International Journal of Engineering and Technology Innovation, vol. 4, no. 4, 2014, pp. 223-233

Applying Sequential Particle Swarm Optimization Algorithm to Improve Power Generation Quality

Abdulhafid Sallama, Maysam Abbod*, Shariq Mahmood Khan

School of Engineering and Design, Brunel University, London, UK.

Received 26 June 2014; received in revised form 31 July 2014; accepted 18 August 2014

Abstract

Swarm Optimization approach is a heuristic search method whose mechanics are inspired by the swarming or collaborative behaviour of biological populations. It is used to solve constrained, unconstrained, continuous and discrete problems. Swarm intelligence systems are widely used and very effective in solving standard and large-scale optimization, provided that the problem does not require multi solutions. In this paper, particle swarm optimisation technique is used to optimise fuzzy logic controller (FLC) for stabilising a power generation and distribution network that consists of four generators. The system is subject to different types of faults (single and multi-phase). Simulation studies show that the optimised FLC performs well in stabilising the network after it recovers from a fault. The controller is compared to multi-band and standard controllers.

Keywords: Particle swarm optimization, Fuzzy logic controller, Power system stabilizer and Adaptive neuro fuzzy interference system, Multi band stabilizer.

1. Introduction

Fuzzy logic control is a powerful tool which is used in various power system applications. The application of fuzzy logic is most suitable when a well-defined control objective cannot be specified, or the system is complex and an exact mathematical model is not available. Most power system stabilizers (PSS) are used in electric power systems to employ the classical linear control theory approach based on a linear model of a fixed configuration of the power system, such a fixed-parameter PSS [1]. However, FLC can be improved further when its parameters are optimized. Optimization algorithms such as particle swarm optimization (PSO) is capable 6f find5ng multiple optima and peaks in a single run as defined by machine learning problems [2].

Most advanced classic power system stabilizer controllers and try to compensate the power and change in frequency in the national grid as a result of changing the capacity in renewable energy plant [3] or after fault occurrence in order to amend the system to normal situation as soon as possible [4], in addition to maintaining stability and preventing the system from going into out of control situation (oscillation), which are presented by high voltage overshooting and oscillation for a long period of time [5]. This problem has been handled by designing complex control devices, containing sophisticated electronic circuits, which monitor the network and the generators at different levels and stages [6]. In order to reduce the oscillation and overshot, a neuro-fuzzy controller was utilized [7]. Such controller is trained for several faults stages in order to get the best result by

* Corresponding author. E-mail address: maysam.abbod@brunel.ac.uk
Tel.: +44-01895-267061; Fax: +44-01895-269695

using Adaptive Neuro-Fuzzy Inference Systems (ANFIS) which monitors the system and tries to manipulate the power generator in order to improve the transient behavior and reduce the oscillation which can occur at the start switching the generator or due to changing in capacity variation at deferent renewable energy plant or when faults that can occur at the transmission lines [8].

In this paper a sequential PSO system is being developed which is used to optimise the scaling factors of fuzzy logic controllers (FLC) that is used to control generators on power generation and distribution networks. The system is also controlled using multi-band controller and a non-PSS controller system. Different faults are simulated such as single and multi-phase faults. Results show that the performance of optimized FLC system is improved in terms of recovering from faults and stabilising the system.

2. Particle Swarm Optimisation

PSO is invented by Kennedy and Eberhart in the mid 1990s while attempting to simulate the choreographed, graceful motion of swarms of birds as part of a socio-cognitive study investigating the notion of “collective intelligence” in biological populations [9: 3]. In PSO, a set of randomly generated solutions (initial swarm) propagates in the design space towards the optimal solution over a number of iterations (moves) based on large amount of information about the design space that is assimilated and shared by all members of the swarm. PSO is inspired by the ability of flocks of birds, schools of fish, and herds of animals to adapt to their environment, found rich sources of food, and avoided predators by implementing an “information sharing” approaches, hence, developing an evolutionary advantage [10]. Naturally most of creatures behave during their movement as a swarm. One of the most important researches in artificial life today is to study the natural creatures in swarms, how they act like bird flocking or fish school and others, in addition to reconfiguring through mathematical models which can be handled through computer [11].

Particle swarm optimization is developed as a simulation approach of the bird flocking flow in two space dimensions (x, y). In this approach (vx) represents the agent velocity in the direction of x-axis and (vy) represents the agent velocity in the direction of y-axis. While (x, y) represents the agent current position and (vx, vy) represents the current velocities in two dimensions. The agent can be modified (updated) to the new position based on the velocity and position information [12]. Different versions of PSOs are investigated in this study; some are with either, "merits and demerits". The following versions are considered: Local search PSO (LPSO) [13], Global search PSO (GPSO) [14], Comprehensive Learning PSO (CLPSO) [15], Dynamic Multi Swarm PSO (DMS-PSO) [16], Dynamic Multi Swarm with Sun-regional Search PSO (DMS-SHS-PSO) [17], Adaptive PSO (APSO) [18], and Unified Search PSO (UPSO) [19]. The different algorithms tests are done using benchmarks with the following initial settings: 10,000 iteration (Generation), 20 particles, mean fitness of 10 trials and the initial PSO coefficients are W = 0.9, C1 and C2 = 2.05.

3. Multi-PSOs

After a number of tests on all types of PSOs, each one is evaluated separately on a different set of benchmark functions. The response of each algorithm on all selected benchmark functions is also observed, in terms of both speed response and accuracy in reaching to the closest correct solution. Therefore a new algorithm is generated containing more than one type of PSOs, where all PSOs algorithms share the same population which is updated every iteration via each algorithm.

The SPSO structure is built on all algorithms placed in serial, such that all algorithms address the same population individually one after another and process them through a specific number of iterations. Through the experimentation, it is found that to exchange the population between the algorithms can be advisable every 3 to 7 iterations. In this work, the number of internal repetitions is chosen equal to five iterations. This is established as a result of testing different settings for different problems which are used for comparison.

Each algorithm received the latest updated information such as p-best, g-best, the last velocity and position (v, x) for each particle from the previous algorithm, and the best results obtained will be set as an initial value for the current algorithm. This will be repeated for all the PSOs. The last algorithm will hand over its findings to the first algorithm to complete the cycle until the last iteration. In this case, the population will be the same size during the all iterations which select the best particles in each step. A flowchart of the SPSO algorithm is shown in Fig. 1.

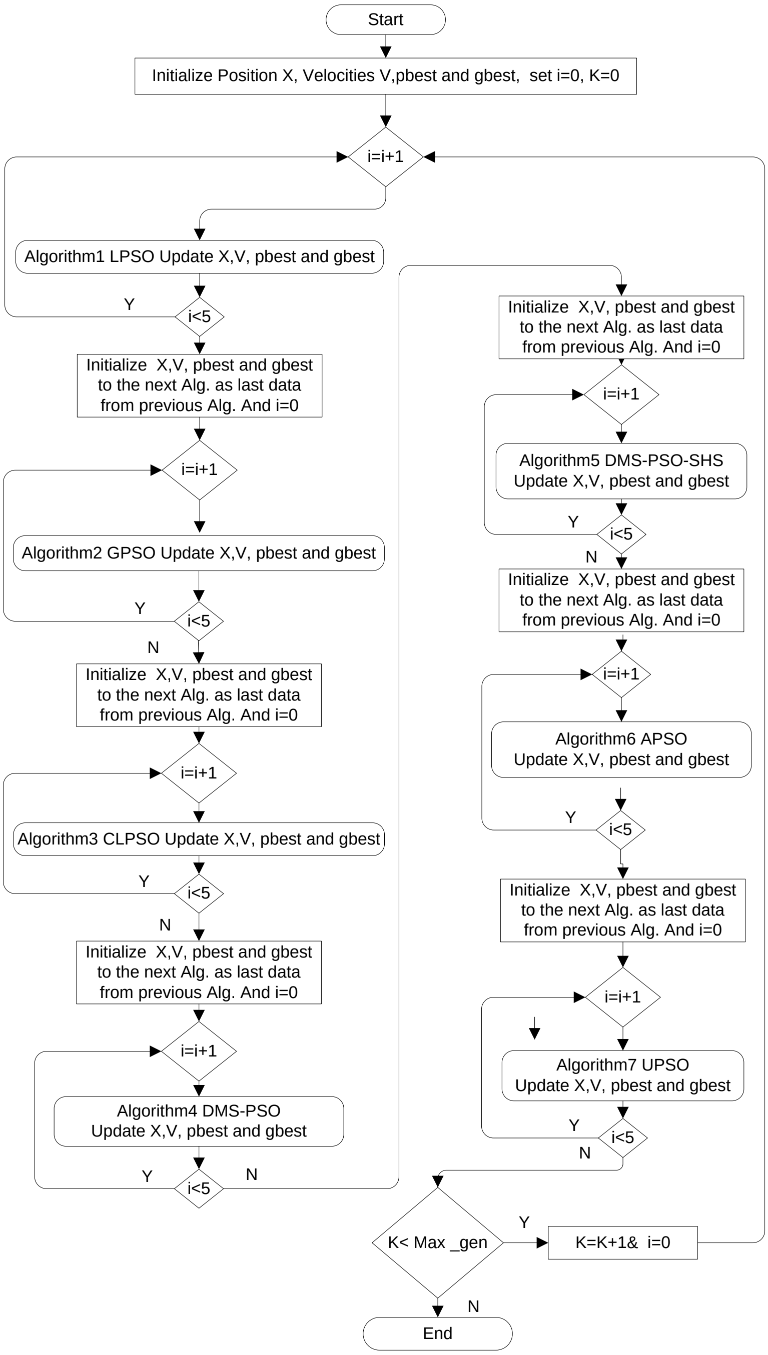

Fig. 1 Creative design methodology

4. Power System Design and Tuning

The Matlab Simulink Power System Toolbox is used to simulate a sample of electric power grid, which includes in this case two generators, one is with 5000 MW capacity and the other is with1000 MW capacity, both are turbine driven. These two generators mutually are connected to the grid via high voltage transformers and bas bar line with length of 700 km together with a load of 5000 MW as well as fault breaker, this breaker closes after 3.8 sec of the start of the simulation for a transition time of 0.1 sec. Two fuzzy logic controllers (FLC) are used to control each turbine as a replacement of a conventional MB stabilizer to improve the performance [20], [21] and [22] as shown in Fig. 2.

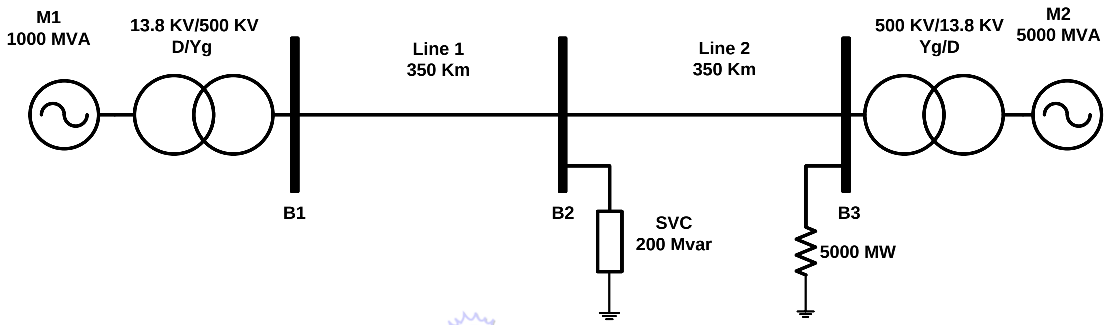

Fig. 2 Single line diagram of the simulated power system.

4.1. Controller design

The controller is designed in several stages using the Matlab Fuzzy Logic Toolbox. Mostly with FLC design, the first stage to choose the correct input signal. In this paper the generator speed deviation ($\Delta\omega$), and its derivative ($\Delta\acute{\omega}$) are two signals considered like two inputs for (FPSS) controller as shown in Fig. 3.

These two signals are used as rule-antecedent (IF-part) in the formation of rule base, and the output of controller (Δu) is used to represent the contents of the rule consequent (THEN-part) in performing of rule base [23], which is injected into the input of the excitation circuit controller.

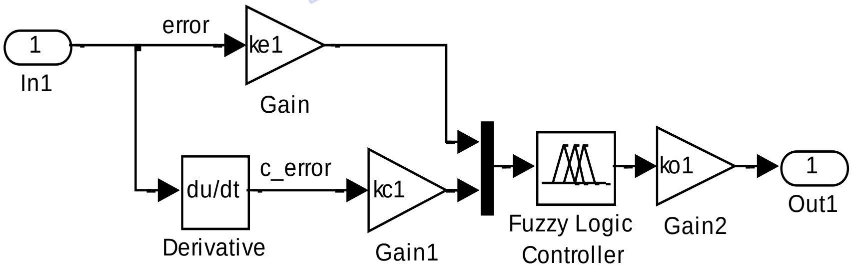

Fig. 3 Block diagram for FLC controller.

The Automatic Voltage Regulators (AVR) the generator terminal voltage by controlling the amount of current supplied to the generator field winding by the exciter. The general block diagram of the AVR subsystem is shown in Fig. 3. The measured element senses the current, power, terminal voltage and frequency of the generator. The measured generator terminal voltage V_g is compensated for the load current I_g and compared with the desired reference voltage V_{ref} to produce the voltage ΔV. This error is then amplified and used to alter the exciter output, and consequently the generator field current, so that the voltage error is eliminated. This represents a typical closed-loop control system. The regulation process is stabilized using a negative feedback loop taken directly from either the amplifier or the exciter as shown in Fig. 4. At this stage the controller is based on manly tuned

FLC architecture with two inputs as the error (Δω) and the change in error (Δώ), while the output (Δu) is directed to the excitation voltage loop driver then to the generator winding. Three membership functions are selected for each of the input variables, while the output is selected as a linear function since the inference engine used is a Takagi-Sugeno-Kang type TSK [24].

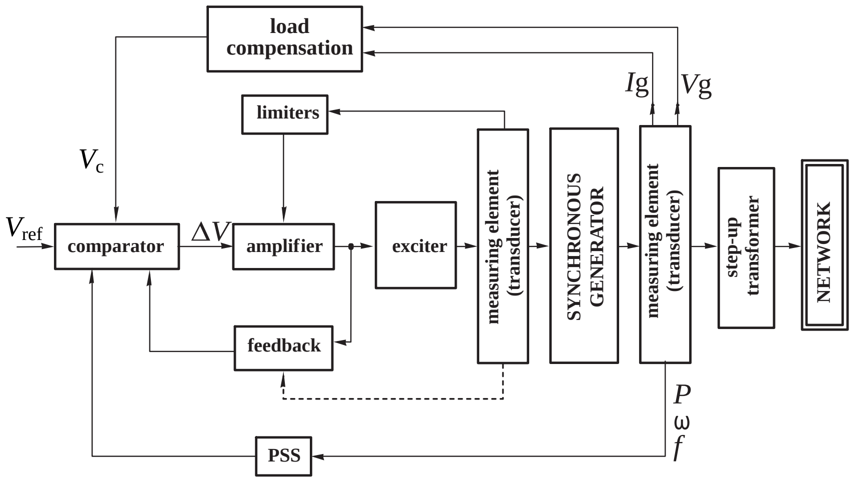

Fig. 4 Block diagram of excitation control system, PSS and power system stabilizer [25].

4.2. Manual tuning of scaling factors

In the proposed fuzzy power system stabilizer, FPSS is with two inputs and single output, which means three scaling factors are considered: the error (Ke), change of error (Kc) and output (Ko). To improve the FLC response, the FLC scaling factors are manually tuned. The scaling factors for the first generator FLC1 are Ke1 for the error gain, Kc1 is the gain for the change of error and Ko1 is the output gain. While the second generator FLC2 are Ke2 is the error gain, Kc2 is the gain for change of error and Ko2 is the output gain. The best values established for the scaling factors are as shown in Table 1.

Table 1 Manual Tuning of the FLC Scaling Factors.

	FLC1			FLC2		
Type	Ke1	Kc1	Ko1	Ke2	Kc2	Ko2
Manual Tuning	2	3.75	2.25	5	3.75	10

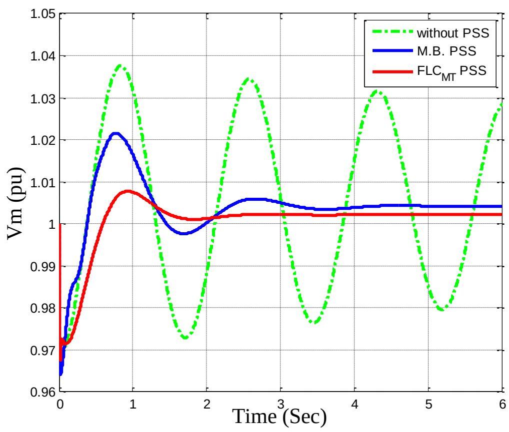

Fig. 5 System's response normal operation without PSS and with M.B and FLC PSS.

Sample result of the manually tuned FLCs in comparison to the M.B. stabilizer is shown in Fig. 5 which illustrates the grid power (Vm) in per unit (pu) of the whole system after Static Var Compensator (SVC) in normal operation and without PSS stabilizer. The system response of FLCs controllers is slightly better than the MB controller, whereas the system without PSS controllers becames unstable. The main reason for simulating this stage is to observe the effect of the proposed FLC controller on the system as a major step in the design stages. The main target is to focus on next stages which are dealing with training of FLC using the Adaptive Neuro-Fuzzy Inference System (ANFIS), and auto-tuning of the scaling factors using SPSO for both controllers (FLC1 & FLC2). In a later stage, a larger network contained four generators and four controllers (FLC1, FLC2, FLC3 &FLC4) works in the process at the same time.

4.3. FLC training

In order to increase the controller response quality, the FLC is trained using a learning signal from the MB stabilizer using the ANFIS architecture [8: 20]. The training is performed in two steps; simulation with disruption in the grid network by the occurrence of short circuit between one phase and the ground, the next fault is a short circuit occurrence between the three phases and the ground for a period of time of 0.1 msec as shown in Fig. 6. Both trained controllers are saved for each generator (FLC1 and FLC2). Experimentally, the FLCs trained based on the three-phase fault condition is better than the single-phase fault condition.

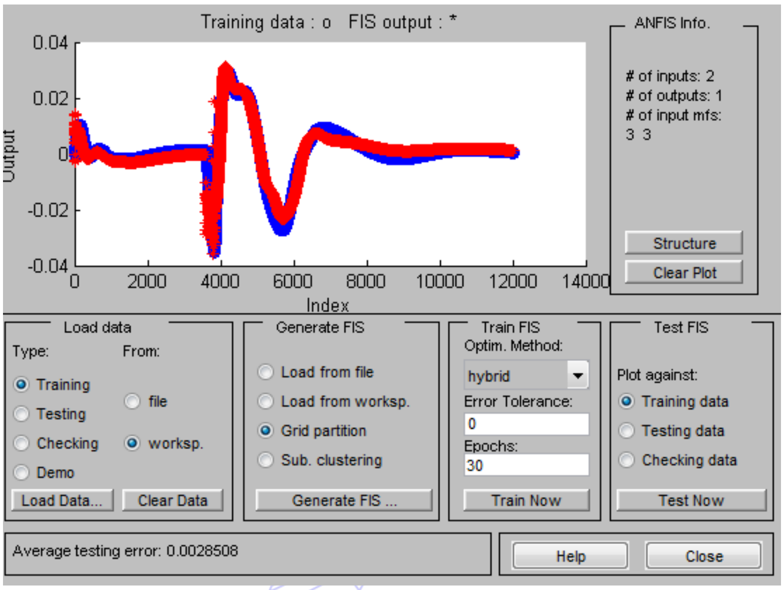

Fig. 6 FLC training in ANFIS editor.

The main reason for performing this stage is to observe the effect of the proposed FLC controller on the system as a major step in the design stages. But the main target is to focus on the next stages which are dealing with training of FLC using the ANFIS, and auto-tuning of the scaling factors using PSO for both controllers (FLC1 & FLC2) of the process at the same time.

4.4. Auto-Tuning FLC

In this stage the FLC scaling factors are selected using the SPSO optimizer [26] which is aimed at improving the response of both controllers keeping in mind the main objectives. The objectives are dependent on the type and requirements of the system to be controlled. For the power system, the objective is to minimize the following four variables on the final output (Vm) after the SVC of whole system:-

- Minimizing the settling time.
- Minimizing steady state error.
- Minimizing the overshoot
- Minimizing the first negative peak

The objective function is set to minimize the weight average of the four variables as follows:

$$\mathrm{MinW_{Averge}} = \sum \left[\frac{|\mathrm{Norm_{De}}*15|+|\mathrm{Norm_{OV}}*20|+|\mathrm{Norm_{p}}*20|+|\mathrm{Norm_{Er}}*45|}{100}\right] \tag{1}$$

where

Norm_De is the normalization of time delay value.

Norm_ov is the normalization of overshooting value.

Norm_P is the normalization of first -peak value.

Norm_Er is the normalization of steady stat error value.

4.5. Auto tuning of scaling factors

The controller is tuned using SPSO to select the scaling factors. The optimizer's objective function is based on three objectives: steady state error, settling time, overshoot, and the negative peak. The first and second objectives are with the highest priority, while the third is with the next priority, whereas the fourth objective is with the least priority [27]. The aim is to reduce the settling time first, then reduce the overshoot, and the last objective is to reduce the first negative peak of oscillation. Simulation result of the multi-band and the auto-tuned FLC for different types of faults are shown in Table 2. The table lists the settling time, overshoot and fluctuation.

Table 2 Simulation Results for Multi Fault with Auto-Tune and M-Training.

	Multi Band Controller (MB)			Fuzzy Logic Controller (FLC_{AT})		
	Settling time (sec)	Overshoot (%)	Fluctuation	Settling time (sec)	Overshoot (%)	Fluctuation
1 phase fault	2.6	5.24	4 peaks	1.4	3.45	2 peaks
2 phase fault	2.9	8	4 peaks	1.6	5.5	2 peaks
3 phase fault	3	11.4	4 peaks	1.9	5.1	2 peaks

5. Scaled-Up System Results

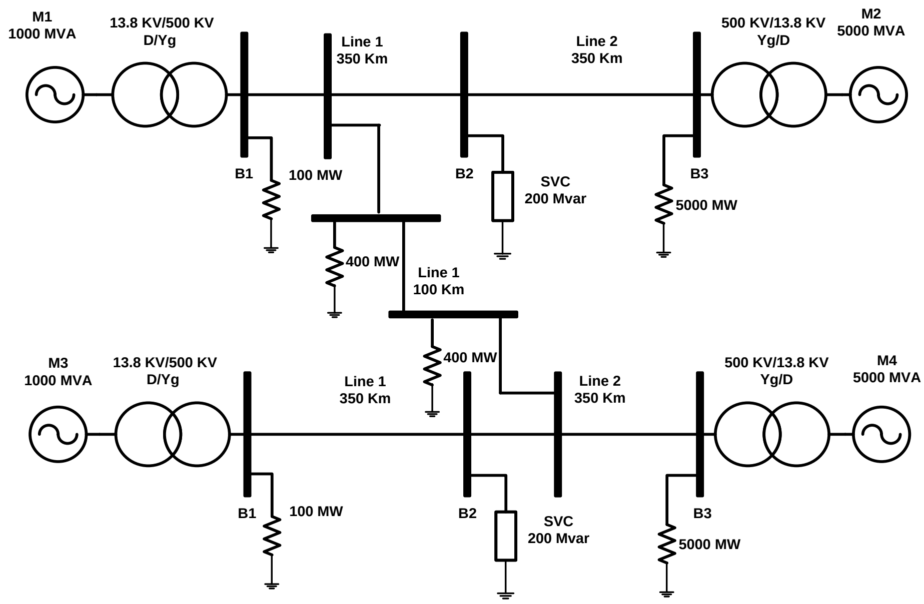

Fig. 7 Single line diagram of the scaled-up power system.

For the purpose of testing the efficiency of the new designed system, the system is upgraded to a larger electrical network grid, using MATLAB Simulink Power System Tools, which include in this case four generators, two with a capacity of 5000 MW, and the other two with a capacity of 1000 MW, all the generators are turbine driven. The four generators mutually connected to the network via high voltage transformer, bas bar and transposed line with length 1,500 km together with a load of 11,000 MW as well as two fault breakers. One of the breakers closes after 4.8 sec of the start of the simulation for a transition time period of 0.1 sec. Four neuro-fuzzy logic controllers are used to stabilize each turbine. The FLCs replace the conventional MB stabilizer to improve stability as shown in Fig. 7.

Same as before the controller is tuned using SPSO to optimize the selection of the scaling factors. The optimizer's objective function is based on three objectives: steady state error, settling time, overshoot, and the negative peak. The first and second objectives are with the highest priority, while the third is with medium priority, whereas the fourth objective is with the least priority.

The optimized values of the scaling factors for both, the auto-tuned and the manual tuned fuzzy logic controller are listed in Table 3. The FLCs are trained on data generated using 3-phase fault conditions which dubbed as 3-phase training. Four FLCs controllers are trained for the generator. The scaling factors are tuned automatically for each controller using SPSO optimizer.

The system is tested and simulated for different fault conditions, namely, single and multi-phase conditions. Simulation results show that controllers are performing well for single-phase fault, two and three-phase in comparison to the performance when the system is driven without PSS. Figs. 8, 9 and 10 show the response of the system to the faults when driven without PSS, MB stabilizer and auto-tuned FLCs respectively.

Table 3 Final Auto-Tuning Scaling Factor Values.

	FLC1			FLC2		
	Ke1	Kc1	Ko1	Ke2	Kc2	Ko2
Manual Tuning	2	3.75	2.25	5	3.75	10
Auto. Tuning	1.100	1.948	6.559	2.096	27.57	5.666
	FLC3			FLC4		
Manual Tuning	2	3.75	2.25	5	3.75	10
Auto. Tuning	1.139	0.941	9.215	1.556	6.006	5.543

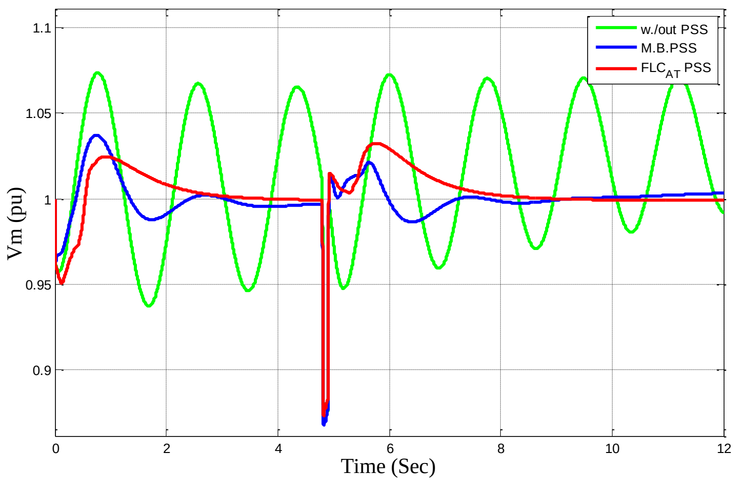

Fig. 8 System's response to 1-phase fault with 3-phase training and Auto-tuning.

Fig. 8 shows the response of the FLCs controller with 3-phase training and auto-tuning scaling factors to four controllers. The first test is conducted for one phase fault, and assumes that the settling time for the system at ±1.2 %.

The fault starts at 4.8 sec and ends at 4.9 sec, the FLCs controllers reduce the stability time from 3.77 sec to 2.5 sec, moreover, with two peaks fluctuation compared to the MB controllers with six peaks and larger overshoot such as FLCs is with 3.2% while MB is with 3.7%. In Fig. 9, the FLCs reduce the stability time from 4.03 sec to 2.43 sec, with less fluctuation from six to two compared to MB controllers. More importantly, when a three-phase fault is simulation as shown in Fig. 10, the FLCs react with high efficiency compared to the MB and reduce the overshoot from 4.9% to 4.2% and stability period from 3.41 sec to 2.5 sec with respect to the MB controller also in long period started appearance of increasing on steady state error. The numerical results are shown in Table 4.

Table 4 Simulation Results for Multi Fault with Auto-Tune and M-Training.

	Multi Band Controller (MB)			Fuzzy Logic Controller (FLC_{AT})		
	Settling time (sec)	Overshoot (%)	Fluctuation	Settling time (sec)	Overshoot (%)	Fluctuation
1 phase fault	3.77	3.7	6	2.5	3.2	2
2 phase fault	4.03	3.7	6	2.43	4.4	2
3 phase fault	3.41	4.9	6	2.5	4.2	2

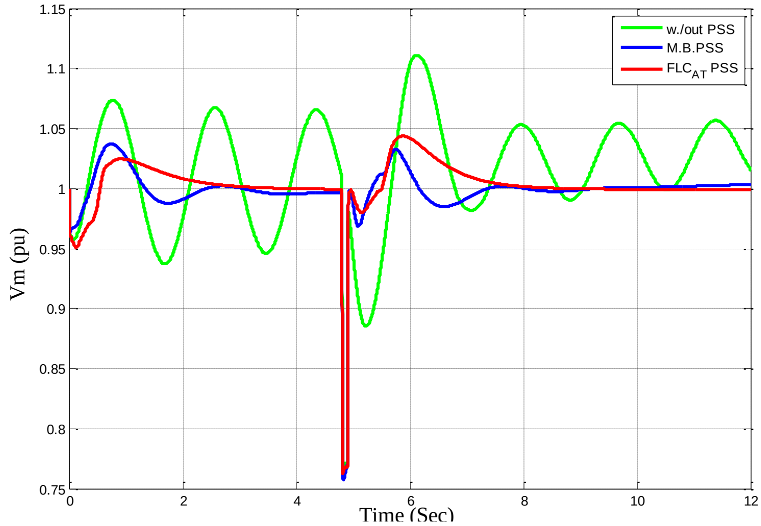

Fig. 9 System's response to 2-phase fault with 3-phase training and Auto-tuning.

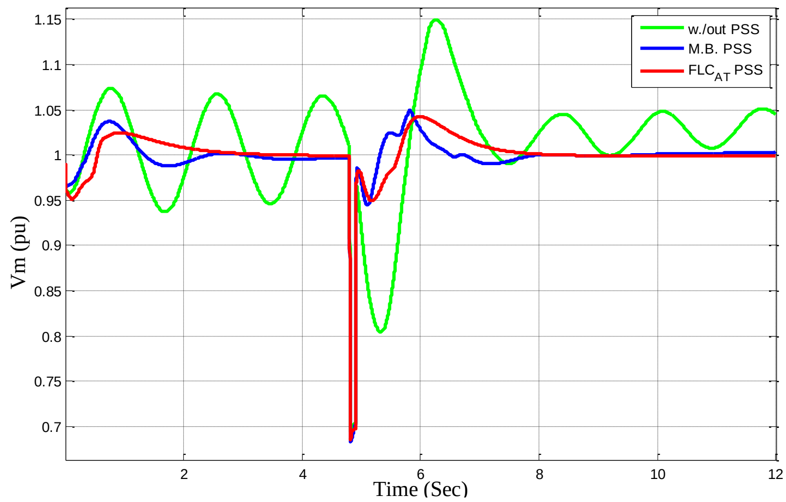

Fig. 10 System's response to 3-phase fault with 3-phase training and Auto-tuning.

6. Conclusions

The work presented in this paper demonstrates that FLC is better compared to the MB controller, in particular when the FLC is trained on 3-phase fault conditions with the aid of SPSO for the selection of scaling factors for each controller automatically. However, single-phase faults are also produce good training data which the controllers perform well. This type of control is far better than the standard controller which needs to be tuned for each type of fault.

References

[1] V. Rupal, H. A. Patel, and A. Mehta "Novel approach for designing a power system stabilizer," National Conference on Recent Trends in Engineering & Technology, May 2011.
[2] K. D. Koper, M. E. Wysession and D. A. Wiens, "Multimodal function optimization with a niching genetic algorithm: A seismological example," Bulletin of the Seismological Society of America, vol. 89, pp. 978-988, August 01, 1999.
[3] K. S. Hook, Y. Liu, and S. Atcitty, "Mitigation of the wind generation integration related power quality issues by energy storage," EPQU J., vol. XII, no. 2, 2006.
[4] F. W. Keay and W. H. South, "Design of power system stabilizer sensing frequency deviation," IEEE Transactions on Power Apparatus and Systems, vol. 90, pp.707- 713, March 1971.
[5] C. J. Wu and Y. Y. Hsu, "Design of self-tuning PID power system stabilizer for multimachine Power System," IEEE Transactions on Power Systems, vol. 3, pp. 1059-1064, Aug. 1998.
[6] C. C. Lee, "Fuzzy logic in control systems—Part I and II," IEEE Trans.Syst., Man, Cybern., vol. 20, pp. 404-435, Mar./Apr. 1990.
[7] K. A. El-Metwally and O. P. Malik, "Fuzzy logic based power system stabilizer," IEEE Proc- Gener.Transm. Distri., vol. 142, pp.277-281, May 1995.
[8] A. Afzalian, and D. A. Linkens, "Training of neurofuzzy power system stabilisers Using Genetic Algorithms," International-Journal-of-Electrical-Power-&-Energy-Systems, vol. 22, no. 2, pp. 93-102, 2000.
[9] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Neural Networks, 1995, Proceedings, IEEE International Conference on, vol.4, pp. 1942-1948, 1995.
[10] D. Bratton and J. Kennedy, "Defining a standard for particle swarm optimization," in Swarm Intelligence Symposium, 2007, SIS 2007, IEEE, pp. 120-127, 2007.
[11] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in ACM SIGGRAPH Computer Graphics, pp. 25-34, 1987.
[12] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in Micro Machine and Human Science, 1995, MHS '95, Proceedings of the Sixth International Symposium on, pp. 39-43, 1995.
[13] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in Systems, Man, and Cybernetics, 1997, Computational Cybernetics and Simulation, 1997 IEEE International Conference on, pp. 4104-4108, vol. 5, 1997.
[14] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in Evolutionary Computation Proceedings, 1998, IEEE World Congress on Computational Intelligence, the 1998 IEEE International Conference on, pp. 69-73, 1998.
[15] J. J. Liang, A. K. Qin, P. N. Suganthan and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," Evolutionary Computation, IEEE Transactions on, vol. 10, pp. 281-295, 2006.
[16] J. J. Liang and P. N. Suganthan, "Dynamic multi-swarm particle swarm optimizer," in Swarm Intelligence Symposium, 2005, SIS 2005, Proceedings 2005 IEEE, pp. 124-129, 2005.
[17] S-Z. Zhao, P. N. Suganthan and S. Das, "Dynamic multi-swarm particle swarm optimizer with sub-regional harmony search," in Evolutionary Computation (CEC), 2010 IEEE Congress on, pp. 1-8, 2010.
[18] Z-H. Zhan, Jun Zhang, Yun Li and H-S. Chung, "Adaptive particle swarm optimization," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 39, pp. 1362-1381, 2009.
[19] K. E. Parsopoulos and M. N. Vrahatis, "Unified particle swarm optimization for tackling operations research problems," in Swarm Intelligence Symposium, 2005, SIS 2005, Proceedings 2005 IEEE, pp. 53-59, 2005.
[20] I. Kamwa, R. Grondin, and G. Trudel, "IEEE PSS2B versus PSS4B: the limits of performance of modern power system stabilizers," IEEE Trans. Power Systems, vol. 20, pp. 903-914, May 2005.

[21] H. P. Inamdar and R. P. Hasabe, “It based energy management through demand side in the industrial sector,” INCACEC International Conference on Control, Automation, Communication and Energy Conservation, pp. 1-7, 2009.

[22] A. T. Ali, E. B. M. Tayeb and M. K. A. Adam, "A multi-machine power system stabilizer using fuzzy logic controller," International Journal of Computational Engineering Research, vol. 2, Issue. 6, Oct 2012.

[23] H. Bevrani, and P. Ranjbar, “Fuzzy logic-based load-frequency control concerning high penetration of wind turbines,” IEEE Systems Journal, 2011.

[24] T. P. Hong and C. Y. Lee, “Induction of fuzzy rules and membership functions from training examples,” Fuzzy Sets and Systems, vol. 84, pp. 33-47, Nov. 1996.

[25] J. Machowski, J. Bialek, and J. Bumby. Power system dynamics: stability and control, John Wiley & Sons, 2011.

[26] Z-H. Zhan, J. Zhang, Y. Li, and H. S-H. Chung, “Adaptive particle swarm optimization,” IEEE Transactions on Systems Man, and Cybernetics. Part B: Cybernetics, vol. 39, no. 6, pp. 1362-1381, 2009.

[27] H. Jiang, J. Zheng, L. Chen, “A particle swarm algorithm for multiobjective optimization problem,” Journal of Pattern Recognition and Artificial Intelligence, vol. 20, pp. 606-611, 2007.