Speed Control of Induction Motor Drives Based on Combining Slime Mould Optimization Algorithm and Sliding Mode Theory

Kuei-Hsiang Chao * and Kuan-Chih Chang

Posted Date: 20 April 2026

doi: 10.20944/preprints202604.1268.v1

Keywords: induction motor; field-oriented control; slime mould algorithm; zebra optimization algorithm; sliding mode controller; exponential reaching law; constant speed reaching law

 doi:10.20944/preprints202604.1268.v1

Article

Speed Control of Induction Motor Drives Based on Combining Slime Mould Optimization Algorithm and Sliding Mode Theory

Kuei-Hsiang Chao * and Kuan-Chih Chang

Department of Electrical Engineering, National Chin-Yi University of Technology, Taichung 41170, Taiwan

* Correspondence: chaokh@ncut.edu.tw; Tel: +886-4-2392-4505 (ext.7272); Fax: +886-4-2392-2156

Abstract

This study addresses the speed control problem of an induction motor (IM) under the field-oriented control (FOC) architecture by proposing a robust controller design that combines the slime mould algorithm (SMA) with sliding mode theory (SMT). Distinct from traditional controller designs with fixed gains, the proposed theory defines the ranges of three gain parameters of the exponential reaching law sliding mode controller—namely, the sliding mode dynamic trajectory control gain, the exponential reaching gain, and the constant speed reaching gain—as the search space for the SMA. An adaptive fitness function is constructed using speed error and the rate of change of speed error to continuously evaluate and update these three gain parameters, thereby determining the optimal gains for the current state. This method allows the system to increase gain values to accelerate reaching when far from the sliding surface, and reduce gains to suppress chattering and minimize overshoot when approaching the sliding surface. Finally, Matlab/Simulink simulation software is used to simulate the proposed robust controller applied to an IM drive system. Its performance is compared with three other controllers: constant speed reaching law, exponential reaching law, and zebra optimization algorithm (ZOA) combined with exponential reaching law. Simulation results confirm that the proposed novel controller demonstrates control performance superior to the other three controllers in both speed command tracking and load regulation response.

Keywords: induction motor; field-oriented control; slime mould algorithm; zebra optimization algorithm; sliding mode controller; exponential reaching law; constant speed reaching law

1. Introduction

Induction motors play a key role in modern industrial automation and precision transmission fields due to their rugged structure, low maintenance costs, and high reliability [1]. However, as industrial applications impose increasingly stringent requirements on speed response and dynamic control precision, traditional scalar control is no longer sufficient. Consequently, the industry has widely adopted field-oriented control (FOC) [2] technology to achieve decoupled control of torque and flux. Traditional FOC typically uses proportional-integral (P-I) controllers. While P-I controllers are relatively easy to design, they struggle to balance speed command tracking and load regulation response simultaneously. Thus, when the system faces parameter inaccuracies or external disturbances, control performance is often insufficient.

To overcome these limitations, academia has proposed various advanced control laws, but each has bottlenecks. For instance, the sliding mode controller (SMC) [3] offers excellent anti-interference capabilities, but high-frequency chattering near the sliding surface can lead to torque ripple. Furthermore, the commonly used constant speed reaching law lacks integral action, leading to steady-state errors, while the exponential reaching law can accelerate response but often causes significant overshoot. To enhance performance, research has attempted to combine different algorithms. For example, extension theory combined with sliding mode controller (ETERLSMC) [3]

 doi:10.20944/preprints202604.1268.v1

can effectively suppress overshoot and improve tracking stability, but its performance relies heavily on the completeness of characteristic data; a lack of sufficient data samples leads to imprecise control. Neural network sliding mode controller (NNSMC) [4] can approximate nonlinear dynamics to reduce chattering, but its complex structure creates a massive computational burden, and performance depends on training convergence, making hardware implementation challenging. Extendable fuzzy theory combined with a two-degree-of-freedom controller (2DOFC) [5] improves system flexibility, but rule switching during load recovery often causes transient oscillations, resulting in insufficient convergence smoothness.

To solve the optimization problem of sliding mode parameters, swarm intelligence algorithms have been introduced. However, particle swarm optimization (PSO) [6], while fast to converge, lacks a mutation mechanism and easily falls into local optima. The zebra optimization algorithm (ZOA) [7] possesses a group defense mechanism to avoid local solutions, but it mostly uses fixed weights or linear decrease strategies during the search, resulting in an insufficient balance between global search and local exploitation when handling high-dimensional coupled control parameters.

The control law proposed in this paper uses the speed difference between the induction motor's speed command and actual speed, along with its rate of change, as features to calculate current fitness. It utilizes the slime mould algorithm (SMA) [8] to set the parameter ranges for the sliding mode dynamic trajectory control gain, exponential reaching gain, and constant speed reaching gain of the combined exponential reaching law SMC as a three-dimensional search space. Unlike traditional optimization algorithms that rely on fixed or linear weights, this study simulates the adaptive weight adjustment of slime mould under different food concentrations and its unique oscillation mechanism. This allows the algorithm to dynamically change step size based on environmental feedback, precisely finding the optimal position coordinates to assign to the three gain parameters of the exponential reaching law sliding mode velocity controller. Consequently, during IM operation, the most suitable gain combination is output in real-time by comparing fitness values, improving speed command tracking response and overshoot caused by the exponential reaching law under different working conditions, and significantly enhancing system robustness under load variations.

2. Field-Oriented Control Architecture of IM System

The core principle of FOC [9] is to transform the three-phase stator voltage and current variables of the IM into a two-axis synchronous rotating (d,q) coordinate system via coordinate transformation, simplifying its mathematical model to reduce controller design complexity [10,11]. In this architecture, the speed controller compensates based on the error between the speed command and feedback speed, while current controllers regulate the d and q axis currents respectively, achieving independent control of flux and torque. Finally, the voltage commands generated by the controllers control the duty cycle of the power semiconductor switches in the inverter [12] via space vector pulse width modulation (SVPWM) [13] to realize speed control of the IM drive.

2.1. Dynamic Equations of FOC

The purpose of FOC is to give the IM control characteristics similar to a separately excited DC motor [14], realizing independent control of field and torque. Based on this, the dynamic equations of the squirrel-cage induction machine in the d and q axis coordinate system are derived from the relationship between voltage and flux linkage. The stator and rotor voltage equations can be expressed as Equations (1)–(4) [10,11].

$$v_{ds} = R_s i_{ds} + L_\sigma \frac{di_{ds}}{dt} - \omega_e L_\sigma i_{qs} + \frac{L_m}{L_r}\frac{d\phi_{dr}}{dt} - \frac{\omega_e L_m}{L_r}\phi_{qr} \tag{1}$$

$$v_{qs} = \omega_e L_\sigma i_{ds} + R_s i_{qs} + L_\sigma \frac{di_{qs}}{dt} + \frac{\omega_e L_m}{L_r}\phi_{dr} + \frac{L_m}{L_r}\frac{d\phi_{qr}}{dt} \tag{2}$$

 doi:10.20944/preprints202604.1268.v1

$$0 = -R_r L_m i_{ds} + R_r \phi_{dr} + L_r \frac{d\phi_{dr}}{dt} - (\omega_e - \omega_r) L_r \phi_{qr} \tag{3}$$

$$0 = -R_r L_m i_{qs} + (\omega_e - \omega_r) L_r \phi_{dr} + R_r \phi_{qr} + L_r \frac{d\phi_{qr}}{dt} \tag{4}$$

where: R_s is stator resistance, R_r is rotor resistance, L_s is stator inductance, L_r is rotor inductance, L_m is mutual inductance, ω_e is synchronous angular velocity, and ω_r is rotor electrical angular velocity. i_{ds} and i_{qs} are the d and q axis stator currents; ϕ_{dr} and ϕ_{qr} are the rotor d and q axis fluxes, and $L_\sigma \triangleq L_s - L_m^2 / L_r$ is the leakage inductance coefficient.

The electromagnetic torque and mechanical equations of the motor are expressed as Equation (5).

$$T_e = \frac{3P}{4} \frac{L_m}{L_r} (i_{qs}\phi_{dr} - i_{ds}\phi_{qr}) = T_L + J \frac{d\omega_{rm}}{dt} + B\omega_{rm} \tag{5}$$

where: T_L is load torque, J is the moment of inertia, B is the viscous friction coefficient, and the relationship between rotor electrical angular velocity ω_r and mechanical angular velocity ω_{rm} is $\omega_{rm} = (2/P)\omega_r$, where P is the number of poles.

If the magnitude and position of the rotor flux linkage can be obtained at any instant, and the d-axis of the synchronous rotating coordinate system is aligned with the rotor flux linkage and maintained in synchronous rotation, then the rotor flux linkage exists only on the d-axis, i.e., $\phi_{dr} = \phi_r$, and the q-axis flux $\phi_{qr} = 0$. Consequently, Equations (1)–(4) can be rewritten as Equations (6)–(9) [11].

$$\frac{di_{ds}}{dt} = \left(-\frac{R_s}{L_\sigma} - \frac{1-\sigma}{\sigma\tau_r}\right) i_{ds} + \omega_e i_{qs} + \frac{1-\sigma}{\sigma\tau_r L_m} \phi_r + \frac{v_{ds}}{L_\sigma} \tag{6}$$

$$\frac{di_{qs}}{dt} = -\frac{R_s}{L_\sigma} i_{qs} - \omega_e i_{ds} - \frac{1-\sigma}{\sigma L_m} \omega_e \phi_r + \frac{v_{qs}}{L_\sigma} \tag{7}$$

$$\frac{d\phi_r}{dt} = -\frac{R_r}{L_r} \phi_r + R_r \frac{L_m}{L_r} i_{ds} \tag{8}$$

$$-R_r \frac{L_m}{L_r} i_{qs} + \omega_{sl} \phi_r = 0 \tag{9}$$

where: $\tau_r = L_r / L_m$, $L_\sigma = \sigma L_s$ is the stator leakage inductance, $\sigma = 1 - L_m^2 / L_s L_r$ is the leakage coefficient, and $\omega_{sl} = \omega_e - \omega_r$ is the slip speed. Since $\phi_{dr} = \phi_r$ and $\phi_{qr} = 0$, by substituting this condition into the electromagnetic torque Equation of (5), it can be rewritten as Equation (10). Furthermore, to describe the dynamic relationship between torque and speed, the mechanical equation of the motor can be further expressed as Equation (11).

$$T_e = \frac{3P}{4} \frac{L_m}{L_r} i_{qs} \phi_r \tag{10}$$

$$\frac{d\omega_{rm}}{dt} = \frac{1}{J} (T_e - T_L - B\omega_{rm}) \tag{11}$$

The flux command is the rated flux of the motor. However, since the rotor flux cannot be directly measured, it is necessary to obtain its estimated value through estimation. Therefore, by performing the Laplace transform on Equation (8), the estimated flux $\hat{\phi}_r$ can be obtained as Equation (12).

$$\hat{\phi}_r = \frac{L_m i_{ds}}{\frac{L_r}{R_r} s + 1} \tag{12}$$

where: $\hat{\phi}_r$ is the estimated flux and s is the Laplace operator.

From Equations (6) and (7), it can be seen that both the d and q axis current equations contain ordinary differential equation terms and nonlinear coupling terms. To eliminate the coupling effect and achieve system linearization, a control technique using feedforward compensation [15] is

 doi:10.20944/preprints202604.1268.v1

adopted. Using the feedback principle, the *d*-axis current error is passed through a P-I controller, and the resulting signal is Equation (13).

$$v_{ds}{}' = \left(K_{Pd} + \frac{K_{Id}}{s} \right) \left(i_{ds}{}^{*} - i_{ds} \right) \tag{13}$$

This signal is then combined with the output signal of the *d*-axis current. After coordinate transformation from two axes to three phases, it forms the three-phase voltage command input to the inverter. To ensure the current loop possesses linear control characteristics, appropriate compensation must be added to $v_{ds}{}'$ before coordinate transformation. Therefore, the *d*-axis voltage command is expressed as Equation (14).

$$v_{ds}{}^{*} = L_{\sigma} \left(v_{ds}{}' - \omega_e i_{qs} - \frac{1-\sigma}{\sigma \tau_r L_m} \hat{\phi}_r \right) \tag{14}$$

By substituting Equation (14) into v_{ds} in Equation (6), it can be expressed as Equation (15).

$$\frac{di_{ds}}{dt} = \left(-\frac{R_s}{L_{\sigma}} + \frac{1-\sigma}{\sigma \tau_r} \right) i_{ds} + v_{ds}{}' \tag{15}$$

Similarly, the *q*-axis current signal and voltage are given by Equation (16) and Equation (17), respectively.

$$v_{qs}{}' = \left(K_{Pq} + \frac{K_{Iq}}{s} \right) \left(i_{qs}{}^{*} - i_{qs} \right) \tag{16}$$

$$v_{qs}{}^{*} = L_{\sigma} \left(v_{qs}{}' + \omega_e i_{ds} + \frac{1-\sigma}{\sigma L_m} \omega_e \hat{\phi}_r \right) \tag{17}$$

By substituting Equation (17) into v_{qs} in Equation (7), it can be expressed as Equation (18).

$$\frac{di_{qs}}{dt} = -\frac{R_s}{L_{\sigma}} i_{qs} + v_{qs}{}' \tag{18}$$

The speed control loop takes the error between the actual speed obtained from the motor feedback signal and the speed command as input to a P-I controller to generate a torque command. This torque command is then converted into a *q*-axis current command via the current control loop, thereby driving the motor to achieve the goal of speed regulation. The mechanical dynamics of the motor can be constructed from its moment of inertia and viscous friction coefficient. Therefore, the Laplace transform [16] can be applied to Equation (5) to establish the mathematical model of the speed loop. Based on this, a controller is designed to ensure the system possesses good dynamic response and the ability to eliminate steady-state errors.

2.2. Sensorless FOC System Architecture of IM

The actual system still requires coordinate transformation and pulse width modulation (PWM) of the inverter to generate the corresponding three-phase voltage signals. In addition, this paper adopts a sensorless direct FOC strategy, so it is necessary to obtain the current flux and speed via a flux estimator [17] and a speed estimator [18]. In view of this, the complete architecture of the traditional sensorless FOC IM drive system is shown in Figure 1 [10].

 doi:10.20944/preprints202604.1268.v1

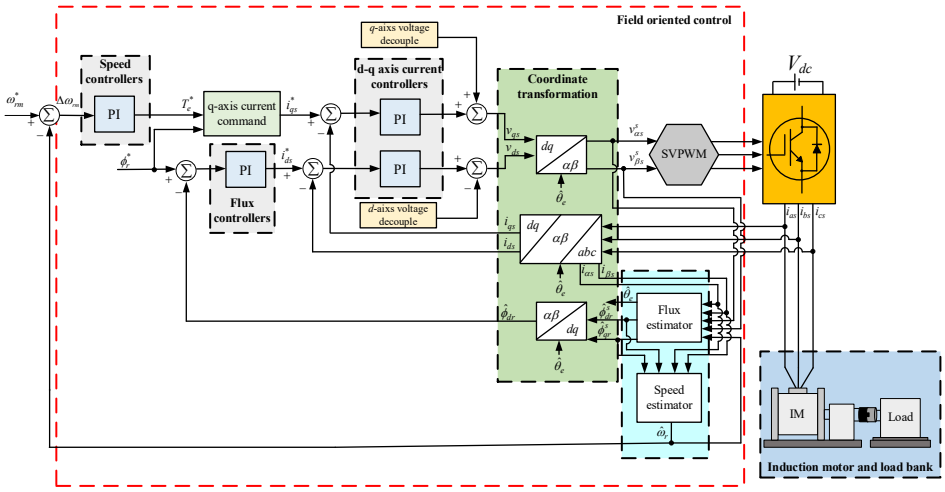

Figure 1. Block diagram of the traditional sensorless FOC IM drive system.

3. The Proposed Novel Controller Algorithm

In traditional FOC of induction motors, the speed loop uses only a P-I controller, which often struggles to simultaneously balance the fast tracking of speed commands and regulation capability under load disturbances. Therefore, this paper proposes an exponential reaching law sliding mode controller combined with the SMA for dynamic gain parameter adjustment to replace the P-I controller in the speed loop. Specifically, taking the real-time speed error and the rate of change of the speed error as inputs, the SMA performs an optimization search to automatically correct the gains of the SMC. This equips the controller with self-adjustment capabilities and robustness, thereby effectively enhancing the overall performance of the IM speed control.

3.1. Slime Mould Algorithm

The SMA is an emerging bio-inspired swarm intelligence optimization method, proposed by Shimin Li et al. in 2020 [8]. Its principle mimics the typical behaviors of slime mould during the food foraging process, including the approach behavior of gradually moving closer to food based on odor gradients, the wrap behavior of forming energy transmission centers through the strengthening and weakening of veins, and the oscillation behavior generated by the periodic contraction of veins. Through the aforementioned behaviors, the SMA is able to balance exploration and exploitation during the search process, gradually approaching the global optimal solution.

3.1.1. Approach Behavior

In nature, slime mould initially spreads out a “radial extension network”, resembling random antennae, to explore its surroundings. If a higher concentration of chemical odor is detected in a particular direction, the slime mould gradually concentrates energy to extend toward that direction, forming a temporary pathway. If the odor is insufficient or the direction is unclear, the slime mould changes direction and performs random drifting to avoid missing potential resources. This process, from rapid perception to approaching and then correcting direction, allows the slime mould to continuously update its movement path. Subsequently, in the iteration process, the movement of each slime mould depends on the fitness value corresponding to the food odor intensity and the best slime mould in the population. The mathematical expression for this approach behavior is represented by Equation (19).

$$\overrightarrow{X(t+1)} = \begin{cases} \overrightarrow{X_b(t)} + \overrightarrow{vb} \cdot (\overrightarrow{W} \cdot \overrightarrow{X_A(t)} - \overrightarrow{X_B(t)}) & , r < p \\ \overrightarrow{vc} + \overrightarrow{X(t)} & , r \geq p \end{cases} \tag{19}$$

 doi:10.20944/preprints202604.1268.v1

where: $\overrightarrow{X(t+1)}$ represents the new position for the approach food behavior, $\overrightarrow{X_b(t)}$ represents the optimal position of the slime mould individual at the *t*-th iteration, $\overrightarrow{vc}$ is a control vector that linearly decreases from one to zero, $\overrightarrow{X_A(t)}$ and $\overrightarrow{X_B(t)}$ represent the positions of two individuals randomly selected from the slime mould, $\overrightarrow{X(t)}$ is the current position, *t* is the current iteration count, and *r* is a random number in the range $[0,1]$.

In this process, in order to simulate the reaction of the slime mould to the strength of the odor, the SMA introduces a switching probability *p*. When the fitness value of a certain individual is better than the average value, *p* will tend toward the positive direction, causing it to move and approach the best individual; conversely, it will bias toward the negative direction, causing it to stay or move away, thereby maintaining population diversity. The mathematical expression for *p* is shown in Equation (20).

$$p = \tanh\left|S(i) - DF\right| \tag{20}$$

where: *p* is the switching probability, $S(i)$ is the fitness value of the *i*-th slime mould, and *DF* is the best fitness value found in all iterations.

Furthermore, to reflect the periodic contraction to expansion of the slime mould veins, the SMA designs an oscillation amplitude parameter $\overrightarrow{vb}$. This vector has a larger range of values ($[-a,a]$) in the initial stage, enabling the slime mould to explore with large steps; as the iteration progresses, a gradually approaches zero, and the amplitude of $\overrightarrow{vb}$ converges synchronously, making the movement in the later stages more fine-grained. The mathematical expression for $\overrightarrow{vb}$ is represented by Equations (21) and (22).

$$\overrightarrow{vb} = [-a, a] \tag{21}$$

$$a = \arctan h\left(-\left(\frac{t}{T}\right)+1\right) \tag{22}$$

where: $\overrightarrow{vb}$ is the oscillation amplitude parameter, and *T* is the maximum number of iterations.

In the mathematical modeling of the SMA, whether an individual is closer to the optimal solution depends not only on the position of the current best individual but is also influenced by its "relative fitness". To reflect this difference, the algorithm introduces a weight $\overrightarrow{W}$, the value of which is dynamically adjusted based on the individual's ranking in the population (SmellIndex) and the difference between the best and worst fitness. When an individual's performance is close to the optimal solution, $\overrightarrow{W}$ takes a larger positive value, making it converge more actively toward the optimal solution; conversely, individuals with poorer performance receive smaller or even lower weights, corresponding to a "weakening of the approach behavior ", to avoid the entire population concentrating too quickly. This design compresses the value range through a log function to ensure $\overrightarrow{W}$ varies within a reasonable interval. The mathematical expression for $\overrightarrow{W}$ is represented by Equations (23) and (24).

$$\overrightarrow{W(SmellIndex(i))} = \begin{cases} 1 + r \cdot \log\left(\dfrac{bF - S(i)}{bF - wF} + 1\right), & condition \\ 1 - r \cdot \log\left(\dfrac{bF - S(i)}{bF - wF} + 1\right), & other \end{cases} \tag{23}$$

$$SmellIndex = sort(S) \tag{24}$$

 doi:10.20944/preprints202604.1268.v1

where: *bF* is the best fitness value in the current iteration, *wF* is the worst fitness value in the current iteration, *SmellIndex* is the sequence of fitness values of the slime moulds, and *condition* refers to the slime mould individuals ranked in the first half of fitness.

3.1.2. Wrap Behavior

When the slime mould finds food, it does not approach from a single direction alone; instead, it gradually contracts and strengthens the surrounding veins, ultimately wrapping around the food to become a center for cytoplasm energy transport. During this process, high-nutrient paths leading to the food gradually thicken, allowing a greater influx of material, whereas ineffective or nutrient-poor paths gradually contract until they vanish to conserve energy. Furthermore, a portion of the slime mould continues to diffuse outward, maintaining population diversity to ensure other potential resources can be located should the current food source become insufficient. This mechanism of "concurrent concentration and contraction" enables the slime mould to pinpoint the food location while avoiding excessive convergence. The mathematical expression for this wrap food behavior is represented by Equation (25).

$$\overrightarrow{X^*} = \begin{cases} rand \cdot (UB\text{-}LB) + LB & , rand < z \\ \overrightarrow{X_b(t)} + \overrightarrow{vb} \cdot (\overrightarrow{W} \cdot \overrightarrow{X_A(t)} - \overrightarrow{X_B(t)}) & , r < p \\ \overrightarrow{vc} + \overrightarrow{X(t)} & , r \geq p \end{cases} \quad (25)$$

where: *UB* is the upper bound of the search space, *LB* is the lower bound of the search space, *rand* is a random number in $[0,1]$, and *z* is the probability threshold.

3.1.3. Oscillation Behavior

In nature, even after the slime mould has found food, its veins still maintain periodic contraction and dilation. This "oscillation" behavior avoids local stagnation, ensures the continuous flow of cytoplasm, and allows for direction adjustment at any time. The random changes brought about by oscillation enable partial pathways to temporarily reverse or weaken, maintaining the population's exploration capability. On the other hand, as time progresses, the oscillation amplitude gradually diminishes, allowing energy to ultimately concentrate on the best food source, completing the transition from "global" to "local". In the mathematical modeling of the SMA, this oscillation behavior is not determined by a single factor but is realized through the combined action of three elements: the weight $\overrightarrow{W}$ mentioned earlier that strengthens or weakens the contribution of individuals with high or low fitness; the stochastic vector $\overrightarrow{vb}$ which simulates the randomness of vein flow direction switching; and the random control vector $\overrightarrow{vc}$ whose range linearly decreases to zero with the total number of iterations. In other words, oscillation is the core mechanism for maintaining the balance between exploration and convergence during the evolutionary process of the SMA, enabling the algorithm to both escape local optima and gradually converge to the global optimum.

3.2. Sliding Mode Controller

In general, under most circumstances, traditional P-I controllers can achieve satisfactory control performance under fixed operating conditions. However, because the dynamic behavior of induction motors is highly nonlinear and involves multivariable coupling, their equivalent parameters change with load, temperature, and magnetic saturation, leading to uncertainty in the system model. Coupled with external disturbances and sudden load changes, it is often difficult for simple P-I control to maintain stable performance. To overcome these limitations, intelligent control methods are often introduced in research, or multiple control strategies are combined, to provide the controller with better robustness and self-adaptive capabilities.

 doi:10.20944/preprints202604.1268.v1

Among them, SMC belongs to the category of variable structure control (VSC) [19]. Its advantage is a lower dependence on the precise model of the system. By designing a suitable sliding mode surface, the nonlinear dynamics of the IM can exhibit characteristics similar to a linear system during the sliding mode phase, thereby providing good anti-interference capabilities and robustness, making it particularly suitable for IM control requiring high-speed response and high reliability.

3.2.1. State Variable Design

In speed tracking control, the objective is to make the error value between the speed command ω_{rm}^{*} and the actual feedback speed ω_{rm} zero, i.e., $\omega_{rm}^{*}-\omega_{rm}=0$. Simultaneously, the rate of change of the error must also converge to 0 to ensure the system achieves a steady state. At this time, the output of the controller is the q-axis current command i_{qs}^{*}, so the state variables of the IM system can be expressed as Equation (26).

$$\begin{cases} x_1=\omega_{rm}^{*}-\omega_{rm} \\ x_2=\dot{x}_1=-\dot{\omega}_{rm} \end{cases} \tag{26}$$

By performing differentiation based on Equations (11) and (26), the dynamic relationship of the state variables can be obtained, expressed as Equation (27).

$$\begin{cases} \dot{x}_1=-\dot{\omega}_{rm}=\dfrac{1}{J}(T_L+B\omega_{rm}-\dfrac{3P}{4}\dfrac{L_m}{L_r}i_{qs}\phi_r) \\ \dot{x}_2=-\ddot{\omega}_{rm}=-\dfrac{1}{J}\left(\dfrac{3P}{4}\dfrac{L_m}{L_r}\dot{i}_{qs}\,\phi_r+B\dot{\omega}_{rm}\right) \end{cases} \tag{27}$$

Let $D=\dfrac{1}{J}\left(\dfrac{3P}{4}\dfrac{L_m}{L_r}\phi_r\right)$, and substitute it into Equation (27) to obtain Equation (28).

$$\dot{x}_2=-\ddot{\omega}_m=-\frac{1}{J}\left(\frac{3P}{4}\frac{L_m}{L_r}\dot{i}_{qs}\,\phi_r\right)+\frac{B}{J}\dot{\omega}_{rm}=-D\dot{i}_{qs}-\frac{B}{J}x_2 \tag{28}$$

3.2.2. Sliding Mode Dynamic Trajectory Design

The core concept of sliding mode control is to cause the system state to move along a sliding mode dynamic trajectory. This sliding mode dynamic trajectory function can be defined through a function composed of a set of state variables and control gains, as shown in Equation (29).

$$s_1=cx_1+x_2 \tag{29}$$

where: s_1 is the sliding mode dynamic trajectory function, c is the sliding mode dynamic trajectory control gain, and x_1 and x_2 are the state variable functions.

To achieve system stability, the conditions $x_1=0$ and $x_2=0$ must be satisfied. When the sliding mode function $s_1(x_1,x_2)=0$, let $x_2=\dot{x}_1$, and we obtain $cx_1+x_2=cx_1+\dot{x}_1=0$. From this, the system state variables x_1 and x_2 can be derived, as shown in Equation (30).

$$\begin{cases} x_1=x_1(0)e^{-ct} \\ x_2=-cx_1(0)e^{-ct} \end{cases} \tag{30}$$

In summary, as time progresses, both state variables x_1 and x_2 gradually decay to 0 at an exponential rate. In other words, when $s_1=0$, it indicates that the system is operating along the designed sliding mode dynamic trajectory. When the sliding mode dynamic trajectory function is designed as $s_1=cx_1+x_2$, and after the system reaches the sliding mode dynamic trajectory $s_1=0$, its state variables will gradually approach zero, thereby achieving the objective of state variable control.

 doi:10.20944/preprints202604.1268.v1

3.2.3. Sliding Mode Reaching Law Design

To ensure that the sliding mode dynamic trajectory function s_1 can converge to 0 at a certain point in time (i.e., the system reaches the sliding mode dynamic trajectory) and maintain stable operation, a suitable reaching law function must be designed [20]. According to the design of the sliding mode dynamic trajectory, to achieve $s_1 = 0$, an appropriate control output function u must be designed to satisfy the control conditions. Therefore, Equation (29) can be rearranged in to Equation (31).

$$\dot{s}_1 = c\dot{x}_1 + \dot{x}_2 = cx_2 + \dot{x}_2 = cx_2 - D\dot{i}_{qs} - \frac{B}{J}x_2 \tag{31}$$

Based on the fact that the controller output i_{qs}^* is the q-axis reference current, let the output function be $u = \dot{i}_{qs}$. Thus, Equation (31) can be rewritten as Equation (32).

$$\dot{s}_1 = \left(c - \frac{B}{J}\right)x_2 - Du \tag{32}$$

where: $\dot{s}_1$ is the reaching law function.

According to the Lyapunov stability second method [21], assuming there exists a continuous function V that satisfies three conditions, it can be determined that the system is at the equilibrium point $s_1 = 0$, i.e., $\lim_{t\to\infty} s_1(t) = 0$, as shown in Equation (33).

$$\begin{aligned} &(1)V(0) = 0 \\ &(2)V(s_1) > 0, s_1 \neq 0 \\ &(3)\dot{V}(s_1) < 0, s_1 \neq 0 \end{aligned} \tag{33}$$

In addition, if $V = s_1^2/2$ is set as the Lyapunov function, it can simultaneously satisfy condition (1) and condition (2) in Equation (33), and the third condition can be further derived from its time differentiation, as shown in Equation (34).

$$\dot{V}(x) = s_1\dot{s}_1 \tag{34}$$

Based on the aforementioned derivation, in order for the system to satisfy the stability condition, it is necessary to design an appropriate reaching law function $\dot{s}_1$ such that $\dot{V}(x) = s_1\dot{s}_1 < 0$ holds. Common forms of reaching laws mainly include the constant speed reaching law and the exponential reaching law. Among them, the constant speed reaching law and the exponential reaching law can be expressed as Equation (35) and Equation (36), respectively.

$$\dot{s}_1 = -\varepsilon\,\mathrm{sgn}(s_1), \varepsilon > 0 \tag{35}$$

$$\dot{s}_1 = -\varepsilon\,\mathrm{sgn}(s_1) - qs_1, \varepsilon > 0, q > 0 \tag{36}$$

where: ε is the constant speed reaching gain, q is the exponential reaching gain, and $\mathrm{sgn}(s_1)$ is a function symbol, expressed as Equation (37).

$$\mathrm{sgn}(s_1) = \begin{cases} 1, & s_1 > 0 \\ -1, & s_1 < 0 \end{cases} \tag{37}$$

Based on the settings of the two aforementioned reaching law functions, an analysis of the sliding mode variable yields its dynamic relationship as $\dot{s}_1 = -\varepsilon\,\mathrm{sgn}(s_1), \varepsilon > 0$. From Equation (32), the controller's $u = -cx_2 - \varepsilon\,\mathrm{sgn}(s_1) - qs_1$ can be derived, serving as the control force function u applied to the motor model. Under this control action, the system state is guided to the designed sliding mode dynamic trajectory, causing the final response to stabilize at the origin point corresponding to said sliding mode dynamic trajectory.

 doi:10.20944/preprints202604.1268.v1

3.2.4. Controller Output Design

According to the aforementioned reaching law design, the first two conditions of the Lyapunov function in Equation (33) can ensure $\lim_{t\to\infty} s_1(t) = 0$. However, if the constant speed reaching law in Equation (35) is adopted, although the conditions of the Lyapunov function can be satisfied regardless of whether $t = 1$ s or $t = 100$ s when $s_1 = 0$ (meaning the sliding mode dynamic trajectory is reached), its dynamic performance is subject to additional limitations because its response speed is a fixed constant value. If its gain value is increased to accelerate the reaching speed, it is easy to cause significant chattering near the sliding mode dynamic trajectory; conversely, if the gain value is lowered to suppress chattering, the system requires a longer time to reach the sliding mode dynamic trajectory. Based on the above phenomena, the constant speed reaching law often struggles to simultaneously balance speed and stability in practical applications. Therefore, the exponential reaching law adopted in this paper introduces an exponential reaching term $-qs_1$. When the sliding mode dynamic trajectory function s_1 is small (i.e., the system state is close to the sliding mode dynamic trajectory), $\dot{s}_1 = -qs_1$ approaches 0, at which time the system dynamics are dominated by $-\varepsilon\,\mathrm{sgn}(s_1)$; conversely, when the sliding mode dynamic trajectory function is large (i.e., the system state is far from the sliding mode dynamic trajectory), the value of $\dot{s}_1 = -qs_1$ will also be large, enabling the system to move rapidly toward the sliding mode dynamic trajectory with a larger reaching speed. From this, it can be seen that the exponential reaching law can automatically adjust the reaching speed according to the system state, making it more suitable for application in control systems with large parameter fluctuation ranges or significant load disturbances.

In summary, the exponential reaching law possesses superior adaptive characteristics compared to the constant speed reaching law; therefore, this paper selects the exponential reaching law as the reaching function for the sliding mode surface. From Equation (36), it can be seen that when only the exponential reaching term $-qs_1$ is considered, the solution for its corresponding sliding mode dynamic trajectory function can be expressed as Equation (38).

$$s_1(t) = s_1(0)e^{-qt} \tag{38}$$

This paper adopts the exponential reaching law function as the applied reaching law. Furthermore, from Equation (29), it is known that i_{qs}^* is the output of the controller. Therefore, by letting the controller function $u = \dot{i}_{qs}$, Equation (32) can be rewritten as Equation (39).

$$\dot{s}_1 = \left(c - \frac{B}{J}\right)x_2 - Du = -\varepsilon\,\mathrm{sgn}(s_1) - qs_1 \tag{39}$$

The expression for the controller function u is represented by Equation (40).

$$u = \frac{1}{D}\left[\left(c - \frac{B}{J}\right)x_2 + \varepsilon\,\mathrm{sgn}(s_1) + qs_1\right] \tag{40}$$

From Equation (40), the q-axis reference current i_{qs}^* can be expressed as Equation (41).

$$i_{qs}^* = \frac{1}{D}\int\left[\left(c - \frac{B}{J}\right)x_2 + \varepsilon\,\mathrm{sgn}(s_1) + qs_1\right]dt \tag{41}$$

3.3. Speed Control of SMA Combined with SMC

Based on the aforementioned discussion, this paper adopts the exponential reaching law sliding mode controller as the speed controller for the IM drive system. However, to effectively suppress speed overshoot and enhance the speed and stability of speed tracking, this paper introduces the SMA to dynamically adjust the three gain parameters of the exponential reaching law sliding mode controller. Specifically, the upper and lower bounds of these parameters are set as the search space, and the speed error between the motor speed command and the actual speed, along with its rate of

 doi:10.20944/preprints202604.1268.v1

change, are used as features to establish the initial fitness value. The algorithm first evaluates the population fitness distribution to identify the current best position and best fitness value. Subsequently, it executes the wrap behavior via the second term of Equation (25), utilizing the aforementioned best solution to guide the population position and strengthen local search capabilities. Finally, the third term of Equation (25) is used to introduce an oscillation mechanism to avoid becoming trapped in local minima, thereby gradually approaching the global optimal solution and determining the suitable three gain parameters for the exponential reaching law sliding mode controller. The dynamic adjustment process of the SMA is detailed below, and its control flowchart is shown in Figure 2.

Step 1: Set relevant parameters: such as slime mould population size N, dimension dim, maximum number of iterations T, search space range $[lb_j, ub_j]$, random numbers r and $rand$, probability threshold z, and weights $\alpha(W_1)$ and $\beta(W_2)$.

Step 2: Initialize the position X and fitness value F of each slime mould.

Step 3: First, a slime mould individual is randomly selected as the initial search slime mould SM_j, and its position coordinate values are directly input into the exponential reaching law sliding mode controller. Subsequently, the integral of absolute error (IAE) is adopted to calculate the system feedback speed error e and the rate of change of the speed error $\dot{e}$, and the initial fitness value is calculated based on the set weight values $\alpha(W_1)$ and $\beta(W_2)$. The weight values represent the relative importance of each feature quantity. According to the analysis of the sliding mode dynamic trajectory function in Equation (23), it is known that this controller possesses compensation and correction effects regarding the rate of change of the speed error; therefore, in the initial stage, $\alpha(W_1)$=0.7 and $\beta(W_2)$=0.3 are set, with $\alpha(W_1)+\beta(W_2)=1$. In subsequent iterations, the weight update mechanism of the SMA in Equation (23) of Step 5 is used to dynamically adjust the aforementioned weights, and based on this, the IAE is utilized to calculate the fitness value F_i of the search slime mould, as shown in Equation (42).

$$F_i = \alpha(W_1)\cdot\sum_{t=1}^{T}\left|e_{i,t}\right| + \beta(W_2)\cdot\sum_{t=1}^{T}\left|\dot{e}_{i,t}\right| \tag{42}$$

where: $e_{i,t}$ and $\dot{e}_{i,t}$ are the error and the rate of change of the error of the i-th slime mould in the t-th iteration, respectively.

Step 4: Based on the fitness values calculated in Step 3, all slime moulds are sorted. The best fitness bF, the worst fitness wF, and the global best position X_{best} in the current population are recorded. This sorting result will serve as the basis for the subsequent calculation of weights.

 doi:10.20944/preprints202604.1268.v1

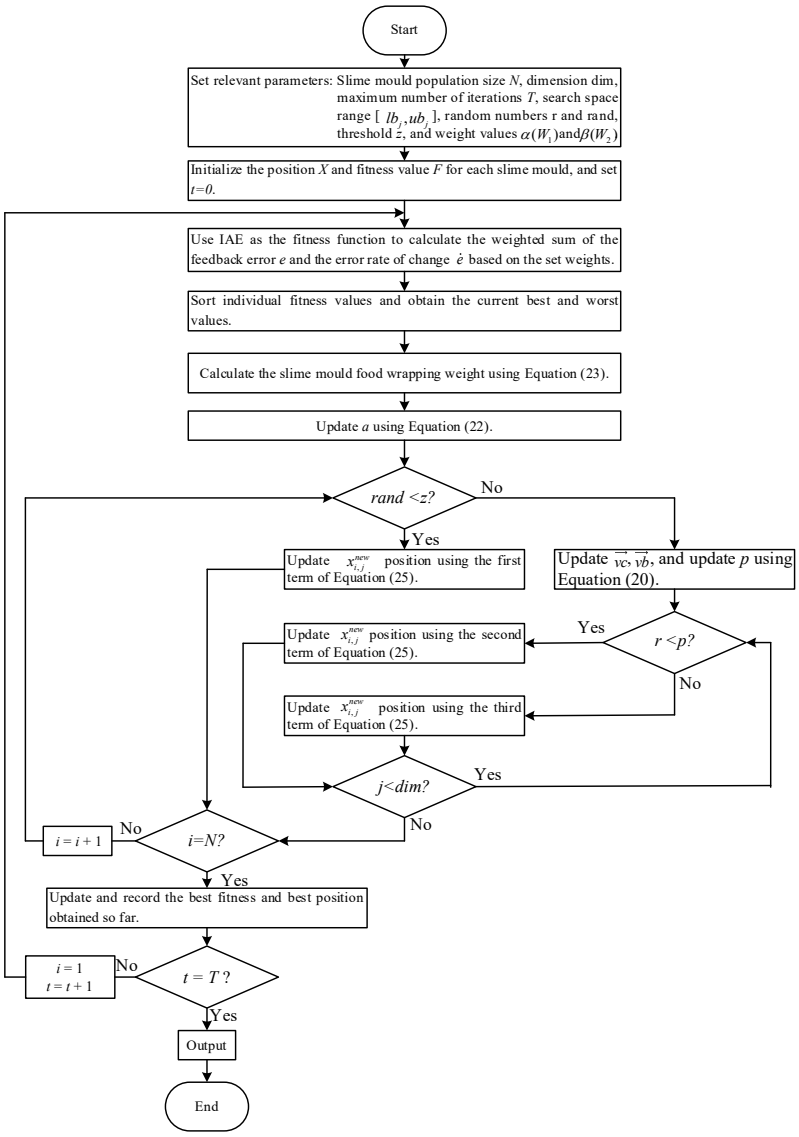

Figure 2. Control flowchart of the proposed novel SMA.

Step 5: At the current iteration number t, for each slime mould individual X_i, first calculate the approach food probability p using Equation (20) based on its current fitness value $S(i)$ and the best fitness bF in the population so far. Then, update the range a of the oscillation control parameter according to Equation (22) to determine the oscillation vector $\overrightarrow{vb}$, and simultaneously update the linearly decreasing control vector $\overrightarrow{vc}$. Finally, calculate the corresponding adaptive weight $\overrightarrow{W}$ via Equation (23) to provide the proportion for regulating exploration and exploitation behaviors during the subsequent position update. Furthermore, feed this weight back into the fitness function to serve as the basis for dynamically adjusting the weight coefficients $\alpha(W_1)$ and $\beta(W_2)$ within the integral of absolute error (IAE) formula.

Step 6: For each dimension j of each slime mould individual X_i, calculate its candidate position X_i^{new} according to the single integrated position update Equation (25) of the SMA. This mechanism first determines whether to execute the first term of Equation (25) by judging if $rand > z$; if the random distribution is not triggered, it decides whether to execute the second term of Equation (25) based on $r < p$; otherwise, it executes the third term of Equation (25). After completing the position update, use it as the basis for the subsequent fitness evaluation and population sorting.

 doi:10.20944/preprints202604.1268.v1

Step 7: After all slime mould individuals have completed their position updates, sort the population fitness, and update the worst fitness wF and the best fitness bF to the global best record values achieved so far.

Step 8: Update and record the best fitness value and best position obtained so far. If the iteration count has reached the preset maximum number of iterations, stop the iteration and output the coordinate values of the best position to adjust the three gain parameters of the exponential reaching law sliding mode controller, and determine the final output i_{qs}^{*} of the controller via Equation (41).

To illustrate the configuration of the overall control architecture, the proposed slime mould algorithm combined with exponential reaching law sliding mode controller (SMAERLSMC) is integrated into a traditional sensorless FOC IM drive system. The overall block diagram is shown in Figure 3.

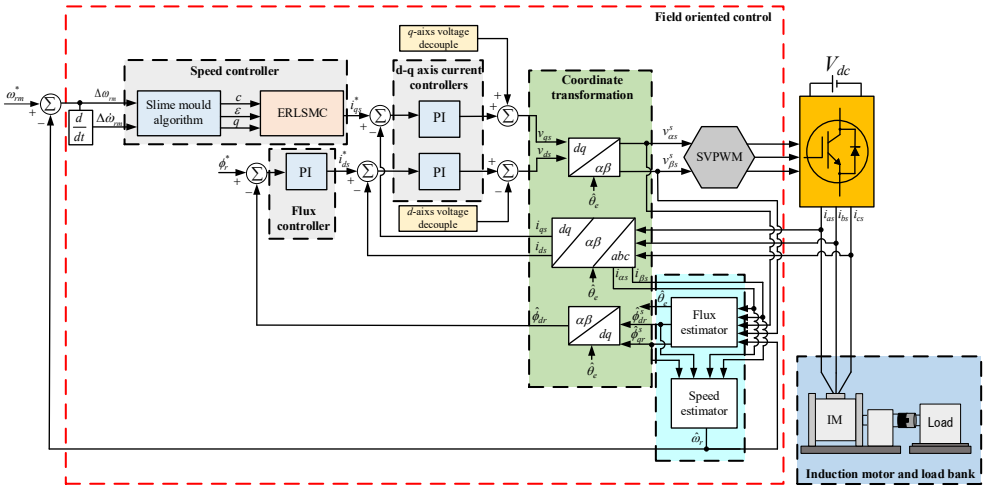

Figure 3. Block diagram of the sensorless FOC IM drive system utilizing the novel sliding mode speed controller combined with SMA.

4. Simulation Results

To verify the performance of the proposed robust controller, this paper utilizes Matlab/Simulink to conduct simulation verification of the speed control for the IM drive system. The motor specifications are shown in Table 1. The speed control performance of the IM is compared under identical test conditions using four types of controllers: the slime mould algorithm combined with exponential reaching law sliding mode controller (SMAERLSMC), the constant speed reaching law sliding mode controller (CSRLSMC), the exponential reaching law sliding mode controller (ERLSMC), and the zebra optimization algorithm combined with exponential reaching law sliding mode controller (ZOAERLSMC). Figures 4a–d and 5a–d present the simulation results of the speed command tracking response for the IM under loads of 1 N-m and 2 N-m, respectively, with the speed command rising from 0 to different speed commands (1000 rpm, 2000 rpm, 3000 rpm, 4000 rpm). Figure 6a–d show the simulation results of the load regulation response for the IM under different speed commands (1100 rpm, 2100 rpm, 3100 rpm, 4100 rpm), where the load torque variation is T_L: $0 \rightarrow 1 \rightarrow 0 \rightarrow 2 \rightarrow 0$ N-m. From the simulation results, it can be observed that the CSRLSMC, due to the absence of integral action in its control structure, cannot ensure that the speed stably converges to the speed command during actual operation control, and the tracking process requires a longer time; furthermore, in terms of load regulation response, its speed recovery time is longer, and it exhibits a larger speed drop. While the ERLSMC presents a faster response in speed command tracking, it easily produces overshoot during the dynamic tracking process, leading to concerns regarding system stability. From Figures 4 and 5, it can be observed that although the ZOAERLSMC

 doi:10.20944/preprints202604.1268.v1

can effectively suppress overshoot during the dynamic response process compared to the pure ERLSMC, this controller can still be further optimized in the balance between speed tracking and load regulation. Specifically, under load variations, its transient response and speed recovery capability remain insufficient. However, if the SMA is adopted to dynamically adjust the three gain parameters of the ERLSMC, it can not only effectively suppress the overshoot caused by the pure ERLSMC but also possess superior control performance in load regulation response compared to the other three types of controllers. Compared to other control strategies, the SMAERLSMC proposed in this paper can simultaneously balance the capabilities of command tracking speed and overshoot suppression, and regarding load regulation response, its overall control performance is superior to the other three types of controllers.

Table 1. Specifications of the adopted IM.

Electrical specifications	Values
Three-phase rated voltage	AC 180 V
Three-phase rated current	AC 7.65 A
Rated power	1.9 kW
Rated speed	4750 rpm
Operating frequency	80 Hz
Number of poles	2
Stator resistance	0.6797 Ω
Rotor resistance	0.6327 Ω
Armature inductance	53.423 mH
Mutual inductance	51.75 mH
Moment of inertia	0.003 $kg \cdot m^2$
Viscous friction coefficient	0.0001 $N \cdot m \cdot s$

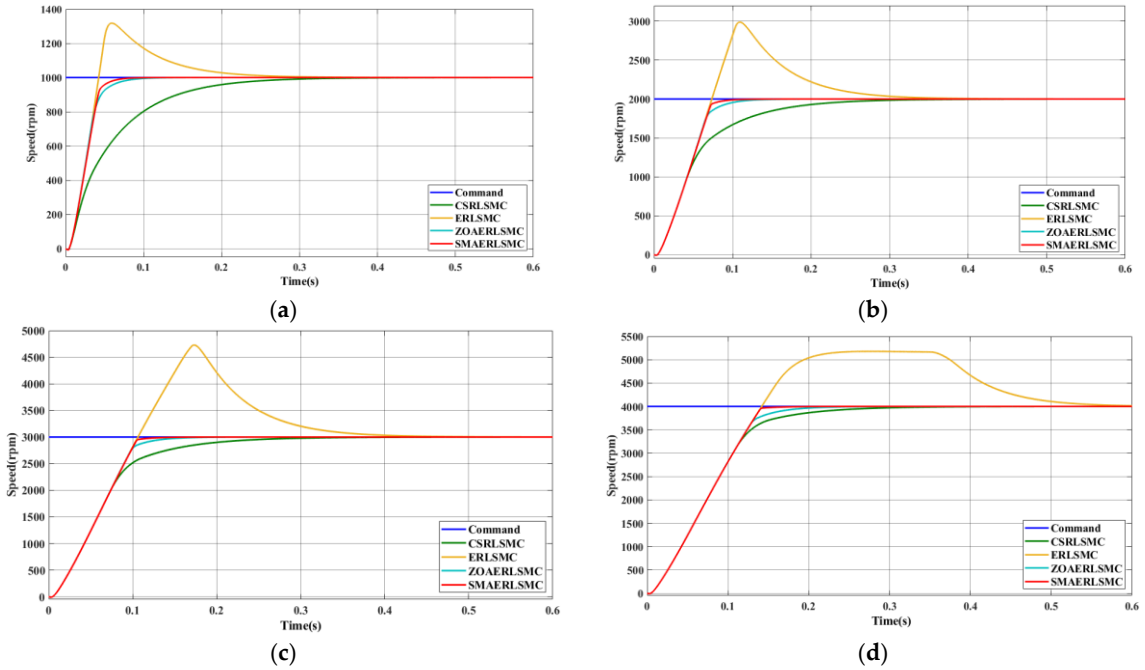

Figure 4. Comparison of tracking responses between the proposed SMAERLSMC, the CSRLSMC, the ERLSMC, and the ZOAERLSMC under a load of 1 N-m, where the speed command rises from 0 rpm to different speed commands: (**a**) $\omega_{rm}: 0 \rightarrow 1000$ rpm; (**b**) $\omega_{rm}: 0 \rightarrow 2000$ rpm; (**c**) $\omega_{rm}: 0 \rightarrow 3000$ rpm; (**d**) $\omega_{rm}: 0 \rightarrow 4000$ rpm.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 April 2026 doi:10.20944/preprints202604.1268.v1

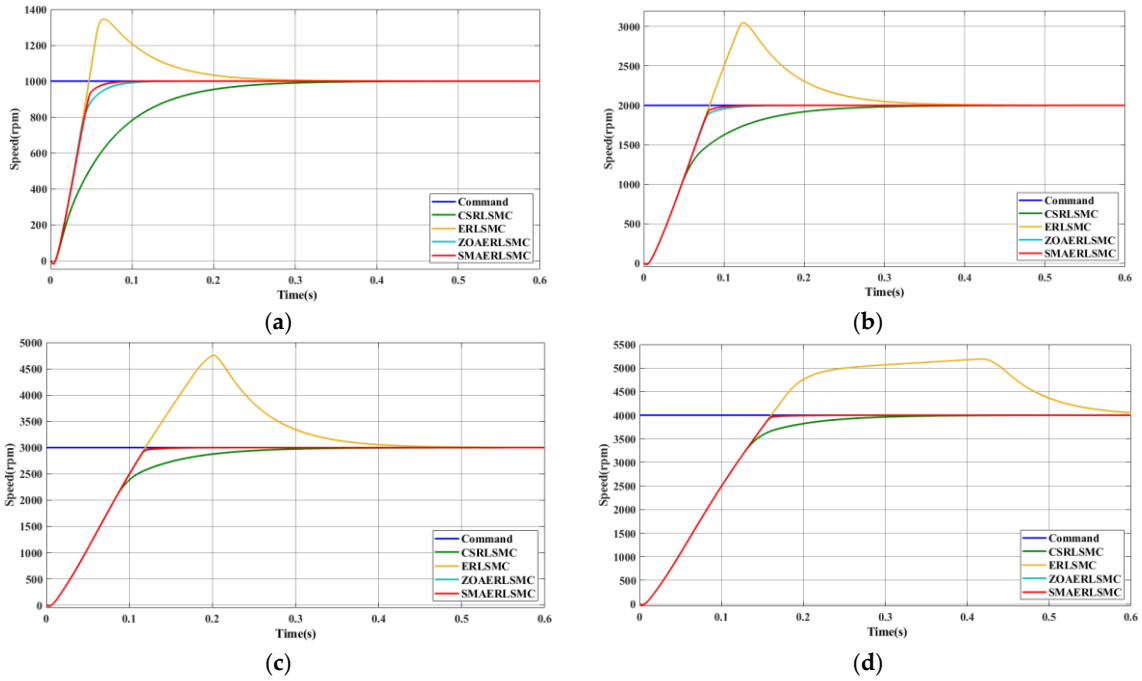

Figure 5. Comparison of tracking responses between the proposed SMAERLSMC, the CSRLSMC, the ERLSMC, and the ZOAERLSMC under a load of 2 N-m, where the speed command rises from 0 rpm to different speed commands: (**a**) $\omega_{rm}: 0 \rightarrow 1000$ rpm; (**b**) $\omega_{rm}: 0 \rightarrow 2000$ rpm; (**c**) $\omega_{rm}: 0 \rightarrow 3000$ rpm; (**d**) $\omega_{rm}: 0 \rightarrow 4000$ rpm.

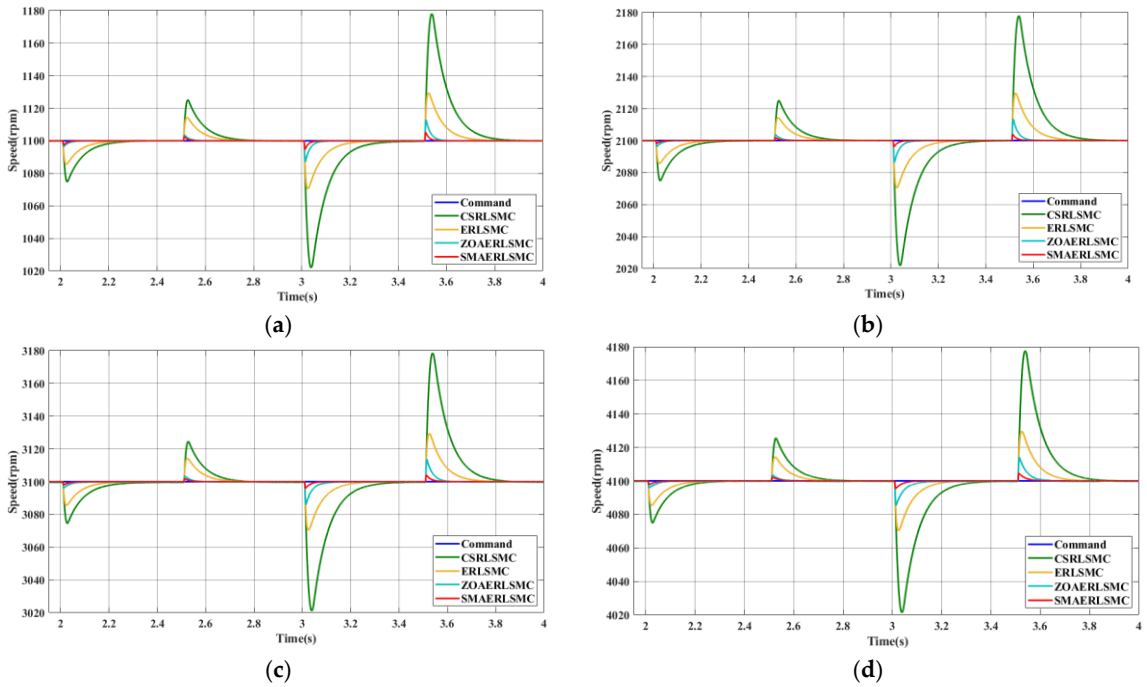

Figure 6. Comparison of load regulation responses between the proposed SMAERLSMC, the CSRLSMC, the ERLSMC, and the ZOAERLSMC, where a load of 1 N-m is applied at 2 seconds and unloaded at 2.5 seconds; then a load of 2 N-m is applied at 3 seconds and unloaded at 3.5 seconds: (**a**) $\omega_{rm} = 1100$ rpm, $T_L: 0 \rightarrow 1 \rightarrow 0 \rightarrow 2 \rightarrow 0$ N-m; (**b**) $\omega_{rm} = 2100$ rpm, $T_L: 0 \rightarrow 1 \rightarrow 0 \rightarrow 2 \rightarrow 0$ N-m; (**c**) $\omega_{rm} = 3100$ rpm, $T_L: 0 \rightarrow 1 \rightarrow 0 \rightarrow 2 \rightarrow 0$ N-m; (**d**) $\omega_{rm} = 4100$ rpm, $T_L: 0 \rightarrow 1 \rightarrow 0 \rightarrow 2 \rightarrow 0$ N-m.

 doi:10.20944/preprints202604.1268.v1

5. Conclusions

This paper proposes a SMAERLSMC to replace the P-I controller in traditional FOC, and applies it to IM speed control. Unlike the design of traditional fixed weights or general linear decreasing strategies, this study introduces the dynamic weight mechanism unique to the slime mould algorithm. This mechanism requires very low additional computational cost to utilize feedback on individual quality, realizing adaptive adjustment of the fitness function, thereby altering the search convergence path to precisely optimize the gains of the sliding mode controller. This control strategy not only effectively suppresses the speed overshoot caused by the ERLSMC but also overcomes the issue of poor load regulation response performance found in the CSRLSMC. Simultaneously, simulation results demonstrate that the robust controller proposed in this paper is superior to the ZOAERLSMC and two common reaching law sliding mode controllers in terms of speed tracking and load regulation performance. In particular, through adaptive weight calculation, it exhibits better robustness under load variations. Furthermore, the proposed controller features a simple architecture, low computational burden, and does not require large amounts of training data, making it easy to implement.

Author Contributions: K.-H.C. planned the project and wrote, edited, and reviewed the manuscript. K.-C.C. is responsible for the design of the robust controller combining slime mould optimization algorithm and sliding mode theory for the induction motor drive. K.-H.C. also managed the project and completed the formal analysis of the sliding mode controller. All authors have read and agreed to the published version of the manuscript.

Funding: The authors gratefully acknowledge the support and funding of this project by the National Science and Technology Council, Taiwan, under the Grant Number NSTC 114-2221-E-167-003-MY2.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Nomenclature

Acronyms

IM	:	induction motor
FOC	:	field-oriented control
SMA	:	slime mould algorithm
SMT	:	sliding mode theory
ZOA	:	zebra optimization algorithm
P-I	:	proportional-integral
SMC	:	sliding mode controller
ETERLSMC	:	extension theory combined with sliding mode controller
NNSMC	:	neural network sliding mode controller
2DOFC	:	two-degree-of-freedom controller
PSO	:	particle swarm optimization
SVPWM	:	space vector pulse width modulation
VSC	:	variable structure control
SMAERLSMC	:	slime mould algorithm combined with exponential reaching law sliding mode controller
CSRLSMC	:	constant speed reaching law sliding mode controller
ERLSMC	:	exponential reaching law sliding mode controller
ZOAERLSMC	:	zebra optimization algorithm combined with exponential reaching law sliding mode controller

Symbols

R_s	:	stator resistance
R_r	:	rotor resistance

 doi:10.20944/preprints202604.1268.v1

L_s	:	stator inductance
L_r	:	rotor inductance
L_m	:	mutual inductance
ω_e	:	synchronous angular velocity
ω_r	:	rotor electrical angular velocity
i_{ds}, i_{qs}	:	d and q axis stator currents
ϕ_{dr}, ϕ_{qr}	:	rotor d and q axis fluxes
L_σ	:	leakage inductance coefficient
T_L	:	load torque
J	:	moment of inertia
B	:	viscous friction coefficient
ω_{rm}	:	mechanical angular velocity
P	:	number of poles
σ	:	leakage coefficient
ω_{sl}	:	slip speed
$\hat{\phi}_r$	:	estimated flux
s	:	Laplace operator
$\overrightarrow{X(t+1)}$	:	new position for the approach food behavior
$\overrightarrow{X_b(t)}$	:	optimal position of the slime mould individual at the t-th iteration
$\overrightarrow{vc}$	:	control vector that linearly decreases from one to zero
$\overrightarrow{X_A(t)}, \overrightarrow{X_B(t)}$	:	positions of two individuals randomly selected from the slime mould
$\overrightarrow{X(t)}$	:	current position
t	:	current iteration count
p	:	switching probability
r	:	random number in the range $[0,1]$
$S(i)$	:	fitness value of the i-th slime mould
DF	:	best fitness value found in all iterations
$\overrightarrow{vb}$	:	oscillation amplitude parameter
T	:	maximum number of iterations
$\overrightarrow{W}$	:	weight
bF	:	best fitness value in the current iteration
wF	:	worst fitness value in the current iteration
$SmellIndex$	:	sequence of fitness values of the slime moulds
$condition$	:	slime mould individuals ranked in the first half of fitness
UB	:	upper bound of the search space
LB	:	lower bound of the search space
$rand$	:	random number in $[0,1]$
z	:	probability threshold
ω_{rm}^*	:	commanded rotor speed
s_1	:	sliding mode dynamic trajectory function
c	:	sliding mode dynamic trajectory control gain
x_1	:	state variable of speed difference
x_2	:	state variable of rate change of the speed difference
u	:	control force function
V	:	Lyapunov's second stability criterion continuous function
i_{qs}^*	:	stator reference current of the q-axis
$\dot{s}_1$	:	reaching law functions
ε	:	constant speed reaching gain
q	:	exponential reaching gain
$\mathrm{sgn}(s)$	:	function symbol of sign
e	:	speed difference
$\dot{e}$	:	rate change of the speed difference
$\alpha(W_1), \beta(W_2)$	:	weight values

 doi:10.20944/preprints202604.1268.v1

References

1. Simões, M.G. A Concise History of Induction Motor Drives—Part 1 [History]. *IEEE Electrif. Mag.* **2023**, 11, 5–11.
2. Paul, O.E.; Kucher, E.S. Synthesis of Induction Motor Adaptive Field-oriented Control System with the Method of Signal-adaptive Inverse Model. In Proceedings of the 21st International Conference of Young Specialists on Micro/Nanotechnologies and Electron Devices (EDM), Chemal, Russia, 29 June–03 July 2020; pp. 470–474.
3. Chao, K.H.; Hsieh, C.T.; Chen, X.J. A Robust Controller Based on Extension Sliding Mode Theory for Brushless DC Motor Drives. *Electronics* **2024**, 13, 4028, 1–26.
4. Gudey, S.K.; Malla, M.; Jasthi, K.; Gampa, S.R. Direct Torque Control of an Induction Motor Using Fractional-order Sliding Mode Control Technique for Quick Response and Reduced Torque Ripple. *World Electr. Veh. J.* **2023**, 14, 137, 1–20.
5. Chao, K.H.; Chang, C.L. Design of a Robust Controller for Induction Motor Drive Systems Based on Extendable Fuzzy Theory. *Mathematics* **2024**, 12, 3235, 1–20.
6. Song, J.; Zheng, W.X.; Niu, Y. Self-Triggered Sliding Mode Control for Networked PMSM Speed Regulation System: A PSO-Optimized Super-Twisting Algorithm. IEEE Trans. Ind. Electron. **2022**, 69, 763–773.
7. Chao, K.H.; Chen, C.C.; Lin, X.Y. Design of a Brushless DC Motor Drive System Controller Integrating the Zebra Optimization Algorithm and Sliding Mode Theory. Electronics **2024**, 13, 3353, 1–24.
8. Li, S.; Chen, H.; Wang, M.; Heidari, A.A.; Mirjalili, S. Slime Mould Algorithm: A New Method for Stochastic Optimization. *Future Gener. Comput. Syst.* **2020**, 111, 300–323.
9. O'Rourke, C.J.; Qasim, M.M.; Overlin, M.R.; Kirtley, J.L. A Geometric Interpretation of Reference Frames and Transformations: dq0, Clarke, and Park. *IEEE Trans. Energy Convers.* **2019**, 34, 2070–2083.
10. Liu, C.H. *AC Motor Control: The Principles of Vector Control and Direct Torque Control*, 4th ed.; Tonghua Books Co., Ltd.: Taipei, Taiwan, 2007; pp. 33–217.
11. Ye, C.C. AC Motor Control and Simulation Technology: A Guidance on Mastering the Core Algorithms of Electric Vehicles and Variable Frequency Technology, 1st ed.; Wunan Books: Taipei, Taiwan, 2023; pp. 11–151.
12. Gupta, S.; George, S.; Awate, V. PI Controller Design and Application for SVPWM Switching Technique Based FOC of PMSM. In Proceedings of the 2nd International Conference on Trends in Electrical, Electronics, and Computer Engineering (TEECCON), Bangalore, India, 23–24 August 2023; pp. 172–177.
13. Vemparala, R.R.B.; Titus, J. Performance Evaluation of an Si+SiC Based Hybrid VSI Using a Modified Space Vector Switching Pattern in a Grid Connected Inverter Application. In Proceedings of the IECON–48th Annual Conference of the IEEE Industrial Electronics Society, Brussels, Belgium, 17–20 October 2022; pp. 1–6.
14. Tapia-Olvera, R.; Beltran-Carbajal, F.; Aguilar-Mejia, O.; Valderrabano-Gonzalez, A. An Adaptive Speed Control Approach for DC Shunt Motors. *Energies* **2016**, 9, 961, 1–16.
15. Lin, S.; Cao, Y.; Wang, Z.; Yan, Y.; Shi, T.; Xia, C. Speed Controller Design for Electric Drives Based on Decoupling Two-degree-of-freedom Control Structure. *IEEE Trans. Power Electron.* **2023**, 38, 15996–16009.
16. Georgiev, Z.; Trushev, I.; Chervenkov, A. Laplace Transform Method to Transient Analysis in Magnetically Coupled Electrical Circuits. In Proceedings of the International Scientific Conference on Computer Science (COMSCI), Sozopol, Bulgaria, 18–20 September 2023; pp. 1–4.
17. Jo, G.J.; Choi, J.W. Rotor Flux Estimator Design with Offset Extractor for Sensorless-driven Induction Motors. *IEEE Trans. Power Electron.* **2022**, 37, 4497–4510.
18. Indriawati, K.; Widjiantoro, B.L.; Rachman, N.R. Disturbance Observer-based Speed Estimator for Controlling Speed Sensorless Induction Motor. In Proceedings of the 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), Yogyakarta, Indonesia, 10–11 December 2020; pp. 301–305.
19. Li, K.; Ding, J.; Sun, X.; Tian, X. Overview of Sliding Mode Control Technology for Permanent Magnet Synchronous Motor System. *IEEE Access* **2024**, 12, 71685–71704.

 doi:10.20944/preprints202604.1268.v1

20. Qin, G.; Wang, M.; Cao, G.; Wang, Q.; Liao, Y. PID Sliding Mode Control of PMSM Based on Improved Terminal Sliding Mode Reaching Law. *Energies* **2025**, 18, 2661, 1–15.
21. Bodur, F.; Kaplan, O. A Novel Sliding Mode Control Based on Super Twisting Reaching Law for PMSM Speed Controller with Fixed-time Disturbance Observer. In Proceedings of the 12th International Conference on Renewable Energy Research and Applications (ICRERA), Oshawa, ON, Canada, 29 August–01 September 2023; pp. 563–568.