DOI: 10.21917/ijsc.2010.0014

HYBRID EVOLUTIONARY ALGORITHMS FOR FREQUENCY AND VOLTAGE CONTROL IN POWER GENERATING SYSTEM

A. Soundarrajan[1], S. Sumathi[2] and G.Sivamurugan[3]

[1] *Department of Information Technology, PSG College of Technology, Tamil Nadu, India*
E-mail: soundaa@yahoo.com

[2,3] *Department of Electrical & Electronics Engineering, PSG College of Technology, Tamil Nadu, India*
E-mail: ssi@eee.psgtech.ac.in

Abstract

Power generating system has the responsibility to ensure that adequate power is delivered to the load, both reliably and economically. Any electrical system must be maintained at the desired operating level characterized by nominal frequency and voltage profile. But the ability of the power system to track the load is limited due to physical and technical consideration. Hence, a Power System Control is required to maintain a continuous balance between power generation and load demand. The quality of power supply is affected due to continuous and random changes in load during the operation of the power system. Load Frequency Controller (LFC) and Automatic Voltage Regulator (AVR) play an important role in maintaining constant frequency and voltage in order to ensure the reliability of electric power. The fixed gain PID controllers used for this application fail to perform under varying load conditions and hence provide poor dynamic characteristics with large settling time, overshoot and oscillations. In this paper, Evolutionary Algorithms (EA) like, Enhanced Particle Swarm Optimization (EPSO), Multi Objective Particle Swarm Optimization (MOPSO), and Stochastic Particle Swarm Optimization (SPSO) are proposed to overcome the premature convergence problem in a standard PSO. These algorithms reduce transient oscillations and also increase the computational efficiency. Simulation results demonstrate that the proposed controller adapt themselves appropriate to varying loads and hence provide better performance characteristics with respect to settling time, oscillations and overshoot.

Keywords:

Load Frequency Control (LFC), Automatic Voltage Regulator (AVR), Evolutionary Algorithm (EA), Enhanced Particle Swarm Optimization (EPSO), Multi Objective Particle Swarm Optimization (MOPSO), and Stochastic Particle Swarm Optimization (SPSO)

1. INTRODUCTION

In recent years, the performance of computers has a great influence over the power systems in maintaining quality and reliable power supply. Also the power system can be controlled easily and efficiently with higher degree of reliability. The growth in size and complexity of electric power systems along with increase in power demand has initiated the need for intelligent systems that combine different techniques and methodologies. The intelligent systems possess human like expert knowledge and adapt themselves in changing environments. In electric power system, as the demand deviates from its normal value with an unpredictable small amount, the state of the system changes. The automatic control system detects these changes and initiates in real time as set of control actions which will eliminate as effectively and quickly as possible the state deviations. The active and reactive power demands are never steady they continuously change with rising and falling trend [1]. Steam input to turbo- generators must therefore be continually regulated to match the active power demand, failing which the machine speed will vary with consequent change in frequency which may be highly undesirable. Also the excitation of generators must be continually regulated to match the reactive power demand with reactive generation; else the voltages at various system buses may go beyond the prescribed limits [2].

Automatic Generation Control (AGC) is used in real-time control to match the area generation changes to area load changes in order to meet tie-line flows and keep frequency at nominal value. By processing frequency and tie-line deviations, AGC can determine the load changes of its own area and in its neighboring area. The function of AGC is to reallocate the generation changes to pre-selected machines after an initial random accommodation of the load by governor action. It is necessary to obtain much better frequency constancy than obtained by speed governor itself [3]. For successful operation of the power system, the load must be fed with constant voltage and frequency. Hence, a suitable control strategy has to be developed to accomplish this task. In practice different control strategies are utilized for AGC like, Proprotional and Integral (PI), Proportional, Integral and Derivative (PID) and optimal control [4]. But these conventional controllers have their own limitations like slow and lack of efficiency in handling system non-linearities [5]. The optimal control is quite often impractical for the implementation, since accurate prediction of load demand is necessary [6].

Several new optimization techniques like Genetic Algorithm (GA), PSO, Ant Colony Optimization (ACO), Simulated Annealing (SA) and Bacterial Foraging have emerged in the past two decades that mimic biological evolution, or the way biological entities communicate in nature [7]. Evolutionary Algorithm has received greater attention in recent past for solving optimization problems. Evolution is an optimization process, where the aim is to improve the ability of individuals to survive. EA is the emulation of the process of natural selection in a search procedure and it is an efficient Meta heuristic method. It finds the solution from the population and not from the single particle and so it can give the global optimum solution from the entire population. EA can be used for wide range of applications like Engineering design, combinational optimization problems and real time problems [7].

Due to its high potential for global optimization, GA has received great attention in control system such as the search of optimal PID controller parameters. The natural genetic operations would still result in enormous computational efforts. The premature convergence of GA degrades its performance and reduces its search capability. Particle swarm optimization (PSO), first introduced by Kennedy and Eberhart, is one of the modern heuristics algorithms. It was developed through simulation of a simplified social system, and has been found to be robust in solving continuous non-linear optimization problems [8]. Researchers including Zwe-Lee Gaing have presented PSO for optimum design of PID controller in AVR system [9]. A new time-domain performance criteria function was defined to estimate their system. To improve the performance of the PID controller, You-bo Wang et al has presented the use of new PSO based auto tuning of PID controllers [10]. Haluk gozde et al has used the PSO algorithm for optimizing PID values of LFC in a single area power system [11]. Their contribution includes selection of optimum parameter value for the integral gain and proportional gain was made equal to the regulation R. All these works have been reported for implementing intelligent techniques for controlling voltage and frequency separately. Hence, a novel approach of combined intelligent control of voltage and frequency in a single area power system is proposed in this paper. The objective of this work is to design and implement an EA based PID controller to search the optimal parameters for efficient control of voltage and frequency. The model of the LFC and AVR of single area power system is designed using simulink in MATLAB. The algorithm was developed to generate the optimum Proportional, Integral and Derivative gains of the controller. These values are sent to workspace and shared with the simulink model for simulation under different loads and regulation parameters. The proposed LFC and AVR contribute to the satisfactory operation of the power system by maintaining system voltages and frequency.

The paper is organized as follows, Section 2 describes the linearized model of the plant, Section 3 describes the theory of Evolutionary Algorithm, Section 4 demonstrates the design of EA based PID controller, Section 5 shows simulation results, Performance comparison of different evolutionary controllers are given in section 6. Section 7 indicates the computational efficiency of EA based controllers and conclusion is derived in Section 8.

2. LINEARIZED MODEL OF THE PLANT

2.1. BASIC GENERATOR CONTROL LOOPS

In an interconnected power system, LFC and AVR equipment are installed for each generator. The schematic diagram of the voltage and frequency control loop is represented in fig.1. The controllers are set for a particular operating condition and take care of small changes in load demand to maintain the frequency and voltage magnitude within the specified limits [3].

Small changes in real power are mainly dependent on changes in rotor angle δ and, thus, the frequency f. The reactive power is mainly dependent on the voltage magnitude (i.e. on the generator excitation). Change in angle δ is caused by momentary change in generator speed. Therefore, load frequency and excitation voltage controls are non-interactive for small changes and can be modeled and analyzed independently. Furthermore, excitation control is fast acting while the power frequency control is slow acting since, the major time constant contributed by the turbine and generator moment of inertia-time constant is much larger than that of the generator field. Thus, the cross-coupling between the LFC loop and the AVR is negligible, and the load frequency and excitation voltage control are analyzed independently [1] [6].

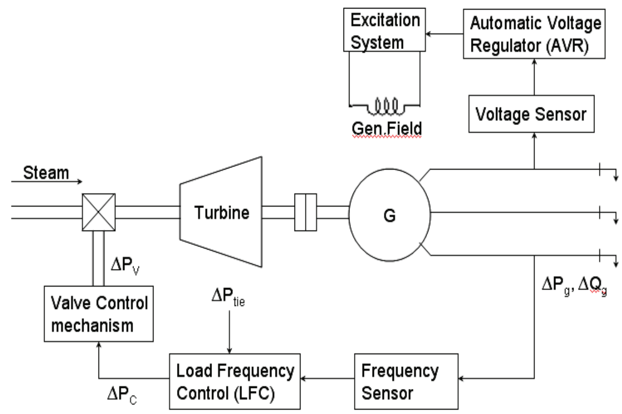

Fig.1. Schematic diagram of LFC and AVR of a synchronous generator

2.2 LOAD FREQUENCY CONTROL (LFC)

The aim of LFC is to maintain real power balance in the system through control of system frequency. Whenever the real power demand changes, a frequency change occurs. This frequency error is amplified, mixed and changed to a command signal which is sent to turbine governor. The governor operates to restore the balance between the input and output by changing the turbine output. This method is also referred as Megawatt frequency or Power-frequency (P-f) control.

2.3. AUTOMATIC VOLTAGE REGULATOR (AVR)

The aim of this control is to maintain the system voltage between limits by adjusting the excitation of the machines. The automatic voltage regulator senses the difference between a rectified voltage derived from the stator voltage and a reference voltage. This error signal is amplified and fed to the excitation circuit. The change of excitation maintains the VAR balance in the network. This method is also referred as Megawatt Volt Amp Reactive (MVAR) control or Reactive-Voltage (QV) control. The simulink models of load frequency controller and automatic voltage regulator is constructed based on the block diagram approach as proposed by Hadi Sadaat [1].

3. EVOLUTIONARY ALGORITHMS

Evolutionary Algorithm (EA) is a basic search algorithm, which is derived from the Darwin's Theory of Evolution, proposed in 1859. According to the Darwin's theory, if an environment can host only a limited number of populations then each and every individual in the environment tries to attain the best position. As a result, the individuals will begin to compete among themselves for the given resources to attain a better position and at last, the individuals that are best fit to the environmental conditions will survive. This phenomenon is also known as 'survival of the fittest'. The mechanisms used by EAs are inspired from biological evolution to find the solution for the given problem. It performs well in approximating a set of solution for all types of problems because they ideally do not make any assumption about the underlying fitness landscape. The Evolutionary process is simulated in a computer and hence, millions of generations can be executed in a matter of hours and can be repeated under various circumstances. Evolution is an optimization process, where the aim is to improve the ability of individuals to survive. Evolution via natural selection of a randomly chosen population of individuals can be thought of as a search through the space of possible chromosome values. In that sense, an EA is a stochastic search for an optimal solution to a given problem. An EA utilizes a population of individuals, where each individual represents a candidate solution to the problem.

The EA selects the best fit value from the given population and it is less sensitive to the scaling of algorithm parameters. EA also has good fault tolerance and it also takes care of social and cognition behavior of the individual particle. In Evolutionary algorithm, the self adaption of the particle is an important strategy which varies the EA parameters during run time in a specific manner. This feature is inherent in modern evolution strategies and it is provided for the context of changing fitness landscapes also. In this case, even the objective function changes, the EA always aims at the moving target i.e. the present population will be reevaluated, and quite naturally it is tested whether individuals have been adapted to the new objective function. The EA techniques provide robust performance and global search characteristics. The most significant advantage of using EA technique is the gain of flexibility and adaptability to the task [12].

In this paper, the objective is to find the optimum values of Proportional, Integral and Derivative gains of a PID controller. These optimum gains are used in AVR and LFC models of the power system. The PID controller based Automatic Voltage Regulator (AVR) and Load Frequency Controller (LFC) is designed to maintain the terminal voltage of the generating system in constant level and to reduce the frequency deviation. The evolutionary algorithms used in this paper are Enhanced Particle Swarm Optimization (EPSO), Multi Objective Particle Swarm Optimization (MOPSO), Stochastic Particle Swarm Optimization (SPSO).

3.1 BASIC PSO AND ITS VARIANTS

Particle swarm optimization (PSO) is a stochastic population based optimization algorithm, firstly introduced by Kennedy and Eberhart in 1995 [13]. In PSO algorithm, each member of the population is called a ''particle'', and each particle ''flies'' around in the multidimensional search space with a velocity, which is constantly updated by the particle's own experience and the experience of the particle's neighbours or the experience of the whole swarm. It has already been applied in many areas, such as function optimization, artificial neural network training, pattern classification and fuzzy system control. The advantages of PSO are that PSO is rapidly converging towards an optimum, simple to compute, easy to implement and free from the complex computation in genetic algorithm (e.g., coding/decoding, crossover and mutation). However, PSO does exhibits some disadvantages: it sometimes easy to be trapped in local optima, and the convergence rate decreased considerably in the later period of evolution; when reaching a near optimal solution, the algorithm stops optimizing, and thus the accuracy the algorithm can achieve is limited.

In standard PSO algorithm, each particle in the swarm represents a solution to the problem and it is defined with its position and velocity [9]. In D-dimensional search space, the position of the *i*th particle can be represented by a D-dimensional vector, $X_i = (X_{i1},\dots, X_{id}, \dots, X_{iD})$. The velocity of the particle v_i can be represented by another *D*-dimensional vector $V_i = (V_{i1},\dots, V_{id}, \dots, V_{iD})$. The best position visited by the *i*th particle is denoted as $P_i=(P_{i1},\dots,P_{id},\dots,P_{iD})$, and P_g as the index of the particle visited the best position in the swarm, then P_g becomes the best solution found so far, and the velocity of the particle and its new position will be determined according to the Eq. (1) and (2). [10].

$$V_{id}= WV_{id}+ C1\ R\ (P_{id}-X_{id})+ C2\ R\ (P_{gd}-X_{id}) \tag{1}$$

$$X_{id}= X_{id}+ V_{id} \tag{2}$$

The parameter W in (1) is inertia weight that increases the overall performance of PSO. It is reported that a larger value of W can favor higher ability for global search while lower value of W implies a higher ability for local research. To achieve a higher performance, we linearly decrease the value of inertia weight W over the generations to favor global research in initial generations and local re-search in the later generations. The linearly deceased value of inertia is according the (3) [14]

$$W=W_{max} - iter * \frac{W_{min}- W_{min}}{iter_{max}} \tag{3}$$

Where $iter_{max}$ is the maximum of iteration in evolution process, W_{max} is maximum value of inertia weight, W_{min} is the minimum value of inertia weight, and iter is current value of iteration. The objective function represents the function that measures the performance of the system. The fitness function (objective) function for PSO is defined as the Integral of Time multiplied by the Absolute value of Error (ITAE) of the corresponding system. Therefore, it becomes an unconstrained optimization problem to find a set of decision variables by minimizing the objective function. The algorithm of PSO based PID controller is represented as flow chart in fig.2, and its sequence of operations are as follows

1. Initialize the algorithm parameters like number of generation, population, inertia weight and constants.
2. Initialize the values of the parameters K_P, K_i and K_D randomly.
3. Calculate the fitness function of each particle in each generation.
4. Calculate the local best of each particle and the global best of the particles.
5. Update the position, velocity, local best and global best in each generation.
6. Repeat the steps 3 to 5 until the maximum iteration reached or the best solution is found.

Evolutionary Algorithm spans a family of optimization algorithms, differing such as the way of selecting the best, how to create new solutions from existing ones, and the data structures used to represent those solutions. Although they correspond to the same basic algorithmic skeleton called Evolution Strategies, the proposed algorithm is differentiated by the way the velocity and position parameters are updated. In PSO it is the inertia weight that is used to balance the global and local search ability. By changing the inertia weight dynamically, the search ability is dynamically adjusted. By linearly decreasing the inertia weight from a relatively large value to a small value, PSO tends to have more global search ability [15]. This paper introduces new variants called Constriction Factor, Acceleration Factor and Time Varying Acceleration Co-efficient to enable the PSO in providing best solution with increased computational efficiency.

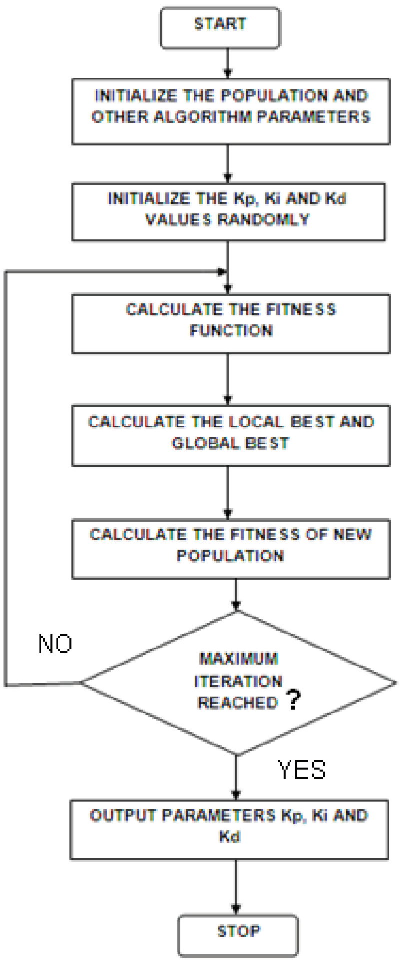

Fig.2. PSO Algorithm for PID Controller

3.2 ENHANCED PARTICLE SWARM OPTIMIZATION (EPSO)

The EPSO is a kind of swarm intelligence, being inspired by the study of birds and fish flocking. EPSO is an improved version of the Conventional PSO. In EPSO, the Constriction Factor approach is introduced in the velocity update formula to ensure faster convergence. And the Inertia weight of the particles is made to decrease linearly to avoid local best solution. The velocity and position of the particle is updated using the Eq. (4) and (5) respectively.

$$V_{id} = WKV_{id} + C_1 R (P_{id} - X_{id}) + C_2 r (P_{gd} - X_{id}) \quad (4)$$

$$X_{id} = X_{id} + V_{id} \quad (5)$$

where, Constriction Factor (K) is evaluated from the given values of C_1 and C_2 by the Eqn (6).

$$K = \frac{2}{2 - c - \sqrt{c^2 - 4c}} \quad (6)$$

Where, $c = C_1 + C_2,\ c > 4$

where C_1 and C_2 in Eqn.6 are the cognitive and social co-efficients of the particles in the search space. As a result, the proposed EPSO algorithm provides stable and faster convergence towards global best solution in a minimal computational time [16]. The fitness function is possibly the most important component of an EA. The purpose of the fitness function is to map a chromosome representation into a scalar value. Since, each chromosome represents a potential solution, the evaluation of the fitness function quantifies the quality of chromosome, i.e., how close the solution is to the optimal solution. Selection, cross-over, mutation and elitism operators usually make use of the fitness evaluation of chromosomes [17]. Also, the probability of an individual to be mutated can be a function of its fitness. It is therefore extremely important that the fitness function accurately models the optimization problem. The fitness of each particle is evaluated using the ITAE fitness function as in Eq. (7). The Integral of Time is multiplied by the Absolute value of Error (ITAE) [16] [18].

$$F = \int_0^\infty t.e(t).dt \quad (7)$$

3.3 MULTI OBJECTIVE PARTICLE SWARM OPTIMIZATION (MOPSO)

Real world problems often have multiple conflicting objectives. In certain problems, there is no single solution that is the best when measured on all objectives. These problems give raise to the multi-objective optimization. In this method, the Acceleration Factor (K) is introduced in the velocity update formula. And the inertia weight of the each particle is made to decrease linearly in all iterations and by adjusting the different objectives, the MOPSO seeks to discover what possible combinations of these objectives are available and then the best solutions can be found for the PID controller [19]. In this algorithm original PSO which operates in continuous space is extended to operate on discrete binary variables. The extended version of PSO has been proven to be very effective for static and dynamic optimization problems. The multi objective particle swarm optimization technique is based on the idea of combining several objective functions that are need to be satisfied by solution. In this algorithm t denotes the generation index, p_t the population and A_t the archive at generation. The Acceleration Factor (K) is evaluated from the Eq. (8)

$$\alpha = \alpha_0 + t / T \quad (8)$$

where t denotes current generation and T denotes the total number of generation. After evaluating population p_t, initial archive A_t is generated with non-dominated solutions in p_t The weight of the particle is linearly decreased with each iteration according to the Eqn.9.

$$W = w_0 + r^*(1 - w_0) \quad (9)$$

where w_0 is the initial weight and r is a random number [0,1] weight varies randomly[20-sabahi]

The velocity and position of the particle is updated using the Eq. (10) and (11).

DOI: 10.21917/ijsc.2010.0014

$$V^i_{j,t+1} = wV^i_{j,t} + \alpha * [c_1R_1(p^i_{j,t} - x^i_{j,t}) + c_2R_2(p^{i,g}_{j,t} - x^i_{j,t})] \quad (10)$$

$$X^i_{j,t+1} = x^i_{j,t} + V^i_{j,t+1} \quad (11)$$

The Local and Global best positions are updated after each iteration based on the fitness values of particles. The fitness value is calculated considering both objectives using the relation given in Eq. (12) [20].

$$\text{Eval}(k) = \sum_{i=1}^{n} w_i f_i(k) \quad (12)$$

3.4 STOCHASTIC PARTICLE SWARM OPTIMIZATION (SPSO)

In this method, the 'Time Varying Acceleration Co-efficient' (TVACs) are introduced for Cognitive and Social co-efficient. The implementation these TVACs reduce the cognitive component (c1) meanwhile; it increases the social component (c2) acceleration coefficient with time. Here, the inertia weight and acceleration coefficients are neither set to a constant value nor set as a linearly decreasing time varying function. Instead, these values are updated non-linearly in each generation and so the better convergence rate is obtained towards the optimal PID gains in minimal iterations [21]. The Time Varying Acceleration Co-efficient (TVAC) i.e. Cognitive and Social Co-efficients are initialized as in Eq. (13) and (14).

$$c_1 = (c_{1i} - c_{1f}) * \left(\frac{max\_iter - iter}{max\_iter}\right) + c_{1f} \quad (13)$$

$$c_2 = (c_{2i} - c_{2f}) * \left(\frac{max\_iter - iter}{max\_iter}\right) + c_{2f} \quad (14)$$

where, Initial Cognitive factor $c_{1i} = 2.05$, Initial Social factor $c_{2i} = 2.05$, Final Cognitive factor $c_{1f} = 3$ and Final Social factor $c_{2f} = 3$. Now the Weight of the each particle is updated non-linearly by using the formula in Eq. (15).

$$w = (w_{max} - w_{min}) * \left(\frac{max\_iter - iter}{max\_iter}\right) + w_{min} \quad (15)$$

Where W_{max} is the maximum inertia weight and W_{min} is the minimum value of inertia weight. *max_iter* is maximum number of iteration that algorithm can evolve each particle and *iter* is the current iteration value. Fitness function is applied for each particle and it is evaluated in each iteration for updating the particles towards the best solution in every step. The Fitness function used in this algorithm is Integral Time Absolute Error (ITAE) function given in Eq. (16) i.e., the Integral of Time is multiplied by the Absolute value of Error [22].

$$F = \int_0^\infty t.e(t)dt \quad (16)$$

4. DESIGN OF EA BASED PID CONTROLLER

In Conventional PID controller, the gains are randomly selected by trial and error method. In this paper, EA finds the Proportional, Integral and Derivative gains of the PID controller and the values are passed to the PID controller of the plant. The block diagram in Fig.3 shows the EA based PID controller.

The Simulink model for LFC and AVR with PID controller is designed based on the transfer function approach with the different values for each constant. For AVR model K_a=10, K_e=1, K_g=1, K_r=1, T_a=0.1, T_e=0.4, T_g=1, T_r=0.05. For LFC model T_g=0.2, T_t=0.5,H=10,D=0.8 [23].This model depicts a plant which encloses an AVR and LFC loop within it and the PID controller getting a step input and the regulated output is seen from the scope. The Simulink model of LFC and AVR is shown in Fig.4 and Fig.5 respectively. The AVR model consists of a step input, EPSO based PID controller, an amplifier that amplifies the signal to the exciter which in turn controls the voltage of the generator and a scope to display the terminal voltage. It also contains a sensor that determines the difference between load demand and power generated and feeds it to the controller based on the load changes. The LFC model in Fig.4 shows a step input, PID controller based on EPSO, a governor that controls the speed of the turbine that drives the generator and the scope that shows the frequency deviation.

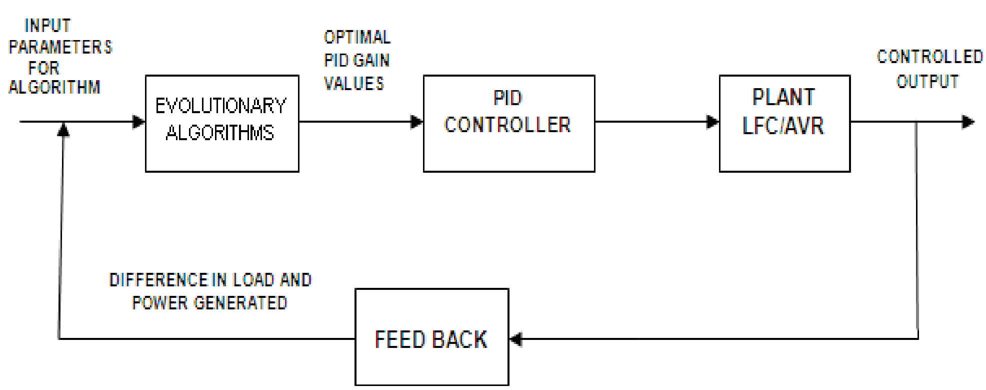

Fig.3. EA Based PID Controller

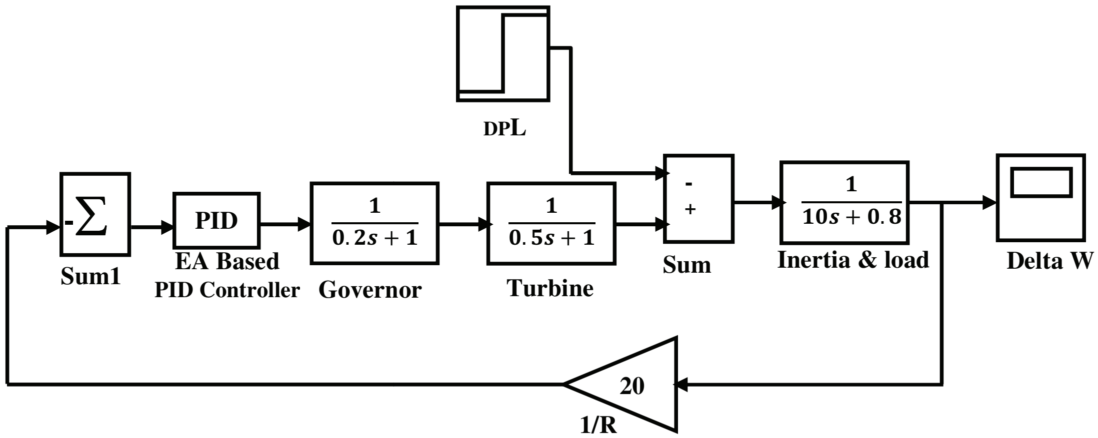

Fig.4. Simulink Model of Load Frequency Control with EA Based PID Controller

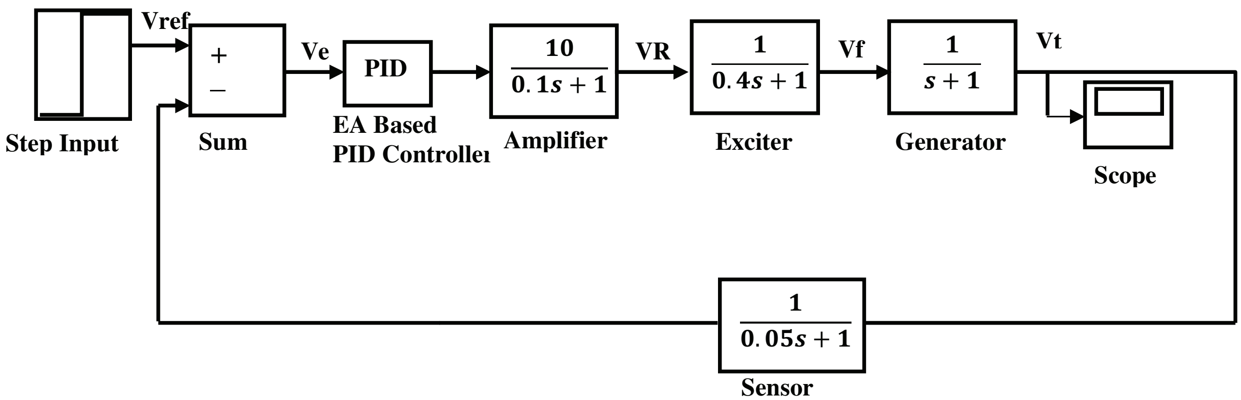

Fig.5 Simulink Model of Automatic Voltage Regulator with EA Based PID Controller

5. SIMULATION RESULTS

The simulation results for Load Frequency Control (LFC) and Automatic Voltage Regulator (AVR) are given for a single area system to quantify the benefits of EPSO, MO-PSO and SPSO based PID controller. The simulation was done using the Simulink package available in MATLAB R2008b. The LFC and AVR were simulated on Intel core 2 Duo (2.4 GHz), 3GB RAM PC. Simulink model for LFC and AVR with EA based PID controller is constructed based on the generalized model of the turbo alternator. The K_p, K_I and K_d values for the PID controller is obtained by running the M-file that calls the fitness function to evaluate the fitness of the solution. The simulation was performed for different load and regulations and the results are discussed in this section.

5.1 EPSO BASED PID CONTROLLER

The simulation results for AVR and LFC with EPSO based controllers are presented in Fig.6 and Fig.7 to validate the efficiency of the proposed algorithm.

The algorithm is simulated by keeping the population size and number of generations as 50. The inertia weight minimum is kept at 0.4 and the maximum inertia weight at 0.9. The cognitive and social co-efficient are maintained at 2.05 and 2 respectively. The step load change (ΔP_L) of 20 % (0.2 p.u) disturbance is considered for the single area power system. It can be found that EPSO generates the relatively better results with faster convergence rate and higher precision. The time taken for the computation of the PID gains using this algorithm is 10.2 seconds. It is observed that the settling time of AVR with EPSO based PID controller is 4.9 seconds and there is no transient peak overshoot. Also the settling time of LFC is 9.5 seconds and the peak overshoot is -0.0093, for the same disturbance and at speed regulation value of 20.

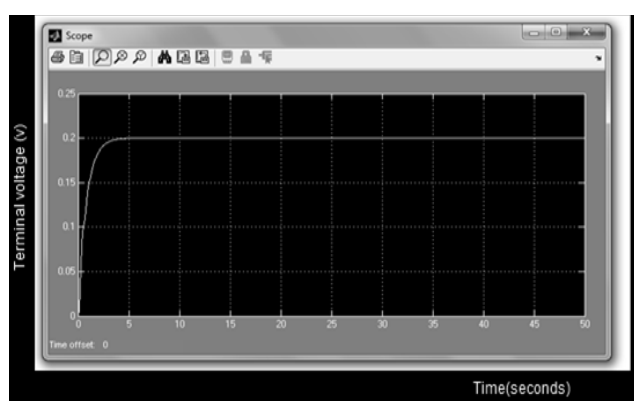

Fig.6. AVR with EPSO Based PID Controller for ΔP_L = 0.2 p.u.

DOI: 10.21917/ijsc.2010.0014

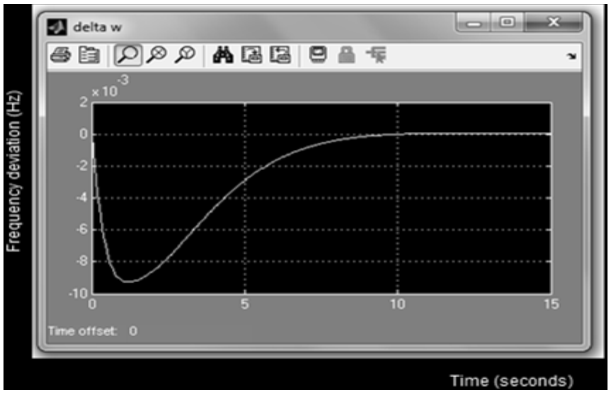

Fig.7. LFC with EPSO Based PID Controller for R=20 and ΔP_L = 0.2 p.u.

When compared to conventional PSO-PID controller the settling time of AVR is reduced by 88% and no overshoot and the oscillation occurred is due to load fluctuations. The results also indicate that the proposed controller could create very perfect step response of the terminal voltage in AVR system. The settling time, peak overshoot and oscillations of LFC is reduced by 83%, 33% and 33%, respectively.

5.2 MO-PSO BASED PID CONTROLLER

The K_p, K_I and K_d values for the PID controller is obtained by running the MO-PSO code developed as an M-file in MATLAB R2008b. The optimal parameter values for population and number of generations is maintained at 50 and 25 for both LFC and AVR. The cognitive and social co-efficient are maintained at 2.05 and 3 respectively. The PID gain values are transferred to the AVR and LFC Simulink model for simuating with different load and regulation values. In this technique the objective functions are collectively minimized by means of assigning weight for different objective functions. The computational time for the particle convergence to the optimum values of PID gains in MO-PSO is 14.35 seconds. In this technique the PID gains are obtained by giving priority to the objective function that needs to be satisfied. The terminal voltage response and frequency deviation of turbo generator for a change in load of 0.2 p.u is shown in Fig.8 and Fig.9.

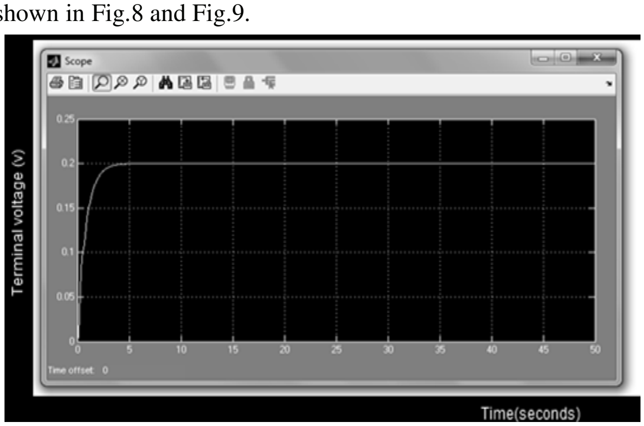

Fig.8. AVR with MO-PSO Based PID Controller for ΔP_L=0.2p.u

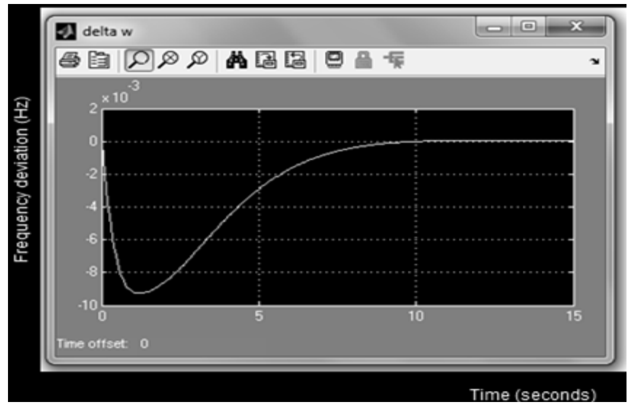

Fig.9. LFC with MO-PSO Based PID

It is observed form the graph that the settling time of AVR with MO-PSO based PID controller is 5.0 seconds and there is no transient peak overshoot. Also the settling time of LFC is 9.7 seconds and the peak overshoot is -0.0091, which is much less than the conventional PSO-PID controllers. When compared to conventional controller it is observed that the settling time, peak overshoot and oscillations of LFC is reduced by 81%, 34% and 34%, respectively. The settling time of AVR is reduced by 87% as compared to the conventional controller.

5.3. SPSO BASED PID CONTROLLER

The standard PSO algorithm determined by non-negative real parameter tuple {w, c1, c2} is analyzed using stochastic process theory. The stochastic convergence condition of the particle swarm system and corresponding parameter selection guidelines are derived. The K_p, K_I and K_d value for the PID controller is obtained by running the SPSO source code as an M-file. The optimal parameter values for population size and number of iterations are maintained 50 and 25 respectively. The Inertia weight is linearly varied between 0.35 and 0.4. The minimum and maximum values for C1 and C2 are selected between the ranges 2 and 3. The LFC and AVR models are simulated for different regulations and loads are analyzed. The frequency deviation and terminal voltage response for a change in load of 0.2 p.u and regulation of 20 is shown in Fig.10 and Fig.11.

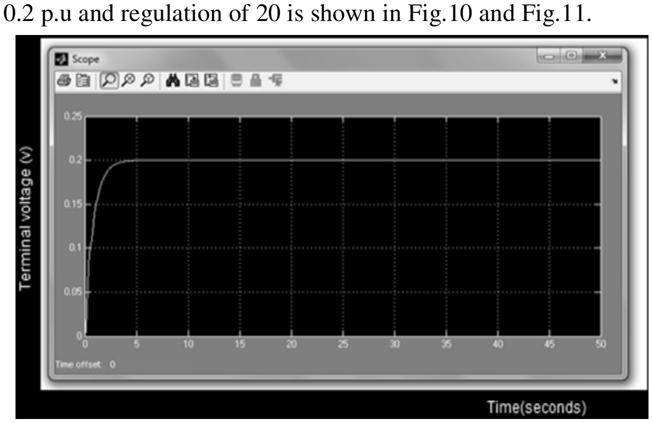

Fig.10. AVR with SPSO Based PID Controller for ΔP_L=0.2 p.u

ISSN: 2229 – 6956(ONLINE)
DOI: 10.21917/ijsc.2010.0014

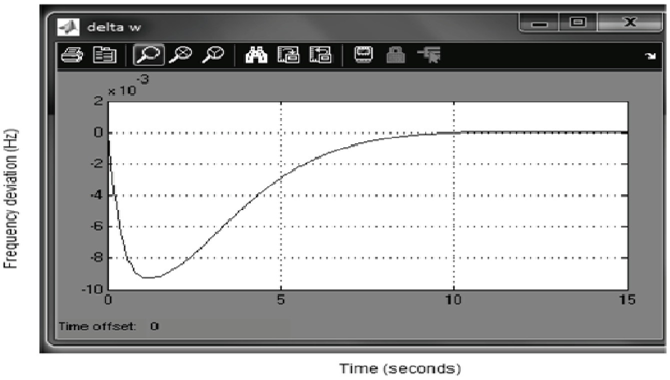

Fig.11. LFC with SPSO Based PID Controller for R=20 and ΔP_L = 0.2 p.u.

From Fig.10, it is found that the settling time of AVR with SPSO based Integral controller is 5.1 seconds and there is a no transient overshoot. The settling time for frequency deviation is 9.8 seconds and the output response varies between 0 to -0.0097 which is very less as compared to PSO-PID controller.

6. COMPARATIVE ANALYSIS

When an electrical load change occurs, the turbine-generator rotor accelerates or decelerates, and frequency undergoes a transient disturbance. The controller should not allow transient oscillations or overshoot, which in-turn trips the under-frequency relay connected in the system. Oscillations, settling time and overshoot are interrelated changes in one parameter will affect the other parameter. Hence, it is important that the designed controller must be efficient in selecting the optimum gains in order to achieve better results. Owing to the randomness of the heuristic algorithms, their performance cannot be judged by the single result; hence the models are simulated for different load changes and regulations to validate the efficiency of the proposed algorithms [24]. The value of R determines the slope of the governor characteristics and it determines the change on the output for a given change in frequency. In practice 'R' is set on each generating unit so that change in load on a system will be compensated by generated output. The speed governor system should be operated within the restricted control range of feedback gains due to the system instability. Therefore higher value of load ΔPL for a small 'R' value will introduce oscillations into the system. Hence ΔPL and R are selected as shown in table 2 & 3 to obtain optimum results in terms of settling time, overshoot and oscillations. Increasing the load ΔPL into higher values will experience large overshoot and settling time. The performance of the proposed EA based controllers developed for AVR of power generating system for various load changes is given in table 1.The results in table 2 and table 3 indicate the efficiency of PSO algorithm for real-time applications and its suitability under varying load conditions. The EA provides better convergence and reveal their superiority with respect to settling time, oscillations and overshoot when compared to the standard PSO method.

Table.1. Performance analysis of EA based AVR

Change in load (ΔP_L)	**Settling Time in Seconds**			
	PSO	**EPSO**	**MO-PSO**	**SPSO**
0.1	9.03	4.5	4.8	4.7
0.2	11.2	4.9	5.0	5.1
0.6	12.9	5.2	5.3	5.4
0.8	14.6	5.5	5.7	5.7

The simulation results of EPSO based controller shows that for a load of 0.2 p.u and regulation of 20 the settling time of LFC is reduced by 83%, the oscillations are decreased by 33%, reduction of 33% in the overshoot and the settling time of AVR is reduced by a factor of 88% when compared to the conventional controllers. The time taken for the algorithm to converge to the optimal gain values is 10.2 seconds. The results of MO-PSO based controller shows that for a load of 0.2 p.u and regulation of 20 the settling time of LFC is reduced by a factor of 81%, the oscillations are decreased by 34%, reduction of 34% in overshoot and the settling time of AVR is reduced by 87% as compared to conventional controllers. The time taken to find the optimal gains by the algorithm is 14.4 seconds. The analysis results of SPSO based controller shows that for a load of 0.2 p.u and regulation of 20 the settling time of LFC is reduced by 80%, the oscillations are decreased by 30%, reduction of 30% in the overshoot and the settling time of AVR is reduced by a factor of 86% when compared to the conventional controllers. The time required for the algorithm to compute the values of PID gains is 18.5 seconds.

Table.2. Performance comparison of EA based LFC for R value of 20

Parameter	**R1=20**							
	ΔP_L=0.2				ΔP_L=0.6			
	PSO	EPSO	MO-PSO	SPSO	PSO	EPSO	MO-PSO	SPSO
Settling Time (s)	10.4	9.1	9.7	9.9	13.4	11.6	11.8	11.9
Overshoot (Hz)	-0.0102	-0.0093	-0.0091	-0.035	-0.042	-0.028	-0.031	-0.035
Oscillation (Hz)	0 to 0.0014	0 to 0.0093	0 to 0.0091	0 to 0.035	0 to 0.042	0 to 0.028	0 to 0.031	0 to 0.035

Table.3. Performance comparison of EA based LFC for R value of 30

Parameter	R1=30							
	ΔP_L=0.2				ΔP_L=0.6			
	PSO	EPSO	MO-PSO	SPSO	PSO	EPSO	MO-PSO	SPSO
Settling Time (s)	12.2	11.5	11.8	11.9	13.4	12.5	12.8	12.8
Overshoot (Hz)	-0.0102	-0.0066	-0.0068	-0.0071	-0.042	-0.020	-0.068	-0.027
Oscillation (Hz)	0 to 0.0102	0 to 0.0066	0 to 0.0068	0 to 0.007	0 to 0.042	0 to 0.020	0 to 0.068	0 to 0.027

7. COMPUTATIONAL EFFICIENCY OF EVOLUTIONARY ALGORITHMS

The time complexity of different evolutionary algorithms used in the optimization of PID gains are analyzed. The mean CPU time taken to complete the fixed number of iterations has been analyzed. The comparison of average computation time or time complexity of different EAs for combinatorial optimization of PID gains for AVR and LFC is shown in Fig. 12.

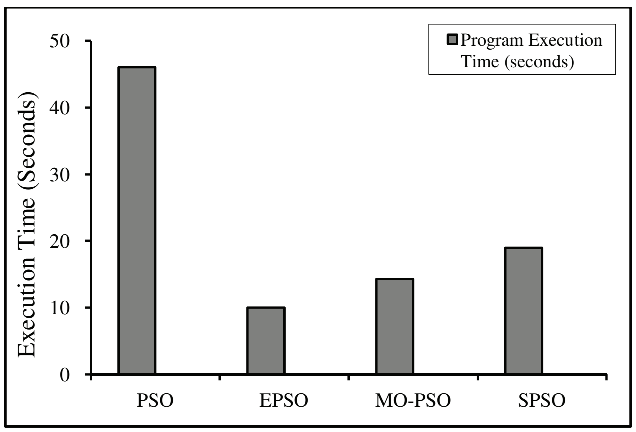

Fig.12 Comparative Analysis of Execution Time for Different Evolutionary Algorithms

The execution time is measured for different evolutionary algorithm with number of iterations as 25 and swarm size as 50. As shown in Fig.12 the EPSO algorithm takes a computational time of 10.1 seconds which is less when compared to MO-PSO and SPSO. These entire algorithms find best solution with less number of iterations and there is marginal difference in time taken to converge into the best solution. For higher values of iterations and swarm size, the computational efficiency and the program execution time is found to be increased. EA uses probabilistic transition rules to move in the search space [25]. Also EA uses a parallel search through the search space; this increases the computational efficiency of the algorithms. The no-free-lunch (NFL) theorem states that there cannot exist any algorithm for solving all problems that is on average superior to any other algorithm. This theorem motivates research in new optimization algorithms, especially EC. The basic PSO method does not perform the selection and crossover operation in evolutionary process, it can save computation time compared with the GA method, thus proving that the EA based PID controller is more superior. The EA search starts from a diverse set of initial points, which allows parallel search of a large area of the search space.

8. CONCLUSION

In this paper, the EA based PID controllers are utilized for the LFC and AVR of power generating system and compared with conventional controllers. The conventional controllers used for this application exhibits poor dynamic characteristics with large settling time, oscillations and overshoot. Hence, an intelligent technique has been proposed for voltage and frequency control in an isolated power system, which is found to be more suitable for controlling in relatively less time. The LFC and AVR models with EPSO, MO-PSO and SPSO based controllers were simulated for different load changes and regulations to validate the efficiency of the proposed algorithms. From the simulation results it can be found that EA based controllers can produce relatively better results with faster convergence rate and higher precision. The proposed algorithm attempts to make a judicious use of exploration and exploitation abilities of the search space and therefore likely to avoid false and premature convergence. The cost of power generation is exorbitant, hence quality and reliable power supply is utmost important for a generating system. Hence application of these evolutionary algorithms will lead to the satisfactory performance of the power generating system.

REFERENCES

[1] Hadi Saadat, 2002, "*Power System Analysis*", McGraw-Hill, 2nd Edition. New Delhi.
[2] P. Kundur, 2006, "*Power System Stability and Control*", Tata McGraw-Hill, 2nd Edition. New Delhi.
[3] J.Wood, F.Wollenberg, 2003, "*Power generation, Operation and Control*", John Wiley & Sons (ASIA) Pte Ltd, 2nd Edition. New York.
[4] D.M.Vinod Kumar, 1998, "Intelligent Controllers for Automatic Generation Control", *Proc. of IEEE region 10 International conference on global connectivity in Energy, Computer, Communication and Control*, Vol.2, pp.557-574.
[5] C.W.de silva, 1995, "Intellligent Control: Fuzzy logic applications", CRC press Inc.
[6] P.S.R. Murthy, 2009, "Power System Operation and Control", BSP books Pvt. Ltd.
[7] M.A. Panduro, C.A. Brizuela, L.I. Balderas and D.A. Costa, 2009, "A comparision of Genetic algorithm, Particel Swarm

Optimization and the Differential Evolution methods for the design of Scannable circular antenna arrays", *Progress in Electromagnetic Research,* Vol.13, pp.171-186.

[8] Kennedy.J and Eberhart, R.C., 1995, "Particle Swarm Optimization", *Proceedings of the 1995 IEEE International conference on Neural Networks*, Perth, Australia, IEEE Service Center, Piscataway, NJ, IV:1942-1948.

[9] Zwe-Lee Gaing, 2004, "A particle Swarm Optimization approach for optimum design of PID controller in AVR system", *IEEE Transactions on Energy Conversion*, Vol.19, No.2, pp.384-391.

[10] You-Bo Wang, Xin Peng, and Ben-Zheng Wei, 2008, "A New Particle Swarm Optimization Based Auto-Tuning of PID Controller", *Proceedings of the Seventh International Conference on Machine Learning and Cybernetics*,Kunming, pp. 1818 – 1823.

[11] Haluk Gozde et al, 2008, "PSO based Load Frequency Control in a single area power system", *University of Pitesti, Scientific Bulletin*, Vol.2, No.8, pp.106-110.

[12] D.B. Fogel, 2000, "*Evolutionary Computation: Toward a new philosophy of machine intelligence*", 2^{nd} Edition. New York: IEEE Press.

[13] J. Kennedy, R.C. Eberhart, 1995, "Particle Swarm Optimization", *Proceedings of IEEE conference on neural networks*, Vol. 4, pp.1942-1948.

[14] Shi, Y., and Eberhart, R. C., 1999, "Empirical Study of Particle Swarm Optimization," IEEE Congress on Evolutionary Computation, Piscataway, NewJersey, Vol. 3, pp. 1945-1950.

[15] T.Back, U.Hammel, and H.-P.Schwefel, 1997, "Evolutionary Computation: Comments on the History and Current state", *IEEE Transactions on Evolutionary Computation*, Vol. 1, No.1, pp.3-17.

[16] Hongqing Fang, and Long Chen, 2009, "Application of an enhanced PSO algorithm to optimal tuning of PID gains", *Proceedings of the Chinese control and Decision Conference*.

[17] R. C. Eberhart and Y. Shi, 2000, "Comparing inertia weights and constriction factors in particle swarm optimization", in *Proc. of the 2000 Congr. Evol. Comput.*, San Diego, pp. 84-88.

[18] Shigenori Naka, Takamu Genji, Kenji Miyazato, and Yoshikazu Fukuyama, 2004, "Hybrid Particle Swarm Optimization using Constriction Factor Approach", *IEEE Transactions on Energy Conversion*, Vol. 19, No. 2.

[19] Andrey Popov, Adel Farag and Herbert Werner, 2005, "Tuning of a PID controller using Multi-objective optimization technique", *Proceedings of the 44^{th} IEEE International Conference on Decision and control, Seville, Spain*.

[20] K. Sabahi, A. Sharifi, M. Aliyari, M. Teshnehlab and M. Aliasghary, 2008, "Load Frequency Control in Interconnected Power system using Multi-objective PID Controller", *Journal of Applied Sciences*, Vol.8, No.20, pp.3676-3682.

[21] M. Janga Reddy and D. Nagesh Kumar, 2007, "An efficient multi-objective optimization algorithm based on swarm intelligence for engineering design", *Engineering Optimization,* Vol.39, No.1, pp. 49-68.

[22] G. Ramana Moorthy, M. Senthil Arumugam and C.K. Loo, 2009, "Hybrid particle swarm optimization with fine tuning parameters", *International Journal of Bio-inspired computations,* Vol.1, No 1/2.

[23] Y.L. Karnavas, D.P. Papadopoulos, 2002, "AGC for autonomous power system using combined intelligent techniques", *Electric Power Systems Research*, Vol.62, pp.225-239.

[24] Liang Qin, A. Suganthan and Baskar. S., 2006, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions", IEEE Trans. on Evolutionary Computation, Vol.10, No.3, pp.281-295.

[25] Zeng J., Jie J. and Hu J. , 2006, "Adaptive particle swarm optimisation guided by acceleration information", International Conference on Computational Intelligence and Security, Vol. 1, pp.351–355.