Multi-machine Power System Stabilizer Design Based on Population Based Incremental Learning

Dereck A. Dombo, *Student Member, IEEE*, Komla A. Folly, *Senior Member, IEEE*
Department of Electrical Engineering
University of Cape Town
Cape Town, South Africa
Dereck.Dombo@alumni.uct.ac.za, Komla.Folly@uct.ac.za

***Abstract*— Population-Based Incremental Learning (PBIL) is one of the Evolutionary Algorithms that has received increasing attention in recent years due to its effectiveness. However, recent studies have shown that PBIL with fixed learning rate may suffer from loss of diversity which can lead to premature convergence. In this paper, PBIL with adaptive learning rate (APBIL) is used to overcome the issues of premature convergence. Frequency and time domain simulation results are presented to show the effectiveness of the APBIL algorithm.**

I. Introduction

With the increase in power system interconnections, the problem of low frequency oscillations has become a great concern. These oscillations ranging from 0.2 to 3 Hz can restrict the power transfer and sometimes affect the security of the power system network [1]-[4]. Over the years, different devices such as Static Var Compensators, Power System Stabilizers (PSSs) have been used to mitigate low frequency oscillations [2], [5], [6]. Power System Stabilizers are the most cost effective devices that can be used to damp low frequency oscillations [2], [6].

Conventional PSSs (CPSS) are traditionally designed using classical control techniques such as eigenvalue analysis, phase compensation, pole placement, root locus, etc. [3]. However, CPSSs based on the classical control techniques cannot guarantee the stability of a power system network over a wide range of operating conditions. To robustly select the parameters of the PSS, Evolutionary Algorithms (EAs) such as Genetic Algorithms (GAs), Differential Evolution (DE), have been proposed [2], [5]. Some of the issues with these algorithms are related to the optimal selection of genetic operators such as crossover, mutation, amplification factor, etc. Recently, Population-Based Incremental Learning (PBIL), has received increasing attention because of its effectiveness and easy implementation. PBIL is a technique that combines Genetic Algorithms (GAs) and simple Competitive Learning. In PBIL, the entire genetic population is represented through a probability vector rather than a myriad of chromosomes. Unlike GAs, there is no crossover operator in PBIL. PBIL does not maintain a population of individuals, but instead works with a probability vector (PV) [7]-[8]. During the search, the values in the probability vector are updated to represent those in high evaluation vectors. The probability update rule is similar to the weight update rule in a competitive learning network [7]. At each generation, the PV is used to sample new individuals through learning. This makes PBIL simpler and more efficient than GAs [7]-[20]. In general, a fixed learning rate is used in the update rule of the standard PBIL. The standard PBIL with fixed learning rate has been successfully applied to engineering optimization [13] and power system controller design, where it was shown that PBIL-PSS outperforms GA-PSS and the CPSS [9]-[12]. However, some authors [14], [15] have suggested that PBIL with fixed learning rate may suffer from loss of diversity which may lead to premature convergence. To overcome this problem, dual PBIL and adaptive learning rate (APBIL) were proposed in [15] and [16], respectively.

This paper extends previous work [17]-[22], whereby PBIL with adaptive learning (APBIL) is applied to PSS design for a multi-machine power system. Frequency and time domain simulations are presented to show the effectiveness of the APBIL algorithm. The results show that the APBIL-PSSs provide better damping to the system than the standard PBIL-PSSs, albeit having slightly higher overshoots and undershoots during the first swing.

II. Overview of PBIL

Over the years, Evolutionary Algorithms (EAs) have been increasingly applied in solving various optimization problems. Some of the EAs are Genetic Algorithms [5], Differential Evolution [6], Estimation of Distribution Algorithms to which PBIL belongs [7]-[15] to name only a few. PBIL was first proposed by Shumeet Baluja in 1994 [7] and later developed in 1995 by Baluja and Caruana, and Green [8], [13]. PBIL is a combination of Genetic Algorithms and Competitive Learning for function optimization [7]-[15]. It uses a probability vector (PV) to generate sample solutions. At each generation, the PV is used to sample new individuals through learning. Learning consists of using the best solution to update the probability vector by pushing it towards generating good individuals in the population [7]-[15]. The probability update rule is similar to the weight update rule in a competitive learning network [7], [8]. At the beginning, the

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.183

values of the probability vector are set to 0.5 so that the probability of generating either 0 or 1 is equal [16]-[22]. Just like in GA, the mutation operator is used to maintain diversity in the trial solutions [18]-[20]. In this paper, a forgetting factor is used to relax the probability vector towards a neutral value of 0.5 [18].

It should be mentioned that the PV is a function of the learning rate (LR) which is responsible for determining the speed at which the PV is shifted towards the best solution vector [20].

The summary of the PBIL used in this paper is given below [7]-[20]:

- Step 1: Initialization of the probability vector (PV) so as to ensure uniformly-random bit strings.
- Step 2: Generation of a population of uniformly-random bit strings and comparing them element by element with the PV. Wherever an element of the PV is greater than the corresponding random element, a ‘1’ is generated, otherwise a ‘0’ is generated.
- Step 3: Interpretation of each bit string as a solution to the problem and evaluation of its merit in order to identify the “Best”.
- Step 4: Adjustment of the PV by slightly increasing *PV(i)* to favor the generation of bit strings which resemble “Best”, if Best*(i)* = 1 and decrease *PV(i)* if Best*(i)* = 0.
- Step 5: Generation of a new population reflecting the modified distribution. Terminate if a satisfactory solution is found or else go to step 3.

If the LR is set too high, the algorithm will not be able to explore the whole search space thoroughly and hence it suffers premature convergence. On the other hand, if the LR is too small, the algorithm may require more time to find the optimal solution, which may be time consuming. Therefore, there is a need to have a trade-off between exploration and exploitation, hence the research on the effect of learning rate is still ongoing [16]-[22]. In the next section, we will discuss the PBIL with adaptive learning rate.

III. Overview of APBIL

As discussed previously, in the standard PBIL, the learning rate is fixed and this limits the capability of the algorithm. Another issue is that a lot of time has to be spent on choosing the most suitable LR, although there is no guarantee that it will be the optimal value to use. If the search space environment is ever changing, which is the case in powers systems, a fixed LR may not be adequate in obtaining the optimal control parameters [18].

Unlike the standard PBIL, the APBIL attempts to change the learning rate during the optimization process. In developing the APBIL, it was assumed that diversity is needed at the start of the optimization process so that the algorithm is able to thoroughly explore the search space. Taking this into consideration, the value of the learning rate is chosen such that it is very small at the beginning of the run. This value is varied as the optimization process is carried out and increases according to the following equation:

$$LR(i) = LR_{max}\frac{G(i)}{G_{max}} \tag{1}$$

where

$LR(i)$ is the learning rate at the *ith* generation
LR_{max} is the final learning rate
$G(i)$ is the *ith* generation
G_{max} is the maximum generation

The optimization process is terminated after specific number of iterations (generations). In this case, the maximum generation was set to 400. Although the above equation is very simple, it has shown to improve the performance of the algorithm.

IV. Power System Model and Operating Conditions

The power system model used in this paper is the two-area, four-machine system which is shown in Fig. 1. In each of the two areas, there are two generators with ratings of 900MVA and 22kV each. For small signal stability analysis purposes, all the machines are modelled using sixth order machine model (e.g., sub-transient model). Each machine is equipped with a simple exciter modelled by a first order differential equation [3], [19]-[22].

In carrying out the investigations, several operating conditions were considered. However, for simplicity, only three operating conditions are discussed here as listed in Table I.

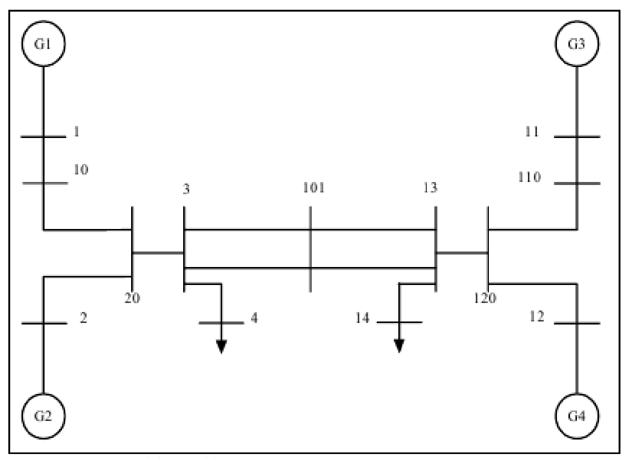

Fig. 1 Two area multi-machine system [22]

The operating conditions involve power transfer which varies between 100MW to 300MW from area 1 to area 2. These operating conditions were obtained by varying the loads at buses 4 and 14.

Table I: Selected Operating Conditions

Case	Real Power Flow[MW]
1	100
2	200
3	300

V. Problem Formulation

The system under study has three fundamental oscillatory modes. There are two local area modes, one in area 1 and the other in area 2. The local mode in area 1 is caused by generator 1 oscillating against generator 2 and the same is true for area 2, where generator 3 is oscillating against generator 4. The third mode is the inter-area mode which is as a result of the generators in area 1 oscillating against the ones in area 2. Since the inter-area modes are the most difficult to control, only these modes will be discussed here. The inter-area modes are listed in Table II for the three operating conditions considered.

Table II: Open-loop poles inter-area oscillations

Case	Inter-area mode
1	-0.007±j4.053(0.00173)
2	0.024±j3.964(-0.00605)
3	0.048±j3.791(-0.0127)

Note: damping ratios are in brackets

In this paper, APBIL and PBIL are used to optimize the parameters of the Power System Stabilizer (PSS) such that adequate damping is provided to the system for pre-specified operating conditions. The objective function considered is given below as follows:

$$J = \max(\min(\zeta_{i,j})) \tag{2}$$

where:

$\zeta_{i,j}$ is the damping ratio of the *i-th* eigenvalue corresponding to the *j-th* operating condition as given below

$$\zeta_{i,j} = \frac{-\sigma_{i,j}}{\sqrt{\sigma_{i,j}^2+\omega_{i,j}^2}}$$

i =1, 2, 3… n

j = 1, 2, 3 …m

where, the eigenvalues are made up of $\sigma_{i,j}$ and $\omega_{i,j}$ which represent the real part and the imaginary part (frequency of oscillation), respectively.

VI. PSS Design

A. PSS Structure

The structure of the PSS which parameters are to be optimized is as follows.

$$K(s) = K_p\left(\frac{sT_w}{1+sT_w}\right)\left(\frac{1+sT_1}{1+sT_2}\right)\left(\frac{1+sT_3}{1+sT_4}\right) \tag{3}$$

where K_p is the gain which is responsible for providing damping to the system. T_w is the washout time constant, which in this paper is set to 10sec. T_1 - T_4 are the time constants of the lead lag compensators.

It should be mentioned that the two controllers in area 1 have the same PSS parameters and the ones in area 2 also have the same parameters.

B. Application of PBIL to PSS Design

Parameter settings of PBIL are as follows:

Population: 100
Generations: 400
Learning rate: Fixed at 0.2
Mutation (Forgetting Factor): 0.005

C. Application of APBIL to PSS Design

Parameters settings of the APBIL are as follows:

Population: 100
Generations: 400
Learning rate: Adaptive, Varying from: 0.0005 to 0.2
Mutation (Forgetting Factor): 0.005

The objective function J is maximized subject to the following constraints:

$$0 \le K_p \le 30$$
$$0 \le T_1, T_3 \le 1$$
$$0.01 \le T_2, T_4 \le 0.3$$

The optimized PSS parameters for both PBIL and APBIL are listed in Table III.

Table III: Optimized PSS parameters

PSSs	Gen	K_p	T_1	T_2	T_3	T_4
PBIL	1 & 2	15.63	0.188	0.019	0.114	0.016
	3 & 4	15.79	0.05	0.016	0.057	0.017
APBIL	1 & 2	15.72	0.10	0.025	0.06	0.01
	3 & 4	18.52	0.029	0.013	0.083	0.0164

Note: Gen is short for generator.

VII. Simulation Results

A. Time Domain Results

The following figures Fig. 2 - Fig. 7 show the time domain simulations that were carried out by applying a 10% step disturbance to the reference voltages of the generators. Change in generator active power responses are plotted for different operating conditions as shown in Table I. Only the results for generator 1 (G1) in area 1 and generator 3 (G3) in area 2 are shown in this paper.

Fig. 2 - Fig. 4 show the responses of PBIL-PSS and APBIL-PSS for G1. It can be seen that both controllers are able to damp the oscillations and improve the stability of the system under all the operating conditions considered. However, APBIL-PSS provides a better damping with quicker settling time than the PBIL-PSS. The settling time of APBIL is 3sec. compared to 5sec. for PBIL-PSS. APBIL-PSS also gives slightly higher overshoots and undershoots than PBIL-PSS during the first swing. This needs to be further investigated.

It can be seen from Fig. 5 - Fig. 7 (plots for G3) that in terms of damping, APBIL-PSS outperforms PBIL-PSS with shorter settling time as in the previous Figures.

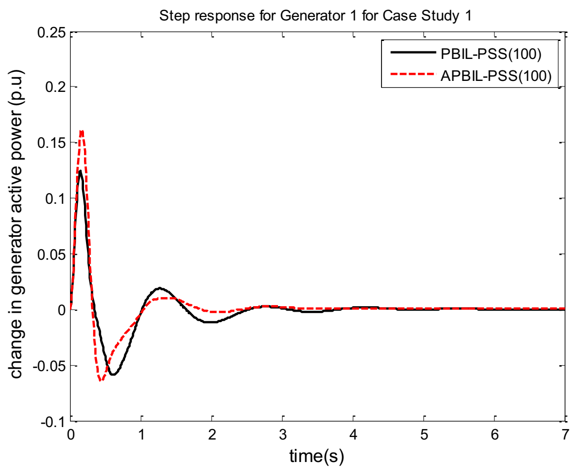

Fig. 2 Change in active power of G1 for case 1

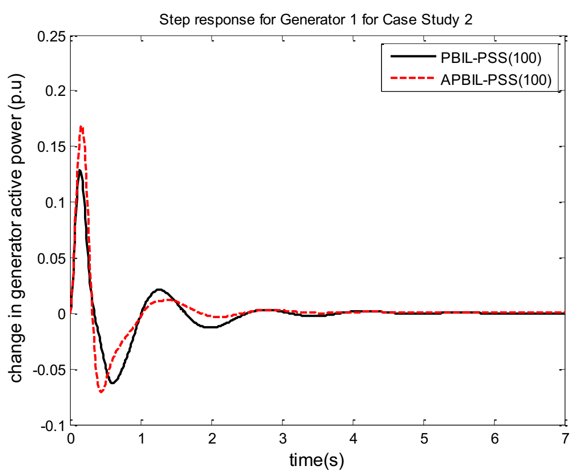

Fig. 3 Change in active power of G1 for case 2

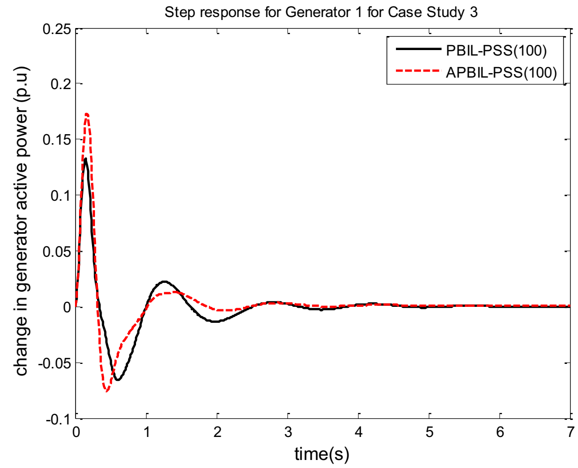

Fig. 4 Change in active power of G1 for case 3

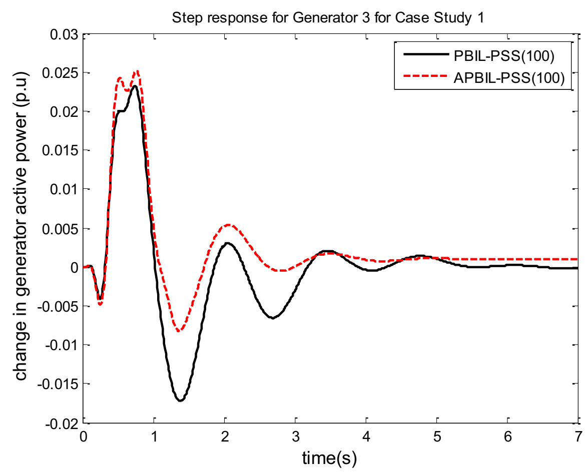

Fig. 5 Change in active power of G3 for case 1

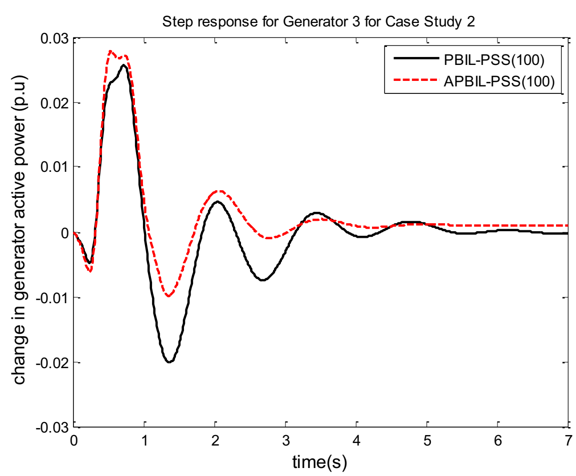

Fig. 6 Change in active power of G3 for case 2

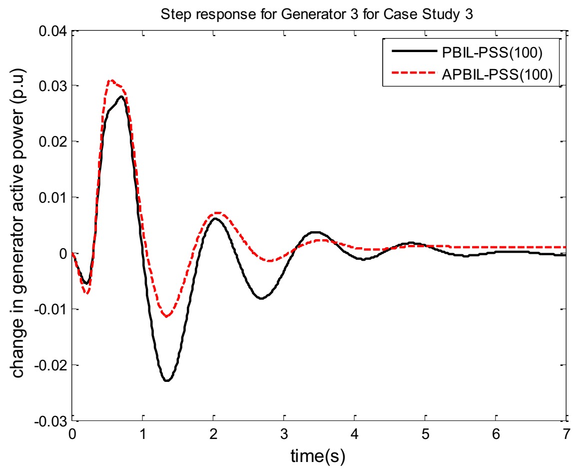

Fig. 7 Change in active power of G3 for case 3

B. *Frequency Domain Results*

Table IV shows the eigenvalues of the closed-loop system together with their damping ratios for the inter-area mode. Case 1 in Table II shows that when the system is equipped with PBIL-PSSs and APBIL-PSSs, the system's damping is improved. However, APBIL displays a better performance having a higher damping ratio of 0.276 compared to 0.191 for PBIL. For cases 2- 3, it can be observed that when the system is equipped with PBIL-PSS and APBIL-PSS, the inter-area modes became stable. However, APBIL-PSSs provide better damping than PBIL-PSSs. In all cases, APBIL-PSSs outperformed the PBIL-PSSs.

In the simulations, we do not include the conventional PSS and the Genetic Algorithms based PSS (GA-PSSs). This is because we have shown previously that PBIL with fixed learning rate outperforms GA and the CPSS [9]-[12].

Table IV: Closed loop Inter-area modes

Case	PBIL	APBIL
1	-0.88±4.53i (0.191)	-1.25±4.36i (0.276)
2	-0.83±4.51i (0.181)	-1.19±4.34i (0.265)
3	-0.78±4.46i (0.172)	-1.12±4.28i (0.254)

Note: damping ratios are in brackets

VIII. CONCLUSION

Robust PSS parameters optimization using APBIL and PBIL has been investigated. The simulation results show that by varying the learning rate of PBIL, a better performance can be obtained. Time domain simulation results show that APBIL-PSSs outperform PBIL-PSSs albeit with slightly higher overshoots and undershoots during the first swing. These results are confirmed by frequency domain results. Further work is being carried out to improve the performance of APBIL.

REFERENCES

[1] R. You and M. H. Nehrir, "A systematic approach to controller design for SVC to enhance damping of power system oscillations," IEEE PES Power Systems Conference and Exposition, vol. 2, pp. 1134-1139, 2004.

[2] P. Zhang and A. H. Coonick, "Coordinated synthesis of PSS parameters in multi-machine power systems using the method of inequalities applied to genetic algorithms," IEEE Transactions on Power Systems, vol. 15, no. 2, pp. 811-816, 2000.

[3] P. Kundur, Power System Stability and Control, McGraw-Hill, 1994.

[4] F. P. Demello and C. Concordia, "Concepts of synchronous machine stability as affected by excitation control," IEEE Transactions on Power Apparatus and Systems, Vols. PAS-88, no. 4, pp. 316-329, 1969.

[5] S. Sheetekela and K. Folly, "Multimachine power system stabilizer design based on evolutionary algorithm," Proceeding of The 44th International Universities' Power Engineering Conference, 2009.

[6] T. Mulumba and K. A. Folly, "Application of self-adaptive differential evolution to tuning PSS parameters," Proc. of Power Engineering Society Conference and Exposition in Africa (Power Africa), 2012 IEEE, 2012, pp. 1-5.

[7] S. Baluja, "Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning," Technical Report Carnegie Mellon University, CMU-CS-49-163, June 2, 1994.

[8] S. Baluja and R. Caruana, "Removing the genetics from the standard genetic algorithm," in Proc 12th International Conference on Machine Learning, Lake Tahoe, CA, July 1995

[9] K. A. Folly, "Design of power system stabilizer: a comparison between genetic algorithms (GAs) and population-based incremental learning (PBIL)," in Power Engineering Society General Meeting, 2006. IEEE, 2006.

[10] K. A. Folly, "Robust controller design based on a combination of genetic algorithms and competitive learning," in International Joint Conference on Neural Networks, 2007. IJCNN, 2007, pp. 3045-3050.

[11] K. A. Folly and G. K. Venayagamoorthy, "A Real-time implementation of a PBIL based stabilizing controller for synchronous generator," in Industry Applications Society Annual Meeting, 2009. IAS 2009. IEEE, 2009, pp. 1-8.

[12] J. P. Sheetekela, K. A. Folly and O. P. Malik, "Design and implementation of power system stabilizers based on Evolutionary Algorithms," in Africon, 2009. Africon'09., 2009, pp. 1-6.

[13] J. R. Green, "Population-based incremental learning as a simple, versatile tool for engineering optimization,"in

Proceedings of the First International Conference on Evolutionary Computation and Its Applications, 1996.

[14] C. Gonzalez, J. A. Lozano and P. Larranaga, "Analyzing the population-based incremental learning algorithm by means of discrete dynamical systems," Intelligent Systems Group, Department of Computer Science and Artificial Intelligence, University of the Basque Country, Spain, 2000.

[15] S. Yang and X. Yao, "Dual population-based incremental learning for problem optimization in dynamic environments," in Proc. of the 7th Asia Pacific Symposium on Intelligent and Evolutionary Systems, pp. 49-56, 2003.

[16] K. A. Folly and G. K. Venayagamoorthy, "Effects of learning rate on the performance of the population based incremental learning algorithm," International Joint Conference on Neural Networks , pp. 861-868, 2009.

[17] K. A. Folly and S. P. Sheetekela, "Application of a simple estimation of distribution algorithm to power system controller design," in Universities Power Engineering Conference (UPEC), 2010 45th International, 2010, pp. 1-6.

[18] K. A. Folly and G. K. Venayagamoorthy, "Performance evaluation of a PBIL based power system damping controller," in Proc. of the IEEE Congress on Evolutionary Computation (CEC), pp. 1-7, 2010.

[19] K. A. Folly and G. K. Venayagamoorthy, "Optimization of power system stabilizer parameters using population-based incremental learning," in 2012 IEEE, Energy Conversion Congress and Exposition (ECCE), 2012, pp. 1468-1474.

[20] K. A. Folly, "Population-based incremental learning with adaptive learning rate strategy," in Advances in Swarm Intelligence, Springer Berlin Heidelberg, 2012, pp. 11-20.

[21] K. A. Folly, "Parallel PBIL applied to power system controller design," Journal of Artificial Intelligence and Soft Computing Research, vol. 3, no. 3, pp. 215-223, 2013.

[22] K. A. Folly, "Comparison of multi-population PBIL and adaptive learning rate PBIL in designing power system controller," in Advances in Swarm Intelligence, Hefei,, Springer International Publishing, 2014, pp. 135-145.