−06	2.4E−03	2.7E−03	3.8E−03	2.1E−04	3.2E−03
F22-07	2.1E−09	3.3E−09	1.2E−06	8.9E−06	5.9E−06	6.8E−06	7.9E−03	5.0E−03	8.8E−04	5.4E−03	5.9E−03
F22-08	7.5E−09	4.3E−09	1.3E−06	2.8E−06	3.6E−06	6.5E−06	5.7E−03	3.6E−03	9.9E−03	6.1E−03	2.4E−03
F22-09	1.0E−09	1.5E−09	2.5E−06	1.6E−06	1.9E−06	2.9E−06	1.7E−03	9.0E−03	8.1E−04	5.2E−03	4.1E−03
F22-10	9.8E−09	1.1E−09	4.0E−06	9.7E−06	8.7E−06	8.2E−06	2.6E−03	1.7E−03	6.7E−03	9.3E−03	5.6E−03
F22-11	5.7E−09	2.8E−09	7.7E−06	1.9E−06	3.2E−06	4.3E−06	5.1E−03	2.4E−03	1.2E−03	6.1E−03	2.9E−03
F22-12	5.8E−09	1.5E−09	4.8E−06	5.3E−06	5.3E−07	3.4E−06	1.4E−03	6.4E−04	9.9E−03	3.2E−03	8.1E−03

Table 6. Wilcoxon's signed rank test on CEC-2022 test suite (Dimension = 10).

Function	HAALO vs PSO	HAALO vs PPSO	HAALO vs GA	HAALO vs ALO	HAALO vs RIME	HAALO vs GEO	HAALO vs MSACO	HAALO vs DBO	HAALO vs FPA	HAALO vs SHO	HAALO vs LUS
F22-01	2.5E−09	6.8E−09	7.6E−06	6.0E−06	4.7E−06	4.1E−06	3.5E−03	9.3E−03	8.3E−03	9.7E−03	1.3E−03
F22-02	7.3E−09	9.4E−09	1.8E−06	6.7E−07	7.4E−06	5.7E−06	8.4E−03	1.4E−03	8.0E−03	2.0E−03	1.6E−03
F22-03	1.6E−09	8.1E−09	6.7E−06	5.2E−06	3.6E−06	8.8E−06	3.9E−03	8.2E−03	4.4E−03	3.8E−03	4.6E−03
F022-4	3.0E−09	7.5E−09	5.0E−06	2.3E−06	9.0E−06	3.8E−06	5.4E−03	9.1E−03	6.2E−03	1.2E−03	9.4E−03
F22-05	6.3E−09	3.3E−09	1.4E−06	7.9E−06	6.2E−06	5.3E−06	8.9E−03	7.9E−03	1.5E−03	3.1E−03	2.5E−03
F22-06	7.4E−09	3.4E−10	5.7E−06	7.6E−06	8.8E−06	3.4E−06	8.2E−03	1.1E−03	8.5E−03	1.3E−03	4.0E−03
F22-07	8.0E−09	1.5E−09	2.3E−06	7.2E−06	7.2E−06	6.4E−06	6.9E−03	5.4E−03	2.5E−03	3.5E−03	1.8E−03
F22-08	9.1E−09	5.8E−09	4.0E−06	4.6E−06	9.5E−06	1.5E−06	5.9E−03	5.1E−03	6.1E−03	1.9E−04	8.7E−03
F22-09	9.3E−09	5.7E−09	7.0E−06	9.2E−06	7.1E−06	1.5E−06	5.8E−03	6.1E−03	4.2E−03	7.4E−03	9.3E−03
F22-10	9.3E−09	4.5E−09	1.1E−06	9.8E−06	8.4E−06	1.3E−06	9.2E−03	8.7E−03	5.2E−03	5.9E−03	4.0E−03
F22-11	5.5E−10	3.4E−09	8.0E−06	5.6E−08	3.3E−06	4.0E−06	5.4E−03	9.2E−03	3.5E−03	3.5E−03	7.4E−03
F22-12	4.5E−09	2.2E−09	4.5E−06	1.4E−06	1.8E−06	5.0E−06	4.2E−03	9.1E−03	3.6E−03	5.8E−03	6.3E−03

Table 7. Wilcoxon's signed rank test on CEC 2022 test suite (Dimension = 20).

Gains	Parameters							
	K_{PS}	K_{IS}	K_{DS}	K_{PC}	K_{IC}	K_{DC}	Hb_{up}	Hb_{down}
Lower limit(min)	0	0	0	0	0	0	0	−20
Upper limit(max)	30	20	20	20	20	20	20	0

Table 8. Range of gains in proposed and other controllers.

Self-adaptive parameter control

To further enhance the performance, HAALO employs self-adaptive mechanisms for controlling parameters such as the inertia weight ω_t and the exploration–exploitation balance. The inertia weight is dynamically adjusted based on the diversity of the population at each iteration *t*:

$$\omega_t = \omega_{min} + (\omega_{max} - \omega_{min}) \times \frac{D(t)}{D_{max}} \tag{9}$$

where $D(t)$ is the diversity of the population at iteration *t*, and D_{max} is the maximum diversity observed. The diversity is calculated as:

$$D(t) = \frac{1}{n} \sum_{i=1}^{n} x_i(t) - x_{mean}(t) \tag{10}$$

Controller	FOPID $\times 10^{-6}$							PI $\times 10^{-6}$; PID $\times 10^{-6}$						
Algorithm	HAALO	PSO	PPSO	ALO	GEO	RIME	GA	HAALO	PSO	PPSO	ALO	GEO	RIME	GA
Best	1.5824	4.9982	3.0050	2.2183	2.9785	1.7234	3.9215	2.5415	5.8413	3.0052	2.8516	2.6872	3.0472	7.2195
Worst	3.2251	8.3945	5.9413	3.8516	5.3416	4.8416	6.9816	5.4168	9.5513	6.8416	5.7143	5.3419	7.1642	10.631
Mean	2.0046	5.7923	4.0053	2.4167	3.9541	3.7719	5.1972	3.5572	6.9417	4.0286	3.5419	4.8816	5.9981	9.9513

Table 9. Values of the OF following 20 rounds of optimization algorithms using various controllers.

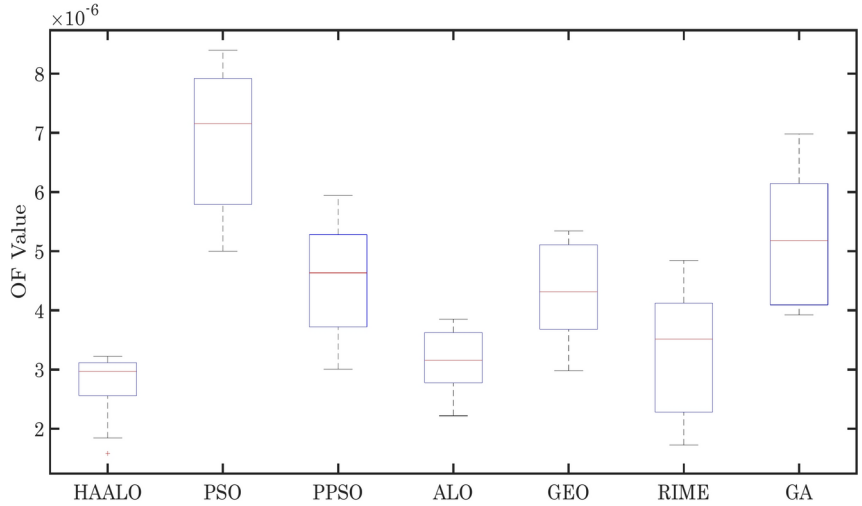


Fig. 5. Boxplot schematic for HAALO optimization and other metaheuristic algorithms.

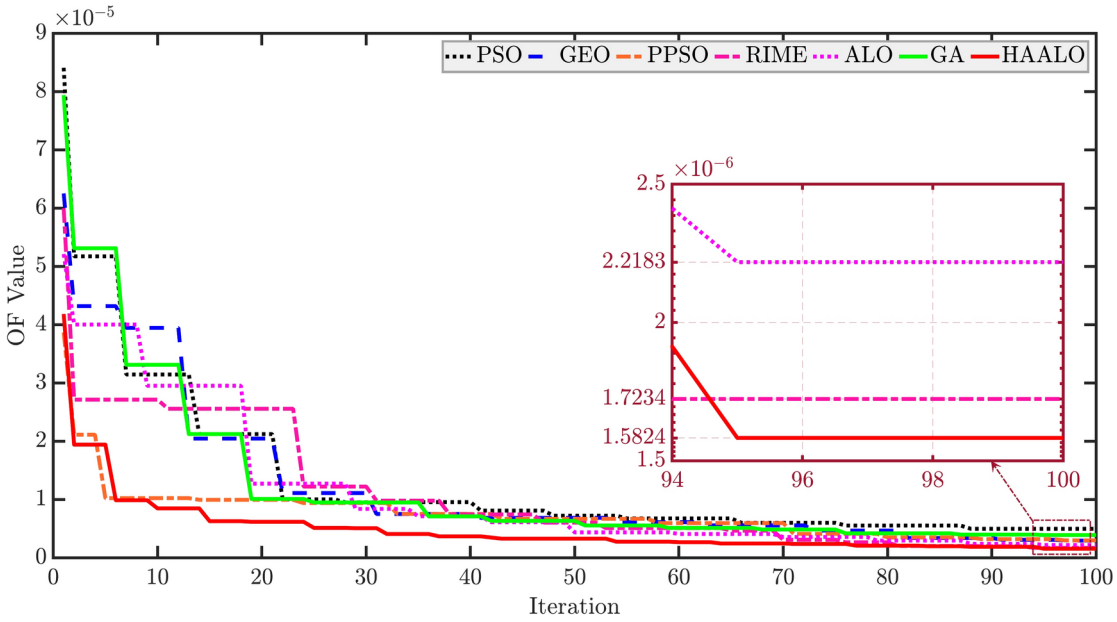


Fig. 6. The best results of the OF by running the proposed HAALO, ALO, PSO, RIME, PPSO, GEO and GA algorithms.

where $x_{mean}(t)$ is the mean position of all ants at iteration t. By dynamically adjusting parameters, HAALO ensures that the balance between exploration and exploitation is maintained throughout the optimization process.

Convergence and final solution
HAALO converges to the optimal solution by combining adaptive exploration and exploitation techniques. The algorithm terminates when the maximum number of iterations T_{max} is reached, and the best solution found is returned as the global optimum:

$$x^* = argminf\left(x_i\left(T_{max}\right)\right) \tag{11}$$

The final solution is x^* the best antlion's position. Table 3 shows the pseudocode table for HAALO optimization method:

Performance analysis of the proposed algorithm on CEC-2022 test suite

This section validates the enhanced optimization capabilities of the HAALO algorithm through rigorous experimentation. The evaluation begins with tests conducted on the latest CEC-2022 benchmark functions, recognized for their complexity and diversity in representing real-world optimization challenges. These benchmarks ensure that the algorithm's performance is assessed against standard and widely accepted criteria. Following this, the algorithm is applied to the proposed SRM problem, focusing on objectives such as torque ripple minimization, speed regulation, and adaptability under dynamic load conditions. Detailed results from both the benchmark and application-specific experiments are presented in the subsequent subsections, highlighting the algorithm's robustness and superiority compared to other optimization methods.

Simulation and benchmarking setup
To rigorously evaluate the effectiveness of the proposed HAALO algorithm, its performance was benchmarked against a comprehensive suite of state-of-the-art metaheuristic optimization methods. The evaluated methods included particle swarm optimization (PSO), progressive particle swarm optimization (PPSO), genetic algorithm (GA), ant lion optimizer (ALO), RIME optimizer, golden eagle optimizer (GEO), multiple stage ant colony optimization (MSACO), dung beetle optimizer (DBO), flower pollination algorithm (FPA), spotted hyena optimizer (SHO), and local unimodal sampling (LUS). These algorithms were selected for their prominence in optimization research and their diverse operational mechanisms, providing a robust and comprehensive comparative framework. The evaluation utilized the CEC-2022 benchmark functions, with 30 independent runs conducted for each algorithm. A maximum of 500 iterations and a population size of 20 were maintained across all experiments to ensure uniformity and reliability in the results. The CEC 2022 benchmarks, featuring 10- and 20-dimensional hyper-spaces, were chosen for their ability to replicate real-world optimization challenges, offering a rigorous testbed for assessing algorithmic performance. All parameter settings for the compared algorithms were initialized based on their recommended default values to provide an ideal environment for performance demonstration. This systematic approach ensured fairness and enabled a detailed analysis of HAALO's capabilities relative to other advanced optimization techniques, highlighting its potential to excel in complex optimization scenarios.

Statistical results
The proposed HAALO algorithm was evaluated using the CEC 2022 test suite, which features 10- and 20-dimensional search spaces. This suite comprises 12 distinct functions, categorized into unimodal, multimodal, hybrid, and composition functions, providing a diverse set of optimization challenges[24]. Further details about these functions can be found in[24]. Tables 4 and 5 respectively provide the comparative statistical results of 10 and 20 dimensional CEC 2022 benchmark functions. Besides, the comparative results of the Wilcoxon's signed

Controllers – algorithms	Parameters											
	K_{PS}	K_{IS}	K_{DS}	λ_S	μ_S	K_{PC}	K_{IC}	K_{DC}	λ_C	μ_C	Hb_{up}	Hb_{down}
PI-HAALO	30	8.299	0	1	0	20	0.8517	0	1	0	0	0
PID-PSO[12]	8.2724	19.1082	0.5159	1	1	16.8093	8.2129	0.3467	1	1	2.8493	− 3.9185
PID-PPSO[12]	20	19.9364	1.0257	1	1	18.9230	0.4408	0	1	1	11.0747	− 18.1291
PID-ALO[12]	4.8662	1.0845	0.0577	1	1	4.3679	0.0351	9.4898	1	1	0.2482	− 9.1854
PID-RIME	17.501	5.69854	874e−7	1	1	9.03235	0.0979	0.256	1	1	0	0
PID-GEO	20	3.199	0	1	1	1.5845	5.36412	9.6521	1	1	1.2907	0
PID-GA[12]	4.5341	10.4119	0.0773	1	1	10.5768	13.9666	15.9050	1	1	8.1984	− 15.8896
FOPID-HAALO	11.0214	8.0854	18.7841	0.9924	3448e−13	11.7069	7.7887	15.5277	0.82317	0.0071	3.2514	− 1.0254
FOPID-PSO[12]	3.0006	12.2041	1.0619	0.9351	0.3260	3.3328	14.0403	1.3530	0.3080	0.0719	20	− 17.7777
FOPID-PPSO[12]	0.3659	11.3479	2.0341	0.5613	0.3577	8.4507	14.4807	1.0683	0.7041	0.0177	6.5840	− 8.0318
FOPID-ALO[12]	2.2623	8.0692	1.7222	0.7845	0.3448	13.7069	17.7887	18.5277	0.2317	5.0788e−5	12.3755	− 0.7385
FOPID-RIME	2.0145	12.2543	10.5412	0.94125	0.00147	15.15	0.6455	1.0365	0.5412	0.00014	1.23	− 0.145
FOPID-GEO	1.02145	16.9854	3.78412	0.9924	0.3448	18.7069	5.7887	10.257	0.3234	75e−7	18.3755	− 0.7385
FOPID-GA[12]	1.5312	12.7514	1.9267	0.9283	0.3410	13.0542	19.0341	0.6248	0.3180	0.0737	4.3712	− 12.1320

Table 10. Optimum gains of proposed and other controllers.

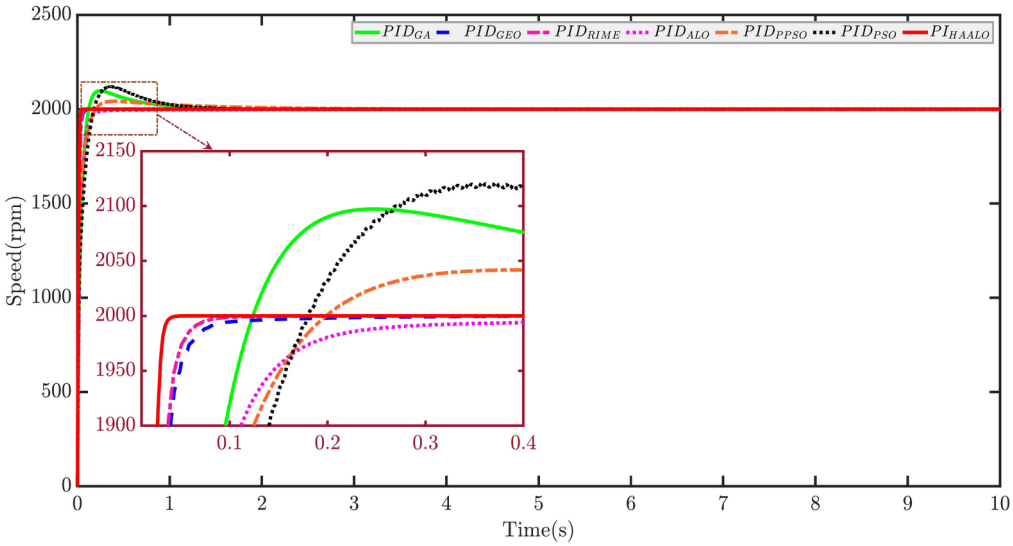


Fig. 7. Time response of speed for PI-HAALO and other controllers in scenario 1.

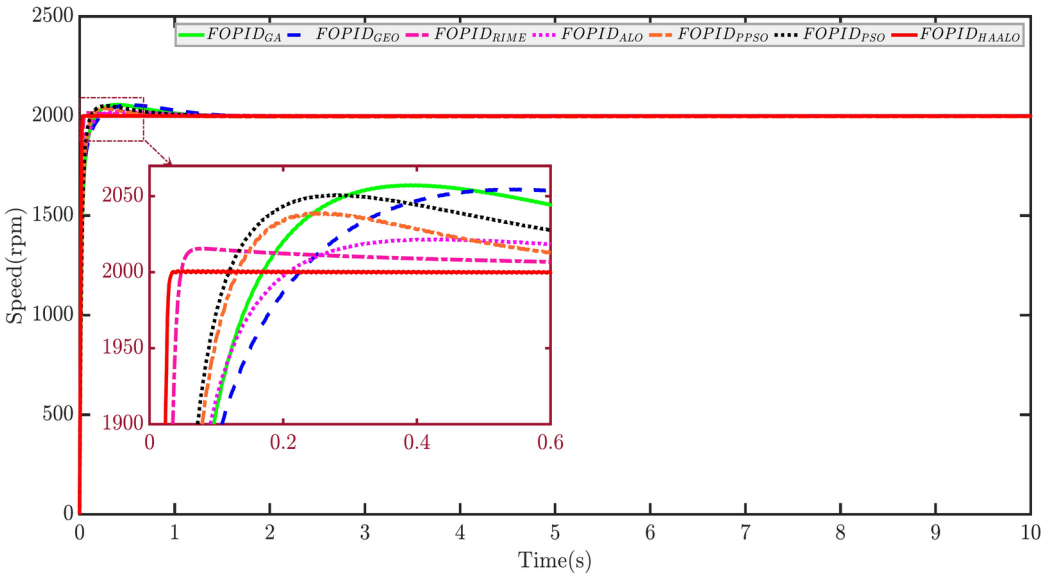


Fig. 8. Time response of speed for FOPID-HAALO and other controllers in scenario 1.

rank test are provided in Tables 6 and 7. The presented results within those tables confirm the superiority of the proposed HAALO in terms of exploration and exploitation.

Objective function definition

The objective function (OF) is a measure used by the designer to evaluate the dynamic response of the system. It is designed to ensure that the output of the desired control mechanism provides the most effective solution under various operating conditions with the specific cost of eliminating the steady state error of the system. The objective function is defined as Eq. (12) [12,21]:

$$OF_{min} = \int_0^{t_{sim}} (\Delta c)^2 + (\Delta s)^2 \, dt \tag{12}$$

Here, t_{sim} represents the simulation time in seconds. Δc and Δs refer to the current error and speed error, respectively. The OF is constrained by the range of the controller coefficients, which sets the boundaries of the search space for the optimization problem, as outlined in Table 8.

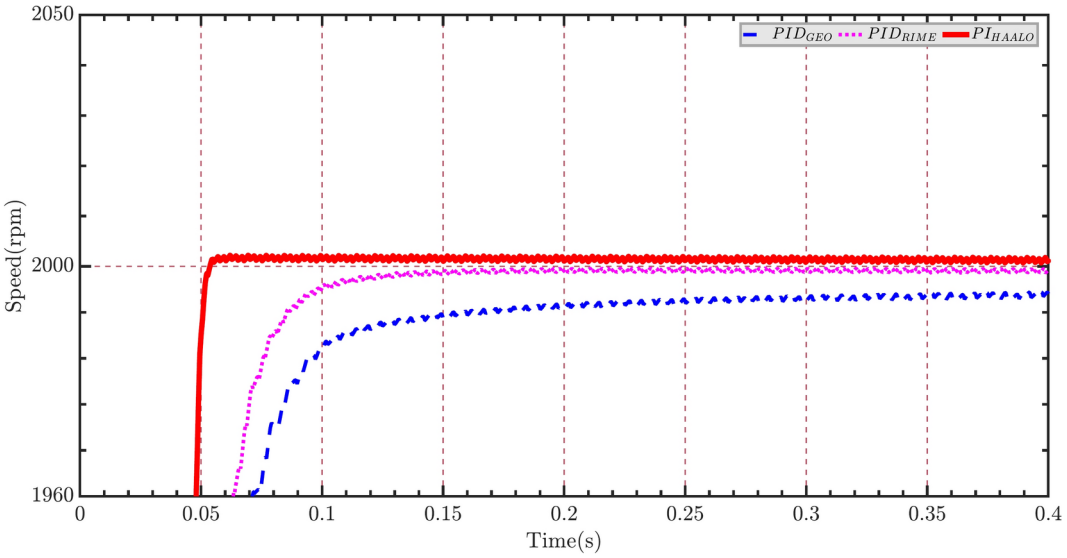


Fig. 9. Time response of speed for PI-HAALO and two best controllers in scenario 1.

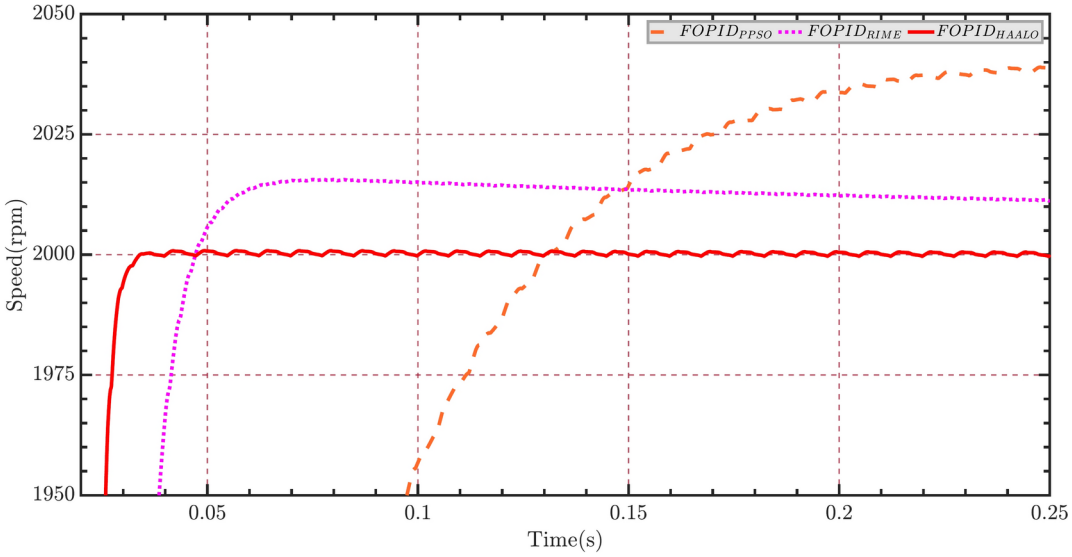


Fig. 10. Time response of speed for FOPID-HAALO and two best controllers in scenario 1.

Results of the simulation and analysis

In this section, we implement the proposed PI and FOPID controllers based on the HAALO algorithm to perform control functions within the SRM, as previously discussed in "Optimization" section. The closed-loop system is modeled in MATLAB 2023a using Simulink. Through 100 iterations with 20 particles, the HAALO algorithm has identified the optimal controller coefficients. The simulation duration is set to 10 s. Fractional-order (FO) operators are modeled using the FOMCON plugin in MATLAB, with a frequency range of [0.001, 1000] and an approximation order of 5. Testing with higher-order approximations showed minimal change in results. The HAALO optimization algorithm was run independently in 20 trials, with Table 9 summarizing the best, worst, and average OF values achieved across different controllers. Figure 5 provides a comparative analysis using boxplots for seven algorithms HAALO, PSO, PPSO, ALO, GEO, RIME, and GA based on their effectiveness in minimizing the objective function. The boxplot in Fig. 5 indicates that even the lowest-performing result from the HAALO algorithm outperforms the highest results of the other six algorithms, highlighting the clear performance advantage of the proposed HAALO algorithm. Figure 6 depicts the best results of the OF by running the proposed HAALO, ALO, PSO, RIME, PPSO, GEO and GA algorithms.

The HAALO algorithm has undergone sufficient iterations, confirming its convergence to the optimal solution. To evaluate the efficiency of the proposed controller, we utilize the HAALO technique for optimization. Figure 5 illustrates the outcomes from this optimized implementation, showing that the lowest OF values were

Controller and optimization method	T_r	T_s	T_p	OS
PI-HAALO	0.0194	0.0295	0.0722	0.0082
PID-PSO[12]	0.1082	0.8528	0.3614	6.0150
PID-PPSO[12]	0.0865	0.4922	0.3942	2.0999
PID-ALO[12]	0.0784	0.1558	9.9877	0.0009
PID-RIME	0.0276	0.0467	9.9563	0.0076
PID-GEO	0.0283	0.0511	1.1207	0.1082
PID-GA[12]	0.0744	0.6692	0.2454	4.8901
FOPID-HAALO	0.0171	0.0262	0.0569	0.0563
FOPID-PSO[12]	0.0495	0.4543	0.2781	2.5505
FOPID-PPSO[12]	0.0531	0.2992	0.2551	2.0807
FOPID-ALO[12]	0.0555	0.1335	0.4288	1.1287
FOPID-RIME	0.0262	0.0394	0.0775	0.7852
FOPID-GEO	0.0604	0.8453	0.5444	2.7291
FOPID-GA[12]	0.0614	0.6504	0.3893	2.8734

Table 11. Transient response specifications in scenario 1.

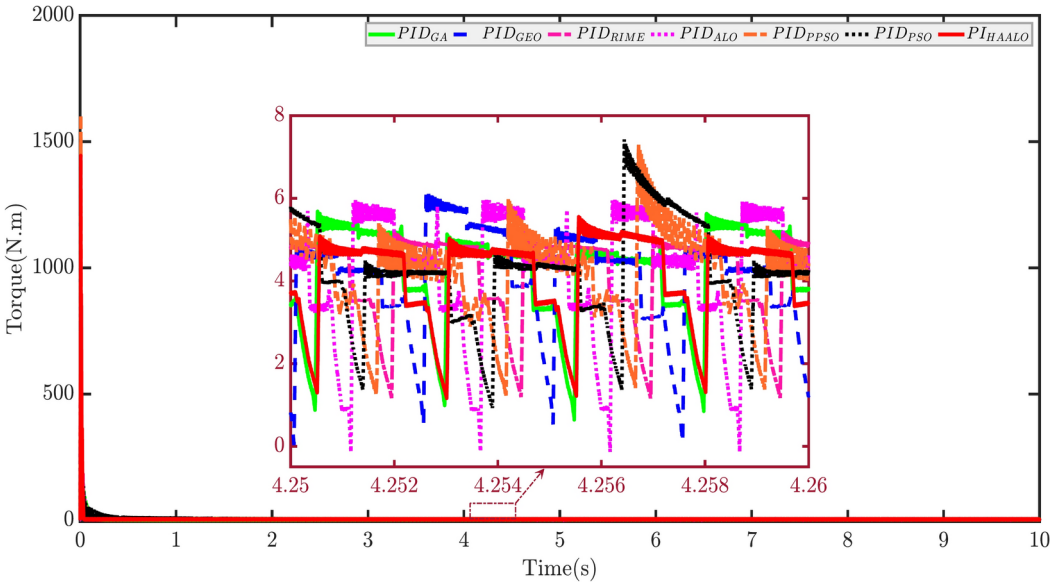


Fig. 11. Time response of total torque for PI-HAALO and other controllers in scenario 1.

achieved after 100 iterations, highlighting the controller's enhanced performance. The HAALO optimization was also independently executed across 20 trials, with Table 9 presenting the best, worst, and average OF values obtained with different controllers.

Table 10 provides the optimal controller parameter values derived from the best results achieved by the HAALO algorithm. To enable a thorough numerical comparison, we calculated and reported the time-domain evaluation indices for each scenario. These indices include the integral of square error (ISE), integral of time-weighted square error (ITSE), integral of absolute error (IAE), and integral of time-weighted absolute error (ITAE). The equations for these indices are provided in Eqs. (13) to (16).

$$ISE = \int_0^x e^2(t)\,dt \tag{13}$$

$$ITSE = \int_0^x t.e^2(t)\,dt \tag{14}$$

$$IAE = \int_0^x |e(t)|\,dt \tag{15}$$

$$ITAE = \int_0^x t.|e(t)|\,dt \tag{16}$$

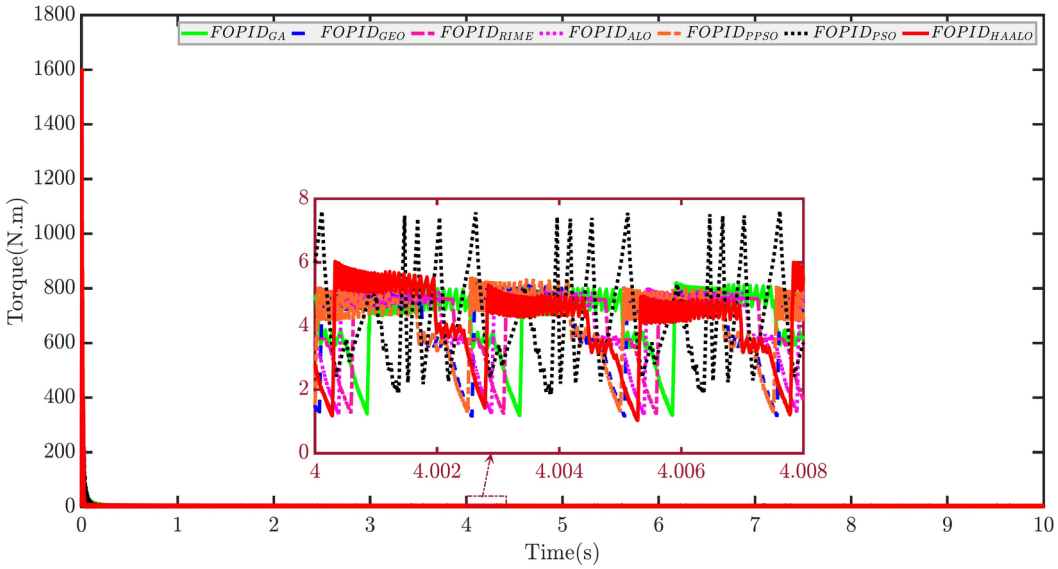


Fig. 12. Time response of total torque for FOPID-HAALO and other controllers in scenario 1.

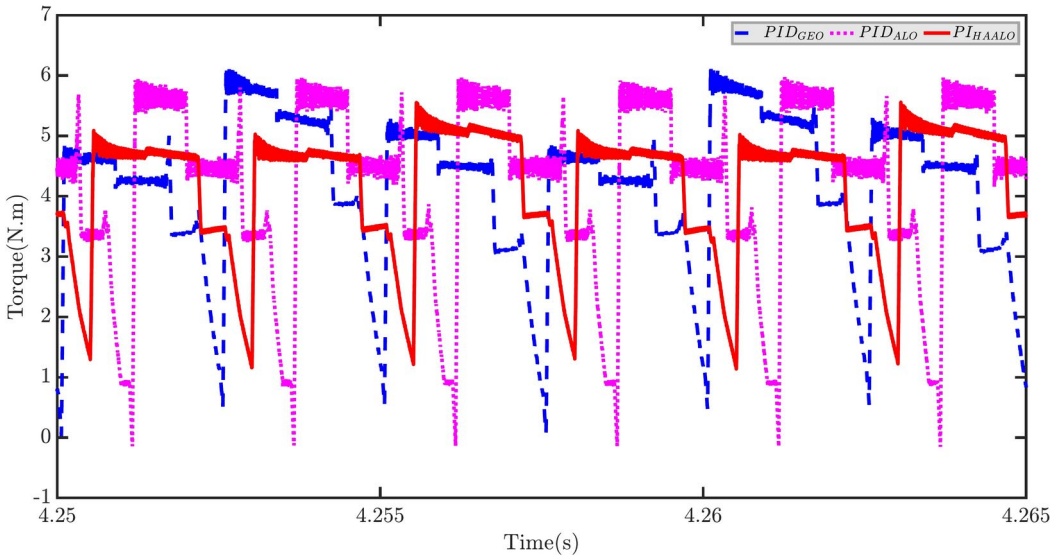


Fig. 13. Time response of total torque for FOPID-HAALO and two best controllers in scenario 1.

where, x is simulation time in s, and e(t) is an error signal between speed and current in SRM.

Comparison in different scenarios

Scenario 1: no load applied SRM

In this scenario, the motor operates without an external load, and we observe its output speed, torque, and current by setting the reference speed to 2000 r.p.m. Figures 7 and 8 display the motor's speed response under no-load conditions for the SRM. Here, the PI controller with the proposed optimization method is compared against the traditional PID controller, showing superior results. Notably, the HAALO-based PI and FOPID controllers achieve the most accurate speed tracking. In addition, for better clarity, Figs. 9, 10 provide a magnified comparison of the proposed method with the two best responses for speed in the SRM.

Table 11 outlines the transient response characteristics, demonstrating that the proposed optimization method achieves faster response times than other meta-heuristic algorithms. Additionally, Figs. 11 and 12 reveal that the HAALO-based PI and FOPID controllers produce the lowest torque ripple at the reference speed compared to other controllers. Additionally, for better clarity, Figs. 13, 14 provide a magnified comparison of the proposed method with the two best responses for torque in the SRM. Figures 15 and 16 illustrate the output current in one phase of the SRM using various controllers. Table 12 provides time-domain evaluation criteria for internal and external controllers, confirming that the proposed controller yields lower error values.

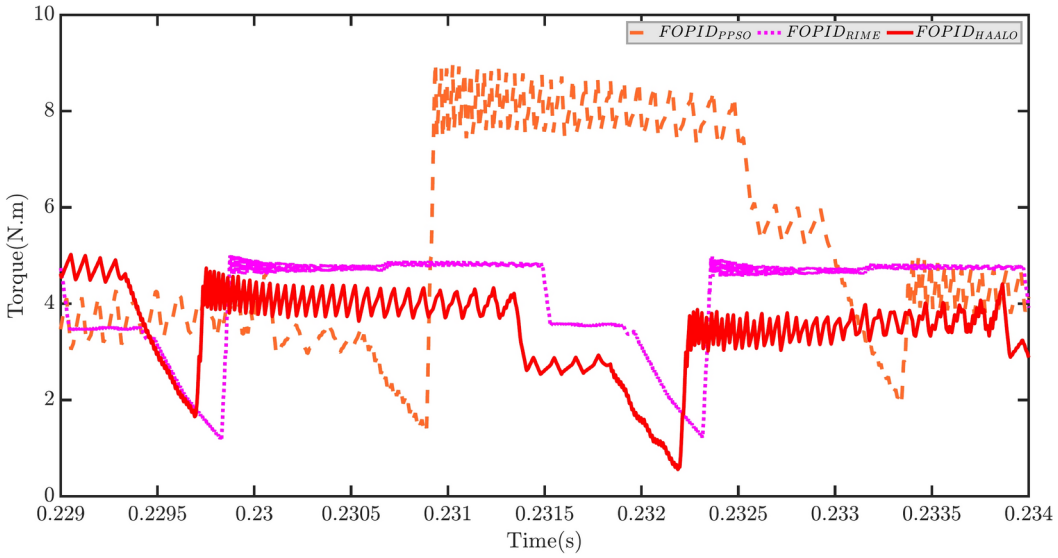


Fig. 14. Time response of total torque for FOPID-HAALO and two best controllers in scenario 1.

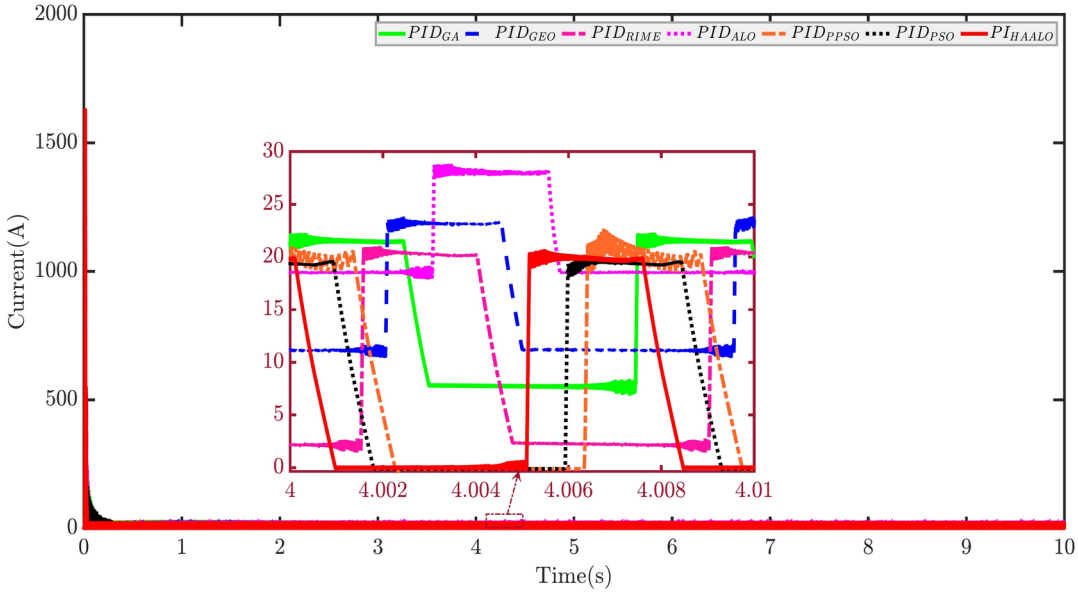


Fig. 15. Time response of phase current for PI-HAALO and other controllers in scenario 1.

Scenario1: no load with time-delay non-linearity for PI-HAALO controller
Managing systems becomes more challenging due to factors such as time delays, non-linearity from power line interference, or SRM malfunctions. In this case, it is assumed that the control system's stimulation signal experiences a 20 ms delay on the outer controller. As shown in Fig. 17, the PI-HAALO controller outperforms others in reaching the reference speed, with other controllers following in rank. Transient response values are listed in Table 13. In addition, for better clarity, Fig. 18 provide a magnified comparison of the proposed method with the two best responses for speed in the SRM.

Figures 19 displays the SRM torque curve, where the proposed controller demonstrates strong performance. Additionally, for better clarity, Fig. 20 provide a magnified comparison of the proposed method with the two best responses for torque in the SRM. Figure 21 shows the phase current for each controller in the SRM. Notably, it achieves a low torque ripple at the reference speed, whereas other controllers have yet to reach this speed.

Scenario 2: change speed in SRM

In practical scenarios, motors often do not operate under ideal conditions. In this case, a reference speed of 3000 r.p.m is applied to the SRM to assess the performance of the optimization method and controllers under these conditions. Table 14 outlines the transient response characteristics, demonstrating that the proposed

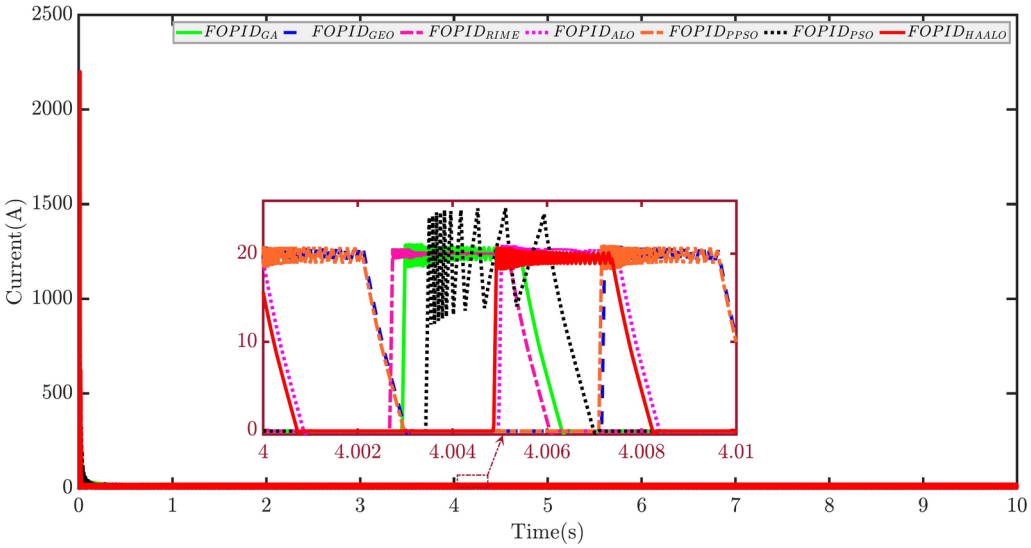


Fig. 16. Time response of phase current for FOPID-HAALO and other controllers in scenario 1.

Controller and optimization method	ISE $\times 10^{-7}$	IAE $\times 10^{-4}$	ITSE $\times 10^{-7}$	ITAE $\times 10^{-4}$
PI-HAALO	3388	42	336	126
PID-PSO[12]	634	708	326	166
PID-PPSO[12]	382	53	316	149
PID-ALO[12]	1861	374	6865	1836
PID-RIME	1948	76	397	295
PID-GEO	3123	231	1806	849
PID-GA[12]	865	187	988	607
FOPID-HAALO	354	45	332	147
FOPID-PSO[12]	560	71	604	280
FOPID-PPSO[12]	494	59	365	230
FOPID-ALO[12]	487	46	302	156
FOPID-RIME	113	40	314	130
FOPID-GEO	501	51	312	149
FOPID-GA[12]	490	53	324	167

Table 12. Time-based indicators illustrating the dynamic response of the SRM in scenario.

optimization method achieves faster response times than other meta-heuristic algorithms. Figures 22 and 23 display the speed response when 3000 rpm is imposed on the SRM. These figures clearly show that the proposed optimization method tracks the reference speed faster than other controllers. In addition, for better clarity, Figs. 24, 25 provide a magnified comparison of the proposed method with the two best responses for speed in the SRM.

A notable advantage of the proposed controller is its stable and acceptable torque ripple at the target speed. In contrast, while other controllers fail to reach the reference speed for the SRM, they exhibit higher torque ripple compared to the PI and FOPID controllers based on the HAALO method. Figures 26, 27, 28, and 29 present a comparison of torque between the proposed controller and other controllers in the SRM. Specifically, Figs. 28 and 29 provide a magnified comparison between the proposed controller and the two best-performing controllers. Figures 30 and 31 display current characteristics of the SRM in this scenario. Table 15 provides time-domain evaluation criteria for internal and external controllers, confirming that the proposed controller yields lower error values.

Scenario 3: dynamic load (stepwise load decrease)

In this section, a stepwise dynamic load is applied to the SRM, starting with a load of 10 N.m from t = 2 s to t = 4 s, followed by an increased load of 30 N.m from t = 4 s to t = 10 s. Figures 32 and 33 illustrate the performance of the proposed optimization method in this scenario. As shown in Figs. 32 and 33, the proposed optimization method experiences less speed drop under loading compared to other metaheuristic algorithms. When the load is removed, it effectively tracks the reference speed. One key advantage of the proposed PI and FOPID controllers based on HAALO method is its stable torque ripple at the target speed, unlike other controllers,

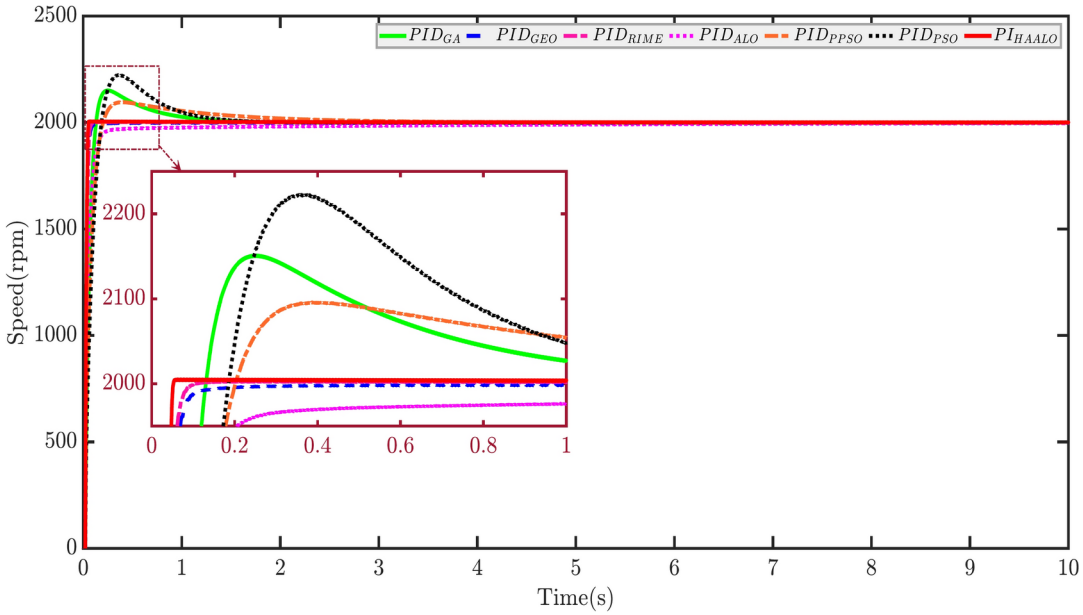


Fig. 17. Time response of speed for PI-HAALO and other controllers in scenario 1 with time delay.

Controller and optimization method	T_r	T_s	T_p	OS
PI-HAALO	0.0177	0.0481	0.0620	0.2296
PID-PSO[12]	0.1192	1.0677	0.3664	11.1793
PID-PPSO[12]	0.1108	1.3017	0.3940	4.7945
PID-ALO[12]	0.0861	0.2294	9.9994	0.0271
PID-RIME	0.0263	0.0634	0.2531	0.1123
PID-GEO	0.0277	0.0708	1.5755	0.0557
PID-GA[12]	0.0733	0.8496	0.2509	7.5589

Table 13. Transient response specifications in scenario 1 with time delay.

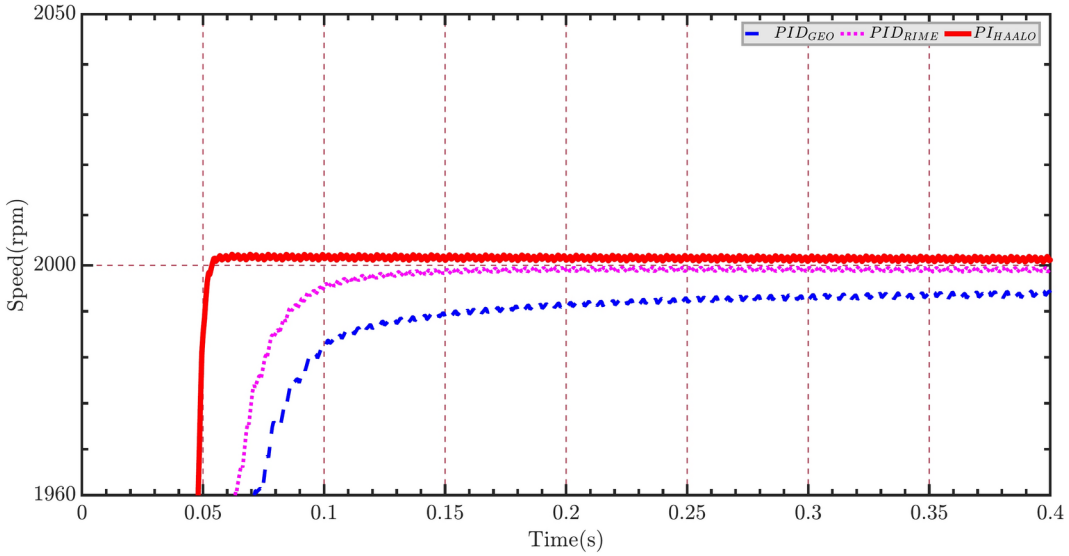


Fig. 18. Time response of speed for PI-HAALO and two best controllers in scenario 1 with time delay.

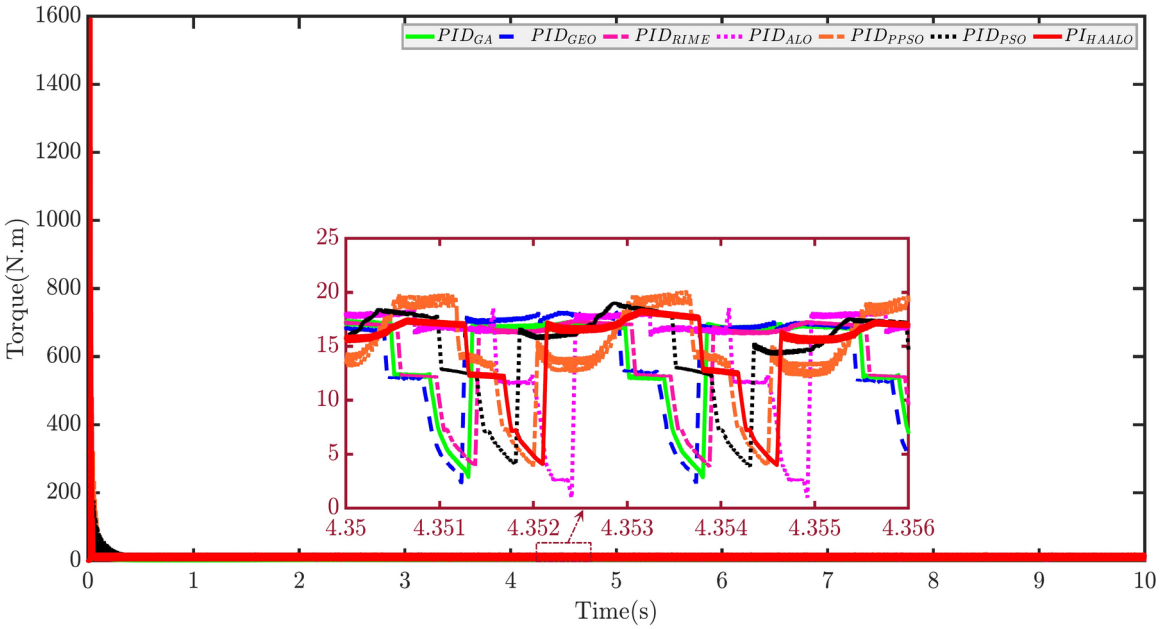


Fig. 19. Time response of total torque for PI-HAALO and other controllers in scenario 1 with time delay.

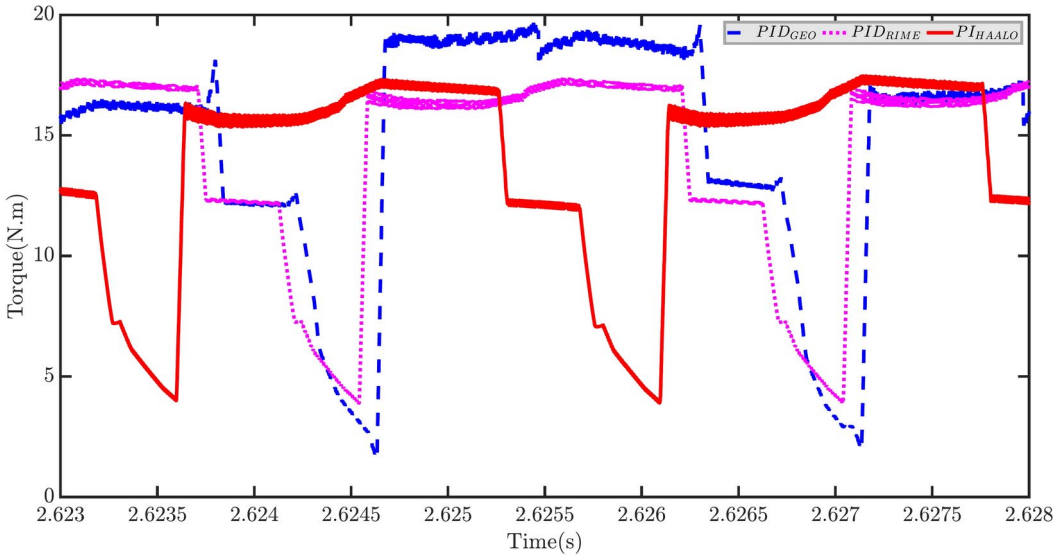


Fig. 20. Time response of total torque for PI-HAALO and two best controllers in scenario 1 with time delay.

which not only struggle to reach the SRM's reference speed but also show higher torque ripple than the PI and FOPID controllers based on the HAALO method. Figures 34, 35, 36, and 37 provide a comparison of torque between the proposed controller and other controllers of the SRM in this scenario. Specifically, Figs. 36 and 37 provide a magnified comparison between the proposed controller and the two best-performing controllers. Figures 38 and 39 display current characteristics of the SRM in this scenario.

Scenario 4: complicated operation

In this scenario, a challenging operation is designed for the SRM to assess how effectively the proposed optimization method and controller functions under rare and demanding conditions. This operation unfolds in three stages:

1. From t = 0 s to t = 2 s, there is no load on the motor, while a reference speed of 2000 r.p.m is maintained from t = 0 s to t = 3 s.
2. At t = 2 s, a load torque of 10 N.m is applied, lasting until t = 4 s, while the reference speed increases to 3000 r.p.m from t = 3 s to t = 10 s.

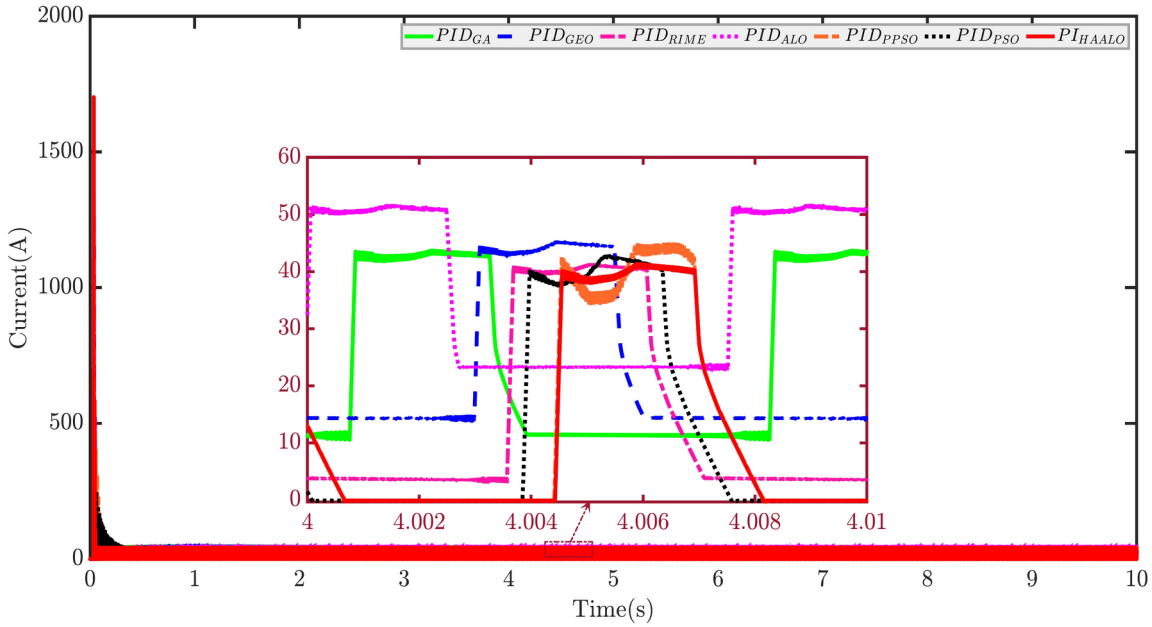


Fig. 21. Time response of phase current for PI-HAALO and other controllers in scenario 1 with time delay.

Controller and optimization method	T_r	T_s	T_p	OS
PI-HAALO	3.0306	3.0571	3.0915	0.1720
PID-PSO[12]	3.0613	3.5627	3.3559	2.7903
PID-PPSO[12]	3.0470	3.1321	3.3866	1.2259
PID-ALO[12]	3.0412	3.1129	9.9825	0.0062
PID-RIME	3.0318	3.0592	3.1960	0.2023
PID-GEO	3.0313	3.0601	3.6213	0.0769
PID-GA[12]	3.0430	3.2855	3.2462	2.0561
FOPID-HAALO	3.0241	3.0461	3.0698	0.1819
FOPID-PSO[12]	3.0307	3.0697	3.2511	1.3777
FOPID-PPSO[12]	3.0308	3.0773	3.2418	1.2288
FOPID-ALO[12]	3.0301	3.0881	3.3715	0.8059
FOPID-RIME	3.0307	3.0572	3.0965	0.7955
FOPID-GEO	3.0303	3.1132	3.5074	1.5510
FOPID-GA[12]	3.0317	3.0922	3.3612	1.5466

Table 14. Transient response specifications in scenario 2.

3. At t = 4 s, the load torque rises to 30 N.m, continuing until t = 10 s, with the reference speed held at 3000 r.p.m from t = 3 s onward.

Figures 40 and 41 demonstrate that the proposed optimization approach, combined with PI and FOPID controllers based HAALO method, achieves more adaptable performance compared to other PID and FOPID controllers, ensuring the SRM operates reliably even in abnormal conditions. Figures 42 and 43 display the SRM's torque during this period, showing that, unlike other controllers that result in motor speeds below the reference, the optimized method enables the motor to achieve a desirable torque at the reference speed. Additionally, for better clarity, Figs. 44 and 45 provide a magnified comparison of the proposed method with the two best responses for torque in the SRM. Figures 46 and 47 illustrate the output current for one phase of the SRM under different controllers.

Conclusions

SRMs by integrating the HAALO algorithm with PI and FOPID controllers. The proposed method significantly reduces torque ripple and enhances speed regulation, offering a more adaptable and robust solution compared to traditional control methods. Simulation results confirm that HAALO-optimized controllers outperform established optimization techniques, showing faster convergence, reduced computational overhead, and superior control accuracy. While the simulation results demonstrate the effectiveness of the proposed method, there are

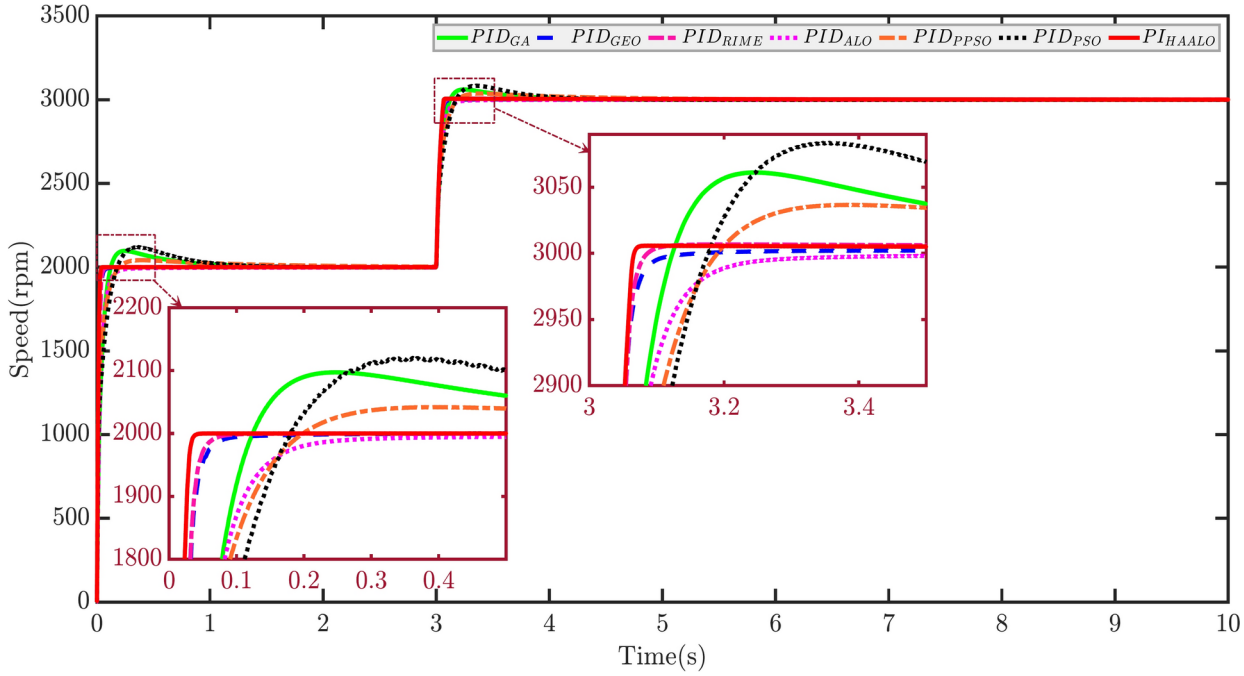


Fig. 22. Time response of speed for PI-HAALO and other controllers in scenario 2.

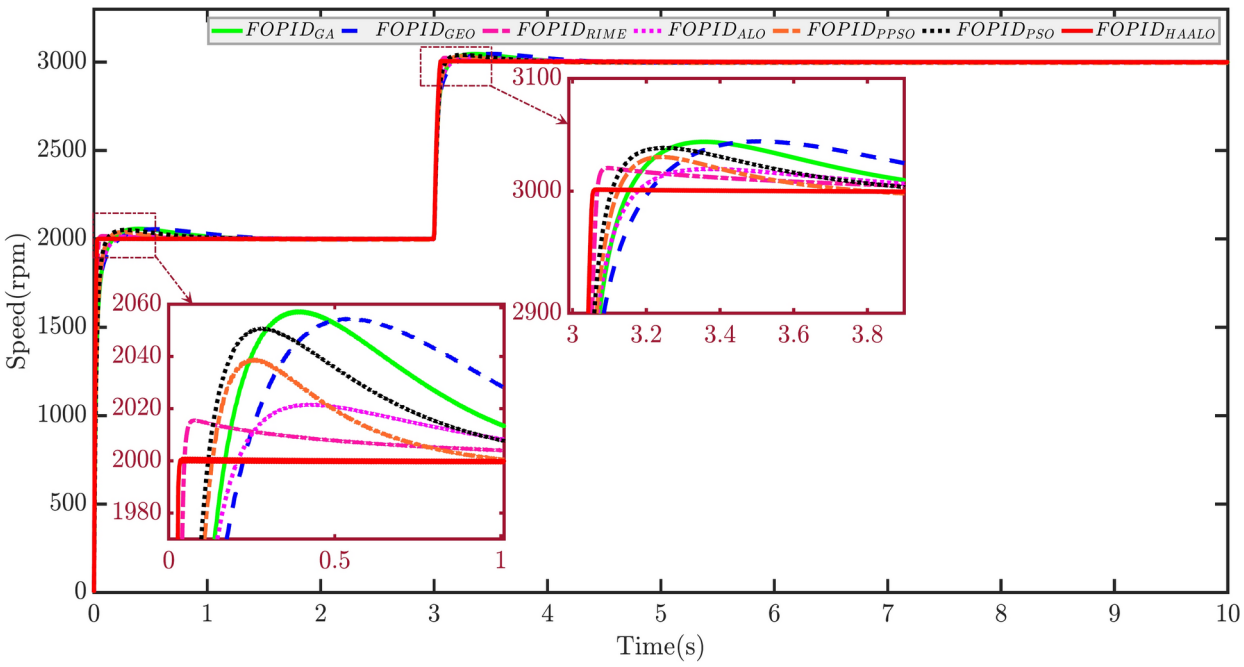


Fig. 23. Time response of speed for FOPID-HAALO and other controllers in scenario 2.

practical limitations to consider. The real-world implementation of the HAALO-optimized controllers may be constrained by factors such as hardware limitations, real-time computational capacity, and the ability to maintain control precision under extreme operating conditions. In practice, challenges such as sensor inaccuracies, external disturbances, and system nonlinearity could affect the accuracy of switching angle optimization and the overall system performance.

Future work should focus on addressing these limitations by investigating the practical implementation of the proposed control system in real-time applications. This would involve testing the system with actual SRM hardware, exploring hardware-specific adjustments, and evaluating performance under real-world conditions. Additionally, integrating machine learning techniques for predictive control and fault detection could further

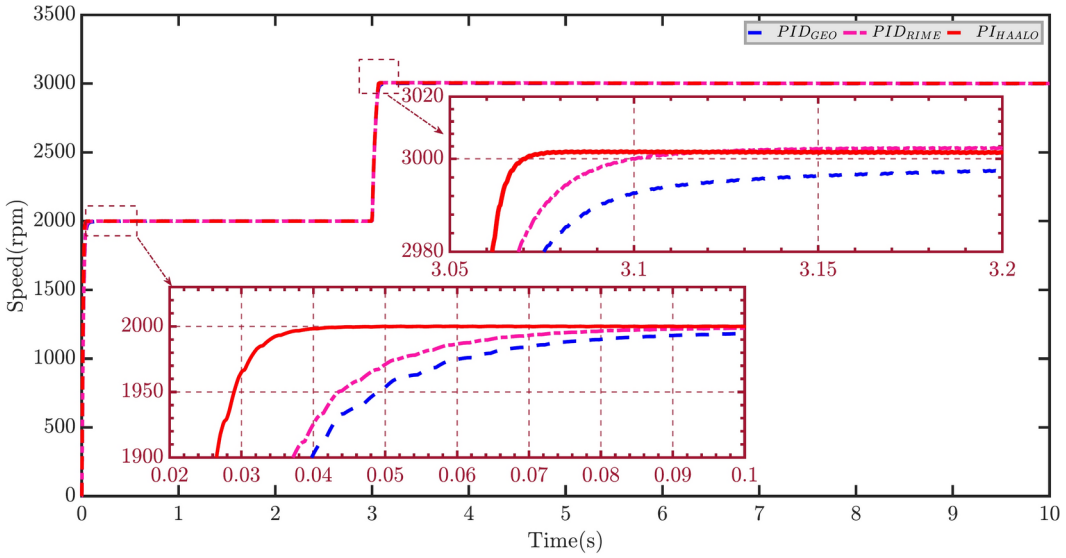


Fig. 24. Time response of speed for PI-HAALO and two best controllers in scenario 2.

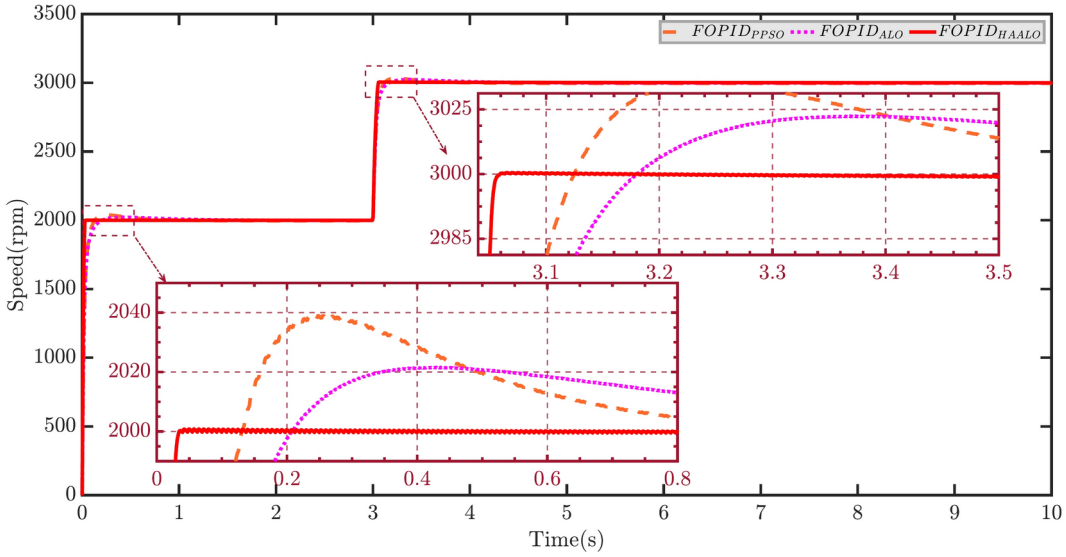


Fig. 25. Time response of speed for FOPID-HAALO and two best controllers in scenario 2.

enhance the adaptability of the system. The extension of the HAALO algorithm to multi-objective optimization, targeting energy efficiency and thermal management, could provide even more practical benefits for high-performance applications. Lastly, exploring hybrid optimization methods that combine HAALO with other advanced control strategies may lead to further improvements in SRM performance in industrial environments.

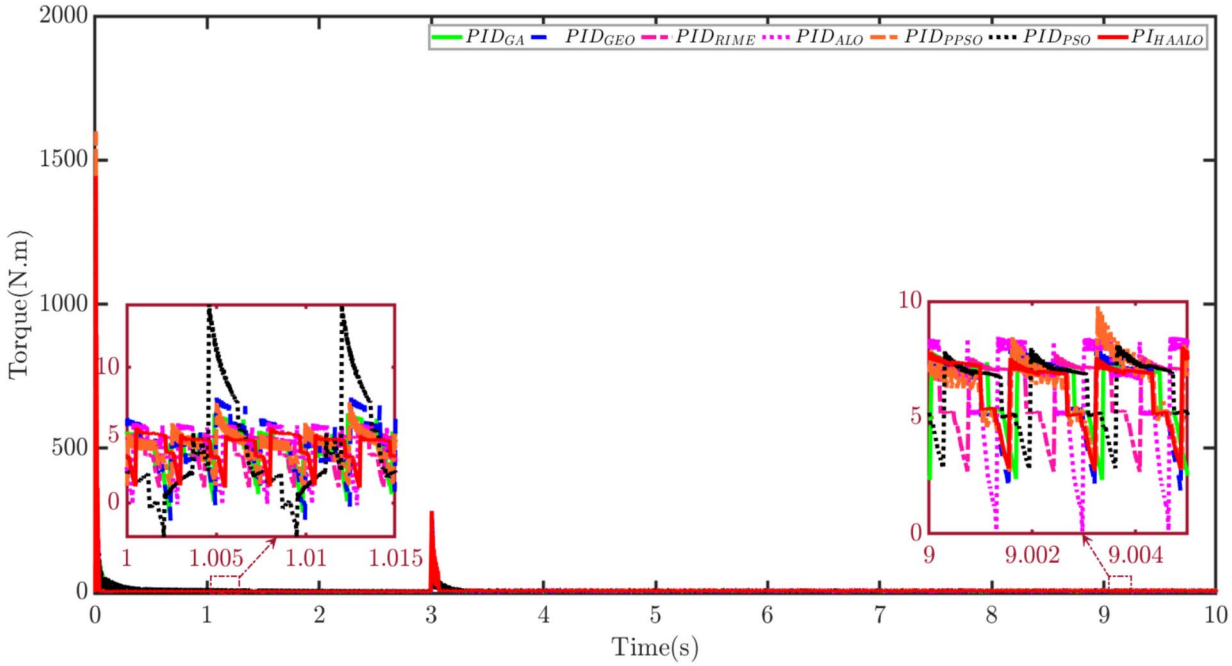


Fig. 26. Time response of total torque for PI-HAALO and other controllers in scenario 2.

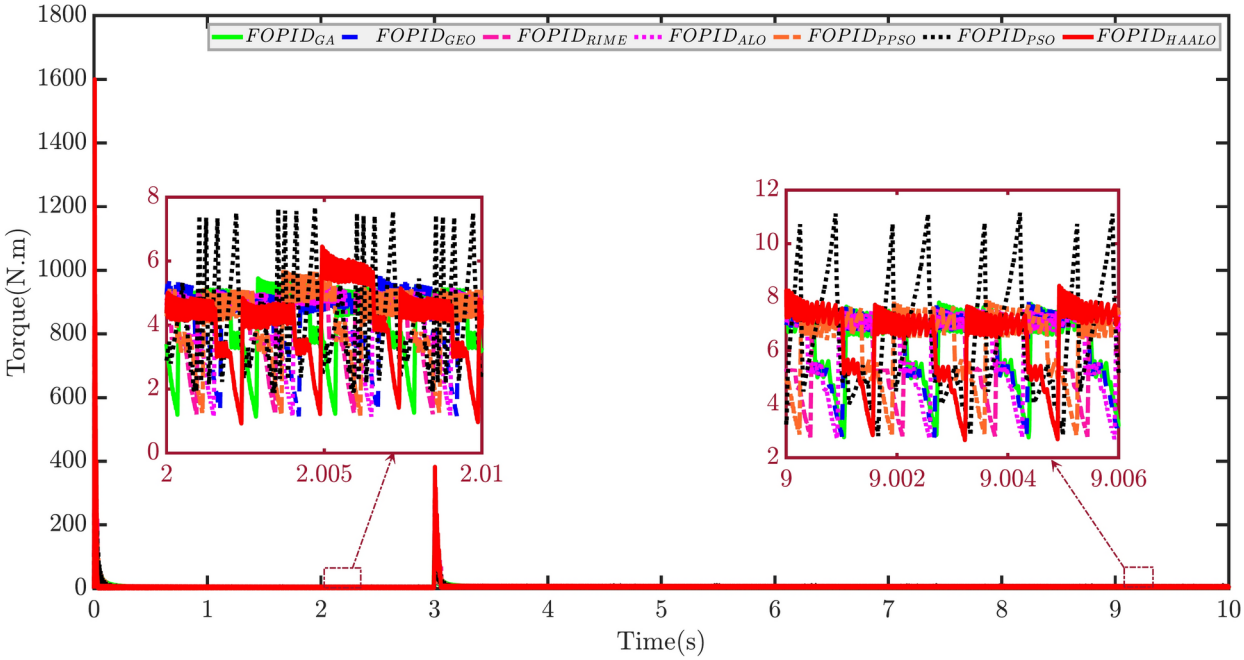


Fig. 27. Time response of total torque for FOPID-HAALO and other controllers in scenario 2.

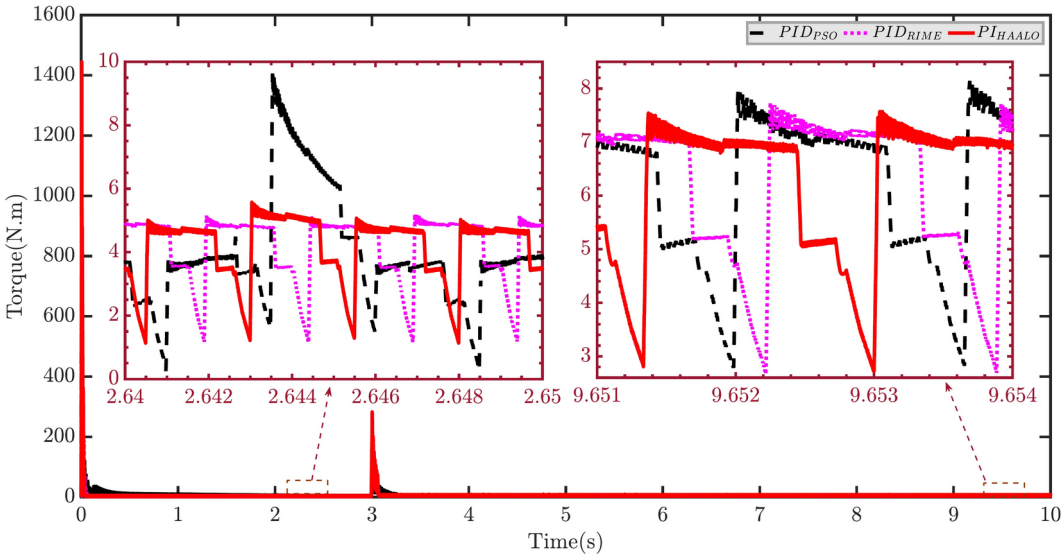


Fig. 28. Time response of total torque for PI-HAALO and two best controllers in scenario 2.

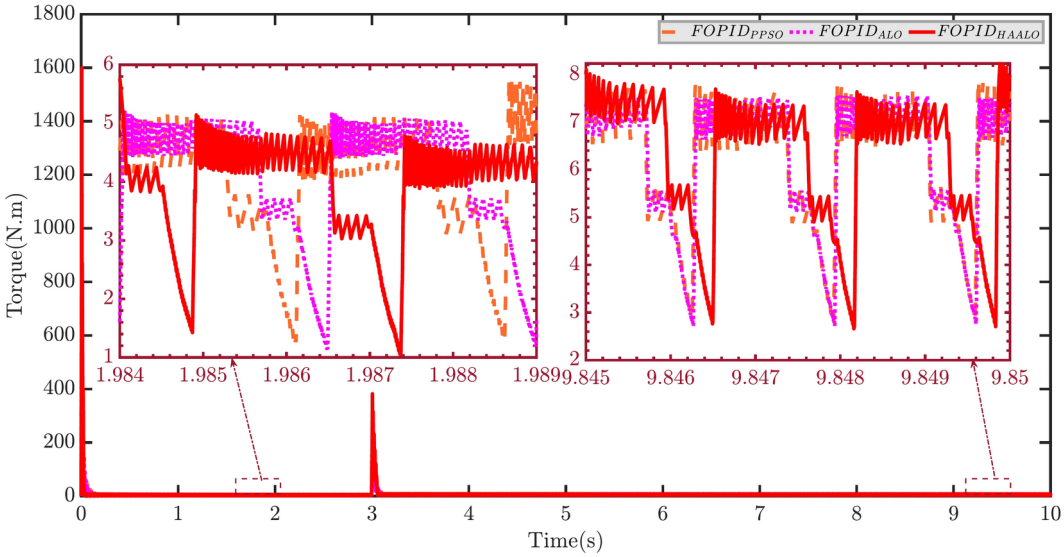


Fig. 29. Time response of total torque for FOPID-HAALO and two best controllers in scenario 2.

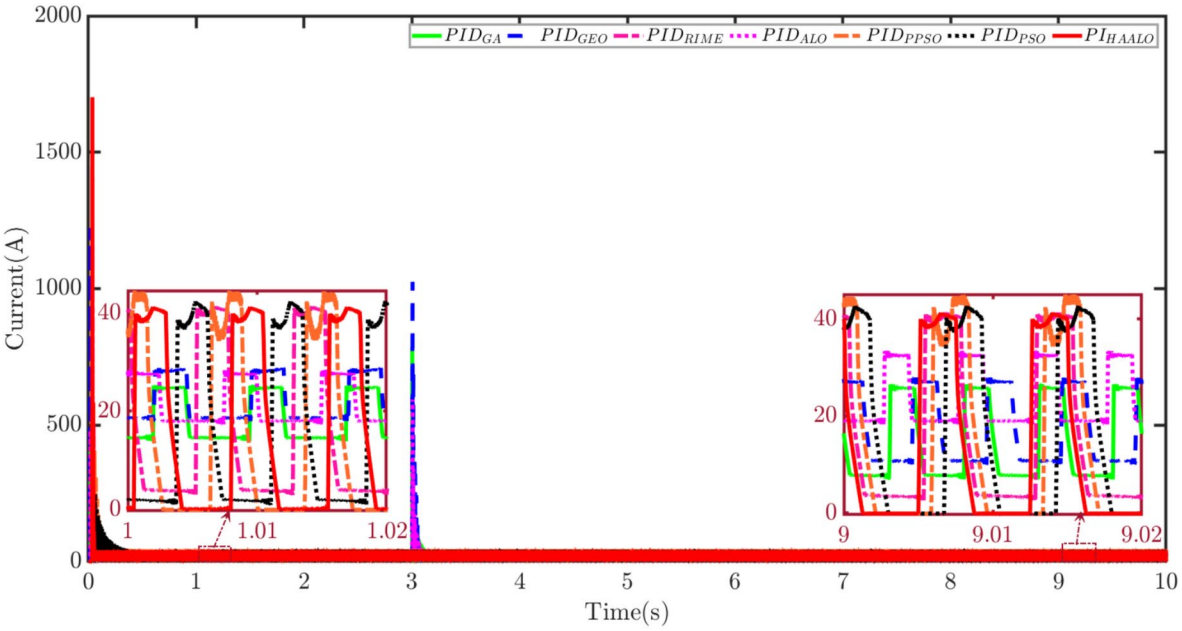


Fig. 30. Time response of phase current for PI-HAALO and other controllers in scenario 2.

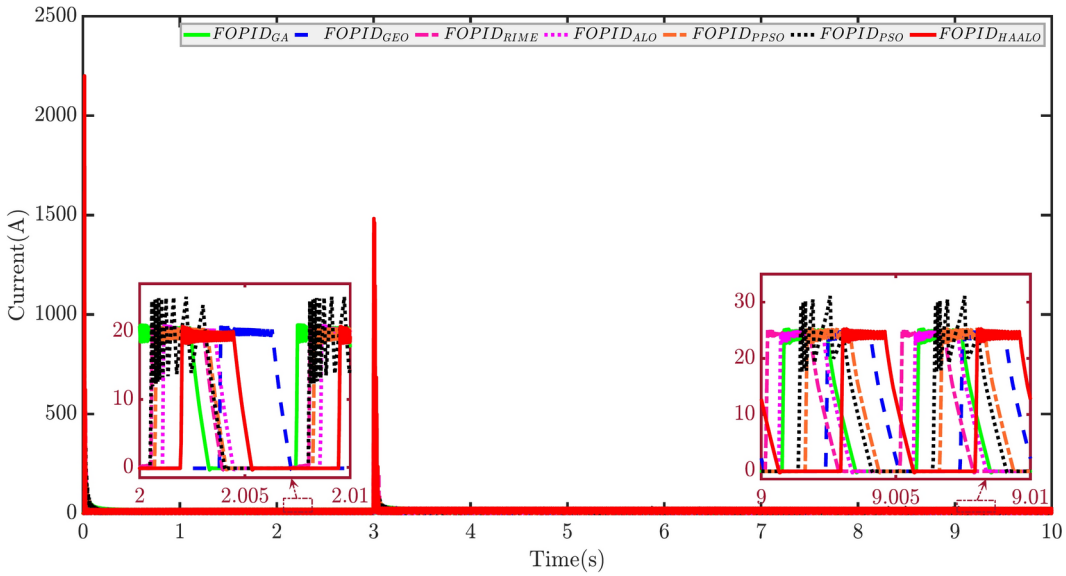


Fig. 31. Time response of phase current for FOPID-HAALO and other controllers in scenario 2.

Controller and optimization method	ISE $\times 10^{-7}$	IAE $\times 10^{-4}$	ITSE $\times 10^{-7}$	ITAE $\times 10^{-4}$
PI-HAALO	23,046	128	59,720	473
PID-PSO[12]	28,946	166	85,580	541
PID-PPSO[12]	85,246	155	255,500	540
PID-ALO[12]	2372	387	8627	1905
PID-RIME	6944	145	15,720	628
PID-GEO	9524	330	22,070	1407
PID-GA[12]	2012	273	5128	1096
FOPID-HAALO	18,432	121	45,360	435
FOPID-PSO[12]	1217	101	2774	428
FOPID-PPSO[12]	2387	100	6308	443
FOPID-ALO[12]	199	83	5053	339
FOPID-RIME	3506	85,340	7665	327
FOPID-GEO	6167	102	17,550	373
FOPID-GA[12]	1362	93	3189	374

Table 15. Time-based indicators illustrating the dynamic response of the SRM in scenario 2.

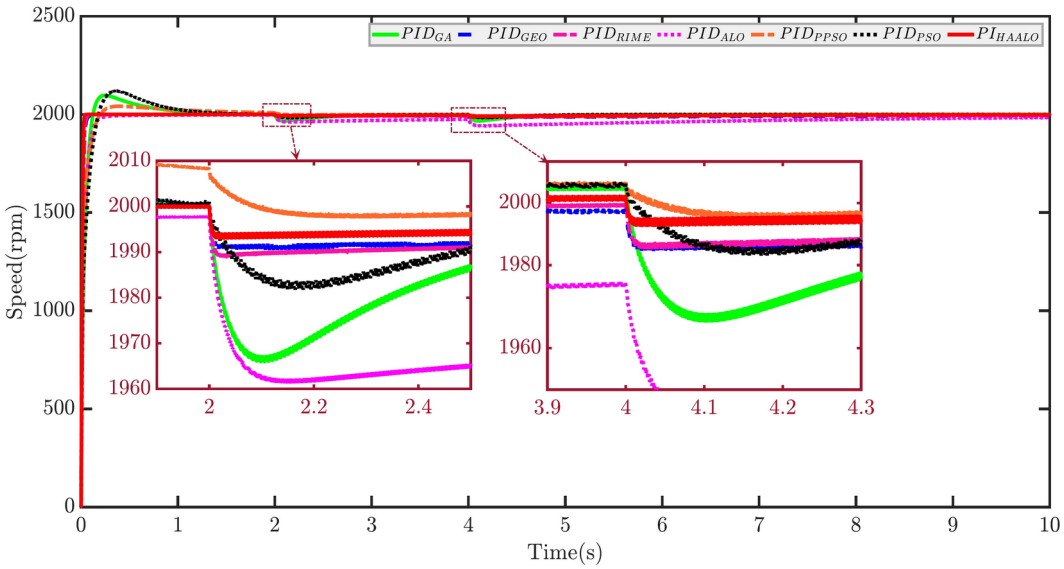


Fig. 32. Time response of speed for PI-HAALO and other controllers in scenario 3.

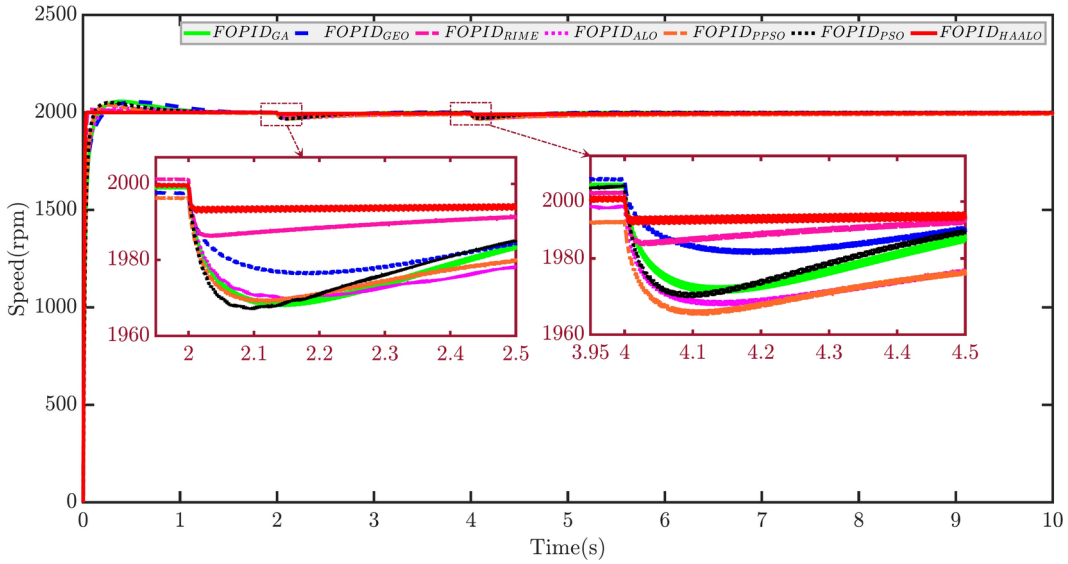


Fig. 33. Time response of speed for FOPID-HAALO and other controllers in scenario 3.

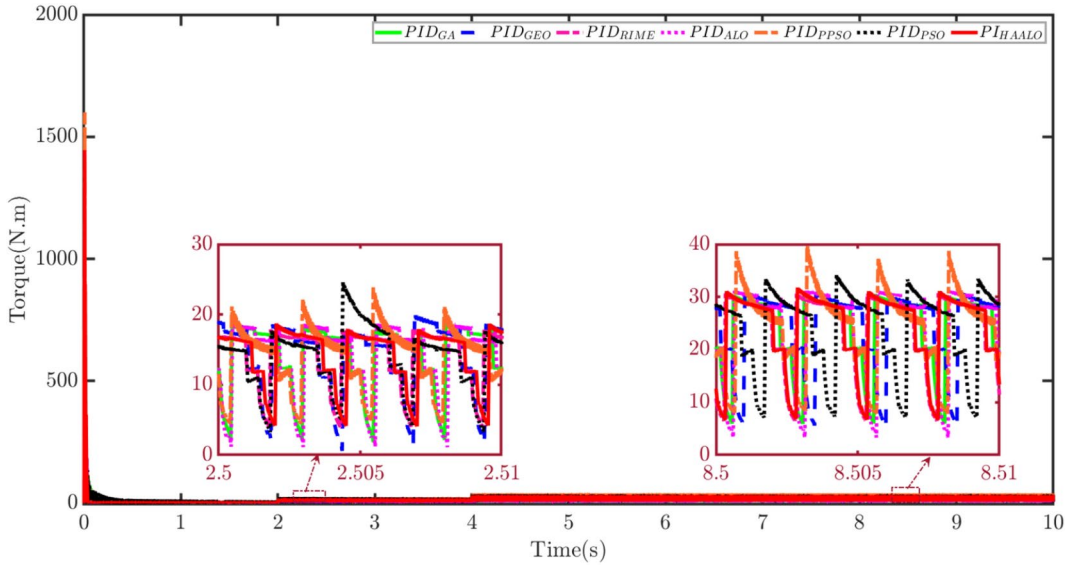


Fig. 34. Time response of total torque for PI-HAALO and other controllers in scenario 3.

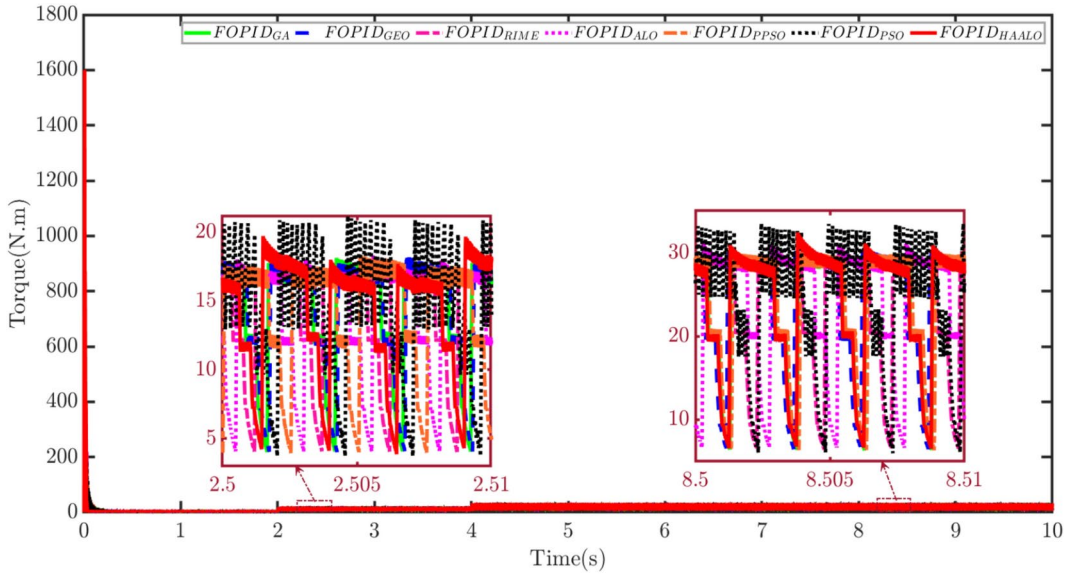


Fig. 35. Time response of total torque for FOPID-HAALO and other controllers in scenario 3.

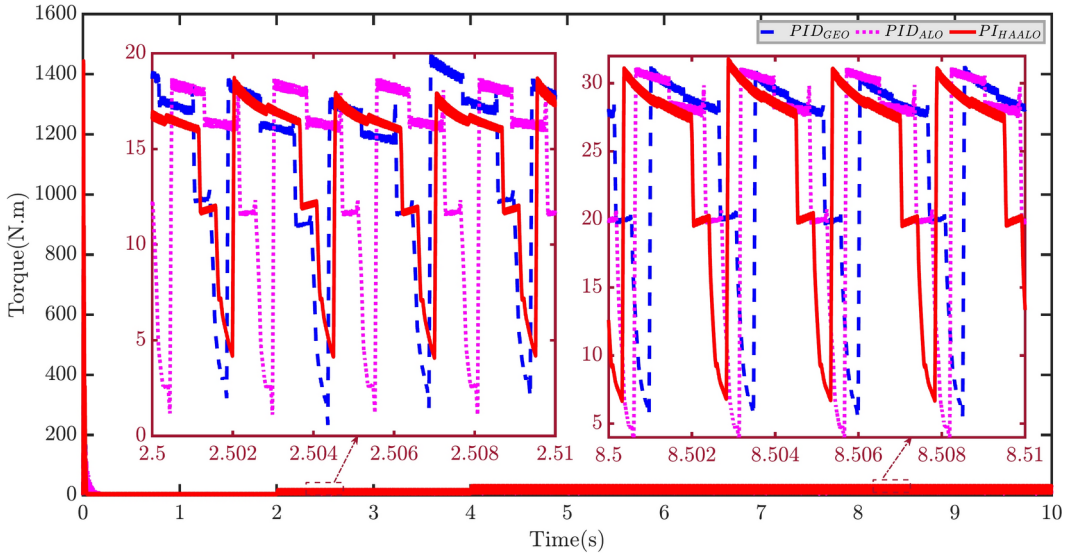


Fig. 36. Time response of total torque for PI-HAALO and two best controllers in scenario 3.

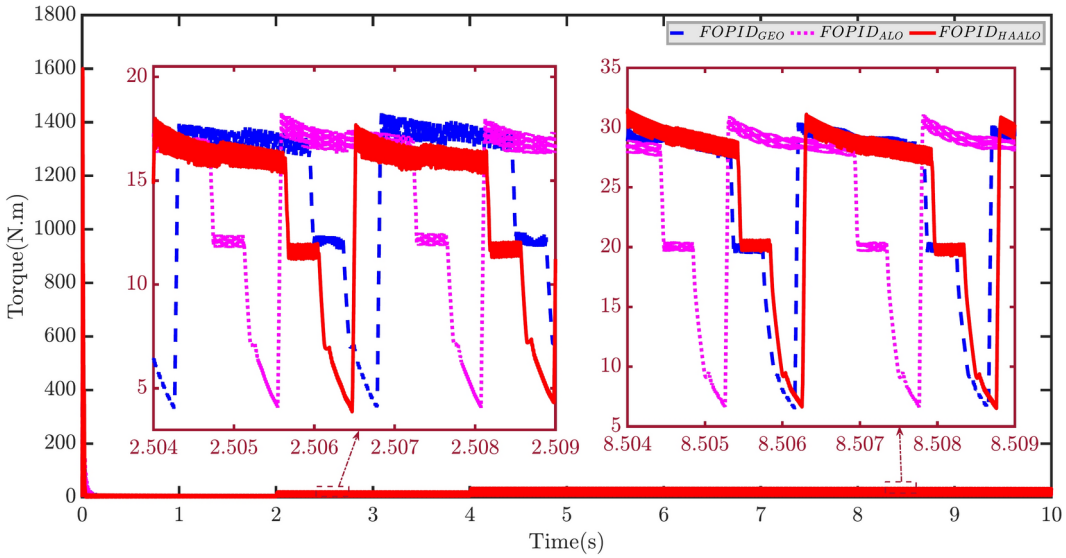


Fig. 37. Time response of total torque for FOPID-HAALO and two best controllers in scenario 3.

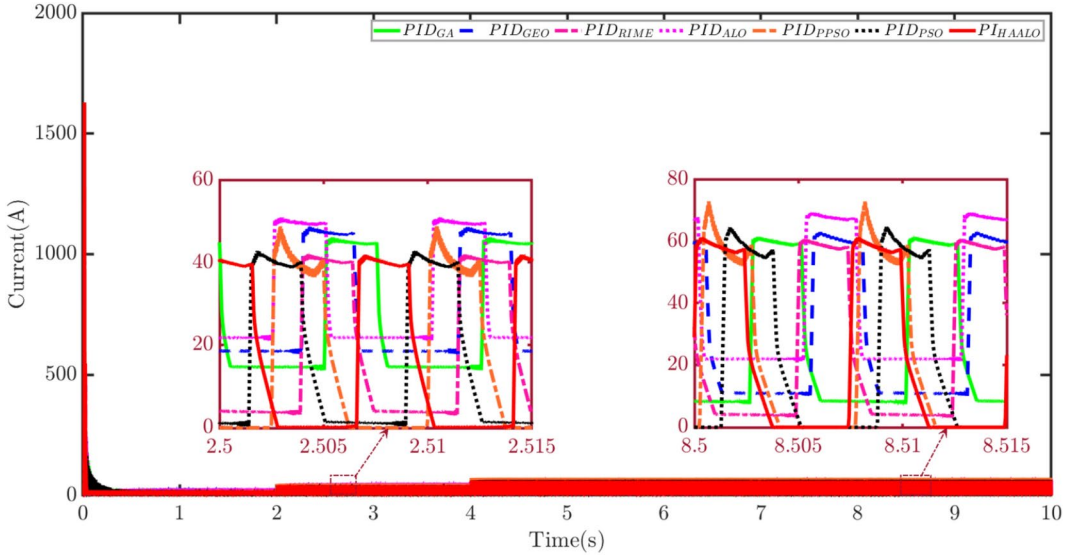


Fig. 38. Time response of phase current for PI-HAALO and other controllers in scenario 3.

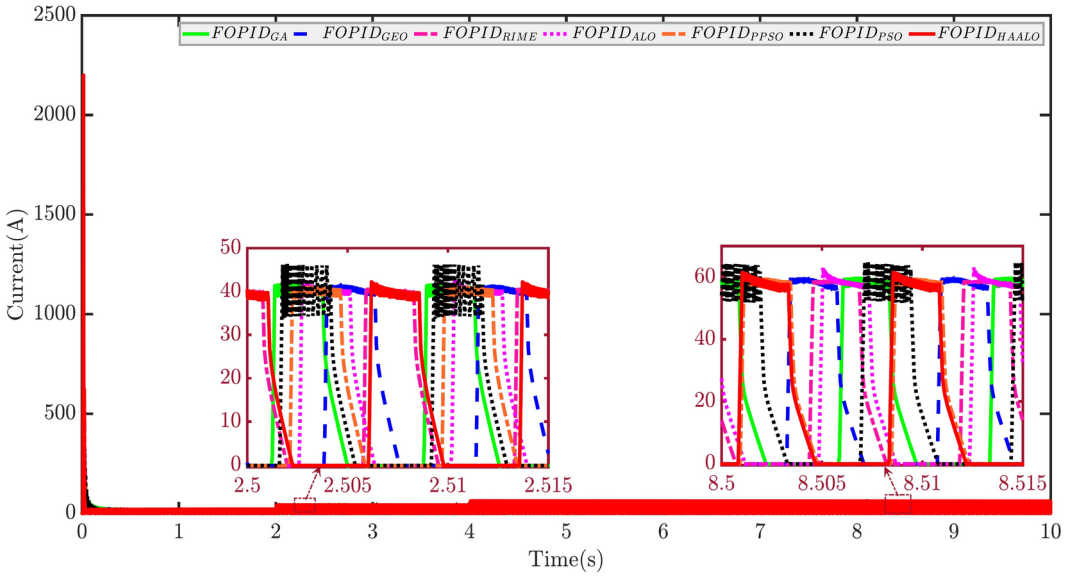


Fig. 39. Time response of phase current for FOPID-HAALO and other controllers in scenario 3.

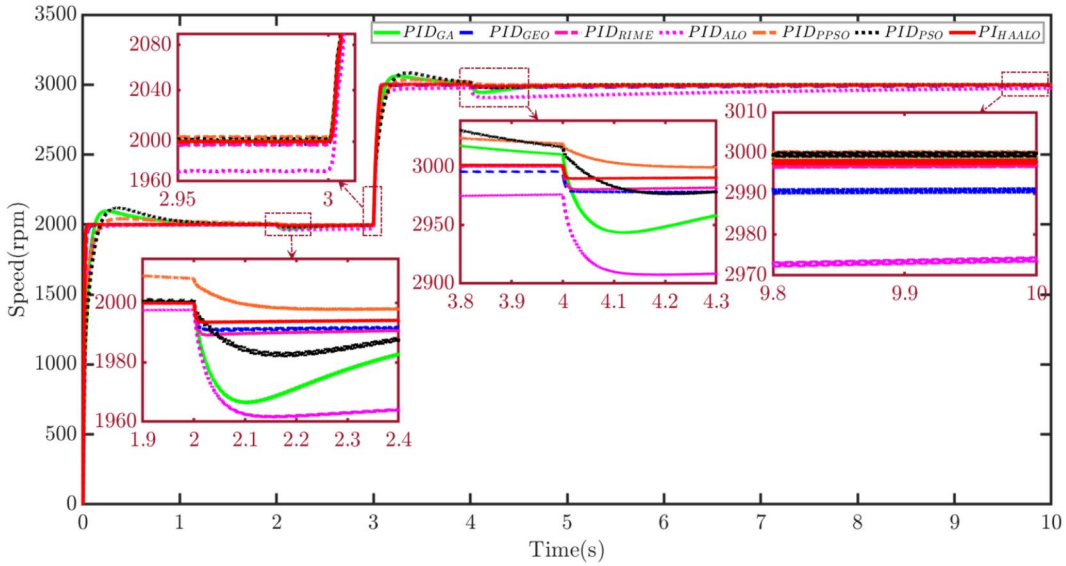


Fig. 40. Time response of speed for PI-HAALO and other controllers in scenario 4.

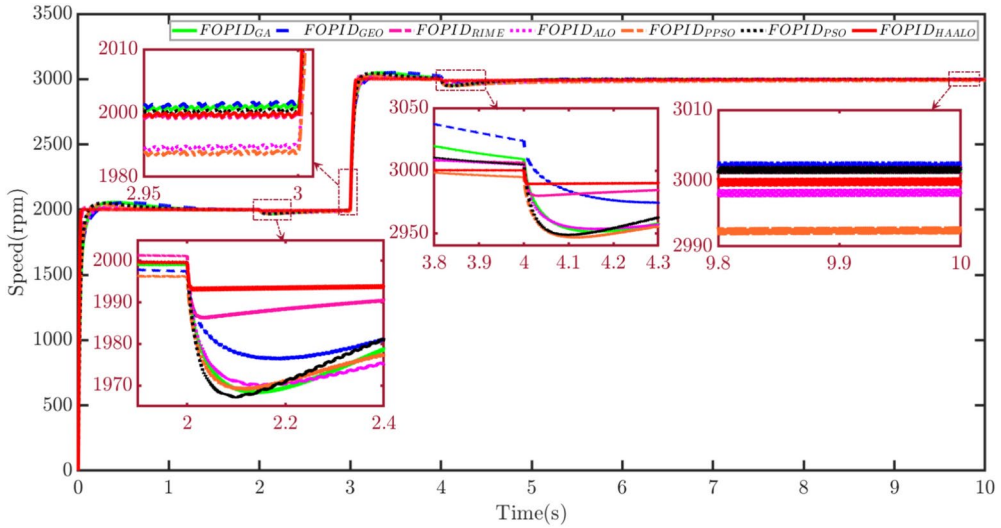


Fig. 41. Time response of speed for FOPID-HAALO and other controllers in scenario 4.

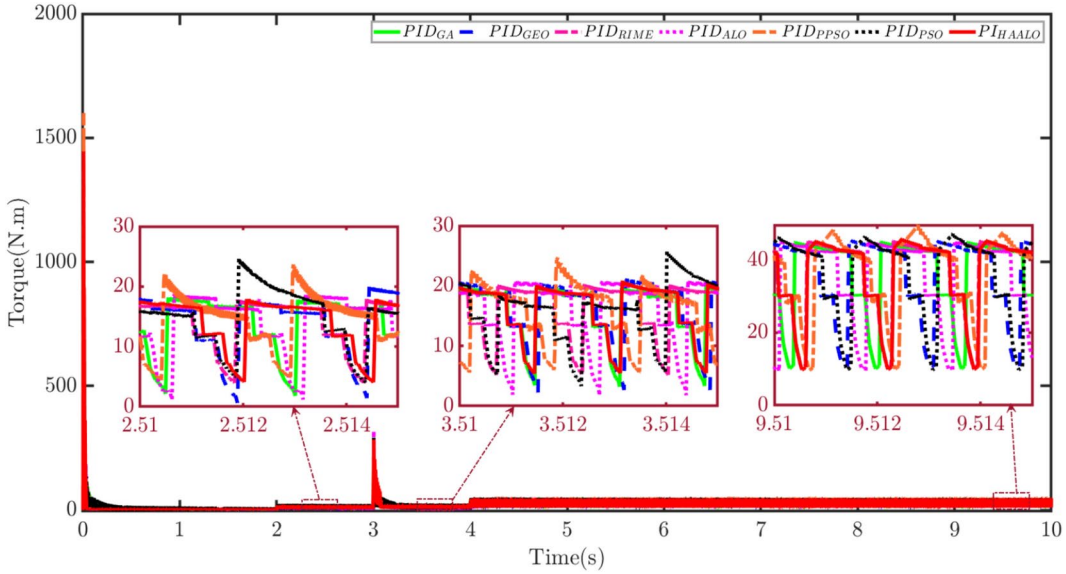


Fig. 42. Time response of total torque for PI-HAALO and other controllers in scenario 4.

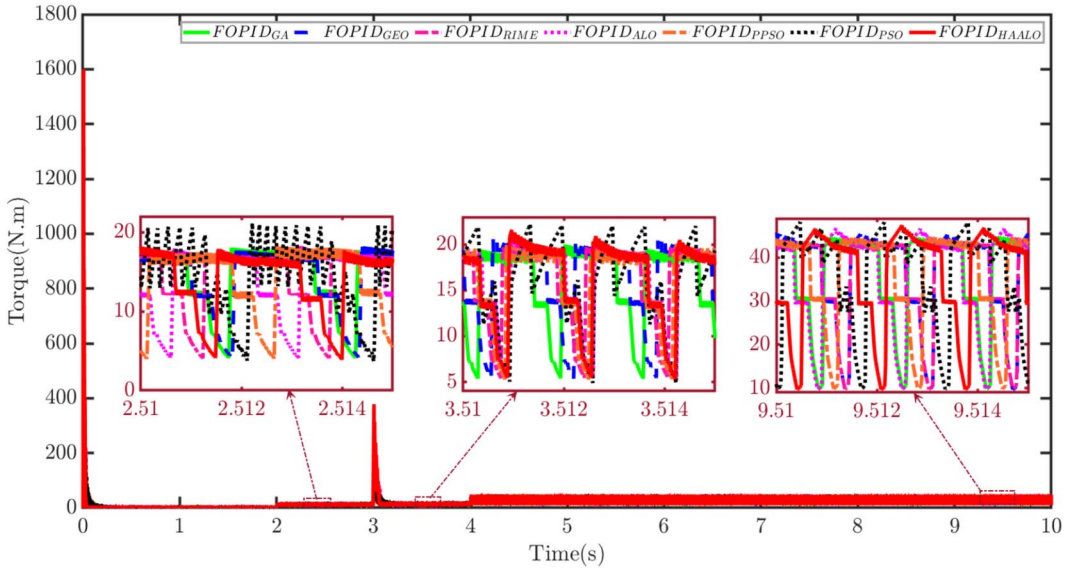


Fig. 43. Time response of total torque for FOPID-HAALO and other controllers in scenario 4.

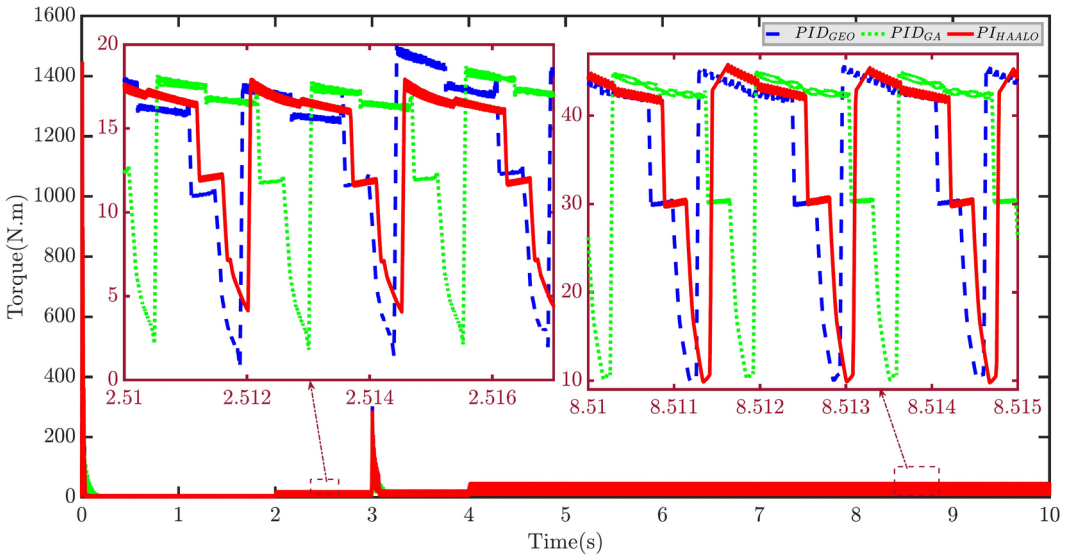


Fig. 44. Time response of total torque for PI-HAALO and two best controllers in scenario 4.

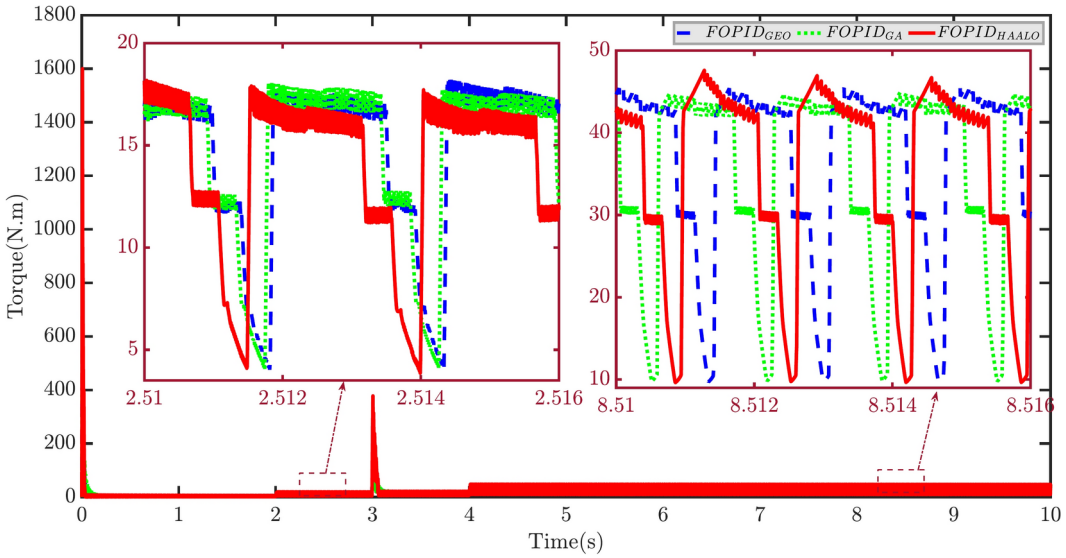


Fig. 45. Time response of total torque for FOPID-HAALO and two best controllers in scenario 4.

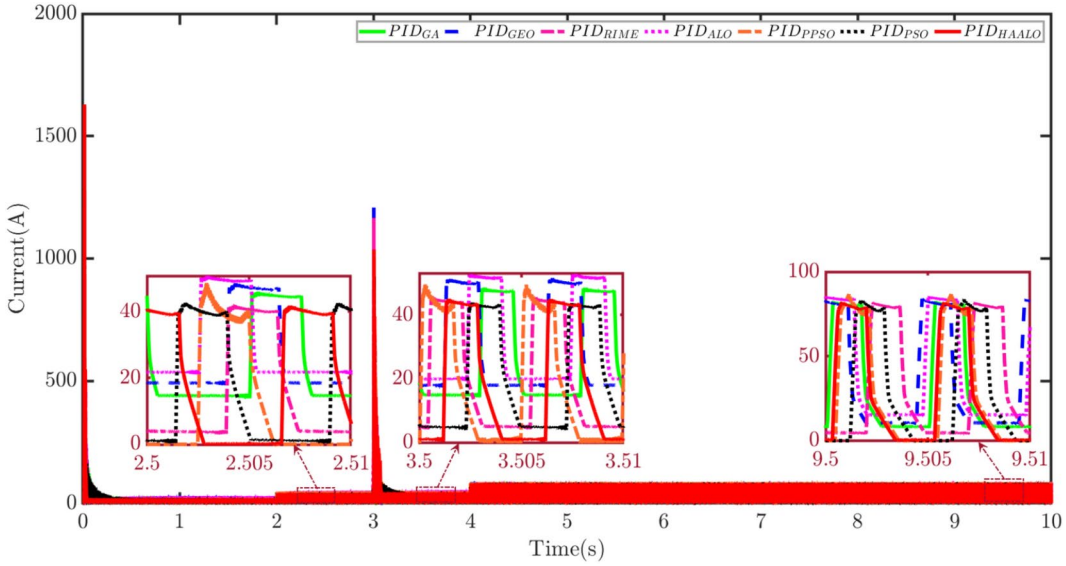


Fig. 46. Time response of phase current for PI-HAALO and other controllers in scenario 4.

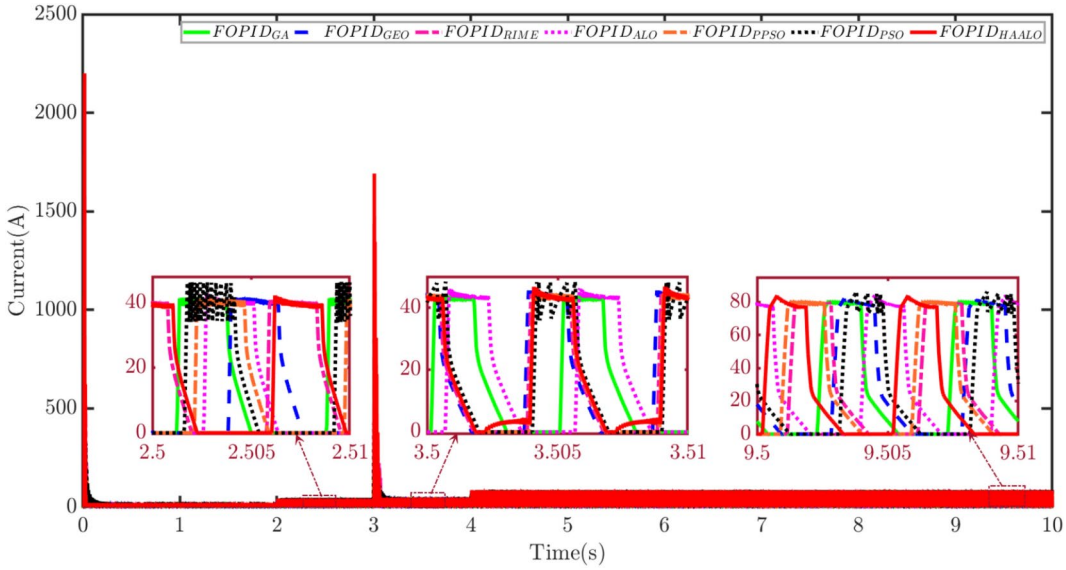


Fig. 47. Time response of phase current for FOPID-HAALO and other controllers in scenario 4.

Data availability

The datasets used and/or analyzed during the current study available from the corresponding author on reasonable request.

Received: 13 November 2024; Accepted: 2 April 2025
Published online: 15 April 2025

References

1. Fang, G. et al. Advanced control of switched reluctance motors (SRMs): A review on current regulation, torque control and vibration suppression. *IEEE Open J. Ind. Electron. Soc.* **2**, 280–301 (2021).
2. Abdel-Aziz, A., Elgenedy, M. & Williams, B. Review of switched reluctance motor converters and torque ripple minimisation techniques for electric vehicle applications. *Energies* **17**(13), 3263 (2024).
3. Husain, I. Minimization of torque ripple in SRM drives. *IEEE Trans. Ind. Electron.* **49**(1), 28–39 (2002).
4. Gan, C. et al. A review on machine topologies and control techniques for low-noise switched reluctance motors in electric vehicle applications. *IEEE Access* **6**, 31430–31443 (2018).
5. Borase, R. P., Maghade, D., Sondkar, S. & Pawar, S. A review of PID control, tuning methods and applications. *Int. J. Dyn. Control* **9**, 818–827 (2021).
6. Kotb, H., Yakout, A. H., Attia, M. A., Turky, R. A. & AboRas, K. M. Speed control and torque ripple minimization of SRM using local unimodal sampling and spotted hyena algorithms based cascaded PID controller. *Ain Shams Eng. J.* **13**(4), 101719 (2022).
7. Senthil Murugan, L. & Maruthupandi, P. Sensorless speed control of 6/4-pole switched reluctance motor with ANFIS and fuzzy-PID-based hybrid observer. *Electr. Eng.* **102**, 831–844 (2020).
8. Kumar, C. V., Ram, A. R. & Rao, G. M. Design and development of remora optimization based controller for speed management in three-phase brushless DC motor. *NeuroQuantology* **20**(16), 1901 (2022).
9. Ren, P., Zhu, J., Jing, Z., Guo, Z. & Xu, A. Minimization of torque ripple in switched reluctance motor based on MPC and TSF. *IEEJ Trans. Electr. Electron. Eng.* **16**(11), 1535–1543 (2021).
10. Masoudi, S., Mehrjerdi, H. & Ghorbani, A. Adaptive control strategy for velocity control of a linear switched reluctance motor. *IET Electr. Power Appl.* **14**(8), 1496–1503 (2020).
11. Alharkan, H. Torque ripple minimization of variable reluctance motor using reinforcement dual NNs learning architecture. *Energies* **16**(13), 4839 (2023).
12. Jabari, M. & Rad, A. Optimization of speed control and reduction of torque ripple in switched reluctance motors using metaheuristic algorithms based PID and FOPID controllers at the edge. *Tsinghua Sci. Technol.* (2024).
13. Kumar, M. N. & Chidanandappa, R. Particle swarm optimization technique for speed control and torque ripple minimization of switched reluctance motor using PID and FOPID controllers. *Int. J. Inf. Technol.* **16**(2), 1185–1201 (2024).
14. Al-Amyal, F., Számel, L. & Hamouda, M. An enhanced direct instantaneous torque control of switched reluctance motor drives using ant colony optimization. *Ain Shams Eng. J.* **14**(5), 101967 (2023).
15. Abualigah, L., Shehab, M., Alshinwan, M., Mirjalili, S. & Elaziz, M. A. Ant lion optimizer: A comprehensive survey of its variants and applications. *Arch. Comput. Methods Eng.* **28**, 1397–1416 (2021).
16. Aguilar-Mejía, O., Minor-Popocatl, H. & Tapia-Olvera, R. Comparison and ranking of metaheuristic techniques for optimization of PI controllers in a machine drive system. *Appl. Sci.* **10**(18), 6592 (2020).
17. da Cunha Reis, M. R. et al. Optimized techniques for driving and control of the switched reluctance motor to improve efficiency. *Control Eng. Pract.* **90**, 1–18 (2019).
18. Divandari, M., Rezaie, B. & Noei, A. R. Speed control of switched reluctance motor via fuzzy fast terminal sliding-mode control. *Comput. Electr. Eng.* **80**, 106472 (2019).
19. Gengaraj, M., Kalaivani, L. & Rajesh, R. Investigation on torque sharing function for torque ripple minimization of switched reluctance motor: A flower pollination algorithm based approach. *IETE J. Res.* **69**(6), 3678–3692 (2023).
20. Rajendran, A. & Karthik, B. Design and analysis of fuzzy and PI controllers for switched reluctance motor drive. *Mater. Today Proc.* **37**, 1608–1612 (2021).
21. Mostafa Jabari, M. A. N., Mohammad, Z., Sanjeevikumar, P. & Muyeen, S.M. Designing a FOPI(1+PIDn) multi-stage controller for super-fast speed control and torque ripple reduction in switched reluctance motors, inpress. *IET Electron. Lett.* (2025).

22. Costa, P. R. M., De Paula, M. V., Neto, P. J. D. S. & Barros, T. A. D. S. A novel LQI-based speed control of switched reluctance motors for high performance applications. *IEEE Access* (2025).
23. Mirjalili, S. The ant lion optimizer. *Adv. Eng. Softw.* **83**, 80–98 (2015).
24. Hu, G., Guo, Y., Wei, G. & Abualigah, L. Genghis Khan shark optimizer: A novel nature-inspired algorithm for engineering optimization. *Adv. Eng. Inform.* **58**, 102210 (2023).

Acknowledgements

This research is funded by European Union under the REFRESH—Research Excellence For Region Sustainability and High-Tech Industries Project via the Operational Programme Just Transition under Grant CZ.10.03.01/00/22_003/0000048; in part by the National Centre for Energy II and ExPEDite Project a Research and Innovation Action to Support the Implementation of the Climate Neutral and Smart Cities Mission Project TN02000025; and in part by ExPEDite through European Union's Horizon Mission Programme under Grant 101139527. The authors would like to express their sincere gratitude to Stanislav Misak for his exceptional supervision, project administration, and overall guidance throughout the course of this project. His expertise and support were instrumental to its success.

Author contributions

Mostafa Jabari, Davut Izci: Conceptualization, Methodology, Software, Visualization, Investigation, Writing-Original draft preparation. Serdar Ekinci, Lukas Prokop: Data curation, Validation, Supervision, Resources, Writing—Review & Editing. Mohit Bajaj, Vojtech Blazek: Project administration, Supervision, Resources, Writing—Review & Editing.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to M.B.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

© The Author(s) 2025