scientific reports

OPEN

Research on online optimization scheme and deployment of PMSM control parameters based on honey badger algorithm

Xiaofeng Zhu[1], Yiming Hu[1]✉, Yinquan Yu[1], Dequan Zeng[1], Jinwen Yang[1] & Giuseppe Carbone[2]

Permanent magnet synchronous motor (PMSM) drive systems are receiving increasing attention due to their superior control quality and efficiency. Optimizing the control parameters are key to achieve the high-performance operation of PMSMs. Although heuristic algorithms demonstrate excellent optimization outcomes in simulations, there are still challenges in deploying optimization schemes in practical drives. In this study, a real-time online deployable control parameter optimization scheme is proposed. The optimization effect is evaluated through the system step response performance, and a framework for deploying optimization algorithms within the driver is developed. A fault suppression mechanism is also designed to mitigate overshoot and vibration issues caused by suboptimal solutions. The proposed scheme is validated on a rapid prototyping control platform. Experimental results confirm that the scheme exhibits good optimization performances across various operating conditions. The honey badger algorithm employed in this paper shows faster convergence and more stable optimization effects than other optimization algorithms. The optimization effect is improved by 2.2% and its performance in terms of consistency across multiple optimization results has increased by 40%.

Keywords PMSM, Honey badger algorithm, Metaheuristic algorithms, Online optimization, Control parameter optimization, Control failure suppression

The permanent magnet synchronous motor (PMSM) is an important type of motor in modern motor applications[1]. Due to its excellent energy conversion efficiency, high power density, high torque, and wide speed range, it has been widely used in electric power systems, such as in cooperative robots[2], electric vehicles[3], wind power generation, and ship propulsion[4,5], and it has received increasing attention.

The application of high-performance controllers is the key to achieve efficient operation of PMSMs[6]. To improve the control performance of PMSMs and enable them to undergo fast and accurate dynamic responses to load changes and external disturbances, various advanced control strategies have been proposed, including fuzzy control[7], sliding mode control[8,9], model predictive control (MPC)[10,11], and adaptive control[12]. Adaptive fuzzy logic speed controllers[13] have been developed to improve the DC bus voltage utilization. The combination of a fuzzy method and memory integral sliding mode control reduces the effect of mismatch disturbances on PMSMs[14], and the introduction of an adaptive law can further improve the robustness of the drive system[15]. An adaptive fuzzy control scheme with a command filtering error compensation mechanism proposed[16] solved the position tracking problem. Model-free predictive control can enhance the stability of motor operation[17,18], and the effects of parameter and external perturbations can be further eliminated by combining it with sliding mode control[19].The above model demonstrates good control performance; however, it has the drawback of high model complexity and application costs. Therefore, in industrial environments, proportional integral (PI) control-based direct torque control (DTC)[20] and field oriented control (FOC)[21] are widely adopted[22,23]. Despite the PMSM having high nonlinearity, the relationship between the output and input of the control object signals is linearized near the operating point[24]. The efficiency of the controller can be ensured by selecting suitable PI parameters that match the specific operating conditions.

However, regardless of which control scheme is used, there are hyperparameters in the control model that need to be adjusted. In addition, the PMSM has high nonlinearity, and when operating conditions change, the

[1]School of Mechatronics and Vehicle Engineering, East China Jiaotong University, Nanchang 330000, China. [2]Department of Mechanical, Energy, and Management Engineering, University of Calabria, 87036 Rende, Italy. ✉email: huyiming139@hotmail.com

control parameters must be adjusted to prevent a decline in the control performance. Conventionally, controller parameters were conventionally designed using the trial-and-error method. However, this approach was considered undesirable due to its blind, lengthy, non-optimal, and potentially unsafe nature. Therefore, to meet different operation requirements and working environment, such as the varying control performance demands of reconfigurable production systems[25], optimizing the hyperparameters in the drives becomes crucial[26]. Such optimization problems are widespread in areas such as materials physics[27], disease prediction[28,29] and power system optimization[30]. To efficiently address these optimization issues, heuristic optimization algorithms[31] and machine learning technologies[32] are widely adopted.

Metaheuristic algorithms have garnered significant attention due to their remarkable search capabilities and excellent global optimization performance, among which the genetic algorithm (GA)[33], particle swarm optimization algorithm (PSO)[34], and differential evolutionary algorithm (DEA)[35] have been used for controller parameter tuning during the design stage of the motor control system. In recent studies, a control algorithm combining PSO and PI control was successfully applied to tune controller parameters of PMSM, and its feasibility was verified by simulations[36,37]. This hybrid method not only improves the performance of a variable-speed drive but also enhances the stability and response speed of the system. By comparing the performances of multiple metaheuristic algorithms in control parameter searches, it was shown that each metaheuristic algorithm is effective in finding the optimal control parameters[38]. In addition to being combined with PI control algorithms, metaheuristic algorithms were used[39] to optimize the output affiliation function of the fuzzy logic controller to achieve the best control performance of the motor. Model downscaling and controller design of the PMSM controller with the help of firefly-based hybrid metaheuristic algorithm was performed[40]. A combination of the GA and an artificial neural network (ANN) was used[41] to adjust the weight factor of the MPC cost function to improve the robustness of motor parameters such as the stator resistance and the permanent magnet flux. Moreover[42], used a PSO algorithm combined with fuzzy PID to optimize the control performance. The algorithm was deployed in experiments, eliminating errors caused by differences between the simulation and experimental environments. An artificial bee colony (ABC) algorithm[43] was employed for parameter optimization, alleviating issues of instability due to poor solutions in real-world deployment environments.

Despite the performances achieved by existing methods in optimizing control parameters for PMSMs, several unresolved issues remain in practical applications. The primary challenge lies in designing the operational logic of search algorithms embedded in the driver to accurately evaluate the performance with various control parameters under different operating conditions. Furthermore, larger population sizes and iteration counts significantly increase the search times, so improving the convergence speeds of algorithms within limited timeframes is crucial for enhancing the search efficiency. Finally, during the optimization process, it is necessary to assess the control effects of suboptimal solutions, which may lead to declines in the control performance or even system instability. Therefore, constructing effective fault suppression mechanisms is critical.

To address the challenges faced in deploying optimization algorithms in the driver, this paper innovatively proposes a real-time online deployable control parameter optimization scheme (ODPOS). The core advantage of this scheme lies in its adaptability to different operating conditions and its excellent convergence capability. It can be deployed online throughout the entire lifecycle of the equipment, significantly enhancing the system's control performance under varying operating conditions. Specifically, the ODPOS defines the fitness of individuals by evaluating the step response performance of the system under different operating conditions, ensuring superior performance throughout the optimization process. Additionally, the honey badger algorithm (HBA)[44] is employed to effectively enhance the convergence speed and quality of the optimization results. Furthermore, a fault suppression mechanism (FSM) is designed to identify and suppress motor out-of-control behaviors. The proposed ODPOS has been experimentally tested on a rapid prototyping experimental platform and compared with other optimization algorithms. Experimental results demonstrated that the proposed ODPOS exhibits outstanding search capabilities in practical applications, providing new insights and solutions for optimizing PMSM control parameters.

The remainder of the paper is structured as follows. "Modeling of PMSM drive systems" describes the PMSM model and control method based on the field-oriented control approach. "ODPOS for PMSM" proposes a real-time ODPOS based on the HBA. Section "Experiments of real-time online optimization" deploys the parameter tuning scheme using a rapid prototyping control platform and discusses the experimental results. Finally, "Conclusions" summarizes the whole paper and provides conclusions. The symbols used in this article can be found in Supplementary Table S1 online.

Modeling of PMSM drive systems

In this study, a surface permanent magnet synchronous motor (SPMSM) was used. Because a PI controller can clearly show control performance differences when various control parameters are selected to facilitate the evaluation of the optimization algorithm, the traditional three-phase PMSM vector control scheme was adopted in this study, which included the speed loop PI controller and the current loop PI controller.

PMSM system model description

For the SPMSM, the $i_d = 0$ control is equivalent to maximum torque per ampere control. To facilitate the design of the controller, a mathematical model of the PMSM in the synchronous coordinate system was chosen. The stator voltage equations of the PMSM are expressed as follows:

$$\begin{cases} u_d = Ri_d + \dfrac{d}{dt}\psi_{\mathrm{d}} - \omega_e \psi_q \\ u_q = Ri_q + \dfrac{d}{dt}\psi_q + \omega_e \psi_d \end{cases}, \tag{1}$$

where *R* is the resistance of the three-phase winding, ω_e is the electrical angular velocity of the motor, u_d and u_q are the voltages of the *d* and *q* axes in the synchronous rotating coordinate system, respectively, and i_d and i_q are the currents of the *d* and *q* axes in the synchronous rotating coordinate system, respectively.

The stator flux linkages ψ_d and ψ_q can be obtained from the $d-q$ axis inductance, calculated as follows:

$$\begin{cases} \psi_d = L_d i_d + \psi_f \\ \psi_q = L_q i_q \end{cases}, \tag{2}$$

where ψ_f represents the permanent magnet flux linkage, and where L_d and L_q are the inductances of the *d* and *q* axes, respectively.

By combining Eqs. (1) and (2), the stator voltage equations can be expressed as

$$\begin{cases} u_d = Ri_d + L_d \dfrac{d}{dt} i_d - \omega_e L_q i_q \\ \\ u_q = Ri_q + L_q \dfrac{d}{dt} i_q - \omega_e (L_d i_d + \psi_f) \end{cases}. \tag{3}$$

In addition to this, the electromagnetic torque T_e of the motor in the natural coordinate system is equal to the derivative of the magnetic field storage energy with respect to the displacement of the mechanical angle θ_m. Thus, the electromagnetic torque can be described as

$$T_\mathrm{e} = \frac{1}{2} p_\mathrm{n} \frac{\partial}{\partial \theta_\mathrm{m}} \left(\boldsymbol{i}_\mathrm{3s}^\mathrm{T} \cdot \boldsymbol{\psi}_\mathrm{3s} \right), \tag{4}$$

where $\boldsymbol{i}_{3s} = \begin{bmatrix} i_A \\ i_B \\ i_\mathrm{C} \end{bmatrix}$ and $\boldsymbol{\psi}_{3s} = \begin{bmatrix} \psi_A \\ \psi_B \\ \psi_C \end{bmatrix}$ are the phase currents and flux linkages of the three-phase windings, respectively, and p_n is the number of pole pairs of the PMSM. Transforming equation into a rotating synchronous coordinate system yields

$$T_\mathrm{e} = \frac{3}{2} p_n i_q \left[i_d (L_d - L_q) + \psi_\mathrm{f} \right]. \tag{5}$$

In addition, the mechanical equations of motion of the motor are described as

$$J \frac{d\omega_m}{dt} = T_e - T_L - B\omega_m, \tag{6}$$

where *J* is the moment of inertia, *B* is the damping factor, and T_L is the load torque.

Control parameter calculation of speed loop

The $i_d = 0$ control scheme was used, and the concept of "active damping" was introduced to set the parameters of the speed loop PI controller, assuming that the motor was started without a load. The following results can be obtained:

$$\frac{\mathrm{d}\omega_m}{dt} = \frac{1.5 p_n \psi_f}{J} \left({i'}_q - B_a \omega_m \right) - B\omega_m / J. \tag{7}$$

Configuring Eq. (7) to the desired closed-loop bandwidth β, the *q*-axis current transfer function corresponding to the rotational speed can be obtained as

$$\omega_m(s) = \frac{1.5 p_n \psi_f / J}{s + \beta} i'_\mathrm{q}(s). \tag{8}$$

Comparing Eqs. (7) and (8) yields the active damping factor B_a:

$$B_a = \frac{\beta J - B}{1.5 p_n \psi_f}. \tag{9}$$

Using a conventional PI regulator, the speed loop controller is expressed as

$$i_q^* = \left(K_{p\omega} + \frac{K_{i\omega}}{s} \right) (\omega_m^* - \omega_m) - B_a \omega_m, \tag{10}$$

where the parameter i_q^* denotes the desired *q*-axis current, and ω_m^* denotes the desired mechanical angular velocity. $K_{p\omega}$ and $K_{i\omega}$ of the PI controller can be adjusted by the following equations:

$$\begin{cases} K_{p\omega} = \dfrac{\beta J}{1.5 p_n \psi_f} \\ K_{i\omega} = \beta K_{p\omega} \end{cases}, \tag{11}$$

where β is the desired bandwidth of the speed loop.

This method of parameter calculation is simple and the relationship between the control parameters and the dynamic quality of the system is clear.

Control parameter calculation of current loop

Using a conventional PI controller and combining it with a feed-forward decoupling control scheme, the voltage in the *d-q* axis coordinates is given as

$$\begin{cases} u_d^* = \left(CK_{pd} + \dfrac{CK_{id}}{s}\right)(i_d^* - i_d) - \omega_e L_q i_q \\ u_q^* = \left(CK_{pq} + \dfrac{CK_{iq}}{s}\right)(i_q^* - i_q) + \omega_e (L_d i_d + \psi_f) \end{cases}, \tag{12}$$

where CK_{pd} and CK_{pq} are the proportional gains of the PI controller, and CK_{id} and CK_{iq} are the integral gains of the PI controller. These parameters can be calculated by the following equation:

$$\begin{cases} CK_{pd} = \gamma L_d \\ CK_{id} = \gamma R \\ CK_{pq} = \gamma L_q \\ CK_{iq} = \gamma R \end{cases}. \tag{13}$$

The response time t_{res} is defined as the time required for the system response to go from 10% to 90% of the step signal, and the relationship between γ and t_{res} can be approximated as $t_{res} = ln9/\gamma$. It can be seen that decreasing γ will lengthen the response time and increasing γ will speed up the response of the system. Since the system response time t_{res} is limited by the electrical time constant, γ cannot be expanded infinitely in practice. The electrical time constant for the *d-q* axes can be calculated by the following equation:

$$\begin{cases} T_d = L_d/R \\ T_q = L_q/R \end{cases}. \tag{14}$$

ODPOS for PMSM

The ODPOS built in this paper has capability to search for the optimal control parameters online. The optimal control target is formulated according to the specific demand. The proposed ODPOS contains a reference speed module and an Optimization Algorithm module to realize the deployment of the HBA in the controller. The Optimization Algorithm module utilizes the HBA as the optimization algorithm to reduce the optimization time. Furthermore, the FSM is constructed to suppress the motor control failure caused by the poor solution during the optimization process.

Principle of HBA

The HBA is a metaheuristic algorithm. The design of the algorithm is inspired by the honey badger's hunting behavior in nature. The algorithm has two parts: an exploration phase and an exploitation phase. The exploration phase mimics the sniffing and digging behavior of the honey badger when locating prey, by extensively searching the solution space to discover new regions that may contain better solutions. The exploitation phase is modeled after the honey badger's foraging behavior of following a honeyguide bird to locate a hive, and this phase further improves and optimizes the known optimal solution to find a better solution.

The HBA avoids falling into local optimal solutions and tries its best to find global optimal solutions by balancing the exploration phase and the exploitation phase. The algorithm can effectively deal with complex nonlinear problems, and experimental results on several standard benchmark functions and engineering design problems have proven its strong global search capability and fast convergence speed. The operation process of the algorithm is mainly based on the following key steps:

(1) Overall initialization. The initialization parameters include the number of iterations $iter_{max}$ and the population size *n*. The candidate solutions to the problem are represented by an $n \times D$ dimensional matrix $\mathbf{P}$:

$$\mathbf{P} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1D} \\ x_{21} & x_{22} & \cdots & x_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nD} \end{bmatrix}, \tag{15}$$

where each row corresponds to an individual and the columns represent the values of the coordinates in the parameter space; *D* denotes the dimension of an individual.

The equation to calculate the location of each individual is as follows:

$$x_i = lb_i + r_1 \times (ub_i - lb_i)\,, \tag{16}$$

where r_1 is a random number between 0 and 1, and ub_i and lb_i denote the upper and lower bounds of the search domain, respectively.

(2) Evaluate fitness of initial individuals. The fitness f_i of each honey badger is evaluated based on the objective function. The individual with the best fitness in the matrix $\mathbf{P}$ is found. The location of this honey badger is defined as x_{prey}, and the fitness is defined as f_{prey} for the algorithm to update the population in the next iteration.

(3) Define intensity I_i and density factor α. I_i is related to the intensity of the concentration of honey badgers in the search space and the distance between the honey badger and the prey. If I_i is high, the honey badger will move better and faster. I_i is defined as follows:

$$I_i = r_2 \frac{S}{4\pi d_i^2}, \tag{17}$$

where $S = (x_i - x_{i+1})^2$ denotes the density of the distribution of honey badgers, $d_i = x_{prey} - x_i$ denotes the distance between the prey and the *i*th honey badger, and r_2 is a random number between 0 and 1. A smaller value of S indicates that the *i*-th honey badger is close to other honey badgers, leading the algorithm to reduce I_i to minimize unnecessary movement and prevent conflicts with other honey badgers. Conversely, a smaller d_i indicates that the honey badger can more clearly perceive the prey's location, thus requiring an increase in I_i to accelerate its approach towards the prey.

Equation (18) is used to update the density factor α, which decreases with the number of iterations, thus ensuring a smooth transition from the exploration phase to the exploitation phase.

$$\alpha = C_1 \times \exp\left(-iter/iter_{max}\right), \tag{18}$$

where C_1 is a constant greater than 1, and $iter_{max}$ is the maximum number of iterations.

(4) Jumping out of local optimum. To prevent the algorithm from falling into a local optimum, a search direction flag *F* is defined, which is determined by a random number through Eq. (19). The algorithm uses this flag to provide a higher chance of searching the solution space for strict scanning. *F* is defined as follows:

$$F = \begin{cases} 1 & \text{if } r_3 \leq 0.5 \\ -1 & \text{else} \end{cases}, \tag{19}$$

where r_3 denotes a random number between 0 and 1.

(5) Honey badger position update. The honey badger algorithm consists of two parts: the "exploration phase" and the "exploitation phase." In the "exploration phase," the updated honey badger position is calculated as follows:

$$x_{new} = x_{prey} + FC_2 I x_{prey} + F r_4 \alpha d_i \left|\cos(2\pi r_5)\left[1 - \cos(2\pi r_6)\right]\right|, \tag{20}$$

In the "exploitation phase", the updated honey badger position is given by

$$x_{new} = x_{prey} + F r_7 \alpha d_i, \tag{21}$$

where r_7 is a random number between 0 and 1.

In the HBA, the determination of whether to use the exploration or exploitation phase of the algorithm to update the location of a honey badger individual is determined by a random number. Specifically, when the random number r_3 is less than or equal to 0.5, the algorithm selects the location update formula for the exploration phase, while when the random number r_3 is greater than 0.5, the algorithm selects the location update formula for the exploitation phase.

(6) Evaluate fitness of updated individuals. The fitness of the new individual is evaluated through the given objective function, and the fitness values of the old and new individuals are compared. If the new individual achieves better fitness, the old honey badger is replaced by the new one; otherwise, the old honey badger position and fitness are retained. After all honey badger positions are updated, the best fitness values of all individuals are found and compared with x_{prey}. The results of the comparison are used to determine whether to update x_{prey} and f_{prey}.

If the algorithm does not reach the end condition after executing step (6), it returns to step (3) and repeats the above steps until the optimum is found. The pseudo-code for HBA is given in Algorithm 1.

```
Input: $iter_{max}, C_1, C_2, n$
  Execute step (1) to initialize population $\mathbf{P}$
  Execute step (2) to evaluate the fitness of each $x_i$
  Save the best position $x_{prey}$ and assign fitness to $f_{prey}$
  for $iter$; $iter$++; $iter < iter_{max}$ do
      Execute step (3) to update $\alpha$ and $I_i$
      for $i$; $i++$; $i < n$ do
          Execute step (4) to calculate $F$
          Execute step (5) to update the $x_{new}$
          Execute step (6) to evaluate $x_{new}$ and assign to $f_{new}$
          Update $\mathbf{P}$, $x_{prey}$, $f_{prey}$
      end for
  end for
Output: $x_{prey}$
```

Algorithm 1.

Online operational logic of the HBA deployed in the driver

In the simulation environment, the single-step solution of the control algorithm requires a short time and can be arbitrarily embedded in the metaheuristic algorithm. However, in practical applications, the evaluation of the fitness of a single individual requires the control system to go through the response phase and the stabilization phase completely, and after completing one evaluation, it is necessary to wait for the system to stabilize and then carry out the next fitness calculation, which results in a long solution time and makes it difficult for the simulation scheme to transfer directly to practical applications. For this reason, the HBA was combined with the FSM proposed in the "Design of control FSM" section as an Optimization Algorithm module embedded in the driver. A reference speed module was also designed to assist the operation of the HBA.

The block diagram of the overall structure of ODPOS is shown in Fig. 1. The reference speed module generates a step speed signal as the reference speed. The control effort of the speed controller is derived from the difference between the reference speed and the actual speed measured by the encoder on the PMSM. The speed controller is constructed using PI control, with its control parameters provided by the Optimization Algorithm module. The q-axis current control effort i_q^* produced by the speed PI controller is not directly used for the motor but rather serves as the current reference for the internal current PI controller, whose task is to generate the q-axis voltage u_q^*. The current signals required by the current PI controller are obtained from the current sampling circuit and transformed into the synchronous rotating coordinate system using the Clark and Park transformations to obtain i_d and i_q. Due to the use of the $i_d = 0$ control strategy, the d-axis current reference is set to zero. The d-axis voltage u_d^* is also generated through the same current PI controller. The reference voltages u_q^* and u_d^* are then used to generate the switching sequence controlling the three-phase voltage inverter through inverse Park transformation and space vector pulse width modulation (SVPWM). Ultimately, the DC power source generates three-phase AC power through the inverter to control the motor and thus drive the motor to operate stably.

The operational logic of the Reference Speed module and the Optimization Algorithm module is illustrated in Fig. 2. The horizontal axis represents the algorithm run time, and the positive part of the vertical axis shows the reference speed output by the Reference Speed module. Below the coordinate axes are the HBA algorithm

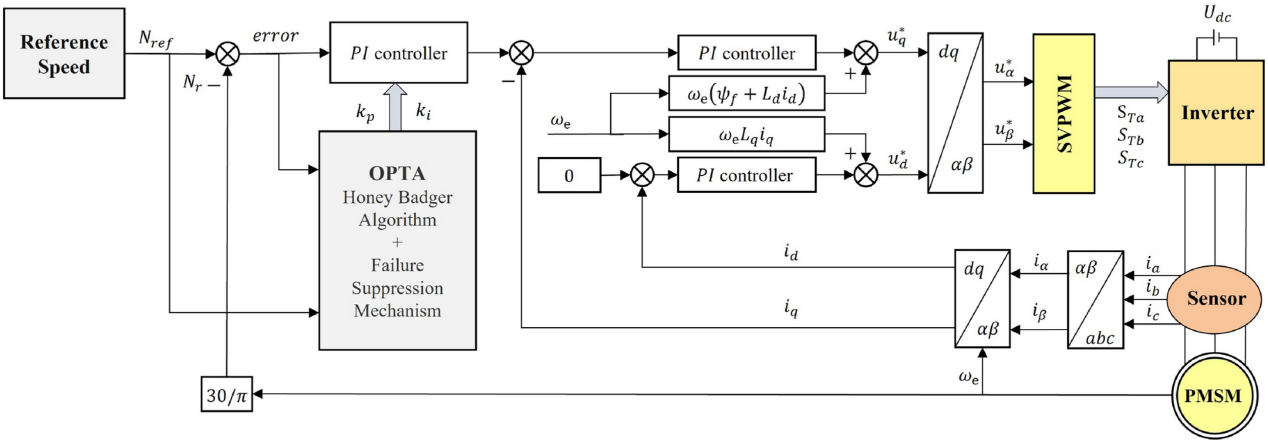


Fig. 1. Control block diagram after embedding the algorithm into the control program.

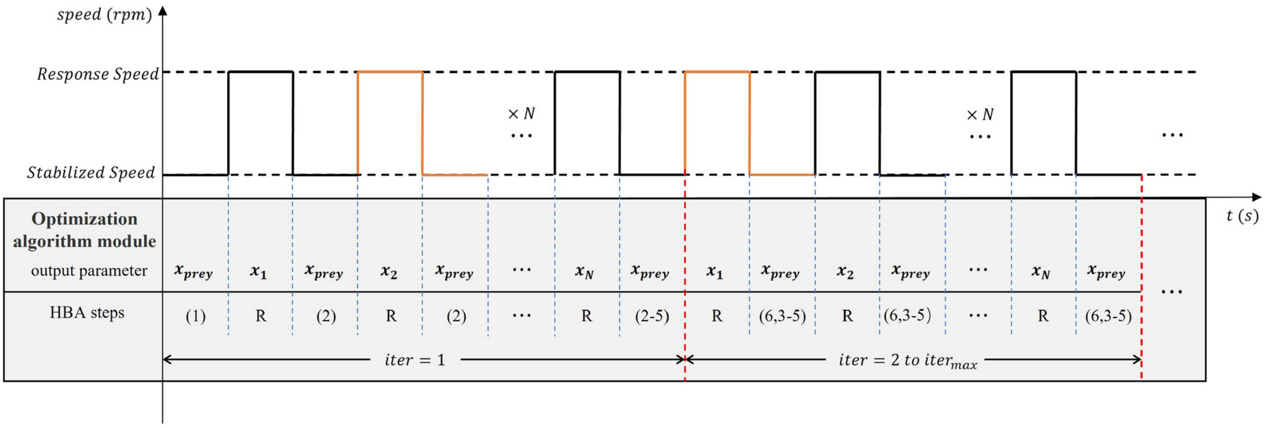

speed (rpm)
Response Speed
Stabilized Speed
× N
⋯
× N
⋯
⋯
t (s)
Optimization algorithm module
output parameter
x_{prey} x_1 x_{prey} x_2 x_{prey} ⋯ x_N x_{prey} x_1 x_{prey} x_2 x_{prey} ⋯ x_N x_{prey}
HBA steps
(1) R (2) R (2) ⋯ R (2-5) R (6,3-5) R (6,3-5) ⋯ R (6,3-5)
⋯
iter = 1
iter = 2 to $iter_{max}$

Fig. 2. Signals output from reference speed module and the corresponding HBA steps performed by the Optimization Algorithm module. *R* indicates that the task of the Optimization Algorithm module is to record the speed error. (2) denotes the execution of step (2) in the HBA to evaluate the fitness of the initial individual. (2–5) means that steps (2) through (5) are executed to update the individual. (6,3–5) indicates that step (6) in the HBA is executed first to evaluate the fitness of the previous individual, and then steps (3) through (5) are executed to update the next individual.

steps executed by the Optimization Algorithm module and the control parameters it outputs during each corresponding time period. The signal output by the Reference Speed module is a square wave, with a maximum value matching the target operating speed and the minimum value representing the stabilized speed, both lasting for 1 s.

This reference speed signal is also sent to the Optimization Algorithm module at the same time. When the signal indicates a response speed, the module directly outputs the control parameters that require fitness evaluation for the PI controller. Under these parameters, the system completes a step response, and the response results are recorded by the Optimization Algorithm module.When the signal indicates a stabilized speed, the Optimization Algorithm module outputs the best control parameters found so far to stabilize the system. Simultaneously, the module calculates the fitness of the previous parameters based on the response results recorded from the previous phase and executes steps 3 to 5 of the HBA to update individuals. An optimization cycle for an individual consists of a response phase and a stabilization phase, completing one evaluation of the individual fitness and one update of the individuals.

The HBA pseudo-code embedded in the control program is given in Algorithm 2. To adapt to the requirements of deploying the algorithm in the drive, this paper breaks down the algorithm into steps as described in “Principle of HBA” section and different parts are excuted in different stages of the optimization process. The frequency of for loop execution in the algorithm is consistent with the sampling frequency of the control system. *t* denotes the current execution time of the algorithm. t_{max} denotes the maximum execution time of the algorithm. The initial value x_1 is calculated according to the mathematical model described in “PMSM system model description” section. This parameter is used to stabilize the system when no better parameter has been found.

```
Set parameters $iter_{max}$, $C$, $C_2$, $n$.
Initialize $P$ with random positions.
Calculate initial parameters $x_i$ using parameter tuning methods and set the optimal individual $x_{prey} = x_i$
Set $i = 1$, $iter = 1$
for $t$; $t = t + t_s$, $t <= t_{max}$ do                ▷ $t_s$ denotes the sampling period of the control system
    Receive signal $N_{ref}$ and $error$
    if $n_{ref}$ changes to stabilized rpm then
        if $iter = 1$ then
            Execute step (2) using Re                    ▷ The recorded errors are stored as an array Re
        else
            Execute step (6) using Re
        end if
        Update P, $x_{prey}$, $f_{prey}$
        $i = i + 1$
        if $i > n$ then
            $i = 1$, $iter = iter + 1$
        end if
        Execute step (3) to update $I_i$ and $\alpha$
        Execute step (4) to update $F$
        Execute step (5) to calculate $x_{new}$
        $x_{out} = x_{prey}$
    end if
    if $n_{ref}$ changes to response rpm then
        $x_{out} = x_{prey}$
        Record the speed error as Re
    end if
      return $x_{out}$
end for
```

Algorithm 2.

Design of control FSM

In the optimization process, to search the global optimal solution rather than a suboptimal solution within the solution space, the algorithm attempts to traverse all parameters in the solution space. When the control parameters fail to satisfy the stabilization conditions, the system may encounter vibration, overshoot, and even loss of control. In practical deployment environments, this failure has the potential to shorten the service life of the equipment and cause economic losses. Therefore, it is necessary to suppress any potential out-of-control behavior.

To avoid control failures in PMSM systems, an FSM is proposed, which includes overshoot suppression and vibration suppression modules.

(1) Overshoot suppression module: For cases where the speed cannot be controlled near the target speed due to poor control parameters, the upper limit H_L and lower limit L_L are introduced to detect whether the speed has exceeded the given threshold.

$$\begin{aligned} H_L &= response\ speed \times 2 \\ L_L &= stabilized\ speed \times 0.5 \end{aligned} . \tag{22}$$

When the speed feedback is higher than H_L or lower than L_L, the module switches the output control parameters to the optimal parameters that have been found so far to stabilize the system. The fitness of this individual is also set to infinity.

(2) Vibration suppression module: To address the problem of rotational speed vibrations near the target caused by poor control parameters, the moving average method is first used to calculate the average speed. This method eliminates short-term fluctuations in the sampled speed data and highlights the long-term trend.

First, the sampled speed data points are denoted as $s_1, s_2, s_3, \cdots, s_n$. Then, the average speed calculated by the moving average method is given by Eq. (23).

$$N_{avg} = \frac{1}{n} \sum_{i=t-n}^{t} s_i, \tag{23}$$

where N_{avg} denotes the average rotational speed, t denotes the tth rotational speed sampling point, and n denotes the size of the window for the moving average method, i.e., the number of data points used to calculate the average.

Second, the upper and lower limits of the rotational speed are set based on the average rotational speed, which is given by

$$\begin{aligned} UN_{avg} &= (1+\delta)N_{avg} \\ LN_{avg} &= (1-\delta)N_{avg} \end{aligned}, \tag{24}$$

where δ indicates that the amplitude of the speed vibration should not exceed a certain percentage of the average speed, UN_{avg} denotes the upper limit of the average speed, and LN_{avg} denotes the lower limit of the average speed.

Finally, a vibration suppression indicator V is introduced as

$$V = \begin{cases} 1 & \text{if } f_u \geq 3 \text{ and } f_l \geq 3 \\ 0 & \text{otherwise} \end{cases}, \tag{25}$$

where f_u denotes the number of times that the motor's speed exceeds the upper limit of the average speed in a given time range T_L, and f_l denotes the number of times that the motor's speed falls below the lower limit of the average speed in T_L.

When the vibration suppression indicator V is equal to 1, it indicates that the algorithm has detected the occurrence of vibration, in which case the output will be switched to x_{prey} to suppress the vibration problem due to unsuitable control parameters.

Optimization target design

In industrial applications, PMSMs are used in machine tools, pressurized plastic machines, fans, and other equipment. Different industrial applications require different control strategies to satisfy the production requirements. In control parameter optimization problems, different control performances are guided by different optimization objectives. And different ways of calculating fitness values in heuristic algorithms represent different optimization objectives. The selection of evaluation indices directly affects the control performance, and the commonly used evaluation indices are error integration indices. These indices mainly include ISE, ITSE, IAE, and ITAE, which are defined in Table 1. In the integral index calculation formulas defined in the table, *e* represents the error between the actual speed and the reference speed, and *t* represents the time.

The system optimization process determines a set of optimal control parameters through the above evaluation indices, and the control parameters obtained under different objectives need to be varied to achieve a trade off between the rise time, overshoot amount, and adjustment time. By analyzing the control effect of the optimal control parameters found by the system for different values of the evaluation indices, the optimization ability of the proposed scheme can be better evaluated.

Experiments of real-time online optimization

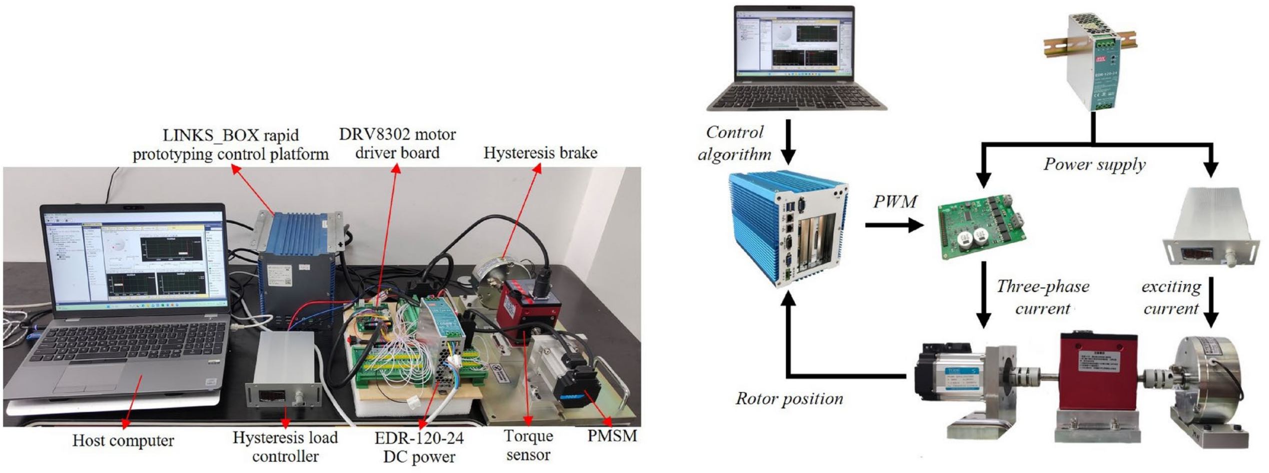
Experimental equipment

The proposed ODPOS was validated on an experimental platform built based on Links-RT semi-physical simulation equipment, which could be directly connected with MATLAB/Simulink, and the algorithm deployment was completed by creating fast control models. The whole experimental platform is shown in Fig. 3(a). The LINKS-BOX rapid prototyping control platform was built by Beijing LINKS^{TM}, using the IBOX-704Plus embedded industrial computer with an Intel Core i5 7200U processor.The motor used was a 100 W permanent magnet synchronous servo motor, model SDGA-01C31BD, manufactured by TODE. The detailed parameters of the motor are shown in Table 2. A hysteresis brake was used to simulate the load conditions, capable of providing a stable load ranging from 0 to 1 NÂ·m. Additionally, the experimental platform included a torque sensor, a DRV8302 motor driver board, an EDR-120-24 switching power supply module and a host computer for developing control programs.

The ODPOS was coded according to the control architecture shown in Fig. 1 in the Simulink environment, and the compiled program was stored on the rapid prototyping control platform. The connections and communication of the experimental equipment are shown in Fig. 3b. The rapid prototyping platform received control commands from the host computer and generated PWM waveforms based on the stored control program to control the motor operation. The motor's operational status was also transmitted in real-time to the

Integrator index	Characteristic
ISE=$\int_0^t e^2 dt$	Fast response, large overshoot, poor stability.
IAE=$\int_0^t \lvert e\rvert dt$	Focuses on late response error, less consideration of pre-response errors.
ITSE=$\int_0^t t\cdot e^2 dt$	Excellent transient responsiveness, but unfavorable for parameter differentiation
ITAE=$\int_0^t t\cdot \lvert e\rvert dt$	Low transient response oscillation, but no distinction is made between overshoot and error.

Table 1. Error integrator index.



(a) Experimental platform for control of PMSM

(b) Block diagram of the experimental platform

Fig. 3. Experimental setup.

Parameter	Value
Rated power (W)	100
Rated DC voltage (V)	24
Rated speed (rpm)	3,000
Rated torque ($N \cdot m$)	0.32
Back EMF (V/1,000rpm)	3.5
Torque constant ($N \cdot m/A$)	0.06
Line resistance (Ω)	0.3
Line inductance (mH)	0.43
Number of pole pairs	4
Rotational inertia $J(kg \cdot m^2)$	0.175×10^{-4}

Table 2. Parameters of the 100 W interior PMSM.

host computer through the rapid prototyping control platform. The load was controlled by the hysteresis loading controller. In this experiment, the current loop frequency of the control algorithm was 10 kHz, and the speed loop frequency was 1 kHz.

Online optimization process

The embedded parameter optimization technique proposed in this paper was tested and validated using a classical PI control structure, as shown in Fig. 1. Different objective functions for the optimization model could be selected and applied according to specific requirements. This section focuses on the optimization process and the effect of the proposed ODPOS when the objective function was set to ITSE.

As mentioned earlier, the optimization algorithm embedded in the driver requires the system to be stabilized at the stabilized speed signal when evaluating the fitness of an individual. Therefore, we used the calculation method proposed in the "PMSM system model description" section to first calculate a control parameter that could stabilize the system. The value of K_P was calculated to be 0.059 and K_I was calculated to be 0.168 in the speed loop of the PMSM control system, while the value of CK_p was calculated to be 0.38 and CK_I was calculated to be 38 in the current loop. These parameters were applied as initial parameters in the control system, and the response performance of the motor is shown as curve BPO in Fig. 5. From the figure, it can be seen that under the control using these parameters, the system exhibited a large amount of overshoot, a long stabilization time, and a certain degree of oscillation. All these indicated that the control effect was not ideal and still needed to be further optimized.

Although the performance using the control parameters obtained through theoretical calculations was not satisfactory, the parameters ensured the convergence and stability of the system. This was sufficient to ensure the normal operation of the optimization algorithm. On this basis, the parameters of the control system were optimized using the ITSE objective function. The HBA parameter settings are shown in Table 3. Figure 4 shows the optimization process of the system.

The horizontal axis in Fig. 4 represents the time spent in the optimization process, and the vertical axes represent the motor speed and the control parameters of the driver. A complete individual fitness calculation

Parameter	Value
Population size	10
Number of iterations	15
C	2
β	10
Duration of response signal(s)	1
Steady signal duration(s)	1

Table 3. Parameter values of embedded HBA.

process started when the reference speed module gave the response speed, and the Optimization Algorithm module output the control parameters that needed to be evaluated. When the reference speed changed to the stabilized speed, the control parameters were reset to the currently found optimal parameters until the next time the reference speed module gave the response speed. This process took about 2 s. The population size in the algorithm was 30, the number of iterations was 10, and the whole parameter optimization process took about 600 s.

As shown in Fig. 4b at 302 seconds, the algorithm enters the sixth iteration, corresponding to $iter = 6, i = 1$ in the Algorithm 2. According to the description in Algorithm 2, the Optimization Algorithm module synchronously changed the output control parameters and started to record the error between the actual speed and the reference speed when the n_{ref} changed to response rpm. After the n_{ref} changed to stabilized rpm at section 303, the control parameters were changed back to the optimal parameters through step(6), (3), (4) and (5) to stabilize the system and prepare for the next control parameter fitness evaluation.

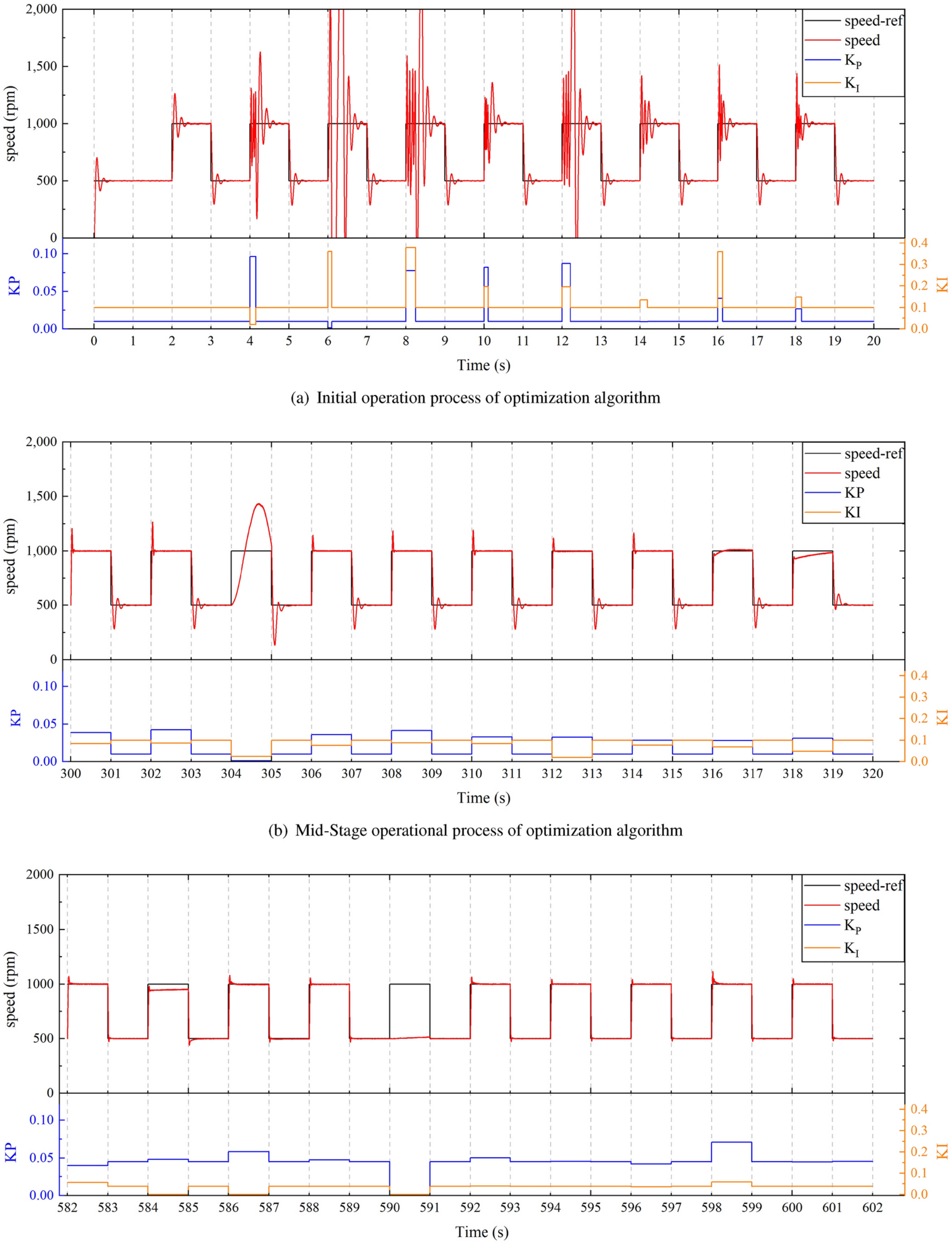
It is worth mentioning that due to the inclusion of the FSM, when the system detected vibrations or runaway failure, the algorithm immediately changed the control parameters back to the optimal values instead of waiting to receive the stabilized speed signal, which is fully demonstrated in Fig. 4a. At the fourth second, for example, the improper control parameters led to severe vibration in the system. The algorithm changed the output control parameters once it detected the vibration, thus preventing the system from getting out of control further.

In the early stage of the optimization process, as shown in Fig. 4a, none of the control effects were satisfactory because all the parameters except the initial control parameters were randomly generated in the first iteration. The control parameters were changed to optimal by the FSM before the end of the response speed signal. With the optimization process, the algorithm started updating the control parameters based on the position of the optimal individual. In the middle stage of the optimization process, as shown in Fig. 4b, most of the control parameters given by the module were able to make the system converge and operate stably. In the final stage of the optimization process, as shown in Fig. 4c, the module performed fine-tuning in the neighborhood of the existing optimal parameters with the aim of finding more ideal control parameters. After 600 s of the optimization process, the Optimization Algorithm module finally provided a set of control parameters that satisfied the control objective, which were the final control parameters.

Figure 5 clearly shows the dynamic response of the system when the reference speed increased from 500 to 1000 rpm before and after the parameter optimization. From the figure, it can be seen that after the optimization of the control parameters by the algorithm, the response time of the motor system was significantly shortened, which means that the system responded more quickly to the speed change. At the same time, the amount of overshoot is also significantly reduced, the fluctuations and vibrations of the system in the process of reaching the steady state are reduced, and the stability of the system was improved. In addition, the stabilization time was reduced, and the system was able to reach a steady state more quickly, further enhancing its reliability in practical applications.

The proposed OPDOS exhibits strong adaptability to different working conditions and can perform optimization for various control requirements. Figure 6 illustrates the response characteristics of the system under no-load conditions when minimizing the IAE as the optimization objective. The reference speed transitioned from 0 rpm to 3000 rpm at time 0. The "calc" curve represents the control effect based on the initial calculation of the control parameters, while the "optimize" curve shows the control effect achieved after the optimization process with the response speed set to 3,000 rpm and the stabilized speed set to 0 rpm. By comparing the two control effects, it can be seen that the optimized control parameters significantly reduced the overshooting phenomenon of the system, which not only helped to smooth the dynamic process of the system, but also significantly shortened the time required for the system to reach a stable state. Based on the results of Fig. 5, it also showed that the proposed ODPOS optimization algorithm has the ability to find the optimum under operating conditions with different speeds.

Figure 7 illustrates the optimization results under various loading conditions. Figure 7a shows the response characteristics of the control parameters optimized by the scheme when the response speed was set to 1,000 rpm and the stabilized speed was set to 0 rpm under no load conditions. After the rated load was suddenly applied, the speed dropped by 102.3 rpm, and it took 2.20 s to recover to the rated speed. After the load was removed, there was a speed fluctuation of 110.5 rpm, which took 2.03 s to eliminate. Figure 7b demonstrates the response characteristics of the system with the control parameters optimized under contirions with a 0.32 NÂ·m load condition. Compared to Fig. 7a, the speed fluctuation after loading was only 29.8 rpm, which was better than the optimization result found under no-load conditions. The time for the speed to stabilize also decreased by 1.31 s compared to the no-load conditions. The response characteristics after unloading were also improved. The load



(a) Initial operation process of optimization algorithm

(b) Mid-Stage operational process of optimization algorithm

(c) Later operational process of optimization algorithm

Fig. 4. Operational status of optimization algorithms at different stages.

experiments showed that applying the proposed ODPOS optimization algorithm under loading conditions can significantly enhance the control performance of the system under load conditions.

Through the above experiments, it was demonstrated that the proposed ODPOS could perform targeted optimizations for different speeds and loading conditions, showing good adaptability to various operating conditions.

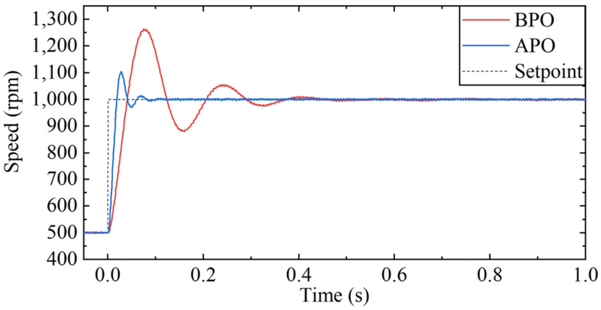


Fig. 5. Comparison of system control effect before and after parameter optimization. BPO represents the control effect generated by the theoretically calculated parameters, and APO signifies the improved control effect achieved through the optimized parameters obtained via the ODPOS.

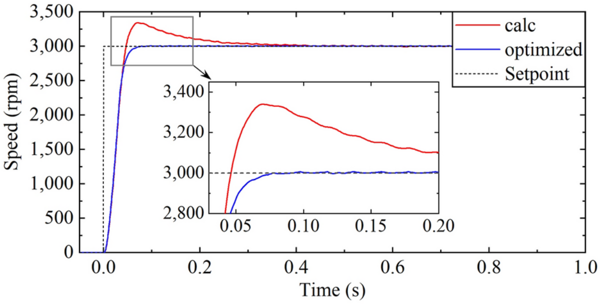
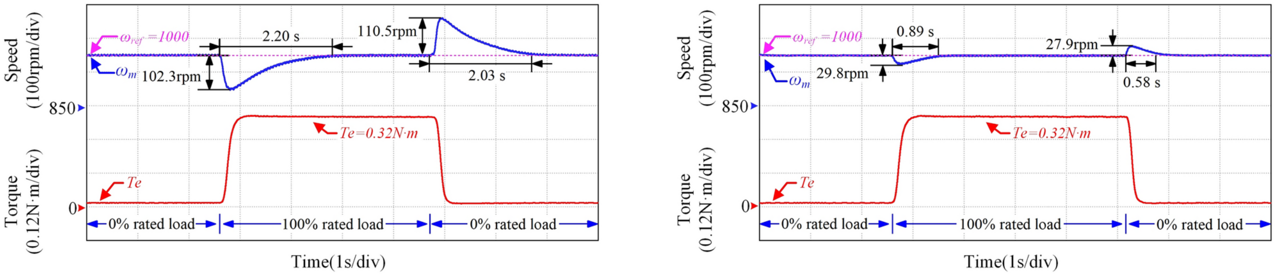


Fig. 6. Comparison of the response characteristics when the reference speed increased from 0 to 3000 rpm.



(a) Optimization result under no-load conditions

(b) Optimization result under loaded conditions

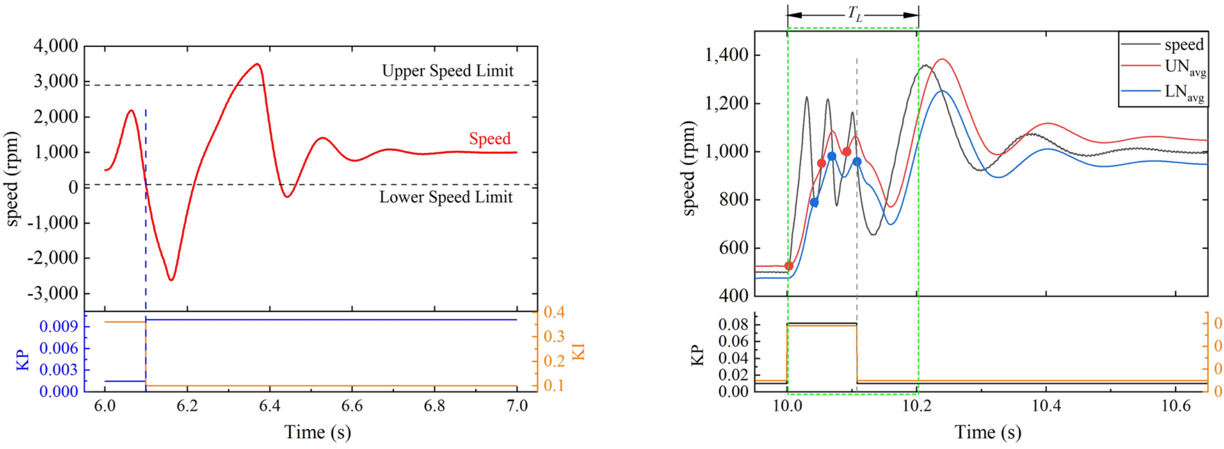
Fig. 7. Optimized system response characteristics under loaded and no-load conditions.

Effectiveness of FSM

This section demonstrates the effectiveness of the FSM during the optimization process. The mechanism included an overshoot suppression module and a vibration suppression module, which aimed to ensure that the system would not experience severe instability due to suboptimal solutions found during the optimization process.

The effect of the overshoot suppression module is shown in Fig. 8a. The Upper Speed Limit and Lower Speed Limit in the figure represent the non-fault speed range set in the overshoot suppression module. When the speed exceeded the upper limit or fell below the lower limit, this module could quickly detect the occurrence of overshoot and immediately adjust the control parameters to the optimal ones found during the optimization process. The blue dashed line in the figure indicates the time when the speed fell below the lower speed limit, at which moment the algorithm simultaneously adjusted the control parameters to prevent further loss of control of the system. Although the system could still experience a certain degree of oscillation due to the inertia of the system and the limitations of the control effect with the current optimal parameters, the adjusted parameters were sufficient to stabilize the system in the end.

The effect of the vibration suppression module is clearly demonstrated in Fig. 8b. The red dots in the figure mark the moments when the speed exceeded UN_{avg}, while the blue dots mark the moments when the speed fell below LN_{avg}. The green rectangular boxes represent the given time range T_L. The window range *n* in Eq. (23)



(a) Effect of overshoot suppression module

(b) Effect of vibration suppression module

Fig. 8. Effect of FSM.

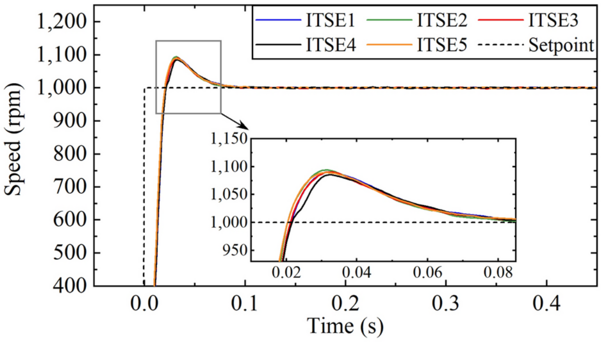


Fig. 9. Comparison of five optimization results.

Metrics	ITSE1	ITSE2	ITSE3	ITSE4	ITSE5
Rise time (ms)	18.26	18.67	18.58	18.16	18.22
Overshoot	9.02%	9.41%	9.08%	8.76%	8.95%
Adjustment time (ms)	77.37	78.16	76.74	76.05	75.91
Fitness value	725,413	729,119	717,330	720,158	722,066

Table 4. Control performances of five optimizations of the ITSE objective function.

was set to 500, the upper and lower limit percentage δ in Eq. (24) was set to 5%, and the given time range T_L was set to 0.2 s. T_L started timing when the speed first exceeded the threshold. Over the next 0.2 s, the speed exceeded both the upper and lower limits of the average speed more than three times, as indicated by the red and blue dots in the figure, respectively. The last occurrence of overshoot satisfied the vibration suppression trigger condition, and the vibration suppression indicator V changed from 0 to 1, indicating that the module had detected vibration in the system. Subsequently, the output parameters were quickly adjusted to the optimal values found thus far. With this adjustment, the vibration of the system was effectively suppressed, which ensured the stable operation of the system.

The effectiveness of the proposed FSM is fully demonstrated in Fig. 4a. As can be seen in the figure, during the early stages of parameter optimization, inappropriate control parameters often led to overshoot and vibration of the system. However, the FSM ensured the stable operation of the optimization process and played an important role in the whole parameter optimization process.

Comparison of optimization effects under different objectives

The proposed parameter optimization scheme should be capable of finding the optimal control parameters corresponding to different objective functions. For this purpose, four different objective functions,IAE, ISE,

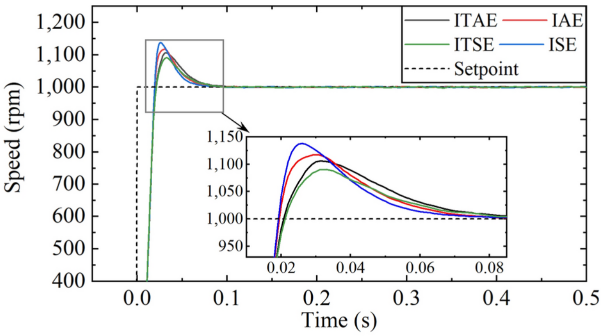


Fig. 10. Comparison of optimization search results with different optimization objectives.

Metrics	ITAE	IAE	ITSE	ISE
Rise time (ms)	18.01	17.65	18.58	17.56
Overshoot	10.61%	11.78%	9.08%	13.97%
Adjustment time (ms)	74.13	68.88	76.74	64.85
Fitness value	5,508	169,527	717,300	107,921,516

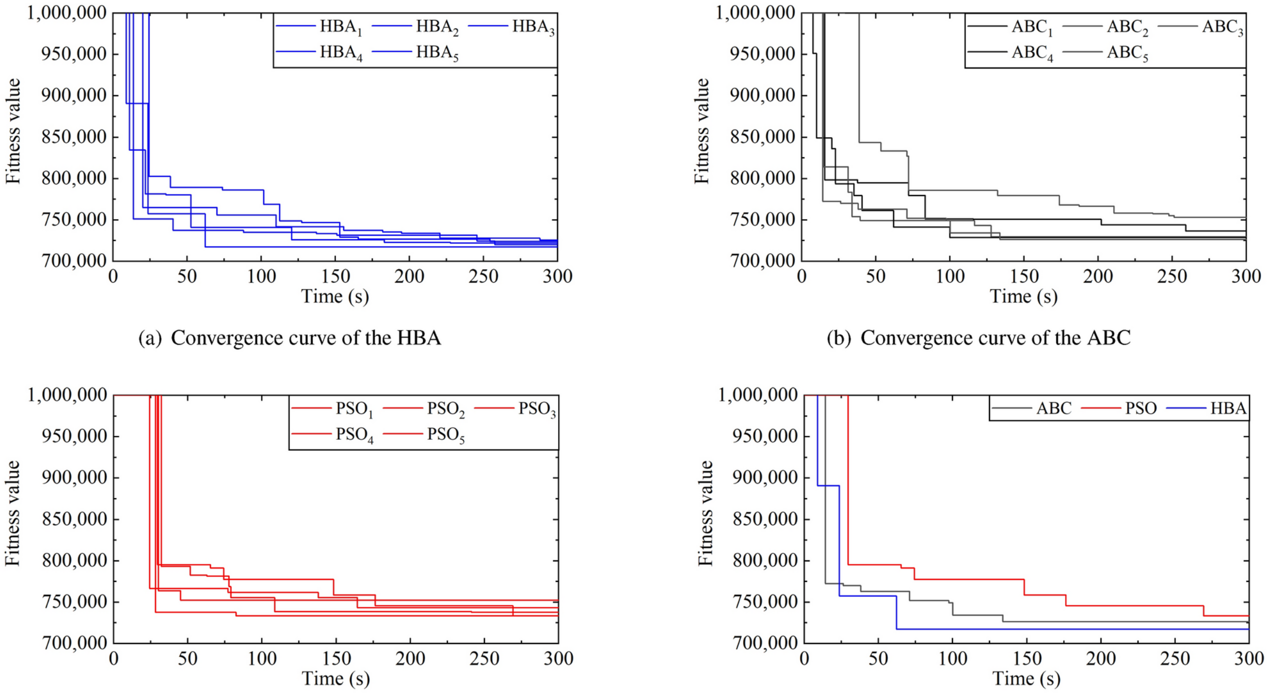
Table 5. Optimal control performance found with different objective functions.

ITAE, and ITSE, were used to optimize the control performance at 1000 rpm under no-load operating conditions. Due to the stochastic nature of the optimization algorithm, each objective function was optimized five times.

Figure 9 displays the control performance of the optimal parameters found through five optimizations of the ITSE objective function. The corresponding control performance metrics and fitness values are shown in Table 4. In the table, rise time represents the time when the rotational speed reached 90% of the Setpoint. Overshoot indicates the percentage the percentage by which the maximum speed exceeds the reference speed. Adjustment time was the time when the actual rotational speed enterd within 1%. In Table 4, the third optimization had the smallest fitness value, however, its corresponding rise time, overshoot, and adjustment time were not the smallest. This was primarily because the ITSE is a comprehensive index that takes into account the overall error in a time period, so the optimization goal was not to minimize a certain control performance metric. By comparing the control performance metrics obtained from each optimization, the optimal control performances achieved by each optimization scheme were very similar despite the randomness of the search process.

Figure 10 displays the motor control performance obtained from optimizations using different objective functions. For each objective function, the best result from five optimizations was selected for comparison. To further evaluate the optimal performance of the system under different objective functions, Table 5 lists three indicators for assessing the system's response performance. Among the indicators for the rise time, ISE performed the best. This was because ISE squared the error, causing the algorithm to prioritize reducing large errors during the optimization process, thereby accelerating the system's response speed. However, this characteristic also led to an increase in the overshoot. Compared with IAE and ISE, the ITAE and IISE indicators do not square the error, so the penalty for the error is linear. The algorithm did not aggressively attempt to reduce errors during the optimization process, thereby reducing the risk of overshoot. This also led to a longer adjustment time. These results demonstrated that the proposed approach can flexibly address diverse objective requirements and achieve good control performance, thereby validating its optimization capability.

The HBA used in the scheme was compared with the PSO[42] and ABC[43] algorithms, which have also been used for parameter optimization in motors. All three algorithms were used to optimize the no-load performance of the motor at 1,000 rpm with ITSE as the optimization index. Each algorithm was run five times. Figure 11 shows the convergence curves for each algorithm, and the final results of each optimization are displayed in Table 6. A smaller fitness value indicates a better control performance achieved by the currently found control parameters. The optimization results obtained using the HBA algorithm were all within 730,000. The optimization results had good consistency, which indicated the stable optimization ability of the algorithm. In contrast, two of the optimization results obtained using the ABC algorithm had fitness values exceeding 730,000, suggesting that the stability of the algorithm was poor. The PSO algorithm's optimization results were scattered between 733,509 and 752,224, and the large difference in each optimization result indicated that the optimization effect of the PSO algorithm was poor. Additionally, in table 6, the coefficient of variation was calculated using multiple operational results from each algorithm. A smaller coefficient of variation indicates a higher consistency in the search results of the algorithm. As shown in the table, the HBA algorithm has a smaller coefficient of variation, indicating that its optimization results are more concentrated and stable. Notably, the HBA algorithm shows a 40% improvement in the consistency of its optimization results compared to the ABC algorithm, which demonstrates the superiority of the HBA algorithm.



(a) Convergence curve of the HBA

(b) Convergence curve of the ABC

(c) Convergence curve of the PSO

(d) Comparison of the best convergence curves

Fig. 11. Convergence curves of different algorithms.

	First run	Second run	Third run	Fourth run	Fifth run	Coefficient of variation
HBA	717,330	720,158	722,006	725,413	729,119	0.0057
ABC	726,259	728,644	729,516	736,387	753,240	0.0133
PSO	733,509	733,603	737,815	743,369	752,224	0.0095

Table 6. Final fitness value obtained from each optimization.

The best results of the different algorithms were compared, as shown in Fig. 11d, The HBA had a faster convergence rate, and the final optimization results were improved by 1.2% and 2.2% compared to the ABC and PSO algorithms, respectively. The PSO had poorer optimization capabilities, which was mainly due to the long computation time required for the fitness calculation of a single individual in practical applications. This necessitated a reduction in the number of individuals and the number of iterations to decrease optimization time. The PSO relies heavily on a large number of search individuals, and the reduction of the number of particles has a greater impact on its search effect. The ABC algorithm, which relies on the division of labor among employed bees, onlooker bees, and scout bees to find the best solution, requires more iterations to achieve satisfactory results in the face of complex problems. The reduction in the number of iterations led to a slower convergence rate. The HBA algorithm, benefiting from its "exploration phase" and "exploitation phase," had strong evolutionary capabilities and fast search speeds. It was also able to achieve good optimization results even when the population size was reduced. Therefore, the HBA is more suitable for solving controller parameters optimization problems.

It is worth noting that the experimental environment differed from the simulation environment, and there was a certain degree of variation in the initial state of the system's response each time, as well as external disturbances. These factors led to a deviations in the system's response results under the same parameters. Such deviations could impact the system's optimization process. However, in terms of overall effectiveness, the proposed ODPOS was still able to find control parameters more suitable for the control objective in the actual deployment environment.

Conclusions

This paper presents a unique online deployable ODPOS for PMSMs, utilising the HSA to tackle the issues of real-time optimisation in drive systems. The ODPOS is intended for integration into the motor control system, facilitating adaptive adjustment of control parameters to accommodate diverse operational requirements. This research introduces a novel evaluation metric that customises the optimisation process to specific control performance indicators, including overshoot, settling time, and stability accuracy, instead of depending simply on conventional integral metrics.

The ODPOS underwent thorough testing on a rapid prototyping control platform, showcasing its efficacy in attaining optimal control performance under various operating situations. The findings demonstrated that the HSA-based optimisation technique converges more rapidly and yields more stable outcomes than alternative algorithms, achieving a 2.2% enhancement in optimisation efficacy over current options.

This discovery is significant due to its potential to revolutionise the optimisation of control parameters in PMSMs, particularly in reconfigurable production settings where adaptability is essential. Integrating the optimisation method within the motor controller has facilitated real-time, self-tuning control systems capable of dynamically adapting to variations in operating conditions, hence improving performance and reliability.

We intend to improve the ODPOS by including ANNs optimised by the Levenberg-Marquardt technique (LMT) to minimise optimisation time. Furthermore, we create new optimisation indices to better align the metrics with the system's reaction characteristics, facilitating more precise performance optimisation. We also plan to modify the criteria for determining whether the system is stable or not in the stabilized state to further reduce the optimization time. Furthermore, we intend to apply the proposed scheme to the hyperparameter optimization in other more advanced control methods to expand the application scope of this scheme.This research advances the development of intelligent motor control systems, presenting a significant opportunity to enhance the efficiency and adaptability of PMSMs in industrial applications.

Data availability

The datasets used during the current study are available from the corresponding author upon reasonable request.

Received: 11 June 2024; Accepted: 21 October 2024
Published online: 04 November 2024

References

1. Sun, X., Shi, Z., Lei, G., Guo, Y. & Zhu, J. Analysis and design optimization of a permanent magnet synchronous motor for a campus patrol electric vehicle. *IEEE Trans. Veh. Technol.***68**, 10535–10544 (2019).
2. Luu, P. T., Lee, J.-Y., Lee, J.-H. & Park, J.-W. Electromagnetic and thermal analysis of permanent-magnet synchronous motors for cooperative robot applications. *IEEE Trans. Magn.***56**, 1–4 (2020).
3. Zhou, C., Huang, X., Li, Z. & Cao, W. Design consideration of fractional slot concentrated winding interior permanent magnet synchronous motor for EV and HEV applications. *IEEE Access***9**, 64116–64126 (2021).
4. Faiz, J. & Nejadi-Koti, H. Demagnetization fault indexes in permanent magnet synchronous motors–an overview. *IEEE Trans. Magn.***52**, 1–11 (2016).
5. Chen, Y., Liang, S., Li, W., Liang, H. & Wang, C. Faults and diagnosis methods of permanent magnet synchronous motors: A review. *Appl. Sci.***9** (2019).
6. Ullah, K., Guzinski, J. & Mirza, A. F. Critical review on robust speed control techniques for permanent magnet synchronous motor (PMSM) speed regulation. *Energies***15**, 1235 (2022).
7. Jin, F., Wan, H., Huang, Z. & Gu, M. PMSM vector control based on fuzzy PID controller. *J. Phys. Conf. Ser.***1617**, 012016 (2020).
8. Xu, W., Qu, S. & Zhang, C. Fast terminal sliding mode current control with adaptive extended state disturbance observer for PMSM system. *IEEE Trans. Emerg.***11**, 418–431 (2023).
9. Hou, Q., Ding, S. & Yu, X. Composite super-twisting sliding mode control design for PMSM speed regulation problem based on a novel disturbance observer. *IEEE Trans. Energy Convers.***36**, 2591–2599 (2021).
10. Zhang, Y., Jin, J. & Huang, L. Model-free predictive current control of PMSM drives based on extended state observer using ultralocal model. *IEEE Trans. Ind. Electron.***68**, 993–1003 (2021).
11. An, X., Liu, G., Chen, Q., Zhao, W. & Song, X. Adjustable model predictive control for IPMSM drives based on online stator inductance identification. *IEEE Trans. Ind. Electron.***69**, 3368–3381 (2022).
12. Bi, G. et al. Multi-optimization objective online tracking-based parameter self-tuning method for sensorless PMSM drives. *IEEE Trans. Transp. Electr.***9**, 1390–1402 (2023).
13. Wang, C. & Zhu, Z. Q. Fuzzy logic speed control of permanent magnet synchronous machine and feedback voltage ripple reduction in flux-weakening operation region. *IEEE Trans. Ind. Appl.***56**, 1505–1517 (2020).
14. Kuppusamy, S. & Joo, Y. H. Memory-based integral sliding-mode control for T–S fuzzy systems with PMSM via disturbance observer. *IEEE Trans. Cybern.***51**, 2457–2465 (2021).
15. Deng, Y., Wang, J., Li, H., Liu, J. & Tian, D. Adaptive sliding mode current control with sliding mode disturbance observer for PMSM drives. *ISA Trans.***88**, 113–126 (2019).
16. Zou, M., Yu, J., Ma, Y., Zhao, L. & Lin, C. Command filtering-based adaptive fuzzy control for permanent magnet synchronous motors with full-state constraints. *Inf. Sci.***518**, 1–12 (2020).
17. Sun, Z., Deng, Y., Wang, J., Li, H. & Cao, H. Improved cascaded model-free predictive speed control for PMSM speed ripple minimization based on ultra-local model. *ISA Trans.***143**, 666–677 (2023).
18. Liu, S. & Liu, C. Virtual-vector-based robust predictive current control for dual three-phase PMSM. *IEEE Trans. Ind. Electron.***68**, 2048–2058 (2021).
19. Zhao, K. et al. Model-free fast integral terminal sliding-mode control method based on improved fast terminal sliding-mode observer for PMSM with unknown disturbances. *ISA Trans.***143**, 572–581 (2023).
20. Liu, K. & Cheng, Q. Research on new direct torque control strategy of TLDMC-PMSM system. *Electr. Eng.***105**, 4213–4227 (2023).
21. 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.***34**, 5267–5282 (2022).
22. Borase, R. P., Maghade, D. K., Sondkar, S. Y. & Pawar, S. N. A review of PID control, tuning methods and applications. *Int. J. Dyn. Control***9**, 818–827 (2021).
23. Wang, H., Xu, S. & Hu, H. PID controller for PMSM speed control based on improved quantum genetic algorithm optimization. *IEEE Access***11**, 61091–61102 (2023).
24. Grimble, M.J. & Majecki, P. Introduction to nonlinear systems modelling and control. In *Nonlinear Ind. Control Syst.*, 3–63 (Springer, 2020).
25. Rösiö, C. & Säfsten, K. Reconfigurable production system design-theoretical and practical challenges. *J. Manuf. Technol. Manag.***24**, 998–1018 (2013).
26. Injeti, S.K. & Divyavathi, M. Optimal gain scheduling of pid controller for the speed control of pmsm drive using bio-inspired optimization algorithms. *Int. J. Electr. Eng. Inform.***11** (2019).

27. Baithalu, R., Mishra, S. & Ali Shah, N. Sensitivity analysis of various factors on the micropolar hybrid nanofluid flow with optimized heat transfer rate using response surface methodology: A statistical approach. *Phys. Fluids***35** (2023).
28. Anwar, N. *et al.* Stochastic runge–kutta for numerical treatment of dengue epidemic model with brownian uncertainty. *Modern Phys. Lett. B*, 2450408 (2024).
29. Nasab, M. A., Al-Shibli, W. K., Zand, M., Ehsan-maleki, B. & Padmanaban, S. Charging management of electric vehicles with the presence of renewable resources. *Renew. Energy Focus***48**, 100536 (2024).
30. Nasab, M. A. et al. Uncertainty compensation with coordinated control of evs and der systems in smart grids. *Solar Energy***263**, 111920 (2023).
31. Nasab, M.A. *et al.* Predicting solar power potential via an enhanced ann through the evolution of cub to predator (ecp) optimization technique. *Electr. Eng.* 1–12 (2024).
32. Anwar, N., Ahmad, I., Kiani, A. K., Shoaib, M. & Raja, M. A. Z. Novel neuro-stochastic adaptive supervised learning for numerical treatment of nonlinear epidemic delay differential system with impact of double diseases. *Int. J. Model. Simul.* 1–23 (2024).
33. Lambora, A., Gupta, K. & Chopra, K. Genetic algorithm-a literature review. In *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, 380–384 (IEEE, 2019).
34. Juneja, M. & Nagar, S. K. Particle swarm optimization algorithm and its parameters: A review. In *2016 International Conference on Control, Computing, Communication and Materials (ICCCCM)*, 1–5 (IEEE, 2016).
35. Deng, W. et al. An improved differential evolution algorithm and its application in optimization problem. *Soft Comput.***25**, 5277–5298 (2021).
36. Injeti, S. K. & Divyavathi, M. Optimal gain scheduling of PID controller for the speed control of PMSM drive using bio-inspired optimization algorithms. *Int. J. Electr. Eng. Inform.***11** (2019).
37. Mahto, R. K. & Mishra, A. Self-tuning vector controlled PMSM drive using particle swarm optimization. In *2020 IEEE First International Conference on Smart Technologies for Power, Energy and Control (STPEC)*, 1–5 (2020).
38. Templos-Santos, J. L., Aguilar-Mejia, O., Peralta-Sanchez, E. & Sosa-Cortez, R. Parameter tuning of PI control for speed regulation of a PMSM using bio-inspired algorithms. *Algorithms***12**, 54 (2019).
39. Ünsal, S. & Aliskan, I. Investigation of performance of fuzzy logic controllers optimized with the hybrid genetic-gravitational search algorithm for PMSM speed control. *Automatika***63**, 313–327 (2022).
40. Ganguli, S., Kumar, A., Kaur, G., Sarkar, P. & Rajest, S. S. A global optimization technique for modeling and control of permanent magnet synchronous motor drive. *Innov. Inf. Commun. Technol. Ser.*, 74–81 (2021).
41. Yao, C. et al. ANN optimization of weighting factors using genetic algorithm for model predictive control of PMSM drives. *IEEE Trans. Ind. Appl.***58**, 7346–7362 (2022).
42. Wang, S., Jiang, C., Tu, Q., Shu, H. & Zhu, C. Particle swarm optimization of fuzzy pi control for pmsms. *J. Power Electron.***23**, 1585–1593 (2023).
43. Ciabattoni, L. et al. A robust and self-tuning speed control for permanent magnet synchronous motors via meta-heuristic optimization. *Int. J. Adv. Manuf. Technol.***96**, 1283–1292 (2018).
44. Hashim, F. A., Houssein, E. H., Hussain, K., Mabrouk, M. S. & Al-Atabany, W. Honey badger algorithm: New metaheuristic algorithm for solving optimization problems. *Math. Comput. Simul.***192**, 84–110 (2022).

Acknowledgements

We thank LetPub (www.letpub.com) for its linguistic assistance during the preparation of this manuscript.

Author contributions

X.Z. wrote the main manuscript text, Y.H. contributed to the conception and design of the work, Y.Y. proposed original research idea and provide research funding, Y.H. and D.Z. reviewed and revised the article, J.Y. provided guidance on the theoretical section, and G.C. reviewed paper and gave the commens. All authors read and approved the final manuscript.

Funding

This study was funded by Less Developed Regions of the National Natural Science Foundation of China [grant number 52067006], Young Scientists Fund of the National Natural Science Foundation of China [grant number 52302470], Natural Science Foundation of Jiangxi Province [grant number 20224BAB214045], Ganpo Talent Support Program-Leading Academic and Technical Personnel in Major Disciplines of Jiangxi Province [grant number 20225BCJ22015], Key R & D Program of Jiangxi Province [grant number 20243BBG71011], The Open bidding for selecting the best candidates Program of Nanchang [grant number 2022JBGS001], 03 Special Program and 5G Project of Jiangxi Province [grant number 20232ABC03A30], Natural Science Foundation of Jiangxi Province [grant number 20232BAB214092], and Ganpo Talent Support Program-Leading Academic and Technical Personnel in Major Disciplines of Jiangxi Province [grant number 20232BCJ23091].

Declarations

Competing Interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Y.H.

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

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