J. Basic. Appl. Sci. Res., 2(2)1441-1450, 2012
© 2012, TextRoad Publication
ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

Analysis of Intelligent Control Methods for Efficiency Optimization of Induction Motor Drive

Y. Hoseynpoor, T. PirzadehAshraf, P. MosadeghArdabili

Department of Electrical Engineering, Ardabil Branch, Islamic Azad University, Ardabil, Iran

ABSTRACT

This paper presents a review of the developments in the field of efficiency optimization of three-phase induction motor through optimal control and design techniques. Particle Swarm Optimization and Genetic Algorithms have been used for efficiency optimization of induction motor drives. The proposed techniques minimize the input power which means increasing the motor efficiency by selecting the optimum reference flux. Simulation results show that the PSO method has better response than the GA method.
KEYWORDS*:* Intelligent Control, Induction motors, Motor Efficiency, Reference Flux, Genetic Algorithm, Particle Swarm Optimization.

I. INTRODUCTION

Induction motor is an electromechanical device that converts electrical energy to mechanical energy, Field oriented vector control drives are capable of both speed and torque regulation because they control both current components and the angle (vector sum) between them. They provide excellent torque characteristics plus tighter speed regulation. These types of drives give independent torque and flux control and allowing a continuous regulation of the motor speed and torque. These types of drives give the best performance in controlling the AC motors. In recent years, optimization algorithms have received increasing attention by the research community as well as the industry to solve various complex control problems as an alternative or complement to the conventional methods [1]. Optimization techniques using analogy of swarming principle have been adopted to solve a variety of engineering problems in the past decade. Swarm Intelligence (SI) is an innovative distributed intelligent paradigm for solving optimization problems that originally took its inspiration from the biological examples by swarming, flocking and herding phenomena in vertebrates. A population of particles exists in the n-dimensional search space in which the optimization problem lives in. Each particle has a certain amount of knowledge, and will move about the search space based on this knowledge. The particle has some inertia attributed to it and so it will continue to have a component of motion in the direction it is moving [2]. Parsopoulos and Vrahatis attempted to improve the search efficiency in PSO by performing two stage transformation of the objective function which eliminates and elevates the neighborhood of the local minima [3]. Alternative runs and tumbles in Ecoli bacteria found in the human intestine constitute chemo taxis and this foraging mechanism was imitated by Kevin Passino for solving optimization problem in control system [4]. In the earlier PSO algorithms, each particle of the swarm is accelerated by its best previous position and towards the best particle in the entire swarm. Here, the underlying assumption is that each particle in the swarm remembers the best position already visited and also it is informed about the best particle position. After letting the particles to search adequate number of times in the solution space independently for the best possible positions, they are attracted to the basin containing the best particle by establishing proper communication among them about the search environment [5]. Genetic Algorithm also is considered as the famous evolutionary tuning method which has been implemented in twin rotor modeling and controller parameter tuning through recent literature [6-7]. Although GA can provide good solutions in tuning controllers that has a complex model, it requires huge memory and faster processing units with large word lengths to execute huge number of repeated computations [8-9]. Moreover, for highly multi-modal problems, the solutions may lose diversity and get trapped in local minima at some points unless special method is adopted to avoid premature convergence to suboptimal region of the search space [10].

I. Induction Motor Model

The overall dynamics of the induction motor under assumption of equal mutual inductances and linear magnetic circuit are given by the following fifth- order model [11-13]:

***Corresponding Author:** Y. Hoseynpoor, Department of Electrical Engineering, Ardabil Branch, Islamic Azad University, Ardabil, Iran.
E-mail: Yaser.hoseynpoor@gmail.com, Tel/Fax/Mob: +98 912 402 7898

$$\frac{d\omega}{dt} = \mu \psi_d i_q - \frac{T_L}{J}$$
$$\frac{d\psi_d}{dt} = -\alpha \psi_d + \alpha M i_d$$
$$\frac{di_d}{dt} = -\gamma i_d + \alpha\beta\psi_d + n_p \omega i_q + \alpha M \frac{i_q^2}{\psi_d} + \frac{1}{\sigma L_s} u_d \quad (1)$$
$$\frac{di_q}{dt} = -\gamma i_q - \beta n_p \omega \psi_d - n_p \omega i_d - \alpha M \frac{i_q i_d}{\psi_d} + \frac{1}{\sigma L_s} u_q$$
$$\frac{dp}{dt} = n_p \omega + \alpha M \frac{i_q}{\psi_d}$$
$$T = \mu \psi_d i_q$$

Where;

Ψ_d and Ψ_q are the d-q axis components of the motor flux and ω is the motor speed. u_d, u_q are the d-q axis components of the motor voltage, i_d, i_q are the stator current components, the n_p is the motor pair poles, M is the mutual inductance and R_s and R_r are the stator and rotor resistance respectively. L_s and L_r are stator and rotor self inductance respectively and T_l is the load torque.

$$\sigma = 1 - (M^2 / L_r L_s); \alpha = \left(\frac{R_r}{Lr}\right); \beta = \left(\frac{M}{\sigma L_s L_R}\right); \mu = \left(\frac{n_p M}{J L}\right) \quad (2)$$

$$p = \arctan \frac{\psi_b}{\psi_a} \quad \text{And} \quad \gamma = \left(\frac{M R_r}{\sigma L_S L_r^2}\right) + \left(\frac{R_S}{\sigma L_S}\right)$$

II. Field Oriented Vector Control Model

The induction motors have various methods of control and the particular method to be adopted depends on the nature of the application. The current in an ac motor can be separated into two distinct components; I_d or the flux producing current component and I_q or the torque producing current component. The total current is the vector sum of those two current components; the torque produced in the motor is based on the cross product of these vectors [14]. Different technologies in drive system implement different levels of control over one or more of these components and the vector angle between them [15]. Thus FOC has the flux and torque independent of each other and as a result of the FOC is the advantages of increasing starting torque, increasing low speed torque, increased shock load capability, tighter speed and torque regulation. The FOC de-coupling control is shown in Fig.1 [16].

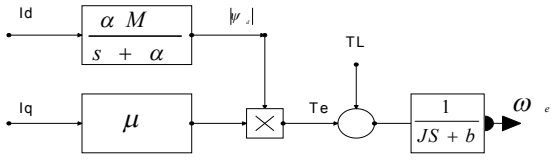

Fig. 1. FOC de-coupling control

The voltage is the command action, required to cancel the nonlinearity by using voltage state feedback. So ψ_d is regulated directly by V_d, after ψ_d becomes constant, the equation of speed becomes linear and voltage V_q regulates the speed ω directly. The voltage feedback equations [16] are:

$$u_d = \frac{\beta}{L_r}\left(-p\omega i_q - \frac{M}{T_r}\frac{i_q^2}{\psi_d} - \frac{M}{\beta T_r} + V_d\right)$$
$$u_q = \frac{\beta}{L_r}\left(p\omega i_d + \frac{M}{T_r}\frac{I_d i_q}{\psi_d} + \frac{M}{T_r} p\omega \psi_{rd} + V_q\right) \quad (3)$$

The closed loop system become

$$\frac{d\omega}{dt} = \mu \psi_d i_q - \frac{Tl}{J}$$
$$\frac{d\psi_d}{dt} = -\alpha \psi_d + \alpha M i_d \quad (4)$$
$$\frac{di_d}{dt} = -\alpha i_d + V_d$$
$$\frac{di_q}{dt} = -\alpha i_q + V_q$$

Where; the time constant is $T_r = \frac{L_r}{R_r}$

The stator current command in current vector control or its desired value in vector voltage control is chosen to satisfy the equation

$$I_S = \frac{T_r}{M}\frac{d\varphi}{dt} + \frac{1}{M}\varphi \quad (5)$$

Where φ: is the flux reference value.

The current commands to voltage commands equations are the following:

$$V_d = (I_{dr} - I_d)(k_6 + \frac{k_7}{s})$$

$$V_q = (I_{qr} - I_q)(k_8 + \frac{k_9}{s}) \quad (6)$$

Where I_d and I_q are the actual d-q stator current components respectively. The current command equations in term of flux and speed set points are the following:

$$I_{dr} = k_1\int_0^t (f_{ref} - \psi_d)dt + k_2(f_{ref} - \psi_d) + \frac{\psi_d}{M}$$

$$I_{qr} = (k_3(\theta_{ref} - \theta) + k_4\int_0^t (\theta_{ref} - \theta)dt + k_5(\omega_{ref} - \omega) + \alpha_{ref})/\mu \quad (7)$$

The main purpose of this paper is to optimize the performance of the FOC technique by improving the motor efficiency by identifying the optimum reference flux. Also selecting the optimal flux set point optimal selection of the controller gains (K_1, K_2, K_3, K_4 and K_5). The optimization is done using genetic algorithm and practical swarm optimization methods.

IV. Proposed Genetic Algorithm

Genetic algorithms (GAs) are global optimization techniques that avoid many shortcomings exhibited in conventional search techniques on a large and complicated search space. The application of GAs to control engineering can broadly be classified into two distinct areas: off-line design and on-line optimization. On-line applications tend to be quite rare because of the difficulties associated with using a GA in real-time and directly influencing the performance of the system. GAs has been applied to controller design problems as well as to system identification techniques. In each case, either the parameters or the structure can be optimized, or potentially both [17]. Other applications include fault diagnosis, stability analysis, sensor-actuator placement, and other combinatorial problems. In any design problem there is a multi dimensional space of possible solutions. Some of these solutions may be acceptable, but not the best (Local Optimum). Optimal mathematical solutions could be obtained from a control system with linear plant dynamics. The GA can be regarded as a research method from multiple directions to solve for problem solutions, since it contains three evolutionary operations: reproduction, crossover, and mutation. In the traditional binary-coded GA, all the variables of interest must be encode as binary digits (genes) and a collection of binary digits further forms a string (chromosome) [18]. Then three standard genetic operations, i.e., reproduction, crossover, and mutation are performed to produce a new generation. Such procedures are repeated until the pre-specified number of generations is achieved, or the required accuracy is satisfied. Some studies applying traditional GA with binary coding to solve optimization problems such as the PID controller design [19]. After a manipulation of binary-coded GA, the final binary digits are then decoded as original real numbers. This is an indirect optimization problem searching .The GA can use single or multiple crossovers algorithm. The most important problem in the design of GA is to choose the fitness function, chromosomes, reproduction, crossover, mutation and the stopping rules of algorithm.

In this paper the optimization process was done using MATLAB/ SIMULINK/GA toolbox parameters as follows:
In case of optimal controller gains, the number of variables are five, the population type is double vector, population size is 20, the initial range of variables are [200-600] for K_1, K_2, [0-20] for K_3 and K_4 ,[500-5000]. For the reproduction, the elite count is 2 and the crossover friction is 0.8, the mutation function is Gaussian, the crossover function is scattered, the stopping rules is the no of generation is 100, and the stall time limit is 200 sec, in case of efficiency optimization, the no of variables are one and the initial range is [0.1-2], all other parameters are the same.

1. Fitness function

The most critical step in applying GA Algorithms is to choose the objective functions that are used to evaluate fitness of each chromosome. In this paper there will be two genetic algorithms, one is to improve the drive performance as in by selecting the optimal gains of K_1, K_2, K_3, K_4, K_5 of the equations (7), and the other algorithm is to optimizing the motor efficiency by selecting the optimal flux set point.

The following two fitness functions are used for optimization process:

- For the case of improving the motor efficiency, the fitness function will be motor input power.

$$f_2 = \max(p_{in}) = \max(I_{sa} * u_a + I_{sb} * u_b) \quad (8)$$

For the case of optimum control gains, the fitness function will be

$$f_1 = \int [(f_{ref} - f_r)^2 + (w_{ref} - w_r)^2] dt \quad (9)$$

Such that f_{ref} is the reference flux and f_r is the motor rotor flux, w_{ref} is the reference speed and ω_r is the motor speed.

2. Chromosomes

Let parent chromosomes are selected to be crossed and parameters (gains) be a random number chosen from [0, 10]. And for the parameter of flux reference will be chosen in the range from [0.1, 2].

3. Mutation

In this paper, dynamic mutation is applied. The mutation process randomly picks up Pm × N Chromosomes to be mutated. The algorithm stops if the pre-specified number of generations is achieved. Figure 2 shows the outline of design steps and data flow for the real-coded GA.

3. Stopping Rules

The processes of generating new chromosomes and selecting those with better function values are continued until the given stopping conditions are satisfied. The process can be stopped after a fixed number of generations, or when any significant improvement in the solution ceases to occur. In this paper, GA is run for a fixed number of generations 100 Generations and the size of population or the number of individuals in each generation is:

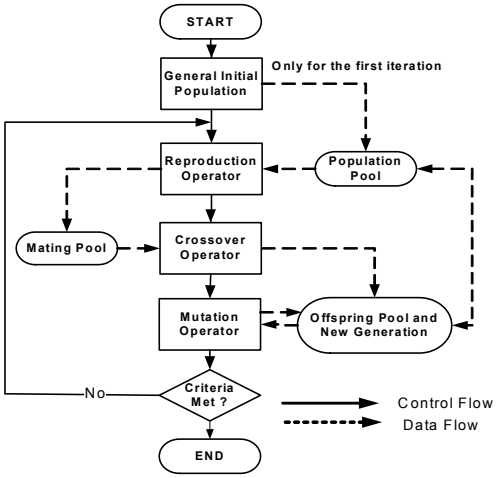

Fig. 2. The data flow of RCGA

V. Proposed Particle Swarm Algorithm

Optimization techniques using analogy of swarming principle have been adopted to solve a variety of engineering problems in the past decade. Swarm Intelligence (SI) is an innovative distributed intelligent paradigm for solving optimization problems that originally took its inspiration from the biological examples by swarming, flocking and herding phenomena in vertebrates. A population of particles exists in the n-dimensional search space in which the optimization problem lives in. each particle has a certain amount of knowledge, and will move about the search space based on this knowledge. The particle has some inertia attributed to it and so it will continue to have a component of motion in the direction it is moving [19-20].

The particle swarm optimization PSO algorithm we employ here is based on that Particles fly through the solution space and are influenced by both the best particle in the particle population and the best solution that a current particle has discovered so far. The best particle in the population is typically denoted by (global best), while the best position

that has been visited by the current particle is donated by (local best) [21]. The (global best) individual conceptually connects all members of the population to one another. That is, each particle is influenced by the very best performance of any member in the entire population. The (local best) individual is conceptually seen as the ability for particles to remember past personal success. The particle swarm optimization makes use of a velocity vector to update the current position of each particle in the swarm. The position of each particle is updated based on the social behavior that a population of individuals adapts to its environment by returning to promising regions that were previously discovered [22].

Let the i^{th} particle of the swarm is represented by the D-dimensional vector $x_i = (x_{i1}, x_{i2}, ..., x_{iD})$ and the best particle in the swarm, i.e. the particle with the smallest function value, is denoted by the index g. The best previous position (the position giving the best function value) of the i^{th} particle is recorded and represented as $p_i = (p_{i1}, p_{i2}, ..., p_{iD})$, and the position change (velocity) of the i^{th} particle is $v_i = (v_{i1}, v_{i2}, ..., v_{iD})$. The particles are manipulated according to the equations:

$$v_{id} = w.v_{id} + c_1.r_1.(p_{id} - x_{id}) + c_2.r_2.(p_{gd} - x_{id}) \quad (10)$$

$$x_{id} = x_{id} + v_{id} \quad (11)$$

Where; d = 1, 2...D; i = 1, 2…N and N is the size of swarm (no of birds); w is the inertia weight; c_1 and c_2 are two positive constants; r_1 and r_2 are the random values. Equation (10) is used to calculate i^{th} particle's new velocity by taking into consideration three terms: the particle's previous velocity, the distance between the particle's best previous and current position, and, finally, the distance between swarm's best experience (the position of the best particle in the swarm) and i^{th} particle's current position. Then, following the second equation, the i^{th} particle flies toward a new position. In general, the performance of each particle is measured according to a predefined fitness function, which is problem dependent. The role of the inertia weight w is considered very important in PSO convergence behavior. The inertia weight is employed to control the impact of the previous history of velocities on the current velocity. In this way, the parameter w regulates the trade-off between the global (wide-ranging) and local (nearby) exploration abilities of the swarm. A large inertia weight facilitates global exploration (searching new areas); while a small one tends to facilitate local exploration, i.e. fine-tuning the current search area. A suitable value for the inertia weight w usually provides balance between global and local exploration abilities and consequently a reduction on the number of iterations required to locate the optimum solution. A general rule of thumb suggests that it is better to initially set the inertia to a large value, in order to make better global exploration of the search space, and gradually decrease it to get more refined solutions, thus a time decreasing inertia weight value is used.

In this paper, the PSO will be used to evaluate the optimal gains of the field oriented vector controller and for selecting the optimal flux set point for efficiency improvement of the motor, the fitness function to be minimized in the two cases are the same as in equations (8) and (9).

For the optimal gains to be chosen, there will be five variables (r_1 to r_5) which are the controller gains, there will be five constants (c_1 to c_5) to be chosen as c_1= 0.8, c_1= 0.6, c_3= 0.5, c_4= 0.15, c_5= 0.12, w=0.9, n=50, max no of bird step is 30 and the variable boundary are [0 to 50].

For the optimal flux set point, there will be one variable (r_1) which is the flux reference, there will be one constants (c_1) to be chosen as c_1= 0.12, w=0.9, n=50, max no of bird step is 30 and the variable boundary are [0.1 to 2].

VI. SIMULATION RESULTS

The simulation targets are:

- From the behavior of the motor without any control during the speed and flux manual change. For FOC the speed and the flux are independent variables. The optimum flux reference value can be identified which has the minimum input power. The motor behavior at this case will be shown in part A.
- For the optimization process is to identify the optimum flux reference automatically by GA and PSO techniques. These results are illustrated in part B.
- The optimum gains selection for the controller of field oriented vector control shown in equations (6) and (7) using GA and PSO. These results are introduced in part C.

A. Performance during Flux Reference Change without Control

Starting the simulation with speed reference 100 r.p.m from zero to 100 sec, then it increased to 200 r.p.m from time of 100 sec to 200 sec as in Fig. 3. While the reference flux was increased from 0.15 weber to 0.45 weber in steps as shown in Fig. 4. As shown in Fig. 5, the motor input power is starting with high value then decreasing with flux reference increasing to reach its minimum value then increasing. The I_d and I_q currents response are illustrated in Fig. 6 and Fig. 7 respectively. These results indicate that the optimum reference flux is 0.24 weber which has the minimum motor input power.

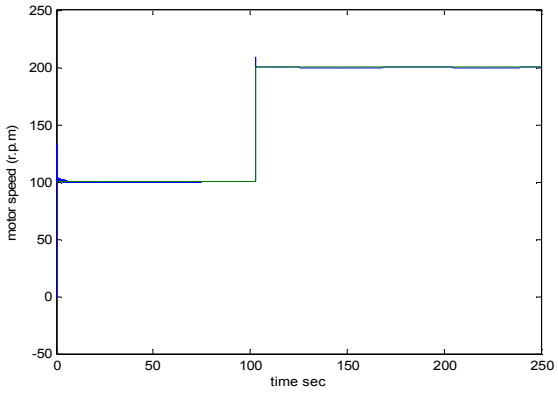

Fig. 3. Motor speed

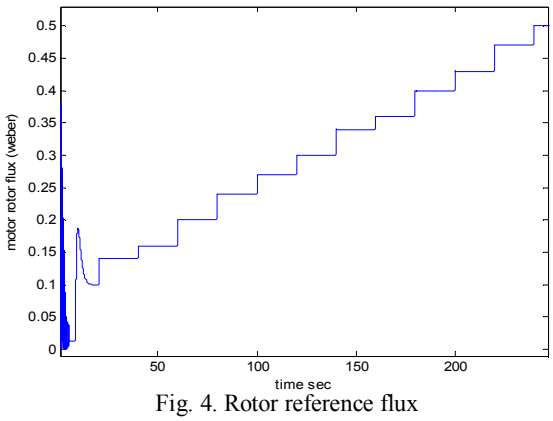

Fig. 4. Rotor reference flux

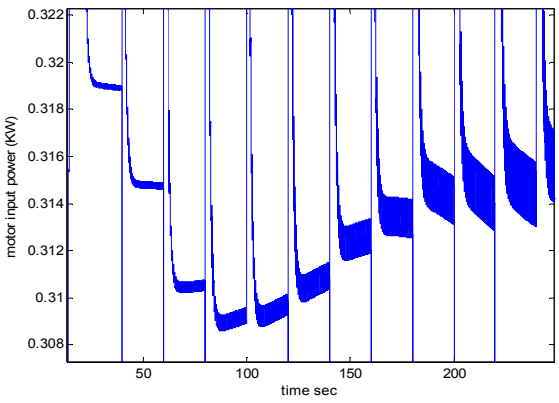

Fig. 5. Motor input power

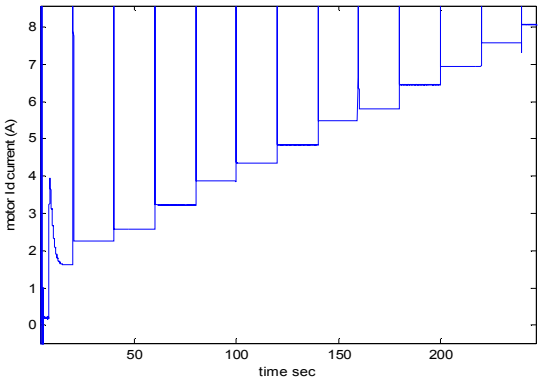

Fig. 6. Motor I_d current

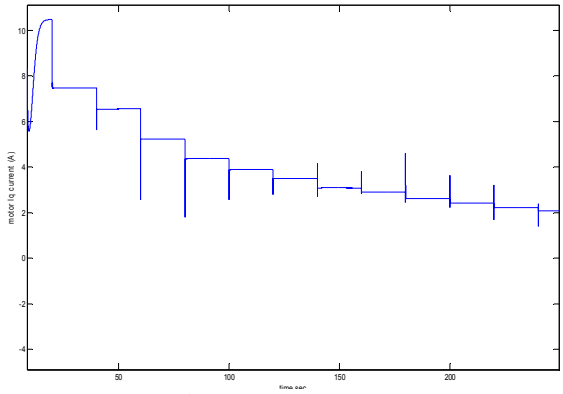

Fig. 7. Motor I_q current

B. Performance during Optimum reference Flux Selection by GA and PSO

Using GA and PSO, there will be only one variable in the proposed flux fitness function shown in equation (8), it is the value of the optimal flux reference. The results are illustrated in Fig's from (8-11), it show that by using GA, the flux reference, f_{ref}=0.17944, and for PSO, the reference will be 0.244 weber. So The PSO reference value is so closely to the target optimum reference flux that was determined before without control. The input power of PSO gives better motor efficiency than GA with the same previous parameters and speed reference as shown in Fig. 8.

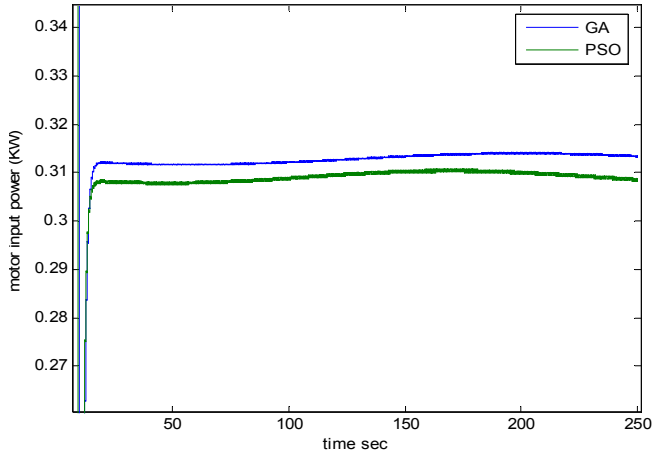

Fig. 8. Motor input power with GA & PSO

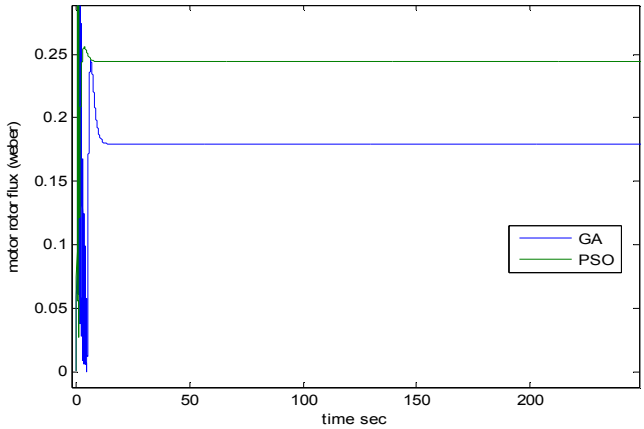

Fig. 9. Motor rotor flux using GA and PSO

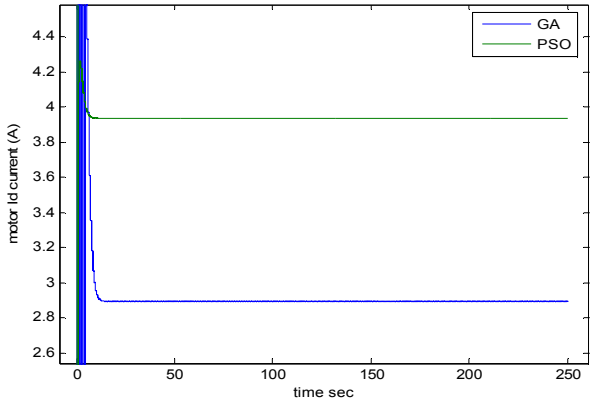

Fig. 10. Motor I_d using GA and PSO

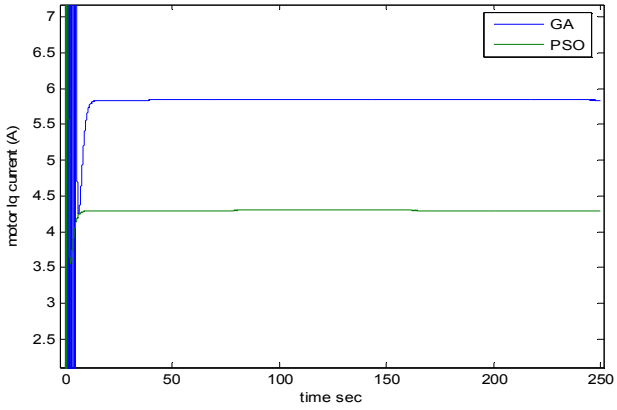

Fig. 11. Motor I_q current using GA and PSO

C. Performance during Optimum Gains Selection by GA and PSO

In this part, selecting of the optimal value of the controller gains (k_1, k_2, k_3, k_4 and k_5), is done by using GA with the same motor parameters and with the same speed reference using the gains fitness function as in equation (9). The optimum gains using GA are (k_1= 450.5, k_2= 510, k_3= 9.8, k_4= 11, k_5= 960), while their value by using PSO are (k_1= 231.9, k_2= 355.5, k_3= 20.6, k_4= 534.5, k_5= 737.5).

The response of the motor speed in the two control methods shown in Fig. 12. The motor speed by using PSO is better than the GA performance as it is closely to the reference value. The rotor flux at both optimization methods are shown in Fig. 13, the PSO flux has minimum overshoot than in the case of GA.

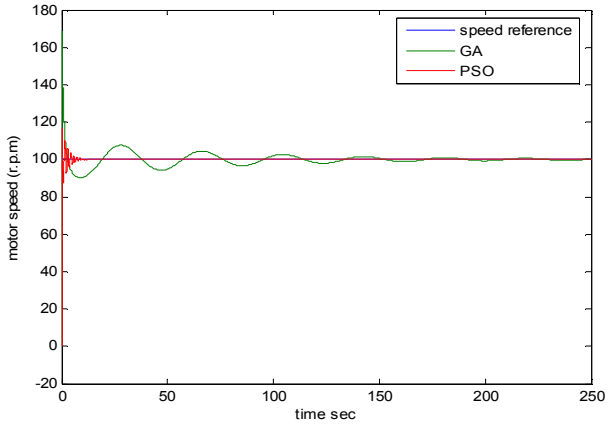

Fig. 12. Motor speed GA and PSO with optimum gains.

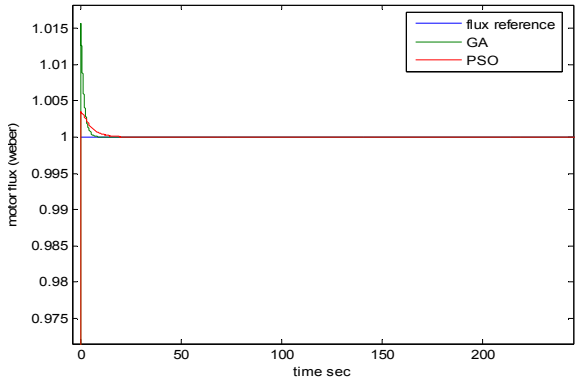

Fig. 13. Rotor flux GA and PSO with optimum gain

VII. Conclusion

In this paper, two different techniques of artificial intelligent which are genetic algorithm and particle swarm optimization have been used to improve the performance of field oriented vector control drive. Selecting of the optimum reference flux was done by the two methods compared to the uncontrolled methods. Also the optimum gains selections were done to enhance the motor response. The results show that the proposed are reliable and capable to enhance the motor behavior using the FOC drive by selecting the optimum gains. The PSO method has better response than the GA method. The proposed techniques minimize the input power which means increasing the motor efficiency by selecting the optimum reference flux.

REFERENCES

[1] M.O. Tokhi and M.S. Alam, "Particle Swarm Optimization Algorithms and Their Application to Controller Design for Flexible Structure Systems", The 1st IEEE International Conference on Computer, Control & Communication (IEEE-IC4), Karachi (Pakistan), pp. 7–18, 2007.

[2] N. Nedjah, "Swarm Intelligent Systems", Studies in Computational Intelligence, Springer, Vol. 26, pp. 3-123, 2006.

[3] K.E. Parsopoulos and M.N. Vrahatis, "Recent approaches to global optimization problems through particle swarm optimization", Natural Computing journal, Vol. 1, Issue 2-3, pp. 235-306, 2002.

[4] C.T. Raj, S.P. Srivastava, and P. Agarwal, "Differential Evolution based Optimal Control of Induction Motor Serving to Textile Industry", International journal of computer science, 2008.

[5] E. El-kholy, "High Performance Induction Motor Drive Based on Adaptive Variable Structure Control", Journal of Electrical Engineering, Vol. 56, No. 3-4, pp. 64-70, 2005.

[6] P. J. Fleming, R. C. Purshouse, "Genetic algorithms in control systems engineering", IEE Colloquium on Genetic Algorithm in Control System Engineering, No. 789, pp 3-32, 2001.

[7] L. Randy, S. Haupt and E. Haupt, "A practical Genetic algorithm", A John Wiley & Sons, INC. Publication, Second Edition, 2004.

[8] J. Andersson, "Applications of a Multi-objective Genetic Algorithm to Engineering Design Problems", Springer Berlin I, SBN 0302-9743.

[9] W. Chang, "Nonlinear System Identification and Control using a Real-Coded Genetic Algorithm", Applied Mathematical Modeling 31, pp. 541-550, 2007.

[10] R. R. Colin and E. R. Jonathan, "Genetic algorithm Principles and Perspective, A Guide to GA theory", Kluwer Academic Publishers ISBN 1-4020-7240-6, 2002.

[11] R. S. Gordon, "Electric Machines and Drives", Addison Wesley, 1992.

[12] D. Mehdi and S. Lassaad, "A New Method for Losses Minimization in IFOC Induction Motor Drives", International Journal of Systems and Control, Vol. 1, pp. 93-99, 2010.

[13] C.R. Sullivan, S. R. Sanders and H. Hofmann, "Stator-Flux-Based Vector Control of Induction Machines in Magnetic Saturation", IEEE Trans on Ind applications, Vol. 33, No. 4, pp. 935-941, 1997.

[14] D.S. Kirshnen, D.W. Novotny and T.A. Lipo, "On-line Efficiency Optimization of a Variable Frequency Induction Motor Drive", IEEE Trans on Ind applications, Vol. IA-20, No. 4, pp. 610-616, 1985.

[15] A. Gastli and M. Nobuyuki, "Stator Flux Controlled V/F PWM Inverter with Identification of IM Parameters", IEEE Trans on Industrial Electronics, Vol. 39, No. 4, pp. 334-340, 1992.

[16] Y.Y. Edward and P.C. Sen, "Decoupling Control of Induction Motor Drives", IEEE Trans on industrial electronics,Vol. 35, No. 2, pp. 253-262, 1988.

[17] R. Bandyobadhyay and U.K. Chakraborty, "Autotuning a PID Controller: A Fuzzy-Genetic approach", Journal of System Architecture 47, pp. 663-673, 2001.

[18] L. RenHou, Y. Zhang, "Fuzzy Logic Controller Based on Genetic Algorithms", Fuzzy Sets and Systems 83, pp. 1-10, 1996.

[19] M. E. El-Telbany, "Employing Particle Swarm Optimizer and Genetic Algorithms for Optimal Tuning of PID Controllers: A Comparative Study", ICGST-ACSE Journal, Vol. 7, Issue 2, pp. 49-54, 2007.

[20] F. Gao and H. Q. Tong, "A Novel Optimal PID Tuning and On-line Tuning Based on Particle Swarm Optimization", Proceedings of the International Conference on Sensing, Computing and Automation, pp. 182 - 186, 2006.

[20] L. Xu-zhou, F. Yu, Y.B. Wang, "PSO Algorithm Based Online Self-Tuning of PID Controller", Computational Intelligence and Security, International Conference, Harbin, pp. 128-132, 2007.

[21] L. Feng-Chieh and Y. Sheng-Ming, "On-line tuning of an Efficiency-Optimized Vector Controlled Induction Motor Drive", Journal of Science and Engineering, Vol. 6, No. 2, pp. 103-110, 2003.