Electrical Engineering
https://doi.org/10.1007/s00202-024-02404-w

ORIGINAL PAPER

Speed control of PMSM using a fuzzy logic controller with deformed MFS tuned by a novel hybrid meta-heuristic algorithm

Tahar Nouaoui[1,2] · Abdelhakim Dendouga[2,3] · Abdelmalik Bendaikha[1,2]

Received: 17 November 2023 / Accepted: 2 April 2024

Abstract

The goal of this study is to design a very robust and efficient standard fuzzy logic controller (FC) with tuned and deformed membership functions (FC-TMFS) using a new hybrid algorithm for speed control of the permanent magnet synchronous motor. The proposed algorithm is a partial combination of three algorithms, which are sewing trainee-based optimization (STBO), particle swarm optimization (PSO), and symbiotic organism search (SOS). To demonstrate the superiority of the proposed algorithm (ST-PS-SO), FC-TMFS optimal parameters obtained using ST-PS-SO were compared to the optimal parameters achieved when using STBO, PSO, and SOS. In terms of the system performance, and the fitness of the global optimum parameters, ST-PS-SO convincingly out-competed all other algorithms. In addition to that, through MATLAB/SIMULINK simulation, ST-PSO-SOS-based FC-TMFS was compared to the classical proportional–integral (PI), and PI–derivative (PID) controllers, as well as the conventional FC with symmetric and untuned MFS, based on the results of speed tracking, torque induction, and robustness test, FC-TMFS convincingly outperformed all other controllers in every aspect by a very significant margin. Finally, processor-in-loop (PIL) implementation was also performed to validate and prove the importance and functionality of the designed FC-TMFS, as well as the capability of the proposed hybrid algorithm.

Keywords Field oriented control · Permanent magnet synchronous motor · Meta-heuristic algorithm · Space vector pulse width modulation · Standard fuzzy logic controller · Processor in loop

1 Introduction

Recently, the use of the permanent magnet synchronous motor (PMSM) in various electrical applications has increased exponentially compared to the induction motor (IM) and the direct current (DC) motor [1], that's got to do with its robust and simple model compared to the IM, and its advantageous viability and low-cost maintenance compared to the DC motor; moreover, the diversity of controlling techniques of the PMSM makes it the more preferable choice for most systems [2, 3]. Each control technique is designed for a specific objective; however, when it comes to the overall performance, whether its speed and position tracking, or traction and transportation, field oriented control (FOC) with a non-classical speed controller is mostly the best choice [4].

FOC can be executed using three main controllers [5], two for the direct and quadratic axis currents regulation, which are usually a classical proportional–integral (PI) controllers, and one for the mechanical speed regulation, the speed controller definitely got the most impact on the performance of FOC [6], non-classical controllers such as sliding mode controller [6], fuzzy-PI controller [7], and standard fuzzy logic controller (FC) [8] are very recommended over classical controllers such as PI and proportional–integral–derivative (PID) controllers, especially for speed regulation of PMSM.

It is no doubt that the fuzzy logic controller (FC) has become one of the most promising and effectual controllers in the field of electronic and electrical engineering [9–12], due

A. Dendouga, A. Bendaikha have contributed equally to this work.

✉ Tahar Nouaoui
tahar.nouaoui@univ-biskra.dz

Abdelhakim Dendouga
a.dendouga@univ-biskra.dz

Abdelmalik Bendaikha
abdlmalik.bendaikha@univ-msila.dz

1 Laboratory of identification, command, control and communication, University of Biskra, Biskra, Algeria

2 Electrical engineering department, University of Biskra, Biskra, Algeria

3 Electrical engineering department, University of M'sila, M'sila, Algeria

Published online: 30 April 2024

to its adaptive functioning and flexible model, the FC does not require thorough tuning to deliver an acceptable outcome; however, when tuned properly by optimizing its membership functions (MFS), and its inputs and outputs factors using an optimization algorithm [13], it can significantly surpass its standard performance.

In the past few years, optimization algorithms capability of solving complicated mathematical problems of nonlinear systems has been validated repeatedly, Yousri D, Allam D, and Eteiba M performed parameters identification of fractional order models of PMSM using a set of chaotic meta-heuristic algorithms [14]. In [15], Aguilar-Mejia O, Minor-Popocatl H, and Tapia-Olvera R published a comparison and ranking of a wide range of meta-heuristic algorithms, for tuning of PI controllers in a machine drive systems. Optimization algorithms performance varies based on the task whether its gains tuning, or parameters identification, etc., and the number of system parameters that needs solving. Particle swarm optimization (PSO), for example, is one of the most used meta-heuristic algorithms [16], it offers fast optimization with good solutions, but it struggles with systems that have too many knobs to adjust, where some algorithms like symbiotic organisms search(SOS) can provide better results [5, 17]. In the case of complicated systems, algorithms with intense exploration and fast convergence like sewing trainee-based optimization (STBO) [18] are way more competent than PSO. In [18], STBO managed to outperform many capable algorithms such as gray wolf optimizer (GWO) [19], whale optimization algorithm (WOA) [20], and genetic algorithm (GA) [21].

Many researchers have implemented meta-heuristic algorithms for the tuning of FC factors and MFS.

Quantum-behaved lightning search algorithm has been implemented for the tuning of a fuzzy-PI controller for indirect FOC of induction motor [22]. In [7], a hybrid fuzzy-PI controller with tuned MFS was used for advanced control of PMSM.

FC forms and implementation methods are very diverse [23–25]. In [23], a fractional order adaptive fuzzy backstepping controller was used for speed control of PMSM. In [24], fuzzy-based and multivariable optimization approach robust control of PMSM was performed. However, the simplest and most common fuzzy speed controllers of PMSM are the standard FC with a single direct output, and the hybrid fuzzy-PI controller where the fuzzy part provides two outputs consisting of adaptive proportional and integral gains that are supplied to the PI part, which provides the controller's output. Since the fuzzy part of fuzzy-PI controller has more outputs, its inference system, tuning difficulty, and level of complexity are way higher than the standard FC with one output.

In the literature, many papers have used the standard FC for speed control of PMSM [8, 26–29].

In [26–28], authors of these papers performed similar research comparing the standard FC against the conventional PI controller. In [29], a comparison of the dynamic response under load disturbances of PMSM using the standard FC and the conventional PI and PID controllers was simulated. In these research papers, the standard FC provided faster response time with better robustness; however, none of them provided an approach to design and tune the FC inputs and output scaling factors.

In [8], AliSkan I and Unsal S tested the speed control performances of the standard FC having different membership functions and inference methods.

The major drawbacks of these studies are the untested robustness against internal disturbances such as motor parameters changes, and the lack of MFS tuning since the full potential of the standard FC can only be achieved with deformed and precisely tuned MFS [13, 22, 30].

In [31], a hybrid algorithm was proposed for the tuning of the output MFS of the standard FC, the main focus of this paper was the proposed algorithm and not the performance of the FC since the output of the controller was saturated using MATLAB/SIMULINK saturation block, which limits the controller's output, and does not showcase its real performance; moreover, the speed tracking was so poor because it delivered a very high steady-state error despite the absence of high external disturbances, tracking of a speed reference higher than the rated speed, or any parameters changes, which could be a result of poorly tuned output MFS, or the untuned inputs MFS.

Thus, this paper main contribution is the design and development of a robust and a very efficient standard FC with thoroughly tuned inputs and output membership functions (FC-TMFS) for speed control of PMSM. In order to determine the best possible parameters for the aforementioned controller, a hybrid algorithm called (ST-PS-SO) was proposed, to investigate the efficiency of the proposed algorithm, when simulating FOC of PMSM, ST-PS-SO-based FC-TMFS (SPS-FC-TMFS) was compared to PSO-based FC-TMFS (PSO-FC-TMFS), STBO-based FC-TMFS (STBO-FC-TMFS), and SOS-based FC-TMFS (SOS-FC-TMFS), with the task if minimizing a specific objective function(OBJ). Finally, running a MATLAB/SIMULINK simulation of FOC of PMSM, FC-TMFS with the best-obtained parameters was compared to the conventional FC, PI, and PID controllers, the simulation included a wide range of speed and load charge references, as well as a massive change in the system parameters to examine the robustness of the designed controller. Processor-in-loop (PIL) simulation was also applied to confirm the integrity of the designed controller.

2 Modeling of the PMSM

Using the dq0 frame reference, equations of PMSM model can be written as in [(1),(2),(3),(4)].

$$v_d = Ri_d - L_d\frac{\mathrm{d}i_d}{\mathrm{d}t} - \omega L_q I_q \quad (1)$$

$$v_q = Ri_q - L_q\frac{\mathrm{d}i_q}{\mathrm{d}t} - \omega L_d I_d + \omega\phi \quad (2)$$

$$T_e = \frac{3}{2}P_\mathrm{n}[(L_d - L_q)i_d i_q + \phi i_q] \quad (3)$$

$$j\frac{\mathrm{d}\omega_r}{\mathrm{d}t} = T_e - T_l - B\omega_r \quad (4)$$

where v_d and v_q represent the voltages, T_e and i_q represent the currents. L_q and L_d are the stator inductances, ω is the electrical velocity, ω_r is the mechanical velocity, P_n is the number of pair of poles, ϕ is the flux induced by the permanent magnet, T_e is the electromagnetic torque, T_l is the load torque, B is the motor viscous friction, and J is the motor inertia. The output of the FC is the electromagnetic torque T_e' required to deliver the targeted speed, after multiplying it by $Kt = \frac{2}{3(P_\mathrm{n}\times\phi))}$. As in (5) and Fig. 1, we get the quadratic axis current reference i_q'.

$$i_q' = T_e' \times Kt \quad (5)$$

Figures 1 and 2 present FOC and PMSM MATLAB/SIMULINK models, respectively.

3 Field oriented control of PMSM

FOC is a vector control technique used to command and drive AC motors, compared to DTC, FOC is more complicated because of its extra two controllers, but it compensates that by delivering a better and more precise current control, and way less electromagnetic torque ripples; moreover, with the introduction of optimization algorithms, the complexity of FOC became less of a concern since the tuning of its controllers parameters became way easier. The goal of FOC is to simplify the control of PMSM by severing the link between the direct and quadratic axis voltages, in order to do that, a decoupling block is used as in Fig. 1 and [32], along with a closed loop control of i_d to drive it into a value very close to zero, hence making the induced electromagnetic torque linear with i_q, and easier to control. With $i_d \simeq 0$, (3) could potentially be reduced to (6):

$$T_e \simeq \frac{3}{2}P_\mathrm{n}(\phi i_q) \quad (6)$$

Figure 1 illustrates in detail the implementation of FOC of PMSM in MATLAB/SIMULINK software.

4 Hybrid ST-PS-SO optimization algorithm

Optimization algorithms are tools designed to solve mathematical and physical problems, either by minimizing or maximizing a certain OBJ, with the aim of finding an optimal solution, through exploring and exploiting a bounded interval of possible solutions. Since each algorithm has its distinctive approach of exploiting and exploring, their performance varies a lot based on the problem structure, for example, PSO is a heavy exploiting algorithm, SOS is a heavy exploring algorithm, and STBO is somewhere in between. ST-PS-SO is a hybrid three-phase algorithm specifically built for the tuning of FC-TMFS, for speed control of PMSM, each phase operates based on one of the three parent algorithms,

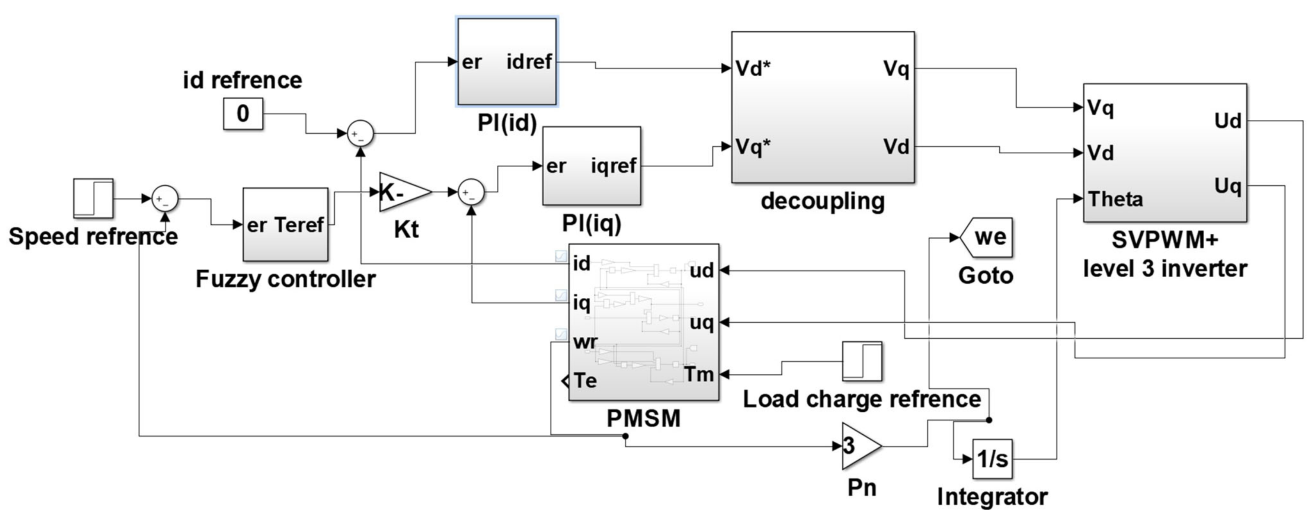


Fig. 1 FOC MATLAB/SIMULINK representation

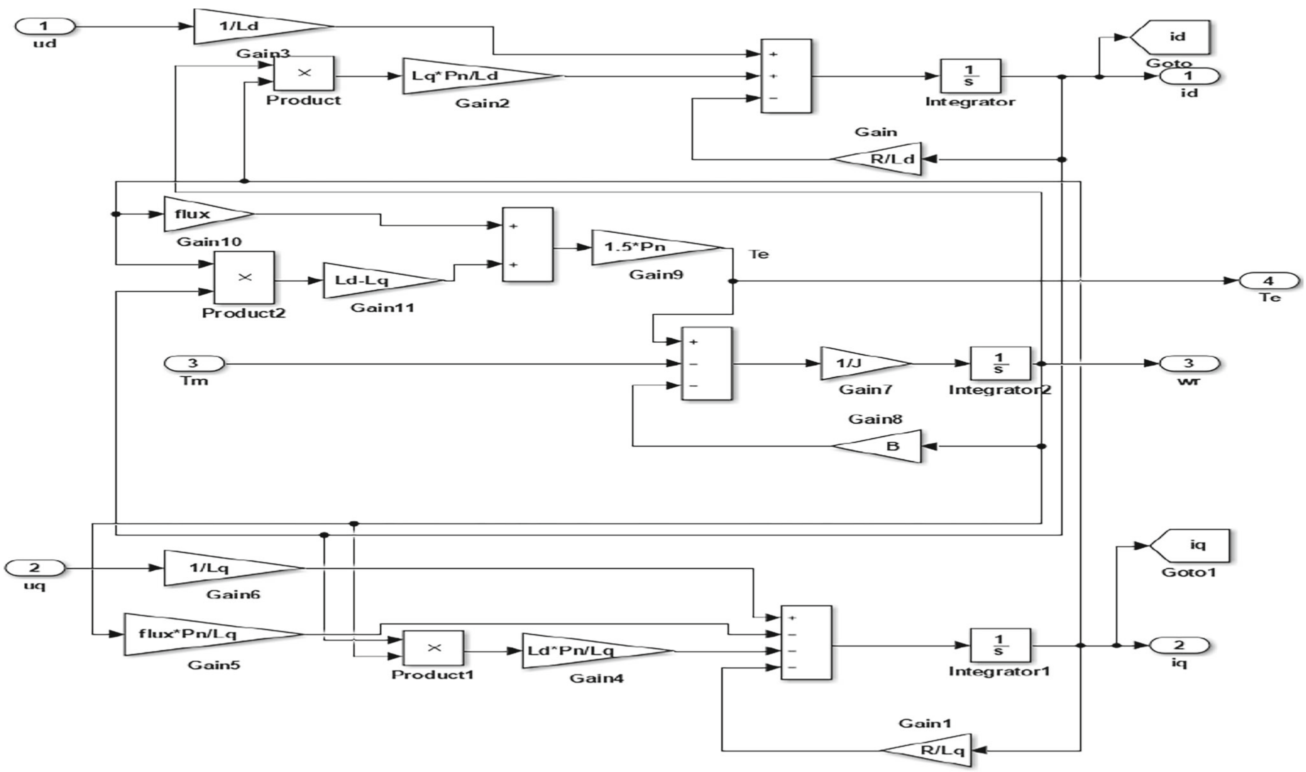


Fig. 2 PMSM SIMULINK model

STBO, PSO, and SOS. Any optimization algorithm can only be initiated by generating a specific number of search agents denoted by $X_{i....N_s}$, where i is the current search agent, and N_s is the amount of search agents, each search agent begins with a random matrix of a possible solution, which is updated every iteration through mathematical equations, the algorithm keeps updating solutions until a certain criteria is met, or until it completes the allowed number of iterations. Finally, a global optimum is deduced from the search agent with the best fitness; in the case of FOC of PMSM, the smaller the OBJ is the better is the fitness.

4.1 ST-PS-SO phases

This section presents the phases of ST-PS-SO in detail.

4.1.1 Phase I

Phase one of ST-PS-SO is a guided exploration that utilizes the first phase of STBO [18], which is called the training phase. In this phase, the current search agent X_i is called a trainee, a set of search agents with better fitness than X_i is established and named S, each agent from S represents a possible instructor for X_i, then a random instructor X_j from S is chosen to help X_i improve its fitness using (7).

$$X_i^P = X_i + r_i \times (X_j - I_i \times X_i) \tag{7}$$

where I_i is either 1 or 2, r_i is a random number equals or between 0 and 1, X_i^P is the potential new solution for X_i depending on:

$$X_i = \begin{cases} X_i^P, & \text{if } F_i^P < F_i \\ X_i, & \text{otherwise} \end{cases}$$

where F_i^P is the new candidate fitness value and F_i is X_i fitness value.

4.1.2 Phase II

Phase two of ST-PS-SO is a partial exploration that imitates the third phase of SOS [5] which is called parasitism phase. Instead of updating the whole matrix of the search agent X_i, parasitism phase replaces the value of one of the dimensions of X_i randomly, by a new random value from the allowed interval; however, in this system, parasitism phase is set to only update the parameter of the output gain factor denoted by $X_{i\text{OUT}}$ since it is the hardest parameter to tune due to its wide interval of possible values. $X_{i\text{OUT}}$ updates based on:

$$X_{i\text{OUT}} = \begin{cases} X_{i\text{OUT}}^P, & \text{if } F_{i\text{OUT}}^P < F_{i\text{OUT}} \\ X_{i\text{OUT}}, & \text{otherwise} \end{cases}$$

4.1.3 Phase III

Phase three of ST-PS-SO is a fast exploit phase mimics the whole algorithm of PSO [16], PSO uses the memory of each search agent to perform a fast and efficient convergence toward a global optimum using (8), (9), and (10)

$$V_i^P = r_1 \times V_i \times \omega + c_1 \times r_2 \times (X_i^{\text{Pbest}} - X_i) + c_2 \times r_3 \times (X_i^{\text{Gbest}} - X_i) \quad (8)$$

$$\omega = \omega\text{Max} - t \times ((\omega\text{Max} - \omega\text{Min})/\text{Maxiter}) \quad (9)$$

$$X_i^P = X_i + V_i^P \quad (10)$$

where r_1, r_2, and r_3 are random numbers from [0–1], t is the current iteration, Maxiter is the maximum number of iterations, $\omega\text{Max} = 0.9$, $\omega\text{Min} = 0.4$, and $c_1 = c_2 = 2$. X_i^{Pbest} represents the parameters of the best fitness achieved by X_i, and X_i^{Gbest} represents the best fitness achieved by the whole population of search agents so far, V_i is the previous iteration velocity of the current search agent, and V_i^P and X_i^P are the new velocity and position of X_i, respectively.

4.2 ST-PS-SO algorithm

This section describes the workflow of ST-PS-SO in the following pseudo-code (1).

Algorithm 1 ST-PSO-SOS pseudo-code

Begin
assign the parameters of [ωMax, ωMin, $Maxiter$, N_s].
assign the number of parameters of search agents $Npar$.
assign the maximum and minimum values of search space $Bn = [VarMin - VarHigh]$.
generate search agents population $i = 1, 2, ...N_s$.
assign random parameters from the interval Bn to all search agents from .
assign zeros to all initial velocities of all search agents.
assign a random X_i^{Gbest} and calculate its fitness F_i^{Gbest}.
calculate all fitness values of search agents and assign X_i^{Pbest} and $F_i^P best$.
initialize Main loop.
While $t < Maxiter$
For $i = 1 : N_s$
initialize Phase I
assign S and X_j
calculate X_i^P using $X_i^P = X_i + r_i \times (X_j - I_i \times X_i)$ then calculate F_i^P
update X_i,$X_i^P best$, and X_i^{Gbest}
initialize Phase II
assign a random new $X_{i_{OUT}}$ for X_i to get X_i^P then calculate F_i^P
update X_i,$X_i^P best$, and X_i^{Gbest}
initialize Phase III
calculate V_i^P using $V_i^P = r_1 \times V_i \times \omega + c_1 \times r_2 \times (X_i^{Pbest} - X_i) + c_2 \times r_3 \times (X_i^{Gbest} - X_i)$
calculate X_i^P using $X_i^P = X_i + V_i^P$
update X_i where $X_i = X_i^P$
update $X_i^P best$, and X_i^{Gbest}
End For
update t where $t = t + 1$
update ω using $\omega = \omega Max - t \times ((\omega Max - \omega Min)/Maxiter)$
End While
End

5 ST-PS-SO-based FC-TMFS

The special thing about the standard FC is its simplicity compared to other types of fuzzy controllers, and its robustness and capability of handling linear and nonlinear systems, without the need of an exact identification of the system parameters [33]. Fuzzy controllers can determine their outputs based on the assigned inputs and inference system, and a set of linguistic rules that are inspired from humans way of thinking [34].

5.1 Design of SPS-FC-TMFS for speed control of PMSM

For speed control of PMSM, SPS-FC-TMFS was based on MAMDANI inference system. SPS-FC-TMFS has two inputs and one output, each has their own shape of deformed and tuned MFS. The MFS of the inputs and outputs of SPS-FC-TMFS are depicted in Figs. 3, 4 and 5.

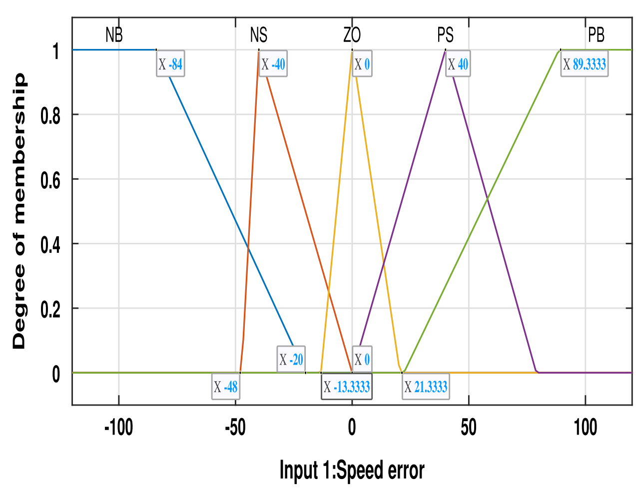


Fig. 3 First input of SPS-FC-TMFS: speed error

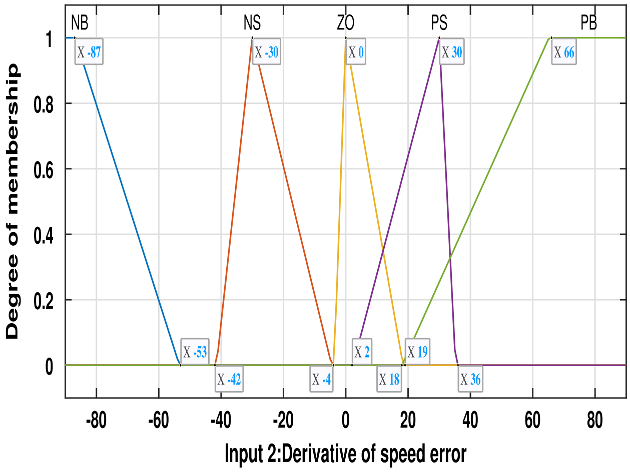


Fig. 4 Second input of SPS-FC-TMFS: derivative of speed error

From here on the standard conventional fuzzy controller will be addressed by FC, and the standard fuzzy controller with tuned membership functions will be addressed by FC-TMFS.

The linguistic rules of SPS-FC-TMFS are presented in Table 1. Figure 6 depicts MATLAB/SIMULINK representation of the standard fuzzy speed controller.

5.2 Process of tuning SPS-FC-TMFS for speed control of PMSM

In this section, the performances of SP-PS-SO, STBO, PSO, and SOS, when tuning FC-TMFS for speed control of PMSM were compared, the selected OBJ is represented in (11), based on [5]. The amount of search agents is 50, and iterations count is 200.

$$\mathrm{OBJ} = \int 2\times \mid (\mathrm{error}_{\mathrm{speed}}) \mid + \int t\times \mid (\mathrm{error}_{i_d}) \mid + \int t\times \mid (\mathrm{error}_{i_q}) \mid \quad (11)$$

Note that the inputs of the FC-TMFS were normalized using MATLAB/SIMULINK saturation blocks, where input one and input two are limited within the intervals [-120; 120] and [-0.292; 0.292], respectively, the output was not saturated. PI controllers of the direct and quadratic axis currents were pre-tuned. The parameters of the tested PMSM are presented in Table 2.

Table 1 The rules set of SPS-FC-TMFS

Control signal		Input 1				
		NB	NS	ZO	PS	PB
Input 2	NB	NB	NB	NB	NS	ZO
	NS	NB	NB	NS	ZO	PS
	ZO	NB	NS	ZO	PS	PB
	PS	NS	ZO	PS	PB	PB
	PB	ZO	PS	PB	PB	PB

The computational complexities of the update processes of SP-PS-SO, STBO, PSO, and SOS are O(3NmT), O(3NmT), O(NmT), and O(4NmT), respectively, where N is N_s, m is the number of the variables, and T is the iterations count. In order to balance the scales, PSO was run with three times the search agents count of other algorithms, which makes O(3NmT) its new computational complexity.

The progress of minimizing OBJ is depicted in Fig. 5, where we can clearly see that SP-PS-SO is the most successful when tuning FC-TMFS.

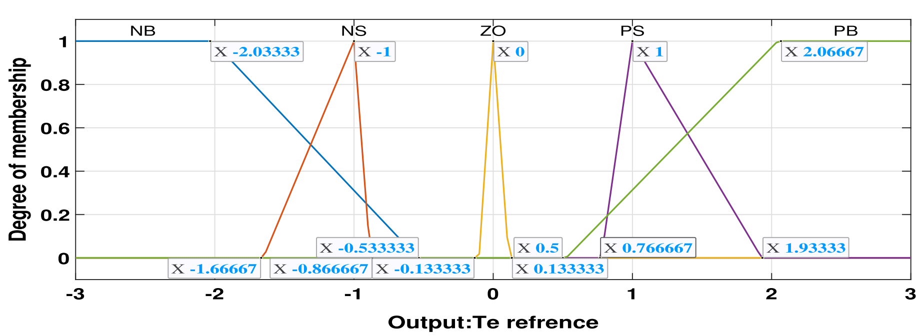
Table 3 shows N_s and the global optimum OBJ of each controller.

5.3 Performance of SPS-FC-TMFS compared to other algorithms-based controllers

In order to inspect and evaluate the performance of FC-TMFS obtained optimal parameters of each algorithm, 4 s of MATLAB simulation of FOC of PMSM was run with a sudden increase in the load charge, and multiple changes applied to the speed reference point. Table 4 presents the speed and load charge values during the simulation.

Figure 8 depicts the speed curves achieved in the simulation, where we can see that SPS-FC-TMFS have executed a very precise speed tracking with 0% overshoot. In Fig. 8a, d, and e, STBO-FC-TMFS and SOS-FC-TMFS performed similarly with fast response time but with a slight 0.5% over-

Fig. 5 Output of SPS-FC-TMFS: T_e reference



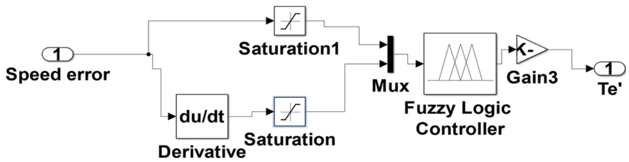


Fig. 6 MATLAB/SIMULINK fuzzy speed controller

Table 2 Parameters of the motor

Parameter	Value
Nominal power Pw (Kw)	1.5
R (Ω)	1.4
L_d (H)	0.0066
L_q (H)	0.0058
J (kg.m^2)	0.00176
Flux (ϕ)	0.1546
P_n	3
B (Nm/rd)	0.00038

Table 3 Tuning performance and the size of N_s of each algorithm

Algorithm	Final OBJ	N_s
SP-PS-SO	**980005**	50
PSO	1000012	150
STBO	990310	50
SOS	990666	50

The best results achieved at that specific test or duration are highlighted in bold

shoot after every sudden change in speed reference, whereas PSO-FC-TMFS performed poorly and could not handle the sudden changes in the speed setpoint, resulting in a very significant steady-state error, especially after the increase in the load torque, which is highlighted in Fig. 8b.

In Fig. 9, after analyzing the induced T_e of each controller, we can see a clear win for SPS-FC-TMFS, since it delivered the most accurate load charge tracking and overshoot suppression, STBO-FC-TMF and SOS-FC-TMFS came second with relatively similar results, and PSO-FC-TMFS could not provide a satisfactory and decent results due to its slow response time.

5.4 Performance of FC-TMFS compared to conventional FC, and the classical PID and PI controllers

This section presents a very comprehensive and intense simulated comparison between FC-TMFS, the conventional FC, and the classical PID and PI controllers. The conventional FC was tuned using ST-PS-SO algorithm, FC inputs scaling is identical to FC-TMFS; however, its output factor and the shape of its MFS are completely different than the ones of FC-TMFS. Tables 5 and 6 present the parameters of each speed controller, along with the parameters of the used currents controllers.

Note that all controllers were tuned with the aim to minimize the response and settling time, and suppress the overshoot of speed tracking as much as possible.

5.4.1 Variable speed reference with no load torque

After applying a variety of speed references, we got the results of response time and overshoot percentage of each controller in Tables 7 and 8.

In Fig. 10, as expected, after every sudden change of the speed reference, the classical PI controller responded poorly with a massive overshoot and undershoot [35], the classical PID managed to lessen the overshoot but at the cost of a very critical kick-back effect [36].

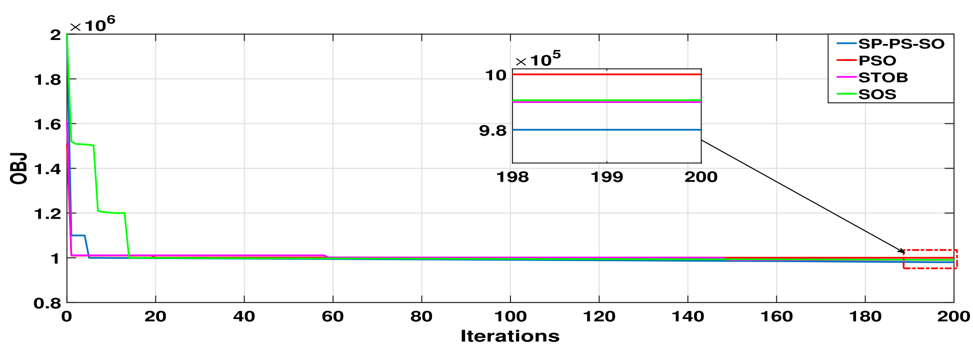


Fig. 7 Progress of tuning of each algorithm

Table 4 Speed and load charge references during the simulation

	0–0.5(s)	0.5–2(s)	2–2.5(s)	2.5–3(s)	3–3.5(s)	3.5–4(s)
Speed(red/s)	100	100	100	−100	50	150
Load charge(N.m)	0	4	0	0	0	0

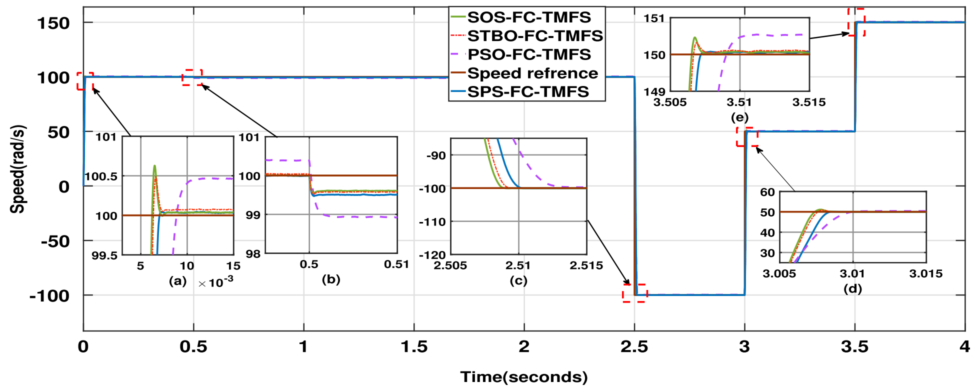


Fig. 8 Speed tracking performance of each controller

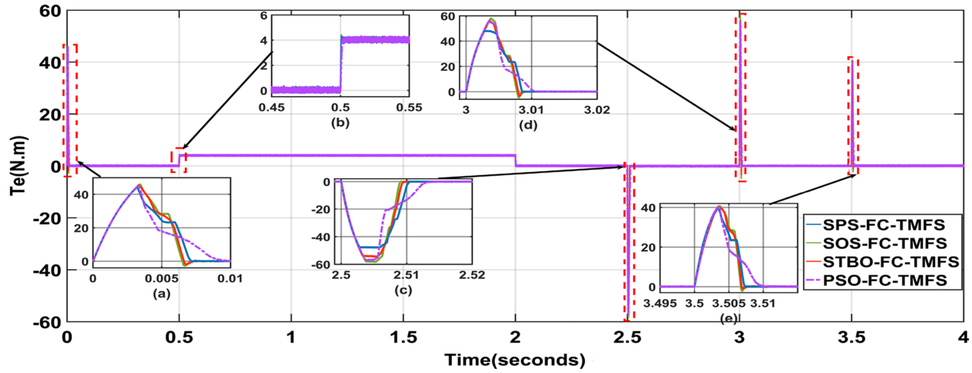


Fig. 9 Induced torque during the change of load charge and speed reference

Table 5 Parameters of FC-TMFS and the conventional FC

	Output gain	Kp(iq)	Ki(iq)	Kp(id)	Ki(id)
Conventional FC	84.3	61.96	143	79.5	54.46
FC-TMFS	23.66	61.96	143	79.5	54.46

Although the conventional FC managed to partially suppress the overshoot, it still had some flaws in terms of settling time. FC-TMFS, however, delivered a flawless speed tracking performance, outmatching all other controllers with faster settling time, and way better overshoot rejection, which is highlighted in Fig. 10a–d, and in Tables 7 and 8.

5.4.2 Fixed speed reference with a variable load torque

Table 9 presents the steady-state error of speed tracking in accordance with every increase in the load charge.

Figure 11 presents the speed tracking achieved with every controller under various load charges. Based on Fig. 11a, b and c, FC-TMFS managed to deliver the best overall results, which is due to its fine-tuned MFS and output gain. The conventional FC could not keep a tight steady state error, mainly because of its symmetric and un-optimized MFS. The classical PID performance was close to FC-TMFS; however, it still showed a significant kick-back effect but only in the case of small load charges. Similarly to the conventional FC, the classical PI also could not effectively handle the increasing value of the load charge, which is mainly because it was tuned to suppress the overshoot, and respond as fast as possible, but because it lacks a derivative component, it cannot deliver a precise speed tracking, and simultaneously handle the introduced load charges.

5.4.3 Fixed speed reference and load charge with a significant increase in the motor parameters

In order to further verify the robustness of the designed controller, an analysis of speed control of PMSM with a fixed

Table 6 Parameters of the classical PID and PI controllers

	Kp(speed)	Ki(speed)	Kd(speed)	Kp(iq)	Ki(iq)	Kd(iq)	Kp(id)	Ki(id)	Kd(id)
PID	5.77	1.2	0.0019	220	105	0.025	86.5	69.1	0.033
PI	3.63	0.062	–	266	5.57	–	73.8	51.2	–

Table 7 Settling time

Duration	Speed reference	Settling time			
		FC-TMFS	FC	PID	PI
0–1	100	**0.006**	0.008	**0.006**	0.007
1–2	−100	**0.011**	0.019	0.015	0.014
2–3	150	**0.012**	0.013	0.012	0.015
3–4	0	**0.008**	0.01	0.01	0.011

The best results achieved at that specific test or duration are highlighted in bold

Table 8 Overshoot percentage

Duration	Speed reference	Overshoot			
		FC-TMFS	FC	PID	PI
0–1	100	**0%**	**0%**	**0%**	6%
1–2	−100	**0%**	2.5%	5%	10%
2–3	150	**0%**	1.52%	1.12%	3.6%
3–4	0	**0%**	4%	5.06%	10.6%

The best results achieved at that specific test or duration are highlighted in bold

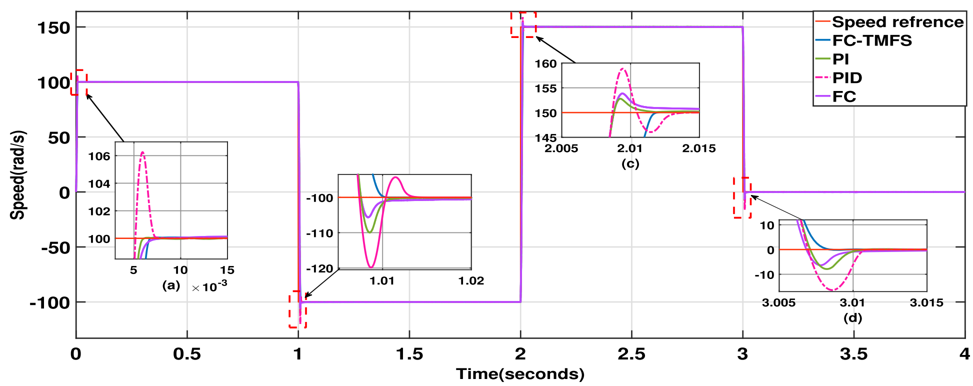


Fig. 10 Speed tracking performance with a fixed load charge and a variable speed reference

Table 9 Speed tracking steady-state error under variable charge load

Duration	Load charge	Steady-state error			
		FC-TMFS	FC	PID	PI
0-0.5	1	**0.11**	0.112	0.18	0.27
0.5-1.5	3	**0.38**	0.89	0.47	0.82
1.5–2	5	0.69	1.6	**0.68**	1.36

The best results achieved at that specific test or duration are highlighted in bold

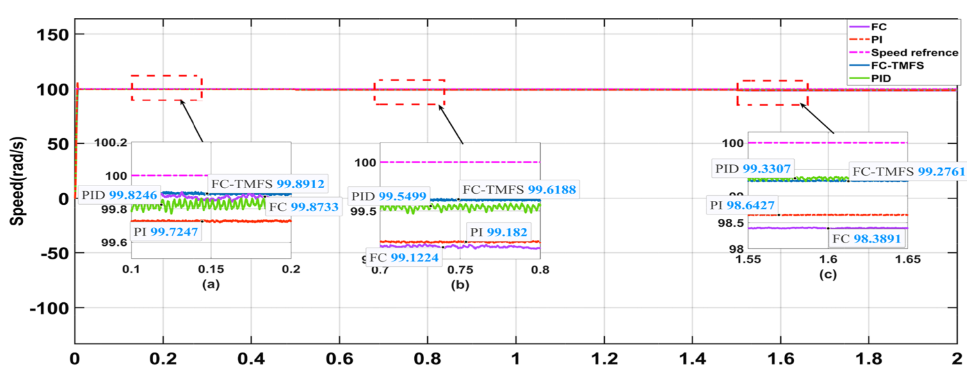


Fig. 11 Speed tracking performance with a variable load charge and a fixed speed reference

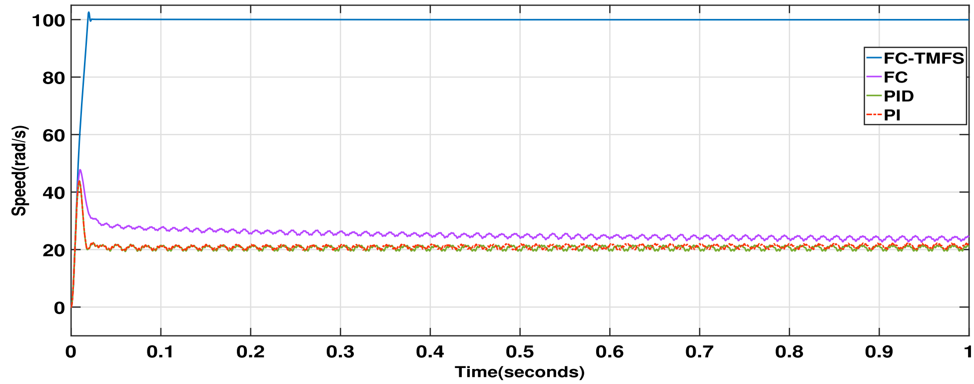


Fig. 12 Speed tracking performance after significantly increasing the values of L_q, L_d, and R

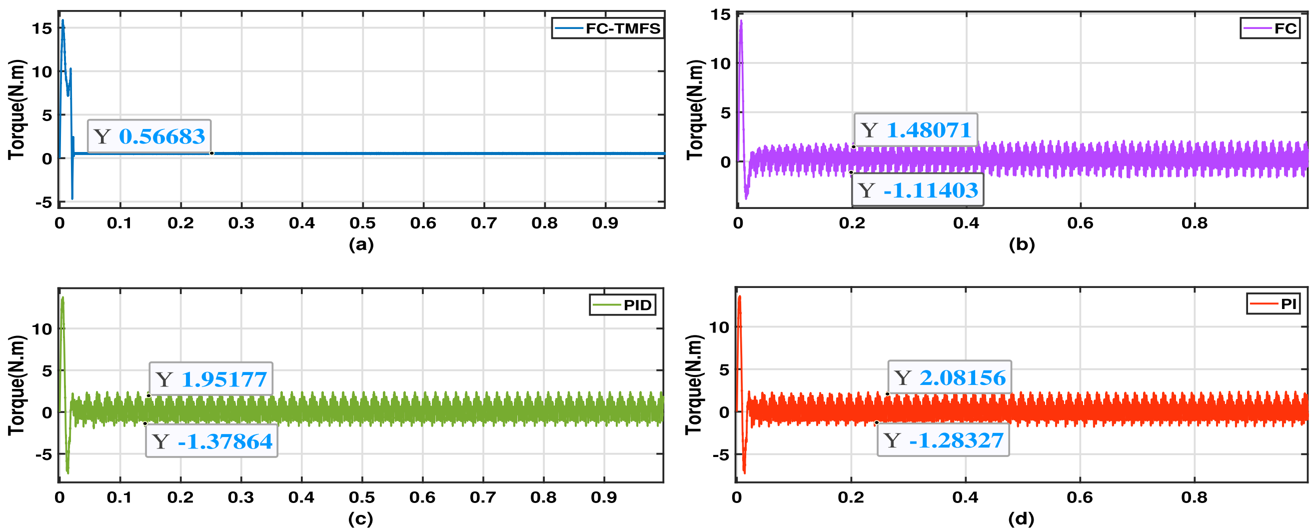


Fig. 13 The produced torque after significantly increasing the values of L_q, L_d, and R. **a** FC-TMFS produced T_e. **b** The conventional FC produced T_e. **c** PID produced T_e. **d** PI produced T_e

speed reference point of 100rad/s, fixed load torque of 0.5N, and a massive increase in the motor internal parameters was simulated using FC-TMFS, the conventional FC, PI, and PID individually.

By increasing the values of the quadratic and direct axis inductances, and the stator resistance, we can determine if these controllers can handle high internal disturbances. The new values of L_q, L_d, and R are 0.022H, 0.013H, and 3.5Ω, respectively.

Figure 12 presents the speed tracking performance of each controller after introducing a significant amount of internal disturbances; obviously, despite the drastic increase in the motor parameters, FC-TMFS still managed to deliver the targeted speed with extreme precision and minimum overshoot and steady-state error, whereas all the other controllers struggled and settled at around 20% of the targeted speed.

The induced electromagnetic torque is presented in Fig. 13. In Fig. 13b, c, and d, the conventional FC, PID, and PI speed controllers produced a very unreliable T_e, since it kept on oscillating with a very dangerous frequency and amplitude, whereas in Fig. 13a, FC-TMFS managed to produce an ideal and compact electromagnetic torque.

5.5 Processor-in-loop simulation of FC-TMFS

In order to confirm the applicability of the designed FC-TMFS, TMS320F28379D board was used to replicate and replace: (FC-TMFS speed controller, PI currents controllers, decoupling block, and SVPWM block), to perform a full PIL control of the simulated PMSM and the three levels inverter. Figure 14 presents and highlights the main components of PIL simulation.

Figure 15 compares the performance of PIL with the standard full MATLAB simulation which is called software-in-loop (SIL) simulation, the comparison was based on 0.4s of FOC of PMSM with a fixed speed reference, and an increase in the load torque at 0.2s. At startup, based on Fig. 15a, b, and c, SIL produced speed settled faster than PIL by 0.001 s; however, PIL startup current and electromagnetic torque were 30% less than SIL, at steady state both methods

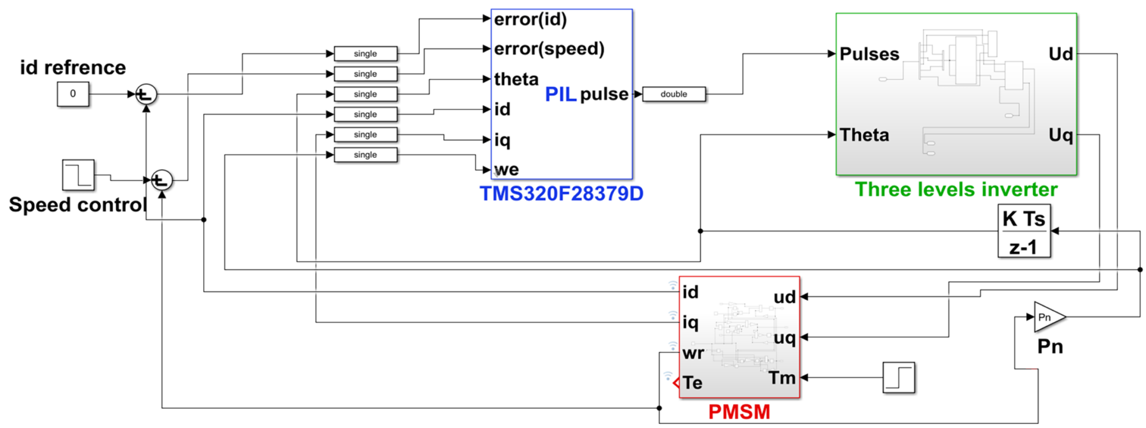


Fig. 14 Processor-in-loop FOC of PMSM

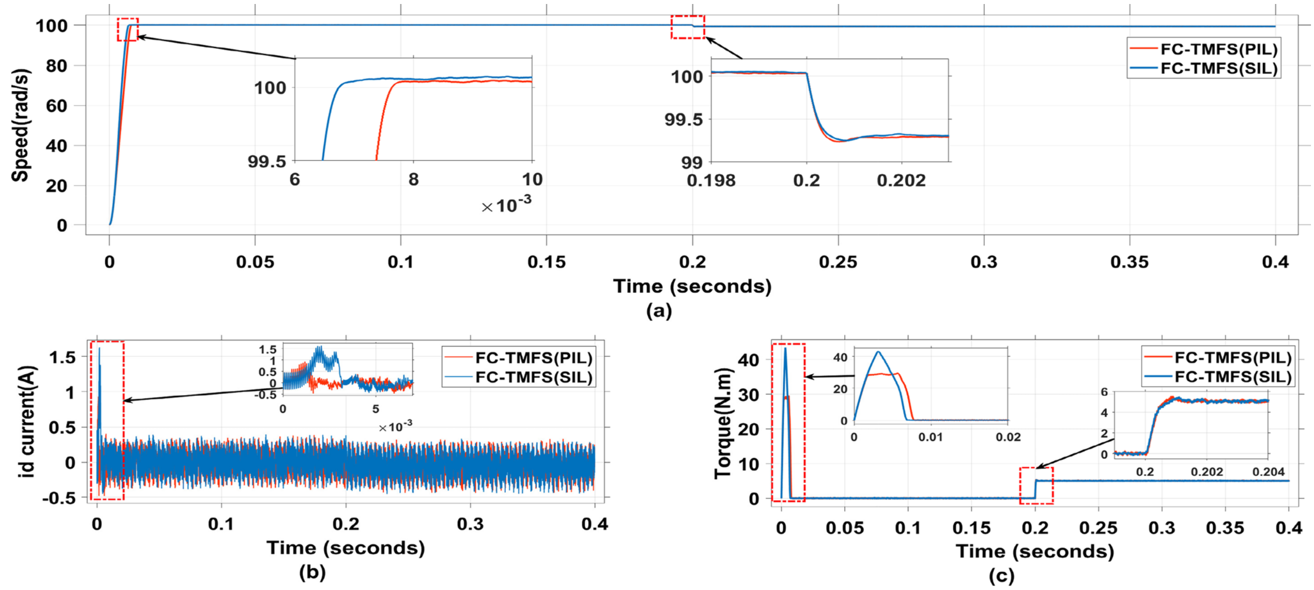


Fig. 15 PIL and SIL FOC of PMSM. **a** Produced mechanical speed. **b** Induced id current. **c** Produced electromagnetic torque

delivered similar results, which strengthens the validity of the proposed controller.

6 Conclusion

In this paper, a hybrid meta-heuristic algorithm called ST-PS-SO was proposed for the tuning of the output gain, and the MFS of FC-TMFS for speed control of PMSM. With the help of MATLAB/SIMULINK simulation, the superiority of the proposed algorithm when tuning FC-TMFS was validated against STBO, PSO, and SOS, which is due to ST-PS-SO combination of guided and efficient exploration, and fast convergence. Then, ST-PS-SO-based FC-TMFS (SPS-FC-TMFS) was thoroughly compared to the conventional FC, and the classical PID and PI controllers. The proposed SPS-FC-TMFS managed to utterly outperform every controller with way more durable speed control of PMSM, no overshoot/undershoot problems, and way less steady-state error with or without a charge load. Then after applying an enormous increase to the parameters of the motor, SPS-FC-TMFS still managed to start and track the setpoint of speed reference, while the conventional FC, PI, and PID destabilized and caused the system to malfunction and fail to deliver a reliable mechanical speed, which again proves the significance of the proposed algorithm, and the necessity of tuning the MFS of the standard FC. Finally, results of processor-in-loop (PIL) simulation of the designed controller were presented and analyzed to further enhance the credibility and practical suitability of speed control of PMSM with the proposed SPS-

FC-TMFS. Although the performance of FC-TMFS is very satisfactory, it can still be further improved by using a Type-2 standard FC or tuned using newer optimization algorithms.

Author Contributions Tahar.N wrote the main manuscript TEXT Abdelhakim.D and Abdelmalik.B prepared figures. All authors reviewed the manuscript.

Data Availability No datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The authors have no competing interest to declare that are relevant to the content of this article.

References

1. Rauth SS, Samanta B Comparative analysis of IM/BLDC/PMSM drives for electric vehicle traction applications using ANN-based FOC. In: 2020 IEEE 17th India Council International Conference (INDICON) pp 1–8. IEEE
2. Guezi A, Bendaikha A, Dendouga A (2022) Direct torque control based on second order sliding mode controller for three-level inverterfed permanent magnet synchronous motor: comparative study. https://doi.org/10.20998/2074-272X.2022.5.02
3. Dendouga A, Guezi A, Bendaikha A Performance improvement of permanent magnet synchronous motor fed by direct matrix converter by using second-order sliding mode control. In: 2022 5th International Conference on Power Electronics and their Applications (ICPEA) Vol. 1, pp 1–6. IEEE
4. Bida VM, Samokhvalov DV, Al-Mahturi FS PMSM vector control techniques-A survey. In: 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus) pp 577–581. IEEE
5. Nouaoui T, Dendouga A, Bendaikha A Fractional Order PID tuned using symbiotic organism search algorithm for field oriented control of the permanent magnet synchronous motor. In: 2023 International Conference on Advances in Electronics, Control and Communication Systems (ICAECCS) pp 1–5. IEEE
6. Junejo AK, Xu W, Mu C, Ismail MM, Liu Y (2020) Adaptive speed control of PMSM drive system based a new sliding-mode reaching law. IEEE Trans Power Electron 35:12110–12121. https://doi.org/10.1007/s001090000086
7. Khanh PQ, Anh HPH (2023) Hybrid optimal fuzzy Jaya technique for advanced PMSM driving control. IEEE Trans Power Electron. https://doi.org/10.1007/s00202-023-01911-6
8. Unsal S, AliSkan I (2017) Investigation of speed control performances of the fuzzy logic controllers having different membership functions and inference methods. Anadolu Univ J Sci Technol A-Appl Sci Eng 18:831–841. https://doi.org/10.18038/aubtda.341397
9. Belman-Flores JM, Rodríguez-Valderrama DA, Ledesma S, García-Pabón JJ, Hernández D, Pardo-Cely DM (2022) A review on applications of fuzzy logic control for refrigeration systems. Appl Sci 12:1302. https://doi.org/10.3390/app12031302
10. Alraih S, Nordin R, Abu-Samah A, Shayea I, Abdullah NF, Alhammadi A (2022) Robust handover optimization technique with fuzzy logic controller for beyond 5G mobile networks. Appl Sci 22:6199. https://doi.org/10.3390/s22166199
11. Shuraiji AL, Shneen SW (2022) Fuzzy logic control and PID controller for brushless permanent magnetic direct current motor: a comparative study. J Robot Control (JRC) 3:762–768. https://doi.org/10.18196/jrc.v3i6.15974
12. Dendouga A, Guezi A, Bendaikha A Robustness evaluation of fuzzy logic control based on feedback linearization of induction motor fed by matrix converter. In: 2022 5th International Conference on Power Electronics and their Applications (ICPEA) Vol. 1, pp 1–7. IEEE
13. García-Gutiérrez G, Arcos-Aviles D, Carrera EV, Guinjoan F, Motoasca E, Ayala P et al (2019) Fuzzy logic controller parameter optimization using metaheuristic cuckoo search algorithm for a magnetic levitation system. Appl Sci 9:2458. https://doi.org/10.3390/app9122458
14. Yousri D, Allam D, Eteiba M (2018) Parameters identification of fractional order permanent magnet synchronous motor models using chaotic meta-heuristic algorithms. Math Tech Frac Order Syst. https://doi.org/10.1016/B978-0-12-813592-1.00018-0
15. Aguilar-Mejía O, Minor-Popocatl H, Tapia-Olvera R (2020) Comparison and ranking of metaheuristic techniques for optimization of PI controllers in a machine drive system. Appl Sci 10:6592. https://doi.org/10.3390/app10186592
16. Wang D, Tan D, Liu L (2018) Particle swarm optimization algorithm: an overview. Soft Comput 22:387–408. https://doi.org/10.1007/s00500-016-2474-6
17. Guha D, Roy PK, Banerjee S Adaptive symbiotic organism search algorithm optimized 3DOF-PID Controller for load frequency control of hybrid power system. In: 2020 IEEE Calcutta Conference (CALCON) pp 1–6. IEEE
18. Dehghani M, Trojovska E, Zuscak T (2022) A new human-inspired metaheuristic algorithm for solving optimization problems based on mimicking sewing training. Sci Rep 12:17387. https://doi.org/10.1038/s41598-022-22458-9
19. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
20. Mirjalili S, Mirjalili SM, Saremi S, Mirjalili S (2020) Whale optimization algorithm: theory, literature review, and application in designing photonic crystal filters. In: Mirjalili S, Dong JS, Lewis A (eds). Springer
21. Grefenstette JJ Genetic algorithms and machine learning. In: Proceedings of the sixth annual conference on Computational learning theory
22. Hannan MA, Ali JA, Mohamed A, Amirulddin UAU, Tan NML, Uddin MN (2018) Quantum-behaved lightning search algorithm to improve indirect field-oriented Fuzzy-PI control for IM drive. IEEE Trans Ind Appl 54:3793–3805. https://doi.org/10.1109/TIA.2018.2821644
23. Zhang L, Ma J, Wu Q, He Z, Qin T, Chen C (2023) Research on PMSM speed performance based on fractional order adaptive fuzzy backstepping control. Energies 16:6922. https://doi.org/10.3390/en16196922
24. Zheng Y, Zhao H, Zhen S, He C (2021) Designing robust control for permanent magnet synchronous motor: fuzzy based and multivariable optimization approach. IEEE Access. 9:39138–39153. https://doi.org/10.1109/ACCESS.2021.3056890
25. Hoai HK, Chen SC, Chang CF (2020) Realization of the neural fuzzy controller for the sensorless PMSM drive control system. Electronics 9:1371. https://doi.org/10.3390/electronics9091371
26. Puci I, Vilanova R, Ferrer CP Comparison of speed control of permanent magnet synchronous motor using pi and fuzzy controller. In: 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)
27. Nanda B, Thakur AN (2015) Fuzzy logic based field oriented control of permanent magnet synchronous motor. Int J Electr Electron Data Commun 3:18479

28. Jeevananthan P, Kumar CS, Nithya RK (2012) Fuzzy Logic Based Field Oriented Control of Permanent Magnet Synchronous Motor (PMSM). Int J Adv Res Electr, Electron Instrum Eng 1
29. Dursun M, Boz AF (2015) The analysis of different techniques for speed control of permanent magnet synchronous motor. Technical gazette 22:947–952. https://doi.org/10.17559/TV-20140912141639
30. Rodriguez M, Arcos-Aviles D, Martinez W (2023) Fuzzy logic-based energy management for isolated microgrid using meta-heuristic optimization algorithms. Appl Energy 355:120771. https://doi.org/10.1016/j.apenergy.2023.120771
31. Unsal S, Aliskan I (2022) Investigation of performance of fuzzy logic controllers optimized with the hybrid genetic-gravitational search algorithm for PMSM speed control. Automatika: casopis za automatiku, mjerenje, elektroniku, racunarstvo i komunikacije 63:313–327. https://doi.org/10.1080/00051144.2022.2036936
32. Abassi M, Khlaief A, Saadaoui ABM O Chaari Performance analysis of FOC and DTC for PMSM drives using SVPWM technique. In: 2015 16th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA) pp 228–233. IEEE
33. Gadoue SM, Giaouris D, Finch JW (2009) MRAS sensorless vector control of an induction motor using new sliding-mode and fuzzy-logic adaptation mechanisms. IEEE Trans Energy Convers 25:394–402. https://doi.org/10.1109/TEC.2009.2036445
34. Liang H, Zou J, Zuo K, Khan MJ (2020) An improved genetic algorithm optimization fuzzy controller applied to the wellhead back pressure control system. Mech Syst Signal Process 142:106708. https://doi.org/10.1016/j.ymssp.2020.106708
35. Ali A, Majhi S (2009) PI/PID controller design based on IMC and percentage overshoot specification to controller setpoint change. ISA Trans 48:10–15. https://doi.org/10.1016/j.isatra.2008.09.002
36. Ozgenc B, Ayas MS, Altas IH (2022) Performance improvement of an AVR system by symbiotic organism search algorithm-based PID-F controller. Neural Comput Appl. https://doi.org/10.1007/s00521-022-06892-4