isrjournals

International Journal of Advanced Research in Electrical and Electronics Engineering

Volume: 3 Issue: 2 Feb,2016,ISSN_NO: 2321-4775

PSO and DE Based Tuning of PID Controller in AVR System

G.E.N.GANESH
Electrical and Electronics Engineering
A.U. College of Engineering (A)
Visakhapatnam, India
eswaranagaganesh@gmail.com

Dr. K.VAISAKH, Ph.D (IISc)
Electrical and Electronics Engineering
A.U. College of Engineering (A)
Visakhapatnam, India
vaisakh_k@yahoo.co.in

Abstract— **In this paper, determination of optimal parameters of a PID controller in an Automatic Voltage Regulation (AVR) system by the approach of Particle Swarm Optimization (PSO) and Differential Evolution (DE) Techniques is presented. This paper demonstrated in detail how to employ the PSO and DE methods to search efficiently the optimal PID controller parameters of an AVR system. A MATLAB simulation has been performed. The proposed approach had superior features, including easy implementation, stable convergence characteristic, and good computational efficiency. In order to assist estimating the performance of the PID controller using PSO and DE, Minimum Error Integral (MEI) criterion Integral Absolute Error (IAE), Integral Square Error (ISE), Integral Time Absolute Error (ITAE) and Integral Time Square Error (ITSE) are used.**

Keywords*— *AVR system, Differential Evolution, Particle Swarm Optimization, PID controller, tuning of controller.

I. INTRODUCTION

The Automatic Voltage Regulator (AVR) is widely used in electrical power field to obtain the stability and good regulation of the electric system. The characteristics of alternator output required are constant voltage and constant current. To get the constant output, alternator field excitation is controlled by Automatic Voltage Regulator (AVR). The role of an (AVR) is to hold the terminal voltage magnitude of a synchronous generator at a specified level. Constant voltage at the generator terminals is essential for satisfactory main power supply. The terminal voltage can be affected by various disturbing factors (speed, load, power factor, and temperature rise), so that special regulating equipment is required to keep the voltage constant, even when affected by these disturbing factors. So to maintain the constant terminal voltage some controller is required to get the desired step response. There are so many controller used such as adaptive control, neural control, recently fuzzy control. Among them PID controller is mostly used in industrial applications because of its simple structure and robust performance in much number of operating conditions. But tuning of PID control parameters (controller gains) little difficult. Because many industrial applications deals with high order, nonlinearities. Several optimization techniques are proposed over the years for tuning of PID controller. The first and most popular method is Ziegler-Nichols method (closed loop method). But by using this method it is quite difficult to find the efficient values of controller gains in desired systems. For these reasons it is highly desirable to adapt a searching technique to increase the capability of PID controller. There several random search methods and heuristic methods increasing attention for efficient values and optimized solution. Eberhart and Kennedy developed particle swarm optimization (PSO) based on the analogy of swarms of birds and fish schooling. Each individual exchanges previous experiences in PSO. These research efforts are called swarm intelligence. Other evolutionary computation (EC) techniques such as genetic algorithms (GAs), utilize multiple searching points in the solution space like PSO. Whereas GAs can treat combinatorial optimization problems, PSO was aimed to treat nonlinear optimization problems with continuous variables originally. Moreover, PSO has been expanded to handle combinatorial optimization problems and both discrete and continuous variables as well. Moreover, unlike other EC techniques, PSO can be realized with only a small program.

As a relatively new population-based optimization technique, differential evolution has been attracting increasing attention for a wide variety of engineering applications including power engineering. Differential evolution is an efficient heuristic algorithm for search and optimization. DE operates on floating point representation of variables to be optimized. Like other evolutionary algorithms, DE is capable of handling non convex, non differentiable complex optimization problems. The main advantages of a DE come from its simple but effective mutation process to ensure the search diversity as well as to enhance the search effectiveness with the information from the objective function directly.

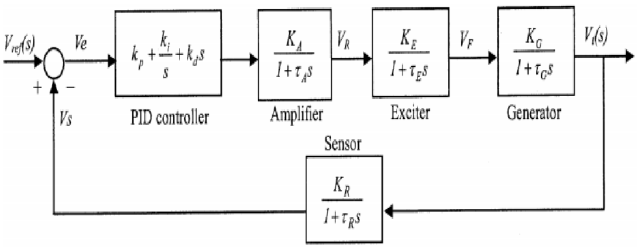

Fig. 1 Block diagram of AVR system with PID controller

II. AVR SYSTEM WITH PID

A. AVR System

The AVR maintains the constant voltage up to certain level of the load current which is independent of the generator speed and load. A simple AVR system comprises four main components, namely amplifier, exciter, generator, and sensor. For mathematical modelling and transfer function of the four components, these components must be linearized, which takes into account the major time constant and ignores the saturation or other nonlinearities. The reasonable transfer function of these components may be represented, respectively, as follows

- Amplifier Model:

The excitation system amplifier may be a magnetic amplifier, rotating amplifier, or modern electronic amplifier. The amplifier is represented by a gain "K_A" and a time constant "τ_A", and the transfer function is Typical values of K_A are in the range of 10 to 400. The amplifier time constant is very small, in the range of 0.02 to 0.1 second, and often is neglected. The transfer function is

$$\frac{V_R(s)}{V_e(s)} = \frac{K_A}{1+\tau_A s}. \qquad (1)$$

- Exciter model:

A reasonable model of a modern exciter is a linearized model, which takes into account the major time constant and ignores the saturation or other nonlinearities. In the simplest form, the transfer function of a modern exciter may be represented by a single time constant "τ_E" and a gain"K_E", i.e.

$$\frac{V_F(s)}{V_R(s)} = \frac{K_E}{1+\tau_E s}. \qquad (2)$$

Typical values of are in the range of 10 to 400. The time constant is in the range of 0.5 to 1.0 s.

- Generator model:

The synchronous machine generated e.m.f is a function of the machine magnetization curve, and its terminal voltage is dependent on the generator load. In the linearized model, the transfer function relating the generator terminal voltage to its field voltage can be represented by a gain "K_G" and a time constant "τ_G" and the transfer function is

$$\frac{V_t(s)}{V_F(s)} = \frac{K_G}{1+\tau_G s}. \qquad (3)$$

These constants are load-dependent, "K_G" may vary between 0.7 to 1, and "τ_G" between 1.0 and 2.0 seconds from full-load to no-load.

- Sensor Model:

The voltage is sensed through a potential transformer and, in one form, it is rectified through a bridge rectifier. The sensor is modelled by a simple first order transfer function, given by

$$\frac{V_s(s)}{V_t(s)} = \frac{K_R}{1+\tau_R s}. \qquad (4)$$

"τ_S" is very small, and we may assume a range of 0.01 to 0.06 second.

B.PID Controller

PID controller is used to increase the dynamic response of the system. It operates with 3 combined controllers P, I, and D. A proportional controller has the effect of reducing the rise time, but never eliminates the steady-state error. An integral controller has the effect of eliminating the steady-state error by adding finite pole, but it may make the transient response worse. A derivative controller has the effect of increasing the stability of the system, reducing the overshoot, and improving the transient response by adding finite zero. The transfer function of the PID controller is

$$TF_{PID} = K_p + \frac{K_i}{S} + K_d S \qquad (5)$$

C.AVR system with PID

PID controller incorporated in AVR system is shown in fig 1.1

D. Performance index used for optimal PID

For the assist of searching efficient values of PID gains minimum error integral criteria is used and their formulae as follows;

$$\mathrm{IAE}=\int_0^\infty |e(t)|.dt \qquad (6)$$

$$\mathrm{ISE}=\int_0^\infty e^2(t).dt \qquad (7)$$

$$\mathrm{ITAE}=\int_0^\infty t\,|e(t)|\,dt \qquad (8)$$

$$\mathrm{ITSE}=\int_0^\infty t.e^2(t).dt \qquad (9)$$

The IAE, ISE, ITAE and ITSE performance criterion formulas are as follows
Conclusions for MEI tuning for disturbances:

- ISE formula result in the tightest tuning (highest gain, shortest integral time)
- ITAE results in the loosest tuning
- IAE results in intermediate tuning
- ITSE would probably fall between IAE and ISE in tightness of tuning.

A set of good control parameters K_p, K_i, and K_d can yield a good step response that will result in performance criteria minimization in the time domain.

III. PSO BASICS AND ALGORITHM

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behaviour of bird flocking or fish schooling. Like other evolutionary algorithms, PSO is also a population-based search algorithm and stats with an initial population of randomly generated solutions called particles which fly through the search space. Each particle represents a candidate solution to the optimization problem, and has a velocity and a position. The position of a particle is influenced by the best position visited by itself i.e. its own experience and the position of the best particle in its neighborhood i.e. the experience of neighboring particles. The best particle in the population is denoted by (global best), while the best position that has been visited by the current particle is denoted by (local best). Consequentially, each particle is influenced by the best performance of any member in the entire population due to the sharing information between them. The performance of each particle is measured using a fitness function that varies depending on the optimization problem. Each particle in the swarm is represented by the following characteristics:

X_i: The current position of the particle i.
V_i: The current velocity of the particle i.
P_i: The best position of particle i so far, and is the best position found in the whole swarm so far. Below equations are used for updating both of the velocity and the position of each particle.

$$V_i = w_i.V_i + c_1.r_1.(P_i - X_i) + c_2.r_2.(P_g - X_i) \qquad (10)$$

$$X_i = X_i + V_i \qquad (11)$$

Where: C_1 and C_2 are the cognitive coefficients and r_1, and r_2 are random real numbers drawn from [0,1)], ω is the inertia weight which is used to achieve a balance in the exploration and exploitation of the search space and plays very important role in PSO convergence behaviour. The inertia dynamically reduces during a run from 1.0 to near 0 in each generation which facilitates a balance in the exploration and exploitation of the search space, it is determined as follows:

$$w_i = w_{max} - \frac{w_{max} - w_{min}}{iter_{max}}.iter \qquad (12)$$

Where $iter_{max}$, is the maximum number of iterations, and iter is the current number of iteration.

IV. PSO EMPLOYED TO PID

In this paper PID controller gains in AVR system are tuned by PSO technique. It is also called PSO-PID controller. By using PSO algorithm proportional, integral, derivative gains can be obtained optimally such that AVR system could obtain good step responses with the assist of minimum error integral criteria.

A. Parameters of the problem

To apply the PSO method for searching the controller parameters, we use the “individual” to replace the “particle” and the “population” to replace the “group” in this paper. We defined three controller parameters K_p, K_i, and K_d, to compose an individual K by K=[K_p K_i K_d] hence, there are three members in an individual. These members are assigned as real values. If there are *n* individuals in a population, then the dimension of a population is $n\times 3$. The matrix representation in a population is as follows.

B.PSO-PID algorithm

Step 1) Specify upper and lower limits of control parameters and initialize the parameter values randomly for each population including positions, velocities, pbest, gbest for each population
Step2) Find the performance index value
Step3) Check whether the present values with pbest and gbest values. update the next positions and velocities by the formulae given by (10) and (11) where w value is set by (12).
Step 4) Check whether these values are within bounds is not. If not modify those values by using

If $v^{t+1}>V^{max}$ then $v^{t+1}=V^{max}$
If $v^{t+1}<V^{min}$ then $v^{t+1}=V^{min}$

Step 5) modify the control parameter values at each iteration and check limits of control parameters if any violation of limits set to minimum and maximum values of respective individuals.
Step 6) the number of iterations reaches maximum then go to step 7 otherwise go to step 2.
Step 7) latest individual gbest values are the optimal control gains.
Limits of three control parameters are

$0 \le K_i \le 1$, $0 \le K_i \le 1$, and $0 \le K_i \le 1$

Without integrating the PID controller in AVR system the step response is observed.

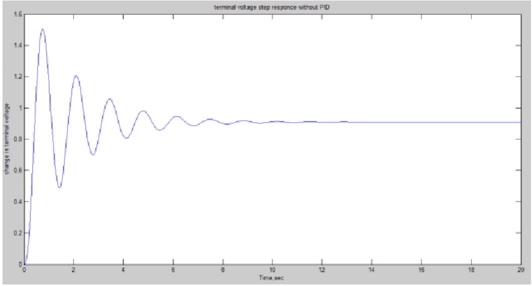

Fig.2 Step response of AVR system without PID.

From the above step response **without PID** controller the time domain parameters calculated are

- Peak overshoot $\%M_p = 65.7034$
- Peak time $t_p = 0.7479$
- Settle time $t_s = 6.9835$
- Steady state error $e_{ss} = 0.0909$

From above results we can conclude that the system is very sluggish taking more time to reach steady state and settle time is very high and system is not restoring its original state after disturbance as steady state error is not zero. Rise time is low but giving more oscillations and maximum peak overshoot is also getting high value. We use the PID controller in AVR system to increase the performance and it is tuned for desired performance. So we need to search efficient values of proportional, integral, and derivative gains for proper and acceptable step response.

Because of this unwanted response PSO-PID is integrated in AVR system.

According to the trials, the following PSO parameters are used for verifying the performance of the PSO-PID controller in searching the PID controller parameters:

- the member of each individual is k_p, k_i and k_d,
- population size = 50,
- inertia weight factor w is set by eq 3.16 where w_{max}=0.9 and w_{min} = 0.4
- the limit of change in velocity $V_{kp}^{max} = k_p^{max}/2$, $V_{ki}^{max} = k_i^{max}/2$,and $V_{kd}^{max} = k_d^{max}/2$
- Acceleration constant $C_1 = 2$ and $C_2 = 2$.

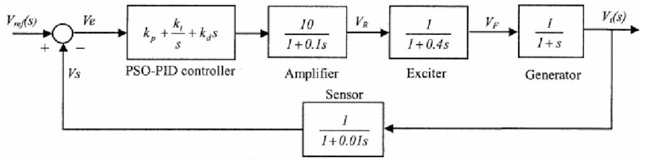

Fig.3 Block diagram of AVR with PSO-PID

The PID controller in the AVR system is tuned by using PSO with the performance criterion IAE, ISE, ITSE, ITAE. Below response is for the system in which PSO- PID controller is used. And the parameters like peak time, rise time, settle time, maximum peak overshoot, and steady state error are observed.

1) Step response with PSO-PID controller, IAE as performance criterion

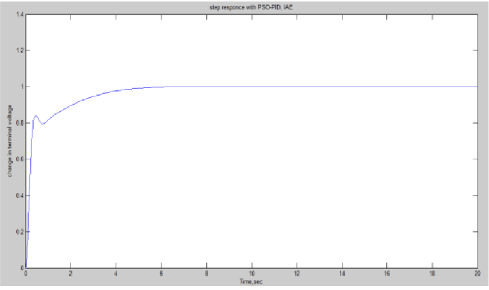

Fig.4 Step response with PSO-PID, IAE

By tuning of PSO-PID controller with the performance criterion IAE even though maximum overshoot is very less (%Mp=0.1562) and the steady state error is zero, it is taking more time to reach the steady state, as rise time t_r is 3.22s and settle time is 4.14s. So the system is very sluggish.

2) Step response with PSO-PID controller, ISE as performance criteria.

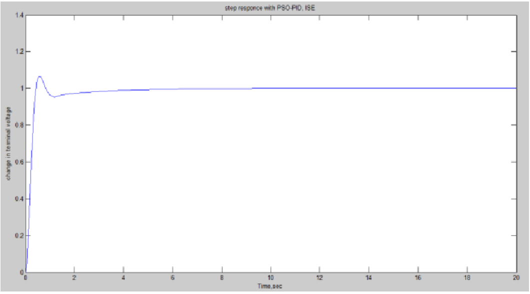

Fig.5 Step response with PSO-PID, ISE

With ISE the obtained step response has the maximum overshoot %Mp = 6.48. The rise time t_r=1.6s and settle time t_s=2.6s are less. So the response is not sluggish and not oscillatory. The steady state error is also zero.

3) Step response with PSO-PID controller, ITAE as performance criteria.

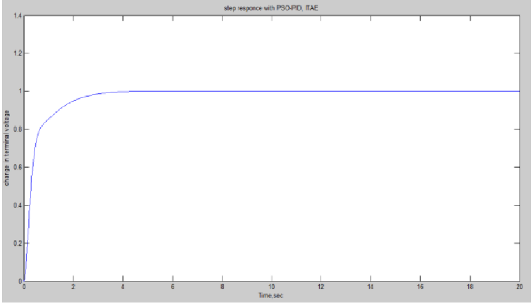

Fig.6 Step response of PSO-PID, ITAE

The step response obtained with ITAE is desired response with the maximum overshoot =0.1 rise time=3.9s and settle time is 2.7s

4) Step response with PSO-PID controller, ITSE as performance criteria.

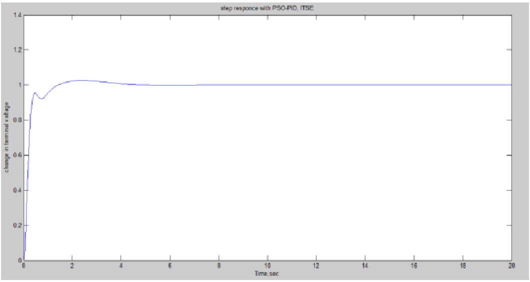

Fig.7 Step response of PSO-PID, ITSE

ITSE gives the response with very less overshoot %Mp= 2.88 and zero steady state error but with high settle time t_s = 3.07s indicating system is little slow. At the end among all criteria ITAE gives the better response for the PSO-PID in AVR system.

VI.DE BASED PID TUNING

Differential evolution was first proposed over 1994 – 1996 by Storn and Price at Berkeley as a new evolutionary algorithm (EA). Differential evolution (DE) is a stochastic direct search optimization method. It is generally considered as an accurate, reasonably fast, and robust optimization method. The main advantages of DE are its simplicity and therefore easy use in solving optimization problems requiring a minimization process with real-valued and multimodal (multiple local optima) objective functions. DE uses a non uniform crossover that makes use of child vector parameters to guide through the minimization process. The mutation operation with DE is performed by arithmetical combinations of individuals rather than perturbing the genes in individuals with small probability compared with one of the most popular EAs, genetic algorithms (GAs). Another main characteristic of DE is its ability to search with floating point representation instead of binary representation as used in many basic EAs such as GAs. The characteristics together with other factors of DE make it a fast and robust algorithm as an alternative to EA, and it has found an increasing application in a number of engineering areas including power engineering.

A. Algorithm of DE-PID:

Step 1) specify the limits of positions and velocities and initialize the random positions and random velocities by using the formulae

$$\mathbf{x}_i^G = \mathbf{x}_{i(L)} + \text{rand}_i[0,1] \cdot (\mathbf{x}_{i(H)} - \mathbf{x}_{i(L)}), \tag{13}$$

Step 2) Calculate the objective function value f(Xi) for all X_i

Step 3) start mutation Select three points from population and generate perturbed individual V_i using equation

$$v_{i,G+1} = x_{r_1,G} + F \cdot (x_{r_2,G} - x_{r_3,G}) \tag{14}$$

Step 4) Recombine the each target vector xiwith perturbed individual generated in step 3 to generate a trial vector U_i using equation

$$u_{j,i.G+1} = \begin{cases} v_{j,i.G+1} \; if \; rand_j \le C_r \vee j = k \\ x_{j,i.G} \qquad otherwise \end{cases} \tag{15}$$

Step 5) Check whether each variable of the trial vector is within range. If yes, then go to step 6 else make it within range using $u_{i,j} = 2 * x_{\min,j} - u_{i,j}$,if $u_{i,j} < x_{\min,j}$ and $u_{i,j} = 2 * x_{\max,j} - u_{i,j}$, if $u_{i,j} > x_{\max,j}$, and go to step 6.

Step 6) Calculate the objective function value for vector U_i.

Step 7) Choose better of the two (function value at target and trial point) using mutation equation for next generation.
Step 8) Check whether convergence criterion is met if yes then stop, otherwise go to step 3.

VII.RESULTS OF DE-PID

The PID controller in the AVR system is tuned by using DE with the performance criterion IAE, ISE, ITSE, ITAE.

1) Step response with DE-PID controller, IAE as performance criterion.

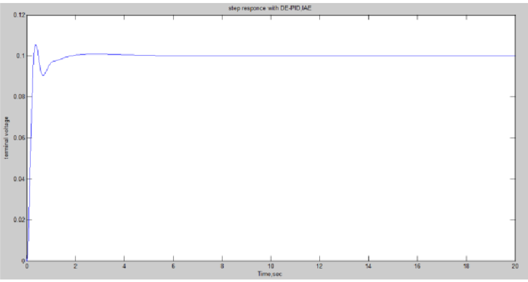

Fig.8 Step response of DE-PID, IAE.

With IAE the obtained step response of AVR system maximum peak overshoot is not too less or high it is 5.4536(%). And rise time and settle time are also less 1.2s and 1.3s respectively. Steady state error is zero.

2) Step response with DE-PID controller, ISE as performance criterion

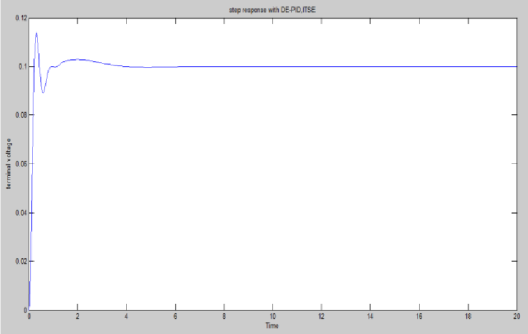

Fig.9 Step response of DE-PID, ISE.

3) Step response with DE-PID controller, ITAE as performance criterion.

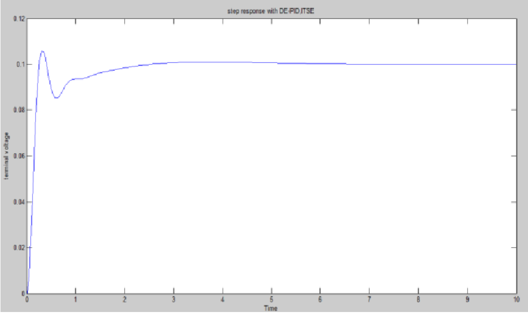

Fig.10 Step response of DE-PID, ITAE.

By considering the ITAE as performance index the step response almost same as response with IAE.

4) Step response with DE-PID controller, ITSE as performance criterion

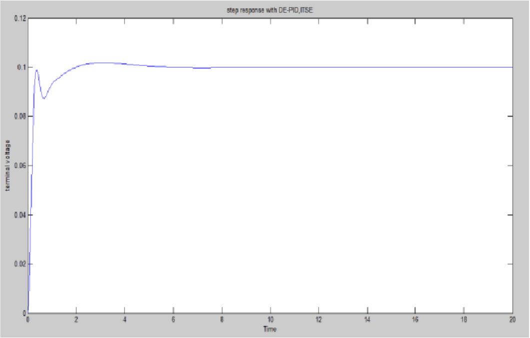

Fig.11 Step response of DE-PID, ITSE

The best result obtained while tuning of PID with DE technique is with performance index as ITSE. The step response has maximum overshoot %Mp=1.8558, rise time t_r=1.4s, settle time is 1.59s. By observing these parameters response is fast and not oscillatory. And the steady state error is completely eliminated.

Table 1 comparison of PSO-PID and DE-PID results

Criteria	Controller type	K_p	K_i	K_d	%Mp	E_{ss}	t_s	t_r
IAE	PSO-PID	0.3316	0.1646	0.2491	0.1562	0	4.4	1.26
	DE-PID	0.7152	0.5521	0.3681	5.4536	0	1.3	1.2
ISE	PSO-PID	0.7727	0.3226	0.1834	6.485	0	2.60	1.6
	DE-PID	0.7587	0.9823	0.3123	9.000	0	2.19	1.4
ITSE	PSO-PID	0.5019	0.4863	0.2631	2.880	0	3.07	1.8
	DE-PID	0.5342	0.4597	0.3510	1.8558	0	1.59.	1.4
ITAE	PSO-PID	0.3000	0.1864	0.1367	0.1069	0	2.73	3.9
	DE-PID	0.6384	0.4635	0.4481	5.9348	0	1.87	1.0

In PID controller tuning process Particle Swarm Optimization and Differential Evolution techniques are compared. By observing the obtained results DE-PID giving better responses for all performance estimation compared with PSO-PID.

VIII. CONCLUSIONS

In this paper the PID controller parameters have been tuned by Particle Swarm Optimization (PSO), Differential Evolution (DE). After inserting PID in the AVR system and after tuning it by PSO, results for four different criteria has been observed.

For the same population, crossover rate and number of generation in Differential Evolution, all tuning methods demonstrated the almost same performance in searching the best values of Minimum Integral Error. Also, the proposed method has resulted in better dynamic performance as well as transient response of the system. Further, the DE algorithm took less time to reach the optimal solution. When PID-PSO is compared with DE-PID later one gives the better responses than earlier one.

These modern optimization techniques can be applied to multi objective functions. How to search a efficient values of gains of multi objective PID controller of multi-machine system is one of future scope of work.

REFERENCES

[1]. Z. L. Gaing, "A particle swarm optimization approach for optimum design of PID controller in AVR system," IEEE Trans. Energy Conversion, vol. 19, no. 2, pp. 384-391, June 2004.

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

[3]. Kwang Y.Lee Mohamed E. El-Hawary *Modern Heuristic Optimization Techniques,* theory and applications to power systems, IEEE press, John Wiley & Sons, 2008.

[4]. S.P. Ghoshal, Optimization of PID gains by particle swarm optimization in fuzzy based automatic generation control, Electr. Power Syst. Res. 72 (2004) 203–212.Y. Shi and R. Eberhart,“A modified particle swarm optimizer,”inProc. IEEE Int. Conf. Evol. Comput., Anchorage, AK, May 1998, pp. 69–73.

[5]. Y. Shi and R. C. Eberhart,“Empirical study of particle swarm optimization,”in Proc. IEEE Int. Conf. Evol. Comput., Washington, DC, July 1999, pp. 1945–1950.

[6]. R. C. Eberhart and Y. Shi,“Comparison between genetic algorithms and particle swarm optimization,”in Proc. IEEE Int. Conf. Evol. Comput., Anchorage, AK, May 1998, pp. 611-616.

[7]. Kennedy J and Eberhart RC (1997), A Discrete Binary Version of the Particle Swarm Algorithm, In Proceedings of the 1997 Conference on Systems, Man, and Cybernetics, IEEE Service Center, Piscataway, NJ, pp. 4104–4109.

[8]. M.S.Saad, H.Jamaluddin “Implementation of PID controller tuning using differential evolution and genetic algorithms” IJICIC (2012) pp.7761-7779.

[9]. T. Andromed, A. yahya, S. Samion, A. Baharom, N. L. Hashim “PID Controller Tuning by Differential Evolution Algorithm on EDM Servo Control System” Applied Mechanics and Materials Vols. 284-287 (2013) pp 2266-2270.

[10]. M. L. Amer Hanafy H. H. Hassan Hosam K. M. Youssef “Modified Evolutionary Particle Swarm Optimization for AVR-PID tuning” Communications & Information Technology 2008.

[11]. D. Ruijun, Differential Evolution versus Particle Swarm Optimization for PID Controller Design. In Fifth International Conference on Natural Computation, ICNC'09. 2009.

[12]. Bingul and Zafer, A New PID Tuning Technique Using Differential Evolution for Unstable and Integrating Processes with Time Delay, in Neural Information Processing, Pal, et al., Editors. 2004 Springer Berlin / Heidelberg. p. 254-260.