



Speed Control of PMSM Using Modified Particle Swarm Optimization Technique Based on Inertia Weight Updating Mechanism

Raja Gandhi^{1,2} · Dibyadeep Bhattacharya^{1,2} · Ajay Anand^{1,2} · Sadhan Gope² · Arpita Banik^{1,2} · Rakesh Roy^{1,2}

Received: 13 March 2023 / Accepted: 26 June 2023

© The Author(s), under exclusive licence to Springer Nature Singapore Pte Ltd 2023

Abstract

In this paper, the modified Particle Swarm Optimization (PSO) algorithm is implemented to tune the gain parameters of the PI speed controller of the PMSM drive system. The PSO is one of the artificial intelligence techniques which is modified with the inertia weight updating mechanism to prevent premature convergence and balance the exploration and exploitation of the particles. The field-oriented vector control PMSM drive is developed in MATLAB/Simulink to examine three different conditions such as start-up, speed command change, and sudden load torque imposition. The different parameters are then examined such as speed overshoot, settling time, peak time, rise time and speed ripple and the results are compared with conventional PSO-tuned PI controllers for the same motor. From the results, it is proved that the modified PSO-PI controller gives better performance compared to the conventional PSO-PI speed controller.

Keywords PMSM · Particle swarm optimization · Field oriented control · Vector control · PI controller

Introduction

Permanent magnet machines have been rapidly gaining popularity as they are being increasingly used in the electrical drive industry, starting from hard disk drives to electric

vehicles [1]. This is due to the fact that these machines provide us with few distinctive advantages over their counterparts. To name a few, the PM machines display high efficiency, high torque, and faster dynamics and they also come in compact sizes. These motors are therefore being preferred for high-performance motor drive applications [2, 3]. One such motor is a permanent magnet synchronous motor (PMSM). In addition to the above-mentioned advantages, the PMSM also exhibits high starting torque. In order to convert the PMSM into a high-performance drive, a robust drive control strategy is needed to deal with its nonlinear behavior. It has been seen through a literature survey that out of the two prevailing topologies of control strategies (scalar and vector control), the vector control technique was found to be the most widely used due to its ability to provide control of the torque and flux components separately [4–6]. It is mandatory for a high performance drive to provide a smooth running system that allows rapid recovery from transients and other disturbances. In order to achieve that, a speed controller is required that prohibits any sudden changes in the speed of the motor drive due to unwanted disturbances. Most of the speed controllers used in such motor drive systems are PI (proportional integral) controllers [7, 8]. The tuning of a PI controller is basically tuning its parameters k_p and k_i which are normally achieved by trial and error methods that are time consuming and not that accurate, especially in case

This article is part of the topical collection “SWOT to AI-embraced Communication Systems (SWOT-AI)” guest edited by Somnath Mukhopadhyay, Debashis De, Sunita Sarkar and Celia Shahnaz.

✉ Raja Gandhi
nitraja06@gmail.com

Dibyadeep Bhattacharya
P19ee002@nitm.ac.in

Ajay Anand
P19ee002@nitm.ac.in

Sadhan Gope
sadhan.nit@gmail.com

Arpita Banik
arpitabanik891@gmail.com

Rakesh Roy
Rakesh.Roy@nitm.ac.in

¹ Department of Electrical Engineering, NIT Meghalaya, Bijnai Complex, Laitumkhrach, Shillong, Meghalaya 793003, India

² Department of Electrical Engineering, NIT Agartala, Barjala, Jirania, Agartala, Tripura 1799046, India

of non-linearity [9, 10]. Therefore, in order for the motor drive to work smoothly the PI parameters (k_p & k_i) had to be tuned correctly.

Numerous optimization techniques, including the genetic algorithm, grey wolf optimization, JAYA, teaching learning, particle swarm optimization (PSO), whale, ant colony optimization, fuzzy logic control, etc have been used in the literature to tune the PI parameters and give better tuning results compared to conventional methods [11–16]. Due to its simplicity and ease of implementation, PSO is one of the most well-liked optimization techniques among them. However, the disadvantage of this technique is that it easily taps into local optimal solutions and therefore often provides inaccurate global optimized solutions. The above reasons warrant an improvement in the PSO-optimized system to find the best solution in space more accurately.

In this paper, the PI controller parameters for the vector controlled PMSM drive are tuned using a modified PSO algorithm. Here, a modified inertia weight updating strategy is applied to balance particle exploitation and exploration in search space. The addition of this modification allows the determination of the optimal values of the PI parameters (k_p and k_i) and prevents the algorithm from premature convergence. The performance of the modified PSO-PI-based PMSM drive is analyzed based on speed, current, and torque responses. To prove the efficiency of the aforementioned method another identical PMSM drive controlled by conventional PSO-PI controller is subjected to the same parameter based analysis. Different parameters such as speed overshoot, rise time, peak time, settling time and speed ripple are analyzed.

The solution for the above-mentioned problems has been dealt in the following way in this paper. First, a brief description of a vector-controlled PMSM drive has been

given. Next, the optimization technique i.e Particle Swarm Optimization, and its modifications used in the work has been introduced. This introduction is then followed by the description of the process for the acquisition of the magnitudes of K_p and K_i using the aforementioned modified PSO algorithm. Finally, the results obtained by the above process on the simulated platform have been shown and discussed, followed by the conclusion.

Field Oriented Vector Control of PMSM Drive

Figure 1 shows the schematic view of a vector control PMSM drive using a modified PSO-optimized PI speed controller. The vector control method is widely accepted due to its ability to independently adjust the signal’s phase and amplitude. The field-oriented vector control PMSM is implemented here due to its simplicity, ease of use and fast dynamic response. Using this control strategy, both the torque and flux parts of the PMSM drive system can be controlled separately. The dynamic equations of a PMSM drive in a rotor reference frame, as well as the electromagnetic torque equations, are as follows:

$$\begin{bmatrix} I_{qs}^r \\ I_{ds}^r \\ I_o^r \end{bmatrix} = \begin{bmatrix} \cos(\delta) & \cos(\delta - \frac{2\pi}{3}) & \cos(\delta + \frac{2\pi}{3}) \\ \sin(\delta) & \sin(\delta - \frac{2\pi}{3}) & \sin(\delta + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} \tag{1}$$

$$T_e = \frac{3P}{4} [\phi_{af} I_{qs}^r + (L_d - L_q) I_{ds}^r I_{qs}^r] \tag{2}$$

In FOC $i_{d_{ref}} = 0$ ($i_{d_{ref}} = I_{ds}^r$) and the torque of the motor is controlled by changing the q -axis current as shown in (3)

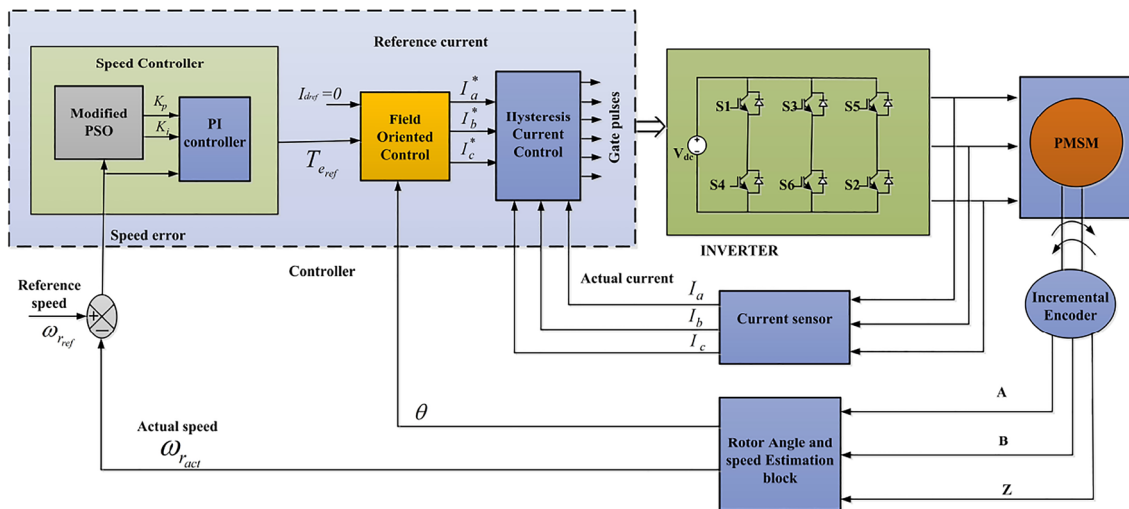


Fig. 1 Block diagram of Field oriented vector controlled PMSM drive with modified PSO-PI speed controller

$$T_e = \frac{3P}{4} [\phi_{af} I_{qs}^r] \tag{3}$$

From Fig. 1, it can be seen that the two feedback loops are present to regulate the PMSM drive system. The inner loop is used to control the supply current and the outer loop is a speed control loop that is used to control the speed response of the motor. In the speed control loop, the speed error has been fed to a modified PSO block to calculate the optimum gain (k_p & k_i) values for the PI controller. These gain values are then used to calculate the reference electromagnetic torque of the motor using (4). By using Eqs. (3) and (4), the reference q -axis current has been calculated and given to the FOC block to calculate the reference stator current. The error of the stator’s actual and reference currents passes through the hysteresis band of the HCC to produce the PWM pulse signal of the inverter. Therefore, to run the motor very smoothly, the optimum gain values of the PI parameter are required, and this has been made possible with the meta-heuristic optimization approaches.

$$T_{e_{ref}} = K_p(\omega_{error}) + \int K_i(\omega_{error})dt \tag{4}$$

Particle Swarm Optimization Technique

The PSO technique is a meta-heuristic algorithm first developed in 1995 and inspired by the social behavior of birds or fish [1]. To achieve the optimum solution it replicates the movement of a flock of birds or fish by adjusting the particle location by rejigging its velocity information. Due to its simplicity and ease of implementation, this technique is widely used in numerous domains, including classification, neural network training, machine study, signal processing etc and this type of artificial intelligence method is also good for optimizing parameters in a continuous search space [2]. Considering the position and velocity of the n number of particles are $x_i = [x_1, x_2, \dots, x_n]$ and $v_i = [v_1, v_2, \dots, v_n]$ for the N -dimensional search space. Historically, the best position of particle i is $P_{i_{best}} = [p_{1_{best}}, p_{2_{best}}, \dots, p_{n_{best}}]$ and the global best position is $g_{best} = [g_{1_{best}}, g_{2_{best}}, \dots, g_{n_{best}}]$. The formula for updating the position and velocity of each particle during each iteration (t) is presented by (5) and (6):

$$V_i(t + 1) = V_i(t) + c_1 r_1 (P_{i_{best}}(t) - x_i(t)) + c_2 r_2 (g_{i_{best}}(t) - x_i(t)) \tag{5}$$

$$x_i(t + 1) = x_i(t) + V_i(t + 1) \tag{6}$$

The second part of Eq. (5) is called the cognitive component and it shows how the particle remembers its own past experiences. The third part is called the social

part and it shows how particles share information and work together. Each particle’s motion is influenced by the learning variables c_1 and c_2 . If c_1 or c_2 is zero, each particle’s movement is determined by either the population’s best historical information (g_{best}) or by its own individual optimal historical information (p_{best}). To enhance performance, many researchers added inertia weight (w) to the first portion of the velocity update formula and the resulting generalized is given by Eq. (7):

$$V_i(t + 1) = wV_i(t) + c_1 r_1 (P_{i_{best}}(t) - x_i(t)) + c_2 r_2 (g_{i_{best}}(t) - x_i(t)) \tag{7}$$

$$\text{where } w = (w_{max} - w_{min}) \frac{(t_{max} - t)}{t_{max}} + w_{min}$$

Modification of the PSO Algorithm Based on Inertia Weight

In PSO, the inertia weight (w) coefficient has a prominent influence on its performance. It is essential in achieving a balance between the particles’ exploration and exploitation to reach the optimum solution. This inertia weight helps to regulate the inertial motion of the swarm particles by damping or accelerating the particle velocity to its original direction. To balance exploration (global search) and exploitation (local search), the value of the w must be within the particle’s searching ability. If the value of w is large, the particle’s velocity is also large, resulting in a large step size while performing a global search. It has a strong global search ability, but poor local search capability. The smaller the value of inertia weight, the smaller the step size of the particles. This is good for local search and has a high local search ability. However, if the inertial weight parameter is too high or too low, the algorithm will fail to converge to its optimum solution. So, to get the best solution, the value of the inertia weight must be carefully chosen. In the traditional PSO algorithm, inertia weight is updated using (7), where w_{max} and w_{min} remain constants throughout the operation and inertia weight (w) changes linearly with each iteration. When a PMSM experiences a disturbance, the typical inertia weight updating approach may encounter premature convergence and offer an unreliable optimum solution. Therefore, the classical PSO algorithm is modified in this work based on a novel inertia weight updating approach. The inertia weight is determined using Eq. (8), where the w_{min} value is fixed but the w_{max} value changes with every iteration to achieve the best balance between the local and global search conducted by particles for a successful analysis of the specified search space:

$$w(t) = w_{\max}(t - 1) - \frac{w_{\max}(t - 1) - w_{\min}}{t_{\max}} t \tag{8}$$

Determination of Optimized k_p and k_i Gains Using Modified PSO Algorithm

Using the flow chart in Fig. 2, a detailed procedure to determine the optimum k_p and k_i gain magnitudes for the PI controller is outlined below:

1. Initialization of parameters of the PSO, such as the total number of particles (n), upper and lower limit of the particle (U_L & L_L), maximum iteration (t_{\max}), learning variables c_1 and c_2 , random variables r_1 and r_2 , and maximum and minimum limit of inertia weight (w_{\max} & w_{\min}), thus calculate the initial position of the particle randomly using Eq. (9):

$$x_{i,j} = (U_L - L_L) * \text{rand}(1, j) + L_L, \tag{9}$$

where j denotes number of dimension, $x_{i,1} = K_p$ & $x_{i,2} = K_i$

2. Using the particle position information, the PMSM drive is running n -times. For each operation, the total speed

error (TSE) is calculated using (10), and the fitness function is saved with the lowest speed error in (11):

$$\text{TSE} = \sum_0^T (\omega_{r_{\text{ref}}} - \omega_{r_{\text{act}}})^2 \tag{10}$$

$$\text{Fitness function} = \min \sum_0^T (\omega_{\text{error}})^2 \tag{11}$$

3. The fitness function is now examined to check the least speed error condition. The particle's p_{best} and g_{best} are updated with new P_{best} and g_{best} values if the fitness function of the new position is better than the previous entry. Otherwise, the memory remains intact.
4. In this step, the algorithm updates the particle's inertia weight, velocity, and position and repeats step 2 if it does not reach its maximum number of iterations. On the other hand, if $t = t_{\max}$, the algorithm will stop. After the program has terminated, the final optimized values are set as the final K_p and K_i values that correspond to the least speed error seen in the fitness function.

Simulation Results and Discussion

To evaluate the performance of the proposed optimized PI speed controller-based PMSM drive system, three different cases such as start-up, speed command change, and sudden load torque imposition are investigated. The simulation model of the FOC vector-controlled PMSM drive is developed in MATLAB/Simulink. For each case, the speed, current, and torque waveform are examined along with various parameters like the speed ripple, rise time, speed overshoot, settling time, and peak time. The rise time and settling time are computed when the motor speed reaches 90% and 99.8% of the reference speed. The peak time of the motor is the time required to reach its maximum speed with overshoot. To demonstrate improved performance with the proposed technique, the results are then compared to those obtained using conventional PSO-optimized PI gain. The specification of the PMSM and PSO parameters is depicted in Table 1, and the sampling time considered for the simulation is 40 μ s.

In the first case, the reference speed is set to 157 rad/s, and the motor's transient behavior is recorded during startup. The speed response of the motor is shown in Fig. 3a where it can be seen that the modified PSO-PI has a speed overshoot of 0.433%, while the conventional PSO-PI has a speed overshoot of 0.445%. The different parameters such as rise time (t_r), peak time (t_p), settling time (t_s), and speed ripple are also examined to evaluate the performance of the PMSM drives. It is seen that the modified PSO-tuned PI controller-based PMSM drive system gives a more precise response and takes

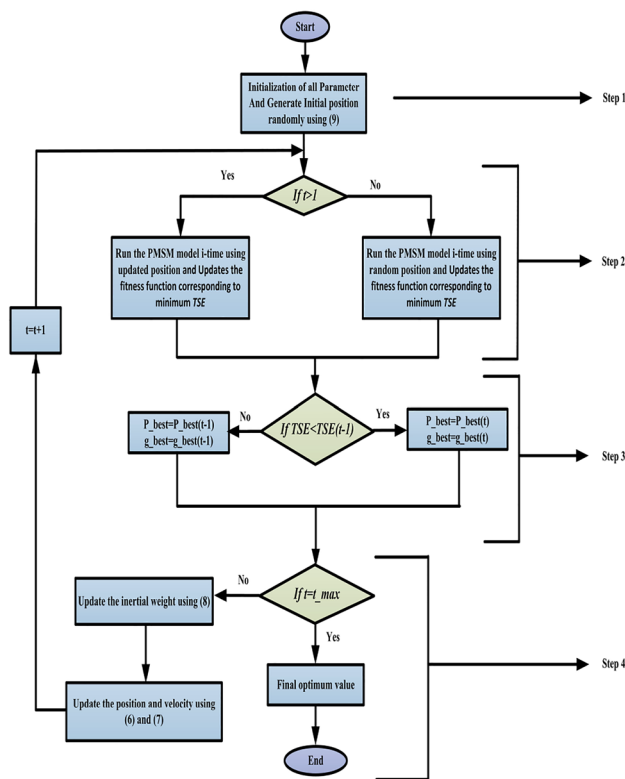


Fig. 2 Flow chart of the modified PSO

Table 1 Configuration of the PMSM and PSO algorithms

| Description of PMSM | Value | Unit | Description of PSO | Value | Symbol |
|---------------------|---------|-------------------|--------------------|-----------|--------------------|
| Rated power | 3.7 | kW | No. of particle | 15 | n |
| Rated current | 6 | A | No. of iteration | 20 | t_{max} |
| L_d & L_q | 28 | mH | Upper limit | 10 | U_L |
| rotor flux | 0.831 | Wb | Lower limit | 0 | L_L |
| Resistance (R) | 3 | Ω | Learning variables | 1.5 | c_1, c_2 |
| Motor inertia | 0.00198 | kg m ² | Inertial weight | [0.9 0.4] | w_{max}, w_{min} |
| Pole pair | 2 | | | | |
| Rated voltage | 415 | V | | | |

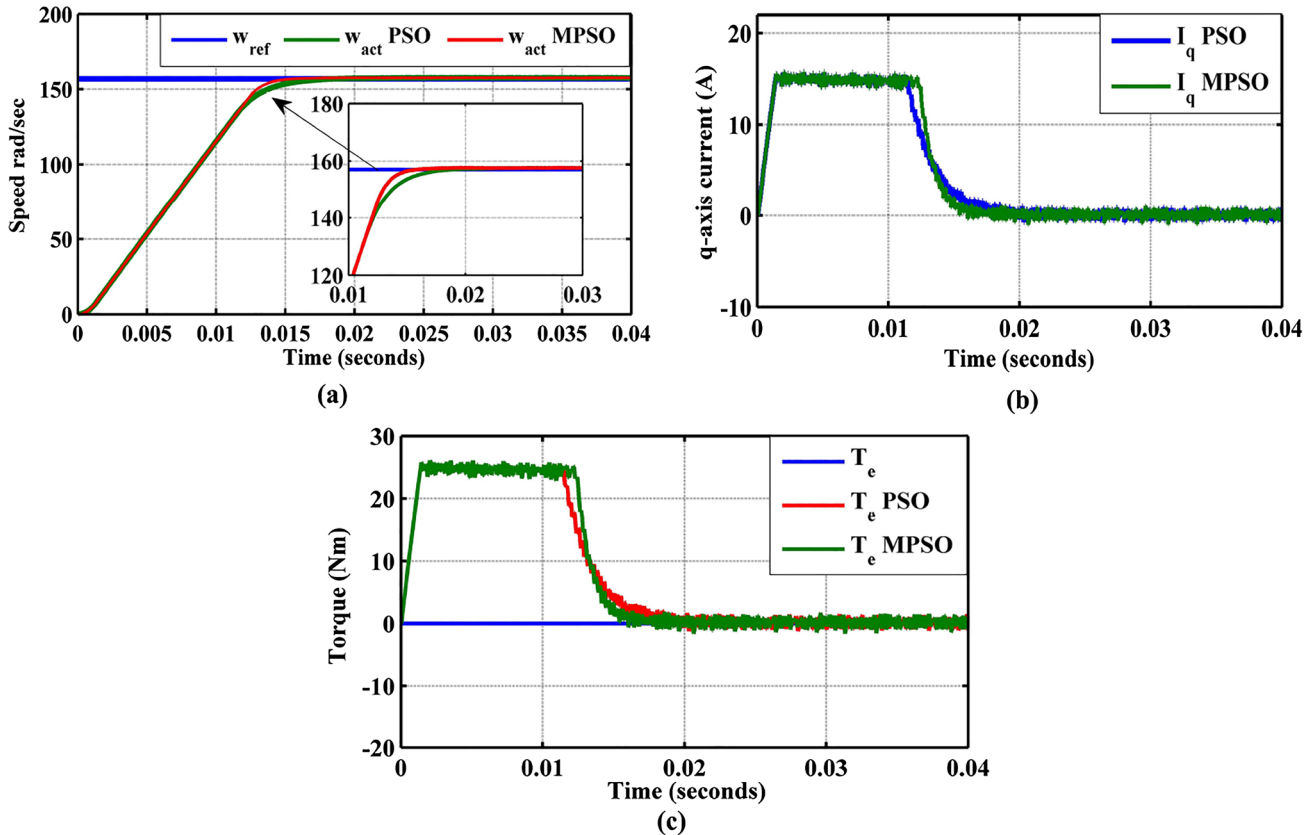


Fig. 3 Response of the motor at starting condition **a** speed, **b** q -axis current, **d** Electromagnetic torque

less time than the traditional PSO-tuned PI controller, as is shown in Table 2. As seen from Fig. 3b and c, initially, the stator draws a high current to produce the high electromagnetic torque to accelerate the motor speed to its reference speed quickly in both the drives.

In the second case, a sudden change in the reference speed command is given to each motor drive to test their reliability. The motor speed is initially set at 157 rad/s, then at $t = 0.5$ s, it is changed to 120 rad/s, and at $t = 2.5$ s, it is changed back to 157 rad/s. The response of the drives to the speed, q -axis current, and torque are shown in Fig. 4a–c. In this case, the modified PSO-PI controller

Table 2 Different parameter analysis at motor startup

| Parameters | Conventional PSO | Modified PSO |
|---------------------|------------------|--------------|
| t_s (s) | 0.025 | 0.02 |
| t_r (s) | 0.1224 | 0.1213 |
| t_p (s) | 0.0275 | 0.02 |
| Speed overshoot (%) | 0.445 | 0.433 |
| Speed ripple | 0.24 | 0.22 |

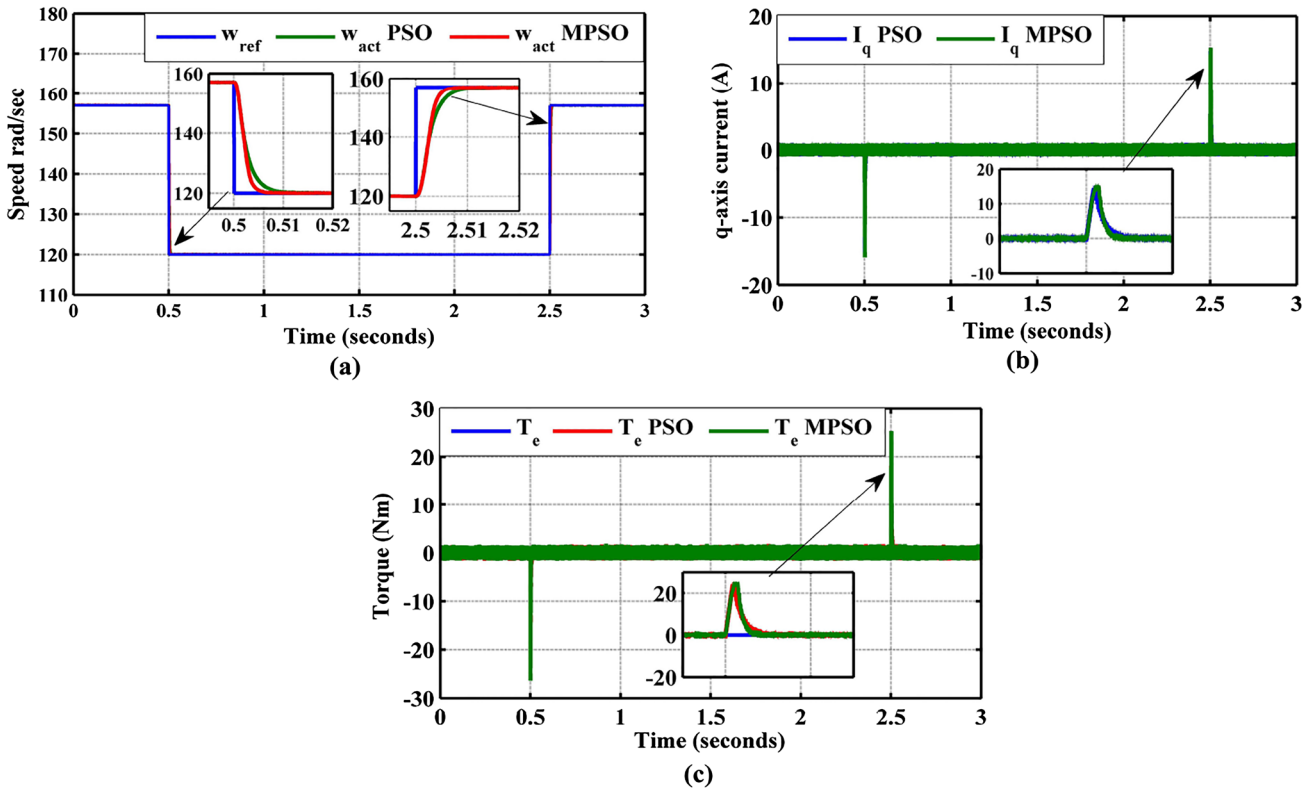


Fig. 4 Response of the motor at speed command change condition a speed, b *q*-axis current. c Electromagnetic torque

Table 3 Different parameter analysis at motor speed command change

| Parameters | Series conventional PSO | | Series modified PSO | |
|--------------|-------------------------|-------------|---------------------|-------------|
| | Falling edge | Rising edge | Falling edge | Rising edge |
| t_s (s) | 0.01 | 0.16 | 0.0055 | 0.0085 |
| t_r (s) | 0.0035 | 0.0046 | 0.0015 | 0.0031 |
| t_p (s) | 0.01 | 0.017 | 0.006 | 0.012 |
| Speed ripple | 0.2 | 0.34 | 0.18 | 0.2 |

outperforms the conventional PSO-PI controller for transients at $t = 0.5$ and 2.5 s, as shown by the speed response in Fig. 4a. Similar results are shown by the speed overshoot, rise time, peak time, settling time, and speed error, as in Case 1. When the speed command is changed to 120 rad/s and then back to 157 rad/s, the stator draws a large current and generates a large electromagnetic torque, as shown in Fig. 4b and c. This allows the rotor to survive the change in reference speed command quickly. Analysis of Fig. 4b and c show the modified PSO-PI PMSM drive responds to the command change quicker than the conventional PSO-PI PMSM drive as shown in Table 3.

In the third case, a 15 Nm load torque is applied to the motor drive at $t = 1$ s to examine the controller’s robustness. The motor’s speed response is depicted in Fig. 5a, where it can be observed that the speed response suffers from a speed dip when the load torque disturbances are applied. But compared to a traditional PSO-based speed controller, the proposed modified PSO-based speed controller works better under load torque disturbances and has less speed dip. Figure 5b and c show the stator current and electromagnetic torque responses, and it is clear that the current and thus electromagnetic torque rise to accelerate the motor to its reference speed when a load torque of 15 Nm is imposed.

Conclusions

In this paper, the PSO algorithm is modified to improve the speed response of the field-oriented vector control PMSM drive system. The algorithm is modified with an inertia weight updating mechanism to solve the premature convergence of the particle and give an optimum solution. The three different conditions are examined to analyze the performance of the motor. From the results, it can be said that the speed overshoot and speed ripple is less in the modified PSO-based PI controller compared to the conventional PSO.

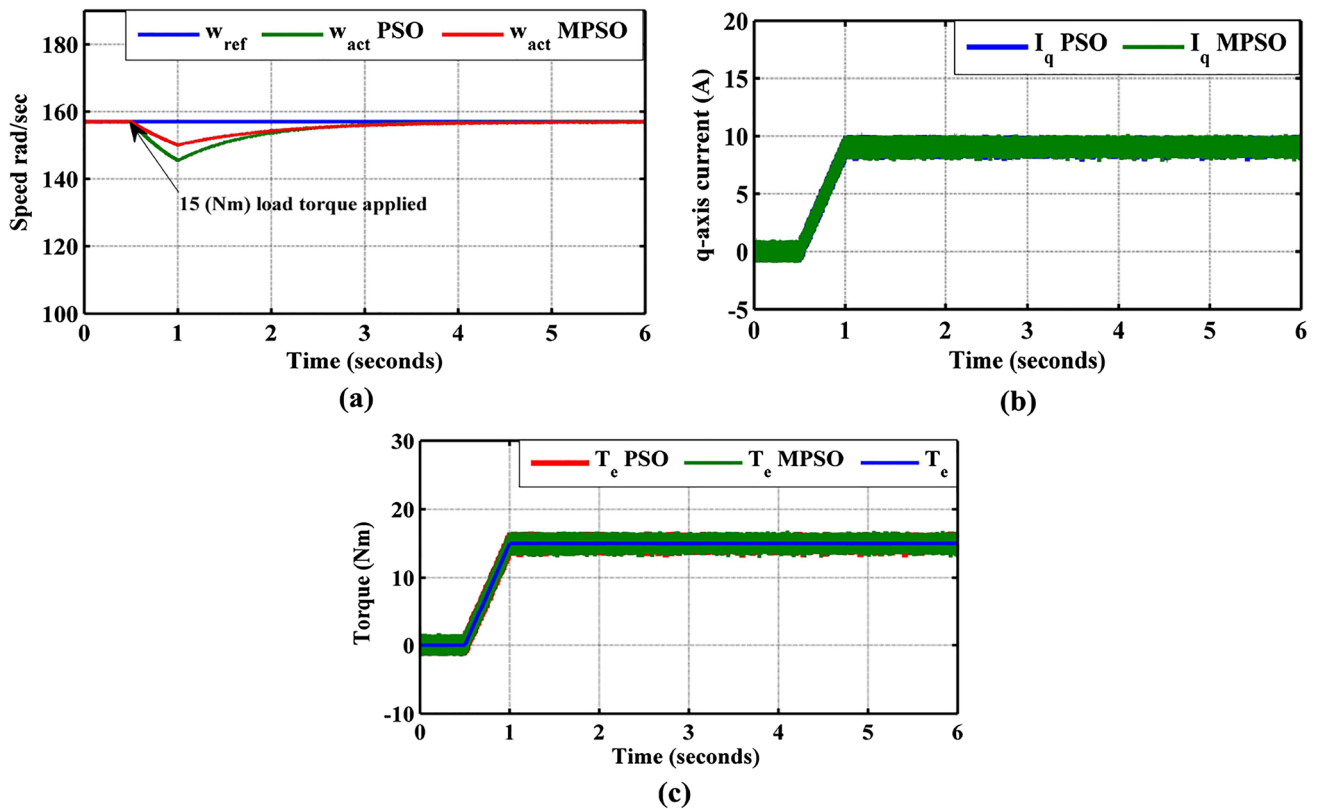


Fig. 5 Motor's response at a sudden load torque of 15 Nm impose **a** speed, **b** q -axis current, **c** electromagnetic torque

The speed response of the motor is better and faster in a modified PSO-PI-based PMSM drive. Therefore, it can be concluded that the modified PSO-tuned PI controller-based PMSM drive system gives a better response in comparison to the conventional PSO-tuned PI controller.

Declarations

Conflict of Interest The authors declare that they have no known conflict of interest.

Ethical Approval This article does not contain any studies with animals performed by any of the authors.

References

- Wang M. A novel mathematical nonlinear PMSM realization method for electric machine emulator. *IEEE J Emerg Sel Top Power Electron.* 2022;10(4):4171–81. <https://doi.org/10.1109/JESTPE.2022.3152429>.
- Iwama K, Noguchi T. High-efficiency drive method of adjustable field IPMSM utilizing magnetic saturation. *IEEE Access.* 2022;10:125499–508. <https://doi.org/10.1109/ACCESS.2022.3226335>.
- Wu J, Wang J, Gan C, Sun Q, Kong W. Efficiency optimization of PMSM drives using field-circuit coupled FEM for EV/HEV applications. *IEEE Access.* 2018;6:15192–201. <https://doi.org/10.1109/ACCESS.2018.2813987>.
- Wang Z, Chen J, Cheng M, Chau KT. Field-oriented control and direct torque control for paralleled VSIs fed PMSM drives with variable switching frequencies. *IEEE Trans Power Electron.* 2016;31(3):2417–28. <https://doi.org/10.1109/TPEL.2015.2437893>.
- Ullah Z, Kim K-T, Park J-K, Hur J. Comparative analysis of scalar and vector control drives of IPMSM under inter-turn fault condition considering nonlinearities. In: 2015 IEEE energy conversion congress and exposition (ECCE). 2015; IEEE, p. 366–72.
- Filho CJV, Xiao D, Vieira RP, Emadi A. Observers for high-speed sensorless PMSM drives: design methods, tuning challenges and future trends. *IEEE Access.* 2021;9:56397–415. <https://doi.org/10.1109/ACCESS.2021.3072360>.
- Hussain HA. Tuning and performance evaluation of 2DOF PI current controllers for PMSM drives. *IEEE Trans Transp Electr.* 2021;7(3):1401–14. <https://doi.org/10.1109/TTE.2020.3043853>.
- Mehta ND, Haque AM, Makwana MV. Modeling and simulation of P, PI and PID controller for speed control of DC Motor Drive. *Int J Sci Eng Res.* 2017;8(7):556–62.
- Jiang X, Wang Y, Dong J. Speed regulation method using genetic algorithm for dual three-phase permanent magnet synchronous motors. *CES Trans Electr Mach Syst.* 2023;7(2):171–8. <https://doi.org/10.30941/CESTEMS.2023.00013>.
- Fan Y, Chen J, Zhang Q, Cheng M. An improved inertia disturbance suppression method for PMSM based on disturbance observer and two-degree-of-freedom PI controller. *IEEE Trans Power Electron.* 2023;38(3):3590–9. <https://doi.org/10.1109/TPEL.2022.3218842>.

11. Kumar A, Khan YA, Verma V. Comparative evaluation of PSO, TLBO, JAYA, whale optimization, and grey Wolf optimization based tuning of PI controllers for vector controlled synchronous reluctance motor drive. In: 2022 IEEE IAS global conference on emerging technologies (GlobConET). 2022; IEEE. p. 261–66.
12. Gandhi R, Wilson R, Kumar A, Roy R. Comparative analysis of vector controlled PMSM drive with particle swarm optimization and ant colony optimization technique. In: 2020 international conference on computational performance evaluation (ComPE). 2020; IEEE. p. 744–50.
13. Lai C, et al. PMSM drive system efficiency optimization using a modified gradient descent algorithm with discretized search space. *IEEE Trans Transp Electr.* 2020;6(3):1104–14. <https://doi.org/10.1109/TTE.2020.3004463>.
14. Fodorean D, Idoumghar L, Brévilliers M, Minciunescu P, Irimia C. Hybrid differential evolution algorithm employed for the optimum design of a high-speed PMSM used for EV propulsion. *IEEE Trans Ind Electron.* 2017;64(12):9824–33. <https://doi.org/10.1109/TIE.2017.2701788>.
15. Song J, Zheng WX, 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(1):763–73. <https://doi.org/10.1109/TIE.2021.3050348>.
16. Wilson R, Gandhi R, Kumar A, Roy R. Optimized vector control strategy for dual-rotor axial flux permanent magnet synchronous motor for in-wheel electric drive applications. In: 2020 international conference on computational performance evaluation (ComPE). 2020; IEEE. p. 676–81.

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.