Multimachine Power System Stabilizer Design Based on Evolutionary Algorithm

Severus Sheetekela
Department of Electrical Engineering
University of Cape Town
Cape Town, South Africa
spsheetekela@gmail.com

Komla Folly
Department of Electrical Engineering
University of Cape Town
Cape Town, South Africa
komla.folly@uct.ac.za

***Abstract*- This paper discusses the design of multimachine power system stabilizers based on three evolutionary algorithm techniques, namely: Genetic Algorithm (GA), Population Based Incremental Learning (PBIL) and the Breeder Genetic Algorithm (BGA) with adaptive mutation. The three PSSs are designed using eigenvalues analysis, whereby the lowest damped ratio is maximized. A comparison is done to determine which type of algorithm gives better results. Theoretically, the BGA and the PBIL based power system stabilizers perform better than the GA based power system stabilizer. The three PSSs (based on BGA, PBIL and GA) are tested against the Conventional Power System Stabilizer (CPSS), to verify their effectiveness. A four machine theoretical system is used in the simulations. Time domain simulations are presented to show that PBIL and BGA based PSSs perform better than the GA based PSS. However, the PSSs based on the evolutionary algorithms perform better than the CPSS.**

***Index Terms*-- Stability, genetic algorithm, population-based incremental learning, breeder genetic algorithm, adaptive mutation, cross over, premature convergence, low frequency oscillations, power system stabilizer**

I. INTRODUCTION

Electromechanical oscillations between interconnected synchronous generators are phenomena inherent to power systems. The stability of power systems is of vital importance and it is a major requirement for secure system operation [1-2]. Over the past decades, the oscillations of major concern in power systems were associated with single generators or a single machine connected to an infinite bus system. Some of the low frequency oscillations were also observed when large systems were connected by relatively weak tie lines. These low frequency modes were found to be associated with a group of generators in one area oscillating against a group of other generators in another area [1-2]. The low frequency oscillation modes that are more frequently encountered in an interconnected power systems are usually in the range of 0.1 to 3 Hz, but of importance are the inter-area modes which have a frequency range of 0.1Hz to 0.8Hz. These oscillations usually occur during and after a large or a small disturbance in the system. These oscillations can severely restrict system operations by requiring the curtailment of power transfer from one area to another as an operational measure if not taken care of adequately [2]. Therefore it is imperative that special control methods are used to stabilize and adequately damp these low frequency oscillations. Over the years, power system utilities have been using Power System Stabilizers (PSSs) to damp low frequency oscillations in power systems. These PSSs are usually designed over a certain nominal operating conditions, with their parameters tuned using conventional control approach such as root locus, phase compensation techniques, etc. These PSSs are often called Conventional Power System Stabilizers (CPSSs). Even though CPSSs perform decent job in stabilizing the system at nominal operating conditions, they have a problem when the system operating conditions change. That is, their performances degrade as the system condition changes [3-8]. Therefore, the control design and tuning of PSSs have a very significant influence on their performance and effectiveness in enhancing overall system stability [4-5]. As a result of this, modern control theory techniques have been applied to the design of PSS in recent years. These techniques vary from adaptive control, optimal control, variable structure control and H infinity control [6].

Recently, new techniques that have received increasing attention have been employed in designing robust and reliable PSSs to stabilize the system over a wide range of operating conditions. GAs are biologically motivated adaptive systems based on natural selection and genetics, they represent a heuristic search technique based on the evolutionary ideas of natural selection and genetics, and they operate by virtue of survival of the fittest [7-8]. GAs have been popular in academia and most recently is being accepted by some industry mainly because of its intuitiveness, ease of implementation, and the ability to solve highly non- linear, mixed integer optimization problems that are typical of engineering problems [7, 9]. GAs are usually used to solve optimization problems by exploitation of a random search [7-8]. Other family of Evolutionary Algorithms that will be considered in this work are the Breeder Genetic Algorithm (BGA) and the Population Based Incremental Learning (PBIL). BGA employs the same concept of survival of the fittest as employed in GAs; however BGA uses the artificial breeding similar to the one practiced in animal breeding. The result is an extremely versatile and effective function optimizer with very few parameters to be selected by the user

[10].
This work uses a slightly different version of BGA, called the Adaptive Mutation Breeder Genetic Algorithm (AMBGA). In AMBA, the mutation rate changes according to the nature of the fitness values [10].
PBIL is based on combining GAs and the competitive learning for function optimization. In other words, PBIL is an extension to the Evolution Genetic Algorithm achieved through the re-examination in terms of competitive learning [11].
This paper presents the simulation results of an interconnected multi-machine power system equipped with power system stabilizers that have been designed using the classical genetic algorithm (GA), Breeder genetic algorithm (BGA) and the Population-based Incremental learning (PBIL).

II. DESCRIPTION OF THE STUDY SYSTEM

The power system considered is the hypothetical two area network system [1]. The system consists of two identical areas separated by a relatively weak tie- line. Each area includes two identical generating units with equal power outputs. Each generating unit is equipped with a fast acting AVR and a speed governor.
This system allows for the analysis of a hypothetical interconnected network. It allows for the observation of all the low frequency oscillations that exist in the interconnected system. Specifically, special attention will be given to inter – area oscillations, which exist as a result of the power transfer from one area to another area. This results in the swinging of generators in one area against generators in another area. This will all be stabilized with the much more robust PSS designed using evolutionary algorithms. There are two loads connected in the system, on bus 4 and bus 14. The system model is shown in Figure 1.

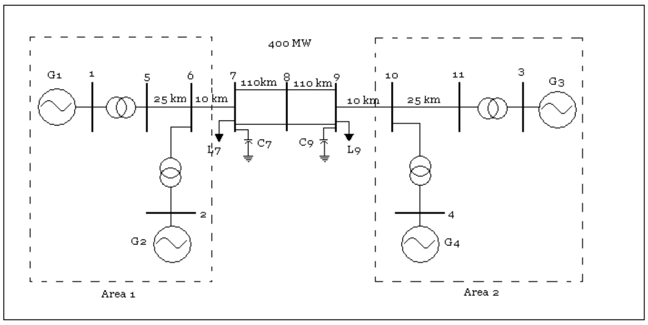

Figure 1: Two area system model

The generators G1 and G2 are in Area 1, while generators G3 and G4 are in Area 2. The generator G1 acts as the reference generator (slack bus on bus 1).

III. GENETIC ALGORITHM OVERVIEW

This section gives an overview of the traditional Genetic Algorithm (GA). GA has been used to solve complex and difficult problems in engineering that are hard or impossible to solve using conventional optimization methods. GA manipulates a set of potential solutions in a view to generate solutions which are better and fitter therefore using the principal of survival of the fittest. The fittest individuals in a population reproduce and survive to the next generation [9].

IV. OVERVIEW OF BGA

As has been mentioned before, BGA is a relatively new evolution algorithm; it is also based on survival of the fittest as in GAs with the exception that BGA is based on artificial selection. This work uses a modified version of BGA called the Adaptive Mutation BGA or AMBA [10]. The BGA usually uses real-valued representation as opposed to GA which mainly uses binary and sometimes floating or integer representation. The BGA uses truncation selection method, whereby selected top T% of the fittest individuals are chosen from the current generation to go through recombination and cross over to form the next generation. The rest of the individuals are discarded. In truncation method, the fittest individual called an *ellist* is guaranteed a place in the next generation. The other top (T-1) % go through recombination and mutation to form up the rest of the individuals in the next generation; recombination is similar to crossover in GAs. The process is repeated until an optimal solution is obtained or the maximum number of iterations have been reached. BGA is very versatile and easy to program [8]. Like GAs, BGAs also have recombination and mutation.

A. Recombination

Adaptive Mutation Breeder Genetic Algorithm (AMBGA) allows various possible recombination processes to be used, each of them searching the space with a particular bias and there is no prior knowledge as to which bias is likely to suit the task at hand. It will be best to include several of them and allow selection to do the elimination. The recombination processes that were used are volume and line recombinations [6, 10].
In volume recombination, a random vector r equal to the parents in length is generated and the child z_i is produced by the following expression [6, 10].

$$z_i = r_i x_i + (1 - r_i) y_i \quad (1)$$

where:
z_i is a component of the child
x_i and y_i are the two respective parent components
r_i is the random vector component

In other words, the child can be said to be located at a point inside the hyper box defined by the parents.
In line recombination a single random number r is generated between 0 and 1, the child is obtained by the following expression [10].

$$z_i = r x_i + (1 - r) y_i \quad (2)$$

where:
z_i is a component of the child
x_i and y_i are the two respective parent components
r is a random number between 0 and 1
In light of this, a child can be said to be located at a randomly chosen point on a line connecting the two parents.

B. *Mutation process*

One problem that has been of concern in GA is premature convergence, where the search might converge on local optima than the desired global one. This has been minimized by preserving the diversity of the population by adding a small injection of randomness or mutation [10]. This is achieved by adding a small vector of normally distributed zero mean random numbers to each child before inserting it into the population. The standard deviation **r** of the vector is very critical, as a small **r** might lead to a premature convergence or a large **r** might impair the search and reduce its ability to converge optimally, therefore it is better to use an adaptive approach where by the rate is modified during the course of the search. The population is divided into two halves X and Y. A mutation rate of 2**r** is applied to X while a mutation of **1/(2r)** to Y. The mutation **r** is adjusted depending on the population (X or Y) that is producing better and fitter solutions on average. If X individuals are fitter, then the mutation rate, **r** is increased by l0%, while if Y is fitter then the mutation rate, **r** is reduced by similar amount.

V. OVERVIEW OF PBIL

Population Based Incremental Learning has been preferred by many researchers over GA due to its simplicity, less computation time and capacity that is needed and which on numerous occasions outperforms the GA. PBIL was originally proposed by and developed in [11,12]. PBIL combines some aspects of GAs and competitive learning [11-12]. PBIL is an extension to the Evolution Genetic Algorithm (EGA) achieved through the re-examination of the performance of the EGA in terms of competitive learning [11, 12]. The crossover operator is taken away in PBIL, redefining the role of the population. PBIL works with probabilistic vectors, the probability vectors control the random bit strings generated by PBIL and are used to create other individuals through learning [13]. Learning in PBIL consists of using the current probability distribution to create N individuals. Using the objective function, the performance of these individuals is vindicated. Using the best individual, the probability vector is updated, increasing the probability of producing solutions similar to the current best individuals. Mutation is used to maintain diversity in PBIL. Mutation in PBIL can be performed into two of the methods. The first method is performing mutation on the population generated. The second is to perform mutation on the probability vector.
PBIL has the following properties [12, 13]:

- There is no crossover or fitness operators
- It works with probability vectors, this probability vector controls the random bit strings generated by the PBIL and is used to create other individuals through learning
- No need to store all solutions, it only stores the current best solution and the current solution being evaluated

Recent works have indicated that PBIL is simpler, faster and more effective than the GA, and outperforms the GA in many optimization applications.

VI. OBJECTIVE FUNCTION

The purpose of this paper is to optimize the parameters of the PSS simultaneously such that the controller can stabilize the system over a wide range of operating conditions. The parameters that were to be optimized are K (gain of the PSS) as well as T_1, T_2, T_3 and T_4 which forms up the lead lag circuit. The controllers were designed over a wide range of multi operating conditions. The objective function used was to maximize the lowest damped ratio over a wide range of operating conditions [8]. This objective function was used both in GA, BGA and PBIL. This objective function is given as:

$$val = \max(\min(\zeta_i)) \qquad (3)$$

$$i = 1,2,3 \ldots\ldots\ldots..n$$

i represents the number of the eigenvalue, while
$\zeta_i = \frac{-\sigma_i}{\sqrt{\sigma_i^2+\omega_i^2}}$ is the damping ratio of the i^{th} eigenvalue.
σ_i, ω_i are the real part and the imaginary part (frequency) of the i^{th} eigenvalues respectively.

In this work no constraints were imposed on the damping ratio. The only requirement was to maximize the minimum damping ratio.

VII. PSS DESIGN

For comparison purpose a CPSS was also designed. In total four PSSs were designed and their performances compared. For the BGA, PBIL and GA, there are 5 parameters for each optimization type that were being optimized: K, T_1, T_2, T_3 and T_4. The washout time constant (T_w) was set at 10 seconds, this is because T_w is not critical. Ten parameters were optimized for Generators 1 to 4.

A. *Application of GA to PSS Design*

The following parameters were used in designing the PSS using GA and configured in the following way:

Chromosome representation: real
Population: 400
Generation: 200
Mutation: 0.01

B. *Application of BGA to PSS Design*

The parameters for the BGA design were configured in the following way:

Chromosome representation: real
Population: 400
Generation: 150
Mutation: adaptive with starting value of 0.1

C. *Application of PBIL to PSS Design*

The parameter's configuration that was used in PBIL is as follows:
Chromosome length: 15
Population: 400
Generation: Learning rate: 0.1
Forgetting factor: 0.005

D. *CPSS Design*

The parameters for the CPSS were tuned based on a certain nominal operating condition using phase compensation method and trial and error approach. The continuous transfer function of the PSS is given below:

$$Upss(s) = K \frac{sT_5}{1+sT_5} \frac{1+sT_1}{1+sT_2} \frac{1+sT_3}{1+sT_4} \Delta\omega(s) \quad (4)$$

E. *Operating conditions considered*

The following operating conditions were considered when tuning the PSS parameters based on the Evolutionary Algorithms.

TABLE I: OPERATING CONDITIONS CONSIDERED IN THE PSS DESIGN

Operating condition	Generator	Active power (pu)	Line Reactance (pu)
1	G1 G2 G3 G4	5 7 8 8	0.11
2	G1 G2 G3 G4	7.2615 7 7 7	0.11
3	G1 G2 G3 G4	5 7 8 8	0.22
4	G1 G2 G3 G4	7.2615 7 7 7	0.22
5	G1 G2 G3 G4	5.2615 9 5 9	0.11
6	G1 G2 G3 G4	5.2615 9 5 9	0.22

In operating conditions 1 and 2, area 1 composed of Generator 1 and 2, exports 200MW to area 2, while in operating conditions 3 to 6, Area 1 exports 400MW to Area 2, but the amount of power generated by the different generators changes according to the operating conditions. In this study, operating condition 2 is chosen as the nominal operating condition.

VIII. SIMULATION RESULTS

The results from the simulations are shown in this section. The results of three operating condition are discussed. Only the time domain simulations will be shown for the 10% step response in the exciter voltage reference of generator 2 and the response to a 6 cycles three phase fault on line 1 and cleared by removing the line.

The three operation conditions considered in the simulations are given in Table II. The results used for analysis were chosen randomly to show the robust stability of the designed PSS outside the designed operating conditions and therefore the cases shown below were not used in the optimization, except for the nominal condition.

TABLE II: OPERATING CONDITIONS CONSIDERED IN THE ANALYSIS OF THE RESULTS

Case	Generator	Active power (pu)	Line Reactance (pu)
1	G1 G2 G3 G4	7 8 6 6	0.11
2	G1 G2 G3 G4	7.2615 7 7 7	0.11
3	G1 G2 G3 G4	8.2615 6 8 6	0.11

A. *Step Responses*

Figures 2 to 4, show the time responses for the power deviation of generator 2 to a step response of a 10% change in the voltage reference of generator 2. The results show that the system is well stable and damped with all four PSSs considered. But comparing the evolutionary algorithm based PSSs with the CPSS, the BGA-PSS, PBIL_PSS and GA-PSS perform much better than the CPSS across all the operating conditions considered. Having said that, the BGA and PBIL based PSSs have a better damping as compared to the GA based PSS. Considering case 1, the maximum overshoot of the second swing of the system with CPSS is approximately 0.15p.u, with GA-PSS is around 0.11 p.u, with BGA-PSS and PBIL-PSS is close to 0.10 p.u. The system with CPSS settles in around 8 to 9 seconds. The settling time for GA-PSS is around 5 to 6 and about 3 to 4 seconds for the BGA-PSS and PBIL-PSS, respectively. This same trend applies to cases 2 and 3. In case 3 of figure 4, the system with CPSS settles after 10 seconds, with GA-PSS settles in around 4.5 seconds while with the BGA-PSS and PBIL-PSS settles in around 3.5 seconds. The seond swing amplitude is still high for the CPSS, which is close to 0.1pu, whereas, the maximum amplitudes for GA-PSS, BGA-PSS and PBIL-PSS are 0.08 pu, 0.06 pu, and 0.06 pu, respectively.

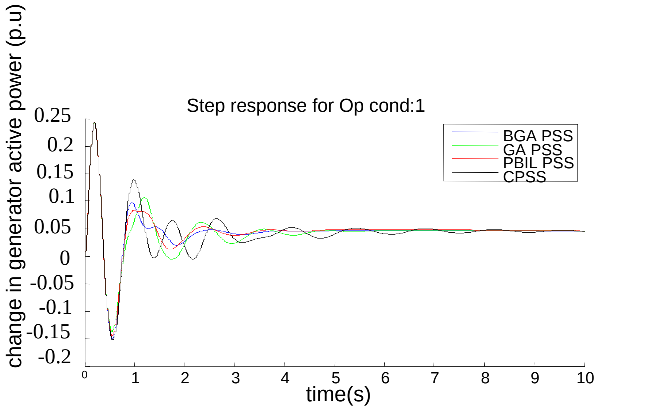

Figure 2: G2 step response to 10% change in Vref at Op cond: 1

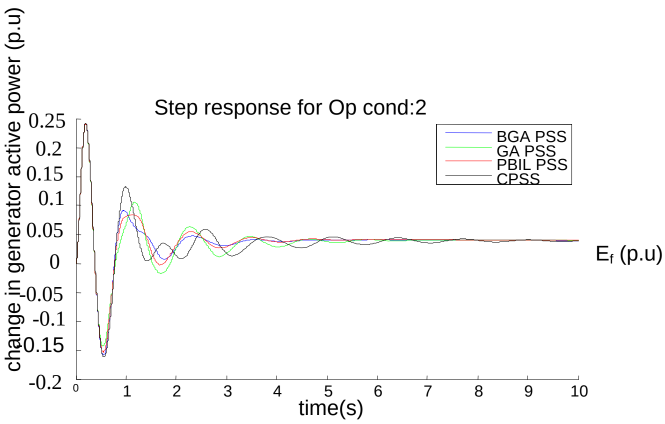

Figure 3: G2 step response to a 10% change in Vref at Op cond: 2

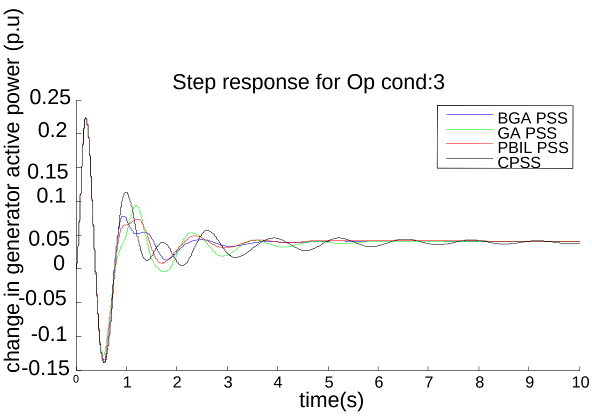

Figure 4: G2 step response to a 10% change in Vref at Op cond: 3

B. *Transient Response*

The system responses to a 6 cycle three phase fault in line 1 are shown below in Figures 5 and 6. The fault was cleared by removing the line.

Figure 5 shows the electrical power output responses. It can be seen that the system equipped with the CPSS settles after 7 seconds, but has a smaller overshoot than all the other PSSs. On the other hand, with GA-PSS settles after 4 seconds while with BGA-PSS and PBIL-PSS, the system settles in around 3 seconds.

Figure.6 shows the AVR field voltage. In all the cases, the field voltage value reaches the ceiling for the first and second swings due to the limitations of the PSS output.

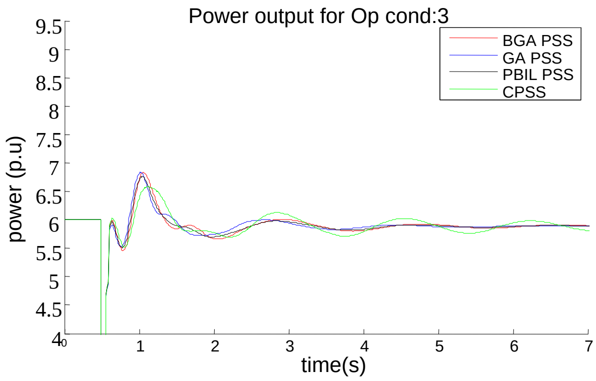

Figure 5: G2 Power response to a 3 phase fault on Line 1

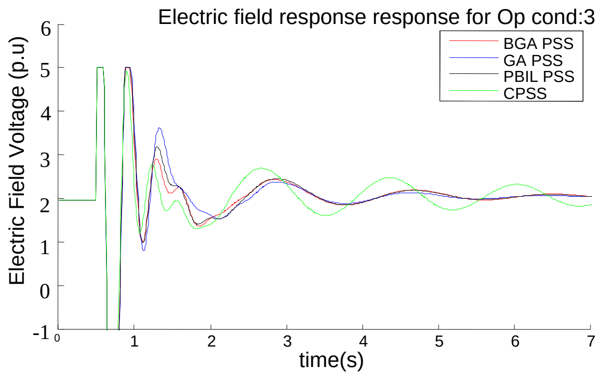

Figure 6: G2 Electric field response to a 3 phase fault on Line 1

IX. CONCLUSION

Power system stabilizers were designed using three different evolutionary algorithms for a multi-machine power system having local and inter-area oscillations. These PSSs were tested using both small and large disturbances and the time domain simulation results were presented. The results showed that the BGA-PSS and PBIL-PSS perform similarly. However, these two PSSs gave better results than the GA-PSS. All the three PSSs based on evolutionary algorithms perform better than the CPSS across all the operating conditions as expected. This can be attributed to the fact that, the three evolutionally algorithm PSSs are designed across a wide range of operating conditions.

X. ACKNOWLEDGEMENT

We would like to thank Mr. David Oyedokun for reading the manuscript and for providing useful comments.

XI. REFERENCES

[1] M. Klein, G.J Roggers, P. Kundur, “A fundamental study of inter – area oscillations in power systems” Transactions on Power Systems, Vol. 6, No. 3, August 1991.

[2] G. J Rogers, “Power System Oscillations”, Kluwer Academic Publishers, 2000.

[3] P. Kundur, M.Klein, G.J. Rogers, M.Zwyno, “Application of Power System Stabilizers for Enhancement of overall System Stability”, IEEE Trans. on Power Systems – Vol. 4, No.2, May 1989.

[4] M.A. Abido, “Robust Design of Multimachine Power System Using Simulated Annealing”, IEEE Trans, Energy Conversion, Vol. 15, No. 3, pp 297- 304, September 2000.

[5] H. Shayeghi, A. Safari, A. Shayanfar, “Multimachine Power System Stabilizers Design Using PSO Algorithm”, Int Journal of Electrical Power and Energy Systems Engineering Vol. 1, No. 4, 2008.

[6] K A. Folly, N. Yorino and H. Sasaki, “Synthesis of Two-input PSS Based on the H infinity Control Theory”, Trans. IEE Japan, Vol. 118-B, No.6, pp. 699-706, 1998.

[7] J. Satheeshkumar Jegadeesan, A. Ebenezer Jeyakumar, ”A novel Approach for Tuning Power System Stabilizer (SMIB system) using genetic local search technique”, Proc. Of the 4th World Scientific and Engineering Academy and Society (WSEAS) International Conference on Artificial Intelligence, 2005.

[8] SPN Sheetekela, K A Folly, “Optimization of power system stabilizer using genetic algorithm techniques based on eigenvalue analysis”, SAUPEC 2009, 28 – 29 January 2009.

[9] Sidhartha Panda and C. Ardil “Real-Coded Genetic Algorithm for Robust Power System Stabilizer Design”,*International Journal of* Electrical, Computer, and Systems Engineering Volume 2 No.1, pp 6 – 14, 2008

[10] John Greene,”The basic idea behind the Breeder Genetic Algorithm” Department of electrical engineering, Course Note, Uinversity of Cape Town 2005.

[11] Shumeet 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

[12] S. Baluja, Rich Caruana, “Removing the genetics from the standard Genetic Algorithm”, Proc 12th International Conference on Machine Learning, Lake Tahoe, CA, July 1995

[13] K.A. Folly, ”Multimachine Power System Stabilizer Design Based on a Simplified Version of Genetic Algorithms Combined with Learning”, International Conference on Intelligence Systems Application to Power Systems (ISAP), 2005