ORIGINAL ARTICLE

A comparative experimental study of direct torque control based on adaptive fuzzy logic controller and particle swarm optimization algorithms of a permanent magnet synchronous motor

H. Mesloub[1] · M. T. Benchouia[1] · A. Goléa[1] · N. Goléa[2] · M. E. H. Benbouzid[3]

Received: 18 February 2016 / Accepted: 26 June 2016 / Published online: 26 August 2016

Abstract Direct torque control (DTC) of permanent magnet synchronous motor (PMSM) drives is receiving increasing attention due to important advantages, such as fast dynamic and low dependence on motor parameters. However, conventional DTC scheme, based on comparators and the switching table, suffers from large torque and flux ripples. In this paper, two intelligent approaches are proposed in order to improve DTC performance. The first approach is based on two adaptive fuzzy logic controllers (AFLC). The first AFLC replaces the conventional comparators and switching table and the second AFLC adjusts in real time the outer loop PI parameters. In the second approach, particle swarm optimization (PSO) is used as another alternative to adjust the PI parameters. Simulation and experimental results demonstrate the effectiveness of the proposed intelligent techniques. Besides, the system associated with these techniques can effectively reduce flux and torque ripples with better dynamic and steady state performance. Quantitatively, PSO-based DTC approach reduces greatly flux and torque ripples. Further, PSO-based approach maintains a constant switching frequency which improves the PMSM drive system control performance.

Keywords Permanent magnet synchronous motor (PMSM) · Direct torque control (DTC) · Adaptive fuzzy logic controller (AFLC) · Particle swarm optimization (PSO) · Practical validation

1 Introduction

Permanent magnet synchronous motor (PMSM) has high efficiency, high reliability, and simple structure and has been used in many fields. PMSM has wide development because of different kinds and wide application and has many advantages over an AC induction motor. PMSM generates the rotor magnetic flux with rotor magnets, so achieves higher efficiency. The classification of the PMSM, their merits and demerits, magnetic characteristics of magnets used for PMSM and their comparison with Induction Motors are presented in [1]. Therefore, the PMSM are used in electric and hybrid vehicles, high-end white goods (refrigerators, washing machines, dishwashers, etc.), high-end pumps, fans, and in other appliances, which require high reliability and efficiency. The PMSM is a synchronous which certainly requires accompanying power electronics, but it also provides the basis for achieving high-quality actuator control [2]. Despite its advantages, such as high efficiency, high power density, and high torque to current ratio, the PMSM remains complicated and difficult to control when good transient performance under all operating conditions is desired. This is due to the fact that the PMSM is a nonlinear, multivariable, time-varying system subjected to unknown disturbances and variable parameters. Over the past decades, various robust control techniques have been developed in order to improve the performances of the PMSM in the presence of external disturbances. However, the

✉ H. Mesloub
sassamesloub@yahoo.fr

[1] LGEB Laboratory, University of Biskra, BP. 145, 07000 Biskra, Algeria

[2] LGEA Laboratory, Larbi Ben M'hidi University, 04000 Oum El-Bouaghi, Algeria

[3] LBMS, University of Brest, Brest Cedex 03, France

widely used approach consists in using linear control theory with the disturbance estimate [3, 4]. In [5], the robustness is ensured by using H^{∞} control theory.

Direct torque control (DTC) has a relatively simple control structure yet performs at least as good as the field-oriented control (FOC technique). It is also known that DTC drive is less sensitive to parameters de-tuning (only stator resistor is used to estimate the stator flux) and provides a high dynamic performances than the classical vector control (fastest response of torque and flux) [6, 7]. This method allows a decoupled control of flux and torque without using speed or position sensors. This type of command involves hysteresis nonlinear control, for both stator flux magnitude and electromagnetic torque, which introduces limitations such as a high and uncontrollable switching frequency [8]. This controller produces a variable switching frequency and consequently large torque and flux ripples and high currents distortion. The DTC is mostly used in the objective to improve the reduction of the undulations or the flux's distortion and to have good dynamic performances. It is essentially based on a localization table which allows selecting the vector tension to apply to the inverter according to the position of the stator flux vector and of the direct control of the stator flux and the electromagnetic torque.

Many artificial intelligence techniques and Random search methods have been employed to improve the controller performances. Neural network, fuzzy system, and genetic algorithm have been widely applied to proper tuning of PID controller parameters [9–12]. But all have some shortages. GA has a big computational complexity. Fuzzy system itself has many parameters to be optimized; the results of these experiments showed that fuzzy controllers perform better, or at least well as, adaptive controllers. Moreover, this technique offers the advantage of requiring only a simple mathematical model to formulate the algorithm, which can easily be implemented by a digital computer [13–15]. These features are appreciated for nonlinear processes for which there is no reliable model and complex systems where the model is useless due to the large number of equations involved. Nevertheless, the main problem with fuzzy logic is that there is no systematic procedure for the design of fuzzy controller [16–18].

With the superiority of fuzzy controller, it can adapt its structure, acting on a number of factors which constitute the internal configuration of this type of controller, such as fuzzification blocks, fuzzy rules, block defuzzification, and input and output gains. Additionally, it is possible to use the fuzzy logic to adjust or supervise the parameters of traditional PI regulator [19] and to replace, on the one hand, the conventional comparators and the switching table in order to reduce torque and flux ripples, nonlinear controller [20], neural network algorithms [9, 21, 22], and particle swarm optimization (PSO) [23–25].

Particle swarm optimization is a stochastic global optimization technique. PSO find optimal regions of complex search spaces through the interaction of individuals in a population of particles and has been found to be robust in solving continuous nonlinear optimization problems [26–28]. The PSO technique can generate a high-quality solution within shorter calculation time and stable convergence characteristic than other stochastic methods. Because the PSO method is an excellent optimization methodology and a promising approach for solving the optimal PI controller parameters problem [29, 30]; therefore, this study develops the PSO-PI controller to search optimal PI parameters. This PI controller is called the PSO-PI controller. The PSO Control has been employed to generate the required torque to implement the DTC technique, while minimizing the torque and the flux ripples and the switching frequency.

In this paper, a direct torque controlled PMSM based on the adaptive fuzzy logic controllers (AFLC) and the PSO is presented. A comparison study between these techniques is also presented. To show the performances of the DTC of the PMSM based on the AFLC and the PSO, simulation results are presented. To validate the simulation results, these algorithms are implemented on a test bench around a DSPACE 1104. These techniques show high performances compared to the conventional DTC. Comparing these techniques, we see that the DTC based on the PSO is more efficient than the DTC based on the AFLC.

2 DTC principal

The electrical and the mechanical equations of PMSM in the stator reference frame (α–β) are as follows:

$$\begin{cases} L_s i_{s\alpha} = -R_s i_{s\alpha} + \omega\phi_m \sin(\theta) + v_{s\alpha} \\ L_s i_{s\beta} = -R_s i_{s\beta} + \omega\phi_m \cos(\theta) + v_{s\beta} \\ \phi_{s\alpha} = -R_s i_{s\alpha} + v_{s\alpha} \\ \phi_{s\beta} = -R_s i_{s\beta} + v_{s\beta} \end{cases} \tag{1}$$

Where $i_{s\alpha}$, $i_{s\beta}$ and $v_{s\alpha}$, $v_{s\beta}$ are, respectively, stator currents and voltages $\phi_{s\alpha}$, $\phi_{s\beta}$ are separate magnetic flux linkage generated by stator currents; ϕ_m is the permanent magnet rotor flux. R_s is armature winding resistance, and L_s denotes the total inductance for each phase.

The mechanical equation for the motor dynamics, on the other hand, is

$$T_e - T_r - f\Omega = J\frac{d\Omega}{dt} \tag{2}$$

Where J is the total moment of inertia of the rotor and f is the friction coefficient. T_r is the load torque and T_e is the electromagnetic torque which is given by

$$T_e = \frac{3}{2}p\left(\phi_{s\alpha} i_{s\alpha} - \phi_{s\beta} i_{s\alpha}\right) \tag{3}$$

With p as number of pole pairs.

The main idea of the DTC is to directly control the torque and flux produced by the machine, without current control, as it is the case in FOC [31, 32]. Different approaches have been developed [33, 34]. Figure 1 is a typical DTC system. Usually, a DC bus voltage sensor and two phase currents sensors are needed for the flux and torque estimator. The stator flux amplitude, $\phi_s = \sqrt{\phi_{s\alpha}^2 + \phi_{s\beta}^2}$ and position $\delta = \tan^{-1}(\phi_{s\beta}/\phi_{s\alpha})$ are computed from the flux components estimation given by:

$$\begin{cases} \phi_{s\alpha} = \int_0^{T_e} (V_{s\alpha} - R_s I_{s\alpha})dt + \phi_{s\alpha_0} \\ \phi_{s\beta} = \int_0^{T_e} \left(V_{s\beta} - R_s I_{s\beta}\right)dt + \phi_{s\beta_0} \end{cases} \tag{4}$$

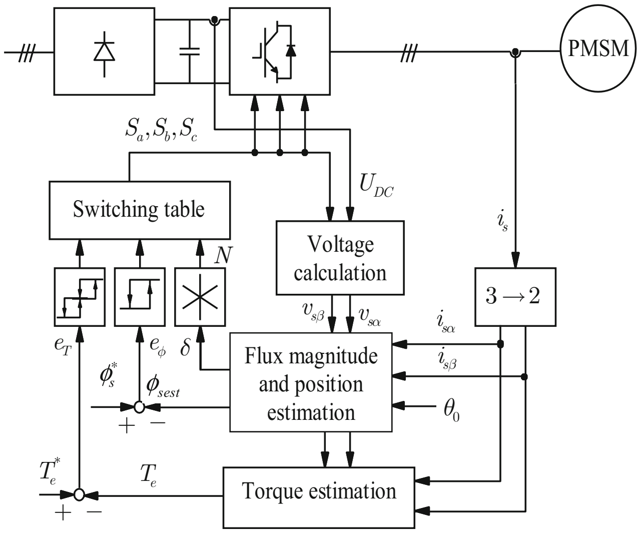


Fig 1 PMSM DTC block diagram

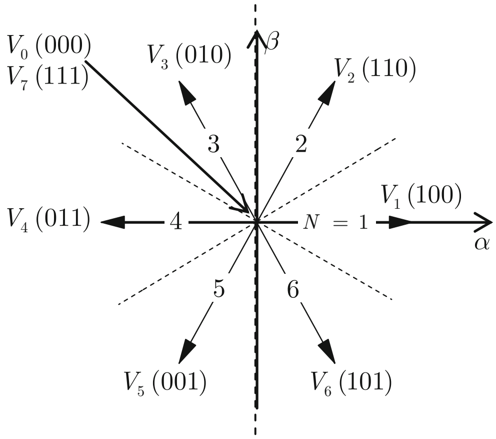


Fig 2 Two-level VSI voltage vectors and sectors

Where $\phi_{s\alpha_0}$ and $\phi_{s\beta_0}$ are the initial stator flux values.

The torque is estimated from (3).

The switching state of the inverter is updated, at each sampling time T_s, depending on flux and torque hysteresis comparators outputs and stator flux position sector as shown in Fig. 2 and Table 1. Therefore, the switching frequency is usually not fixed; it changes with the rotor speed, load, and bandwidth of the flux and torque controllers. The main advantages of DTC are the absence of coordinate transformation and current regulator and the absence of separate voltage modulation block. Common disadvantages of conventional DTC are high torque ripple and slow transient response to the step changes in torque during start-up.

3 DTC based on adaptive fuzzy logic controller

The introduction of adaptive fuzzy concept adjusts real time the PI parameters K_p and K_i, through fuzzy inference mechanism (AFLC$_1$). According to electromagnetic torque and

Table 1 PMSM DTC switching table

		Sector N					
$e_\varnothing$	e_T	1	2	3	4	5	6
1	1	V_2	V_3	V_4	V_5	V_1	V_4
	0	V_7	V_0	V_7	V_0	V_7	V_0
	−1	V_6	V_1	V_2	V_3	V_4	V_5
1	1	V_3	V_4	V_5	V_6	V_1	V_2
	0	V_7	V_0	V_7	V_0	V_7	V_0
	−1	V_6	V_1	V_2	V_3	V_4	V_5

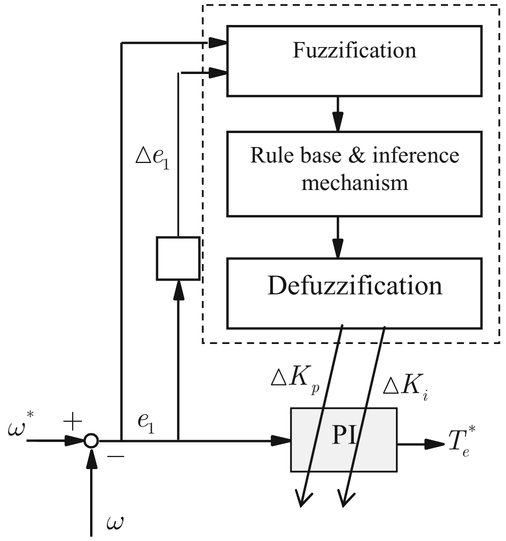


Fig 3 Block diagram of the fuzzy adaptive PI control

stator flux linkage error, a method that fuzzy controller uses in selecting the proper voltage vectors controls the system (AFLC$_2$). The proposed system configuration is shown in Fig. 5.

3.1 Structure of fuzzy adaptive logic controller (AFLC$_1$_PI)

Figure 3 shows a schematic of the AFLC1 with its three main parts: fuzzification, fuzzy rules, and inference and defuzzification. The AFLC1 inputs are speed error $e_1(k)$ and change of speed error $\vartriangle e_1$. The AFLC1 outputs $\vartriangle K_p$ and $\vartriangle K_i$ are changes to adjust the PI proportional gain K_p and the integral gain K_i.

3.1.1 Fuzzification of input and output variables

The first step in designing a fuzzy logic controller is fuzzification of the input and output variables, which converts the input and output data to semantic value. For the fuzzification purpose, the input and output variables are normalized to the interval $[-1, 1]$ using appropriate gains values. Then, normalized variables are fuzzified using seven fuzzy linguistic variables as shown in Fig. 5, where NB is negative big; NM is negative middle; NS is negative small; ZO is zero; PS is positive small; PM is positive middle; and PB is positive big. All of the membership functions are triangular. The fuzzy membership functions are presented in Fig. 4.

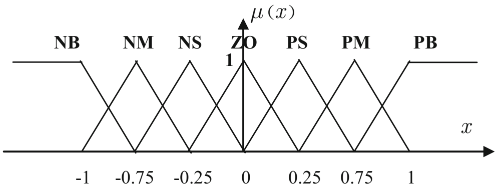


Fig. 4 Fuzzy membership functions for AFLC1

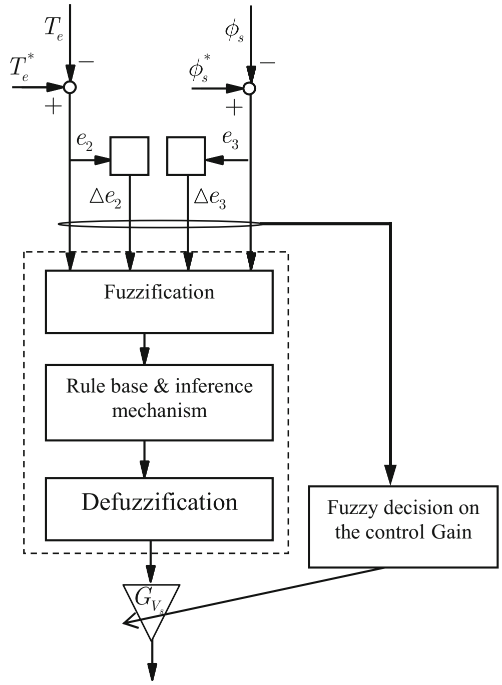


Fig. 5 Adaptive fuzzy controller structure

3.1.2 Fuzzy inference and defuzzification

The most significant step in fuzzy control design is establishment of fuzzy inference rule between the input variables e_1, $\vartriangle e_1$ and the output variables $\vartriangle K_p$, and $\vartriangle K_i$ according to the knowledge and experience of experts or input–output data. The fuzzy rules are summarized in Tables 2 and 3.

Table 2 Fuzzy rules for *Kp*

e_1 ΔK_p Δe_1	NB	NM	NS	ZO	PS	PM	PB
NB	PB	PB	PM	PM	PS	ZO	ZO
NM	PB	PB	PM	PS	PS	ZO	NS
NS	PM	PM	PM	PS	PS	ZO	NS
ZO	PM	PM	PS	ZO	NS	NM	NM
PS	PS	PS	ZO	NS	NS	NM	NM
PM	PS	ZO	NS	NM	NM	NM	NB
PB	ZO	ZO	NM	NM	NM	NB	NB

Table 3 Fuzzy rules for K_i

e_1 ΔK_i Δe_1	NB	NM	NS	ZO	PS	PM	PB
NB	NB	NB	NM	NM	NS	ZO	ZO
NM	NB	NB	NM	NS	NS	ZO	ZO
NS	NB	NM	NS	NS	ZO	PS	PS
ZO	NM	NM	NS	ZO	PS	PM	PM
PS	NM	NS	ZO	PS	PS	PM	PB
PM	ZO	ZO	PS	PS	PM	PB	PB
PB	ZO	ZO	PS	PM	PM	PB	PB

There are 49 rules in Table 2, and the implication applied in the rules is as follows:

$$R_{ij} : \text{If } e_1 \textit{ is } A_i \text{ and } \triangle e_1 \textit{ is } B_j \text{ then } \triangle K_p/\triangle K_i \textit{ is } C_{ij}/D_{ij} \tag{5}$$

where A_i, B_j, C_{ij}, D_{ij} are corresponding to the fuzzy subsets of e_1, $\triangle e_1$, $\triangle K_p$, and $\triangle K_i$. The Mamdani's Min–Max operator is implemented for doing fuzzy inference. For instance, the degree of membership of the fuzzy subsets C_{ij} for the parameter $\triangle K_p$ can be obtained as follows:

$$u_c(\Delta K_p) = \bigvee_{i,j=1}^{7} \left\{ \left[u_i(e_1) \wedge u_j(\Delta e_1)\right] \wedge u_{cij}(\Delta K_p) \right\} \tag{6}$$

Where $u(.)$ is the degree of membership.

Defuzzification is the process of transforming fuzzy variables to crisp values. The center of gravity method is used to get the crisp values. The parameter $\triangle K_p$ ($\triangle K_i$ is similar) can be calculated from the following equation:

$$\Delta K_p(e_1, \Delta e_1) = \frac{\sum_{k=1}^{7} \Delta K_p u_c(\Delta K_p)}{\sum_{k=1}^{7} u_c(\Delta K_p)} \tag{7}$$

After defuzzification, the parameters K_p, K_i can be derived:

$$\begin{cases} K_p = K_{p0} + \Delta K_p \\ K_i = K_{i0} + \Delta K_i \end{cases} \tag{8}$$

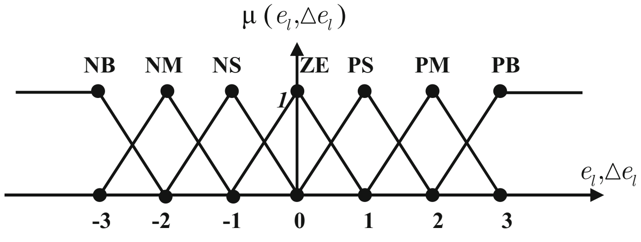


Fig. 6 Membership functions for inputs $e_l(k)$ and $\Delta e_l(k)$

Table 4 Relationship between the variation of the control and its gain

V_s	G_{Vs}
PVS-NVS	PVS
PS-NS	PS
PM-NM	PM
PB-NB	PB
PVB-NVB	PVB

Where K_{p0}, K_{i0} are the PI controller initial parameters.

3.2 Structure of fuzzy adaptive logic controller (AFLC$_2$)

The AFLC$_2$ inputs are flux and torque errors and their variations, and the AFLC$_2$ output vector (V_s) is led to a switching table. The proposed control system shown in Fig. 5 is composed of four stages: fuzzification, rules execution, defuzzification (classical fuzzy control), and adaptation mechanism of the output gain.

The rule base is the principal component of the fuzzy controller; it indicates how the controller behaves to response to any input situation. The rule base is constituted by collection of *If-Then* rules of the form:

$$R_j : \text{if } e_l(k) \textit{ is } A_j \textit{ and } \Delta e_l(k) \textit{ is } B_j \text{Then } V_s(k) \textit{ is } C_j, j = 1, .., m, l = 2, 3 \tag{9}$$

Where A_j, B_j, and C_j are fuzzy sets such as NL (negative large), NM (negative medium), etc., defining fuzzy partition on the controller input space Fig. 6, and $e_l(k)$ and $\Delta e_l(k)$ are scaled and normalized version of:

$$\begin{cases} e_2(k) = T_e{}^*(k) - T_e(k) \\ \Delta e_2(k) = e_2(k) - e_2(k-1) \\ e_3(k) = \phi^*(k) - \phi(k) \\ \Delta e_3(k) = e_3(k) - e_3(k-1) \end{cases} \tag{10}$$

With ge and gce, constant inputs gain which play an essential role, since they determine the control performances. The expression that $e(k)$ is A_j is implemented by membership function indicating the grade of membership of $e_l(k)$ in the fuzzy set A_j as in Fig. 6; this operation

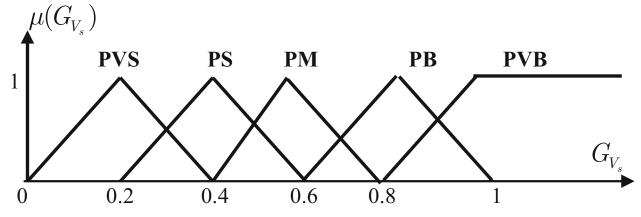


Fig. 7 Membership functions of the output

Table 5 Decision on control gain

Δe_l e_l	NB	NM	NS	ZE	PS	PM	PB
NB	PVB	PVB	PB	PM	PS	PVS	PVS
NM	PVB	PB	PM	PS	PVS	PVS	PVS
NS	PB	PM	PS	PVS	PVS	PVS	PS
ZE	PM	PS	PVS	PVS	PVS	PS	PM
PS	PS	PVS	PVS	PVS	PS	PM	PB
PM	PVS	PVS	PVS	PS	PM	PB	PVB
PB	PVB	PVB	PS	PM	PB	PVB	PVB

is called fuzzification. The shape of the membership function is quite arbitrary and depends on the user's preference. For simplicity, triangular and trapezoidal shapes are usually used. The logical operators "and" and "Then" can be interpreted as min or algebraic product, and various inference and defuzzification algorithms can be used to produce crisp output value. If the operators "and" and "Then" are implemented as algebraic product, the max-product inference and the center of gravity defuzzification methods are adopted in this paper.

In most fuzzy control studies [21, 35–37], the gain associated with the control output must be constant and as low possible in order to avoid the instability problem. This increases considerably the response time of the system. To solve this problem, the output gain is considered as a fuzzy variable (Fig. 6).

Therefore, the gain must be adapted at every situation of the system as a function of the error and its variation. To improve the fuzzy decision table, we associate, for each fuzzy set of the control, the fuzzy set gain, similar but always strictly positive (Table 4).

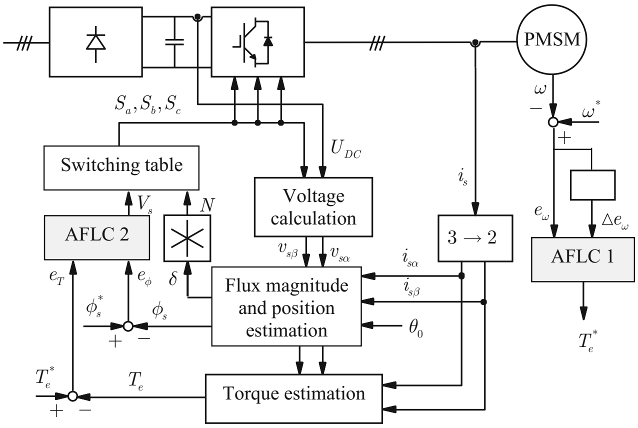


Fig. 8 Direct torque adaptive fuzzy logic controller (AFLC) scheme for PMSM

The fuzzy sets of variable gain that choose their corresponding membership functions are represented by Fig. 7. The decision matrix on the control of the output gain is given in Table 5.

4 Simulation model and structure of DTC system based AFLC

In the DTC scheme proposed in Fig. 1, a hysteresis controller is used. The output of hysteresis controller has only two states, which naturally leads to tacking the same action for the big torque error and small one. As a consequence, a poor performance is made in response to step change and large torque ripple. To improve the performances of PMSM, a DTC method based on an AFLC is used. The AFLC concept adjusts the real time parameters of K_p and K_i and hysteresis controller is replaced by two inputs AFLC$_2$ and the vector output of the AFLC$_2$ is led to a switching table to decide which vector should be applied. This method has the advantage of AFLC classification and has the advantage of simplicity and easy to implement. The basic structure of the proposed algorithm is shown in Fig. 8. The main blocks will be illustrated in the following.

5 DTC based on particle swarm optimized (DTC_PSO)

In PSO, this behavior of animals is imitated by particles with certain positions and velocities in a searching space, wherein the population is called a swarm, and each member of the swarm is called a particle. Members of a swarm communicate good positions to each other and dynamically adjust their own position and velocity based on these good positions. The velocity adjustment is based upon the historical behaviors of the particles themselves as well as their neighbors. In this way, the particles tend to fly towards better and better searching areas over the searching process [38]. Particle swarm optimization has been enormously successful in several and various industrial domains. It has been used across a wide range of engineering applications [39–42]. These applications can be summarized around domains of robotics, image and signal processing, electronic circuits design, and communication networks, but more especially the domain of plant control design, as shown in [43–45].

The PSO technique can generate a high-quality solution within shorter calculation time and stable convergence characteristic than other stochastic methods [46].

Because the PSO method is an excellent optimization methodology and a promising approach for solving the optimal PI controller parameters problem [41, 47, 48]; therefore, this study develops the PSO-PI controller to search optimal PI parameters. This PI controller is called the PSO-PI controller.

5.1 Implementation of a PSO-PI controller

Model and parameters are varying while PMSM is running. The changing of model and parameters will reduce the performance of the dynamic system. To select the PI parameter automatically at the initial stage of the system and to fine adjust the PI parameter according to the operating condition, the PSO controller has been introduced to tuning PI parameter for PMSM. Based on this idea, take the possible PMSM PI parameters as the position of a particle, through finding the optimal particle in the solution space to implement PI parameters' online self-tuning.

The basic principle of PSO controller used in PMSM can be described as follows. Considering a group which consists of n particles, each particle searches the best position under a certain velocity. It updates its position according to the best record of its own and others' in the history.

The current position of particle "l" is represented as

$$K^l = \left(K_p^l,\ K_i^l\right) \quad l = 1,\ 2, \ldots,\ n \tag{11}$$

The current velocity of particle "l" is represented as

$$V^l = \left(V_p^l,\ V_i^l\right) \quad l = 1,\ 2, \ldots,\ n \tag{12}$$

The best position of particle "l" in its search history is represented as

$$P^l = \left(P_p^l,\ P_i^l\right) \quad l = 1,\ 2, \ldots,\ n \tag{13}$$

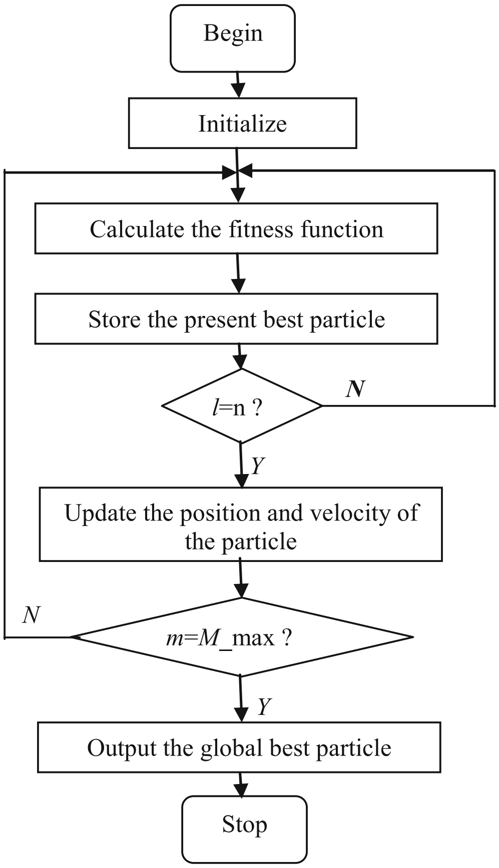

Begin
Initialize
Calculate the fitness function
Store the present best particle
l=n ?
N
Y
Update the position and velocity of the particle
N
m=M_max ?
Y
Output the global best particle
Stop

Fig. 9 The program flowchart

The best position of the population in its search history is represented as

$$P_{best}^g = \left(P_p^g,\ P_i^g\right) \quad l = 1,\ 2, \ldots,\ n \tag{14}$$

Update the velocity and position by the following equations:

$$\begin{cases} V^{l+1} = w\cdot V^l + C_1\cdot R_1\cdot\left[P^l - K^l\right] + C_2\cdot R_2\cdot\left[P_{best}^g - K^l\right] \\ K^{l+1} = K^l + V^{l+1} \quad l = 1,\ 2, \ldots,\ n \end{cases} \tag{15}$$

Table 6 Result of the fitness values after generation

Iteration	Parameters	Fitness
1	Velocity (1,1)=========>Kp = Current_position (1,1) = 0.698 Velocity (2,1)=========>Ki = Current_position (2,1) = 0.701	0.1400
7	Velocity (1,1)=========>Kp = Current_position (1,1) = 0.6000 Velocity (2,1)=========>Ki = Current_position (2,1) = 0.6500	0.040
16	Velocity (1,1)=========>Kp = Current_position (1,1) = 0.0300 Velocity (2,1)=========>Ki = Current_position (2,1) = 0.5000	0.018
20	Velocity (1,1)=========>Kp = Current_position (1,1) = 0.0100 Velocity (2,1)=========>Ki = Current_position (2,1) = 0.6000	0.018

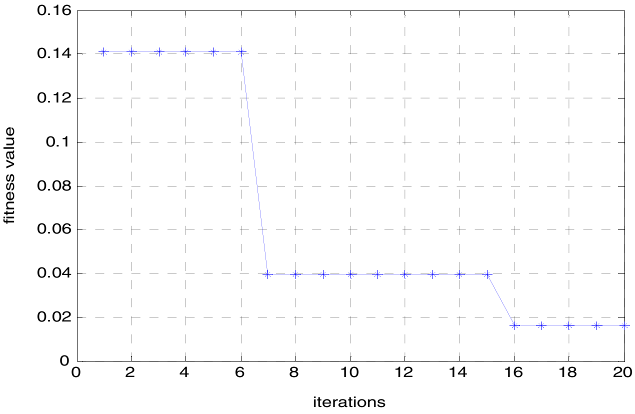


Fig. 10 Optimization process of the PSO

Where ω is the constant inertia weight, C_1, C_2 is the learning factor (C_1, C_2 is usually on [0, 4] interval); R_1, R_2 are uniformly distribution pseudo-random numbers on [0, 1] interval. The velocity of a particle is usually limited to a maximum speed. It can prevent the system unstable from the affect of some bad particles. In general, the motor needs the PI parameters be tuned at the appropriate value quickly during start-up. Then, make fine adjustments according to the load torque to improve the performance of the system. So, we use time-varying weights in the speed updating (14) to substitute the constant weight and set the weight range on $[\omega_{\max}, \omega_{\min}]$ interval. At each sampling time, the population iterates $M\_\max$ times. The iteration in the time “m” of the inertia weight is

$$\omega^m = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{M\_\max} \cdot m \tag{16}$$

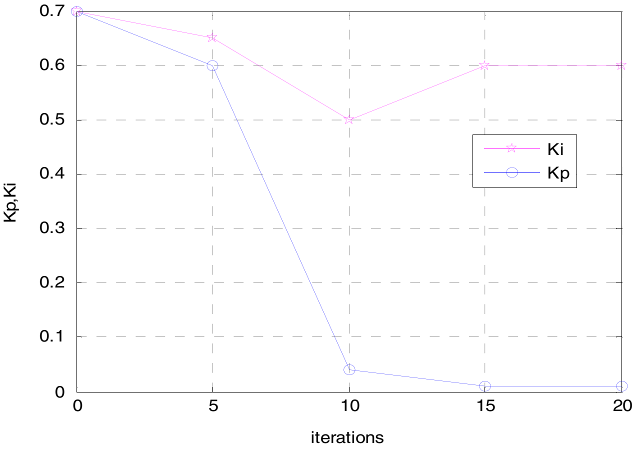
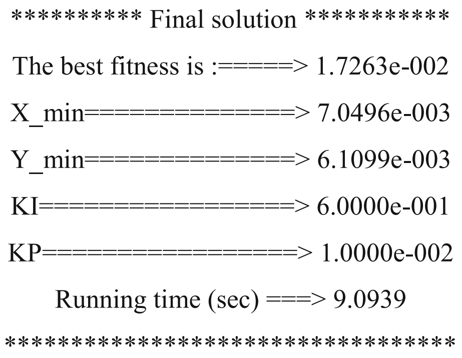


Fig. 11 Adjust the PI parameter



```
********** Final solution ***********
The best fitness is :=====> 1.7263e-002
X_min==============> 7.0496e-003
Y_min==============> 6.1099e-003
KI================> 6.0000e-001
KP================> 1.0000e-002
Running time (sec) ===> 9.0939
**********************************
```

Ring topology is used as the neighbor topology of particle swarm. The influence of neighbors is delivered one by one until the best particle is found. The fitness function composed of the speed error (e_ω) and the speed error change (Δe_w) of PMSM is as

$$F = R_1 \cdot |e_w(l)| + R_2 \cdot |\Delta e_w(l)| \qquad R_1, R_2 \in [0,\ 1] \tag{17}$$

Through the iteration of $M\_\max$ times, we found the best particle which produces a minimum fitness function. The position value (K_p, K_i) of this particle in the search space is the optimal PI parameters. The pseudo-code of PSO tuning algorithm is given below. The general particle swarm optimization algorithm may be applied to any optimization problem. Figure 9 presents a flowchart of the particle swarm optimization algorithm (Table 6):

From Fig. 10, it can be seen that the algorithm usually converges within 20 iterations because of a good initial guess. But from the figure, we can also see that the initial fitness values are not as good as others. This is

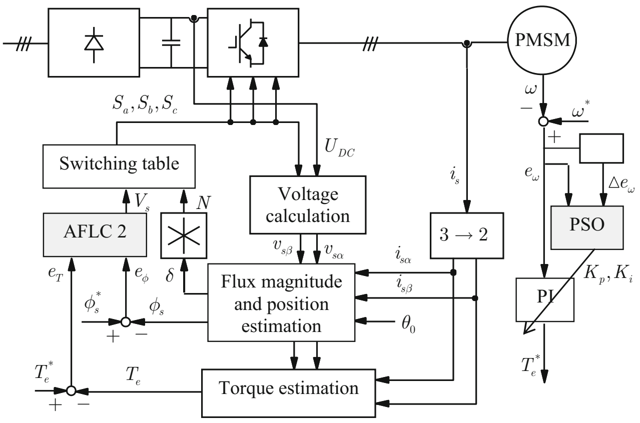
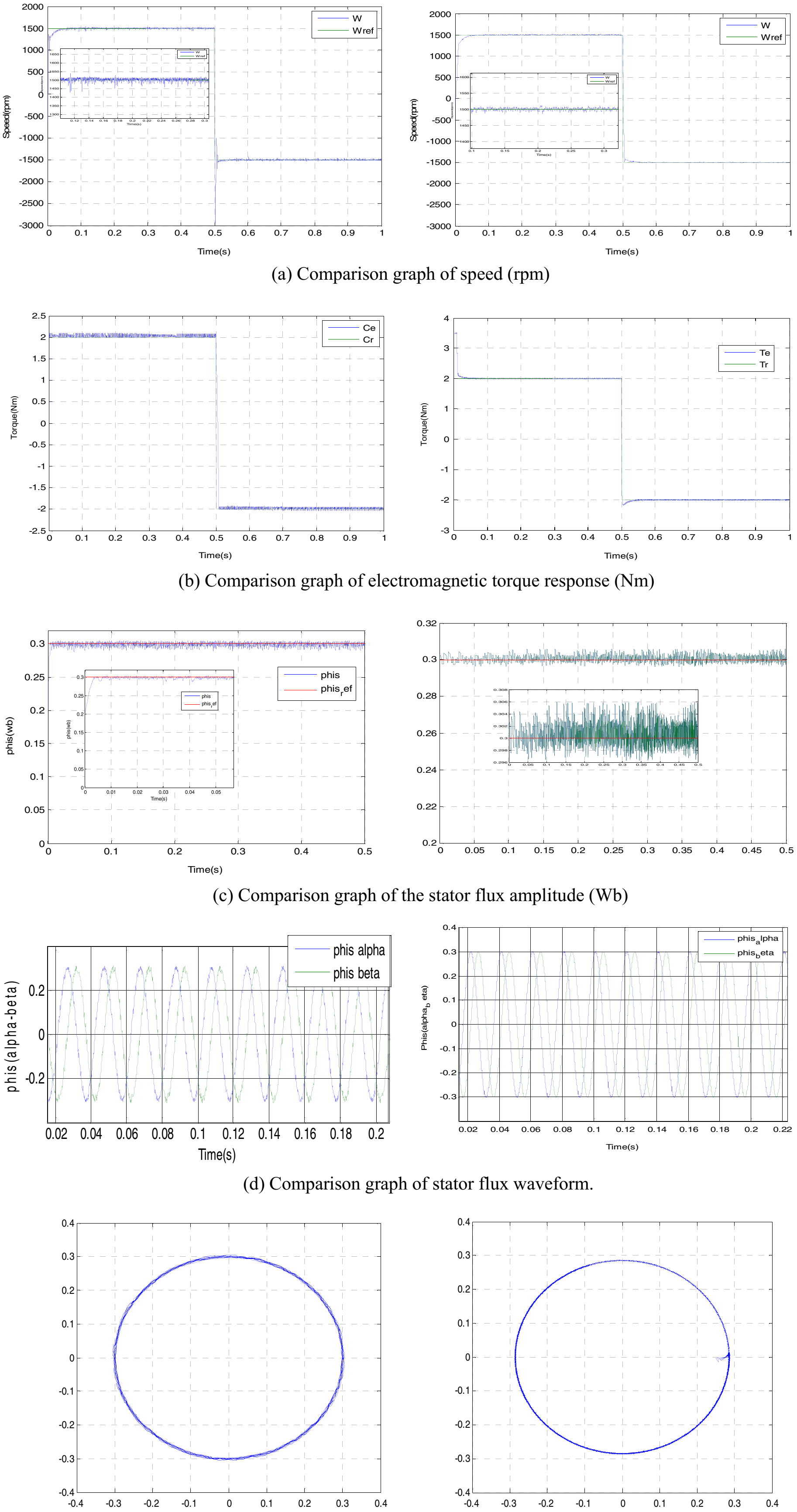


Fig. 12 PMSM direct torque control scheme with PSO_PI and AFLC



(a) Comparison graph of speed (rpm)

(b) Comparison graph of electromagnetic torque response (Nm)

(c) Comparison graph of the stator flux amplitude (Wb)

(d) Comparison graph of stator flux waveform.

(e) Comparison graph of stator flux ($\phi_{s\alpha}$, $\phi_{s\beta}$) trajectory.

Fig. 13 Simulation comparison of DTC_AFLC and DTC_PSO. **a** Comparison graph of speed (rpm). **b** Comparison graph of electromagnetic torque response (Nm). **c** Comparison graph of the stator flux amplitude (Wb). **d** Comparison graph of stator flux waveform **e** comparison graph of stator flux ($\varphi_{s\alpha}$, $\varphi_{s\beta}$) trajectory

because the fitness's sensitivity to parameter changes is different at different operating points. Sometimes, even a small change in the parameters will result in a large change in the fitness (Fig. 11, Table 7).

6 Simulation model and structure of DTC system based AFLC and PSO_PI

To improve the performances of PMSM, a DTC method based on a *PSO* is used. The PSO concept adjusts the real time parameters of K_p and K_i. The basic structure of the proposed algorithm is shown in Fig. 12. The main blocks will be illustrated in the following.

7 Simulation and experimental results

A simulation model of the control system was established by Simulink in MATLAB. The speed and flux references used are 1500 rpm applied at 0 s, −1500 rpm applied at 0.5 s and 0.3 Wb, respectively. The sampling period used is 10–4 ms. A comparative study using simulation between the first approach based on two AFLC and the second one based on the PSO which adjusts the PI parameters was carried out. To validate the simulation results, a test bench based on DSP1104 was developed and the experimental results are compared to the simulation results. The sampling period is the same (10^{-4} ms).

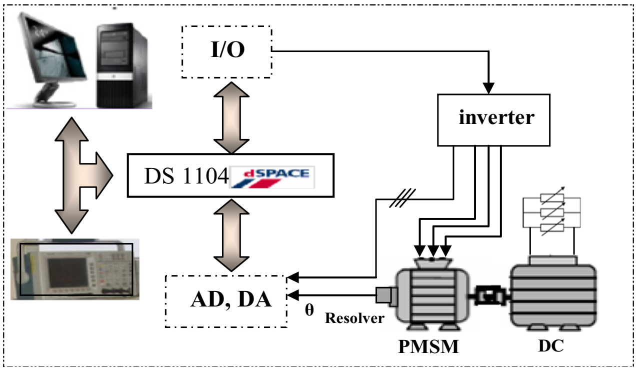


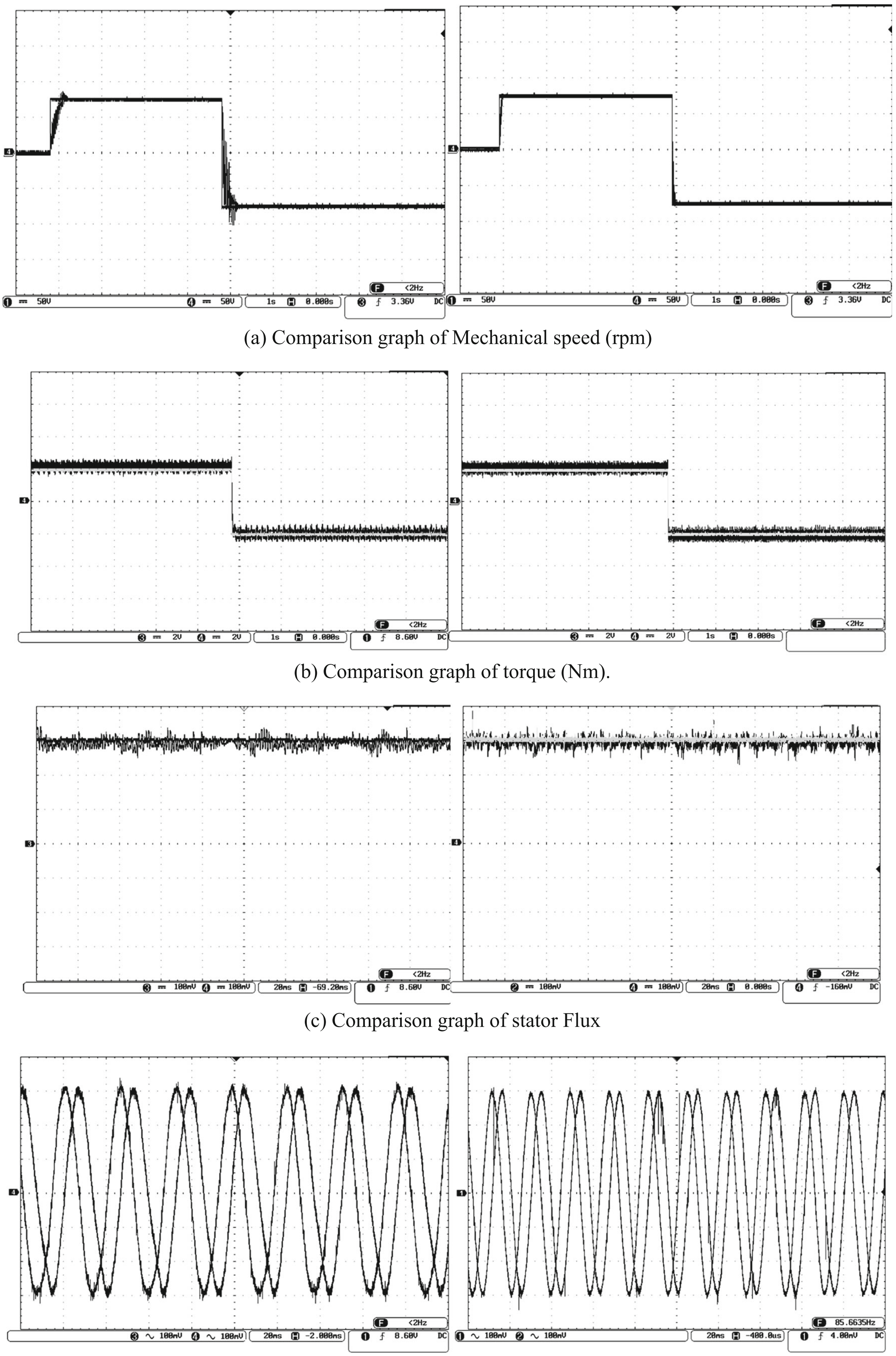
Fig. 14 The hardware setup used in the experiment tests

Fig. 15 Experimental comparison graph of DTC_AFLC and DTC_PSO. **a** Comparison graph of mechanical speed (rpm). **b** Comparison graph of torque (Nm). **c** Comparison graph of stator flux. **d** Comparison graph of flux curve. **e** Comparison graph of trajectory of flux ($\phi_{s\alpha}$, $\phi_{s\beta}$). **f** Comparison graph of spectrum of the current for phase a

Simulation and experimental results (Figs. 13a and 15a) show the speed response in both cases, for a step variation in the reference speed from 1500 to −1500 rpm. In both cases, the speed response is faster without any steady state error; but the DTC based on the PI-PSO gives a good performance compared with the DTC based on two AFLC. The torque responses are presented in Fig. 13b for the simulation and in Fig. 15b for the experimental, respectively. The DTC based on two AFLC shows a better dynamic and steady state performances compared with the DTC based on the PI-PSO. The torque ripples are significantly reduced in the first approach, for the same operating conditions. The stator flux magnitude is shown in Fig. 13c for the simulation and in Fig. 15c for the experimental, respectively. The results show a high dynamic performance, and the ripples in the second approach are remarkably reduced compared with the first one. The simulation and the experimental flux components for both approaches are shown in Figs. 13d and 15d, respectively. The flux components show that the ripples are greatly reduced in the second approach. The stator flux vector describes a trajectory almost circular (Figs. 13e and 15e). Finally, as a consequence of the reduction of the torque and flux ripples, the waveform of the stator current is improved, in particular in the second approach. The stator current is very smooth in the second approach compared with the first one, so in nearly sinusoidal, thereby minimizing the harmonics.

After the simulation tests and in order to verify the performance of the proposed methods, the experimental activities were carried out using the benchmark presented in Fig. 14, a simulation program and the laboratory setup with 0.415-KW permanent magnet synchronous motor drive and all the parameters used are the same as simulation study. A dc motor is mechanically coupled to the PMSM to serve as a load. The data acquisition system is developed utilizing dSPACE DS 1104. The board is equipped with analog-to-digital converters and digital-to-analog converters. A PC is used for software development and results visualization. The software is written in high-level language C.

Several tests are performed and the results are shown in Fig. 15a–f.

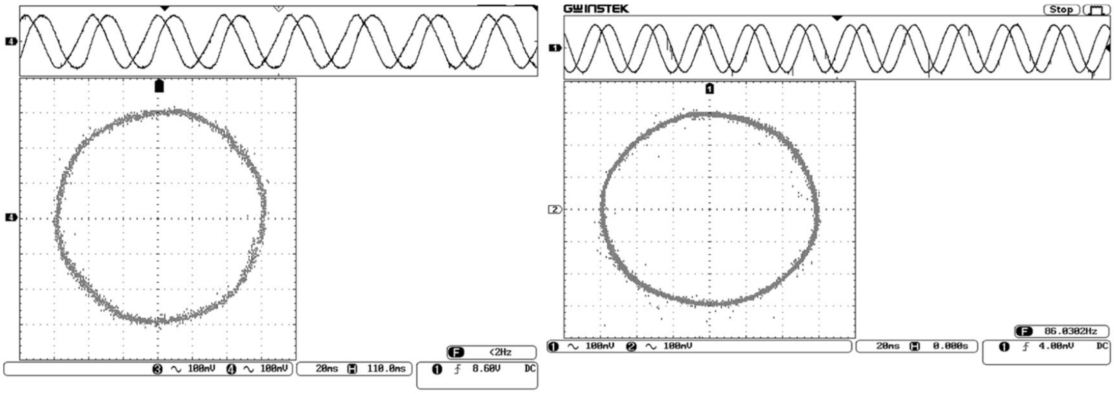


(a) Comparison graph of Mechanical speed (rpm)

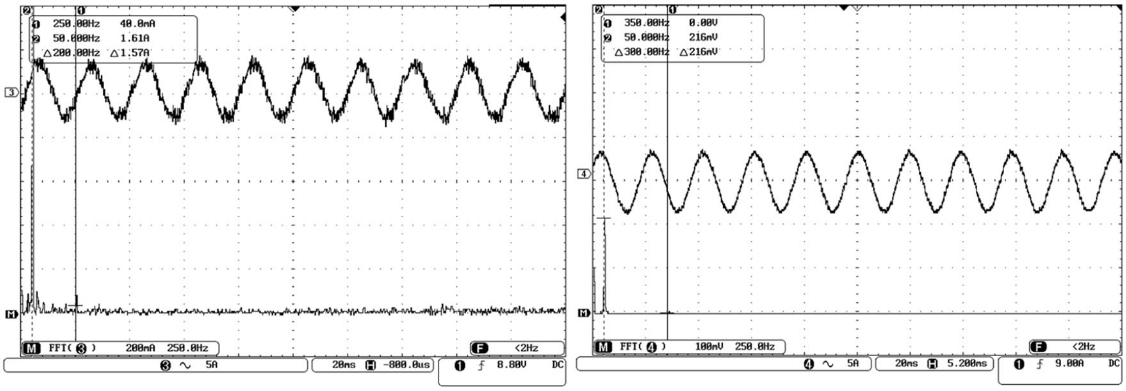
(b) Comparison graph of torque (Nm).

(c) Comparison graph of stator Flux

(d) Comparison graph of flux curve



(e) Comparison graph of trajectory of flux ($\phi_{s\alpha}$, $\phi_{s\beta}$)



(f) Comparison graph of spectrum of the current for phase a

Fig. 15 continued.

8 Conclusion

This paper describes the design and the implementation of a direct torque controlled PMSM, based on the Adaptive Fuzzy Logic Controller and the Particle Swarm Optimized methods. These advanced techniques are used in order to improve the performances of the DTC, in particular reduction of torque ripples, high dynamic response, and the switching frequency reduction. To show the performances of the proposed methods, simulations were performed using Matlab-Simulink. To validate the simulation results, a test bench has been constructed based on the DSP-1104.

The simulation and the experimental results show good performances for both methods compared to classical DTC. The DTC_PSO shows more performances than the DTC_AFLC, in particular the reduction of the torque and the flux ripples. An optimization scheme of the flux has been proposed to reduce the torque ripples. The optimization was tested using simulation and experimental results. The results show a reasonable improvement by flux optimization. The main improvements shown are the following:

- Reduction of torque, flux, and current ripples in transient and steady state response.
- No flux droppings caused by sector changes circular trajectory.
- Fast response of the stator flux in transient state.

Simulation and experimental results both show that compared with DTC_AFLC, the flux and torque ripples are reduced greatly in PMSM DTC based on particle swarm optimized control, and the switching frequency of the inverter is constant, improving the control performance of the system. The DTC_PSO controller enables the system to be of good fastness, no-overshoot, and very high accuracy, especially if it can enhance the ability of rejecting the load disturbance and takes on good robustness.

Appendix

Table 7 Main parameters of the prototype

PMSM parameters			
Number of pairs of poles	p		*2*
Nominal current	I_N		4.5A
Inductance *d*-axis Ld	Ld		43 mH
Inductance *d*-axis Lq	Lq		43 mH
PM flux linkage	ϕ_f		0.178 Wb
moment of inertia	J		85e-6 kgm2
PI_AFLC parameters			
K_{p0}	0.00220	K_p	0.00459
K_{i0}	0.06000	K_i	0.10000
PI_PSO parameters			
K_p		*1.0000e-002*	
K_i		*6.0000e-001*	
PSO parameters			
Acceleration coefficient related to pbest “cognitive acceleration”		C_1	2
Acceleration coefficient related to gbest “social acceleration”		2	2
The minimum of the weights		$w_{\min}$	0.4
The maximum of the weights		$w_{\max}$	0.9
Random number		R_1	[0,1]
Random number		R_2	[0,1]

References

1. Tulasi Ram Das G (2008) J. NTUCE Hyderabad, Introduction and control of permanent magnet synchronous machines. Workshop on ‘Modern AC Drives’ at RGMCET, Nandyala, Febrauary-2008
2. Finch JW, Giaouris D (2008) Controlled AC electrical drives. IEEE Trans Ind Electron 55(2):481–491
3. Rodriguez J, Kennel RM, Espinoza JR, Trincado M, Silva CA, Rojas CA (2012) High-performance control strategies for electrical drives: an experimental assessment. IEEE Trans Ind Electron 59: 812–820
4. Zhang Y, Akujuobi CM, Ali WH, Tolliver CL, Shieh L-S (2006) Load disturbance resistance speed controller design for PMSM. IEEE Trans Ind Electron 53(4):1198–1208
5. Dongemei X, Daokui Q, Fang X (2005) Design of H8 feedback controller and IP position controller of PMSM servo system. IEEE Int Conf Mechatron Autom 2(29):578–1108
6. Nash JH (1986) Direct torque control, induction motor vector control without an encoder. IEEE Trans Ind Appl 33(2):333–341
7. Idris NRN, Yatim AHM (2004) Direct torque control of induction machines with constant switching frequency and reduced torque ripple. IEEE Trans Ind Electron 51(4):758–767
8. Bossoufi B, Karim M, Ionita S, Lagrioui A (2010) Performance analysis of direct torque control (DTC) for synchronous machine permanent magnet (PMSM). In: Proc. IEEE-SIITME’2010, pp. 275–280, 23–26 Sep 2010, Pitesti, Romania
9. Kumar V, Gaur P, Mittal AP (2014) ANN based self tuned PID like adaptive controller design for high performance PMSM position control. Expert Syst Appl 41:7995–8002
10. Nyanteh YD, Srivastava SK, Edrington CS, Cartes DA (2013) Application of artificial intelligence to stator winding fault diagnosis in permanent magnet synchronous machines. Electr Power Syst Res 103:201–213
11. Labiod S, Guerra TM (2012) Fuzzy adaptive control for a class of nonlinear systems with unknown control gain. Evol Syst 3:57–64
12. Gao Y, Wang J, Qiu X (2011) The improvement of DTC system performance on fuzzy control. Procedia Environ Sci 10:589–594
13. HAZZAB A, BOUSSERHANE IK, ZERBO M, SICARD P (2006) Real time implementation of fuzzy gain scheduling of PI controller for induction motor machine control. Neural Process Lett 24:203–215
14. Hajebi P, Taghi AlModarresi SM (2012) Online adaptive fuzzy logic controller using genetic algorithm and neural network for networked control systems. ICACT Transactions on Advanced Communications Technology (TACT) 1(3)
15. Boudana D, Nezli L, Tlemcani A, Mahmoudi MO, Djemai M (2008) DTC based on fuzzy logic control of a double star synchronous machine drive. Nonlinear Dyn Syst Theory 8(3):269–286
16. Al Gizia A, Mustafaa MW, Jebur HH (2014) A novel design of high-sensitive fuzzy PID controller. Appl Soft Comput 24:794–805
17. Maldonado Y, Castillo O, Melin P (2013) Particle swarm optimization of interval type-2 fuzzy system for FPGA applications. Appl Soft Comput 13:496–508
18. Shahnazi R, Shanechi HM, Pariz N (2008) Position control of induction and DC servomotors: a novel adaptive fuzzy PI sliding mode control. IEEE Trans Energy Convers 23(1)
19. Sangram K, Routray NN, Pravat Kumar RA (2012) Robust fuzzy sliding mode control design for current source inverter based STATCOM. Appl Procedia Technol 4:342–349
20. Singh M, Chandra A (2011) Application of adaptive network-based fuzzy inference system for sensor less control of PMSG-based wind turbine with nonlinear-load-compensation capabilities. IEEE Trans Power Electron 26(1):165–175
21. Bhattacharya A, Chakraborty C (2011) A shunt active power filter with enhanced performance using ANN based predictive and adaptive controllers. IEEE Trans Ind Electron 58(2):421–428
22. Kashif SA, Saqib MA (2014) Sensorless control of a permanent magnet synchronous motor using artificial neural network based

estimator—an application of the four-switch three-phase inverter. Electr Power Compon Syst 42(1)

23. Zhan Z-H, Zhang J, Li Y, Chung HS-H (2009) Adaptive particle swarm optimization. IEEE Trans Syst Man Cybern B Cybern 39: 1362–1381
24. Kiranyaz S, Ince T, Yildirim A, Gabbouj M (2009) Fractional particle swarm optimization in multidimensional search space. IEEE Transactions on: Accepted for future publication Vol. pp 1–1, Forthcoming
25. Vasumathi B, Moorthi S (2012) Implementation of hybrid ANN–PSO algorithm on FPGA for harmonic estimation. Eng Appl Artif Intell 25:476–483
26. Pourjafari E, Mojallali H (2011) Predictive control for voltage collapse avoidance using a modified discrete multi-valued PSO algorithm. ISA Trans 50:195–200
27. Mohkami H, Hooshmand R, Khodabakhshian A (2011) Fuzzy optimal placement of capacitors in the presence of nonlinear loads in unbalanced distribution networks using BF-PSO algorithm. Appl Soft Comput 11:3634–3642
28. Oh S-K, Jang H-J, Pedrycz W (2011) A comparative experimental study of type-1/type-2 fuzzy cascade controller based on genetic algorithms and particle swarm optimization. Expert Syst Appl 38: 11217–11229
29. Chiou J-S, Tsai S-H, Liu M-T (2012) A PSO-based adaptive fuzzy PID-controllers. Simul Model Pract Theory 26:49–59
30. Bouallegue S, Haggege J, Ayadi M, Benrejeb M (2012) PID-type fuzzy logic controller tuning based on particle swarm optimization. Eng Appl Artif Intell 25:484–493
31. Takahashi I, Noguchi T (1986) A new quick response and high-efficiency control strategy of an induction motor. IEEE Trans Ind Appl IA-22(5):820–827
32. Boldea I (2000) Direct torque and flux control (DTFC) of A.C. drives: a review. In: Proceedings of EPEPEMC'2000, vol. 1, Kosice, Slovakia, pp 88–97
33. Pacas M, Weber J (2005) Predictive direct torque control for the PM synchronous machine. IEEE Trans Ind Electron 52(5):1350–1356
34. Gatto G, Marongiu I, Serpi A, Perfetto A (2008) A predictive direct torque control of induction machines. In: Proceedings of SPEEDAM 2008, June 2008, Ischia, Italy, pp 1103–1108
35. Sharmeela C, Mohan MR, Uma G, Baskaran J (2007) Fuzzy logic based controlled three phase shunt active filter for line harmonics reduction. J Comput Sci 3(2):76–80
36. Mikkili S, Panda AK (2012) Real-time implementation of power theory using FLC based shunt active filter with different fuzzy M.F.s, 38th Annual Conference on IEEE Industrial Electronics Society (IECON 2012) pp 702–707
37. Benchouia MT, Ghadbane I, Golea A, Srairi K, Benbouzid MH (2014) Implementation of adaptive fuzzy logic and pi controllers to regulate the DC bus voltage of shunt active power filter. Appl Soft Comput 56:1826–1838
38. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, vol 4, Perth, Australia, pp 1942–1948
39. Marinakis Y, Marinaki M, Dounias G (2010) A hybrid particle swarm optimization algorithm for the vehicle routing problem. Eng Appl Artif Intell 23:463–472
40. Samanta B, Nataraj C (2009) Use of particle swarm optimization for machinery fault detection. Eng Appl Artif Intell 22:308–316
41. Liu L, Liu W, Cartes DA (2008) Particle swarm optimization-based parameter identification applied to permanent magnet synchronous motors. Eng Appl Artif Intell 21:1092–1100
42. Venayagamoorthy GK, Smith SC, Singhal G (2007) Particle swarm-based optimal partitioning algorithm for combinational CMOS circuits. Eng Appl Artif Intell 20:177–184
43. Bouallegue S, Haggege J, Benrejeb M (2010a) Structured loop-shaping HN controller design using particle swarm optimization. In: Proceedings of the 2010 I.E. International Conference on Systems, Man, and Cybernetics SMC'10, Istanbul
44. Bouallegue S, Haggege J, Benrejeb M (2010b) Structured mixed-sensitivity HN design using particle swarm optimization. In: Proceedings of the 7th IEEE International Multi-Conference on Systems, Signals and Devices SSD'10, Amman
45. Bouallegue S, Haggege J, Benrejeb M (2011) Particle swarm optimization-based fixed-structure H^{∞} control design. Int J Control Autom Syst 9(2):258–266
46. Ray RN, Chatterjee D, Goswami SK (2009) An application of PSO technique for harmonic elimination in a PWM inverter. Appl Soft Comput 9:1315–1320
47. Kao C-C, Chuang C-W, Fung R-F (2006) The self-tuning PID control in a slider–crank mechanism system by applying particle swarm optimization approach. Mechatronics 16:513–522
48. Wain R-J, Lin Y-F, Chuang K-L (2014) Total sliding-mode-based particle swarm optimization control for linear induction motor. J Frankl Inst 351:2755–2780