An Improved PMSM Drive Architecture Based on BFO and Neural Network

Regular Paper

Flah Aymen1,* and Sbita Lassâad1

1 National School of Engineering in Gabes, Tunisia
* Corresponding author E-mail: flahaymening@yahoo.fr

Received 17 Feb 2012; Accepted 3 Sep 2012

DOI: 10.5772/53011

© 2013 Aymen and Lassâad; licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract In this paper, an improved robust vector control strategy is designed to drive the Permanent magnet synchronous motor in a wide speed range mode. The designed control method guarantees the precision and robustness of speed regulation performance by using recurrent neural network architecture. The stator current controller parameter tuning problems, which characterize this control strategy, are resolved using a bacterial foraging optimization algorithm to find the optimal parameters of the current controllers used. A field weakening control algorithm generates an adaptive magnetizing current command to achieve the desired high speed mode. The robustness and effectiveness of the global control scheme are verified through computer simulations established under a Matlab-Simulink environment.

Keywords Motor Speed Drive, PMSM, Vector Control, High Speed, BFO Algorithm, Recurrent Neural Network

1. Introduction

Robotics has been of interest to mankind for many years. Nowadays, the use of robots is wide-spread in the manufacturing industry, the military, space exploration, transportation and medical applications. A consideration of performance robustness, rapidity and precision is required for many of these applications. An autonomous car can be considered as robotic or self-driving [1]; an electrical vehicle can be used in this robot application [2] for its efficiency, simplicity and ease of maintenance, and especially for its role in environmental protection. Recent research has therefore focused on studies of these vehicles to improve drive performance.

Many electrical motor types are used in these vehicles. Zhu [3] made a comparative study on these electrical machines and drives. In his approach, induction machines, switched reluctance machines and permanent magnet synchronous motors are compared. The obtained results prove that the permanent magnet synchronous motor is more efficient than the other types.

Based on this motor type, much control research has sought to ensure a high speed control phenomenon. Several control strategies have been proposed and tested for controlling this device, such as vector control or direct torque control, as presented in [4], [5] and [6]. The fast torque, speed responses, and especially the simple architecture make the vector control strategy most popular in these applications; however, the robustness
problems are already present in this strategy, especially under motor parameter variations.

Various solutions have been suggested for these problems. The main idea is to change or replace the three commonly used PI controllers by neuron network, fuzzy and neuro-fuzzy controllers, as presented in [7], [8], [9] and [10], or controllers characterized by robustness, as in the IMC or MRAC methods presented respectively in [11] and [12]. The efficiency of the neural network approach has been proven, as well as its robustness and precision in various applications, as presented in [13] and [6]. Therefore, this technique is implemented in this application to increase the system outer loop control performance.

In this work, we seek to build a robust PMSM control loop which can be operated in a wide speed range and which is useful in relation to the problems that can appear in this speed region with PMSM parameters variations due to external effects as high temperatures, vibrations or dust. To resolve this problem, the recurrent neural network has proved advantageous and has been used to replace the conventional PI speed controller.

The other two current controllers used are characterized by their parameter tuning problems. These values can influence system robustness and efficiency, and optimum PI parameters are therefore of the utmost importance. Many methods are presented in the literature to find these values, such as particle swarm optimization (PSO) [14]-[15] and bacterial foraging optimization (BFO) [16] algorithms. In [17], the author proves the precision of BFO; we have therefore used this method to adjust the desired current’s PI parameters.

As this application needs a high speed running mode, the field weakening operation is implemented based on Morimoto’s approaches presented in [18] and [19].

Consequently, the obtained new control scheme is implemented in the MATLAB/Simulink environment and tested in a wide speed range. Robustness tests are executed to verify the system performance.

This paper is organized as follows. After a general introduction, the second section describes the proposed high speed algorithm (HAS). The recurrent neural network speed controller architecture is given in the third part. The bacterial foraging optimization algorithm is described in general terms in the fourth part, and the simulation results are then presented and discussed. Finally, the conclusion is presented.

2. High speed algorithm

The establishment of the high speed control algorithm is based on the PMSM mathematical model. The dynamic properties of a PMSM can be described by a set of nonlinear differential equations linking the stator currents and voltages with the mechanical quantities torque, speed and angular position. These mathematical expressions can be formulated as in Eq (1) and Eq (2), which present, respectively, the stator voltage and the electromagnetic torque expressions in the rotating (d, q) reference frame. The mechanical expression can be written as presented in Eq (3) [14].

\[
\begin{cases}
v_d = R_s i_d + L_s \frac{di_d}{dt} - \omega L_q i_q \\
v_q = R_s i_q + L_s \frac{di_q}{dt} + \omega L_d i_d + \omega \lambda_m
\end{cases}
\]

(1)

\[
\begin{align*}
T_e &= \frac{3}{2} P \left( \lambda_d i_q - \lambda_q i_d \right) \\
\lambda_d &= L_d i_d + \lambda_m \\
\lambda_q &= L_q i_q
\end{align*}
\]

(2)

and

\[
\left( T_e - T_l \right) = \left( \frac{P}{2} \right) \left( \frac{d \omega}{dt} + f \omega \right)
\]

(3)

In the PMSM, the flux is mainly produced by the rotor side magnet and denoted \( \lambda_m \). In the field-oriented control strategy and within the rated speed regime, the direct stator current \( i_d \) is maintained equal to zero. However, to go beyond the rated regime, the total flux is minimized following the field weakening strategy (FW).

The direct stator field component depends principally on the direct stator current, stator inductance and the mainly permanent magnet flux. However, the quadrature element depends on the quadrature stator current, which is generally introduced by the electromagnetic torque demand. Therefore, to minimize the total flux, it is evident that the direct component flux must also be minimized. This occurs if the direct stator current goes to the negative region \( i_d < 0 \). However, it is necessary to fix the minimization limitations to avoid the magnet demagnetization problems [18].

Therefore, the field weakening principle is described in Fig. 1, which presents three zones. The first gives the current limitation as described in the first part of Eq (4). The second is the ellipse form given to the voltage limitation, as indicated in the second part of Eq (4). The field weakening region is illustrated by the hatched area in the circle, which designs the interception between the two last zones. Generally, the current limitation circle is centred on zero and its radius depends on the inverter maximum current. Whether the limitation voltage shape is an ellipse is dependent on \( L_d \neq L_q \) or \( L_d = L_q \).
respectively. The voltage limitation curve characteristics are essentially defined by the centre point, which depends on the magnet flux and the stator inductance values. The radius value depends also on the target speed. If the speed increases, the radius value is reduced. The maximums of current and voltage are usually set by the inverter [20]. These two constraints can be expressed by the system of equations in (4) [19], where \( i_{\text{max}} \) and \( V_{\text{max}} \) are respectively the maximum inverter phase-current and phase-voltage amplitudes. After the tow inverter limit conditions are applied in the high speed range, shown in Eq (4), and when all the PMSM parameters are taken into account, a new direct stator currents expression can be established, which is presented in Eq (6). To avoid the demagnetization problems, it is necessary to define the direct stator current limit condition, which can be obtained if the quadrature component is equal to zero. The direct stator current limit is then expressed in Eq (5).

![Field weakening operation under inverter voltage and current limits](image)

**Figure 1.** Field weakening operation under inverter voltage and current limits

\[
\begin{align*}
\begin{cases}
I_d^2 + I_q^2 & < I_{d,\text{max}}^2 \\
V_d^2 + V_q^2 & < V_{d,\text{max}}^2
\end{cases}
\end{align*}
\]

\[ i_{\text{d,lim}} \bigg|_{i_q=0} = \frac{-b - \sqrt{\Delta}}{2a} \]   

where  

\[
\Delta = 4V_{\text{max}}^2 \left[ R_s^2 + \alpha^2L_d^2 \right] - 4\alpha^2R_s^2L_m^2 \\
b = 2L_d\alpha^2L_m \\
a = R_s^2 + \alpha^2L_d^2
\]

The direct reference stator current is:

\[
i_d^* = \sqrt{\frac{V_{\text{max}}^2 - R_s^2I_{\text{q,lim}}^2 - 2R_sL_i\omega}{\alpha^2L_s} - \lambda_m}
\]

3. Recurrent Neural Networks Speed Controller

Recurrent neural networks (RNN) have been an interesting and important part of neural network research since the 1990s. Much architecture is presented in the literature, characterized sometimes by total connection and sometimes by partial connection. In this work, the recurrent neural networks speed controller (RNNSC) is proposed and its architecture is given in Fig. 2. The structure corresponds to three neurons at input layer, two neurons in hidden layer and one in output layer. \( S_j(k) \) represents the input signals of the hidden layer, \( X_j(k) \) is the output signal of the hidden layer, \( i_q^* \) is the final output signal of the RNNSC and \( I(k) \) is the input vector. \( W_{ij}(k) \) is the weight between the same neuron in the hidden layer, \( W_{ij}(k) \) is the weight between two successive neurons in the hidden layer, \( W_i(s) \) is the weight from the input layer to the hidden layer and \( W_i(s) \) is the weight from the hidden layer to the output layer. The sigmoid activation function is used in the two first layers and the linear function on the output layer.

![Recurrent neural network speed controller architecture](image)

**Figure 2.** Recurrent neural network speed controller architecture

The mathematical relations between the different neurons in the different layers are given by Eqs (7), (8) and (9):

\[
S_j(k) = W_{ij}(k)X_j(k-1) + \sum_{j=1}^{m} W_{ij}(k)I_j(k) \\
+ W_{ij}(k)X_{j-1}(k-1)
\]  

\[
X_j(k) = f(S_j(k)) = \frac{1}{1 + \exp(-S_j(k))}
\]

\[
i_q^*(k) = \sum_{j=1}^{n} W_{ij}^0(k)X_j(k)
\]

The gradient method is used to adjust the weights and the mathematical expressions can be presented as in Eq (10). More details are given in [14]. \( \eta \) represents the RNNSC learning rate.

\[
\begin{align*}
W(k+1) &= W(k) + \Delta W(k) = W(k) - \eta(k) \frac{\partial E(k)}{\partial W(k)} \\
E(k) &= \frac{1}{2} \left[ i_q^*(k) - i_q^*(k) \right]^2
\end{align*}
\]
In the chemotaxis step, the bacterium is said to be swimming if it moves in a predefined direction, and tumbling if it moves in an altogether different direction. The bacterial position is updated mathematically by Eq (11).

$$\theta(i+1,k,l) = \theta(i,k,l) + C(i) \frac{\Delta s(i)}{\sqrt{\Delta s^2(i) + \Delta r(i)}}$$ (11)

5. Simulations And Results

To verify the effectiveness of the proposed high speed PMSM control algorithm, a digital simulation based on the Matlab/Simulink software package has been carried out. The PMSM characteristics are given in Tab. 2.

The entire proposed control scheme, built as connected subsystems, is illustrated in Fig. 5. The first subsystem is the recurrent neural network speed controllers (RNNSC) used for online generating the quadrature reference stator current $$i_q^*$$. The high speed algorithm is used for online generating the direct reference stator current $$i_d^*$$. Once the reference and actual quadrature stator currents have been compared, the PI-QSC controller will generate the quadrature reference stator voltage $$v_q^*$$. A PI-DSC is used for the d side to generate the direct reference stator voltage $$v_d^*$$, as a response of the processed direct stator current error as a controller input signal. Park and Clark transformations are added to build the three corresponding reference voltages, $$v_d^*$$, $$v_q^*$$ and $$v_C^*$$.

A Pulse Width Modulation (PWM) subsystem is driven by the voltage source inverter feeding the PMSM.

After obtaining the speed controller architecture, the remaining control problem is with the PI current controller’s parameters, which must be given. In this control scheme, these PI parameters are established by trial and error in two different cases, as presented in Tab. 1. The influence of these PI parameters, first on the direct and quadrature stator current responses and then on the speed behaviour, is clearly shown in Fig. 7, Fig. 8 and Fig. 9. There is clearly a large oscillation in the reported signals. The stator current components’ worst control deteriorates the PMSM regulation performances and power efficiency. This is of great importance in robots’ electric drive systems. Therefore, in order to minimize these oscillations, an intelligent optimization method based on the BFO architecture is performed here, in order to help in adjusting these PI-DSC and PI-QSC parameters for a high-performance PMS drive.

The BFO tuning algorithm needs to start with data such as: bacterium number: $$s = 8$$, chemotaxis steps number: $$Ne = 4$$, swim length: $$Ns = 4$$, reproduction steps number: $$Nre = 2$$, elimination-dispersal events number: $$Ned = 2$$, number of bacteria reproductions (splits) per generation $$Sr = s/2$$ and the probability that each bacteria will be

---

**Figure 3.** Learning database for the RNNSC

After the training process, using the database presented in Fig. 3 it can be evaluated whether the RNNSC is then able to achieve the desired speed. In this case and for 200 learning epochs, the obtaining learning error is approximated as equal to 1.10e-10. This last is sufficient enough to be relied upon in the designed RNNSC.

4. Bacteria Foraging Optimization: A Brief Overview

Many theories for solving the optimization problems have been presented in the literature, such as the particle swarm optimization and genetic algorithms [17]. All these methods are derived from the context of natural behaviour and environment. In bacteria behaviour law, all the elements support the species with the best ability to search for food and eliminate others. Three processes characterize this algorithm as presented in [21] and in [21] Passion gives the corresponding optimization theory method.

The bacterial foraging optimization algorithm is illustrated in Fig. 4. $$J(i, j, k, l)$$ denotes the function cost value and $$\theta(i, j, k, l)$$ the i-th bacterium position. In the present algorithm, the most important loop is the chemotaxis, designed by “B”. Here, each bacterium passes through the tumbling and swimming step “F”, where the corresponding objective value will be calculated and compared to the less one saved one. After making sure that all the bacteria, “G”, are executing the entire chemotaxis step, “B”, reproduction begins; i.e., the half with the highest objective function cost dies, while the half with the lowest cost function divides. The location of the obtained copies is the same as their parent. After reproduction, “elimination-dispersal” begins. With a probability $$P_{ed}$$, elimination and dispersion occurs for each bacterium (this keeps the number of bacteria in the population constant) [21] and [22]. The elimination and dispersal events assist chemotaxis progress by aligning the bacteria with the nearest required values. Each bacterium, according to a fixed probability, is dispersed from its original position and moves to the best position within the search space. These events may prevent local optima trapping but lead to disturbance of the optimization process.
eliminated/dispersed: Ped = 0.25. Also: run length c(:, 1) = 0.05 * ones(s, 1). The initial PI-QSC and PI-DSC parameters [Kp, Ki, Kp, Ki] are randomly chosen and the desired fitness function is presented in Eq (12). Here, the integral square error ISE method is used and α=β=100.

\[
\begin{align*}
\text{min} f &= \text{min} f_{\text{DSC}} + \text{min} f_{\text{QSC}} \\
\text{min} f_{\text{DSC}} &= \alpha_1 (c_{\text{DSC}}^2 + \beta_1 \times \Delta_{\text{DSC}}) \\
\text{min} f_{\text{QSC}} &= \alpha_1 (c_{\text{QSC}}^2 + \beta_1 \times \Delta_{\text{QSC}})
\end{align*}
\] (12)

\[
\theta(j+1,k,l) = \theta(j+1,k,l) + C(i) \frac{\Delta x(i)}{\sqrt{\Delta x^2(i) + \Delta x(i)}}
\]
While the algorithm runs, the “s” bacteria search for the best positions, i.e., those with the minimum of fitness cost. These positions will be transferred to the other bacteria so that they adjust their own places. At the end of the iterations, the obtained positions, which are equal to the minimum cost function, are chosen as the PI-DSC and PI-QSC proportional and integral parameters.

It is important to indicate that, normally four cases would be shown, because there are two reproduction steps and two elimination steps. Here, for each elimination phase there are two reproduction steps, as shown in Fig. 6. Four colours are shown in the diagram, each corresponding to a chemotaxis step for the bacteria.

Fig. 6 shows the evolution of the fitness function cost and the corresponding PI parameters proportionally to the bacteria number. The case of the PI-DSC tuning parameters is presented in Fig. 6a, Fig. 6c and Fig. 6e; the first presents the cost fitness function and the others present the integral and the proportional PI-DSC gain coefficients, respectively. The case of the PI-QSC tuning parameters is illustrated in Fig. 6b, Fig. 6d and Fig. 6f.

For the case of the PI-DSC tuning parameters, the best fitness cost function corresponds to the green colour for bacteria number two, where the corresponding PI parameters are shown in Tab.1.

In the PI-QSC case, there appears to be a strongly unstable and oscillating fitness cost signal response (Fig. 6b). The BFO algorithm fails to give optimum values for all the bacteria used. However, some of these particles, such as the bacteria number three, four and five, reach the best positions. These positions are chosen and used as PI-QSC tuning parameters. It is important to note that these positions will be the target for the other bacteria after more running time, and after the reproduction and elimination steps number in the BFO architecture has been modified.

After these new PI parameter values are applied in the global control algorithm, the novel stator currents results are obtained, which are illustrated in Fig. 7 and Fig. 8. These show good performances, especially when a stator resistance variation is applied at t=1sec in the high speed region. These results prove the current stability in this speed zone and the effectiveness of the BFO algorithm in torque ripple minimization. The efficiency of this algorithm is also shown in the speed results presented in Fig. 9, where the oscillations are minimized.
Once the desired speed controller is designed and the optimum PI-QSC and PI-DSC parameters are tuned, the global control scheme performances will be tested in relation to robot trajectory and drive tracking. A wide speed range is considered, with the appliance of the PMSM load torque dynamics and parameter variations.

The applied target variable speed is changed at 0 rpm to the rated value (2500 rpm), then from the rated value to a high speed of 4000 rpm. In this case, the load torque applied to the motor is in the context of realistic behaviour of a constant power high speed regime, where, the load torque decreases if the desired speed exceeds the rated one. Abrupt load torque variations of 0.1 N.m are also added and removed several times. The motor working regimes imply parameter variation, which mainly consists of stator resistance variations. At 1 second a rise of 100% of stator resistance is introduced as an abrupt variation.

![Figure 7. Direct stator current results with three PI-DSC parameters tuning](image)

![Figure 8. Quadratic stator current results with three PI-QSC parameters tuning](image)

### Table 1. PI-DSC and PI-QSC parameters

|                  | PI-DSC | PI-QSC |
|------------------|--------|--------|
|                 | $K_p$  | $K_i$  | $K_p$  | $K_i$  |
| Case1            | 0.1    | 100    | 100    | 10     |
| Case2            | 10     | 100    | 1      | 10     |
| BFO              | 300.4  | -0.16  | 299.7  | 0.38   |

As previously cited, the first robustness test is carried out for load torque variations. Fig. 10 shows the speed behaviour results. The load torque dynamics appliance is given as:

At rated speed range:
- At $t=0.15s$, $T_l(t)=T_{lrated}+0.1$
- At $t=0.22s$, $T_l(t)=T_l(t)-0.1$

At high speed range:
- At $t=0.8s$, $T_l(t)=T_{min}+0.1$

The obtained speed and stator current results shown in Fig. 1a and Fig. 11 verify the robustness of the proposed high speed control algorithm. The PMSM parameters are highly sensitive to the motor operation regime conditions, which induce a high temperature or hard vibration. Therefore, the robustness of the proposed control scheme against these variations is required. The application of a stator resistance variation in the high speed mode is appropriate, and the system behaviour is verified. This PMSM parameter variation is applied at $t=1s$, as indicated above. The effectiveness of the proposed high speed PMSM vector control architecture is verified and highlighted in the speed zooms shown in Fig. 10, and illustrated by the stator current results in Fig. 11.
As presented in Fig. 10, two speed regions are obtained. From $t=0\text{sec}$ to $t=0.58\text{sec}$, the rated speed zone is illustrated: it is clear that the direct stator current in Fig. 11 is equal to zero. However, if the desired speed exceeds the nominal one for the high speed zone, changing from $t=0.58\text{sec}$ to $t=1.2\text{sec}$, the direct stator current command is reduced to the negative region. The field weakening principle can be simultaneously shown from the direct stator current and the flux dynamic results. The flux magnitude is reduced and moves towards the weakening regime in terms of circle radius, as presented in Fig. 12.

Field weakening results (HSA):

In the stationary frame “α, β”, the stator flux trajectory can be proven by the circle radius presented in Eq (13). If the direct stator current $i_d$ is reduced in the high speed case the circle radius will be minimized to $R_2$.

$$\begin{align*}
\lambda_\alpha^2 + \lambda_\beta^2 &= R^2 \\
R^2 &= (L_{q}i_q + \lambda_m)^2 + (L_{d}i_d + \lambda_m)^2 
\end{align*}$$

Table 2. PMSM parameters and rated values

| Parameter                  | Value       |
|----------------------------|-------------|
| Power                      | 400W        |
| Rated current ($I_{\text{rated}}$) | 2A          |
| Maximum current ($I_{\text{max}}$) | 5A          |
| Rated speed ($\omega_{\text{rated}}$) | 2500rpm     |
| Rated torque ($T_{\text{rated}}$) | 1.5N.m     |
| Stator inductance ($L_s$)   | 0.00215 H   |
| Stator resistance ($R_s$)   | 5.2Ω        |
| Permanent flux ($\lambda_m$) | 0.24Wb      |
| Inertia factor ($J$)        | 8.5e-6kg.m² |

6. Conclusion

This paper has outlined a robust PMSM control design, based on recurrent neural network and the BFO algorithm. In this application, the used motor must be operated in a wide speed range and especially in the high speed mode. For this purpose, a global control and high performance drive scheme has been built and described. Based on the robustness of the recurrent neural network, the vector control strategy is modified, whereby the conventional PI speed controller is replaced by a recurrent neural architecture.

This study has demonstrated the importance of the two current controllers used in this control strategy in terms of the system performance. To improve drive characteristics, an intelligent optimization method based on the BFO algorithm has been designed for tuning the direct and quadrature stator current controllers. The importance of this algorithm is clearly shown on the stator current and on the speed behaviour results.
In the robustness study, the global control scheme proved its efficiency in two different tests. In the first, a load torque was applied in rated and high speed modes. The second test investigated the robustness of the control schemes when the stator resistance of the PMSM was varied.

7. Nomenclature

| Abbreviation | Description |
|--------------|-------------|
| PMSM | Permanent magnet synchronous motor |
| RNNSC | Recurrent neural network speed controller |
| BFO | Bacterial foraging optimization |
| HSA | High speed algorithm |
| \(v_d, v_q\) | Direct and quadrature stator voltages |
| \(\lambda_d, \lambda_q\) | Direct and quadrature stator fields |
| \(i_d, i_q\) | Direct and quadrature stator currents |
| \(L_d, L_q\) | Direct and quadrature stator inductions |
| \(\omega\) | Rotor speed |
| \(\lambda_m\) | Permanent magnet flux |
| \(R_s\) | Stator resistance |
| \(T_i\) | Load torque |
| \(P\) | Pairs poles number |
| \(J\) | Rotor inertia coefficient |
| \(f_{soc} \text{ and } f_{soc}\) | Fitness function for DSC and QSC |
| \(e_{soc} \text{ and } e_{soc}\) | Direct and quadrature stator current error |
| \(\Delta_{soc} \text{ and } \Delta_{soc}\) | Overshoot in DSC and DQC cases |
| PI-DSC | PI-Direct stator current |
| PI-QSC | PI-Quadratique stator current |

8. References

[1] Frazzoli E, Dahleh M.A., Feron E (2000) Robust hybrid control for autonomous vehicle motion planning, 39th IEEE Conference on Decision and Control, Vol. 1, pp. 821–826.

[2] Bormann F, Braune E, Spitzner M (2010) The C2000 autonomous model car, 4th European Education and Research Conference (EDERC), pp. 200–204.

[3] Zhu Z.Q, Howe D (2007) Electrical machines and drives for electric vehicle, hybrid and fuel cell vehicles, Proceedings of IEEE Transaction Vol. 95, No. 4, pp. 746–765.

[4] Aguirre M, Calleja C, Lopez-de-Heredia A (2011) FOC and DTC comparison in PMSM for railway traction application, European Conference on Power Electronics and Applications, pp. 1–10.

[5] Boulghasouli Z, Elbaca A, Elwarraki E, Yousfi D (2011) Combined Vector Control and Direct Torque Control: an experimental review and evaluation, Multimedia Computing and Systems (ICMCS), pp. 1–6.

[6] Gerasimos G. Rigatos and Pierluigi S (2011), Sensorless Control of Electric Motors with Kalman Filters: Applications to Robotic and Industrial Systems, International Journal of Advanced Robotic Systems, Vol. 2, No. 6, pp. 62–80.

[7] Cao X and Fan L (2006) Vector Controlled PMSM Drive Based on Adaptive Neuro-fuzzy Speed Controller, Intelligent Control and Automation Lecture Notes in Control and Information Sciences, Vol. 344, pp.616–622.

[8] Wang Z, Ahn T (2006) Design of the Optimal Self-tuning Fuzzy-PI Controller for Rectifier Current Control in HVDC, System Control Conference, pp. 1224–1227.

[9] Douiri M.R, Cherkaooui M, Nasser T, Essadki A (2011) A neuro fuzzy PI controller used for speed control of a direct torque to twelve sectors controlled induction machine drive, International Conference on Multimedia Computing and Systems (ICMCS), pp. 1–6.

[10] Rodriguez-Martinez A, Garduno-Ramirez R, Vela-Valdes L.G (2011) PI Fuzzy Gain-Scheduling Speed Control at Startup of a Gas-Turbine Power Plant, IEEE Transactions on Energy Conversion, Vol. 26, No. 1, pp. 310–317.

[11] Kim K.H (2011) Simple on-line compensation scheme of nonlinearity in inverter-fed PMSM drive using MRAC and co-ordinate transformation, Electronics Letters, Vol. 47, No. 8, pp. 514–515.

[12] Sun L, Fan X, Liu H (2011) IMC-fed PMSM control system based on fuzzy PI, IEEE International Conference on Computer Science and Automation Engineering (CSAE), Vol. 3, pp. 489–492.

[13] Ehsan S and Soltanpour M (2009), Robust Neural Network Control of Electrically Driven Robot Manipulator using Backstepping Approach, International Journal of Advanced Robotic Systems, Vol. 6, No. 4, pp. 285–292.

[14] Flah A et al. (2011) A Robust PMSM Speed Control using an Artificial and a Recurrent Neural Network, International Journal of Research and Reviews in Computer Science (IJRRCS), Vol. 2, No. 4, pp. 1023–1028.

[15] Mehdi H and Boubaker O (2011) Impedance Controller Tuned by Particle Swarm Optimization for Robotic Arms, International Journal of Advanced Robotic Systems, Vol. 2, No. 5, pp. 93–103.

[16] Flah A, Sbita L (2012), BFO control tuning of a PMSM high speed drive, 16th IEEE Mediterranean Electrotechnical Conference (MELECON), pp. 273–276.

[17] Edelman D, Hochreiter R, Brabazon A (2010) Swarm intelligence-based stochastic programming model for dynamic asset allocation, IEEE Congress on Evolutionary Computation (CEC), pp. 1–8.

[18] Morimoto S, Sanada M and Takeda Y (1994) Wide speed operation of interior permanent magnet synchronous motors with high performance current regulator, IEEE Trans. on Industry Applications, Vol. 30, No. 4, pp. 920–926.

[19] Morimoto S, Takeda Y, Hirasa T (1990) Expansion of operating limits for permanent magnet motor by current vector control considering inverter capacity, IEEE Transaction. Industry Application Vol. 26, No. 5, pp. 866–871.
[20] Flah A, Sbita L, Ben hamed M (2011) Online MRAS-PSO PMSM parameters estimation, International review on modeling and simulation IREMOS, Vol. 4, No. 3, pp. 980–987.

[21] Abraham A, Konar A (2009) On Stability of the Chemotaxis Dynamics in Bacterial-Foraging Optimization Algorithm, IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans, Vol. 39, No. 3, pp. 670–679.

[22] Passino K.M (2002) Biomimicry of bacterial foraging for distributed optimization and control, IEEE Control Systems Magazine, Vol. 22, No. 3, pp. 52–67.