UNIVERSITY OF MAURITIUS RESEARCH JOURNAL – Volume 16 – 2010
University of Mauritius, Réduit, Mauritius

Research Week 2007

Optimization of AVR Parameters of a Multi-machine Power System Using Particle Swarm Optimization

Hervin K. Shewtahul

Robert T F Ah King*
Faculty of Engineering
University of Mauritius
Email: r.ahking@uom.ac.mu

Harry C S Rughooputh
Faculty of Engineering
University of Mauritius
University of Mauritius.
Email: r.rughooputh@uom.ac.mu

Paper Accepted on 18 January 2010

Abstract

In this paper, a method for optimizing the parameters of Automatic Voltage Regulation (AVR) system installed on the generators of a multi-machine power system using Artificial Intelligence (AI) techniques is presented. Each AVR system is equipped with a PID (Proportional, Integral and Derivative) controller and a Power System Stabilizer (PSS). Two methods are presented, which are the Particle Swarm Optimization (PSO) and Genetic Algorithm (GA). The robustness of the AI algorithms is examined by studying the time-domain behavior of the system following different disturbances. The AI techniques provide a much simpler way to solve this non-linear system compared to classical techniques.

Keywords: multi-machine power system stability, AVR system, power system stabilizer, PID controller, particle swarm optimization, genetic algorithm.

* *For correspondences and reprints*

1. INTRODUCTION

A stable voltage level is crucial in a power system. Insulation, security, economy and quality of service are some of the factors that put upper and lower bounds to voltage levels. In modern power systems, there exist various mechanisms on the different networks (that is, the generation, transmission and distribution networks) to control the voltage level. In this work, the voltage profile at generators' terminal after a major disturbance is of concern. Other types of voltage control mechanisms include changing taps of transformers and shunt/series capacitor compensation.

Steady-state frequency is also needed in a power system. One major reason for this is that at some consumers' place, the production line is synchronized to the power line. Constant frequency is also necessary for parallel operation of generators and proper functioning of other equipment which have stringent frequency specifications like some protective relays.

The major cause of variations in voltage levels and operating frequency is the ever-changing active and reactive power demand. As a consequence, the power input to generators must be changed to match the demand in order to prevent undesirable frequency change. Also, the excitation of generators must be continuously regulated to match the reactive power demand with reactive power generation; otherwise, the voltages at various system buses may go out of the prescribed limits.

Many methods have been proposed for tuning PID (Proportional, Integral and Derivative) controllers and for the design of PSS blocks, both in sequential and simultaneous ways for multi-machine system. The most popular methods of tuning PID controllers are probably the Ziegler-Nichols methods. It remains, however, tedious to obtain an optimal (or even near-optimal) solution with these methods for a multi-machine system. Some methods for designing PSS in a multi-machine system were presented in (Abido 2002) and (Chen & Hsu 1987). These methods are limited due to the linearization and simplification involved. And they are often limited to linear controllers. This is what has motivated the use of PSO for this optimization problem, since it can be quite easily extended to more complex controllers and more comprehensive system models. Some recent approaches using intelligent techniques applied to multi-machine AVR control and PSS designs include genetic algorithms, fuzzy systems, neural and neuro-fuzzy systems and more recently particle swarm optimization.

Minimum phase control loop method and genetic algorithm (GA) have been proposed for off-line tuning of PSS in a multi-machine power system (Hongesombut et al. 2002). The PSS parameter problem was converted to an optimization problem and solved using micro-GA and Hierarchical GA was used for automatically identifying the PSS locations. In (Park et al. 2005), the dual heuristic programming optimization algorithm was applied for the design of two local nonlinear optimal neuro-controllers on a practical multi-machine power system. One neuro-controller was designed to replace the conventional linear controllers: automatic voltage regulator and speed-governor for a synchronous generator. The other was an external neuro-controller for the series capacitive reactance compensator, flexible ac transmission systems device.

Elshafei et al. (2005) proposed a power system stabilizer based on adaptive fuzzy systems. The proposed controller was a fuzzy-logic-based stabilizer that has the capability to adaptively tune its rule-base on line. The change in the fuzzy rule base was done using a variable-structure direct adaptive control algorithm to achieve the pre-defined control objectives. A robust artificially intelligent adaptive neuro-fuzzy power system stabiliser (ANF PSS) design for damping electromechanical modes of oscillations and enhancing power system synchronous stability was presented in (Barton 2004). The power system was decomposed into separate subsystems; each subsystem consisting of one machine and the local ANF PSS was associated with each subsystem where the local feedback controllers relied only on information particular to their subsystem and the input signals were the speed, power angle and real power output.

The problem of simultaneous and coordinated tuning of stabilizers parameters and automatic voltage regulators gains in multi-machine power systems was considered in (El-Zonkoly 2006). This problem was formulated as an optimization problem and solved using particle swarm optimization technique. The objective of the parameters optimization was formulated as nonlinear problem with constraints to represent the allowable region of the system parameters. More recently, a craziness based particle swarm optimization and binary coded genetic algorithm were used to obtain the optimal PID gains (Mukherjee & Ghoshal 2007). For on-line off-nominal system parameters Sugeno fuzzy logic (SFL) was applied to get on-line terminal voltage response. SFL was used to extrapolate the nominal optimal gains in order to determine off-nominal optimal gains.

Recently, many artificial intelligence (AI) techniques have been applied to power system design and control. These methods have proven to be very promising. PSO is one such method and is based on swarm intelligence (Chatterjee et al. 2009, Zamani et al. 2009). PSO was introduced in 1995 by Eberhart R. and Kennedy J. (Eberhart & Kennedy 1995). The technique tries to implement the social and cognitive behavior of individuals in a flock, such as fish schooling or locust swarms. The algorithm used is very simple and, thus, computer efficient.

This paper deals with the application of PSO technique aiming to find the optimal values of the proportional, integral and differential gains and time constants for a PID controller installed on the generators.

The paper overview is as follows:
In section 2, the particle swarm optimization algorithm is presented. In section 3, the PID Control Scheme is explained together with the search-space parameters while section 4 presents the simulation results. Section 5 concludes the paper outlining the advantages of the method.

2. PARTICLE SWARM OPTIMIZATION (PSO)

Figure 1 shows the diagram of the AVR system used for the generators (including a PSS block) (Anderson & Fouad 1993). The sensors' model is approximated to first order transfer functions as described in (Saadat 1999). The lead network block is, in fact, two lead compensators in cascade, as denoted by the following expression 1.

$$\frac{1+\tau_1 s}{1+\tau_2 s}\frac{1+\tau_3 s}{1+\tau_4 s} \quad \ldots (1)$$

The reset block has the following transfer function (2):

$$\frac{K_0 s}{1+\tau_0 s} \quad \ldots (2)$$

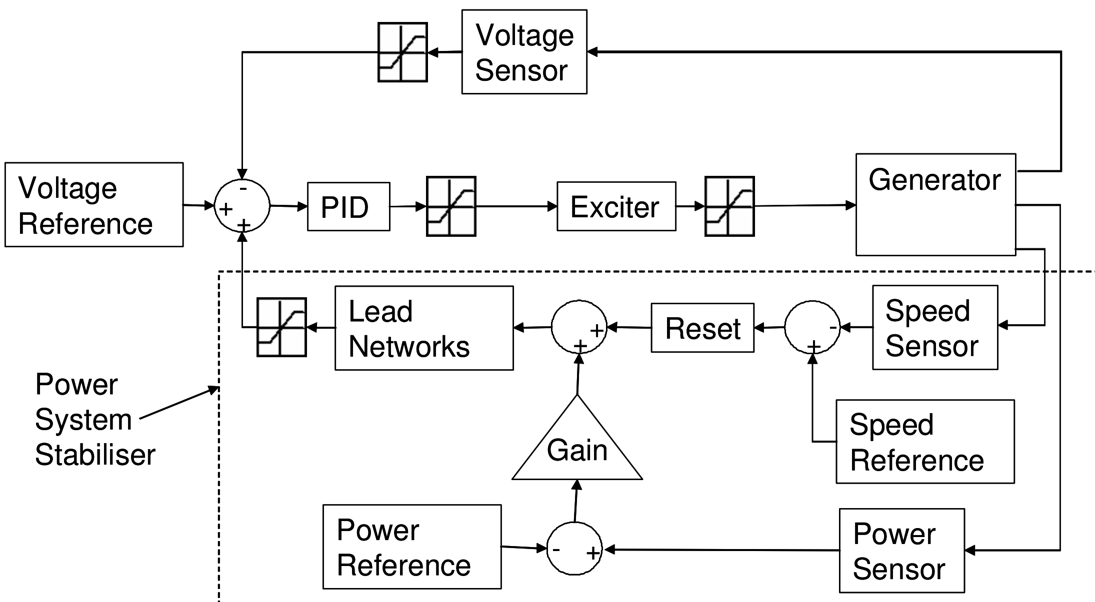

Figure 1: AVR system (Anderson & Fouad 1993)

In this section, we shall review the PSO method detailed in (Parsopoulos & Vrahatis 2002) and applied in section 3 for designing the PID controller of Fig.1.

PSO works by using a population of particles each of which travels through a search-space (usually bounded). Each particle's motion is affected by the flock's (social) and its own (cognition) experience. A particle represents a possible solution, which is evaluated with an objective function. The particle remembers the best solution (position in the search-space) it has been through (cognition) and the best solution of the entire flock (social). In addition to that, there are random variations in the velocity of each particle which allow a more thorough exploration of the search-space. Recent research (Eberhart & Shi, 1998) has shown that PSO works better by concentrating on global search at the start and emphasizing on local exploration at the end of the search (Parsopoulos & Vrahatis 2002). PSO offers many advantages over conventional methods for optimization.

a) Each particle is evaluated using an objective function. Hence, PSO can easily deal with very complex (and even non-differentiable) system of equations, where many conventional methods, such as gradient-based techniques, will fail. Moreover, many of the approximations needed by traditional methods can be dropped when using PSO.

b) Since there is a population of particles flying around the workspace, each carrying individual searches, there exist a sort of parallel search which decreases the risk getting trapped in local optima.

c) Initialisation is not of capital importance in PSO, unlike many other traditional methods.

d) PSO provides the flexibility of balancing between global and local exploration, which enhances the search results. This feature helps PSO to, sometimes, outperform other AI techniques (such as genetic algorithms).

The PSO algorithm is based on particles flying around the search-space. Mathematically, the particle is a vector of dimension *m*, where *m* is the number of parameters to be optimized. This vector is a candidate solution to the problem. Particle *i* is thus denoted as:

$$P_i(t) = \left[p_{i,1}(t)\ , p_{i,2}(t)\ , \ldots, p_{i,m}(t)\right] \qquad ,i=1,2,\ldots,n \qquad \ldots (3)$$

where, the *p*'s are the optimized parameters,

t is the time index (iteration number), and,

n is the number of particles.

The velocity of a particle is defined as the change in position of the particle over iteration. It is thus a vector of dimension *m*, denoted as:

$$V_i(t) = \left[v_{i,1}(t)\ , v_{i,2}(t)\ , \ldots, v_{i,m}(t)\right] \qquad ,i=1,2,\ldots,m \qquad \ldots (4)$$

The algorithm used for the implementation of PSO for this work is given in equations (5), (6) and (7) stated in (Parsopoulos & Vrahatis 2002).

$$\begin{aligned} v_{i,j}^{(t+1)} &= (w^{(t)} \times v_{i,j}^{(t)}) + (c_1 \times rand \times (pbest_{i,j}^{(t)} - p_{i,j}^{(t)})) \\ &\quad + (c_2 \times rand \times (gbest_j^{(t)} - p_{i,j}^{(t)})) \end{aligned} \qquad \ldots (5)$$

$$p_{i,j}^{(t+1)} = p_{i,j}^{(t)} + v_{i,j}^{(t)} \qquad \ldots (6)$$

$$w^{(t+1)} = w_{\max} - a \times t \qquad \ldots (7)$$

$$i = 1, 2, \ldots, n$$

$$j = 1, 2, \ldots, m$$

where,

n	number of particles in the swarm,
m	number of optimized parameters,
t	time index (iteration number),
$v_{i,j}^{(t)}$	component of the velocity of the i^{th} particle with respect to the j^{th} dimension at iteration *t*,
$p_{i,j}^{(t)}$	position of the i^{th} particle in the j^{th} dimension at iteration *t*,
$w^{(t)}$	inertia weight factor at iteration *t*,
c_1	constant governing the cognitive behavior of each particle,
c_2	constant governing the social behavior of the group,

rand a random number from the standard uniform distribution,
$pbest_{i,j}^{(t)}$ a vector of dimension m representing the best position visited by particle *i* as at iteration *t*,

$gbest_j^{(t)}$ a vector of dimension m representing the global best position visited by the swarm as at iteration *t*,
a inertia weight's decrement
w_{max} maximum inertia weight's value

The constants c_1 and c_2 determine the behavior of the individual particles in the swarm. A high value of c_1 (or small value of c_2) results in an "individualistic" behavior, which decreases the chances of finding the global optima. Nevertheless, a low value of c_1 (or large value of c_2) will result in all the particles behaving almost similarly, which diminishes the parallel search feature. Another constant known as the neighborhood constant is sometimes included. The neighborhood constant models the attraction to the best solution in the "neighborhood" of the particle.

The velocity of each particle is bounded as,

$$V_{j\min} \le v_{i,j}^{(t)} \le V_{j\max}$$

By bounding the velocity, the resolution of the search is set. Usually,

$$V_{\min} = -V_{\max}\,. \qquad \ldots (8)$$

Thus small value of V_{max} gives high resolution. It, however, increases the convergence time and it can also cause the particles to stick to local optima.

3. PID CONTROL SCHEME

The PID controller has three adjustable parameters, which are the proportional gain (k_p), integral gain (k_i) and derivative gain (k_d). The PSS has four parameters to be optimized: k_0, k_1, τ_1 and τ_3. k_0 is the gain of the speed offset signal and k_1 is the gain of the power offset signal. Each individual are defined as a vector of dimension seven.

$$p = [k_p, k_i, k_d, k_0, k_1, \tau_1, \tau_3]$$

The search-space is, thus, of dimension seven. The boundaries of the search-space were defined as follows.

$$0 \le k_p \le 50,$$
$$0 \le k_d \le 20,$$
$$0 \le k_i \le 20,$$
$$0 \le k_0 \le 150,$$
$$0 \le k_1 \le 150,$$
$$0 \le \tau_1 \le 2,$$
$$0 \le \tau_3 \le 2.$$

The generator speeds are then set to a random value from a normal distribution with mean equal to the synchronous speed and variance equal to ζ/(inertia constant). The action of the AVR systems and the power system is then simulated over 10 seconds. The simulation algorithms are based on the work from (Kusic 1986, Stagg & El-Abiad 1968, deMello & Concordia 1969). The response is evaluated in the time domain using an objective function, defined below. ζ is a constant, which was set to 30 for the New England power system.

$$performance = \sum_{n=0}^{N} \int_{t=5}^{t=10} \left(change\, in\, speed \times (t-5)\right) dt + \beta \times \sum_{n=0}^{N} \int_{t=1.5}^{t=10} \left(change\, in\, voltage\, level \times (t-1.5)\right) dt \quad \ldots (9)$$

where N is the number of generators present.

As outlined in (Chen & Hsu 1987), a compromise must be set between the voltage profile and the frequency deviation. This can be done by varying β, which is a weight constant. For the New England 10-generator 39-bus system, β was set to 0.0005. Moreover, a condition is inserted in the objective function, which assigns a large performance value (a penalty) if the voltage deviation exceeds 6% 1.5 seconds after the occurrence of the fault. This condition simulates the power systems' restrictions on voltage level. In the optimization, any individual giving fluctuations of more than 6% was thus rejected.

A MATLAB program was written to perform the optimization process. The procedure adopted for the search can be broken down into the following steps:

Step 1: The search-space was defined and a population of 10 random individuals was initialized.

Step 2: Constants c_1 is set to 1.5 and c_2 is set to 0.8. w (inertia weight constant) = 0.9 and a =0.01. V_{max} is set to [k_p/2, k_i/2, k_d/2, k_0/2, k_1/2, τ_1/2, τ_3/2] and the number of iteration is set to 100. These parameters were found to be most effective after several runs of the optimization process.

Step 3: Each particle is evaluated with the objective function, and $pbest_j$ and $gbest$ are updated.

Step 4: The velocity of each particle is updated using equation (5).

Step 5: The position of each particle is adjusted according to their velocities as follows,

$$p_{i,j}^{(t+1)} = p_{i,j}^{(t)} + v_{i,j}^{(t+1)}, \quad \ldots (10)$$

and limited by the following inequalities,

$$0 \le p_{i,j}^{(t)} \le K_{i\max}, \quad \ldots (11)$$

where K_{imax} =[k_{pmax} , k_{imax} , k_{dmax} , k_{0max}, k_{1max}, τ_{1max}, τ_{3max}]

Step 6: Inertia weight constant is updated and the iteration number is increased. If maximum is not reached, step (3) is repeated else step (7) is executed.

Step 7: The latest *gbest* gives the best set of parameters found.

4. RESULTS

The optimization process was run for the New England power system shown in Figure 2. The US-Canadian link is modeled as a constant excitation generator (generator 1). Since an objective function was already defined, a Genetic Algorithm (GA) (Chipperfield et al. 1994) was also used to optimize the system, as a comparison. The same number of function evaluations was used for the GA for the comparison. The GA used generalized rank-based fitness assignment, stochastic universal sampling for selection, real-value mutation and discrete recombination (Chipperfield et al. 1994).

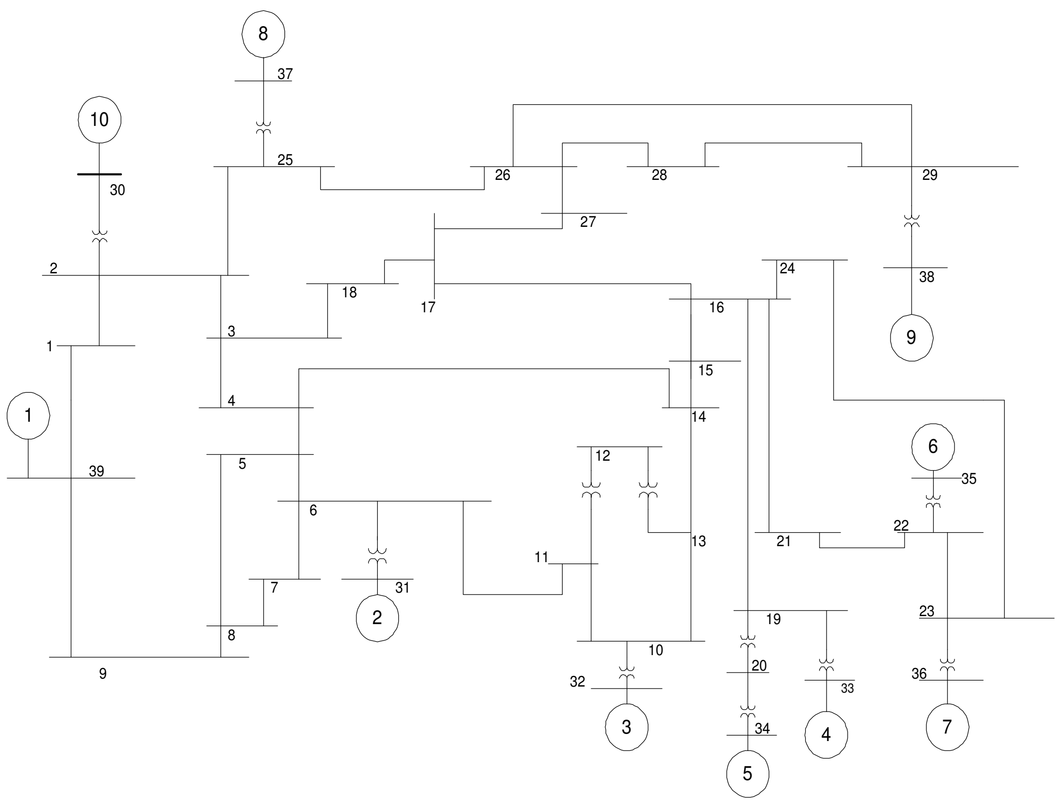

Figure 2: New England power system

All parameters of the PID controller and the PSS were initially set at random numbers within the boundaries stated earlier (same as in Anderson & Fouad 1993). A set of ten independent runs was made for both PSO and GA and the results of the performance values (eqn. (9)) obtained are presented in Table 1.

Table 1: Performance of PSO and GA on ten independent runs

Run Number	*PSO*	*GA*
1	0.01113	0.01345
2	0.01358	0.01229
3	0.01259	0.01310

4	0.01482	0.01295
5	0.01588	0.01274
6	0.01287	0.01673
7	0.01246	0.01248
8	0.01299	0.01610
9	0.01301	0.01297
10	0.01256	0.01301
Minimum	***0.01113***	***0.01229***
Maximum	***0.01588***	***0.01673***
Mean	***0.01319***	***0.01358***
Standard Deviation	***0.00132***	***0.00153***

By inspecting the values given by PSO and GA:

$$performance_{PSO} = mean \pm standard\ \ deviation = 0.01319 \pm 0.00132$$

$$performance_{GA} = mean \pm standard\ \ deviation = 0.01358 \pm 0.00153$$

we find that theses ranges are very close to each other, hence if we define

$$\Delta performance_{\max} = performance_{PSO,\max} - performance_{GA,\max} = 0.01511 - 0.01451 = 0.00060$$

and

$$\Delta performance_{\min} = performance_{PSO,\min} - performance_{GA,\min} = 0.01205 - 0.01287 = -0.00082$$

.

These differences are very small to judge which one of PSO or GA is better. Hence, we can say that PSO and GA performance are very close on this particular problem.

Tables 2 and 3 give the best parameters obtained by PSO and GA respectively. It is to be noted that both PSO and GA converge to a value of zero for K_1 in all runs. These results may suggest that it is advisable to set the power offset signal gain to zero for a more stable system. Moreover, only the PSO algorithm converges to a value of zero for k_d in all runs. The implication is that the system gives best performance when k_d is zero according to PSO. In practice, it is well established that k_d causes a system to be less stable.

Table 2: Parameters obtained by PSO

Generator	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***
k_p	0.1203	0.1548	0.11151	0.12321	0.8143	0.11219	0.77566	0.1208	0.76398	0.21568
k_i	0.1077	0.1357	0.001	0.001	0.001	0.001	0.0135	0.1227	0.001	0.001
k_d	0	0	0	0	0	0	0	0	0	0
K_0	34.276	45.809	23.4765	21.0603	35.63	48.5948	46.8033	28.9452	55.3822	98.8966
K_1	0	0	0	0	0	0	0	0	0	0
T_1	0.9808	0.6963	0.787	0.9198	0.3348	0.5401	0.3685	0.9741	1.738	1.1114
T_3	0.9756	0.6515	0.7618	0.6042	0.3311	0.9233	0.2253	0.9947	1.2992	0.9356

Table 3: Parameters obtained by GA

Generator	1	2	3	4	5	6	7	8	9	10
k_p	0.5	5.9869	0.6314	0.6216	0.5807	3.8934	31.8357	8.2631	8.4071	6.5763
k_i	0.4839	6.006	4.3181	0.3846	1.7606	0.8704	1.9796	0.1851	0.2696	6.5229
k_d	0.2221	8.6838	6.1548	5.695	1.5485	0.0354	1.6131	0.0289	0.4887	0.0022
K_0	0.1	150	149.9878	146.8611	0.1244	149.9756	149.9756	149.9756	150.0244	150.366
K_1	0	0	0	0	0	0	0	0	0	0
T_1	1.5573	1.6828	0.3233	1.3695	1.3747	0.4124	1.1166	1.0111	1.1621	1.1405
T_3	1.2362	1.0328	0.9154	1.1624	0.5569	1.6237	0.2998	0.7157	1.4739	1.3397

The optimized system was then tested with different disturbances and system conditions. Figures 3 to 9 show the response of the system. The responses for a classical machine represented as a constant voltage source behind its transient reactance (no controller in this case) are also shown.

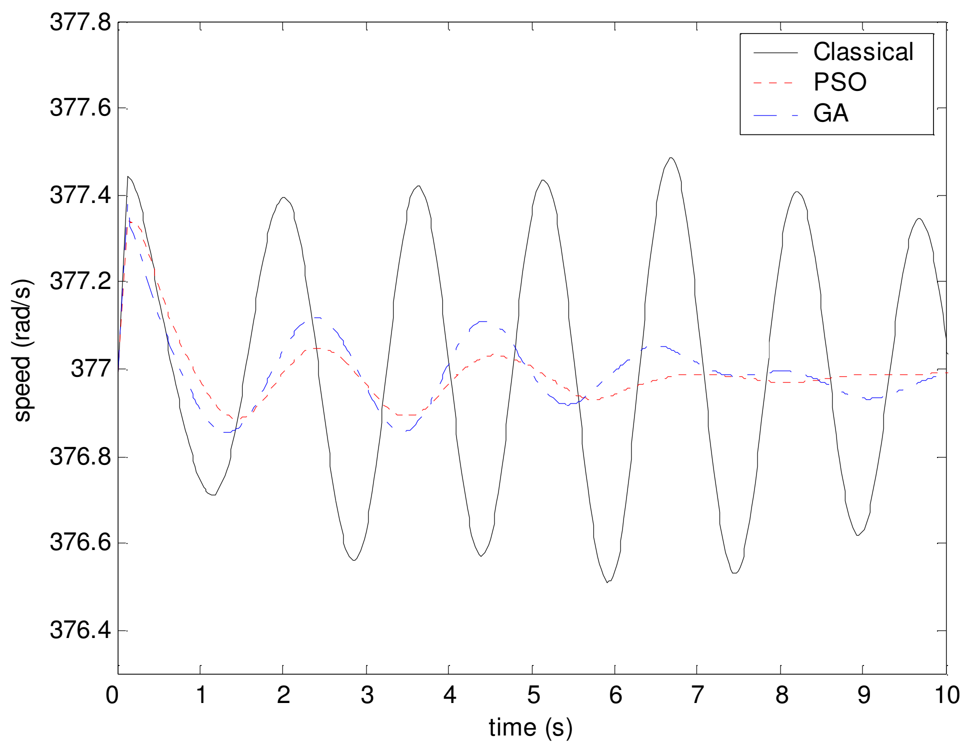

Figure 3: Speed of generator 6 following a 6-cycle 3-phase fault near bus 36

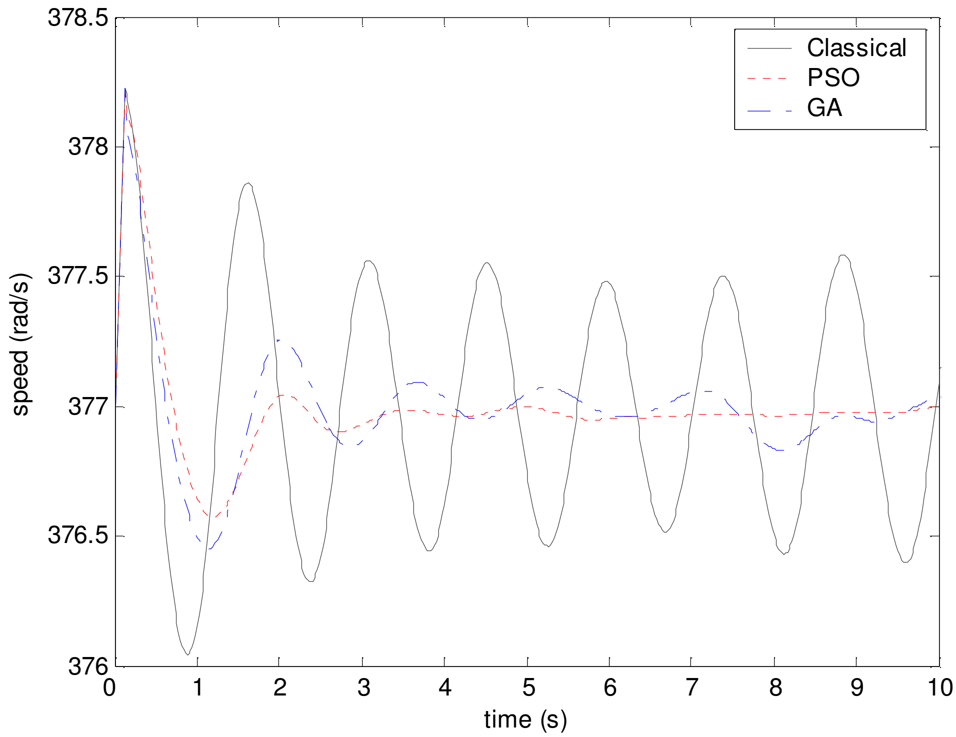

Figure 4: Speed of generator 7 following a 6-cycle 3-phase fault near bus 36

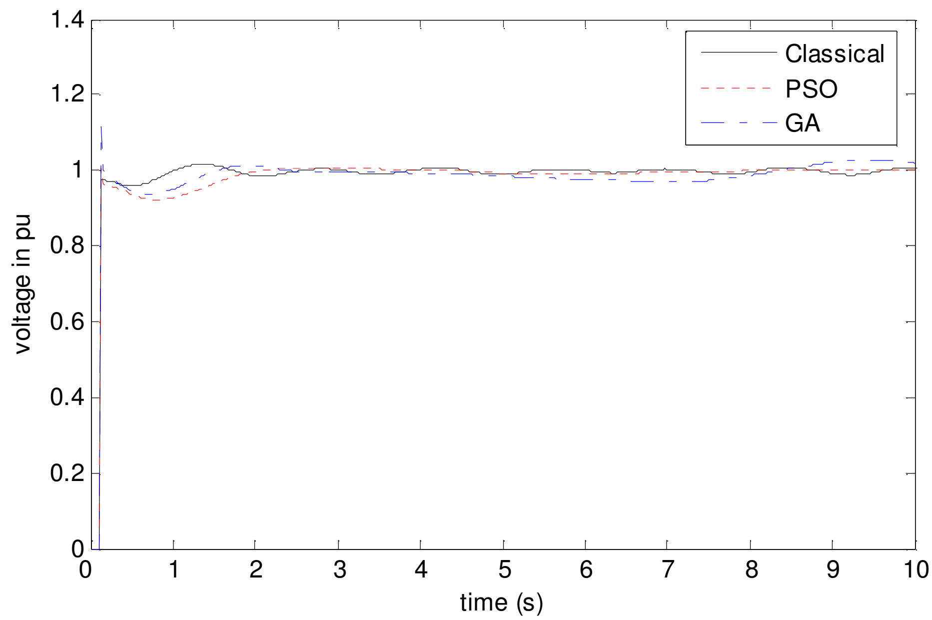

Figure 5: voltage at bus 36 following a 6-cycle 3-phase fault near bus 36

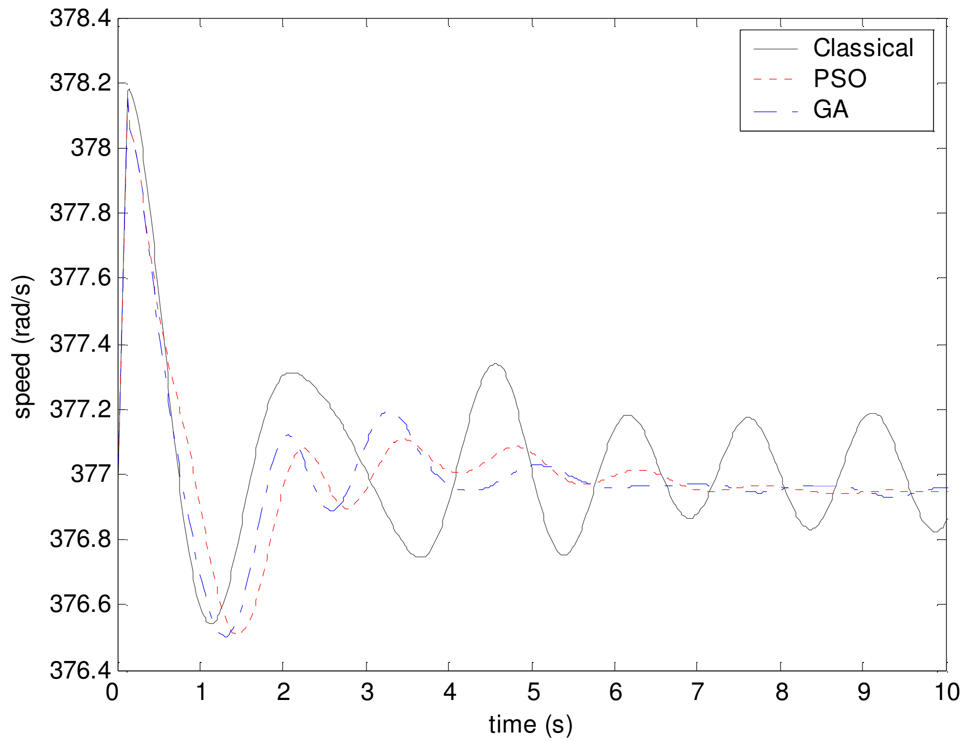

Figure 6: Speed of generator 7 following a 6-cycle 3-phase fault near bus 16 cleared by opening line 16-17

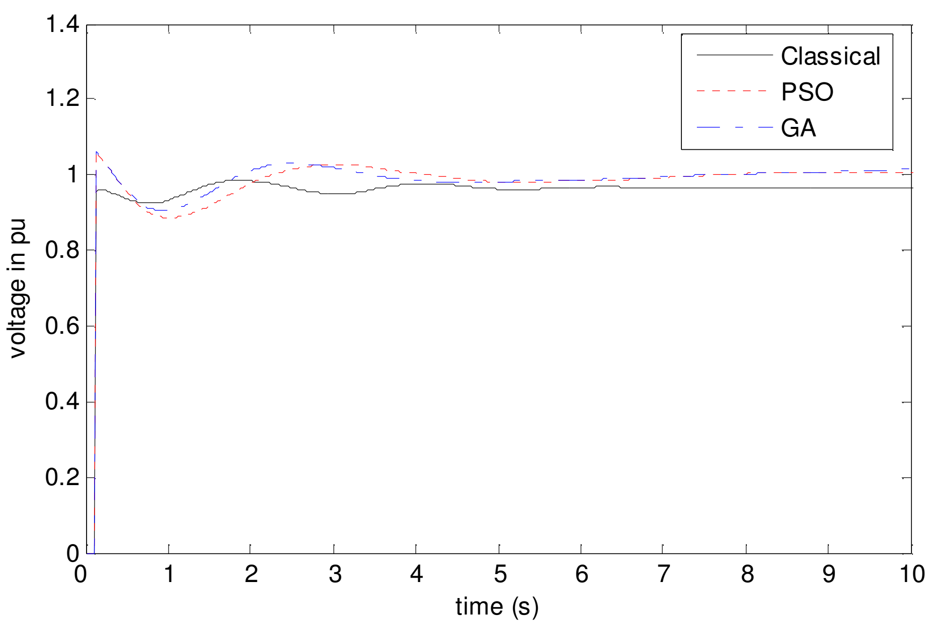

Figure 7: Voltage at bus 16 following a 6-cycle 3-phase fault near bus 16 cleared by opening line 16-17

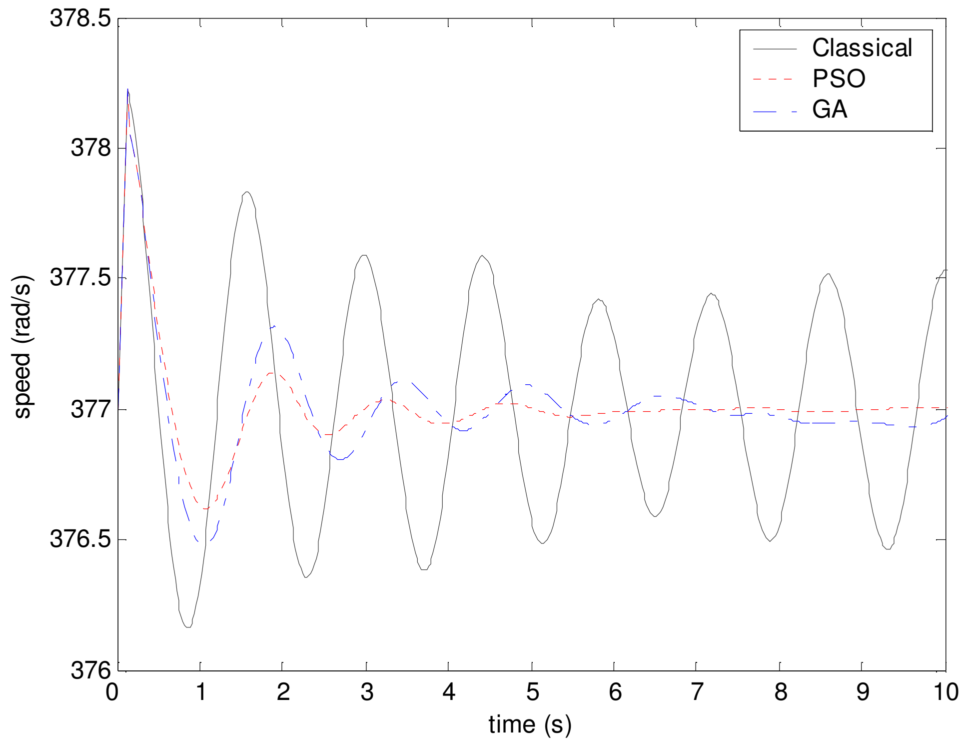

Figure 8: Speed of generator 7 following a 6-cycle 3-phase fault near bus 36 with loading increased by 25%

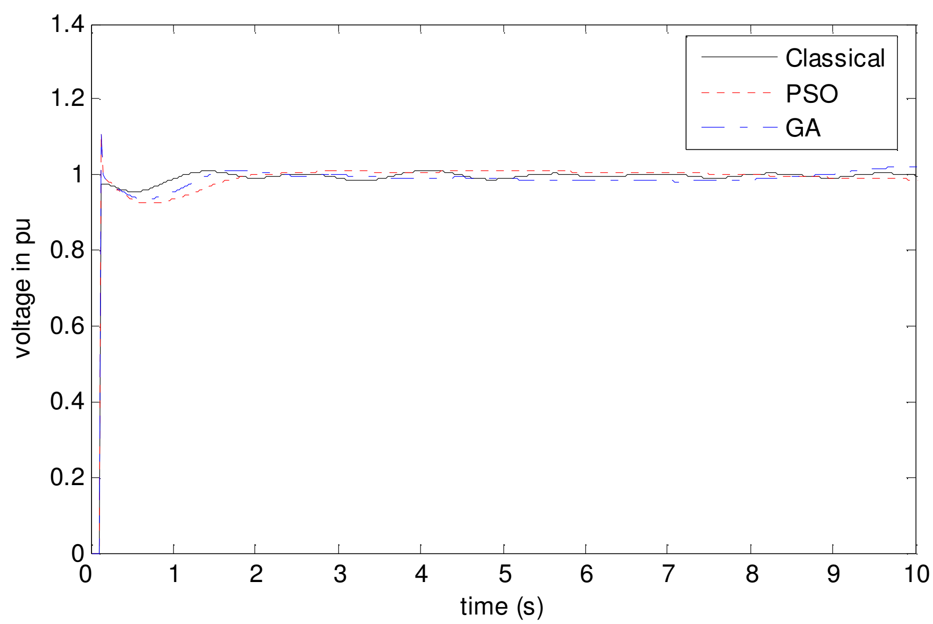

Figure 9: Voltage at bus 36 following a 6-cycle 3-phase fault near bus 36 with loading increased by 25%

As observed from the responses, PSO and GA give comparable results for the particular set of disturbances considered. Oscillations are sustained for the classical machine representation whereas the PSO optimized AVR and GA optimized AVR achieve good damping characteristics in both generator speed and voltage profiles.

5. CONCLUSIONS

This paper has presented methods for optimizing AVR system of a multi-machine power system using AI techniques. The proposed methods have been applied successfully to conventional controllers (PID and lead networks). The methods presented take into account some real-life constraints such as the boundaries imposed on the voltage levels. The designed system has been tested against a number of disturbances and under different system configurations.

Satisfactory performance of the optimized power system network is noted. The AI methods used provide a simple solution to solving a non-linear and discontinuous system, compared to classical methods, which involves more advanced mathematics. The fine tuning of the different optimization parameters of PSO and GA is one of the future works.

The main advantage of this method is that it can be extended to more complex controllers and more comprehensive system models.

6. REFERENCES

ABIDO, M.A. (2002). Optimal design of power-system stabilizers using particle swarm optimization. IEEE Transactions on Energy Conversion 17 (3), 406-413.

ANDERSON, P.M. & FOUAD, A.A. (1993). Power system control and stability. IEEE Power Systems Engineering Series. IEEE Press, New York.

BARTON, Z. (2004). Robust control in a multimachine power system using adaptive neuro-fuzzy stabilizers. IEE Proceedings Generation, Transmission and Distribution 151 (2), 261-267.

CHATTERJEE, A., MUKHERJEE V. & GHOSHAL, S.P. (2009). Velocity relaxed and craziness-based swarm optimized intelligent PID and PSS controlled AVR system, International Journal of Electrical Power & Energy Systems 31 (7-8), 323-333.

CHEN, C.L. & HSU, Y.Y. (1987). Coordinated synthesis of multi-machine power system stabilizer using an efficient decentralised modal control algorithm. IEEE Transactions on Power Systems 2 (3), 543-550.

CHIPPERFIELD, A., FLEMING, P., POHLHEIM, H. & FONSECA, C. (1994). Genetic algorithm toolbox (version 1.2) user's guide. Department of Automatic Control and Systems Engineering, University of Sheffield.

DEMELLO, F.P. & CONCORDIA, C. (1969). Concepts of synchronous machines stability as affected by excitation control. IEEE Transactions on Power Apparatus and Systems PAS-88 (4), 316-329.

ELSHAFEI, A.L., EL-METWALLY, K.A. & SHALTOUT, A.A. (2005). A variable-structure adaptive fuzzy-logic stabilizer for single and multi-machine power systems. Control Engineering Practice 13 (4), 413-423.

EL-ZONKOLY, A.M. (2006). Optimal tuning of power systems stabilizers and AVR gains using particle swarm optimization. Expert Systems with Applications 31 (3), 551-557.

HONGESOMBUT, K., MITANI, Y. & TSUJI, K. (2002). Power system stabilizer tuning in multimachine power system based on a minimum phase control loop method and genetic algorithm. In Proceedings of the 14th Power Systems Computation Conference (PSCC'02), Sevilla, Spain, 24-28 June 2002.

KENNEDY, J. & EBERHART, R. (1995). Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, vol. IV, Perth, Australia, 1942-1948.

KUSIC, G.L. (1986). Computer-aided power system analysis. Prentice-Hall, Inc., Englewood Cliffs, NJ.

MUKHERJEE, V. & GHOSHAL, S.P. (2007). Intelligent particle swarm optimized fuzzy PID controller for AVR system, Electric Power Systems Research 77 (12), 1689-1698.

PARK, J.-W., HARLEY, R.G. & VENAYAGAMOORTHY, G.K. (2005). Decentralized optimal neuro-controllers for generation and transmission devices in an electric power network. Engineering Applications of Artificial Intelligence 18 (1) 37-46.

PARSOPOULOS, K.E. & VRAHATIS, M.N. (2002). Recent approaches to global optimization problems through Particle Swarm Optimization. Natural Computing 1 (2-3), 235-306.

SAADAT, H. (1999). Power system analysis. WCB/McGraw-Hill Co. Ltd., Boston, MA.

STAGG, G.W. & EL-ABIAD, A.H. (1968). Computer methods in power system analysis. McGraw-Hill Kogakusha Ltd., Tokyo.

ZAMANI, M., KARIMI-GHARTEMANI, M., SADATI, N. & PARNIANI, M. (2009). Design of a fractional order PID controller for an AVR using particle swarm optimization, Control Engineering Practice 17 (12), 1380-1387.