ISSN 2250-3153

Performance Comparison between GA and PSO for Optimization of PI and PID controller of Direct FOC Induction Motor Drive

Tista Banerjee , Sumana Chowdhuri, Gautam Sarkar, Jitendranath Bera

Abstract- This Paper presents a comparative study of Genetic Algorithm method (GA) and Particle swarm optimization (PSO) method to determine the optimal proportional-integral (PI) and proportional-integral-derivative (PID) controller parameters, for speed control of a Field Oriented Control (FOC) induction motor. The FOC induction motor has been modeled in MATLAB (SIMULINK); the GA and PSO algorithm has been programmed and implemented in MATLAB. Comparing with traditional Ziegler-Nicholson method and the evolutionary algorithms (EA), it has been observed that during optimizing the controller parameters of a FOC IM drive with PSO the performance of the controller is improved for the step input in speed control as well as for speed tracking problem more efficiently, than Ziegler – Nicholson method and EAs.

Index Terms- FOC (IM), fitness function, GA, PSO, PI Control, PID control

I. Introduction

Induction motors has been widely used in various industries due to its robustness maintenance free operation, better efficiency and lower cost. In different industries, wide range of speed control with fast torque response regardless of load variation is required. This can be achieved very efficiently for induction motor using Field Oriented Control (FOC) [1, 2]. For speed control of induction motor, PI (proportional-integral) and PID (Proportional-integral-derivative) controllers are generally used. To find out the optimum parameters of the controller to obtain a good closed loop response at different operating conditions is a trivial task and these parameters can be optimized by conventional tuning methods, such as Ziegler-Nicholson (Z-N) method [6]. Other tuning methods like pole placement optimization technique are also done [4]. Now a day, Evolutionary methods like Genetic Algorithm (GA), Particle Swarm Optimization (PSO) are used for tuning the parameters. These new tuning techniques can very efficiently solve complex problems like speed tracking problems, where demand speed is a complex function of time, where the conventional methods may not optimize the controller parameter so easily. Genetic Algorithm is a heuristics search method based on Charle's Darwin principle of Natural Selection which narrates 'the survival of the fittest' of each and every individual on earth. At each step, the GA selects individuals from the current population as parents and uses them to produce the offspring's for the next generations. The fitness of all the individual of the population is calculated and the convergence of the generation is based on this fitness criterion. It is well suited for its solving complex design optimization problem as it can handle discrete and continuous variables, nonlinearity and different constrain functions of a system, without requiring gradient information [9].

PSO is inspired by the ability of flock of birds or herd of animals to adapt to their environment. It was developed in 1995 by James Kennedy and Russ Eberhart while attempting to simulate the choreographed, graceful motion of the swarm of birds as a part of socio-cognitive study investigating the motion of collective intelligence in biological population. In PSO, a set of randomly generated solutions propagates in the designed space towards the optimal solution over a number of iteration based on large amount of information about the designed space [10,].

Both GA and PSO are similar in the sense that these two techniques are population based heuristic search methods and they approach for the optimal solution by updating generations. Since the two approaches are supposed to find a solution to a given objective function but employ different strategies and computation effort, it is appropriate to compare their performance. Many researchers have applied both GA and PSO in different fields of engineering e.g. In [17], the authors have compared GA with PSO for designing a TCSC based controller for power system stability improvement.

The major objective of this work is to compare efficiency of both PSO and GA optimization technique applied to a direct field oriented Control Induction motor drive for a simple speed demand problem as well as for a complex speed problem. Here both PSO and GA have been applied to search for the optimal PI and PID controller parameters of FOC IM drive. The error criteria for both the methods are set to improve transient error and steady state error. Hence the fitness function is taken here are Integral Square Error (ISE), Integral Absolute Error (IAE) and Integral Time Square Error (ITSE) [6,8,14]. The performance of both optimization techniques in terms of convergence rate, error minimization and time complexity are compared. The initial parameters are chosen in the neighborhood of the parameters obtained by tuning through Z-N methods.

II. BASIC PLAN OF THE WORK

In this work, optimization of the parameters of PI/PID controller for speed control of a FOC IM has been done using GA and PSO and a comparative study between these two methods has been carried out for a speed tracking problem where the demand speed is a time varying function. The design has been considered for an induction motor (3-Φ, 1.5 HP, 415V, 50Hz) in MATLAB SIMULINK environment. The gain parameters for a step input are initially calculated by Ziegler-Nicholson's optimization technique which serves as the parent for the first generation for both the algorithms, so as to optimize the system much earlier and faster than what it is supposed to be. The algorithms have been developed in matlab 'm-file' and have been interfaced with SIMULINK model as shown in Fig 1.

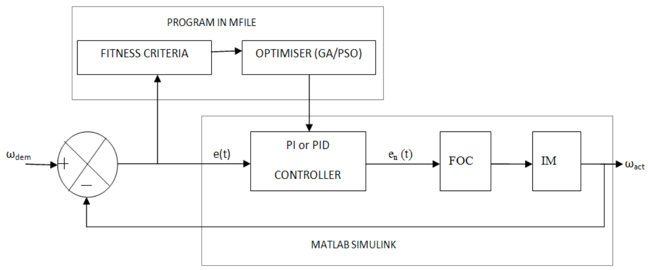

Fig.1: Schematic diagram of GA or PSO based optimizer for PI and PID controller of FOC IM

The PI and PID controller's gain parameters viz. K_p, K_i and K_d are optimized, by either GA or PSO, to have the optimum output of the controllers are given by Eqn.4 and Eqn.5 [6,7,8]. Here e(t) is the difference between the demand speed and the actual speed of the system is denoted by ω_{dem} and ω_{act} respectively. For optimization, three different, criterion namely IAE, ISE and ITSE are chosen which served as the fitness function of the algorithms shown in Eq.6. For the speed tracking problem, the parameters are optimized obeying the same procedure as stated above.

III. THEORITICAL BASICS

A. Field Oriented Control:

By Field Oriented Control (FOC) the transient response of IM improves since IM can be controlled like dc machine where its torque component and field flux component are separated virtually and independent control of each component is possible [1,3,8]. FOC is based on phase transformation, where a three phase time and space variant system is transformed to a synchronously rotating, time invariant system, leading to a structure similar to that of a dc machine [1, 2, 3, 4, 5,14]. Thus by this transformation, the three phase time varying system is transformed into two phase time-invariant system, where i_{qe} corresponds to the Armature component and i_{de} corresponds to the flux component like a dc machine. Hence the torque (T_e) of an IM can be computed as shown in Eq. 1

$$\left\{\begin{aligned} T_e &= K' \times \psi_r \times i_{qe} \\ T_e &= K' \times i_{de} \times i_{qe} \end{aligned}\right\} \ldots.(1)$$

where ψ_r is the peak value of the field flux space vector. In FOC control here i_{ds} is analogous to the field current, i_f and i_{qe} is analogous to armature current of the dc machine [1]. This means that when i_{qe} is controlled it affects the torque directly, ψ_r remaining unaffected. Similarly when i_{ds} is controlled it affects the flux only and does not affect the i_{qs} component of current. Thus an induction motor can be treated as a dc machine [1,8,14]. Independent direct torque control can be possible by only controlling the q-axis current control [1,3,4,8].

B. Genetic Algorithm:

Evolutionary Algorithm (EA) has been employed for solving optimization problems quite successfully. Instead of minimizing or maximizing the object function, it starts with an initial set of parameter values selected randomly. The objective function is evaluated with these parameters and those sets for which the value of the objective function is lower (or higher, as the need may be) are retained and other sets are manipulated. Thus a new set of parameters is evolved from the initial ones and the process is repeated until a "best" choice of parameters is obtained for which the objective function is minimum (or maximum)[8,15].

Among the various evolutionary approaches, Genetic Algorithm can effectively tackle the optimization problem. It is characterized by the chromosome representation, population size, crossover and mutation, their probability rate settings, selection mechanism and fitness function [8, 9,15].

GA requires encoding the solution of an optimization problem in the form of binary strings. The coded parameter in a string represents the chromosome of a particular individual in a population. A large population size incorporates more variation, i.e diversity into the population, but the convergence becomes slower [8, 9,15].

This ensures the possibility of producing individuals with better fitness. Two randomized methods are incorporated in the algorithm for producing future generations, crossover and mutation. In crossover a partial exchange of genes occur between two parent chromosomes [8,9,15]. The simplest way to achieve this is single point crossover where, a random location of chromosome is selected. If probability of a particular bit in a chromosome exceeds of a pre specified probability i.e., $p > p_{cross}$ (defined), the portion of the chromosome of one of the parent, preceding the selected point is combined with the portion of other parent, following the selected bit. In mutation, the parental characteristics are transferred with a slight change in gene[8,9,15]. A selected cell at random is toggled if the probability exceeds the probability of mutation, i.e. if $p > p_{mut}$. The choice of the probability rate for crossover and mutation is very much dependent on the population size[8,9,15].

The fitness function provides information regarding the goodness of a particular individual of a generation. Fitness functions generally consist of physical equations representing the objective function that is subject to optimization. In a GA, the fitter chromosomes are allowed to reproduce with higher probability and thus propagate into future generations [8.9,15]. According to their fitness, individuals are selected for next generation. This is called selection, which ensures that the latter generations contain fitter individuals. This selection mechanism may be deterministic or stochastic. Tournament selection involves running several "tournaments" among a few individuals chosen at random from the population. The winner of each tournament (the one with the best fitness) is selected for crossover. The performance of any selection procedure is guided by the biasing and it affects the time complexity [8, 9,15].

C. Particle swarm optimization:

Particle swarm optimization (PSO) is a population based stochastic optimization technique. It shares many similarities with evolutionary computation techniques such as genetic

ISSN 2250-3153

algorithm but the features of PSO, like easy way of implementation, stable convergence characteristics and computational efficiency has made it much superior than others [10, 11, 12, 13,14].

PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In every iteration, each particle is updated by following two "best" values. The first one is the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called p_{best}. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called g_{best} [10, 11, 17, 18]. When a particle takes part of the population as its topological neighbors, the best value is a local best and is called $l_{best.}$ After finding the two best values, the particle updates its velocity and positions with following equation for d^{th} generation [10, 11, 13, 14], i^{th} particle as shown in Eq. 4.

$$\left.\begin{array}{l} v_{id}[n+1] = v_{id}[n] + \\ \qquad c_1 \times rand() \times \left(pbest_{id}[n] - present_{id}[n]\right) + \\ \qquad c_2 \times rand() \times \left(gbest_{id}[n] - present_{id}[n]\right) \\ present_{id}[n+1] = present_{id}[n] + v_{id}[n+1] \end{array}\right\} \ldots.(2)$$

Where v[n] is the particle velocity, persent[n] is the current particle (solution). $P_{best}[n]$ and $g_{best}[n]$ are defined as stated before. Rand (n) is a random number between (0, 1). C_1, c_2 are learning factors. The value of c_1 and c_2 has been calculated as shown in eqn. 5

$$\left.\begin{array}{l} C_1 = (C_{1f} - C_{1i}) \times (\frac{i_{ter}}{m_{axiter}}) + C_{1i} \\ C_2 = (C_{2f} - C_{2i}) \times (\frac{i_{ter}}{m_{axiter}}) + C_{2i} \end{array}\right\} \ldots.(3)$$

where C_{1i}, C_{1f}, C_{2i}, and C_{2f} are constants, i_{ter} is the current iteration number and m_{axiter} is the number of maximum allowable iterations. The objective of this modification was to boost the global search over the entire search space during the early part of the optimization and to encourage the particles to converge to global optima at the end of the search. Actually C_1 was decreased from 2.5 to 0.5 whereas C_2 was increased from 0.5 to 2.5[10, 11, 12, 13, 14].

D. PI/PID controller:

The PID controller is a generic control loop feedback mechanism (controller) widely used in industrial control systems [6, 8, 15]. A PID controller calculates an "error" value as the difference between a measured process variable and a desired set point. The controller attempts to minimize the error by adjusting the process control inputs (6, 7, 8, 15). The PID controller calculation involves three separate parameters; the proportional, the integral and derivative values, and is given by

$$r(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{de(t)}{dt} \ldots.(4)$$

Where K_p, K_i, K_d are the proportional, integral and derivative gain of the system. u(t) is the input signal and e(t) is the error signal. For PI controller calculation involves two parameters, proportional and integral values and is given by

$$r(t) = K_p e(t) + K_i \int e(t)dt \ldots(5)$$

E. PI/PID performance criteria:

In PID controller design methods, the most common performance criteria are integrated absolute error (IAE), the integrated of time weight square error (ITSE) and integrated of squared error (ISE) that can be evaluated analytically in the frequency domain[6,8,14,15]. The IAE, ISE, and ITSE performance criterion formulas has been given below

$$\left.\begin{array}{l} IAE = \int_0^t |e(t)|dt \\ ISE = \int_0^t |e^2(t)|dt \\ ITSE = \int_0^t t \times |e^2(t)|dt \end{array}\right\} \ldots.(6)$$

3.6 Optimization of PI/PID controller using Ziegler-Nichol's method:

From all the methods designed to optimize PID controller, Ziegler and Nichols' method is mostly used[8,14,15]. The methods are based on characterization of process dynamics by a few parameters and simple equations for the controller parameters. The first method is applied to plants with step responses[8,14,15]. This type of response is typical of a first order system with transportation delays. The second method targets plants that can be rendered unstable under proportional control. The technique is designed to result in a closed loop system with 25% overshoot[8,14,15]. This is rarely achieved as Ziegler and Nichols determined the adjustments based on a specific plant model. Here the second methods have been used, K_{cr} is the gain at critical oscillation and P_{cr} is the time period[8,14,15]. The controller gains are specified in the table no:1.

Table1:PID Controller Gain calculation by z-n method

PID type	K_p	K_i	K_d
P	0.5 K_{cr}	∞	0
PI	0.45 K_{cr}	$P_{cr/1.2}$	0
PID	0.6 K_{cr}	$P_{cr/2}$	$P_{cr/8}$

ISSN 2250-3153

IV. FLOWCHART

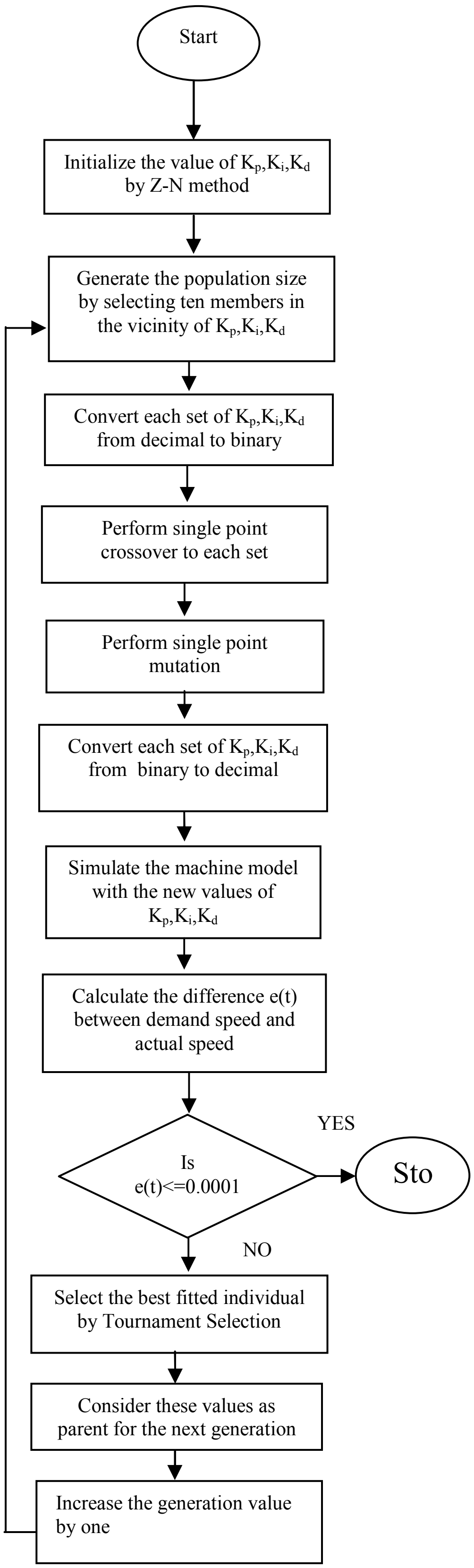

Fig2: Flow chart for optimizing PI/PID controller parametersusing GA

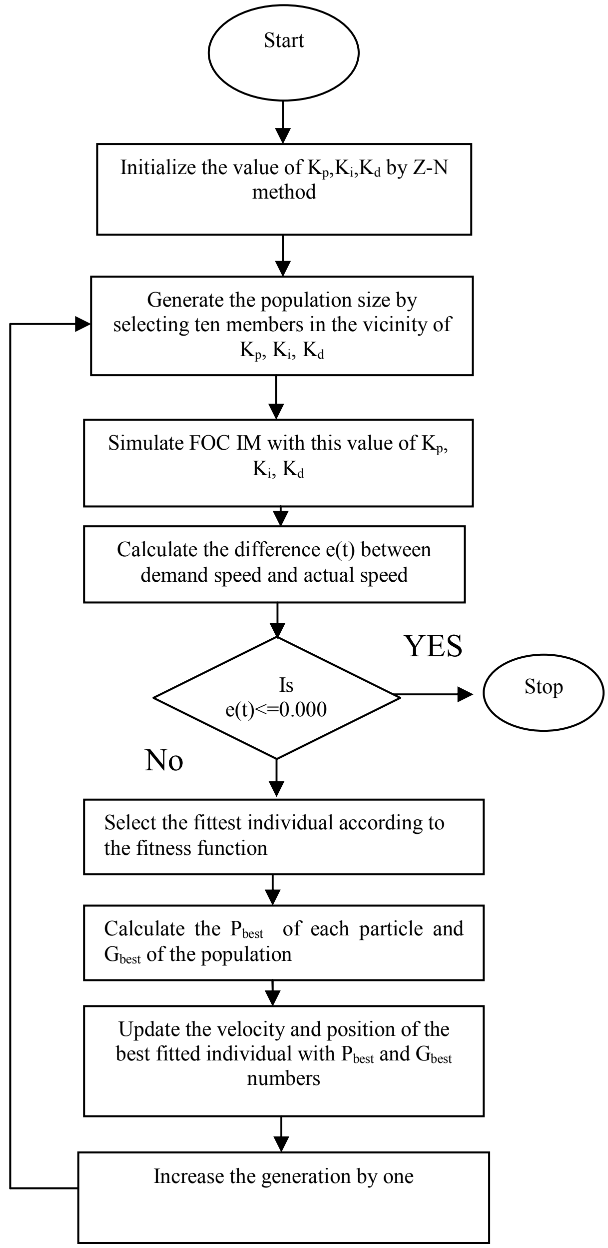

Fig3: Flow chart for optimizing PI/PID controller parameters using PSO

V. RESULTS

Initially a step input to the PI and PID controller has been given. As mentioned, initially the values of proportional, integral gain has been calculated using Z-N method and the system has been simulated and Fig 4 shows the result with this initial parameters settings. Then GA and PSO have been implemented, and the optimized result with GA and PSO has been shown in Fig 5 and Fig 6, respectively. Similarly with Z-N method the parameters of a PID controller for FOC controlled IM has been designed (Fig 7). With this initial values GA and PSO has been evaluated and the result of optimization (Fig 8, Fig 9) gives almost same results but computational time by PSO was found much less than by GA.

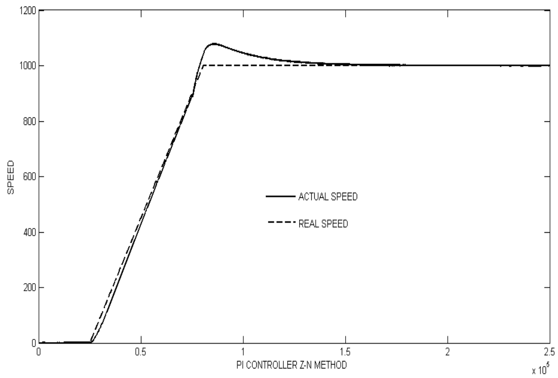

Fig 4: Step (speed) response of PI controlled FOC IM drive using Z-N method.

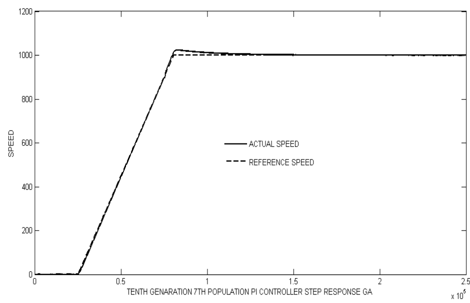

Fig 5: Step (speed) response of PI controlled FOC IM drive of tenth generation using GA.

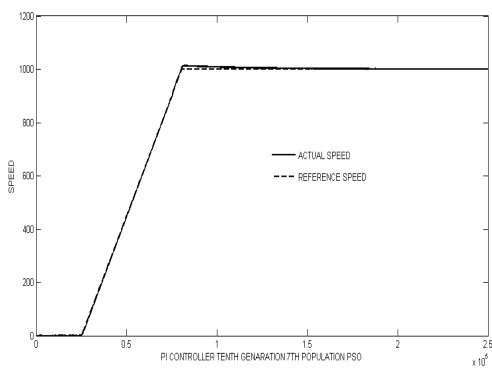

Fig 6: Step (speed) response of PI controlled FOC IM drive of tenth generation using PSO.

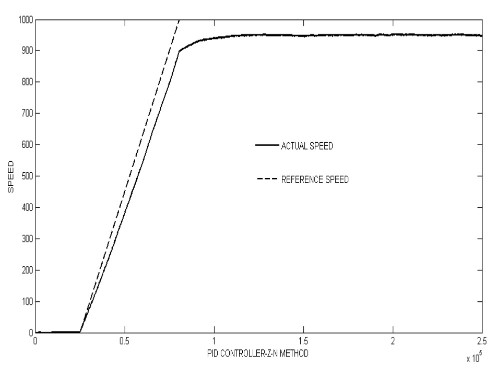

Fig 7: Step (speed) response of PID controlled FOC IM drive using Z-N method.

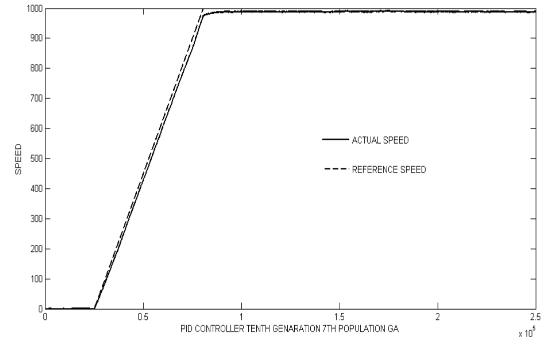

Fig 8: Step (speed) response of PID controlled FOC IM drive of tenth generation using GA.

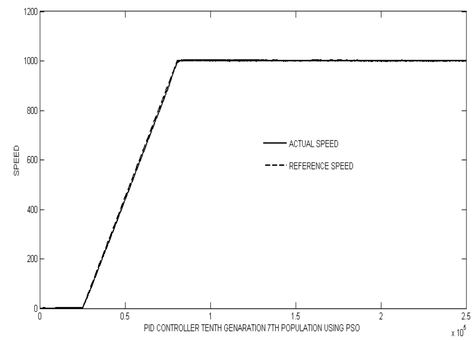

Fig 9: Step (speed) response of PID controlled FOC IM drive of tenth generation using PSO.

The convergence characteristics of GA and PSO has been shown below in the Fig 10,which shows that as the system progresses towards its optimum condition all the values of proportional, integral gain converges to one point. The first graph shows the convergence characteristics of PI controller using Z-N, initially the first generation populations has been plotted and the points were scattered but the optimized population tried to converge to one point. The second graph shows the convergence characteristics of the PI controller using GA, where the optimum condition was found at the tenth generation, and the third shows convergence characteristics using PSO, where the system almost converged at the eight generation but as the error speed was not less than 0.0001 so it took two more generation to converge. The first generation populations were almost linearly scattered but he optimized parameters converged to a point using PSO.

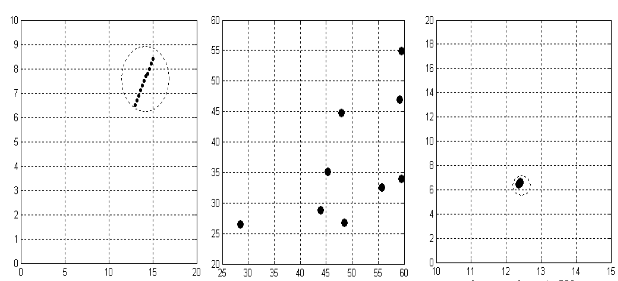

Fig.10: Convergence characteristics of GA and PSO used algorithm to optimize the PI controller

The following Table 1 and Table 2 shows the variation of maximum overshoot (M_p), peak time (T_p) and settling time (T_s)

over five generations of PI and PID controllers , first by using GA and then by PSO. It shows that the the transient response of PSO is better that than of GA. And the convergence characteristics of the gain parameters i.e proportional gain parameter K_p and Integral gain parameter K_i is better in PSO than GA. It has been observed that PSO gave better result than GA.

Table 1: Variation of settling time, maximum overshoot and peak time of ten populations over Ten generations of PI controller

Using GA						Using PSO					
Gen	Kp	Ki	Mp	Tp	Ts	Gen	Kp	Ki	Mp	Tp	Ts
1st	10.23	10.21	17	1.65	2.9	1st	7.08	14.5	3.2	1.75	2.8
2nd	52.4	52.47	12	1.6	2.83	2nd	6.56	12.34	3.1	1.75	2.7
3rd	69.29	69.2	9	1.6	2.72	3rd	7.1089	12.36	3.1	1.75	2.6
4th	28.08	28.08	5	1.6	2.6	4th	7.2	12.37	3.05	1.75	2.4
5th	30.38	30.37	4	1.6	2.3	5th	7.1147	12.53	3	1.75	2.2
6th	54.9800	59.4300	3.7	1.6	2.28	6th	7.1197	14.5951	2.9	1.75	2.16
7th	28.8500	43.9400	3.32	1.6	2.28	7th	7.1240	14.6051	2.9	1.74	2.10
8th	26.7700	48.4500	3.1	1.59	2.25	8th	7.1277	12.4594	2.7	1.72	2
9th	46.9100	59.1300	2.8	1.59	2.2	9th	6.6105	10.6212	2.64	1.72	1.9
10th	44.8100	47.9400	2.7	1.59	2.1	10th	6.1302	10.6017	2.5	1.70	1.8

Table 2: Variation of settling time, maximum overshoot and peak time of ten populations over Ten generations of PID controller

Using GA							Using PSO						
Gen	Ki	Kp	Kd	Mp	Tp	Ts	Gen	Kp	Ki	Kd	Mp	Tp	Ts
1st	10.23	10.21	0.25	17.3	1.77	3.7	1st	7.08	14.5	1.59	6	2.5	3.2
2nd	52.4	52.47	0.27	13.2	1.75	3.6	2nd	6.56	12.34	1.48	5.7	2.5	3.2
3rd	69.29	69.2	0.51	12	1.8	3.5	3rd	7.1089	12.36	1.37	3	2.5	3.15
4th	28.08	28.08	0.25	8.0	1.8	3.3	4th	7.2	12.37	1.27	2.8	2.4	3.1
5th	30.38	30.37	0.51	6.7	1.8	3.2	5th	7.1147	12.53	1.1	2.6	2.3	3
6th	54.9800	59.4300	0.7600	6.2	1.77	3.17	6th	7.1197	14.5951	1.0709	2.4	2.3	2.8
7th	28.8500	43.9400	1.020	6	1.77	3.14	7th	7.1240	14.6051	1.0713	2.1	2.3	2.76
8th	26.7700	48.4500	1.0200	5.8	1.76	3.1	8th	7.1277	12.4594	1.0716	2	2.28	2.7
9th	46.9100	59.1300	0.7600	5.5	1.76	2.9	9th	6.6105	10.6212	1.0710	1.8	2.28	2.7
10th	44.8100	47.9400	0.7600	4	1.76	2.8	10th	6.1302	10.6017	1.0602	1.8		2.7

The complexity curve for PI and PID controllers has been shown in the Fig 11, which shows that time complexity of PSO is lesser than GA.

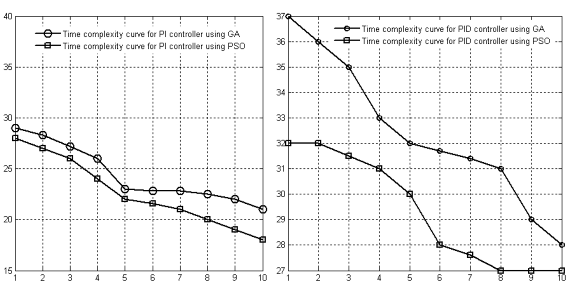

Fig 11: Complexity curve for PI and PID controllers speed Response over ten generations

Now the Table 3 shows the error between the demand speed and the actual speed of the FOC IM system that has been calculated by using various performance indices IAE, ISE, ITSE at different value of demand speed amongst which ITSE gave better result when used with PSO algorithm for PI controller.

Table 3: Comparision table showing different performance indices at different speed for PI Controller of tenth generation tenth population

Speed	Using GA			Using PSO		
	IAE	ISE	ITSE	IAE	ISE	ITSE
100	0.000031	0.000030	0.000031	0.00037	0.00027	0.00022
400	0.000313	0.00029	0.00028	0.00033	0.00026	0.00028
800	0.000303	0.000267	0.000259	0.00012	0.000211	0.00017
1000	0.000301	0.000251	0.00025	0.00017	0.0000162	0.00015
1500	0.000021	0.000020	0.000019	0.00011	0.000015	0.000134
1800	0.000019	0.0000181	0.000012	0.000011	0.0000189	0.00001
2000	0.000009	0.0000081	0.000006	0.000002	0.000017	0.000001

The Table 4 shows the error between the demand speed and the actual speed of the FOC IM system that has been calculated by using various performance indices IAE, ISE, ITSE at different value of demand speed amongst which ITSE gave better result when used with PSO algorithm for PID controller.

Table 4: Comparison table showing different performance indices at different speed for PID Controller for tenth population of the tenth generation

Using GA				Using PSO		
speed	IAE	ISE	ITSE	IAE	ISE	ITSE
100	0.003643	0.003107	0.004341	0.004843	0.004210	0.003527
400	0.00493	0.004636	0.001936	0.003774	0.003192	0.000344
800	0.003082	0.003065	0.00613	0.003143	0.003123	0.0002817
1000	0.002701	0.002471	0.00215	0.003400	0.002800	0.02053
1500	0.000197	0.00146	0.00066	0.0003595	0.0000812	0.000061
1800	0.000169	0.000051	0.000050	0.0000150	0.0000135	0.0000137
2000	0.000009	0.0000032	0.0000021	0.0000053	0.0000039	0.0000015

From the above tables and figures it can be said that higher generation, integral, proportional and derivative gain parameters were much better evolved than the first generation one. The setteling time (T_s), Maximum Overshoot (M_p), peak time (T_p) has reduced considerably generation wise as shown in table 1 and in table 2 for PI and PID Controller respectively. The steady state error at tenth generation is observed by setting different performance criteria (ASE, ISE, ITSE) as fitness function and tabulated for PI and PID Controller Table 3 and Table 4 respectively for GA and PSO. All the results for GA were better than Z-N method but PSO is even better than GA. The result shows very little difference for different fitness function. ITSE is better than the other in both type of evolutionary algorithms.

After this a speed tracking problem has been considered where the speed demand consists of two step input with 1.5s delay time and it has been applied to FOC controlled IM as shown in Fig 12. Just like the previous one, the PI and PID controllers has been optimized by using Z-N method first, and then the Evolutionary algorithms has been used to optimize the same. Here also it has been seen that PSO gave better result that optimization with GA as shown in the following figures.

ISSN 2250-3153

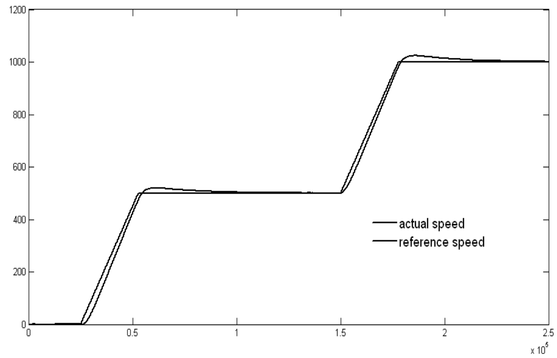

Fig12: Step (speed) response of PI controlled FOC IM drive of tenth generation using Z-N Method for the speed tracking problem

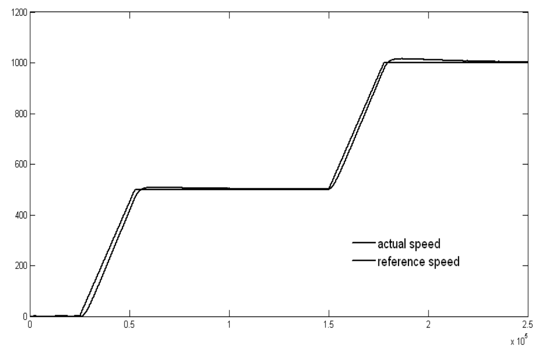

Fig 13: Step (speed) response of PI controlled FOC IM drive of tenth generation using GA for the speed tracking problem

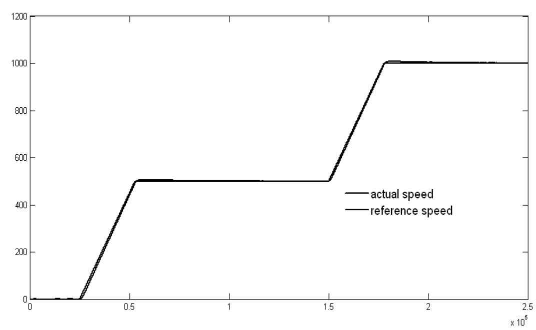

Fig 14: Step (speed) response of PI controlled FOC IM drive of tenth generation using PSO for the speed tracking problem

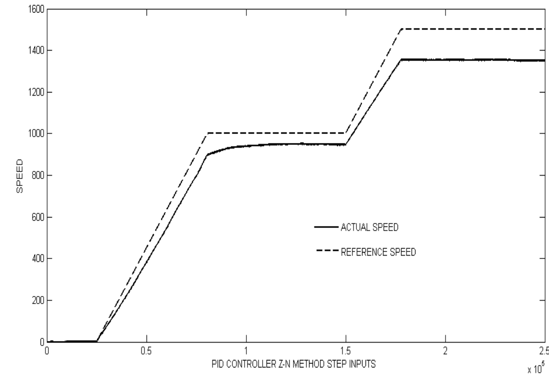

Fig.15: Step (speed) response of PID controlled FOC IM drive using Z-N method for speed tracking problem

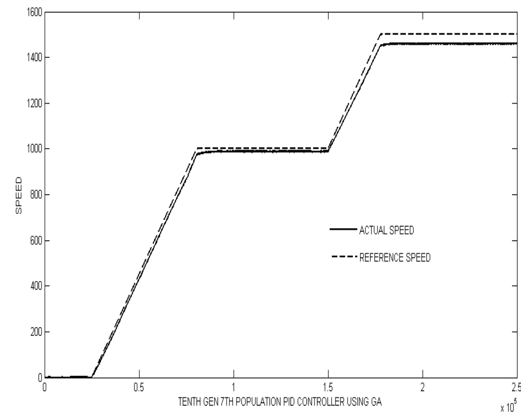

Fig 16: Step (speed) response of PID controlled FOC IM drive of tenth generation tenth population using GA for speed tracking problem.

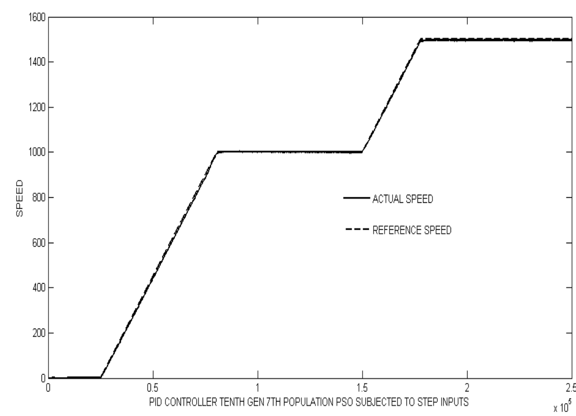

Fig 17: Step (speed) response of PID controlled FOC IM drive of fifth generation tenth population using PSO for speed tracking problem

Thus from Fig 12 the response has become much better in Fig 13 and Fig 14 for the PI controller and Fig 17 and Fig 18 was better than Fig 15 for the PID controllers. But Fig 14 and Fig 17 was even better than Fig 13 and Fig 16 i.e the response due to PSO algorithm was much better than both GA and Z-N method.

VI. CONCLUSION

Thus it can be concluded that, the evolutionary algorithms provide much better speed response than that of conventional Z-N method but among them PSO works with much better efficiency as computational time minimizes, simple, and has stable convergence characteristics than GA

REFERENCES

[1] Bose,B.K., "Modern Power Electronics and AC Drives", (Pearson Education Pte. Ltd ,First Indian reprint 2002,third reprint 2003)

[2] Blaschke ,F., "The Principle of Field Orientation as applied to the new transvector closed loop control system for rotation field machine", Siemens Rev XXXXIX, 1972,200-217

[3] "Field Orientated Control of 3-PhaseAC-Motors", Texas Instruments Europe , Literature Number: BPRA073 February 1998

[4] Casadei, D., Profumo , F. , Serra, G., Tani,A., "FOC and DTC: two viable schemes for induction motors torque control", IEEE trans. Power Electron, 17(September) (2002),779-787.

[5] Novotonyand, R., Tipo,T., "Introduction to field orientation and high performance ac drives", IEEE Ind , Applicat.Soc.Ann Meeting , Tutorial Course1986.

[6] Ogata,Katshuhiko.,"Modern Control Engineering" (PHI Learing Private Limited ,Fifth edition,2010)Ang ,K., Chong, G., Li,Y., "PID control system analysis, design, and technology," IEEE Trans.Control System Technology, vol. 13, pp. 559- 576, July 2005

[7] Kapsiotis,George., Tzafestas, Spyros., "PID self –tuning control combining pole placement and parameter optimization features (original research article) Mathematics and Computers in Simulation", Volume 37, Issues2-3,30November1994,Pages133-142

[8] Banerjee, T., Choudhury, S., Bera, J.N, "Offline optimization of PI/PID controller for a Vector Controlled Induction Motor Drive using Genetic Algorithm.-International Conference on "Electrical Power and Energy Systems" ICEPES 2010, 26-28th August 2010. On Pages192-195.

[9] Krohling, R. A., Jaschek,H., Rey, J. P., "Designing PI/PID controller for a motion control system based on genetic algorithm", in Proc. 12th IEEE Int. Symp. Intell. Contr., Istanbul, Turkey, July 1997, pp. 125–130.

[10] Eberhart,R., Kennedy,J., "Particle swarm optimization," in Proc. IEEE Int. Conf. Neural Networks, vol. IV, Perth, Australia, 1995, pp. 1942–1948.

[11] Mehdi, Nasri., Hossein Nezamabadi-pour, and Malihe Maghfoori, "A PSO-Based Optimum Design of PID Controller for a Linear Brushless DC Motor", in World Academy of Science, Engineering and Technology 26 2007

[12] Gaing, Z.-L., "A particle swarm optimization approach for optimum design of PID controller in AVR system," IEEE Trans. Energy Conversion, vol. 19, pp. 384-391, June 2004

[13] Particle Swarm Optimisation,http://www.swarmintelligence/tutorial.php

[14] Banerjee, T., Choudhury, S., Bera, J.N,Maity,A., "Off-line optimization of PI and PID controller for a vector controlled induction motor drive using PSO", (ICECE), 2010 International Conference on Electrical and Computer Engineering, 18-20 Dec. 2010, On Page(s): 74 – 77, ISBN: 978-1-4244-6277-3

[15] Mahapatra, T.S,Roy, R., Mishra, S., Saha, N., Banerjee, T., Das, P., "Offline Optimisation of PI/PID Controller of a DC Shunt Machine Drive using Genetic Algorithm', ICNEAC 2011, July 8-10 2011, Narsapur, http://www.iacqer.com.

[16] "Parameter estimation of an induction machine using advanced particle swarm optimization algorithms", IET Electr. Power Appl , November 2010,Volume 4, Issue 9, p.748–760

[17] Panda, Sidhartha., Padhy, "Comparison of particle swarm optimization and genetic algorithm for TCSC-based controller design", Department of Electrical Engineering, Indian Institute of Technology, Roorkee, Uttaranchal 247667, India, International journal of Electrical and Electronics Engineering 1:5 2007. http://www.waset.org/journal/ijeee/v1/v1-5-46.pdf

AUTHORS

First Author – T. Banerjee is with the B.P.Poddar Institute of Management and Technology under WBUT, Kolkata-700 052,India (ph. 9748632295 e-mail: tista.banerjee@gmail.com)
Second Author – S. Chowdhuri., G. Sarkar and JN Bera are with Applied Physics Department of Calcutta University, Kolkata, India.(e-mail: sumana_cu05@rediffmail.com,gsaphy@caluniv.ac.in, jitendrabera@rediffmail.com).