

Optimum Induction Motor Speed Control Technique Using Particle Swarm Optimization

M. M. Eissa¹, G. S. Virk², A. M. AbdelGhany¹, E. S. Ghith^{3,*}

¹Electrical Power and Machines Engineering at Helwan University, Egypt

²Electrical Power and Machines Engineering at South Westphalia University, Germany

³A master candidate in Systems Automation and Engineering Management; Master Degree Program Faculty of Engineering Helwan University, Egypt

Abstract Industrial processes are subjected to variation in parameters and parameter perturbations, which when significant makes the system unstable. In order to overcome this problem of parameter variation the PI controllers are widely used in industrial plants because it is simple and robust. However there is a problem in tuning PI parameters. So the control engineers are on look for automatic tuning procedures. In recent years, many intelligence algorithms are proposed to tuning the PI parameters. Tuning PI parameters using different optimal algorithms such as the simulated annealing, genetic algorithm, and particle swarm optimization algorithm. In this paper a scheduling PI tuning parameters using particle swarm optimization strategy for an induction motor speed control is proposed. The results of our work have showed a very low transient response and a non-oscillating steady state response with excellent stabilization. The simulation results presented in this paper show the effectiveness of the proposed method, with satisfied response for PSO-PI controller.

Keywords Particle Swarm Optimization, PI Controller, Induction motor, Matlab, Simulink

1. Introduction

Induction motors play a vital role in the industrial sector especially in the field of electric drives and control. Without proper controlling of the speed, it is impossible to achieve the desired task for a specific application. AC motors, particularly the squirrel-cage induction motors (SCIM), enjoy several inherent advantages like simplicity, reliability, low cost and virtually maintenance-free electrical drives. However, for high dynamic performance industrial applications, their control remains a challenging problem because they exhibit significant nonlinearities and many of the parameters, mainly the rotor resistance, vary with the operating conditions[1]. Field orientation control (FOC) or vector control[2] of an induction machine achieves decoupled torque and flux dynamics leading to independent control of the torque and flux as for a separately excited DC motor. FOC methods are attractive, but suffer from one major disadvantage (they are sensitive to motor parametric variations such as the rotor time constant and an incorrect flux measurement or estimation at low speeds[3]).

Induction motors are widely used in various industries as prime work-horses to produce rotational motions and forces. Generally, variable-speed drives for induction motors

require both wide operating range of speed and fast torque response, regardless of load variations. The classical control is used in majority of the electrical motor drives. Conventional control makes use of the mathematical model for the controlling of the system. When there are system parametric variations or environmental disturbance, behavior of system is not satisfactory and deviates from the desired performance[4].

In addition, usual computation of system mathematical model is difficult or impossible. To obtain the exact mathematic model of the system, one has to do some identification techniques such as the system identification and obtain the plant model. Moreover, the design and tuning of conventional controller increases the implementation cost and adds additional complexity in the control system and thus, may reduce the reliability of the control system. Hence, the fuzzy based techniques are used to overcome this kind of problems. Efficient torque control of induction motor drives in combination with resonant DC-link input filters can lead to a type of stability problem that is known as negative impedance instability. To overcome this[5], proposed a solution to the above problem by using the concept of non-linear system stabilizing controller with the plant.

There are a number of significant control methods available for induction motors including scalar control, vector or field-oriented control, direct torque and flux control, sliding mode control, and the adaptive control[4]. Scalar control is aimed at controlling the induction machine

* Corresponding author:

ehabsai.felnaser@yahoo.com (E. S. Ghith)

Published online at <http://journal.sapub.org/ijee>

Copyright © 2013 Scientific & Academic Publishing. All Rights Reserved

to operate at the steady state, by varying the amplitude and frequency of the fundamental supply voltage[6]. A method to use of an improved V/f control for high voltage induction motors and its stability was proposed in[7]. The scalar controlled drive, in contrast to vector or field-oriented controlled one, is easy to implement, but provides somewhat inferior performance. This control method provides limited speed accuracy especially in the low speed range and poor dynamic torque response.

Many researchers had worked on the fuzzy logic based on-line efficiency control for an indirect vector controlled IM drive. Bimal Bose *et al.*[8] extended the same control technique to a stator flux oriented electric vehicle IM drive and then implemented the fuzzy controller by a dynamic back propagation algorithm using an ANFIS controller. They further verified the simulated results using a DSP based hardware. Haider *et al.*[9] Presented the design and implementation of Fuzzy-SMC-PI methodology to control the flux and speed of an induction motor. The Fuzzy-SMC-PI was basically a combination of Sliding Mode Control (SMC) and PI control methodologies through fuzzy logic, but the drawback being the chattering during the time of switching.

In[10] and[11], the researchers implemented a fuzzy logic controller to adjust the boundary layer width according to the speed error. The drawback of their controller is that it depends on the equivalent control and on the system parameters. Two researchers, Takagi and Sugeno developed a excellent control scheme for control of various applications in the industrial sector. This controller had many advantages over the other methods discussed so far. Many researchers started using their models for their applications. Zie, Ling and Jhang[12] presented a TS model identification method by which a great number of systems whose parameters vary dramatically with working states can be identified via Fuzzy Neural Networks (FNN). The suggested method could overcome the drawbacks of traditional linear system identification methods which are only effective under certain narrow working states and provide global dynamic description based on which further control of such systems may be carried out.

In the above mentioned paper, there were contribution for their research and they solved many problems but there are some limitation in the settling time of the responses and the proper selection of the rule base. The responses had taken a lot of time to reach the final steady state value.

The Particle Swarm Optimization (PSO) is a population-based optimization method first proposed by Kennedy and Eberhart (Kennedy and Eberhart, 1995). Some of the attractive features of the PSO, as reviewed by Bergh (Bergh, 2001). Include the ease of implementation and the fact that no gradient information is required. It can be used to solve a wide array of different optimization problems, including most of the problems that can be solved using Genetic Algorithms; some example applications include neural network training and nonlinear optimization problems with continuous variables and it is easily

expanded to treat problems with discrete variables[13, 14,15].

In this paper, a sincere attempt is made to reduce the settling time of the responses and make the speed of response very fast by designing an efficient controller using PSO-PI control strategy. Here, we have control strategy for the speed control of IM, which has yielded excellent results compared to the others mentioned in the literature survey above. The results of our work have showed a very low transient response and a non-oscillating steady state response with excellent stabilization.

2. Modeling of Induction Motor

In this study, the mathematical model of the system consists of space vector PWM voltage source inverter, induction motor, direct flux and the torque control. An induction motor model to predict the voltage required achieving a desired output torque is given in[16]. Fig.1.shows the power circuit of the 3-phase induction motor and Fig.2. shows the equivalent circuit used for obtaining the mathematical model of the IM.

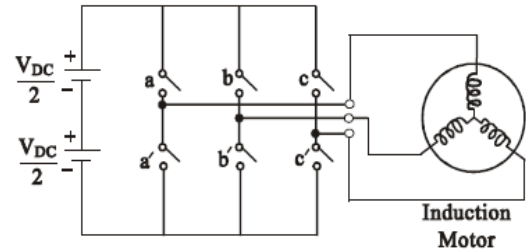


Figure 1. Power circuit connection diagram for the IM

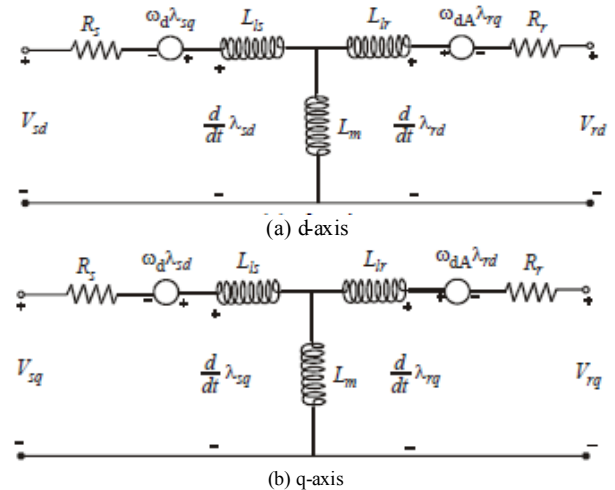


Figure 2. Equivalent circuit of induction motor in d - q frame

An induction motor model is then used to predict the voltage required to drive the flux, torque and the speed to the demanded values. This calculated voltage is then synthesized using the space vector modulation. The stator and rotor voltage equations are given by[16].

$$V_{sd} = R_s i_{sd} + \frac{d}{dt} \lambda_{sd} - \omega_d \lambda_{sq} \quad (1)$$

$$V_{sq} = R_s i_{sq} + \frac{d}{dt} \lambda_{sq} + \omega_d \lambda_{sd} \quad (2)$$

$$V_{rd} = R_r i_{rd} + \frac{d}{dt} \lambda_{rq} - \omega_d \lambda_{rq} \quad (3)$$

$$V_{rq} = R_r i_{rq} + \frac{d}{dt} \lambda_{rq} + \omega_d \lambda_{rd} \quad (4)$$

where V_{sd} and V_{sq} , V_{rd} and V_{rq} are the direct axes and quadrature axes stator and rotor voltages [16].

The squirrel-cage induction motor considered for the simulation study in this thesis, has the d and q-axis components of the rotor voltage zero. The flux linkages to the currents are related by the Equation (5) as

$$\begin{bmatrix} \lambda_{sd} \\ \lambda_{sq} \\ \lambda_{rd} \\ \lambda_{rq} \end{bmatrix} = M \begin{bmatrix} i_{sd} \\ i_{sq} \\ i_{rd} \\ i_{rq} \end{bmatrix}; M = \begin{bmatrix} L_s & 0 & L_m & 0 \\ 0 & L_s & 0 & L_m \\ L_m & 0 & L_r & 0 \\ 0 & L_m & 0 & L_r \end{bmatrix} \quad (5)$$

The electrical part of an induction motor can thus be described by a fourth-order state space model (4*4) which is given in Equation (6), by combining equations (1-5) as

$$\begin{bmatrix} v_{qs} \\ v_{ds} \\ v_{qr} \\ v_{qr} \end{bmatrix} = \begin{bmatrix} R_s + sL_s & -\omega_e L_s & sL_m & -\omega_e L_m \\ -\omega_e L_s & R_s + sL_s & -\omega_e L_m & pL_m \\ sL_m & (\omega_e - \omega_r)L_m & R_r + sL_r & (\omega_e - \omega_r)L_r \\ -(\omega_e - \omega_r)L_m & sL_m & -(\omega_e - \omega_r)L_r & R_r + sL_r \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{qr} \\ i_{qr} \end{bmatrix}, \quad (6)$$

where s is the laplacian operator. By superposition, i.e., adding the torques acting on the d-axis and the q-axis of the rotor windings, the instantaneous torque produced in the electromechanical interaction is given by

$$T_{em} = \frac{3}{2} \left(\frac{P}{2} \right) \left(\lambda_{rq} i_{rd} - \lambda_{rd} i_{rq} \right) \quad (7)$$

The electromagnetic torque expressed in terms of inductances is given by

$$T_{em} = \frac{3}{2} \left(\frac{P}{2} \right) L_m \left(i_{sq} i_{rd} - i_{sd} i_{rq} \right), \quad (8)$$

The mechanical part of the motor is modeled by the equation (9) as [16].

$$\frac{d}{dt} \omega_m = \frac{T_{em} - T_L}{J_{eq}} = \frac{\frac{3}{2} \left(\frac{P}{2} \right) L_m (i_{sq} i_{rd} - i_{sd} i_{rq}) - T_L}{J_{eq}} \quad (9)$$

where, ω_m is angular speed, T_{em} is electromagnetic torque, T_L is load torque, J_{eq} is equivalent moment of inertia, L_m is mutual inductance, L_r is rotor inductance, L_s is stator inductance, and VDC is dc voltage.

3. Particle Swarm Optimization

3.1. Background of Particle Swarm Optimization

Natural creatures sometimes behave as a swarm. One of the main streams of artificial life researches is to examine how natural creatures behave as a swarm and reconfigure the swarm models inside a computer. Swarm behavior can be modeled with a few simple rules. School of fishes and swarm of birds can be modeled with such simple models. According to the research results for a flock of birds, birds find food by flocking (not by each individual). The observation leads the assumption that all information is shared inside flocking. PSO is basically developed through simulation of bird flocking in two-dimension space). The position of each agent is represented by XY axis position and also the velocity is expressed by v_x (the velocity of X axis) and v_y (the velocity of Y axis). Modification of the agent position is realized by the position and velocity information. Bird flocking optimizes a certain objective function. Each agent knows its best value so far (pbest) and its XY position. This information represents the personal experiences of each agent. Moreover, each agent knows the best value so far in the group (gbest) among (pbests). Namely, each agent tries to modify its position using the following information: the current positions (x, y), the current velocities (v_x , v_y), the distance between the current position and pbest, the distance between the current position and gbest. This modification can be represented by the concept of velocity. Velocity of each agent can be modified by the following equation (10,11)

$$v_i^{k+1} = w v_i^k + c_1 r_1 \times (pbest_i - s_i^k) + c_2 r_2 \times (gbest - s_i^k) \quad (10)$$

Where, v_i^k is current velocity of agent i at iteration k , v_i^{k+1} modified velocity of agent i , r_1 and r_2 are random numbers, uniformly distributed in $[0, 1]$, s_i^k is current position of agent i at iteration k , pbest is pbest of agent i , gbest is gbest of agent i , w is weight function for velocity of agent i , c_1 and c_2 are positive constants, called cognitive and social parameter respectively. The following weighting function is usually utilized in (10)

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} \times iter \quad (11)$$

Where, w_{\max} is initial weight, w_{\min} is final weight, $iter_{\max}$ is maximum iteration number, $iter$ is current iteration number.

Using the above equation, a certain velocity that gradually gets close to (pbests) and (gbest) can be calculated. The current position (searching point in the solution space) can be modified by the following equation:

$$s_i^{k+1} = s_i^k + v_i^{k+1} \quad (12)$$

Figure 3 shows a concept of modification of a searching point by PSO and Figure 4 shows a searching concept with agents in a solution space. Each agent changes its current position using the integration of vectors as shown in Figure 3.

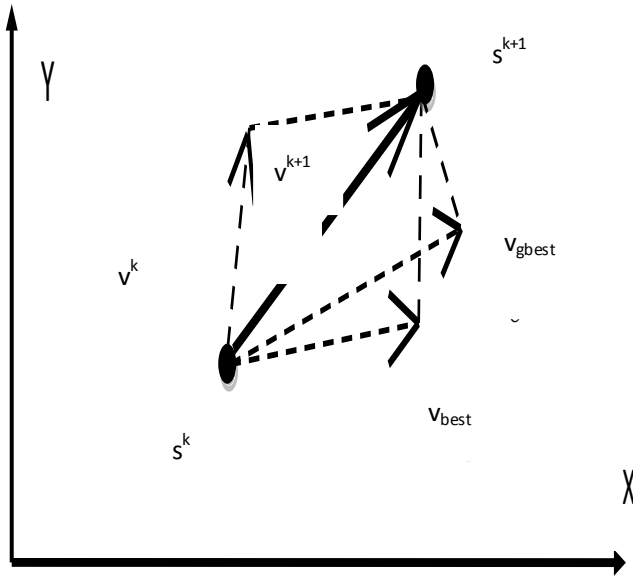


Figure 3. Concept of modification of a searching point by PSO

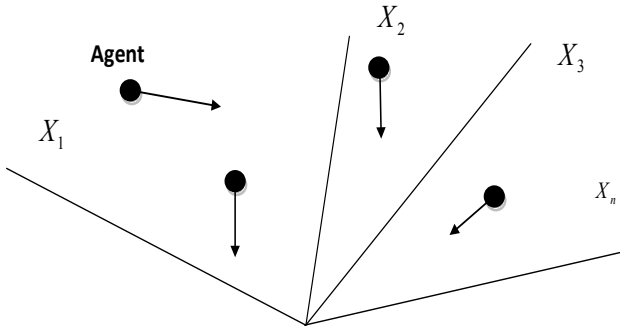


Figure 4. Searching concept with agents in a solution space by PSO

The general flow chart of PSO can be described as follows:

Step 1: Generation of initial condition of each agent

Initial searching points (s_i^0) and velocities (v_i^0) of each agent are usually generated randomly within the allowable range. The current searching point is set to pbest for each agent. The best-evaluated value of pbest is set to gbest and the agent number with the best value is stored.

Step 2: Evaluation of searching point of each agent

The objective function value is calculated for each agent. If the value is better than the current pbest of the agent, the pbest value is replaced by the current value. If the best value of pbest is better than the current gbest, gbest is replaced by the best value and the agent number with the best value is stored.

Step 3: Modification of each searching point

The current searching point of each agent is changed using (10)-(12)

Step 4: Checking the exit condition such as maximum number of iteration

The current iteration number reaches the predetermined maximum iteration number or any other stopping condition (desired accuracy) is reached, then exit. Otherwise, go to step 2.

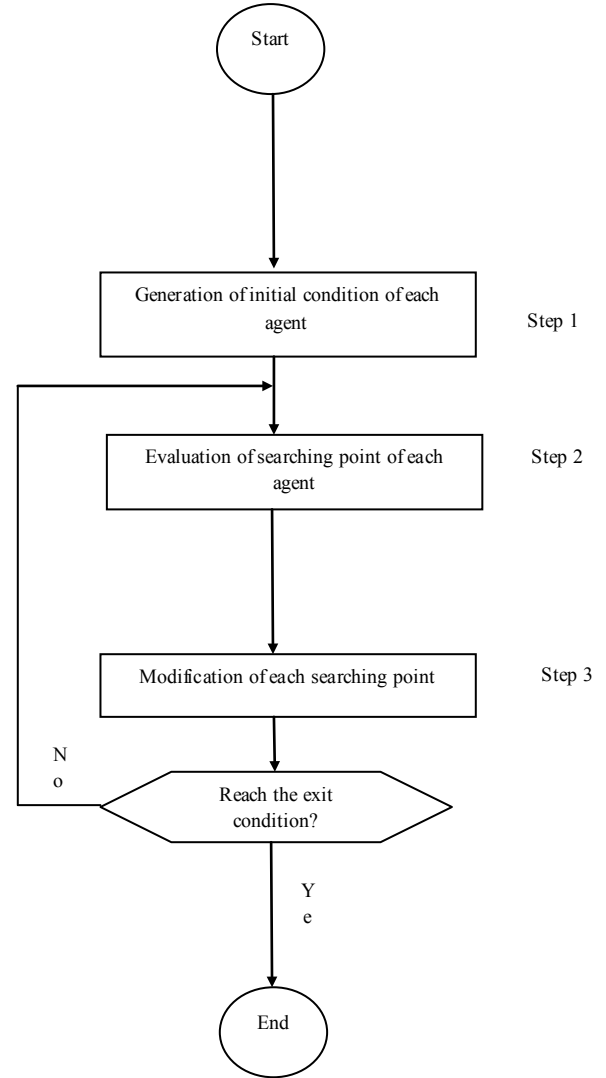


Figure 5. General flow chart of PSO

Figure 5 shows the general flow chart of PSO. The features of the searching procedure of PSO can be summarized as follows:

(a) As shown in (10)-(12), PSO can essentially handle continuous optimization problem.

(b) PSO utilizes several searching points like genetic algorithm (GA) and the searching points gradually get close to the optimal point using their (pbests) and the (gbest).

(c) The speed of each bird, Equation (10), has three components, namely, a momentum component trying to keep the same search direction, a component towards the birds self-best (cognitive term) and a component toward the global swarm best (social term).

(d) The above concept is explained using only XY-axis (two-dimension space). However, the method can be easily applied to n-dimension problem. Namely, PSO can handle continuous optimization problems with continuous state variables in an n-dimension solution space.

The PSO method appears to adhere to the five basic principles of swarm intelligence

1. Proximity, i.e., the swarm must be able to perform

simple space and time computations;

2. Quality, i.e., the swarm should be able to respond to quality factors in the environment;

3. Diverse response, i.e., the swarm should not commit its activities along excessively narrow channels;

4. Stability, i.e., the swarm should not change its behavior every time the environment alters; and finally

5. Adaptability, i.e., the swarm must not be able to change its behavior, when the computational cost is not prohibitive.

Indeed, the swarm in PSO performs space calculations for several time steps. It responds to the quality factors implied by each particle's best position and the best position in the swarm, allocating the responses in away that ensures diversity. Moreover, the swarm alters its behavior only when the best particle in the swarm changes, thus, it is both adaptive and stable.

3.2. Parameter Selection of PSO

The role of the inertia weight w in Equation (10) is considered critical for the PSO's convergence behavior. The inertia weight is employed to control the impact of the previous history of the velocities on the current one. Accordingly, 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 facilitates local exploration, i.e., fine-tuning the current searching area. A suitable value of inertia weight w usually provides balance between global and local exploration abilities and consequently results in reduction of the number of iterations required to locate the optimum solution. Initially, the inertia weight was constant. However, experimental results indicated that it is better to initially set the inertia to large value, in order to promote global exploration of the searching space, and gradually decrease it to get more refined solutions. Thus, an initial value around 1.2 and a gradual decline towards 0 can be considered as good choice for w .

The parameters c_1 and c_2 , in Equation (10), are not critical for PSO's convergence. However, proper fine-tuning may result in faster convergence and alleviation of local minima. An extended study of the acceleration parameter, indicates that $c_1 = c_2 = 0.5$ might provide even better results. Recent work reports that it might be even better to choose a larger

cognitive parameter, c_1 than a social parameter, c_2 but with $c_1 + c_2 \leq 4$.

The parameters r_1 and r_2 are used to maintain the diversity of the population, and they are uniformly distributed in the range $[0, 1]$.

The constriction factor χ is a way of choosing the values of w , c_1 , and c_2 so that convergence is ensured. A modified velocity update equation, corresponding to one of several constriction factor is presented in equation (13).

$$v_i^{k+1} = \chi (v_i^k + c_1 r_1 \times (pbest_i - s_i^k) + c_2 r_2 \times (gbest - s_i^k)) \quad (13)$$

Where

$$\chi = \frac{2}{2 - \varphi - \sqrt{\varphi^2 - 4\varphi}} \quad (14)$$

and

$$\varphi = c_1 + c_2, \varphi > 4.$$

As [14] study the performance of PSO using the constriction factor. Their results indicated that using the constriction factor usually resulted in a better rate of convergence.

4. Realization of PSO-PI Controller Tuning Optimal Parameters

4.1. Proposed PSO-PI Controller

In this paper the particle swarm optimization algorithms (PSO) (each particle contains two members P and I) mean that the search space has two dimension and particles that must 'fly' in a two dimensional space, (PSO are applied to search globally optimal parameters of PI) [17]. PSO Algorithms are used to find the optimal parameters of IM speed control system. The structure of the PI controller with PSO algorithm is shown in Figure 6.

The conventional control system performance behaves poorly in characteristics and even it becomes unstable, when improper values of the controller tuning constants are used. The proposed PSO technique has the feature of tuning at every time, the particles are assumed new positions, they are ensured to update the best particle by comparing the costs corresponding to these positions with the previously selected best particle cost [17]. The flowchart shows the parameters selection using PSO-PI, see Figure 7.

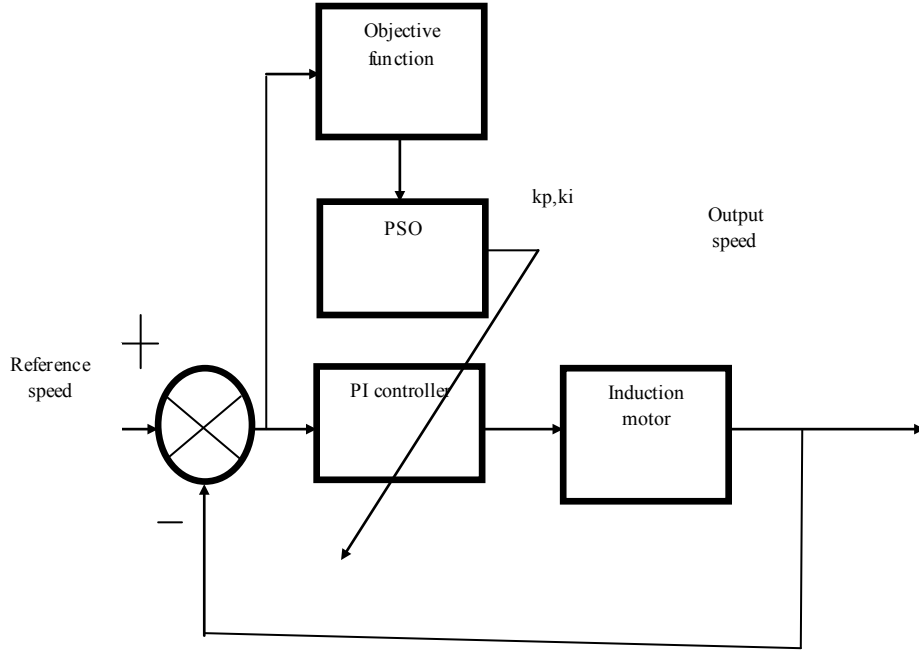


Figure 6. Optimal PI controller with PSO algorithm

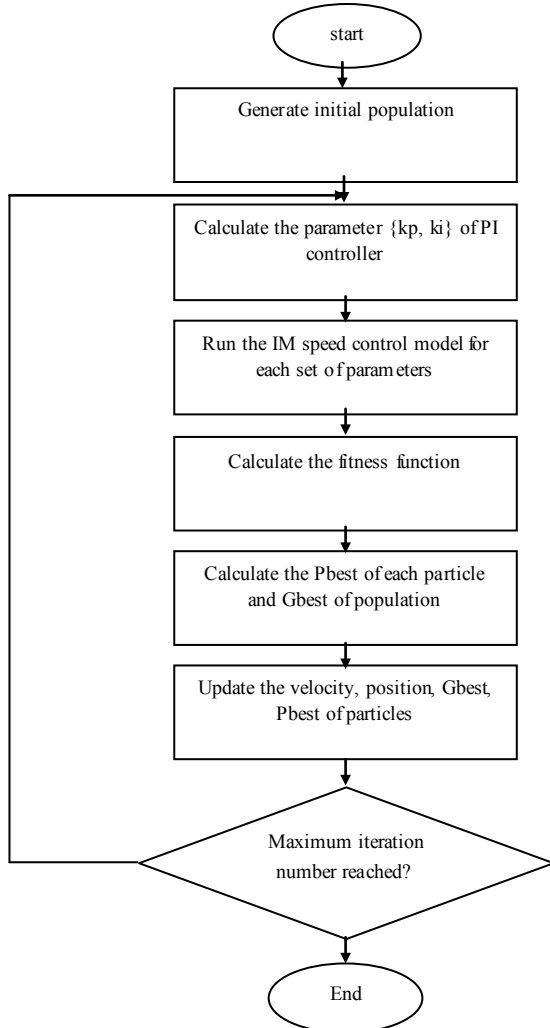


Figure 7. The Flowchart of the PSO-PI Control System technique

4.2. Fitness Function

In PI controller design methods, the most common performance criteria are integrated absolute error (IAE), the integrated of time weight square error (ITSE), integrated of squared error (ISE) and integrated of time weight absolute error (ITAE) that can be evaluated analytically in the frequency domain[18, 20]. These four integral performance criteria in the frequency domain have their own advantage and disadvantages. For example, disadvantage of the IAE and ISE criteria is that its minimization can result in a response with relatively small overshoot but a long settling time because the ISE performance criterion weights all errors equally independent of time. Although the ITSE performance criterion can overcome the disadvantage of the ISE criterion, the derivation processes of the analytical formula are complex and time-consuming[18,19]. The IAE, ISE, ITAE and ITSE performance criterion formulas are as follows:

$$IAE = \int_0^{\infty} |e(t)| dt \quad (15)$$

$$ISE = \int_0^{\infty} e^2(t) dt \quad (16)$$

$$ITAE = \int_0^{\infty} t \cdot |e(t)| dt \quad (17)$$

$$ITSE = \int_0^{\infty} t e^2(t) dt \quad (18)$$

In order study we will select the fitness function (FF) is reciprocal of the performance criterion, in the other words: $FF = 1/ITAE$.

In this paper a time domain criterion is used for evaluating the PI controller. A set of good control parameters P and I can yield a good response that will result in performance

criteria minimization in the time domain. These performance criteria in the time domain include the overshoot, rise time, settling time, and steady-state error.

5. PSO –PI Controller Selection

5.1. PSO –PI Controller

The control system parameters (k_p , k_i), for two-variable optimization problem, a flock of particles are put into the two-dimensional search space with randomly chosen velocities ($\Delta K_{i,1}$, $\Delta K_{i,2}$) and positions ($K_{i,1}$, $K_{i,2}$) knowing their best values so far (p_{best}), and the positions in the two-dimensional space. The velocity of each particle ($\Delta K_{i,1}$, $\Delta K_{i,2}$), adjusted according to its own flying experience and the other particle's flying experience. For example; the i -th particle is represented as $K_i = (K_{i,1}, K_{i,2})$. In the two-dimensional space, the best previous position of the i -th particle is recorded and represented as $P_{besti} = (P_{besti,1}, P_{besti,2})$. The best particle among all of the particles in the group is g_{best} , the velocity for particle i is represented as $\Delta K_i = (\Delta K_{i,1}, \Delta K_{i,2})$. After finding the two best values, the particle updates its velocity and position, and then each particle can be calculated using the current velocity and the distance from P_{best} to g_{best} as shown in the following formulas [21,22,23]:

$$\Delta K_{i,m}(t+1) = w * \Delta K_{i,m}(t) + c_1 * \text{Rand}() * (P_{besti,m} - K_{i,m}(t)) + c_2 * \text{Rand}() * (P_{bestg,m} - K_{i,m}(t)) \quad (19)$$

$$K_{i,m}(t+1) = K_{i,m}(t) + \Delta K_{i,m}(t+1) \quad i=1,2,\dots,n$$

$$m=2(P,I) \quad (20)$$

n = Number of particles in the group, c_1 , c_2 = Acceleration constant, t = Pointer of iterations (generations), $\Delta K_{i,m}(t)$ = Velocity of particle i at iteration t , w = Inertia weight factor, $\text{Rand}()$ = Random number between 0 and 1

$K_{i,m}(t)$ = Current position of particle i at iterations, P_{best} = Best previous position of the i -th particle, g_{best} = Best particle among all the particles in the population.

5.2. PSO-PI Controller Parameter Selection

To control the speed of an induction motor at 150 rad/sec, the following PSO parameters are used to verify the performance of the PSO-PI controller parameters shown in table 1.

Table 1. Parameters Selection of PSO Technique

Number of iterations	50
Population Size	14
Inertia weight factor= w	1
Acceleration constant = $c_1 = c_2$	2.035
The constriction factor= χ	1/1.3

Table 2. Variable of PI-PSO Controller

[k_p k_i]	[2296.6 0.4069]
Rise time	0.75
Maximum overshoot	0
Settling time	0.82
Steady States error	0
ITAE	18.534

Table 2. variable of PI-PSO controller. Particle swarm optimization Algorithm values are given as ($K_p = 2296.6$, $K_i = 0.4069$).

Figure 8 shows the rotor speed performance at $T_L = 2N.m$ and time zero load condition.

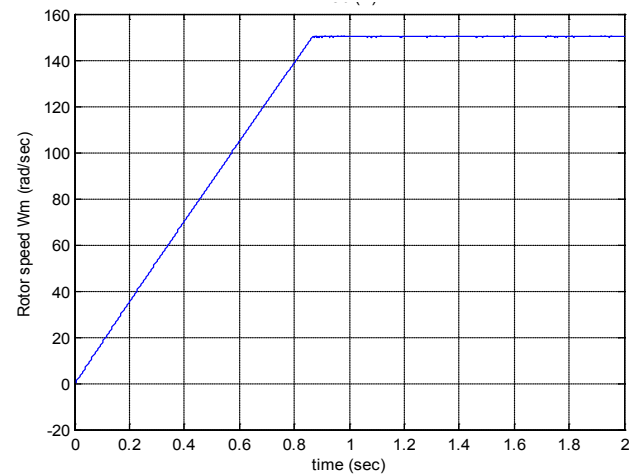


Figure 8. Speed in case of PSO-PI ($T_L = 2N.m$ at $t=0$ sec)

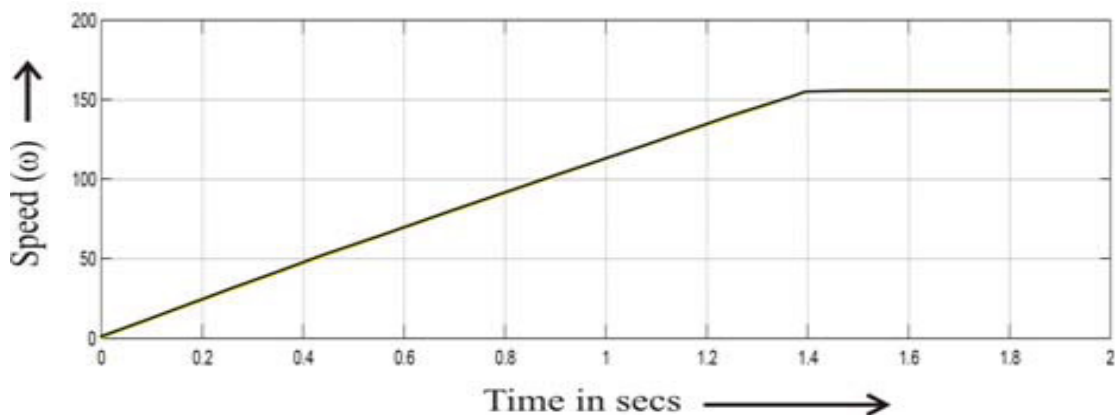


Figure 9. Speed in case of Mamdani control strategy [24]

From the simulation result shown in the Figure 8, it is observed that the response of the speed takes lesser time to reach settle value and obtains the desired value in a min time. A comparison study between the control performance using PSO with another technique used Mamdani control strategy[24] shown in the Figure 9, for the same model and the same parameter. The comparison shows that the motor speed reached to 0.82 sec compared with other technique that reached to 1.4 sec(ITAE=18.534).

Another case study is taken to check the performance of PSO. The test is accured at $T_L=200$, $t=1.2$ sec. The motor starts from standstill at load torque = 2 N.ms and, $t=0.0$ s, a sudden full load of 200 N.ms at $t=1.2$ sec is applied to the system controlled by PSO-PI control.

Figure 8 shows the speed characteristics during step change in load torque at $t = 1.2$ sec with PSO-PI controller. This figure shows that the PSO-PI-based drive system that can handle the sudden increase in command speed quickly without overshoot, under- shoot, and steady-state error (ITAE=18.3831).

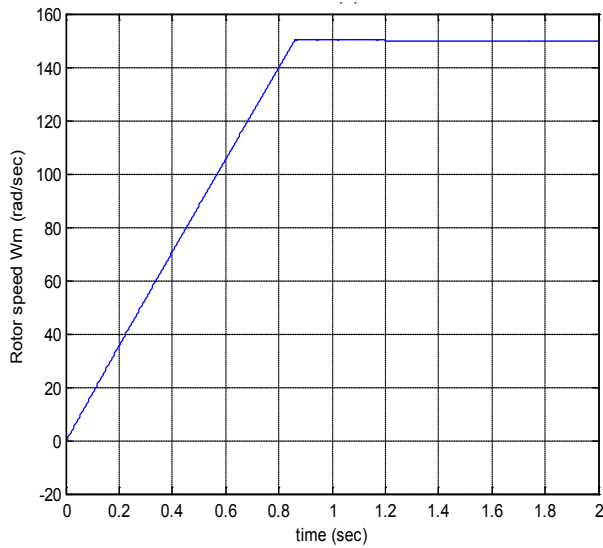


Figure 10. Speed in case of PSO-PI ($T_L=200$ N.m at $t=1.2$ sec)

6. Conclusions

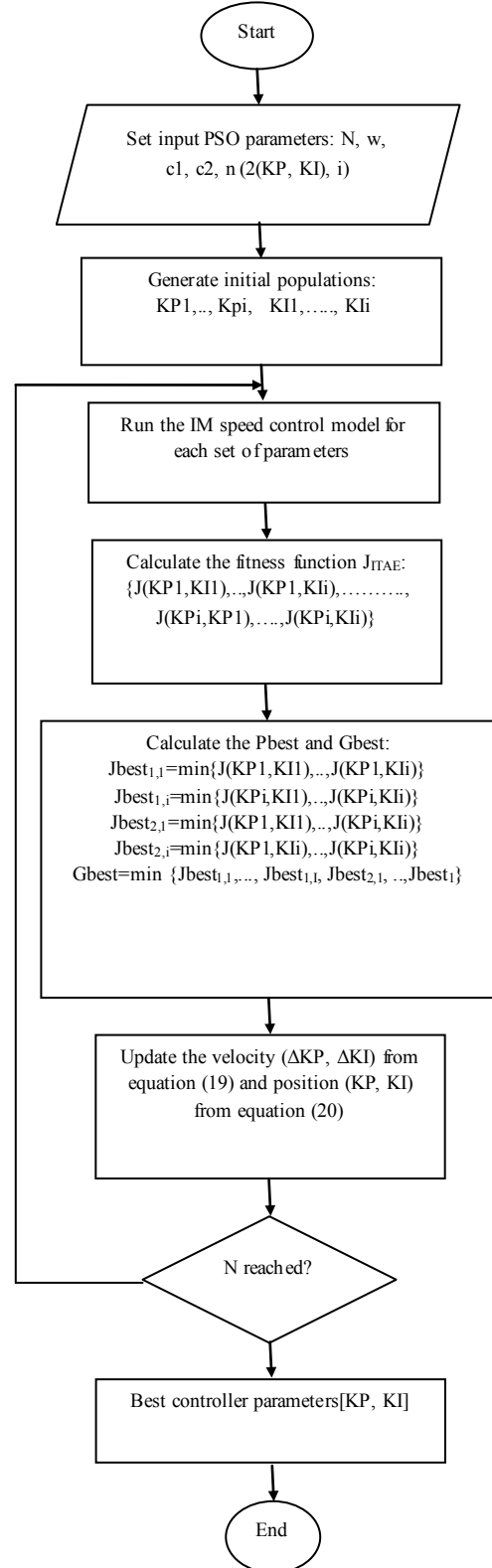
In this paper, a sincere attempt is made to reduce the settling time of the responses and make the speed of response very fast by designing an efficient controller using PSO-PI control strategy. Here, we have control strategy for the speed control of IM, which has yielded excellent results compared to the others mentioned in the literature survey above. The results of our work have showed a very low transient response and a non-oscillating steady state response with excellent stabilization.

APPENDIX

A1. Squirrel Cage Induction Motor (SCIM) specs:

50 HP, 1800 rpm, 460 V, 60 Hz. 3-Phase, 2 pair of poles, Squirrel Cage type IM, $R_s=0.087 \Omega$, $L_s=0.8$ mH, $R_r=0.228 \Omega$, $L_r=0.8$ mH, $L_m=34.7$ H, $J=1.662$ kg.m².

A2.The Flowchart of the PSO-PI Control System technique



REFERENCES

- [1] Denai, M.A. and Attia, S.A, "Fuzzy and Neural Control of an Induction Motor", Proc. Int. J. Appl. Math. Comput.Sci., Vol.12, No. 2, pp. 221-233, 2002.
- [2] Vas P., "Vector Control of AC Machines", Oxford University Press, London, UK, 1990.
- [3] Trzynadlowski, A.M., "The Field Orientation Principle in Control of Induction Motors", Kluwer Pub., Dordrecht, 1994.
- [4] Bose, B.K., "Modern Power Electronics and AC Drives", Pearson Education, Inc., India, 2002.
- [5] Mosskull,H., Gali'c,H., and WahlbergB. "Stabilization of Induction Motor Drives With Poorly Damped Input Filters", IEEE Transactions on Industrial Electronics, Vol. 54, No. 5, pp. 2724-2734, 2007.
- [6] Martins,C. A. and Carvalho, A. S. , "Technological Trends in Induction Motor Electrical Drives", IEEE Porto Power Tech Conference, Vol. 2, Sep. 2001.
- [7] Ben-BrahimL., "Improvement of the stability of the V/f controlled induction motor drive systems", IEEE Proceedings of the 24th Annual Conference, Vol. 2, pp. 859-864, 1998.
- [8] Bimal, K.B., Nitin, R.P. and Kaushik, R., "A neuro-fuzzy based on-line efficiency optimization control of a stator flux oriented direct vector controller IM drive", IEEE Trans. Industrial Electronics, Vol. 44, No. 2, pp. 270-273, Apr. 1997.
- [9] Haider, A. F., Mohamed, E. L., Lau, S. S. and Yang, M. M., "Fuzzy-SMC-PI Flux and Speed Control for Induction Motors", Proc. of RAM-2008, pp. 325-328, 2008.
- [10] Chen, T.C., and Hsu, J.U., "A fuzzy sliding mode controller for induction motor position control", IECON'94., 20th Int. Conf on Industrial Electronics, Control and Instrumentation, Vol. 1, pp. 44-49, 1994.
- [11] Kim, D.H., Kim H.S., Kim J.M., Won C.Y. and Kim S.C., "Induction motor servo system using variable structure control with fuzzy sliding surface", IEEE Int. Conf. Industrial Electronics, Control, and Instrumentation, Vol. 2, pp. 977-982, 1996.
- [12] Xie, K., Lin,T.Y. and Zhang, J., "The Takagi-Sugeno fuzzy model identification method of parameter varying systems", Proc. Rough Sets Current Trends Conf., RSCTC'98, Warsaw, Poland, Jun. 22-26, 1998.
- [13] Kennedy, J.andEberhart, R., "Particle swarm optimization", Proc. IEEE Int. Conf. on Neural Network, Vol. 4, p. 1942 – 1948, 1995.
- [14] Shi,Y.,andEberhart, R., "A modified particle swarm optimizer", Proc. 1998 Int. Conf. Evolutionary Computation, The IEEE World Congress on Computational Intelligence. Anchorage, p. 69 – 73,May 1998.
- [15] Clerc,M.,and Kennedy, J., "The particle swarm-explosion, stability, and convergence in multidimensional complex space", IEEE Trans. Evolutionary Computation, Vol. 6, p. 58 –73, 2002
- [16] Zhang Y., Jiang, Z. and Yu, X., "Indirect Field-Oriented Control of Induction Machines Based on Synergetic Control Theory", IEEE Int. Conf. on Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, 2008, pp. 20-24, 2008.
- [17] Mukherjee, V., and Ghoshal, S. P., "Intelligent particle swarm optimized fuzzy PID controller for AVR system", Electric Power Systems Research, V. 77, Issue 12, Oct. 2007, p. 1689-1698.
- [18] Mitsukura, Y., Yamamoto T., and Kaneda M., "A design of self-tuning PID controllers using a genetic algorithm", in Proc. Amer. Contr. Conf., San Diego, CA, June 1999, p. 1361-1365.
- [19] Popov A., Farag A., and Werner H., "Tuning of a PID controller Using a Multi-objective Optimization Technique Applied to A Neutralization Plant", 44th IEEE Conference on Decision and Control, and the European Control Conference 2005.
- [20] Kim T. H., Maruta I., and Sugie T., "Robust PID controller tuning based on the constrained particle swarm optimization", Automatica, Vol. 44, Issue 4, Apr. 2008, p. 1104 – 1110.
- [21] Kennedy J., and Eberhart, R., "Particle swarm optimization", Proc. IEEE Int. Conf. on Neural Network, Vol. 4, p. 1942 – 1948, 1995.
- [22] Shi Y., and Eberhart, R., "A modified particle swarm optimizer", Proc. 1998 Int. Conf. Evolutionary Computation, The IEEE World Congress on Computational Intelligence. Anchorage, May 1998, p. 69 – 73
- [23] Clerc, M. and Kennedy J., "The particle swarm-explosion, stability, and convergence multidimensional complex space", IEEE Trans. Evolutionary, in a computation, 2002, Vol. 6, p. 58 –73.
- [24] Kusagur, A., Kodad, S.F., and SankarRam, B.V., "AI based design of a fuzzy logic scheme for speed control of induction motors using SVPWM technique", Int. Jr. Comp. Sci. & Network Security, Vol. 9, No. 1, pp. 74 -80, 2009.