RESEARCH ARTICLE

Improved GOA-based fuzzy PI speed control of PMSM with predictive current regulation

Taochang Li*, Ang Li, Limin Hou

Faculty of Electrical and Control Engineering, Liaoning Technical University, Huludao, Liaoning, China

* litaochang@163.com

Citation: Li T, Li A, Hou L (2025) Improved GOA-based fuzzy PI speed control of PMSM with predictive current regulation. PLoS ONE 20(1): e0318094. https://doi.org/10.1371/journal.pone.0318094

Editor: Chiranjit Sain, Ghani Khan Choudhury Institute of Engineering and Technology, INDIA

Received: February 8, 2024

Accepted: January 8, 2025

Published: January 30, 2025

Data availability statement: All relevant data are within the manuscript and its Supporting Information files.

Funding: The authors received funding from the National Natural Science Foundation of China under Grant 52177047 and the Huludao Science and Technology Project under Grant 2024JH(2)2/05b.

Abstract

To address the susceptibility of conventional vector control systems for permanent magnet synchronous motors (PMSMs) to motor parameter variations and load disturbances, a novel control method combining an improved Grasshopper Optimization Algorithm (GOA) with a variable universe fuzzy Proportional-Integral (PI) controller is proposed, building upon standard fuzzy PI control. First, the diversity of the population and the global exploration capability of the algorithm are enhanced through the integration of the Cauchy mutation strategy and uniform distribution strategy. Subsequently, the fusion of Cauchy mutation and opposition-based learning, along with modifications to the optimal position, further improves the algorithm's ability to escape local optima. The improved GOA is then employed to optimize the contraction-expansion factor of the variable universe fuzzy PI controller, achieving enhanced control performance for PMSMs. Additionally, to address the high torque and current ripple issues commonly associated with traditional PI controllers in the current loop, Model Predictive Control (MPC) is adopted to further improve control performance. Finally, experimental results validate the effectiveness of the proposed control scheme, demonstrating precise motor speed control, rapid and stable current tracking, as well as improved system robustness.

1. Introduction

The Permanent Magnet Synchronous Motor (PMSM) has attracted significant attention due to its compact structure, high efficiency, rapid dynamic response, and exceptional speed control performance [1]. In recent years, the application of PMSMs has expanded rapidly across various fields, including industrial automation, electric transportation, renewable energy systems, and other sectors, largely driven by advancements in permanent magnet material technologies. However, like other electric motors, PMSMs face challenges related to parameter uncertainties and load disturbances during practical operation. Conventional fixed-parameter Proportional-Integral-Derivative (PID) control strategies often fall short of meeting the stringent requirements of modern industrial control systems in terms of performance and robustness [2,3].

Consequently, numerous innovative techniques have been proposed by researchers to improve the control performance of PMSMs. These methods include fuzzy control [4,5], model predictive control [6,7], sliding mode control [8,9], among others. Additionally,

Competing interests: The authors have declared that no competing interests exist.

there has been increasing interest in leveraging intelligent algorithms and neural networks in the control of PMSM systems, as evidenced by recent studies [10–17]. For instance, in reference [18], an adaptive fuzzy PI controller is introduced for the vector control system of PMSMs. This approach dynamically adjusts the proportional and integral parameters based on the system error and error rate, thereby effectively enhancing the control performance of PMSMs. However, conventional fuzzy PI control methods often operate within fixed domains, which limits their ability to accommodate complex and dynamic system behaviors. To address this limitation, reference [19] proposes a particle swarm optimization (PSO)-based fuzzy PI control method as a parameter optimization strategy. In this method, the proportional factor and quantization factor of the fuzzy PI controller are optimized through iterative updates of the PSO algorithm, resulting in improved convergence accuracy. This approach enhances the response speed and robustness of PMSMs under varying operational conditions.

In a rigorous examination of motor control methodologies, reference [20] investigates the application of the Extreme Learning Machine (ELM) technique to enhance the dynamic performance and robustness of the system by accurately estimating system disturbances. Building upon this, reference [21] introduces a novel approach that integrates the ELM with the Artificial Bee Colony (ABC) algorithm. This hybrid technique is applied within the PMSM fuzzy PI speed control system, yielding significant reductions in settling time and marked improvements in steady-state accuracy. Nonetheless, while intelligent algorithms and neural networks present promising avenues for advancing motor control, they often encounter challenges such as parameter tuning and algorithm convergence. These issues increase the complexity of implementing such methodologies in practical applications [22].

In summary, this manuscript proposes an enhanced GOA-based fuzzy PI speed control strategy for PMSMs, incorporating predictive current regulation to optimize the performance of high-efficiency PMSM speed control systems. The key contributions of this study are summarized as follows:

(1) This study introduces an improved control framework that integrates an enhanced variable domain fuzzy PI controller with a model predictive current controller (MPCC) in a cascaded structure to achieve high-performance speed regulation for PMSMs.

(2) An enhanced GOA is developed by incorporating mixed Cauchy variation and uniform distribution to address the issues of premature convergence and local optima. Furthermore, a variable domain fuzzy PI speed controller is designed for PMSMs, enabling dynamic adjustments of the fuzzy control domain through the improved GOA, thereby enhancing control flexibility and robustness.

The structure of this manuscript is organized as follows: Section 2 presents the design of the improved GOA-based fuzzy PI speed controller for PMSMs. Section 3 introduces the model predictive current control methodology. Simulation and experimental results are discussed in Section 4. Finally, the conclusions of the study are summarized in Section 5.

2. Improved GOA-based fuzzy PI speed controller design for PMSM

2.1. Mathematical model of PMSM

The primary objective of this study revolves around a surface-mounted PMSM. Specifically, the mathematical representation of the motor within the d-q coordinate system is presented as (1).

$$\begin{cases} u_d = R_s i_d + L\dfrac{di_d}{dt} - \omega_e L i_q \\ u_q = R_s i_q + L\dfrac{di_q}{dt} + \omega_e (L i_d + \psi_f) \end{cases} \tag{1}$$

where u_d and u_q are the stator voltage, while i_d and i_q are stator current in the d-q frame, R_s is stator resistance, L is the stator inductance, ω_e is the electric angular velocity, ψ_f is permanent magnet flux linkage.

2.2. Variable domain fuzzy PI control

The conventional PI controllers have a limitation in achieving online parameter tuning, resulting in the inability to adapt to different operating conditions and hindering the improvement of their dynamic performance. To overcome this limitation and enhance the system's dynamic performance and disturbance rejection ability, a fuzzy PI self-tuning controller is developed by integrating the principles of fuzzy control theory with the conventional PI strategy, as illustrated in Fig 1.

The conventional fuzzy PI control method involves the continuous monitoring of the system's states, and then computes the velocity error e and its rate of change e_c within the motor system, which are then subjected to fuzzification. utilizing the pre-designed fuzzy rules, a fuzzy inference is carried out. The next step is defuzzification, which converts the fuzzy quantities into precise values. Subsequently, this process effectively enables real-time adjustment of the PI controller's parameters. The variables e, e_c, ΔK_p and ΔK_i make use of the identical fuzzy subsets {NB, NM, NS, ZO, PS, PM, PB}.

However, the arrangement of fuzzy rules heavily depends on user expertise. While a large number of rules can accurately depict system outputs, it also introduces complexity into the system. On the other hand, a limited number of rules may simplify the system, but it may compromise the effectiveness of control.

The variable domain fuzzy PI does not necessitate extensive domain-specific expert knowledge. Instead, it only requires a general understanding of the rule trends. The main concept behind this approach is to dynamically adjust the domain of the fuzzy controller by utilizing a scaling factor. The scaling factor's variation is directly proportional to the variable deviation.

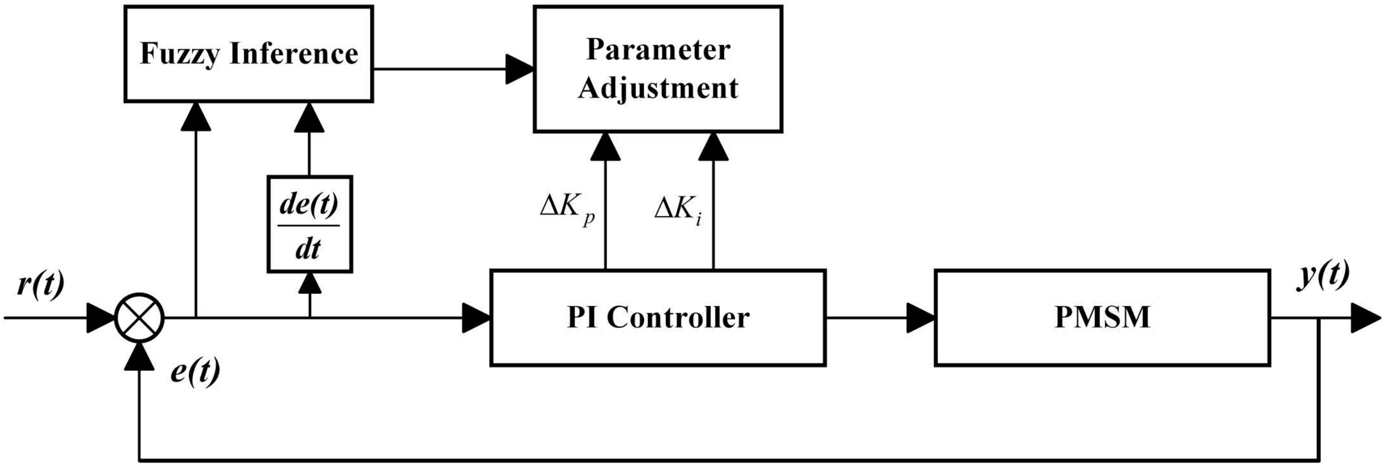

Fig 1. Schematic diagram of Fuzzy PI controller.

https://doi.org/10.1371/journal.pone.0318094.g001

Fig 2 presents a schematic diagram illustrating this concept, where α represents the scaling factor, and $[E, - E]$ represents the domain of the fuzzy controller. As the input to the controller increases or decreases, the scaling factor adjusts accordingly. Currently, there is no standardized form for the scaling factor in variable domain fuzzy control algorithms. Typically, the scaling factor of the domain is designed based on a function as shown:

$$\alpha(x) = 1 - \lambda e^{-kx^2}, \lambda \in (0,1) \tag{2}$$

where $k > 0$ is the adjustment coefficient of the expansion factor.

In light of the challenges associated with determining the optimal values for parameter, the paper employs an improved GOA to optimize the scaling factor. Consequently, a variable domain fuzzy PI controller is developed based on the improved GOA algorithm. The domains of the input and output variables are defined as shown:

$$\begin{cases} X_E = \ [-\alpha(e)E, \alpha(e)E] \\ X_{E_c} = [-\beta(e_c)E_c, \beta(e_c)E_c] \end{cases} \tag{3}$$

2.3. Improved GOA-based variable domain fuzzy PI control

The GOA is an innovative heuristic algorithm that draws inspiration from nature. It mimics the behavior of grasshoppers, where the adult grasshoppers conduct a wide-scale search while the nymphs engage in small-scale movements, both ultimately converging towards a food source. This corresponds precisely to the global exploration and local exploitation aspects of heuristic algorithms. The process of the grasshopper swarm searching for resources is viewed as the process of seeking the optimal solution [23].

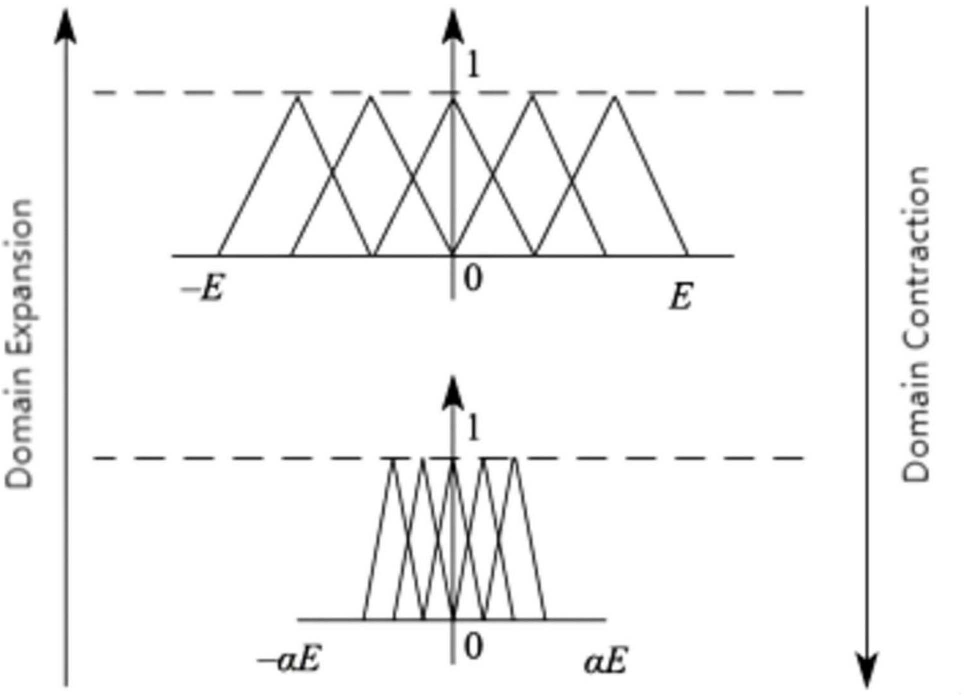

Fig 2. Schematic diagram of variable domain principle.

https://doi.org/10.1371/journal.pone.0318094.g002

During the exploration phase, it is essential for the algorithm to exhibit strong random behavior in order to thoroughly explore the entire search space. Moreover, the initial solutions should demonstrate a wide range of variations, thereby increasing the likelihood of uncovering the region where the optimal solution is expected to reside. Subsequently, in the exploitation phase, the algorithm narrows down the search space, leading to smaller variations in the search process, thus facilitating a more refined exploration. Furthermore, achieving an appropriate balance between the exploration and exploitation phases through the utilization of an improved GOA algorithm is crucial for effectively identifying the optimal solution to a given optimization problem. This approach helps to address the limitations of the original GOA, including its vulnerability to local optima and premature convergence. The specific implementation process involves the following steps.

1) Initialize the population space size N, space dimension, maximum iteration count T_{max}, as well as parameters c_{max} and c_{min}. Additionally initialize the population positions $X_i i = (1, 2 \ldots N)$.

2) Designing fitness functions. In the context of the PMSM speed control system, the objective is to develop a fitness function that effectively reduces both overshoot and rise time. To achieve this simultaneous optimization, the fitness function can encompass the integration of the absolute value of the error as the primary performance indicator. Moreover, to ensure that the control input remains within acceptable limits, a squared term of the control input is incorporated into the fitness function. This results in the following fitness function Eq (4).

$$F = \omega_1 \int_0^\infty \tau \left| e(\tau) \right| d\tau + \omega_2 \int_0^\infty \tau u^2(\tau) d\tau \tag{4}$$

where ω_1 and ω_2 are the weight coefficients for the cumulative speed deviation and the cumulative input square, and $\omega_1 + \omega_2 = 1$.

The setting of these weights is grounded in empirical considerations, and the ultimate determination of weights for the objective fitness function is achieved through experimentation involving various combinations of weights.

3) Calculate individual fitness. The optimal value is stored in the target value T_d. Furthermore, the parameters c and positions of the grasshoppers are to be updated. The calculation formulas for the conventional GOA are denoted by Eq (5) and (6).

$$c = c_{\max} - t \frac{c_{\max} - c_{\min}}{T_{\max}} \tag{5}$$

$$X_i^d = c\left(\sum_{j=1, j \neq i}^{N} c \frac{ub_d - lb_d}{2} s\left(\left| x_j^d - x_i^d \right| \right) \frac{x_j - x_i}{d_{ij}} \right) + T_d \tag{6}$$

where and are the maximum and minimum values of the adaptive parameter. is the current iteration count. is the position of the $i - th$ grasshopper in the $d - th$ dimension. and are the upper and lower bounds of the d-dimensional variable. T_d is the target position. s is the inter-individual impact function. d_{ij} is the distance between the $i - th$ grasshopper and the $j - th$ grasshopper in the $d - th$ dimension. and c is the key to coordinating exploration.

According (5), it can be seen that the function value decreases with the iteration count, which does not meet the practical exploration and adaptation optimization process. Therefore, this paper introduces a random number γ following a uniform distribution to dynamically improve the variation of *c*. By introducing the uniform distribution number γ through the improved function, early-stage exploration can be more comprehensive, and at the same time, it accelerates the convergence speed of the algorithm in the later stages. The ability to modify the parameters dynamically facilitates the acceleration of algorithm development. The Eq (5) is rewritten as (7).

$$c = \gamma(\lg \frac{c_{\max}}{T_{\max}})^{-20t} + \exp(\frac{t}{T_{\max}})^{-20(c_{\max} - c_{\min})} \tag{7}$$

4) Using the Cauchy operator and a segmented approach. The process of updating individual positions is accomplished using the Cauchy operator and a segmentation approach. During the initial stage of the algorithm, the optimal solution of each generation is stored in memory, and the Eq (6) is enhanced as (8) for better performance.

$$X_{\text{new}}(t+1) = a_1 \times W + a_2 \times r_1(X_{best}(t) - X_i(t)) + a_2 \times r_2(X_j(t) - X_k(t)) \tag{8}$$

where $X_{new}(t + 1)$ is the position of the $i - th$ grasshopper in the $(t + 1) - th$ generation. $X_{best}(t)$ is the optimal position in the $t - th$ generation. $X_j(t)$ and $X_k(t)$ are two random positions in the $t - th$ generation. r_1 and r_2 are two random numbers generated in the range [0,1]. a_1 is the memory coefficient. a_2 is the information exchange coefficient.

$$a_1 = \exp(c(t) - 30\frac{t}{T_{\max}})^{-t} \tag{9}$$

$$a_2 = (\frac{T_{\max} - t}{T_{\max}})^N \tag{10}$$

$$W = \sum_{j=1, j \neq i}^{N} c\frac{ub_d - lb_d}{2} s(\left|x_j^d - x_i^d\right|)\frac{x_j - x_i}{d_{ij}}) \tag{11}$$

The enhanced algorithm incorporates the entirety of position information, thereby facilitating the exchange of information among individuals. Furthermore, the exploration aspect is expanded through the random selection of two individuals to spearhead the position update. The latter half of the algorithm introduces the Cauchy operator for the mutation operation, enabling the algorithm to break free from local optima, preventing premature convergence and accelerating the convergence rate. The formula is as follows.

$$X_i^d(t+1) = X_i^d(t) + Cauchy \oplus (X_{best}^d(t) - X_k^d(t)) \tag{12}$$

where Cauchy is the Cauchy operator. $X_k^d(t)$ is the random position of the $k - th$ grasshopper in the $d - th$ dimension. The algorithm update is determined by a random probability P_1. When P_1 is less than $T_{\max} / 2$ individual position updates are performed by using Eq (8), (9), (10), and (11). Otherwise, updates are done by using Eq (12).

5) Incorporating Cauchy mutation. The introduction of Cauchy mutation serves as an active mechanism to perturb and update the target position in order to prevent the algorithm from being confined to local optima.

$$X_{\text{new}}(t+1) = \text{Cauchy} \oplus X_{\text{best}}(t) \tag{13}$$

where $X_{new}(t+1)$ is the target solution for the $t+1$ generation.

In order to guarantee the superiority of the new position in comparison to the target position, a greedy mechanism is implemented subsequent to the position update operation. This mechanism involves the evaluation of the fitness of both the new and old target positions, enabling the determination of whether the position should be updated. This approach ensures that the algorithm progresses towards the desired position.

6) Selecting operation. In order to ascertain if the individual fulfills the termination condition, it is necessary to determine whether this condition is met. If indeed it is, the global optimal solution should be outputted. On the other hand, if the condition is not met, it is necessary to repeat steps 3) to 5) until the algorithm reaches the maximum number of iterations.

The flowchart of the proposed improved GOA is shown in Fig 3.

By incorporating the improved GOA with the variable-domain fuzzy PI controller, the scaling factor is dynamically modified according to the input variables and their rates of change. This leads to the generation of more appropriate PI parameter adjustments, which are then combined with the predefined PI parameters. Finally, these optimized parameters are fed into the controller to complete the process of parameter optimization. In the speed control loop of PMSM, the schematic representation of the variable-domain fuzzy PI controller model, which is based on the improved GOA, is depicted in Fig 4.

3. Model predictive current control

In conventional vector control systems, the utilization of a PI controller in the current loop gives rise to certain challenges, including sluggish dynamic response and elevated current harmonics. To overcome these limitations, this scholarly paper adopts the MPC approach in the current loop instead of the conventional PI controller, aiming to attain enhanced control performance. By regarding the motor current as state variables, Eq (1) can be altered as (14).

$$\begin{cases} \dfrac{\mathrm{d}i_d}{\mathrm{d}t} = u_d / L - R_s \mathrm{i}_\mathrm{d} / L + \omega_e i_q \\ \dfrac{\mathrm{d}i_q}{\mathrm{d}t} = u_q / L - R_s \mathrm{i}_\mathrm{q} / L - \omega_e \psi_f / L - \omega_e i_d \end{cases} \tag{14}$$

In practical systems, it becomes challenging to avoid calculation delays due to the impact of hardware and numerical control. To ensure that the system control performance remains unaffected, it becomes imperative to incorporate a compensation mechanism with a delay of one sample. By discretizing Eq (14) with a sufficiently short sampling time, we can derive the equation for compensating the current with a one-sample delay as presented in (15).

$$\begin{cases} i_d^p(k+1) = (1 - \dfrac{T_s R_s}{L}) i_d(k) + T_s \omega_e i_q(k) + \dfrac{T_s}{L} u_d(k) \\ i_q^p(k+1) = (1 - \dfrac{T_s R_s}{L}) i_q(k) - T_s \omega_e i_d(k) + \dfrac{T_s}{L} u_q(k) - \dfrac{T_s \omega_e \psi_f}{L} \end{cases} \tag{15}$$

The compensating current acquired is utilized to substitute the current within the existing prediction model. Consequently, with the integration of one-sample delay compensation, the current prediction model is subsequently revised as (16).

$$\begin{cases} i_d^p(k+2) = (1 - \dfrac{T_s R_s}{L}) i_d(k+1) + T_s \omega_e i_q(k+1) + \dfrac{T_s}{L} u_d(k+1) \\ i_q^p(k+2) = (1 - \dfrac{T_s R_s}{L}) i_q(k+1) - T_s \omega_e i_d(k+1) + \dfrac{T_s}{L} u_q(k+1) - \dfrac{T_s \omega_e \psi_f}{L} \end{cases} \tag{16}$$

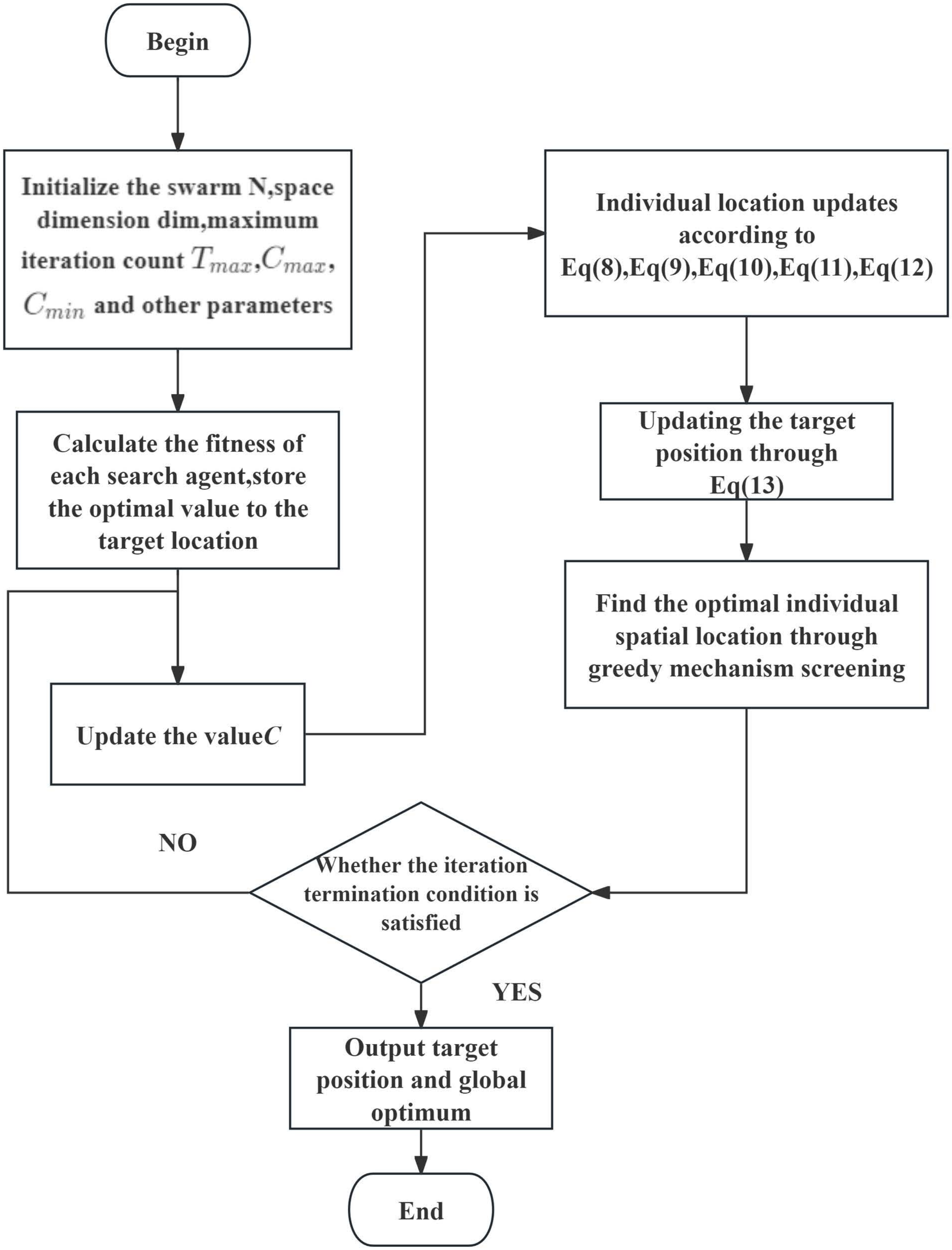
Begin
Initialize the swarm N,space dimension dim,maximum iteration count T_{max},C_{max}, C_{min} and other parameters
Calculate the fitness of each search agent,store the optimal value to the target location
Update the value$C$
Individual location updates according to Eq(8),Eq(9),Eq(10),Eq(11),Eq(12)
Updating the target position through Eq(13)
Find the optimal individual spatial location through greedy mechanism screening
NO
Whether the iteration termination condition is satisfied
YES
Output target position and global optimum
End

Fig 3. Flowchart of the proposed improved GOA.

https://doi.org/10.1371/journal.pone.0318094.g003

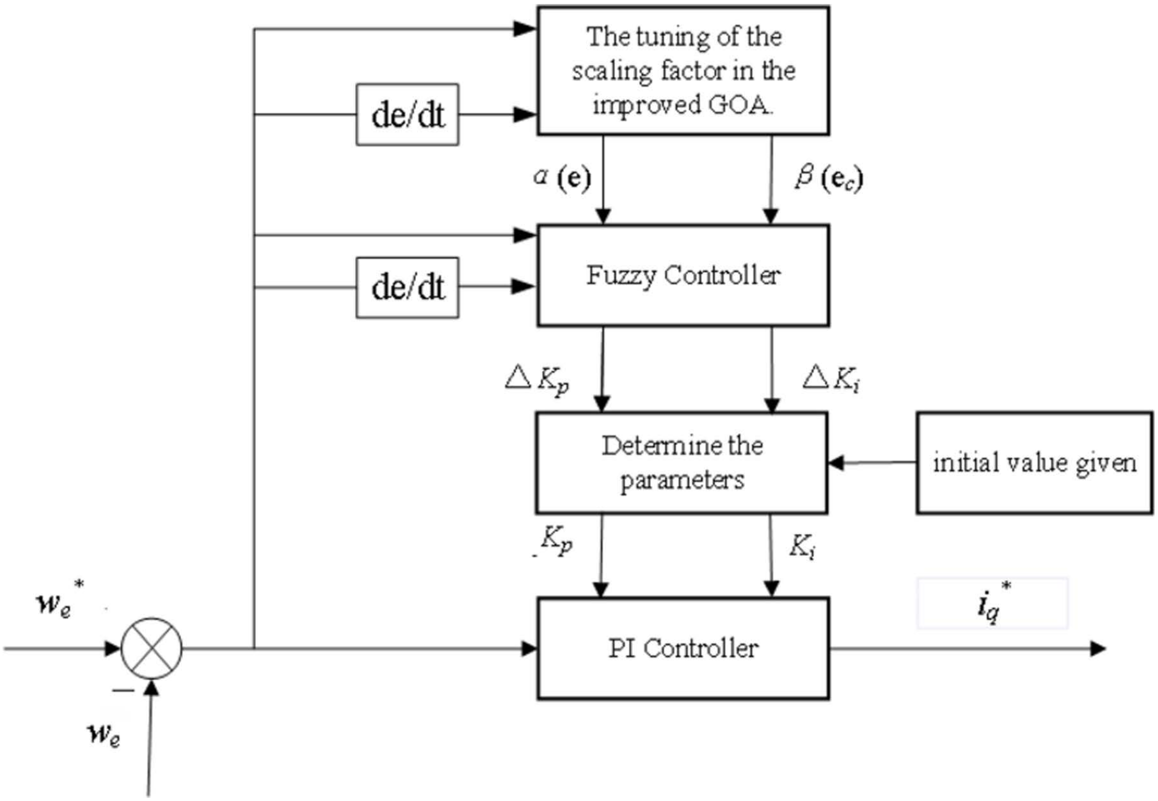

Fig 4. Schematic diagram of improved GOA-based Fuzzy PI controller.

https://doi.org/10.1371/journal.pone.0318094.g004

The objective of the one-step ahead model predictive current control (MPCC) is to minimize the tracking error at the subsequent sampling point. Consequently, the cost function for the conventional MPCC is expressed as:

$$J = (i_d^* - i_d^p(k+2))^2 + (i_q^* - i_q^p(k+2))^2 \tag{17}$$

The cost function is minimized by the prediction current, which in turn corresponds to the optimal voltage vector. This optimal voltage vector represents the most desirable output at the current time.

4. Simulation and Experiment Results

4.1. Simulation Verification

The framework for the control strategy of the PMSM speed control system, as presented in this research, is depicted in Fig 5. In order to validate the effectiveness of the proposed strategy, simulations were conducted within the Simulink environment. The study involved a comparative analysis of three control strategies in regards to load variations, speed adjustments, and rotation direction changes. Scheme 1, referred to as Fuzzy PI, employs the conventional fuzzy PI for speed control and traditional PI control for current control. The parameters of PI controller are $K_p = 30$, $K_i = 500$. Scheme 2 is the model predictive current control strategy of PMSM with GOA variable universe fuzzy PI speed controller. Scheme 3, known as improved GOA-Fuzzy PI, utilizes the control method proposed in this paper.

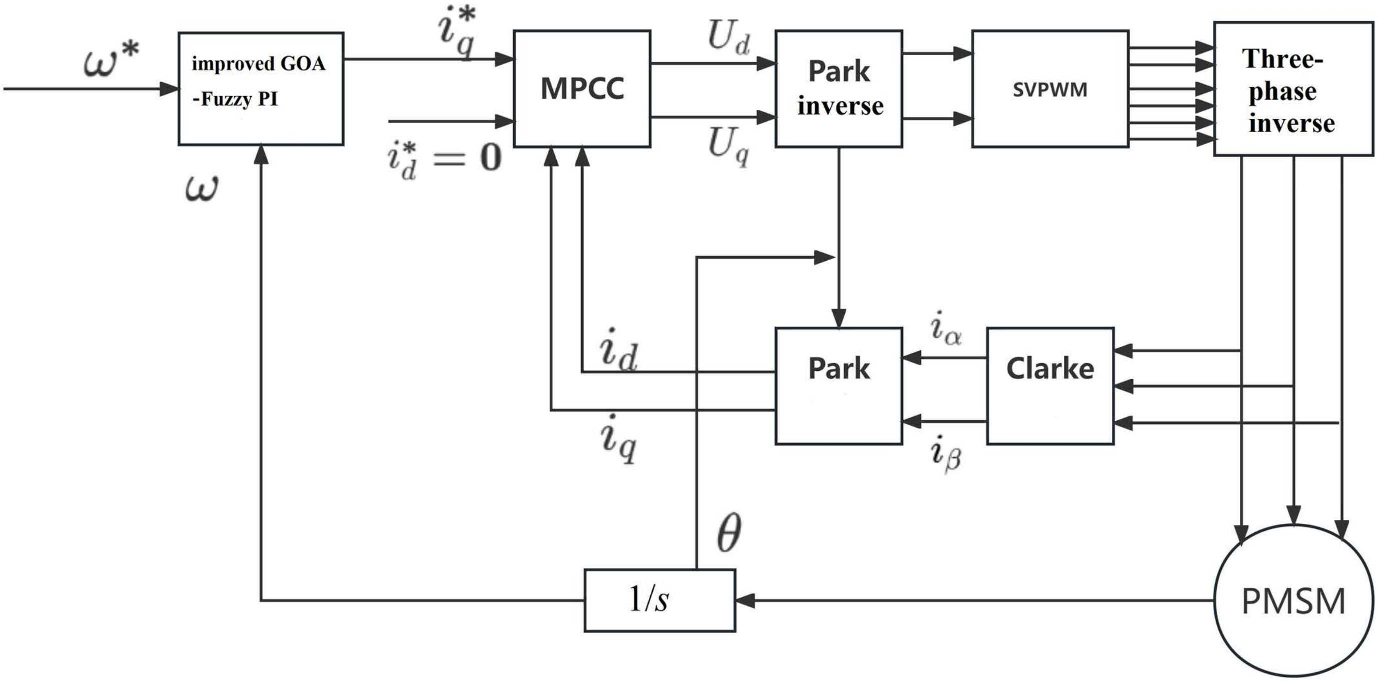

Fig 5. Schematic diagram of PMSM control system.

https://doi.org/10.1371/journal.pone.0318094.g005

The simulation experiment employs the GOA algorithm with predetermined parameters. These parameters include a population size of $N=30$, a maximum iteration count of $T_{\max} = 100$, and the variables to be optimized is $d = 4$, $c_{max} = 1$ and $c_{min} = 2 \times 10^{-5}$. Moreover, the weights in the fitness function are $\omega_1 = 0.9$ and $\omega_2 = 0.1$.

The comparison chart depicted in Fig 6 presents the algorithm fitness values at 800 r/min under load. It becomes evident that the improved version of the GOA displays notable improvements in terms of both convergence speed and optimization capability when compared to the conventional GOA.

The parameters of PMSM, which are used in Simulation, are listed in Table 1.

Firstly, the motor is started without any external load, operating at a prescribed speed of 300 r/min. Subsequently, the load of 10 N•m is imposed upon the motor at 0.5 seconds. And then the load is suddenly reduced at 1.5 seconds. The experiment was then repeated by changing the load to 20 N•m. The speed, torque and current curves for three schemes are shown in Figs 7–9.

Based on the findings depicted in Fig 7A and 7B, when the given speed is 300 r/min, scheme 1 produces about 10.3% overshoot in the start-up stage, scheme 2 and scheme 3 start smoothly without obvious overshoot. However, the rise time of scheme 2 is about 0.1s, while scheme 3 is only 0.08s, which is better in terms of rapidity. Moreover, scheme 3 has less speed overshoot and quickly returns to a steady state when dealing with different loads.

As can be seen from Fig 8, the motor torque is stabilized at 0 N•m faster because of scheme 3, and the time used is approximately 0.1 s, whereas the torque achieved using the traditional Fuzzy PI strategy requires 0.2 s to reach a stable state. In addition to this, scheme 3 enables the motor to achieve faster and consistent torque. Combined with Fig 9, it can be seen more intuitively that scheme 3 can track the change of current faster at the moment of startup and load change, and the current fluctuation is relatively small.

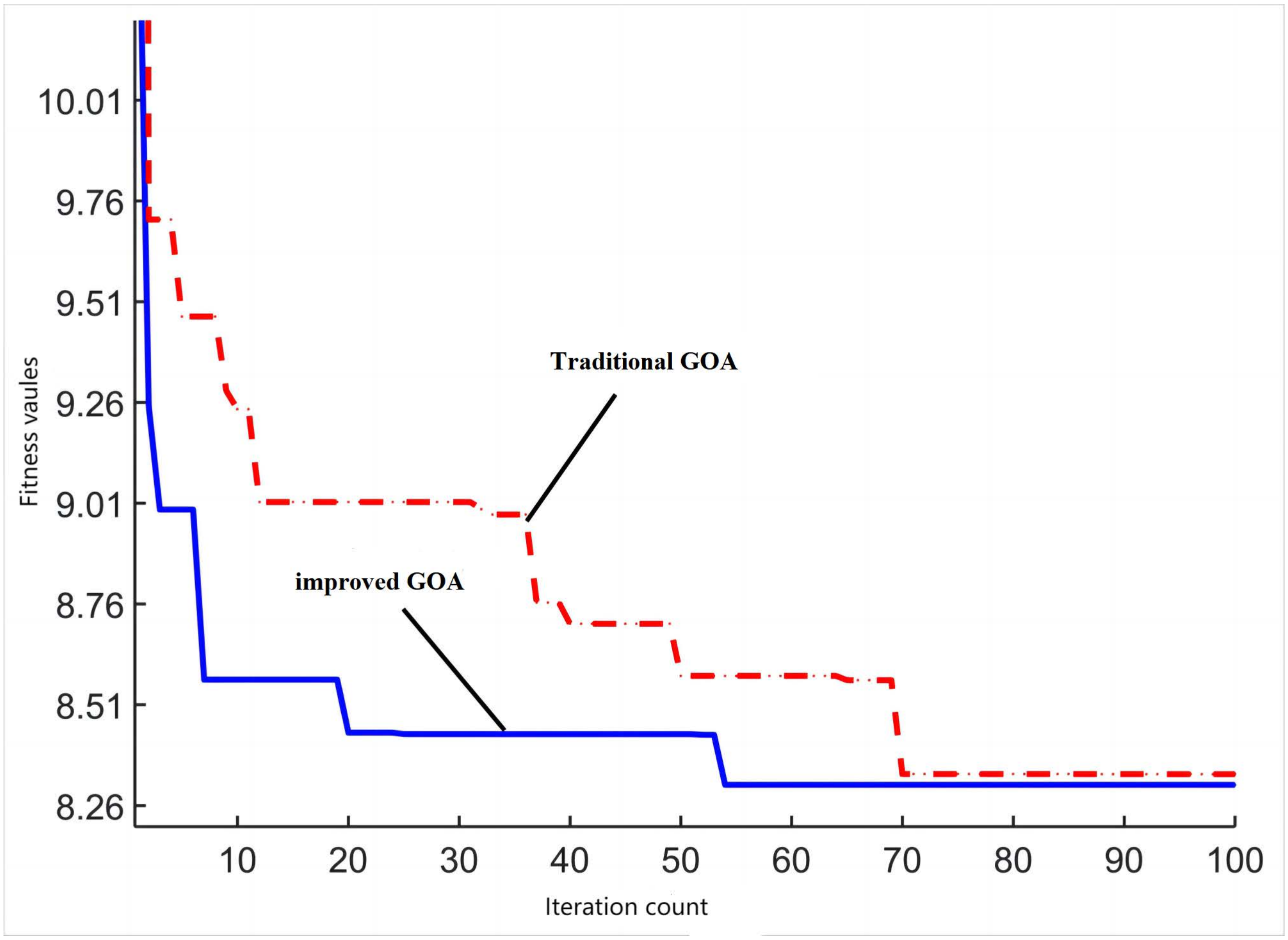

Fig 6. The comparison chart of algorithm fitness values.

https://doi.org/10.1371/journal.pone.0318094.g006

Table 1. Operation parameters of PMSM.

Parameters	Value
Stator resistance	0.346Ω
Stator inductance	7.8mH
Viscous coefficient	0.005N·s/m
Moment of inertia	0.089$kg \cdot m^2$
Parameters magnet flus linkage	0.51825Wb
Number of pole pairs	2
Rated power	10kw
Rated voltage	260V
Rated torque	63N·m
Rated speed	1500r/min

https://doi.org/10.1371/journal.pone.0318094.t001

Three control strategies were simulated under different working conditions. First, the motor starts running at a constant speed of 300 r/min and no-load torque. Then, the motor is loaded with a constant load torque of 10 N • m at 0.5 s. Finally, at 1 s, the given speed changes suddenly, and the speed is changed from 300 r/min to 400 r/min, but is then readjusted to 300

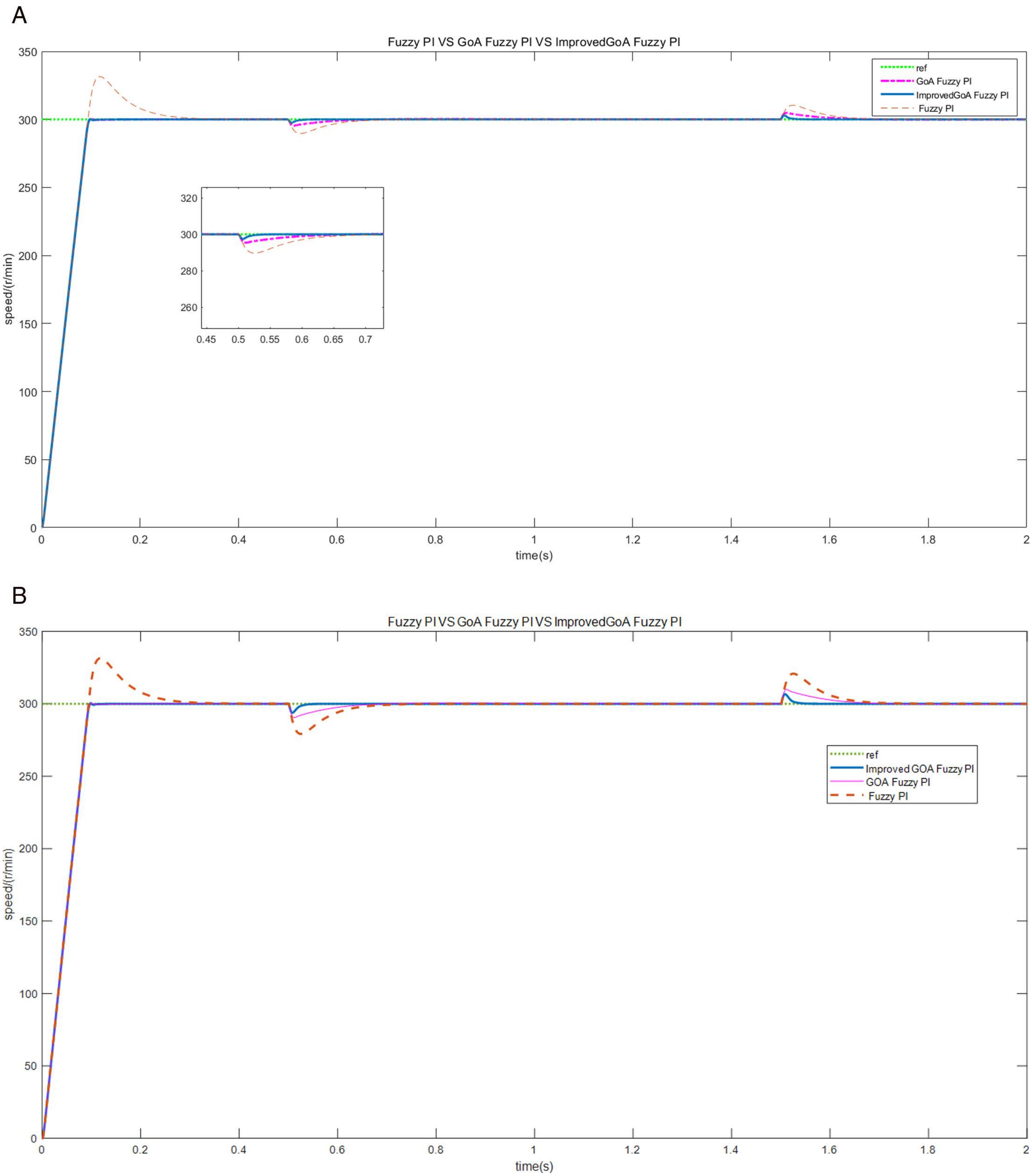

Fig 7. The motor speed curves of load increase/decrease (A) the load of 10 N • m (B) the load of 20 N • m.

https://doi.org/10.1371/journal.pone.0318094.g007

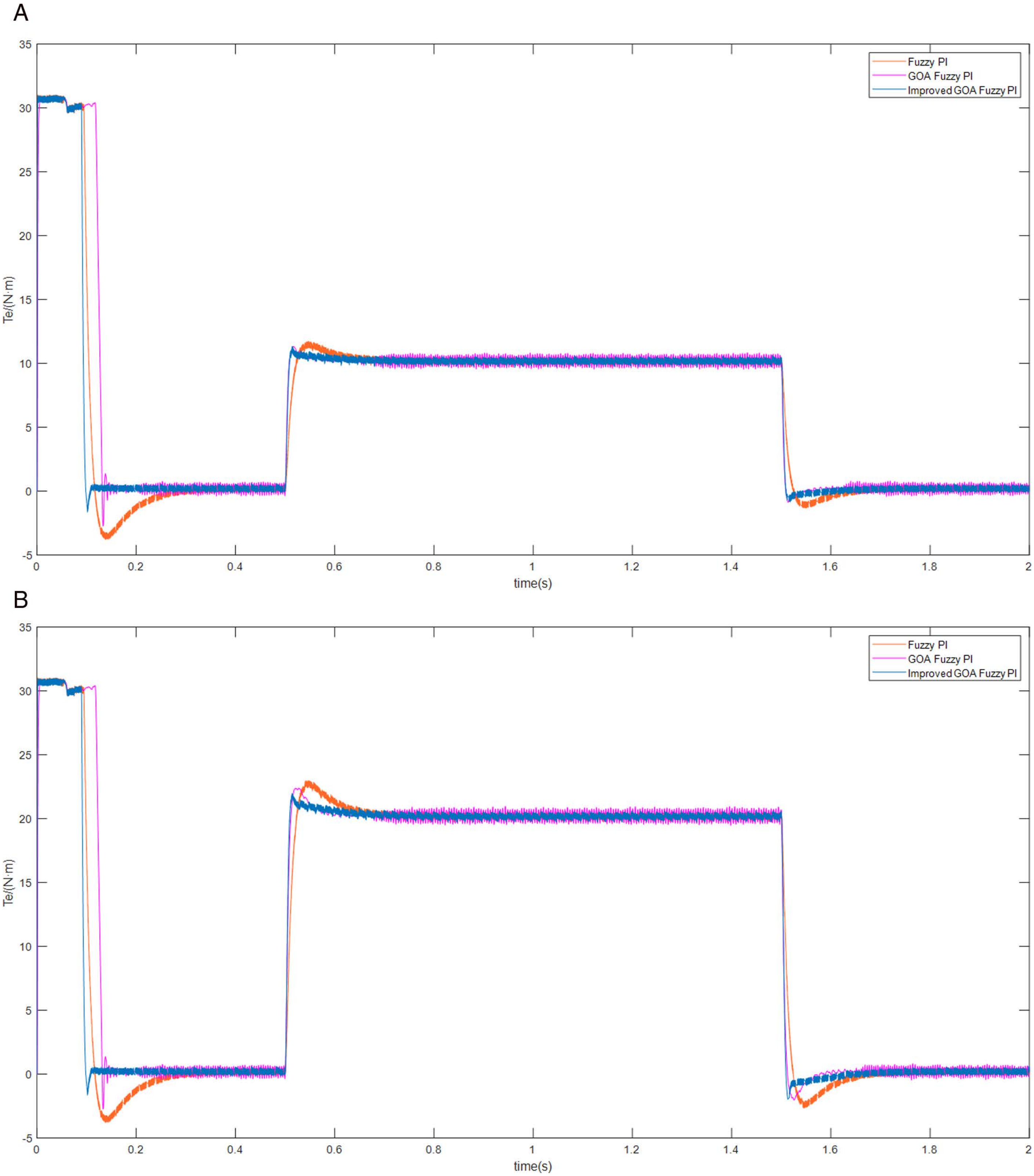

Fig 8. The motor torque curves of load increase/decrease (A) the load of 10 N • m (B) the load of 20 N • m.

https://doi.org/10.1371/journal.pone.0318094.g008

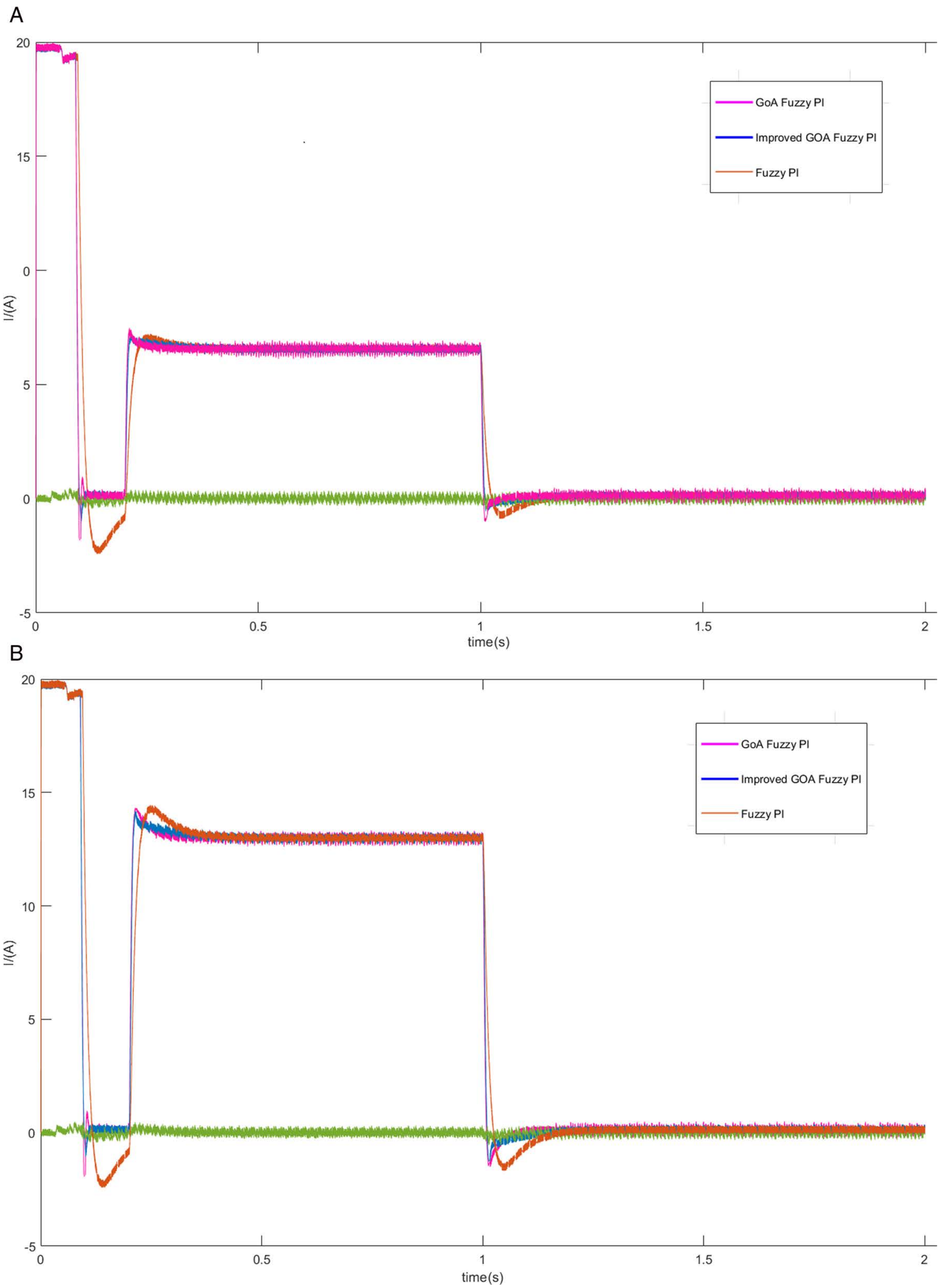

Fig 9. The d-q axis current curves (A) the load of 10 N • m (B) the load of 20 N • m.

https://doi.org/10.1371/journal.pone.0318094.g009

r/min at 2s.The steady-state simulation experiment was carried out under the above conditions. Figs 10 and 11 show the speed and current curves for the three control schemes. Upon comparing the Fig 10, it is evident that scheme 1 displays an overshoot of around 6.8% during the acceleration and deceleration stages, whereas scheme 3 smoothly and rapidly adjusts the speed. From Fig 11, it can be seen the current fluctuation of the q-axis of scheme3 is small, and its operation is stable.

As can be seen from Fig 12, the torque in the scheme3 system becomes 31 N • m and then returns to the load torque value of 10 N • m after about 0.05 s. On the other hand, in the Fuzzy PI system, the torque returns to the load torque value of 10 N • m, the value of the torque increases to 31 N • m and remains there for 0.1 s. In contrast, the current control system predicted by the model has a better immunity to interference.

At the initial stage, the motor operates at a prescribed speed of 300 r/min. However, after 1 second, the motor speed is altered to -300 r/min. The graphical representation in Fig 13 depicts the speed curves for both forward and reverse rotation under the three schemes. The graph clearly illustrates that scheme 3 rapidly and precisely adapts to changes in speed, thus substantiating the effectiveness of the proposed approach.

4.2. Experiment Verification

In order to further verify the effectiveness of the method proposed in this paper, he experiment verification is carried out by using the comprehensive experimental platform of motor speed regulation and loading. The physical representation of the actual platform is shown in Fig 14.

The experimental platform primarily comprises the NI controller, inverter, speed and torque measuring instrument, PMSM, magnetic powder brake, monitoring PC, and essential power distribution lines. The proposed method designed in the Simulink environment is

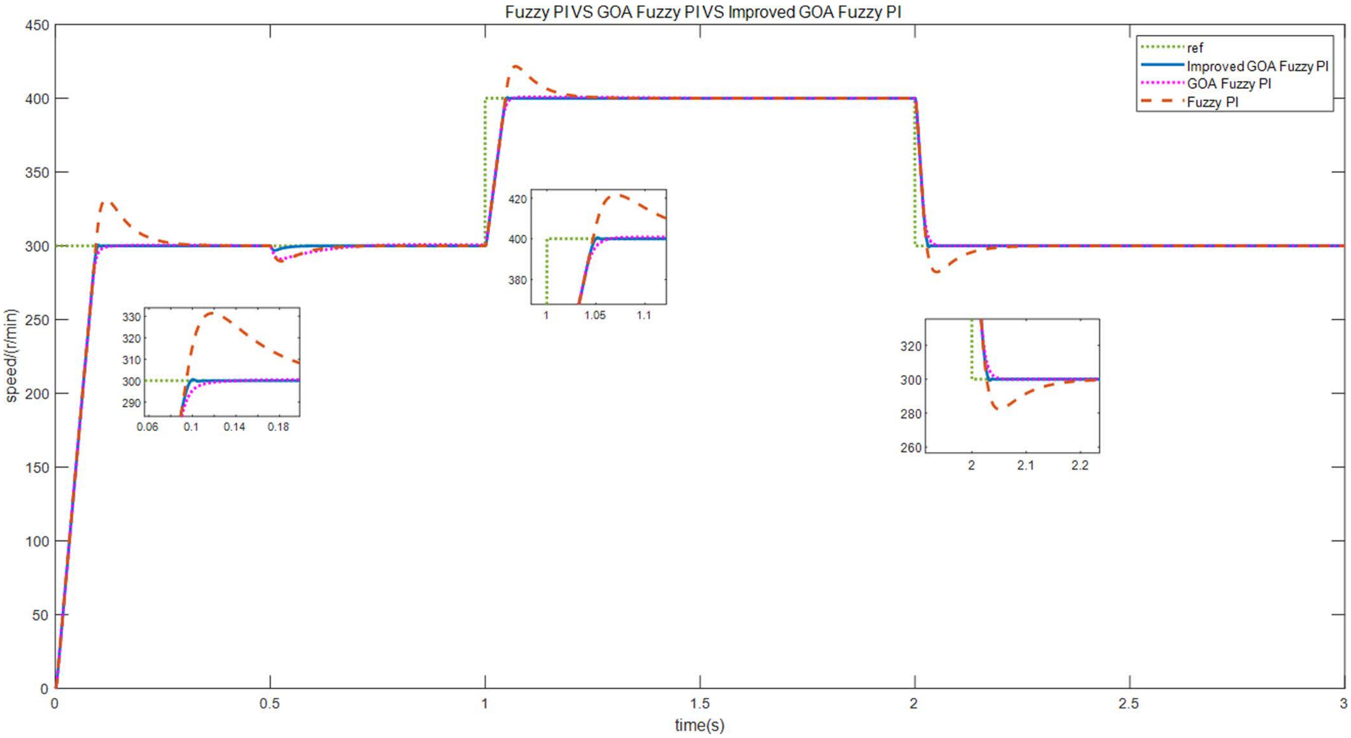

Fig 10. The speed curves of motor speed adjustments.

https://doi.org/10.1371/journal.pone.0318094.g010

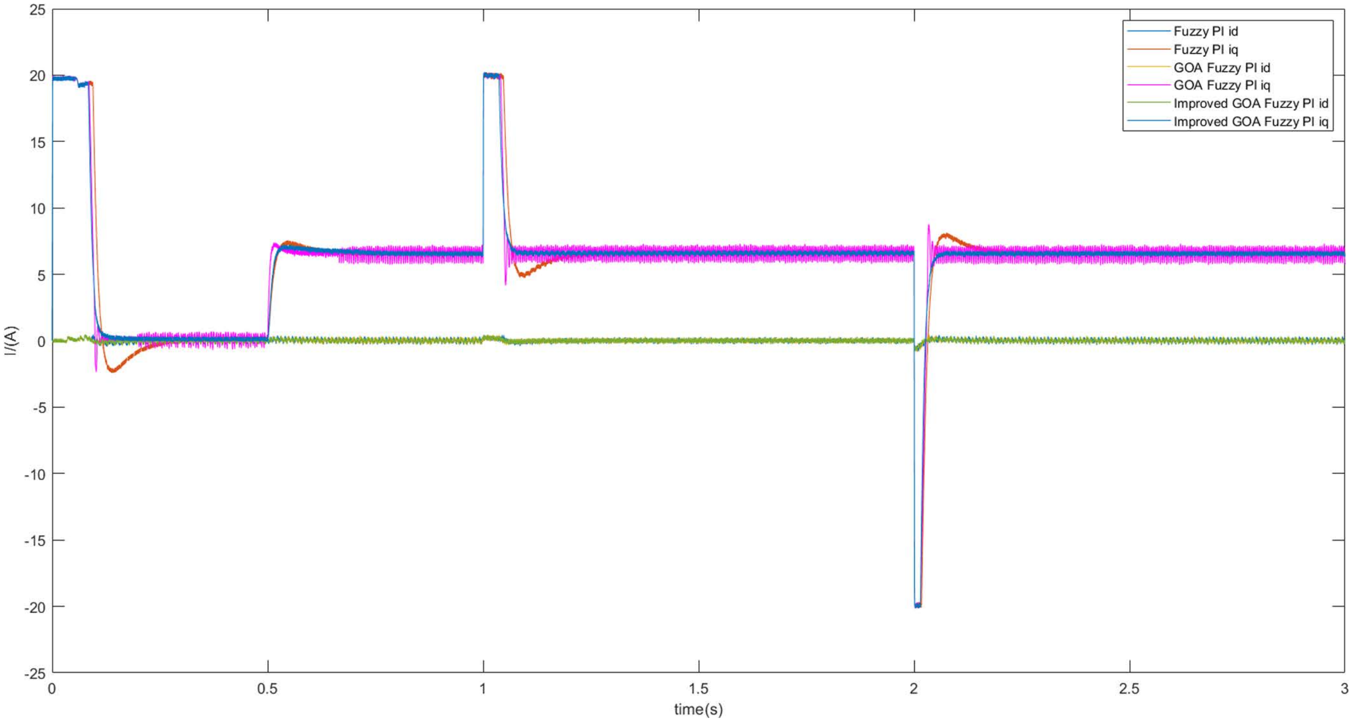

Fig 11. The d-q axis current curves of motor speed adjustments when the motor is loaded with torque.

https://doi.org/10.1371/journal.pone.0318094.g011

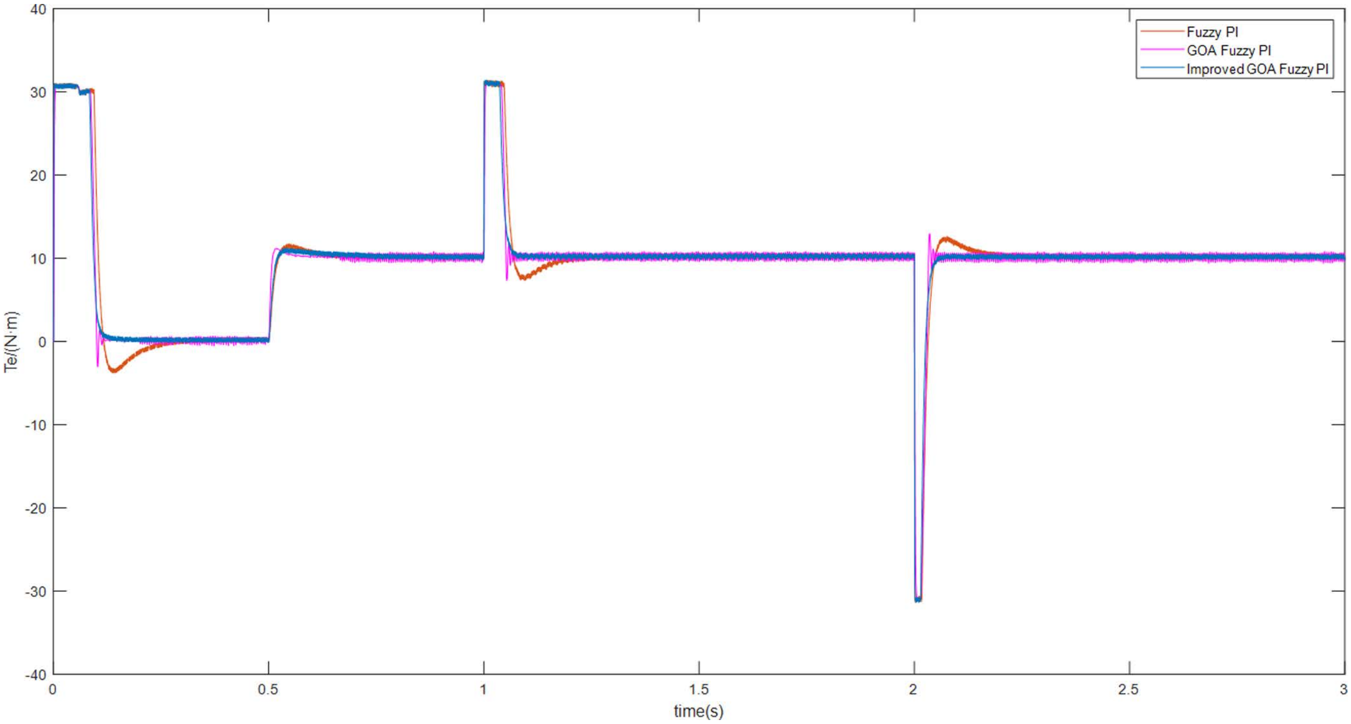

Fig 12. The torque curves of motor speed adjustments.

https://doi.org/10.1371/journal.pone.0318094.g012

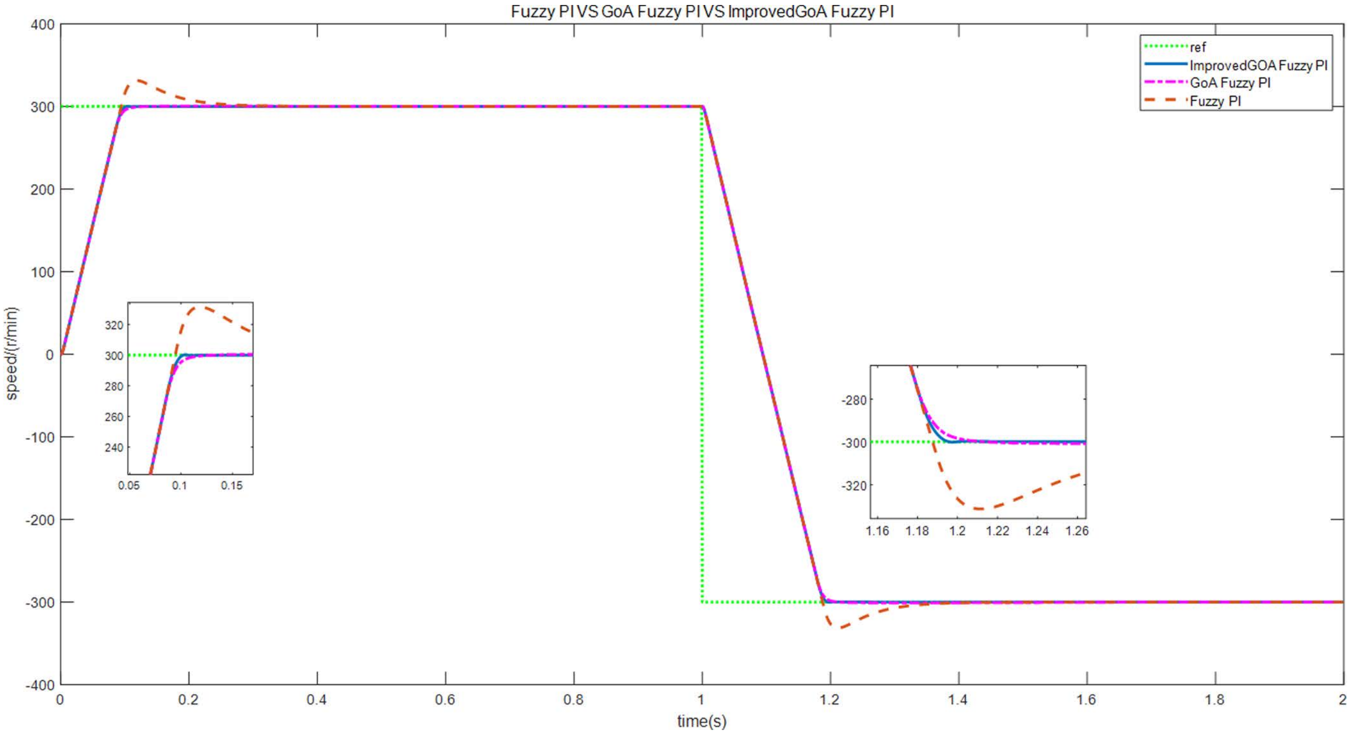

Fig 13. The speed curves of forward and reverse rotation.

https://doi.org/10.1371/journal.pone.0318094.g013

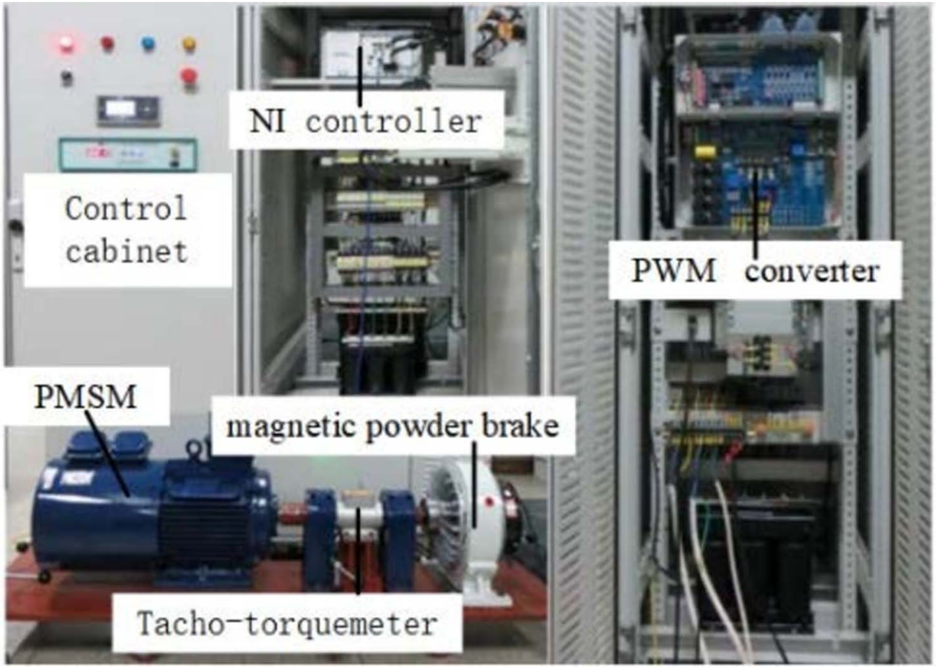

Fig 14. Experiment platform of PMSM control system.

https://doi.org/10.1371/journal.pone.0318094.g014

downloaded to the NI controller. Once the platform parameters are confirmed to be within normal range, the experiment on motor speed control is conducted. The real-time experimental curves, obtained through the monitoring PC, are analyzed to validate the feasibility and effectiveness of the proposed method as presented in this paper.

In the first scenario, the motor speed is adjusted to 300 r/min. Once the motor achieves smooth operation, a torque load of 10 N·m is introduced. Subsequently, the load is subsequently removed after a certain period of running time. Figs 15–20 depict the motor speed curves and current curves for both control schemes. Upon comparing the motor speed curves between the three control schemes, it becomes evident that when the load undergoes changes, the speed fluctuation amplitude of scheme 1 is approximately 15%. Moreover, scheme 1 and scheme 2 exhibit a longer adjustment time in comparison to scheme 3.

Furthermore, upon comparing the present curves in response to variations in motor load, it becomes apparent that scheme 3 displays reduced current fluctuations and faster response throughout the loading and unloading procedures.

In the second scenario, an experiment is conducted to observe the acceleration and deceleration of speed when the motor is loaded with torque. Initially, the motor is set to a speed of 300 r/min. Once the motor is started and running smoothly at this speed, then, the motor is

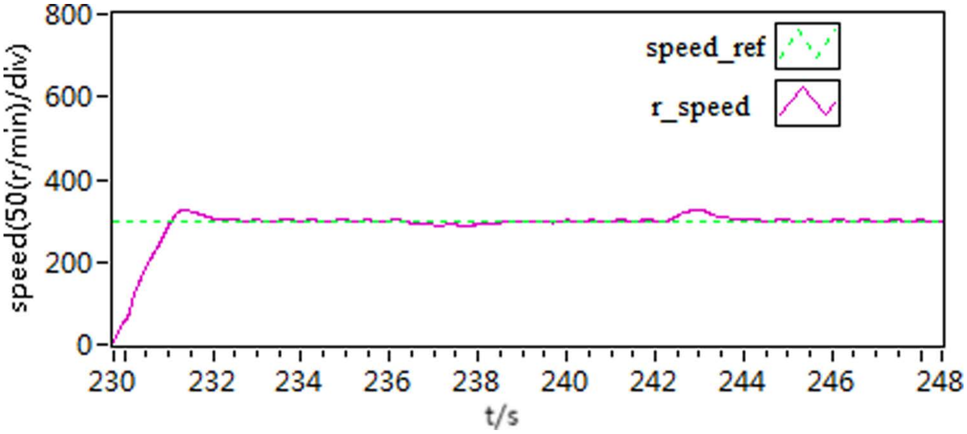

Fig 15. The speed curve of load variations by use of scheme 1.

https://doi.org/10.1371/journal.pone.0318094.g015

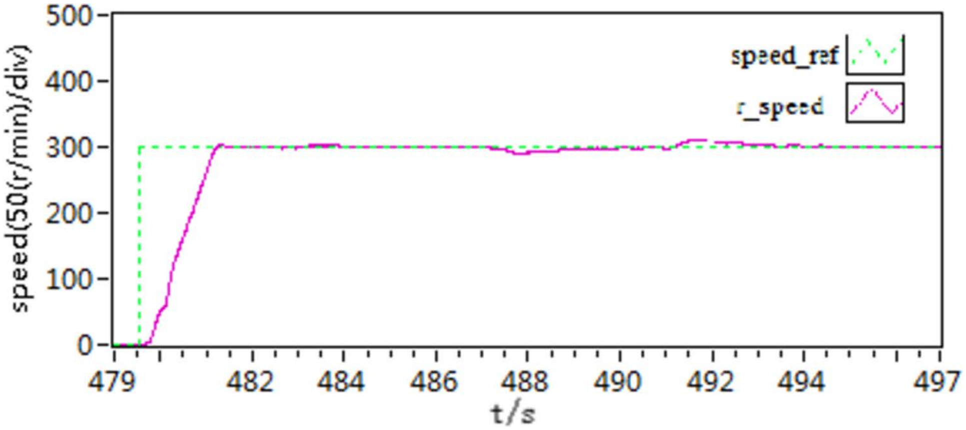

Fig 16. The speed curve of load variations by use of scheme 2.

https://doi.org/10.1371/journal.pone.0318094.g016

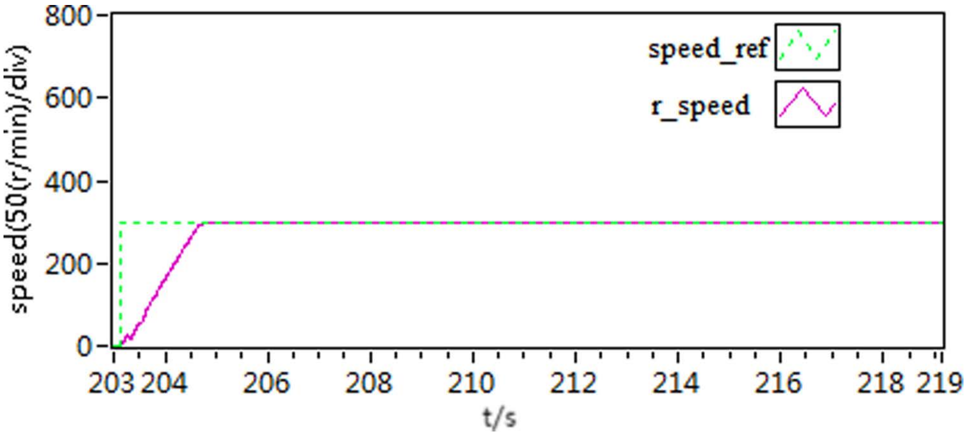

Fig 17. The speed curve of load variations by use of scheme 3.

https://doi.org/10.1371/journal.pone.0318094.g017

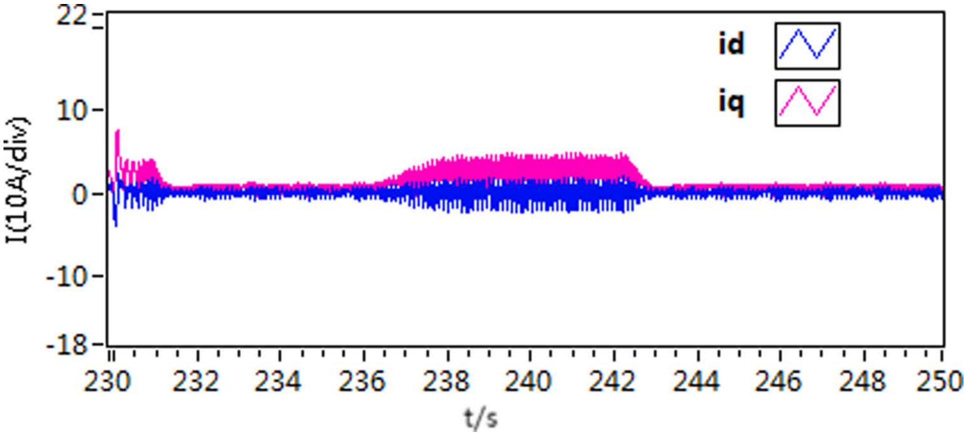

Fig 18. The d-q axis current by use of scheme 1.

https://doi.org/10.1371/journal.pone.0318094.g018

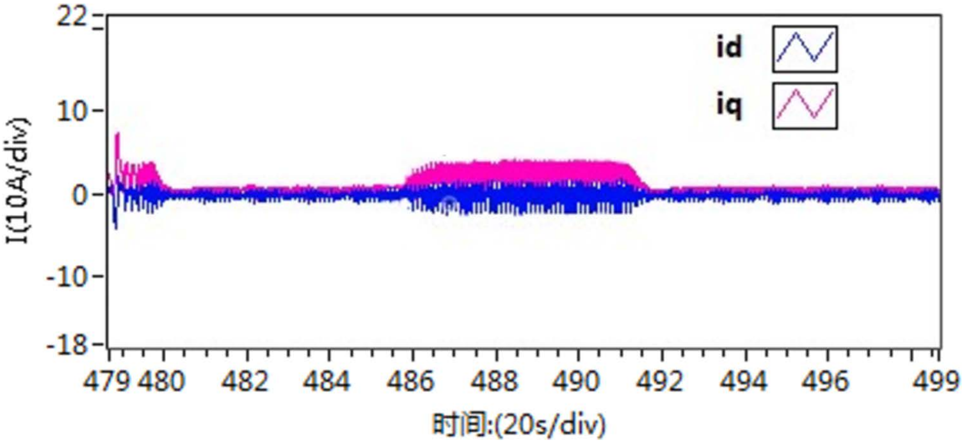

Fig 19. The d-q axis current by use of scheme 2.

https://doi.org/10.1371/journal.pone.0318094.g019

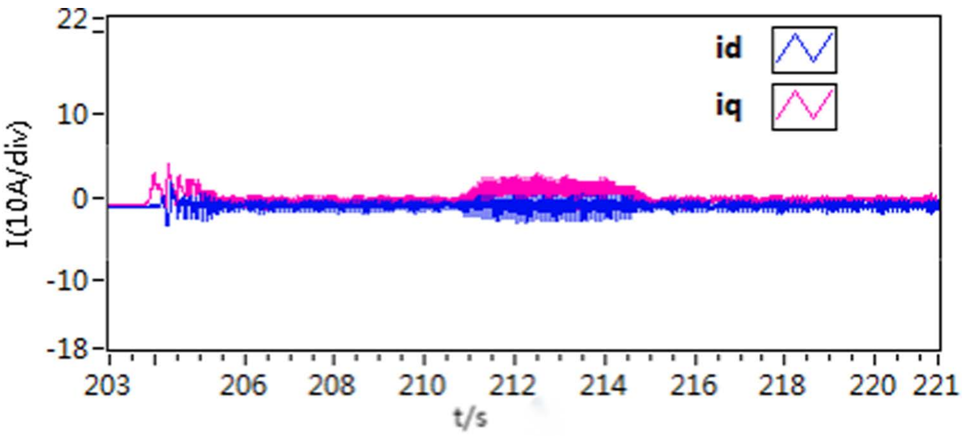

Fig 20. The d-q axis current by use of scheme 3.

https://doi.org/10.1371/journal.pone.0318094.g020

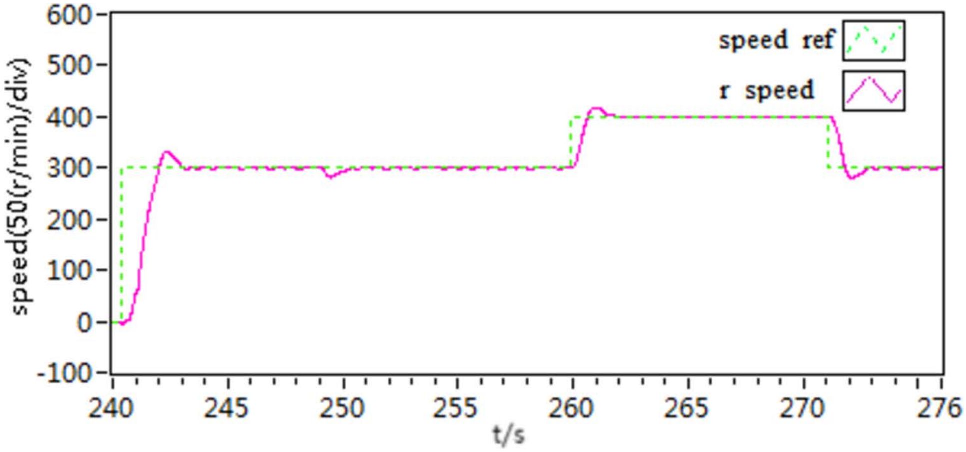

Fig 21. The speed curve of acceleration and deceleration experiment by use of scheme 1.

https://doi.org/10.1371/journal.pone.0318094.g021

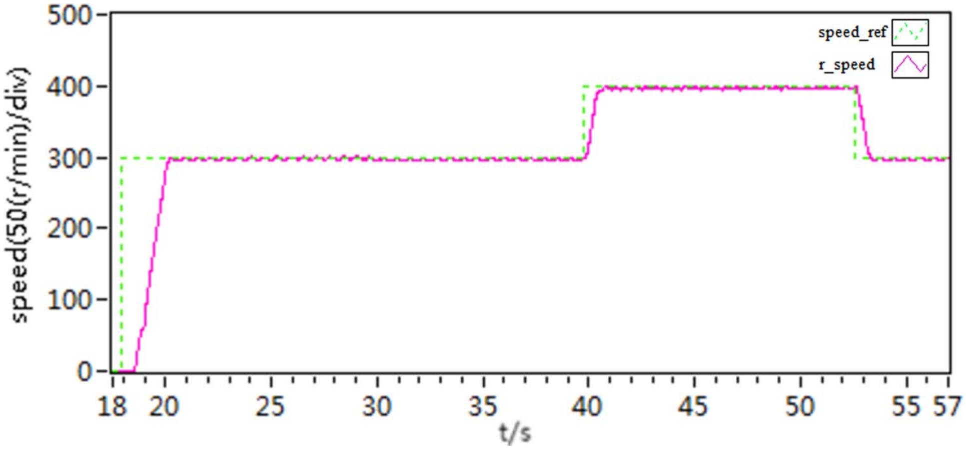

Fig 22. The speed curve of acceleration and deceleration experiment by use of scheme 2.

https://doi.org/10.1371/journal.pone.0318094.g022

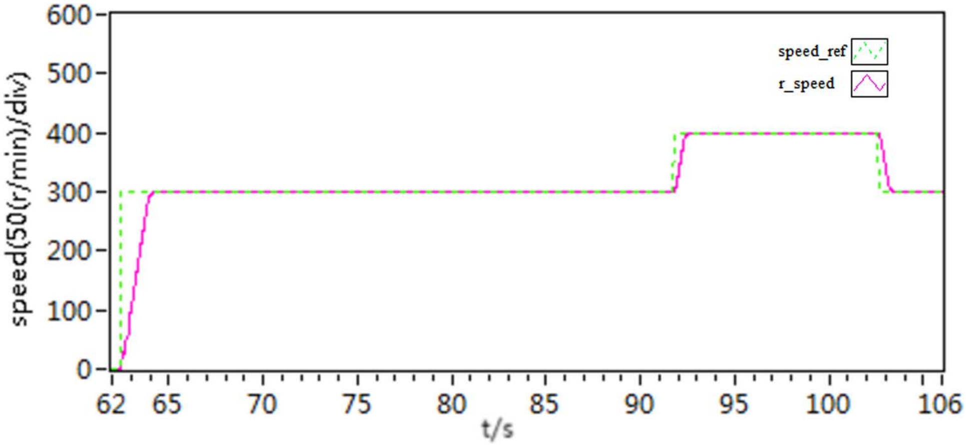

Fig 23. The speed curve of acceleration and deceleration experiment by use of scheme 3.

https://doi.org/10.1371/journal.pone.0318094.g023

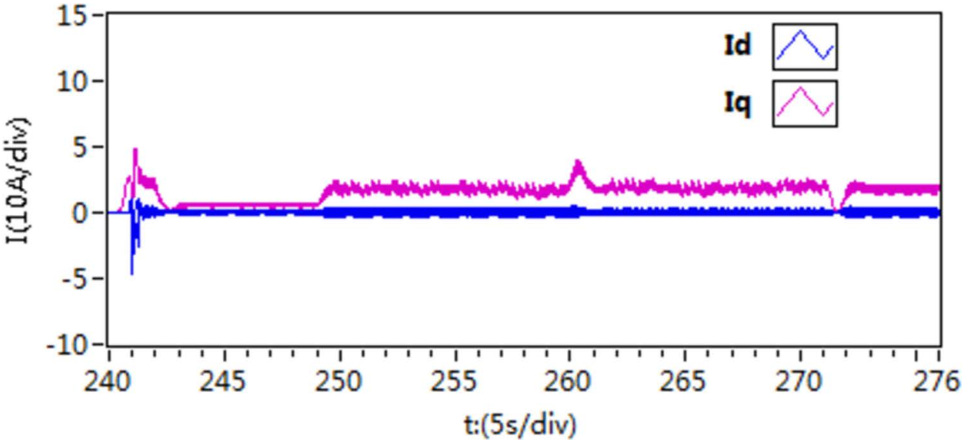

Fig 24. The d-q axis current by use of scheme 1.

https://doi.org/10.1371/journal.pone.0318094.g024

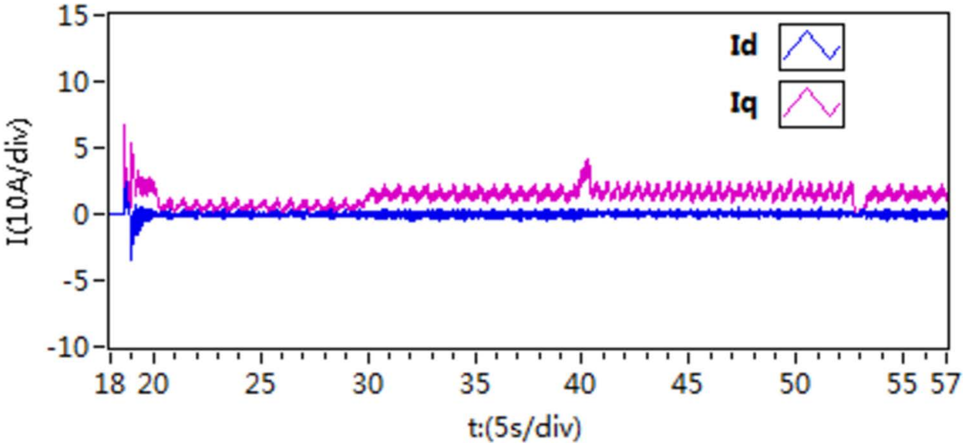

Fig 25. The d-q axis current by use of scheme 2.

https://doi.org/10.1371/journal.pone.0318094.g025

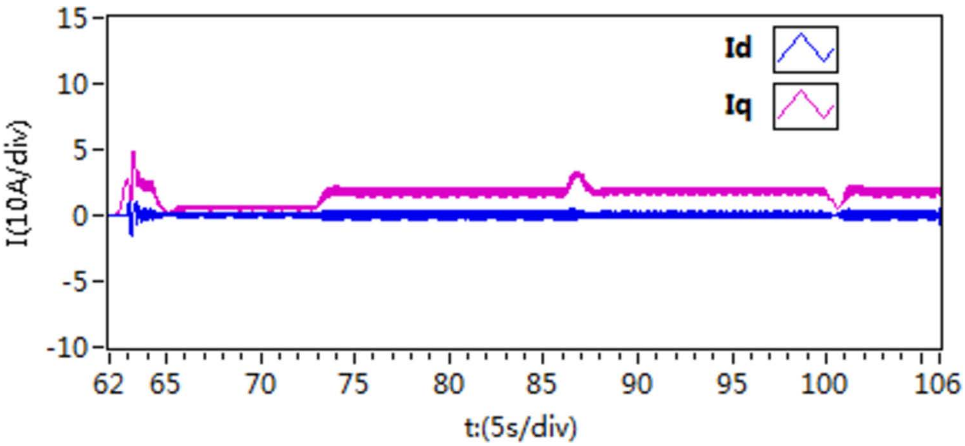

Fig 26. The d-q axis current by use of scheme 3.

https://doi.org/10.1371/journal.pone.0318094.g026

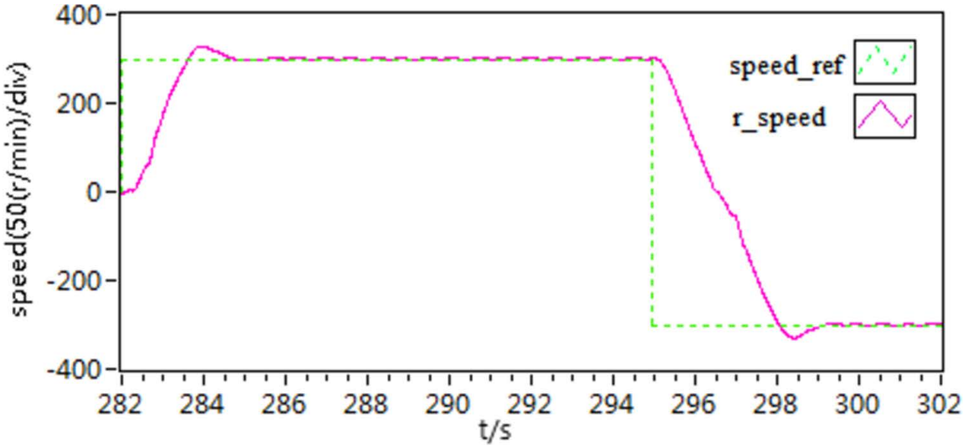

Fig 27. The speed curves of forward and reverse experiment by use of scheme 1.

https://doi.org/10.1371/journal.pone.0318094.g027

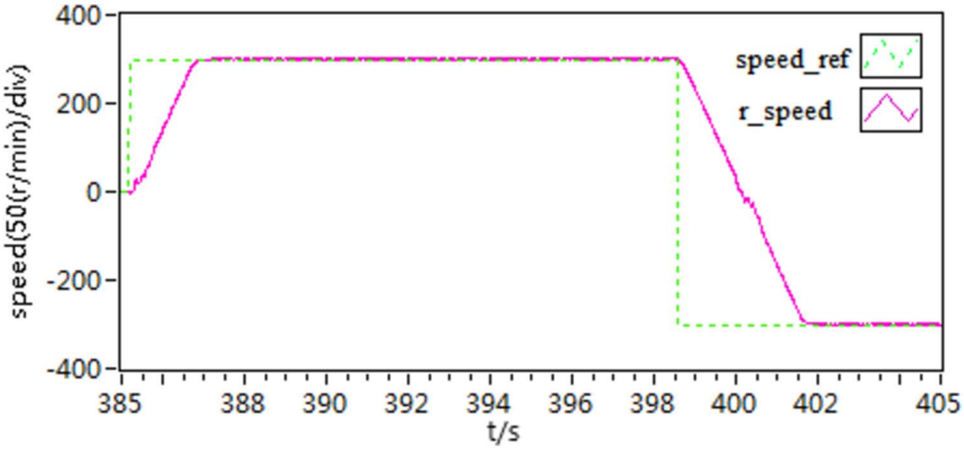

Fig 28. The speed curves of forward and reverse experiment by use of scheme 2.

https://doi.org/10.1371/journal.pone.0318094.g028

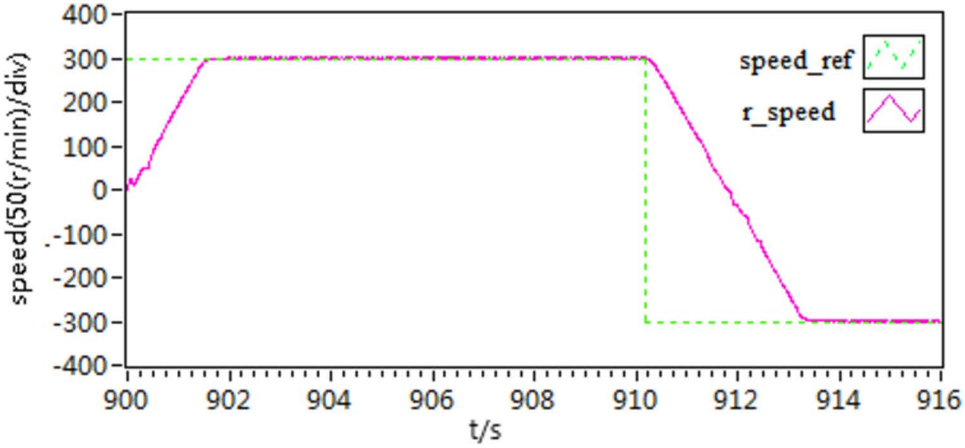

Fig 29. The speed curves of forward and reverse experiment by use of scheme 3.

https://doi.org/10.1371/journal.pone.0318094.g029

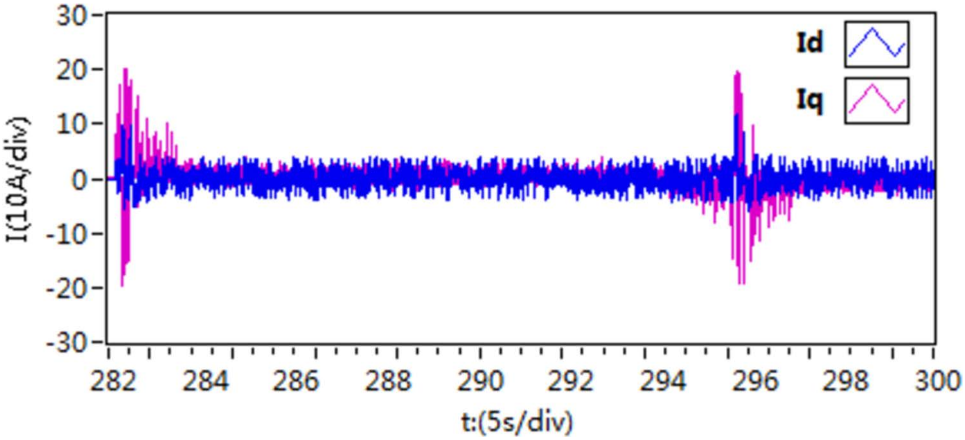

Fig 30. The d-q axis current curves of forward and reverse experiment by use of scheme 1.

https://doi.org/10.1371/journal.pone.0318094.g030

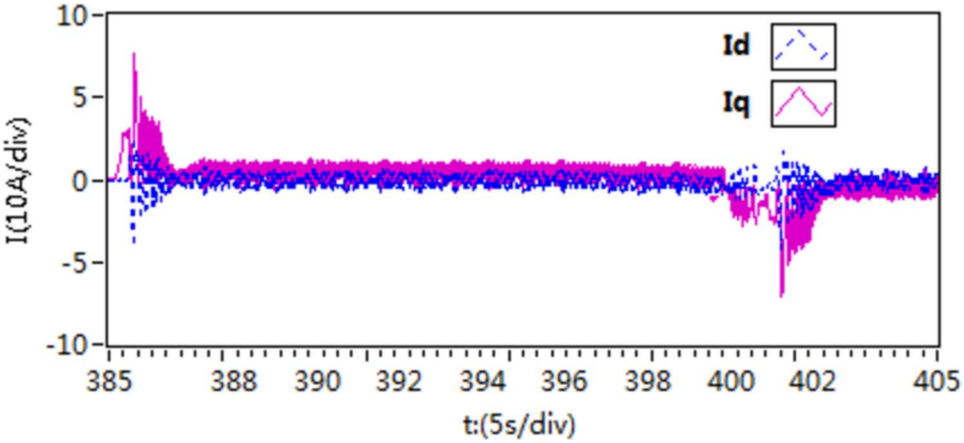

Fig 31. The d-q axis current curves of forward and reverse experiment by use of scheme 2.

https://doi.org/10.1371/journal.pone.0318094.g031

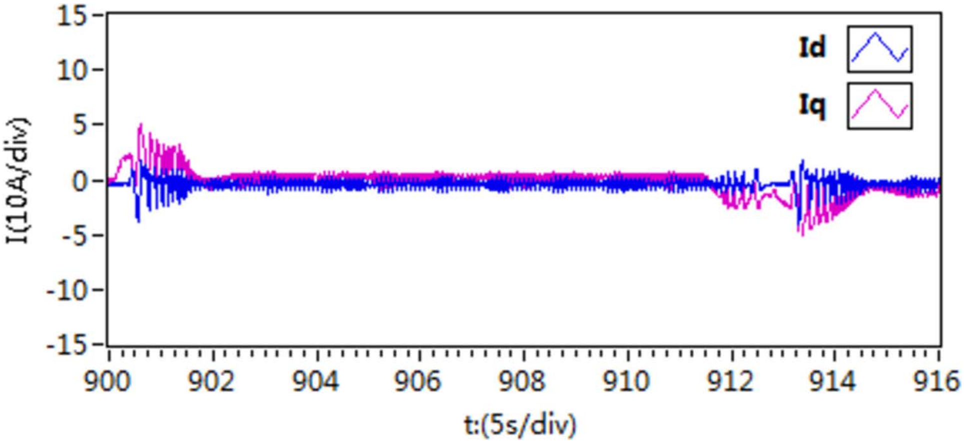

Fig 32. The d-q axis current curves of forward and reverse experiment by use of scheme 3.

https://doi.org/10.1371/journal.pone.0318094.g032

loaded with a constant load torque of 10 N • m, it is then adjusted to 400 r/min. After a certain period, the speed is then readjusted back to 300 r/min. The recorded speed and current curves of this acceleration and deceleration experiment can be seen in Figs 21–26.

According to the Figs 21–26, it is concluded that scheme 3 is capable of achieving precise speed tracking without overshooting, irrespective of whether the motor is running at a higher speed or at a lower speed, and the current fluctuation of file scheme 3 is small, which makes it more resistant to interference.

In the third scenario, an experiment is conducted to test the motor's ability to rotate in both forward and reverse directions. Initially, the motor speed is configured at 300 r/min, and then it is changed to -300 r/min. Figs 27–32 illustrate the results of this experiment. It is evident that when transitioning from forward to reverse rotation, scheme 1 shows a larger overshoot. In contrast, scheme 3 demonstrates the capability to rapidly adjust speed and smoothly transition, thereby demonstrating the robustness of the proposed control scheme.

5. Conclusion

This paper presents a novel methodology for the speed control system of PMSMs by integrating an improved GOA-based fuzzy PI speed controller with a MPCC in a cascaded structure. First, the inherent limitations of traditional GOA, such as susceptibility to local optima and premature convergence, are mitigated by incorporating mixed Cauchy variation and uniform distribution strategies. Subsequently, the enhanced GOA is applied to design a variable-domain fuzzy PI controller for the speed control loop, thereby enhancing the dynamic performance and robustness of the PMSM speed regulation system. Furthermore, to address the limitations of conventional PI control in the motor's current control loop, a one-step-ahead MPCC is developed to minimize tracking errors and improve dynamic performance in the current loop. The proposed control strategy is rigorously validated through both simulations and experimental studies, demonstrating its effectiveness. Future research will focus on integrating disturbance estimation techniques into the proposed approach and addressing the computational burden associated with the control scheme to enhance its practical applicability.

Supporting information

S1 Data. Part of data.
(ZIP)

Author contributions

Conceptualization: Taochang Li, Ang Li.

Data curation: Ang Li.

Investigation: Taochang Li, Ang Li.

Methodology: Taochang Li, Ang Li, Limin Hou.

Project administration: Limin Hou.

Software: Ang Li.

Supervision: Taochang Li, Limin Hou.

Writing – original draft: Ang Li.

Writing – review & editing: Taochang Li, Limin Hou.

References

1. Elmorshedy MF, Xu W, El-Sousy FFM, Islam MdR, Ahmed AA. Recent achievements in model predictive control techniques for industrial motor: a comprehensive state-of-the-Art. IEEE Access. 2021;9:58170–91. https://doi.org/10.1109/access.2021.3073020
2. Li L, Xiao J, Zhao Y, Liu K, Peng X, Luan H, et al. Robust position anti-interference control for PMSM servo system with uncertain disturbance. Trans Electr Mach Syst. 2020;4(2):151–60. https://doi.org/10.30941/cestems.2020.00020
3. Adam AA, Elnady A. Adaptive steering-based HDTC algorithm for PMSM. Asian J Control. 2019;23(1):209–27. https://doi.org/10.1002/asjc.2229
4. Chen Q, Kang S, Zeng L, Xiao Q, Zhou C, Wu M. PMSM control for electric vehicle based on fuzzy PI. IJEHV. 2020;12(1):75. https://doi.org/10.1504/ijehv.2020.104251
5. Sant AV, Rajagopal KR. PM synchronous motor speed control using hybrid fuzzy-PI with novel switching functions. IEEE Trans Magn. 2009;45(10):4672–5. https://doi.org/10.1109/tmag.2009.2022191
6. Jin N, Wang C, Sun D, Li Z, Zhou K. MTPA control of permanent magnet synchronous motor based on dual-vector model predictive control. PLoS One. 2022;17(1):e0262135. https://doi.org/10.1371/journal.pone.0262135 PMID: 35061752
7. Wang F, Zuo K, Tao P, Rodríguez J. High performance model predictive control for PMSM by using stator current mathematical model self-regulation technique. IEEE Trans Power Electron. 2020;35(12):13652–62. https://doi.org/10.1109/tpel.2020.2994948
8. Chebaani M, Mahmoud MM, Tazay AF, Mosaad MI, Nouraldin NA. Extended Kalman Filter design for sensorless sliding mode predictive control of induction motors without weighting factor: an experimental investigation. PLoS One. 2023;18(11):e0293278. https://doi.org/10.1371/journal.pone.0293278 PMID: 38033037
9. Junejo AK, Xu W, Hashmani AA, El-Sousy FFM, Habib HUR, Tang Y, et al. Novel fast terminal reaching law based composite speed control of PMSM drive system. IEEE Access. 2022;1082202–13. https://doi.org/10.1109/access.2022.3196785
10. Yu W, Zhu K, Yu Y. Variable universe fuzzy PID control for active suspension system with combination of chaotic particle swarm optimization and road recognition. IEEE Access. 2024;1229113–25. https://doi.org/10.1109/access.2024.3368762
11. Nicola C-I, Nicola M, Selișteanu D. Sensorless control of PMSM based on backstepping-PSO-type controller and ESO-type observer using real-time hardware. Electronics. 2021;10(17):2080. https://doi.org/10.3390/electronics10172080
12. Wang Q, Yu H, Wang M, Qi X. A novel adaptive neuro-control approach for permanent magnet synchronous motor speed control. Energies. 2018;11(9):2355. https://doi.org/10.3390/en11092355
13. Hoai H-K, Chen S-C, Than H. Realization of the sensorless permanent magnet synchronous motor drive control system with an intelligent controller. Electronics. 2020;9(2):365. https://doi.org/10.3390/electronics9020365
14. Li H, Song B, Chen T, Xie Y, Zhou X. Erratum to "Adaptive fuzzy PI controller for permanent magnet synchronous motor drive based on predictive functional control" [Journal of the Franklin Institute 360.15 (2023) 7333-7364]. J Frank Inst. 2023;360(18):14939. https://doi.org/10.1016/j.jfranklin.2023.10.029

15. Wang L, Tan G, Meng J. Research on model predictive control of IPMSM based on Adaline neural network parameter identification. Energies. 2019;12(24). https://doi.org/10.3390/en12244803

16. Liu Z-H, Wei H-L, Zhong Q-C, Liu K, Xiao X-S, Wu L-H. Parameter estimation for VSI-Fed PMSM based on a dynamic PSO with learning strategies. IEEE Trans Power Electron. 2017;32(4):3154–65. https://doi.org/10.1109/tpel.2016.2572186

17. Wen D, Shi C, Zhang Y, Liao K, Liu J, Luo B, et al. A novel multi-strategy self-optimizing SAPSO algorithm for PMSM parameter identification. IET Power Electronics. 2022;16(2):305–19. https://doi.org/10.1049/pel2.12385

18. Cui J, Li Q, Zhang B, Liu G. Permanent magnet synchronous motor of variable universe adaptive fuzzy PID control. Proceedings of the CSEE. 2013. https://doi.org/10.13334/j.0258-8013.pcsee.2013.s1.030

19. Wang S, Jiang C, Tu Q, Shu H, Zhu C. Particle swarm optimization of fuzzy PI control for PMSMs. J Power Electron. 2023;23(10):1585–93. https://doi.org/10.1007/s43236-023-00643-x

20. Zheng Y, Cao Z, Wang S, Man Z, Chuei R. Extreme learning machine-based field-oriented feedback linearization speed control of permanent magnetic synchronous motors. Neural Comput Appl. 2022;34(7):5267–82. https://doi.org/10.1007/s00521-021-06722-z

21. Vijay Amirtha Raj F, Kamatchi Kannan V. Correction to: Adaptive ELM neural computing framework with fuzzy PI controller for speed regulation in permanent magnet synchronous motors. Soft Comput. 2020;24(14):10981–10981. https://doi.org/10.1007/s00500-020-05071-8

22. Uddin MN, Zou H, Azevedo F. Online loss-minimization-based adaptive flux observer for direct torque and flux control of PMSM drive. IEEE Trans on Ind Applicat. 2016;52(1):425–31. https://doi.org/10.1109/tia.2015.2470642

23. Saremi S, Mirjalili S, Lewis A. Grasshopper optimisation algorithm: theory and application. Adv Eng Softw. 2017;10530–47. https://doi.org/10.1016/j.advengsoft.2017.01.004