

Fractional-Order PID Control of Permanent Magnet Synchronous Motors Based on Based on Improved Squirrel Algorithm

Donglin Tang, Henghui Li

School of mechanical and electrical engineering, Southwest Petroleum University, Chengdu, 610500, China

Abstract: An adaptive differential squirrel algorithm fractional order PID controller is proposed to address the problems of instability and poor accuracy of Permanent Magnet Synchronous Motor PID control systems. In order to improve the global optimisation of the squirrel search algorithm, the crossover operation of the difference algorithm was combined, adaptive predation probability factor, dynamic crossover probability and Gaussian distribution perturbation operation are added to the squirrel search algorithm to improve its exploration and exploitation capability. In the comparative experiment of PMSM speed control, ADESA fractional-order PID was contrasted with conventional PID and other intelligent optimization algorithms. The results indicate that the PMSM control utilizing fractional-order PID through ADESA demonstrated minimal overshoot, the fastest response time, and superior anti-disturbance capabilities compared to PID and other intelligent optimization algorithms. This significantly enhanced the control performance of the permanent magnet synchronous motor.

Keywords: FOPID controller; DE squirrel algorithm; PMSM; Self-adjusting.

1. Introduction

The permanent magnet synchronous motor (PMSM) is widely applied in engineering practices such as machine tools, power systems, and machinery, owing to its advantages of small size, excellent dynamic performance, and high power factor [1-3]. However, due to the nonlinearity and strong coupling characteristics of PMSMs, they exhibit high sensitivity to disturbances and internal parameter variations, making it challenging for traditional PID controllers to achieve the desired performance [4].

Considering the limitations of PID controllers, Professor Podlubny introduced the concept of fractional calculus into PID controllers, creating the Fractional Order Proportional Integration Differentiation (FOPID) controller [5]. Literature [6-8] has employed genetic algorithms for optimizing the parameters of fractional-order PID controllers, demonstrating that compared to integer-order PID controllers, fractional-order PID controllers reduce overshoot, settling time, and steady-state error. In another study [9], the particle swarm optimization algorithm was applied to tune the parameters of fractional-order PID controllers, showcasing superior robustness and control performance compared to integer-order PID controllers. Literature [10] proposed the use of cuckoo search algorithm for designing optimal fractional-order PID controllers, incorporating Levy flights to minimize the risk of converging to local minima. A study by [11] presented a FOPID controller based on the bee swarm algorithm, revealing that the success of PID design heavily relies on the choice of the objective function, and FOPID, with fewer types of objective functions, yields superior responses. Additionally, [12] designed a fractional-order PID controller based on the flower pollination algorithm, demonstrating that, compared to integer-order PID controllers, fractional-order PID controllers can offer faster and smoother velocity responses. The squirrel search algorithm [13], introduced in 2018, is a simple and efficient optimization algorithm with fast convergence and strong search

capabilities, gradually finding applications in various optimization problems, although its global optimization ability remains imperfect. Meanwhile, the crossover mechanism in the differential algorithm [14] introduces a probability of recombining newly generated individuals with individuals from the original population, enhancing population diversity and presenting formidable global search capabilities. Literature [15-16] has demonstrated that incorporating adaptive operations into intelligent algorithms further strengthens their exploratory capabilities.

In order to address challenges such as difficult parameter tuning, unstable response, and poor precision and disturbance rejection in the PID control system of permanent magnet synchronous motors, this paper proposes a novel approach based on research into existing control methods. Specifically, a Fractional Order Proportional Integration Differentiation (FOPID) controller is designed using an adaptive differential squirrel algorithm incorporating a Gaussian distribution disturbance factor for controlling permanent magnet synchronous motors. In the squirrel search algorithm, adaptive predation probability, dynamic crossover probability, and Gaussian distribution disturbance operations are introduced to balance global and local optimization capabilities. The substitution of fractional-order PID for integer-order PID aims to enhance the controller's nonlinear capabilities. Feasibility of the proposed FOPID controller based on the improved squirrel algorithm in the speed control system of permanent magnet synchronous motors is validated through MATLAB simulations.

2. Permanent Magnet Synchronous Motor Model

Assuming the Permanent Magnet Synchronous Motor (PMSM) is an ideal model with a three-phase symmetrical winding and a uniform air gap, the mathematical model of the PMSM under bounded conditions in a coordinate system is as follows:

$$\frac{di_d}{dt} = \frac{1}{L_d}u_d - \frac{R_s}{L_d}i_d + \omega_m i_q \frac{L_q}{L_d} n_p \quad (1)$$

$$\frac{di_q}{dt} = \frac{1}{L_q}u_q - \frac{R_s}{L_q}i_q - \omega_m i_d \frac{L_d}{L_q} n_p - \frac{\varphi_f n_p \omega_m}{L_q} \quad (2)$$

$$\frac{d\omega_m}{dt} = \frac{1}{J}(T_e - T_l - F\omega_m) \quad (3)$$

$$T_e = \frac{3}{2} n_p [\varphi_f i_q + (L_d - L_q) i_d i_q] \quad (4)$$

Here, u_d 、 u_q represent the stator voltage components in the coordinate system; i_d 、 i_q denote the stator current components in the dq coordinate system; L_d 、 L_q represent the inductance components in the dq coordinate system; R_s represents the stator winding resistance; ω_m is the rotor speed of the Permanent Magnet Synchronous Motor (PMSM). T_l 、 n_p 、 F 、 J and φ_f respectively indicate the torque, pole pairs, viscous friction coefficient, rotor inertia, and permanent magnet flux, with T_e representing the electromagnetic torque.

A third-order nonlinear model for the PMSM is established through electrical variables (i_d and i_q currents), mechanical speed (ω_m), and stator voltage (u_d 、 u_q). In the dq coordinate system, the state-space equations for the PMSM are as follows:

$$\dot{x}_1 = \frac{1}{L_d}u_d - \frac{R_s}{L_d}x_1 + x_3 x_2 \frac{L_q}{L_d} n_p \quad (5)$$

$$\dot{x}_2 = \frac{1}{L_q}u_q - \frac{R_s}{L_q}x_2 - x_3 x_1 \frac{L_d}{L_q} n_p - \frac{\varphi_f n_p x_3}{L_q} \quad (6)$$

$$\dot{x}_3 = \frac{1}{J} \left(\frac{3}{2} n_p [\varphi_f x_2 + (L_d - L_q) x_1 x_2] - T_l - F\omega_m \right) \quad (7)$$

$$y_1 = x_3, \quad y_2 = x_1 \quad (8)$$

Here, the stator voltage and state variables are denoted by u_d 、 u_q 、 $x_1 = i_d$ 、 $x_2 = i_q$ and $x_3 = \omega_m$, respectively. The mathematical model and state equations of the aforementioned PMSM provide a theoretical foundation for constructing the experimental framework model in subsequent experiments.

3. Adaptive Differential Squirrel Algorithm with Disturbance Factor

3.1. Squirrel Search Algorithm

The Squirrel Search Algorithm (SSA) is a recently proposed swarm intelligence-based algorithm that simulates the foraging behavior of squirrels in a forest.

In the Squirrel Search Algorithm, the initial positions of squirrels are randomly arranged in the search space. They are

then sorted in ascending order based on the fitness values calculated by the objective function. The fitness of each squirrel's position describes the rank of food sources, including the optimal food source (walnut tree), normal food source (oak tree), and no food source (ordinary tree). During the dynamic foraging process, squirrels may encounter three situations, and their foraging behavior is influenced by predators. The specific movement strategy adopted by a squirrel depends on the probability of predator appearance.

In the first scenario, squirrels on oak trees will move to new positions based on the locations of squirrels on walnut trees. The position update formula (Equation 9) for squirrels is given by:

$$PS_{at}^{new} = \begin{cases} PS_{at}^t + d_g \times G_c \times (PS_{nt}^t - PS_{at}^t), & R_1 \geq P_p \\ \text{Random location}, & \text{otherwise} \end{cases} \quad (9)$$

Here, PS_{at}^t represents the position of a squirrel on an oak tree, PS_{nt}^t is the position of a squirrel on a walnut tree, PS_{at}^{new} represents the updated position of the squirrel on an oak tree, d_g is the randomly generated gliding distance, R_1 is a random number within the range [0,1], t denotes the current iteration, $G_c = 1.9$ is the sliding coefficient that balances between global and local searches, and P_p is the probability of the appearance of predators.

In the second scenario, squirrels on ordinary trees will move to new positions based on the locations of squirrels on oak trees. The position update formula (Equation 10) for squirrels is given by:

$$PS_{nt}^{new} = \begin{cases} PS_{nt}^t + d_g \times G_c \times (PS_{at}^t - PS_{nt}^t), & r_2 \geq P_p \\ \text{random position}, & \text{otherwise} \end{cases} \quad (10)$$

Here, PS_{nt}^t represents the position of a squirrel on an ordinary tree, PS_{nt}^{new} represents the updated position of the squirrel on an ordinary tree, and R_2 is a random number within the range [0,1].

In the third scenario, squirrels on ordinary trees will move to new positions based on the locations of squirrels on walnut trees. The position update formula (Equation 11) for squirrels is given by:

$$PS_{nt}^{new} = \begin{cases} PS_{nt}^t + d_g \times G_c \times (PS_{nt}^t - PS_{nt}^t), & r_3 \geq P_p \\ \text{random position}, & \text{otherwise} \end{cases} \quad (11)$$

According to the position update formulas (Equations 9 to 11), the fitness function values are calculated, and the positions of squirrels are re-sorted. This process is repeated until the iteration is completed.

3.2. Adaptive Differential Squirrel Algorithm

The Squirrel Search Algorithm sometimes faces challenges of getting trapped in local minima when dealing with complex optimization problems. This paper proposes an Adaptive Differential Squirrel Algorithm with a Gaussian distribution disturbance factor. In this approach, the cross-operation of the differential algorithm is integrated into the Squirrel Search Algorithm. This allows newly generated individuals to recombine with individuals from the original population, thereby enhancing population diversity. The algorithm

introduces an adaptive predation probability factor and dynamic crossover probability, improving both the global search capability in the early stages and the local exploitation capability in the later stages. Additionally, Gaussian distribution disturbance operations are incorporated to reduce the likelihood of falling into local minima.

3.2.1. Adaptive Predation Probability Factor

In the Squirrel Search Algorithm, the foraging behavior of each squirrel is influenced by the predation probability. If the probability is too small, the algorithm may face premature convergence issues, while if it is too large, it can impact the global search capability. Therefore, this paper introduces an adaptive predation probability factor, denoted as P_{dp} , with the following formula:

$$P_{dp} = P_d^S + \frac{F_d^M}{1 + \exp((g - \omega_1 G)(\omega_2 / G))} \quad (12)$$

Here, P_d^S is the minimum predation probability, F_d^M is the maximum predation probability, g represents the current iteration count, G is the maximum iteration count, ω_1 , ω_2 are adjustable factors. The adaptive predation probability factor P_{dp} is used to replace P_p in the squirrel position update formula, reducing the likelihood of the algorithm falling into local optima in the early stages.

3.2.2. Dynamic Crossover Probability

The crossover operation in the differential algorithm enhances the algorithm's search capability. However, the crossover probability affects the global optimization ability. In the early stages of exploration, individuals with larger fitness values are farther from the target region. When the crossover probability is larger, the population diversity is richer, leading to an improvement in global search capability. In the later stages, as fitness values decrease, it is crucial to retain the current excellent individuals. Hence, the crossover probability needs to decrease to enhance the algorithm's local exploitation capability. This paper introduces a dynamic crossover probability Cr_i based on the individual fitness function values, with the formula (13) as follows:

$$Cr_i = \frac{f(X_i) - f_{\min}(X)}{f_{\max}(X) - f_{\min}(X)} \quad (13)$$

Here, $f(X_i)$ is the fitness value of the i th squirrel, $f_{\max}(X)$ and $f_{\min}(X)$ are the current population's maximum and minimum fitness values, respectively.

The crossover operation utilizes the current position of the squirrel and its new position obtained from Formula (10) to determine the squirrel's next iteration position on the oak tree. This is expressed in Formula (14):

$$PS_{at}^{cro} = \begin{cases} PS_{at,i,j}^{new}, & \text{if } (\text{rand}_j \leq Cr_i) \text{ or } j = j_{rand} \\ PS_{at,i,j}^{cro}, & \text{if } (\text{rand}_j > Cr_i) \text{ or } j \neq j_{rand} \end{cases}, j = 1, 2, 3, \dots, D \quad (14)$$

Here, i ranges from 1 to NP, where NP is the size of the squirrel population. PS_{at}^{cro} represents the position of the squirrel on the oak tree after the crossover operation. $j = 1, 2, 3, \dots, D$ is the dimensionality of the problem, $\text{rand}_j \in (0, 1)$ is a random number, and $j_{rand} \in (1, D)$ is a

random index.

3.2.3. Gaussian Distribution Disturbance Factor

To explore potential excellent individuals, a disturbance factor based on the Gaussian distribution is proposed to perform domain search operations, enhancing local search capabilities and obtaining the final position of the squirrel. The expression (15) is given as follows:

$$PS_{at}^{t+1} = PS_{at}^{cro} + C \times N(0, 1) \quad (15)$$

Here, PS_{at}^{t+1} represents the final position of the squirrel for the current iteration, C is a preset constant used to control the range of random numbers, and $N(0, 1)$ denotes a random number based on the standard Gaussian distribution.

3.3. Algorithm Effectiveness Experiment

To validate the effectiveness of the Adaptive Differential Squirrel Algorithm with disturbance factor, iterative experiments were conducted using six typical benchmark functions, as shown in Table 1. These benchmark functions are characterized by multiple local minima, large search spaces, and deceptive features. For comparison, Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Differential Evolution (DE), and Squirrel Search Algorithm (SSA) were individually tested. Each algorithm performed 30 optimization runs for each benchmark function.

Table 1. Benchmark function names and expressions

Benchmark Function	Translate expressions
Griewank	$\min f_1(\mathbf{x}) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
Ackley	$\min f_2(\mathbf{x}) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$
Rastrigin	$\min f_3(\mathbf{x}) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$
Salomon	$\min f_4(\mathbf{x}) = -\cos\left(2\pi \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) + 0.1 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} + 1$
Schaffer	$\min f_5(\mathbf{x}) = 0.5 + \frac{(\sin \sqrt{x_1^2 + x_2^2})^2 - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$
Sphere	$\min f_6(\mathbf{x}) = \sum_{i=1}^D x_i^2$

For comparison, Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Differential Evolution (DE), and Squirrel Search Algorithm (SSA) were individually tested. Each algorithm performed optimization runs for each benchmark function 30 times. The comparison results of the benchmark functions are shown in Table 2, where $f(x)Value_{av}$ represents the optimal value found by each algorithm. From the comparison results, it can be observed that, in all benchmark function optimizations, ADESA's $f(x)Value_{av}$ is significantly smaller than the other algorithms. For benchmark functions f1, f2, f5, and f6, ADESA's iteration count is much lower than the other algorithms. Compared to the Squirrel Search Algorithm, ADESA has the highest iteration count difference of 244

iterations. In the optimization of benchmark functions f3 and f4, ADESA's precision has been improved by 0.706 and

0.1702, respectively.

Table 2. Optimisation results of benchmark function under different intelligent algorithms

Functions	GA	PSO	DE	SSA	ADESA
$f_1(x)Value_m$	8.372e-1	6.735e-1	1.76e-3	0	0
Iteration	1000	1000	1000	952	629
$f_2(x)Value_m$	2.7391	5.408e-1	1.25e-2	0	0
Iteration	1000	1000	1000	743	517
$f_3(x)Value_m$	67.534	45.214	140.6	7.381e-1	0.319e-1
Iteration	1000	1000	1000	1000	1000
$f_4(x)Value_m$	2.3741	9.462e-1	6.486e-1	2.462e-1	0.758e-1
Iteration	1000	1000	1000	1000	1000
$f_5(x)Value_m$	7.937e-5	0	0	0	0
Iteration	1000	725	529	418	174
$f_6(x)Value_m$	1.384e-3	2.443e-3	5.934e-5	0	0
Iteration	1000	1000	1000	593	482

The convergence trend curves for the corresponding benchmark functions are shown in Figure 1. From these trend curves, it can be observed that the Adaptive Differential Squirrel Algorithm with disturbance factor exhibits superior search performance for complex optimization problems. In this algorithm, the range of squirrel search is expanded under the influence of adaptive predation and dynamic crossover operations, balancing both global and local optimization

capabilities. The local search capability of the algorithm is enhanced under the influence of the Gaussian disturbance factor. Compared to other algorithms, ADESSA achieves better fitness values in the early iterations, showing significantly faster convergence speed. The spatial search capability and convergence speed of the algorithm have been markedly improved.

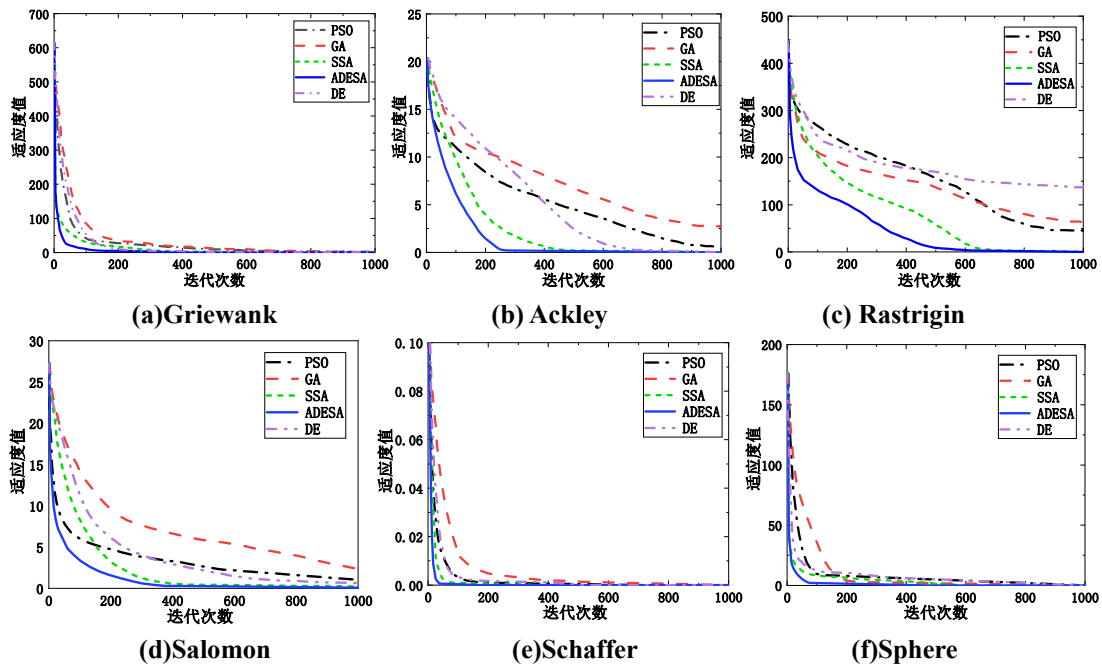


Figure 1. Comparison curve of iterative convergence of benchmark functions

4. Fractional Order PID Control System Design for PMSM

4.1. Fractional Order PID Controller

Fractional calculus is an extension of the basic operators ${}_0 D_t^\alpha$ integration and differentiation to non-integer orders, where t_0 and t are limit factors, and α represents the order of the operation. The definition for fractional order operator ${}_0 D_t^\alpha$ integration is the Grünwald-Letnikov definition, given by Equation (16):

$${}_0 D_t^\alpha f(t) = \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{j=0}^{[(t-t_0)/h]} (-1)^j \binom{\alpha}{j} f(t-jh) \quad (16)$$

Here, h represents the step size factor, $[\cdot]$ denotes the nearest integer function, and $(-1)^j \binom{\alpha}{j} = \frac{(-1)^j \Gamma(\alpha+1)}{\Gamma(j+1)\Gamma(\alpha-j+1)}$ is the binomial coefficient of function $(1-z)^\alpha$.

The fractional-order PID controller is an extension of the traditional integer-order PID. The time-domain differential equation for the fractional-order PID controller is given by Equation (17):

$$u(t) = K_p e(t) + K_i D_t^{-\lambda} e(t) + K_d D_t^\mu e(t) \quad (17)$$

Here, $u(t)$ is the control signal, $e(t)$ is the error signal, and the parameters λ and μ typically range from 0 to 2. K_p is the proportional coefficient, K_i is the differential coefficient, and K_d is the integral coefficient.

Through Laplace transformation, the generalized transfer function of the fractional-order PID is represented by Equation (18):

$$G_c(s) = K_p + \frac{K_i}{s^\lambda} + K_d s^\mu \quad (18)$$

4.2. Fractional-Order PID Controller for PMSM based on ADESA

The fractional-order PID control for Permanent Magnet Synchronous Motor (PMSM) based on Adaptive Differential Squirrel Algorithm (ADESA) is illustrated in Figure 2. The fractional-order PID controller provides closed-loop control for the PMSM, and ADESA is employed to tune the five parameters (K_p , K_i , K_d , λ , and μ) of the fractional-order PID controller. The Iterative Time Absolute Error (ITAE) is chosen as the fitness function due to its advantages such as fast response and small overshoot. Through continuous iterations, the lowest fitness function value is obtained, and the corresponding values for the five parameters are considered as the final parameter values.

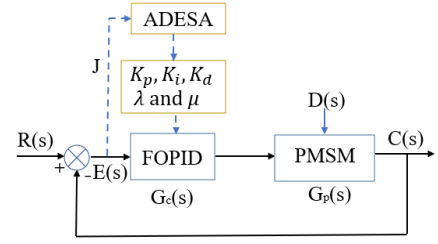


Figure 2. Fractional order PID control design diagram

Here, $R(s)$ represents the desired input, $C(s)$ is the actual output, $E(s)$ is the error, $G_c(s)$ is the controller transfer function, $G_p(s)$ is the controlled object, $D(s)$ is the disturbance input, and J is the performance index (fitness function).

4.3. Simulink Simulation Model Construction

To validate the control effectiveness of the fractional-order PID Permanent Magnet Synchronous Motor (PMSM) based on the ADESA algorithm, this paper employs MATLAB software to construct a simulation system for PMSM speed control. The PMSM speed is set to 700 rad/s, and the Fractional Order PID (FOPID) is implemented using MATLAB and the FOMCON toolbox. The fractional-order simulation is approximated using the Oustaloup differential filtering algorithm. The control block diagram for the PMSM is illustrated in Figure 3.

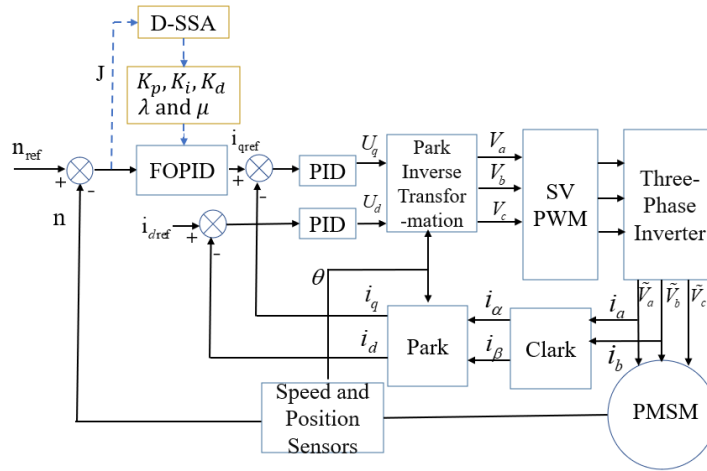


Figure 3. Block diagram of PMSM control system

The Permanent Magnet Synchronous Motor parameters are shown in Table 3.

Table 3. PMSM parameter settings

parameters	values
Speed $n(\text{rad} / \text{s})$	700
Stator Resistance $R(\Omega)$	1.86
Stator Inductance $L(\text{mH})$	2.8
Magnetic Flux $\phi_f(\text{Wb})$	0.109
Rotational Inertia $J(\text{kg} \cdot \text{m}^2)$	1.39E-4
Number of Poles n_p	4

When applying the Adaptive Differential Squirrel Algorithm (ADSA) and other intelligent algorithms to PMSM, to ensure stability and prevent excessive response,

overshooting, oscillation, or distortion, it is necessary to restrict the search range of fractional-order PID parameters, specifically: $K_p, K_i \in (0, 100)$, $K_d, \lambda, \mu \in (0, 1)$. The algorithm is set to iterate 1000 times with an initial population size of 50.

5. Simulation Results and Analysis

5.1. Comparison Experiment of FOPID PMSM Control

The Adaptive Differential Squirrel Algorithm (ADSA) is used to optimize the parameters of both integer-order PID and fractional-order PID controllers. After tuning the controller parameters with the ADSA algorithm, the PMSM speed response comparison curve is shown in Figure 3. Compared to the integer-order PID, the PMSM under fractional-order

PID control exhibits a reduced overshoot from 22.8 rad/s to 7.1 rad/s, peak time from 0.156 s to 0.09 s, and settling time from 0.584 s to 0.323 s, providing a smoother and faster speed response for the PMSM.

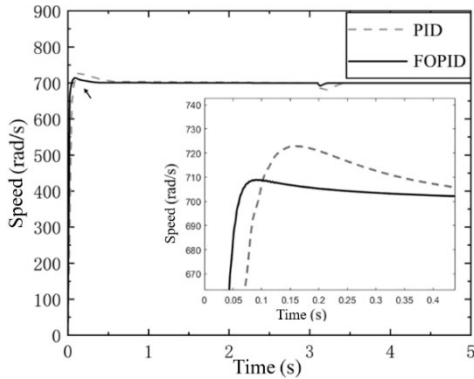


Figure 4. Comparison of PMSM speed response for PID and FOPID

At 3 seconds, a negative load of 5 N·m is applied to the PMSM, and its speed fluctuation response is shown in Figure 5. The PID control generates excessive overshoot and slow response time. Under fractional-order PID control, the speed fluctuation of the PMSM is only 7.6 rad/s, stabilizing after 0.13 seconds. The PMSM exhibits strong anti-disturbance capabilities.

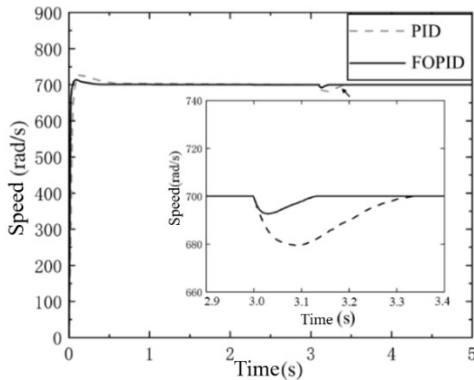


Figure 5. PMSM velocity response under sudden load increase

5.2. ADESA PMSM Control Comparative Experiment

To validate the feasibility of the proposed algorithm, Particle Swarm Optimization (PSO), Firefly Algorithm (FA), Flower Pollination Algorithm (FPA), and the non-improved Squirrel Search Algorithm (SSA) are compared with the Adaptive Differential Squirrel Algorithm (ADESA) in tuning the parameters of the fractional-order PID controller.

Under the control of different intelligent algorithms, the PMSM speed response curves are shown in Figure 6. When calculating the adjustment time with an error of 2%, ADESA achieves control effectiveness at a rise time of 0.048s, while SSA is at 0.052s, and FA and PSO are at 0.129s and 1.3s, respectively. FA and PSO have no overshoot but longer response times than ADESA. ADESA's overshoot of 7.1 rad/s is smaller than SSA's 12.4 rad/s and FPA's 22.3 rad/s. The experiments demonstrate that the improved

ADESA performs better than the original SSA in control effectiveness. The adaptive predation probability factor and dynamic crossover operation enhance the global search and

local exploitation capabilities of SSA, accelerating the convergence of PMSM speed and ensuring smoother speed control.

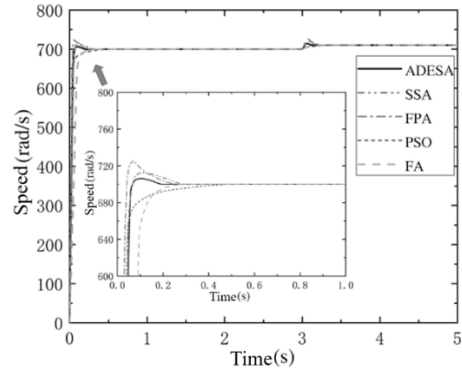


Figure 6. Comparison of PMSM speed response under different intelligent algorithms

At the 3rd second, when accelerating the speed of PMSM from 700 rad/s to 710 rad/s, the speed change response is shown in Figure 7. ADESA exhibits the shortest settling time (0.089s) and the smallest overshoot (4.6 rad/s). The response times of PSO and FA algorithms are too long, making it challenging to reach a steady state quickly. Additionally, FPA and SSA have overshoot values much larger than ADESA.

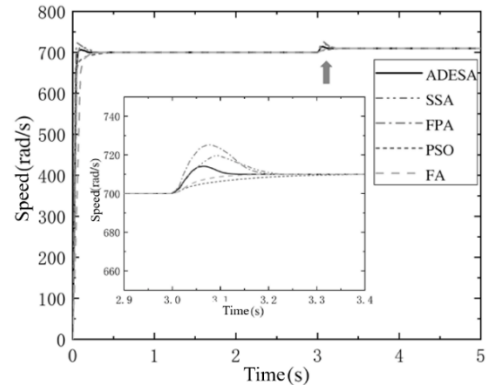


Figure 7. Variable speed response of PMSM with different intelligent algorithms

After applying a negative load of 5 N·m to PMSM at the 3rd second, the speed fluctuations for different algorithms are shown in Figure 8. Compared to the non-improved Squirrel Search Algorithm and other intelligent algorithms, ADESA exhibits the smallest fluctuations (7.6 rad/s), indicating that the PMSM has strong anti-disturbance capabilities under the control of ADESA.

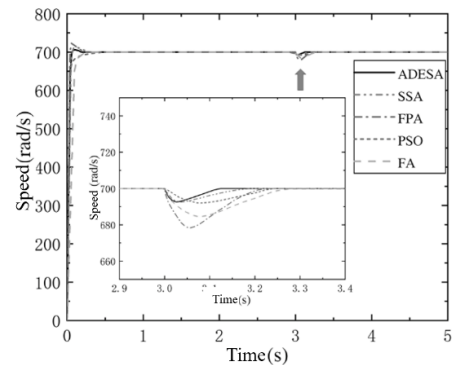


Figure 8. Velocity response of PMSM surge load with different intelligent algorithms

After applying a negative load of $5\text{N}\cdot\text{m}$ to PMSM, the torque variations for different algorithms are shown in Figure 9. Under the ADESA algorithm, the steady-state time of PMSM electromagnetic torque is 0.074s , with an overshoot of $0.34\text{N}\cdot\text{m}$ and minimal fluctuation. This indicates that the PMSM operates more smoothly under the control of ADESA.

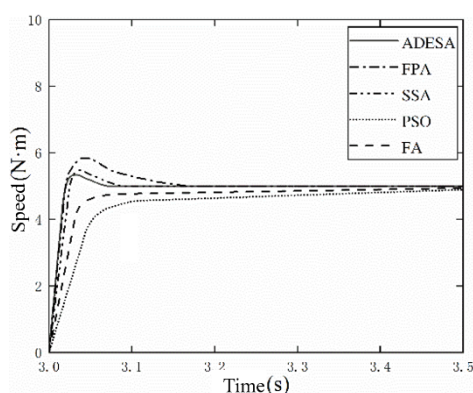


Figure 9. Variation of PMSM torque with different intelligent algorithms

6. Conclusion

In addressing the issues of low precision and slow response time in the PID control system for PMSM, this paper proposes the ADESA fractional-order PID control method for PMSM. By incorporating the crossover operation from differential algorithms and introducing adaptive predation probability factors, dynamic crossover probabilities, and perturbations based on Gaussian distribution, the global search and local exploitation capabilities of the squirrel algorithm are enhanced. The ADESA-tuned fractional-order PID is applied to control PMSM speed, and simulation comparative experiments are conducted.

The results indicate that ADESA fractional-order PID control for PMSM speed exhibits excellent control performance, including fast convergence, small overshoot, disturbance resistance, and stable control. It shows promising prospects for a wide range of applications.

Acknowledgment

This work was supported by the Sichuan Provincial Science and Technology Support Plan Project (2017FZ0033), Sichuan Provincial Bureau of Market Supervision and Pipeline Science and Technology Plan Project (CSCJZ2022007) and Chengdu Technology Innovation R&D Project (1).

Funding This study was supported by the Sichuan Provincial Science and Technology Support Plan Project (2017FZ0033), Sichuan Provincial Bureau of Market Supervision and Pipeline Science and Technology Plan Project (CSCJZ2022007) and Chengdu Technology Innovation R&D Project (2018-YF05-00201-GX).

Declarations

Ethics approval The authors declare that this manuscript was not submitted to more than one journal for simultaneous consideration. The submitted work is original and not has been published elsewhere in any form or language

Competing interests The authors declare that they have no conflict of interest.

References

- [1] Dong JY, Huang YK, Jin L, et al. Review on High Speed Permanent Magnet Machines Including Design and Analysis Technologies[J]. Proceedings of the CSEE, 2014,34(27):4640-4653.
- [2] Zhang C, Zhang RT, Lou YJ, et al. Torque control of permanent magnet synchronous motor based on engineering design method[J]. Manufacturing Automation, 2020, 42(04): 130-133.
- [3] Wang HB, Zhou B, Fang SC. A PMSM Sliding Mode Control System Based on Exponential Reaching Law [J]. TRANSACTIONS OF CHINA ELECTROTECHNICAL SOCIETY, 2009, 24(09):71-77.
- [4] Cui JR, Li Q, Zhang B, et al. Permanent Magnet Synchronous Motor of Variable Universe Adaptive Fuzzy PID Control[J]. Proceedings of the CSEE, 2013,33(S1):190-194.
- [5] Podlubny, Fractional Differential Equations[M], Academic Press, New York, 1999.
- [6] Chen YD, Yang M, Xu HL, et al. Vehicle ABS fractional PID control with parameter tuning using genetic algorithm[J]. manufacturing automation ,2018,40(01):24-27+39.
- [7] Khather S I, Almageed M, Abdullah A I. Fractional order based on genetic algorithm PID controller for controlling the speed of DC motors[J]. International Journal of Engineering & Technology, 2018, 7(4): 5386-5392.
- [8] Kumar P, Chatterjee S, Shah D, et al. On comparison of tuning method of FOPID controller for controlling field controlled DC servo motor [J]. Cogent Engineering, 2017, 4(1): 1357875.
- [9] Sultan G A, Sheet S M I A F, Fare Z K. Speed control of DC motor using fractional order PID controller based on particle swarm optimization[J]. Indonesian Journal of Electrical Engineering and Computer Science, 2021, 22(3): 1345.
- [10] Wu LF, Yang XZ. Design of Fractional Order PID Controller Based on Adjusting Cuckoo Search Algorithm[J]. Control Engineering of China, 2023,30(09):1673-1678.
- [11] Rajasekhar A, Das S, Abraham A. Fractional Order PID Controller Design for Speed Control of Chopper Fed DC Motor Drive Using Artificial Bee Colony Algorithm [J]. WORLD CONGRESS ON NATURE AND BIOLOGICALLY INSPIRED COMPUTING (NABIC), 2013: 259-66.
- [12] Chiranjeevi T, Babu N R, Yadav A, et al. Control of electric machines using flower pollination algorithm based fractional order PID controller [J]. Global Transitions Proceedings, 2021, 2(2): 227-232.
- [13] Jain M, Singh V, Rani A. A novel nature-inspired algorithm for optimization: Squirrel search algorithm[J]. Swarm and Evolutionary Computation, 2019, 44:148-175.
- [14] Storn R, Price K. differential evolution-a simple and efficient heuristic for global optimization over continuous spaces[J]. Journal of Global Optimization, 1997, 11(4): 341-359.
- [15] Chen CB, Hu HT, Gao S. Parameters Adaptive Design of Fractional Order PID Controller Based on Artificial Bee Colony Algorithm[J]. Control Engineering of China, 2020, 27(6): 956-961.
- [16] Liu YF, Peng Z, Wang YH, et al. PID parameter tuning of brushed direct-current motor based on improved genetic algorithm[J]. Journal of Computer Applications, 2022, 42(05): 1634-1641.