

Chapter

Power System Small-Signal Stability Enhancement Using Damping Controllers Designed Based on Evolutionary Algorithms

Komla Agbenyo Folly, Severus Panduleni Sheetekela and Tshina Fa Mulumba

Abstract

This chapter is concerned with the stability enhancement of a power system using power system stabilizers (PSSs) designed based on four evolutionary algorithms (EAs), namely, genetic algorithms (GAs), breeder genetic algorithm (BGA), population-based incremental learning (PBIL), and differential evolution (DE). GAs have been widely applied in many fields of engineering and science and have shown to be a robust and powerful adaptive search algorithm. However, GAs are known to have several limitations. To deal with these limitations, many variant forms of GAs have been suggested often tailored to specific problems. In this research, we investigated the performances of GA-PSS and three other EAs-based PSSs (i.e., BGA-PSS and PBIL-PSS and DE-PSS) in improving the small-signal stability of a power system. These EAs have been selected on the basis of their simplicity, efficiency, and effectiveness in solving the optimization problem at hand. Frequency domain and time-domain simulation results show that DE-PSS, PBIL-PSS, and BGA-PSS performed better than GA-PSS. Time domain simulations suggest that overall, DE-PSS performs better than PBIL-PSS and BGA-PSS in terms of undershoot and subsequent swings, albeit with a relatively large first swing overshoot. The performances of BGA-PSS and PBIL-PSS are similar. On the other hand, GA-PSS gives a better response than the conventional PSS (CPSS).

Keywords: breeder genetic algorithm, damping ratio, genetic algorithms, differential evolution, low-frequency oscillations, power-system stabilizer, population-based incremental learning

1. Introduction

Over the past decades, low-frequency oscillatory modes have been a major concern to power system engineers [1]. These oscillatory modes ranging from 0.1 to 3 Hz tend to be poorly damped especially in moderately to heavily loaded systems that are

equipped with high gain, fast-acting automatic voltage regulators (AVRs) [2, 3]. Generally, we distinguish two main oscillation modes: local and inter-area modes. Local modes (0.8–2 Hz) involve local generators oscillating against each other. On the other hand, inter-area modes are caused by groups of generators in one part of the system swinging against other groups in the interconnected power system having frequencies ranging from 0.1 Hz to 0.8 Hz. Compared to local modes, inter-area modes are generally the most critical modes that need to be damped [4, 5]. These modes are found in almost all interconnected power stems. If they are not adequately damped, the oscillations may sustain and grow, and this may lead to system blackout. Power system stabilizers (PSSs) have been proposed to modulate low-frequency oscillations and increase the damping of electromechanical modes [1, 2]. Tuning the PSS parameters is not a trivial task. Power utilities have preferred using conventional PSSs (CPSSs) designed around a nominal operating condition. The design of the CPSS is generally based on conventional control approaches such as root locus, phase compensation, and pole placement techniques [1–5]. However, since these approaches are not robust, the designed CPSS tends to deviate from optimal operation when the system experiences a range of changes away from the nominal operating conditions. Therefore, new design approaches are required to design a PSS that can operate optimally under a wide range of operating conditions [3, 6]. Evolutionary algorithms (EAs) such as genetic algorithms (GAs) [7–12], differential evolution (DE) and its variants [13, 14], particle swarm optimization (PSO) [15], population-based incremental learning (PBIL) [16–19], and breeder genetic algorithms (BGA) [11, 20–24] are efficient heuristic search methods that are capable of solving complex optimization problems. They do not require the objective function to have properties such as continuity, smoothness, and differentiability. They have many advantages over traditional optimization methods and have attracted considerable attention in recent years. Many of these methods have been applied to power system damping controller design with encouraging results. In particular, GAs have been extensively used to solve global optimization problems in academia and are now being accepted by some industries [9]. DE, PBIL, and BGA are easy to implement yet efficient and robust in solving optimization problems. Therefore, they are considered in this work.

GAs are biologically motivated adaptive systems based on natural selection and genetics. GAs are generally used to solve optimization problems by the exploitation of a random search [7, 8]. Although GAs are seen to be robust and powerful adaptive search mechanisms, they have several drawbacks [9]. One of these drawbacks is related to “genetic drift.” This phenomenon prevents GAs from maintaining diversity in their population. Other issues include the nonexistence of theoretical guidance for selecting optimal GA parameters such as population size, crossover, and mutation rates. Moreover, the natural selection approach used by GAs is not immune from failure [22]. Breeder genetic algorithm (BGA) has been proposed to cope with some of these drawbacks. It applies almost the same ideas as in GA, except that it is based on artificial selection as practiced in animal breeding rather than using natural selection based on Darwinian evolution [23, 24]. Artificial selection (selective breeding) refers to the intentional breeding for certain qualities or a combination of qualities [23]. This is in contrast with the natural selection that is the process whereby organisms survive and produce offspring by naturally adapting to their environment. Generally, individuals in BGA are represented as real numbers instead of binary or integers. The main advantage of using BGA over GA is its simplicity in the selection method and the fewer parameters. The major limitation of this algorithm is that there is a likelihood of

premature convergence that could lead BGA to converge to the local optimum rather than the global one. To deal with the problem of premature convergence, an adaptive mutation is used [23, 24]. In this case, the mutation rate is not fixed but varies according to the convergence and performance of the population. This is the type of BGA that will be discussed later in this chapter.

Population-based incremental learning (PBIL) is a combination of GA and competitive learning. It extends the features of the evolutionary genetic algorithm (EGA) through the reexamination of the performance of the GA in terms of competitive learning [16–19]. It was originally proposed by Baluja [18, 19]. In PBIL, the crossover operator is removed, and the role of the population is redefined. PBIL works on probabilistic vectors (PVs), which control the random bit strings generated by PBIL. The PVs are used to create other vectors through competitive learning. The PV is then updated to increase the likelihood of producing solutions corresponding to the current best individual. It has been shown that PBIL is simpler than GA and in many cases performs better than GA and has less overhead [11, 16–19].

Differential evolution (DE) is a powerful stochastic optimizer whose search mechanism involves a differential mutation technique [12, 13, 25]. The algorithm is both simple and robust, with several variants exhibiting different tradeoffs between convergence speed and robustness. Most often DE outperforms its counterparts in efficiency and robustness [12–14, 25].

This chapter discusses the optimal design of power system stabilizers (PSSs) using four evolutionary algorithm (EAs) techniques, namely, genetic algorithms (GAs), breeder genetic algorithm (BGA) with adaptive mutation, population-based incremental learning (PBIL), and differential evolution (DE). For comparison purposes, the conventional PSS (CPSS) is also included in this work. The performance and effectiveness of the PSSs in damping the electromechanical modes are investigated using both frequency-domain analysis and time-domain simulations. Simulation results show that all the EA-based PSSs (GA-PSS, BGA-PSS, PBIL-PSS, and DE-PSS) perform better than the CPSS for all the operating conditions considered. Frequency domain simulation suggests that DE-PSS, PBIL-PSS, and BGA-PSS have similar performances in terms of the damping ratios that they provided. Time-domain simulations however suggest that overall, DE-PSS performs slightly better than PBIL-PSS and BGA-PSS in terms of undershoot and subsequent swings, albeit with a slightly large 1st swing overshoot. GA-PSS is shown to give the worst performance amount to the EAs. The chapter is organized as follows: Sections 2–4 present the overview of BGA, PBIL, and DE, respectively; Section 5 discusses the system model; Section 6 is concerned with the objective function; Section 7 presents the design of the PSSs; Section 8 discusses the simulation results; and the conclusions are presented in Section 9.

2. Overview of breeder genetic algorithm

As discussed previously, breeder genetic algorithm (BGA) is similar to genetic algorithms (GAs), with the exception that it uses artificial selection and has fewer genetic parameters. Also, BGA uses real-valued representation as opposed to GAs that mainly use binary and sometimes floating or integer representation. BGA is a versatile and effective function optimizer. It has the advantage of being simpler than GA. To deal with the issue of premature convergence that is common with BGA, a modified version of BGA called adaptive mutation BGA is used in this work [11, 20, 23]. In the truncation selection method that has been adopted, the $T\%$ of the fittest individuals is

selected from the current population of N individuals and goes through recombination and mutation to form the next generation. The rest of the individuals are discarded. In the truncation method, the fittest individual in the population is automatically part of the next generation. The other top $T\%-1$ goes through recombination and mutation to form the rest of the individuals in the next generation. The process is repeated until an optimal solution is obtained or the maximum number of iterations has been reached.

2.1 Recombination

Recombination is similar to a crossover in GAs. The adaptive mutation BGA proposed in this work allows various possible recombination methods to be used, each of them searching the space with a particular bias. Because we do not have prior knowledge as to which bias is likely to suit the optimization task, it is better to include several recombination methods and allow selection to do the elimination. Two recombination methods were used in this work: volume and line recombination [11].

In volume recombination, a random vector r_i equal to the parents' length is generated and the child c_i is produced by the following expression:

$$c_i = r_i a_i + (1 - r_i) b_i \quad (1)$$

Where c_i is a component of the child, a_i and b_i are the two respective parent components, and r_i is a random vector component.

The child can be said to be located at a point inside the hyper box defined by the parents as shown in **Figure 1**.

In line recombination, a single uniformly random number r is generated between 0 and 1, and the child is obtained as shown below [23].

$$c_i = r a_i + (1 - r) b_i \quad (2)$$

Where c_i , a_i , and b_i are defined as in Eq. (1).

2.2 Adaptive mutation

As mentioned before, one of the main concerns in GA has been the issue of premature convergence. This issue is also encountered in the classical BGA. This problem can be reduced in BGA by using an adaptive mutation [11, 21, 23]. The

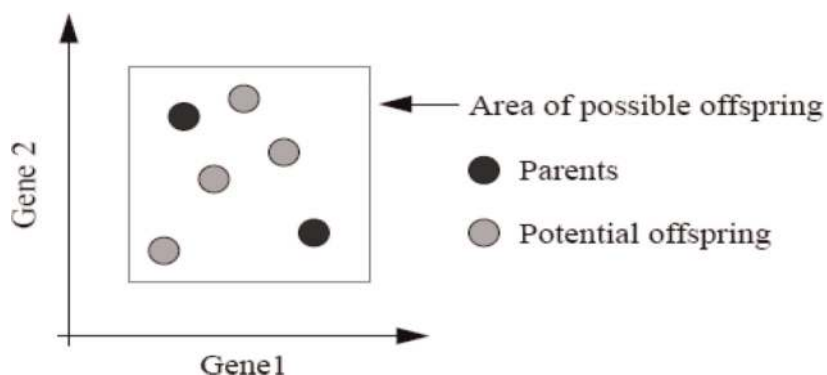


Figure 1.
Volume recombination.

diversity in the population is preserved by adding small, normally distributed zero-mean random numbers to each child before inserting it into the population. The random numbers have a certain standard deviation R [18]. The value of R should be selected carefully because it is critical in determining the convergence of the optimization. If the value of R is too small, the solution might result in premature convergence, while a high value of R might be detrimental to the optimal convergence of the algorithm [11, 23]. The adaptive mutation method proposed here allows us to determine the appropriate value of R . To achieve this, the population is divided into two halves, $P1$ and $P2$. $P1$ is assigned a mutation rate of double R ($2R$), while $P2$ is assigned a mutation rate of half R ($R/2$). The mutation rate R is adjusted depending on the performance of each half of the population ($P1$ or $P2$). If $P1$ gives better and fitter individuals, the mutation rate is increased by a certain percentage (10% in this case); similarly, if $P2$ produces better and fitter individuals, then the mutation rate gets reduced by a similar percentage. The pseudo code for BGA with adaptive learning can be found in [11, 23].

3. Overview of population-based incremental learning algorithm

Population-based incremental learning (PBIL) is a combination of competitive learning derived from artificial neural networks and genetic algorithms [18, 19]. There is no crossover operator in PBIL, instead, the probability vector is updated using a solution with the highest fitness values [18]. The values of the probability vector are initially set to 0.5 to ensure that the probability of generating 0 or 1 is equal. As the search progresses, these values are moved away from 0.5, toward either 0.0 or 1.0.

3.1 Learning rate

Learning in PBIL is based on using the current probability distribution to create N individuals. The probability vector is updated using the best individual so far, thereby increasing the probability of producing solutions similar to the current best solutions. Learning rate is required to update the probability vector. The selection of the learning rate value should be made with care as it determines how fast or slow the prototype vector is shifted toward the best individuals. A larger rate speeds up convergence, but it reduces the function space to be searched, while a smaller rate will slow down the convergence, even though it increases the exploration of a bigger search space, thereby increasing the likelihood of better optimal solutions. The (positive) update rule of the probability vector is given as:

$$PV_i = (1 - LR)PV_i + (LR)B_i \quad (3)$$

where PV is the probability vector, $LR \in [0 \ 1]$ is the learning rate, B is the best solution, and i denotes each locus ($i = 1, 2, \dots, l$) where l is the binary encoding length.

3.2 Mutation

Like in GA, the mutation is used in PBIL to maintain diversity in the population. Mutation in PBIL can be performed in two ways: either on the sample solutions

generated or on the PV. In this study, the mutation is performed on the PV; a forgetting factor is used to relax the probability vector toward a neutral value of 0.5 [11, 16, 17] as shown in the equation below.

$$PV_i = PV_i - FF(PV_i - 0.5) \quad (4)$$

where FF is the forgetting factor that was chosen to be 0.005. The pseudo code for PBIL can be found in [17–19].

4. Overview of differential evolution

Differential evolution (DE) can be defined as a parallel direct search method that uses a population of points to search for a global minimum or maximum of a function over a wide search space [13]. It is a simple and efficient adaptive scheme for global optimization over continuous space. DE is designed to efficiently solve non-differentiable and nonlinear functions and yet retains its simplicity and good convergence to a global optimum [12]. Similar to most EAs, DE explores the search space by maintaining a population of candidate solutions and by using Darwinian evolution theory to direct its search toward prospective areas. The candidates with better fitness values survive and enter the next generation [12–14, 25]. The process continues until the termination criterion is satisfied. It should be mentioned that DE has proved to be one of the best among EAs. It was able to secure competitive rankings in CEC competitions [25]. One of the main advantages of DE over GA is the mutation scheme and the selection process. Unlike GAs where the best solutions are selected for the next generation, in DE, all solutions have an equal chance of being selected as parents independently of their fitness values.

4.1 Mutation

In the context of DE, “mutation” is defined as a process of taking a small random sample of vectors from the current population and combining them algebraically to form a new vector, which is referred to as a *mutant vector* [12, 13]. In the so-called classical version of DE, the mutant vector is formed as follows:

$$V_{i,g} = X_{r1,g} + F(X_{r2,g} - X_{r3,g}) \quad (5)$$

where i , $r1$, $r2$, and $r3$ are all distinct indices in the interval $[1, N_p]$. The mutation scale factor F is a positive real number between 0 and 2 that controls the rate at which the population evolves [13]. The vector X_{r1} is the *base vector*, while $X_{r2} - X_{r3}$ is the *difference vector*, $g = 0, 1, \dots, g_{max}$ are the generations and N_p is the population.

The above process is repeated N_p -times to constitute a mutant population. In the classical version, each base vector is used only once per generation, in order to preserve diversity in the population. The classical version described above is designated as “DE/rand/1” and is widely used, although it has the drawback of relatively slow convergence [12]. Some alternative mutation strategies to the classical version are given below [12–14, 25]:

DE/best/1: This strategy resembles DE/rand/1, except that all mutants use the best vector in the current generation as the base vector:

$$V_{i,g} = X_{best,g} + F(X_{r1,g} - X_{r2,g}) \quad (6)$$

where X_{r1} and X_{r2} are distinct random vectors and X_{best} is the best individual in the current population.

This strategy has faster convergence than DE/rand/1, but often fails to reach the global optimum [12].

DE/best/2: This strategy uses two mutation differences to create a mutant vector:

$$V_{i,g} = X_{best,g} + F(X_{r1,g} - X_{r2,g}) + F(X_{r3,g} - X_{r4,g}) \quad (7)$$

where X_{r1}, X_{r2}, X_{r3} , and X_{r4} are distinct random vectors and X_{best} is the best individual in the current population. This strategy attempts to balance between convergence speed and robustness. However, it may still converge to a local but non-global optimum due to the fact that the base vector X_{best} draws the population toward itself [13].

DE/local-to-best/2: This strategy resembles DE/best/2 in that two mutation differences are used, but the base vector is randomly sampled and the “best” vector is used in one of the scaled differences:

$$V_{i,g} = X_{r1,g} + F(X_{best,g} - X_{r2,g}) + F(X_{r3,g} - X_{r4,g}) \quad (8)$$

This approach has similar convergence properties to DE/best/2 [13].

DE/rand/2: This strategy samples 5 random vectors in the current generation to form two random differences that are scaled and added to the base vector:

$$V_{i,g} = X_{r1,g} + F(X_{r2,g} - X_{r3,g}) + F(X_{r4,g} - X_{r5,g}) \quad (9)$$

where, $r1 \neq r2 \neq r3 \neq r4 \neq r5$. This approach converges more slowly but is very robust [13].

DE/rand/2 has been used in this work due to our objective to appropriately tune the PSS with optimal time constants values for a robust performance.

4.2 Crossover

In DE, “crossover” refers to the process of creating a new vector (called the *trial vector*) by combining a mutant vector with a *target vector* [13]. The target vector for the mutant vector $V_{i,g}$ is $X_{i,g}$. The trial vector $U_i = [u_{1,i}, u_{2,i} \dots, u_{D,i}]$, is then obtained as follows:

$$u_{j,i,g} = \begin{cases} v_{j,i,g} & \text{if } (rand_j(0, 1) \leq CR \text{ or } j = j_{rand}), \quad j = 1, 2, \dots, D \\ x_{j,i,g} & \text{otherwise} \end{cases} \quad (10)$$

where $CR \in [0, 1]$ is the *crossover probability*, and CR is the fraction of the parameter values that are copied from the mutant vector, and $1-CR$ is the fraction of parameter values copied from the trial vector. To determine whether the parameter to be copied is from the mutant or trial vector, a uniformly-distributed random number, $rand_j$ between $[0, 1]$ is generated and compared to the predefined value of CR . In addition, a random index $j_{rand} \in [1, N_p]$ is chosen and the corresponding mutant parameter is copied to ensure that the trial vector is not a duplicate of the target vector.

4.3 Selection

This process consists of choosing the individuals that will enter the next generation. DE employs a “one-to-one survivor selection,” which consists of comparing each trial vector to its corresponding target vector. Mathematically, the vector $X_{i,g+1}$ in the $g + 1$ 'th generation is obtained from the trial vector $U_{i,g}$ and target vector $X_{i,g}$ as follows in the case of a minimization problem:

$$X_{i,g+1} = \begin{cases} U_{i,g} & \text{if } f(U_{i,g}) \leq f(X_{i,g}) \\ X_{i,g} & \text{otherwise} \end{cases} \quad (11)$$

This process ensures that the best vector at each index is retained. Furthermore, this also guarantees that the very best-so-far solution is kept. Once the selection is performed for all target vectors in the current generation g , the processes of mutation, crossover, and selection are repeated with the N_p vectors in the $g + 1$ st generation. This process is iterated until a termination criterion is satisfied.

5. System model

The power system considered in this paper is the two-area four-machine power system as shown in **Figure 2** [1]. Each machine is represented by the detailed six-order differential equations. The machines are equipped with simple exciter systems of first-order differential equations as given in the **Appendix** [11]. The system consists of two similar areas connected by a tie-line. Each area consists of two coupled conventional generator units, each generator is rated 900 MVA and 20 kV. The generator parameters can be found in [1, 11]. The dynamics of the system are described by a set of nonlinear differential equations. However, for the purpose of controller design, these equations are linearized around the nominal operating conditions. The linearized equation of the system is given by:

$$\begin{aligned} \dot{x} &= A_o x + B_o u \\ y &= C_o x + D_o u \end{aligned} \quad (12)$$

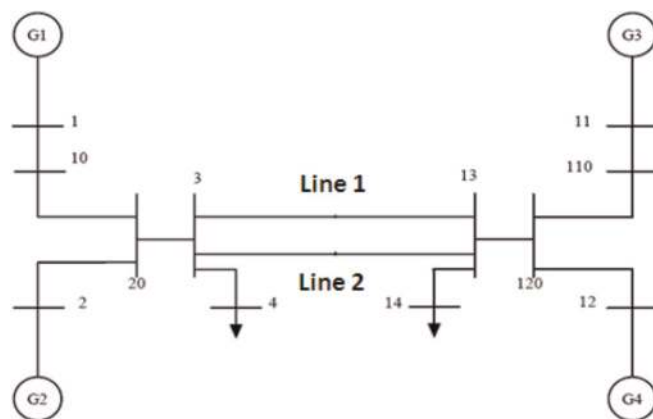


Figure 2.
Two-area system model.

where, x is the state variable, y is the system output, and u is the control input. A_o , B_o , C_o , and D_o are constant matrices of appropriate dimensions.

Several operating conditions have been considered during the design stage of the controller. However, only three operating conditions are listed in **Table 1** for simplicity. Case 1 is the nominal operating condition. At the nominal operating condition, approximately 146 MW is transferred from area 1 to area 2 via the two tie lines, with each line carrying half of the total power. Under these conditions, the load on bus 4 was 1137 MW, while the load on bus 14 was 1367 MW. Case 2 is the moderate load condition, where about 409 MW of real power is transferred from area 1 to area 2. For this case, the load on bus 4 was 967 MW, while the load on bus 14 was 1767 MW. Case 3 is the heavy load condition (worst case scenario) where approximately 512 MW of power is transferred from area 1 to area 2. For this case, the load on bus 4 was 876 MW, while the load on bus 14 was 1876 MW. It should be mentioned that the system exhibits inter-area oscillatory modes due to the flow of power between the two areas that causes the two areas to oscillate against each other. In addition, two local area modes were also observed, one in each area. However, in this chapter, we will concentrate only on the inter-area modes since they are the most critical and difficult to control. **Table 2** shows the open-loop eigenvalues of the inter-area modes. It can be seen that without PSSs, the inter-area modes were stable but poorly damped for case 1, with a damping ratio of 0.011. However, the system became unstable for case 2 and the instability became more pronounced for case 3 with damping ratios of -0.0057 and -0.0130 , respectively. This suggests that with the increase in active power transfer between the two areas, the oscillations have now increased making the system unstable. The frequency of oscillations of the inter-area modes ranges from 0.588 Hz to 0.634 Hz.

Therefore, a supplementary controller known as a power system stabilizer (PSS) will be required to damp the system's oscillations. The block diagram of the PSS is shown in **Figure A.1** in the **Appendix**.

6. Objective function

The objective is to optimize the parameters of the PSSs simultaneously such that the controllers 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 the lead-lag time constants T_1 , T_2 , T_3 , and T_4 . The objective function used was to maximize the lowest damped ratio over a wide range of operating conditions. This objective function was used for GA, BGA, PBIL, and DE. The objective function is given as:

Case	Active power transfer from area 1 to area 2 [MW]	Number of tie-line between areas 1 and 2	Load's active power at bus 4 [MW]	Load's active power at bus 14 [MW]
1	146	2	1137	1367
2	409	2	967	1767
3	512	2	876	1876

Table 1.
 Selected operating conditions.

Case	Inter-area mode	Damping ratio (%)	Frequency of oscillations (Hz)
1	$-0.044 \pm j3.98$	1.10	0.634
2	$0.022 \pm j3.78$	-0.57	0.602
3	$0.048 \pm j3.69$	-1.30	0.588

Table 2.
Open-loop eigenvalues of the inter-area modes for selected operating conditions.

$$val = \max \left(\min \left(\zeta_{ij} \right) \right) \quad (13)$$

where

$$i = 1, 2, \dots, n, j = 1, 2, \dots, m$$

$$\zeta_{ij} = \frac{-\sigma_{ij}}{\sqrt{\sigma_{ij}^2 + \omega_{ij}^2}} \quad (14)$$

where $\zeta_{ij} = \frac{-\sigma_{ij}}{\sqrt{\sigma_{ij}^2 + \omega_{ij}^2}}$ is the damping ratio of the i th eigenvalue of the j th operating conditions. The number of the eigenvalues is n , and m is the number of operating conditions. σ_{ij} and ω_{ij} are the real part and the imaginary part (frequency) of the eigenvalue, respectively.

7. Design of the PSSs

In total 10 PSSs parameters were optimized (i.e., 5 parameters for each area) for generators 1–4. The parameters that were optimized are K , T_1 , T_2 , T_3 , and T_4 . The washout time constant (T_w) was set at 10 seconds and was not optimized since T_w is not critical to the design. The following parameter domain constraints were considered when designing the PSSs.

$$0 < K \leq 20$$

$$0.001 T_i \leq 5$$

where K and T_i ($i = 1, 2, 3, 4$) denote the controller gain and the lead-lag time constants, respectively.

For comparison purposes, a CPSS was also designed using the phase compensation technique. Details can be found in [1, 2].

7.1 Parameters of GAs, BGA, PBIL, and DE

The parameters used in the optimization for GAs, BGA, PBIL, and DE are shown in **Table 3**.

An observation of the parameters given below in **Table 3** shows that PBIL uses few parameters. There is no crossover or selection in PBIL compared to BGA, GA, and DE. In addition, 500 generations were used in the PBIL optimization to allow for adequate learning to take place within the optimization. This is because PBIL that works by learning from the previous best and trying to find the very best individual takes time to explore the

Parameters	GA	BGA	PBIL	DE
Population	100	100	100	50
Generation	120	120	500	180
Selection	Normal geometric	Truncation selection	—	Greedy
Crossover/ Recombination	Arithmetic	Line and volume	—	Binomial (CR: 0.95)
Mutation	Nonuniform	Adaptive random (initial R_{nom} : 0.01)	Forgetting Factor (FF:0.005)	DE/rand/2 (F: 0.95)
Learning rate (LR)	—	—	0.1	—

Table 3.
 Parameters used in GA, BGA, PBIL, and DE.

search space. Another difference is the way in which the initial population is generated. In GA, BGA, and DE, the initial population is selected randomly, while in PBIL the role of the population is redefined using probability vectors (PV). It should be mentioned that a population size of 50 was also tested in PBIL and it was found that it yielded similar results to the population size of 100. However, in this work a population of 100 was used.

7.2 Conventional PSS

The parameters of the conventional PSS (CPSS) were tuned at the nominal operating condition using the phase compensation method and trial and error approach. Details of this approach can be found in [1–3].

8. Simulation results

8.1 Fitness values

Figures 3–6 show the fitness value (minimum damping ratio) of the system when GA, BGA, PBIL, and DE are used in the optimization. The final value obtained from the GA optimization is 0.1867 as compared to 0.205, 0.2095, and 0.227 for BGA, PBIL, and DE, respectively. As discussed previously, GA and BGA were run for 120 generations, DE for 180 generations, while the PBIL was run for 500 generations. Since a smaller population was used for DE, it was decided to increase its generations. The reason for using 500 generations in PBIL is that it starts to settle only around 300 generations and therefore there is a need for a longer simulation period.

8.2 Eigenvalue analysis

Table 4 shows the inter-area modes for the system with the PSSs. It can be seen that with the PSSs, the inter-area modes are very well damped as compared to the open-loop system in Table 2. CPSS performs adequately for the nominal operating condition. The damping ratios provided by the CPSS under the three cases 1, 2, and 3, are 0.1666, 0.1442, and 0.1339, respectively. BGA-PSS provides a damping ratio of 0.2321, 0.2393, and 0.2412 for cases 1, 2, and 3, respectively. On the other hand, the

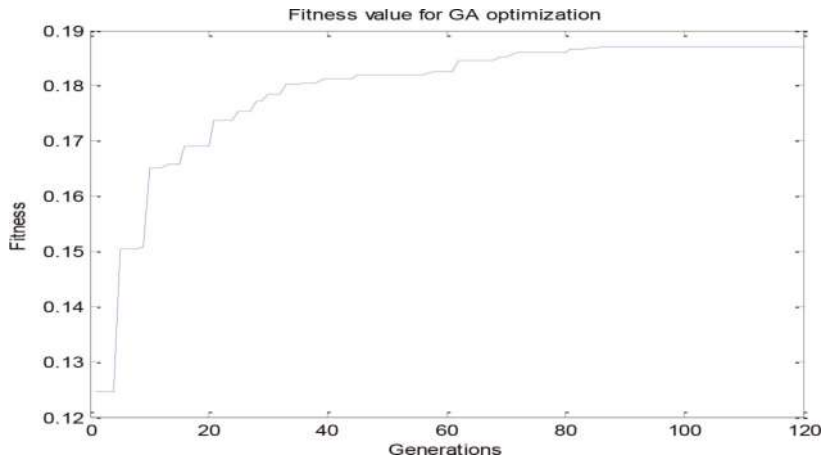


Figure 3.
Fitness value curve from the GA optimization.

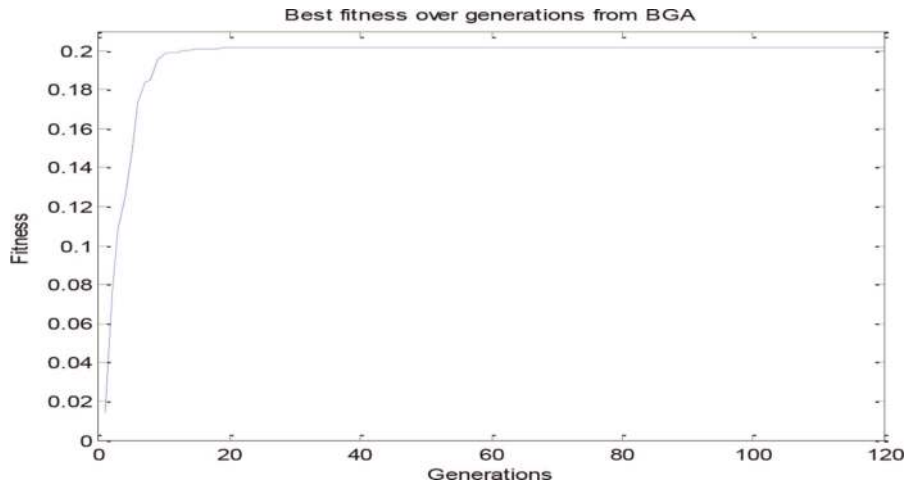


Figure 4.
Fitness value curve from the BGA optimization.

PBIL-PSS and DE-PSS provide a damping ratio of 0.2341 and 0.2377, respectively, for case 1; 0.2387 and 0.2321, respectively, for case 2; 0.2385 and 0.23, respectively, for case 3.

It is observed that PBIL-PSS, DE-PSS, and BGA-PSS provide similar damping ratios to the system for operating condition considered. In case 1, DE provides the best damping ratio, whereas BGA provides the best damping ratios for cases 2 and 3. Among the evolutionary algorithm-based PSSs, GA-PSS provides the lowest damping ratios of 0.2029, 0.2013, and 0.1993 for cases 1, 2, and 3, respectively.

Figure 7 shows the spread of the eigenvalues for the system equipped with the different PSSs. CPSS is the lowest compared to the damping provided by all the other EA-based PSSs. It is observed that among the EA-based PSSs, GA-PSS provides the least damping. The damping provided by the PBIL-PSS, BGA-PSS, and DE-PSS is very similar and higher than that provided by GA-PSS.

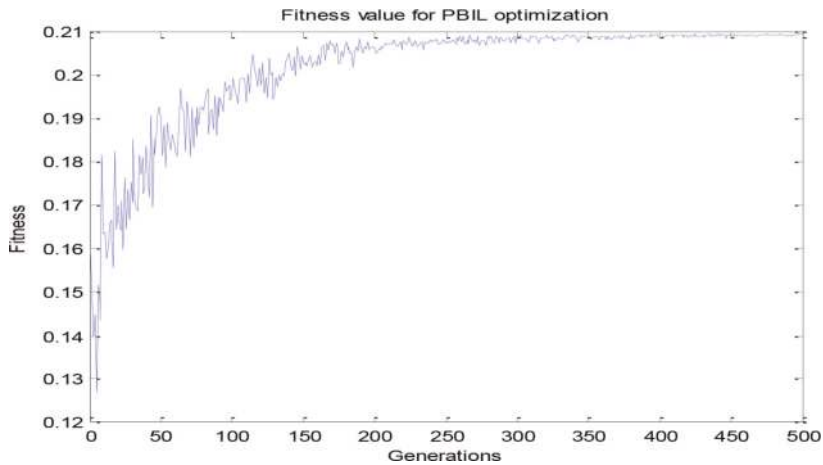


Figure 5.
 Fitness value curve from the PBIL optimization.

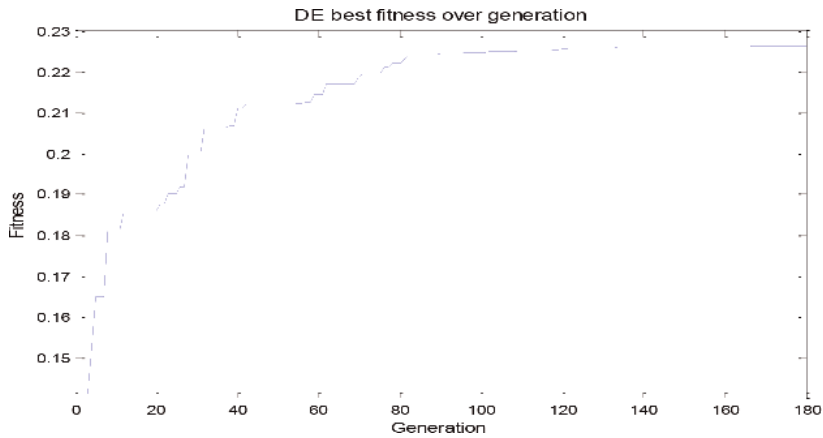


Figure 6.
 Fitness value curve from DE optimization.

Case	CPSS	GA-PSS	BGA-PSS	PBIL-PSS	DE-PSS
1	$-0.62 \pm j3.67$ (0.1666)	$-0.80 \pm j3.86$ (0.2029)	$-0.89 \pm j3.73$ (0.2321)	$-0.91 \pm j3.78$ (0.2341)	$-0.94 \pm j3.84$ (0.2377)
2	$-0.50 \pm j3.43$ (0.1442)	$-0.75 \pm j3.65$ (0.2013)	$-0.86 \pm j3.49$ (0.2393)	$-0.87 \pm j3.54$ (0.2387)	$-0.89 \pm j3.73$ (0.2321)
3	$-0.45 \pm j3.33$ (0.1339)	$-0.72 \pm j3.54$ (0.1993)	$-0.84 \pm j3.38$ (0.2412)	$-0.84 \pm j3.42$ (0.2385)	$-0.87 \pm j3.68$ (0.2300)

Table 4.
 Inter-area modes and the respective damping ratios in brackets.

8.3 Small disturbance

To investigate the performance of the PSSs under small disturbance, a small disturbance of 5% step response is applied to the reference voltage of generator 2 in area 1. The responses of the active power output of generators 2 and 3 are shown in

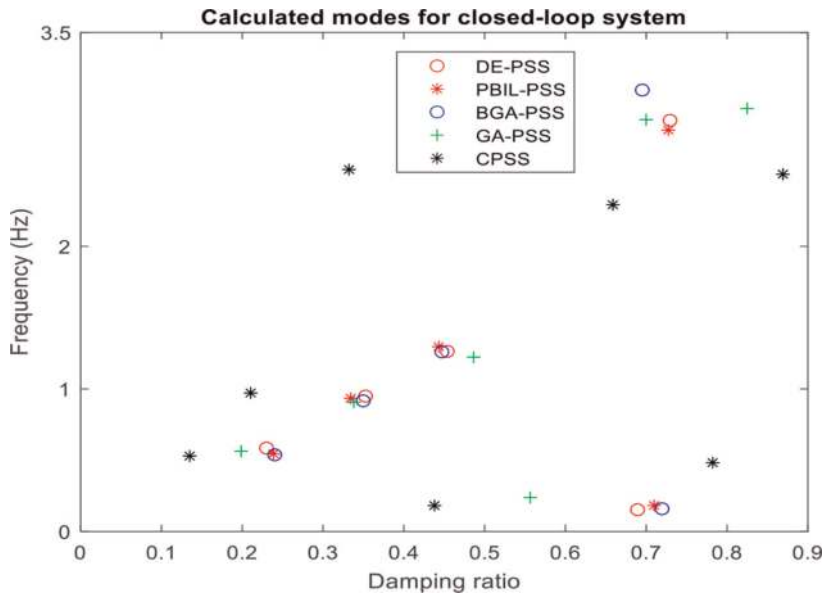


Figure 7.
Spread of the eigenvalues for the different PSSs-nominal condition.

Figures 8–13 for cases 1, 2, and 3, respectively. It can be seen that the system is well-damped across all three operating conditions when it is equipped with DE-PSS, BGA-PSS, GA-PSS, and PBIL-PSS. The CPSS is seen to give the worst performance.

Figures 8 and 9 show the active power output responses of generators 2 and 3, respectively, for case 1. The system equipped with GA-PSS, BGA-PSS, DE-PSS, and

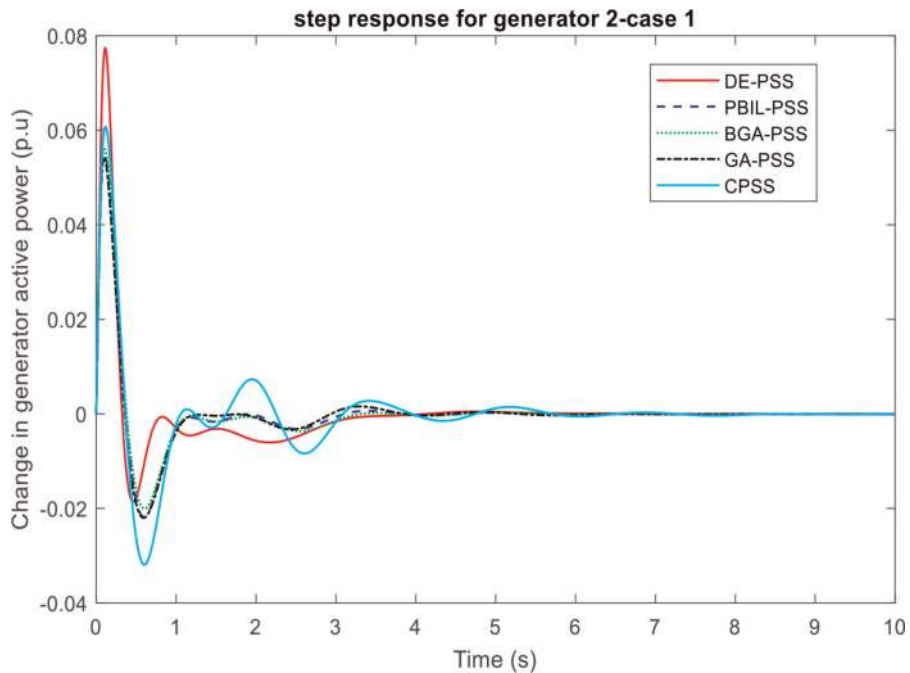


Figure 8.
Response of G2 under the 5% step change in V_{ref} of G2 – Case 1.

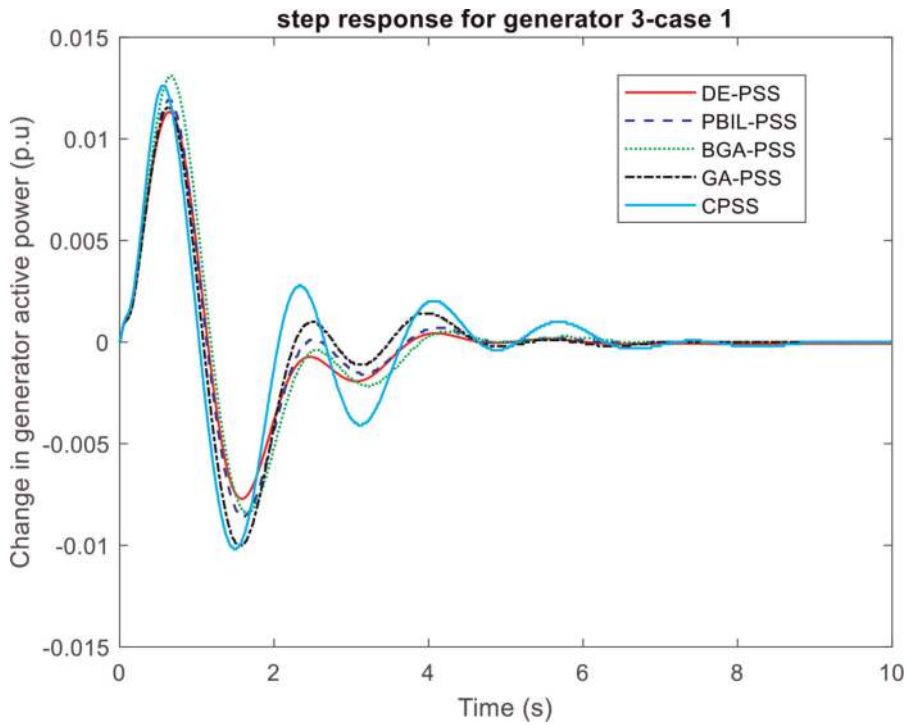


Figure 9.
Response of G_3 under the 5% step change in V_{ref} of G_2 – Case 1.

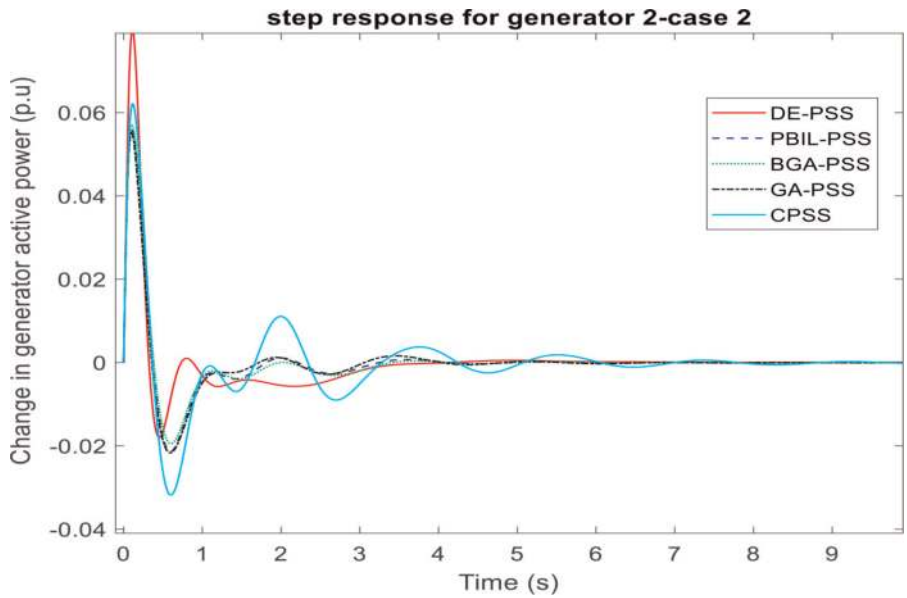


Figure 10.
Response of G_2 under the 5% step change in V_{ref} of G_2 – Case 2.

PBIL-PSS has a similar settling time of approximately 4 sec., whereas the system equipped with CPSS has a longer settling time of around 6 sec. DE-PSS is seen to give the best performance in terms of undershoot and the amplitude of subsequent swings,

albeit with a relatively large 1st swing overshoot as seen in **Figure 8**. It is observed that DE-PSS gives a large 1st swing overshoot in **Figure 8**. The relatively large 1st swing overshoot can be attributed to the high gain of the controller. Note that DE-PSS's gain has almost reached the allowable maximum value [20]. The performance of BGA-PSS is comparable to that of PBIL-PSS. Compared with other EA-based PSS, GA-PSS gives the worst performance. However, it performed better than the CPSS. In **Figure 9**, BGA-PSS is seen to give a slightly high 1st swing overshoot but the subsequent swings are well-damped. Overall, CPSS is seen to give the worst performance.

Figures 10 and **11** show the active power responses of generators 2 and 3, respectively, for case 2. It can be seen that the CPSS has a longer settling time of around 7 sec. Compared to a settling time of around 4 sec. for the EA-based PSSs. This suggests that the oscillations have increased in case 2 compared to case 1. The EA-based PSSs are able to damp the oscillations adequately when compared to the CPSS. In terms of undershoot and subsequent swings, DE-PSS is seen to give the best responses albeit with a relatively large 1st swing overshoot as seen in **Figure 10**. The performances of BGA-PSS and PBIL-PSS are similar. Overall, CPSS gives the worst performance followed by GA-PSS.

Figures 12 and **13** show the active power responses of generators 2 and 3, respectively, for case 3. It can be seen that the system response is similar to case 2 except that the oscillations have now increased as can be seen in the system's responses. The system equipped with the CPSS settled around 10 sec. (see **Figure 13**). It can be seen that the performance of the CPSS has now deteriorated significantly. On the other hand, the performances of GA-PSS, BGA-PSS, PBIL-PSS, and DE-PSS have deteriorated only slightly. This means that the EA-based PSSs are more robust. In terms of settling time, the EA-based PSSs have similar settling times of approximately 6.5 sec., which is comparable to case 2. Although DE-PSS has a larger 1st swing overshoot as seen in **Figure 12**, it gave the best responses in terms of undershoot and subsequent swing amplitudes, followed by BGA-PSS and PBIL-PSS. The performance of GA-PSS although better than that of CPSS is not as good as the other EA-based PSS.

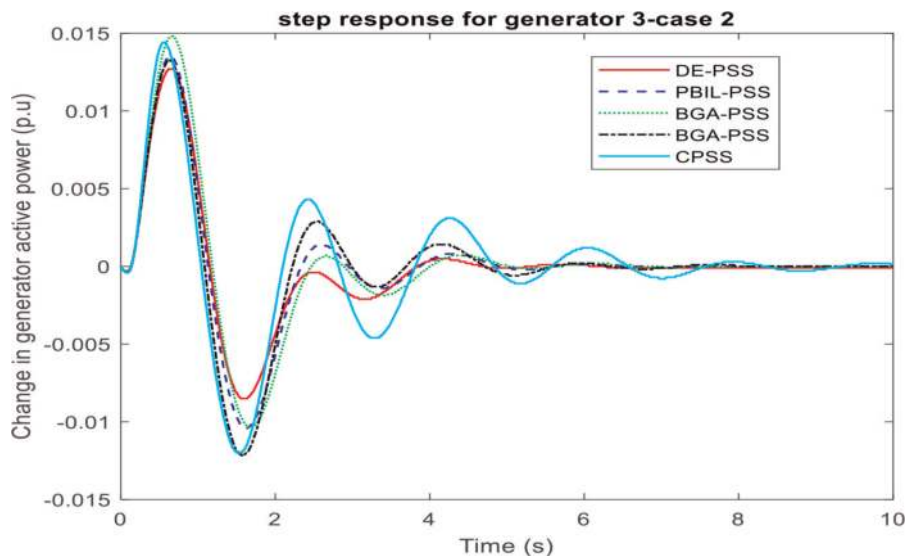


Figure 11.
Response of G_3 under the 5% step change in V_{ref} of G_2 – Case 2.

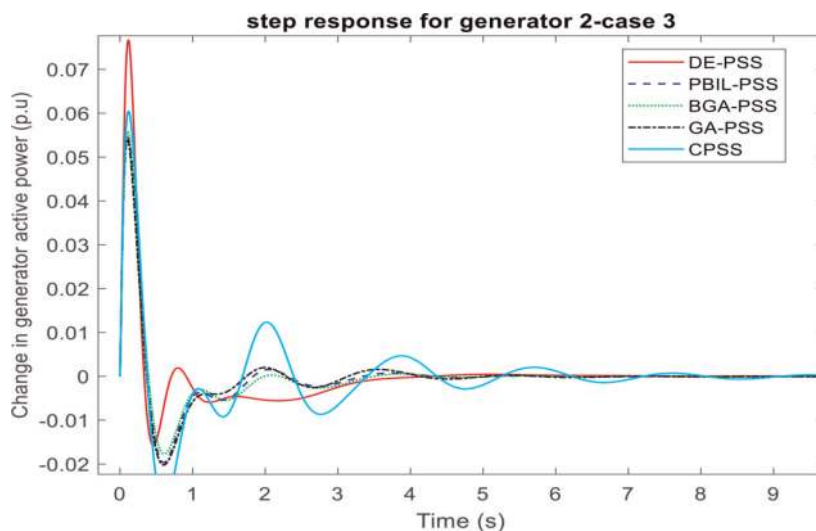


Figure 12.
Response of G_2 under the 5% step change in V_{ref} of G_2 – Case 3.

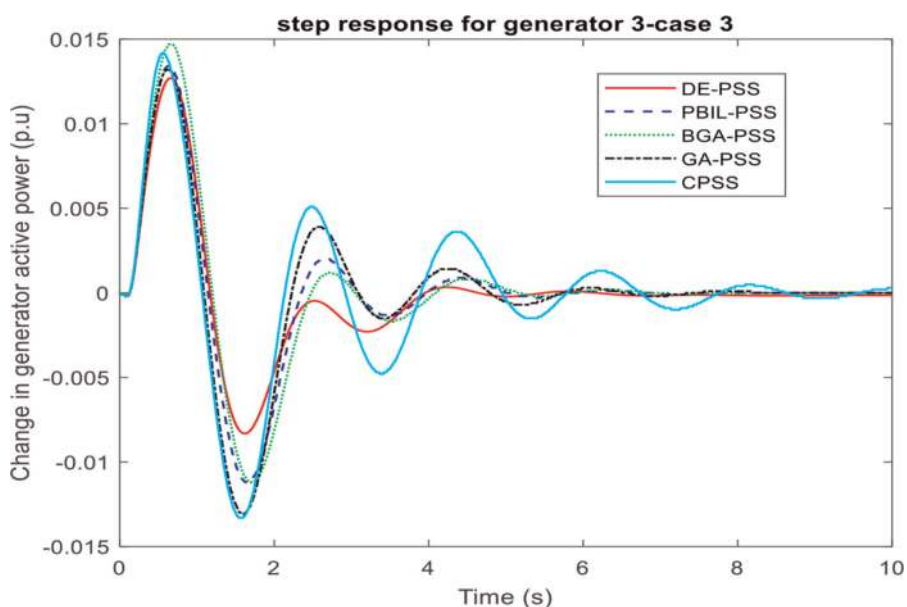


Figure 13.
Response of G_3 under the 5% step change in V_{ref} of G_2 – Case 3.

8.4 Large disturbance

A large disturbance was considered by applying a three-phase fault to the system at bus 3. The fault was cleared by removing one of the transmission lines between bus 3 and bus 13. The fault was applied for 0.1 seconds. After the fault was cleared, the faulted line was removed and the system settled to a different operating condition with only one tie line transmitting power from area 1 to area 2. This means the system is weaker after the fault was cleared compared to its state before the fault. **Figures 14** and **15** show

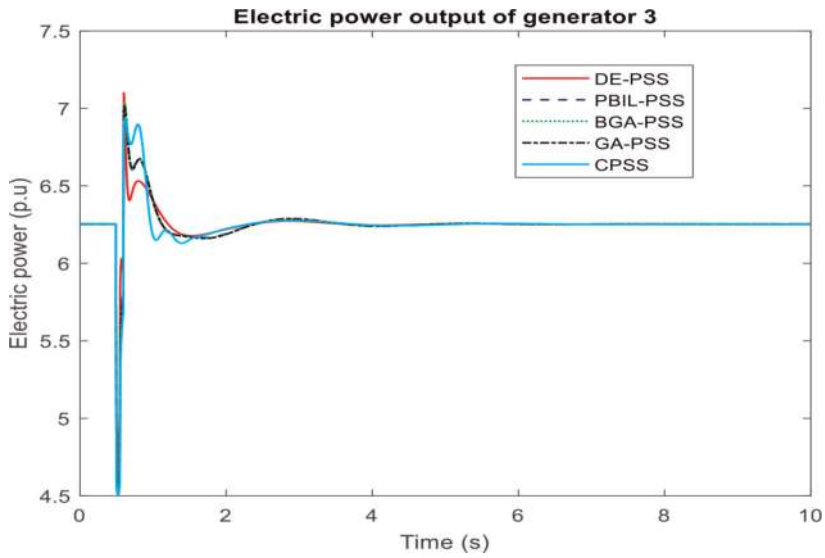


Figure 14.
Electric power output of generator 3 following a three-phase fault on bus 3 for case 1.

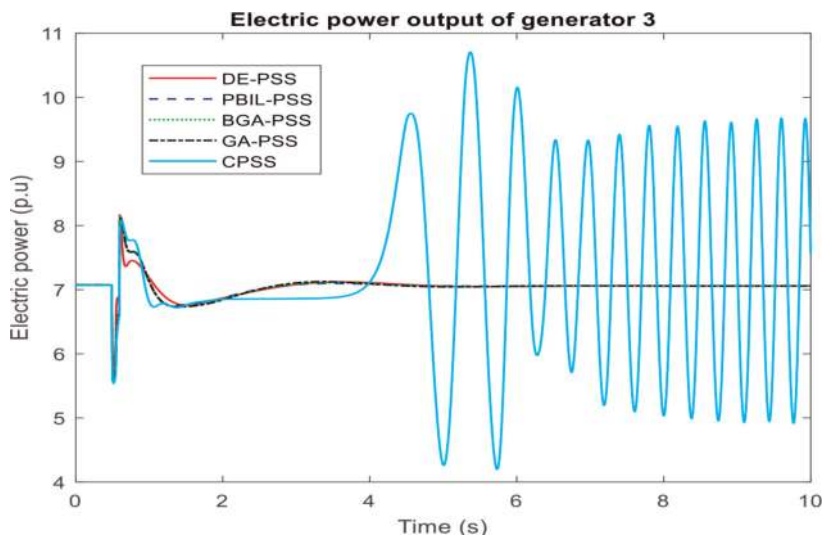


Figure 15.
Electric power output of generator 3 following a three-phase fault on bus 3 for case 2.

the electric power output for generator 3 for case 1 and case 2, respectively. The responses for case 3 are not shown because the system was unable to survive this large disturbance after the fault was removed. It can be seen from **Figure 14** (case 1) that the output power of generator 3 has a high overshoot in the first swing after the fault was cleared but settled down quickly after a few seconds, with all the PSSs providing adequate damping to stabilize the system. However, when the power that was transferred from area 1 to area 2 increased, the CPSS was unable to maintain the stability of the system as seen in **Figure 15** (case 2). On the other hand, all the EA-based PSSs were able to stabilize the system, which suggests that they are more robust than the CPSS.

9. Conclusions

An optimal PSS design for small signal stability improvement of a multi-machine power system using four evolutionary algorithms (GA, BGA, PBIL, and DE) has been presented. Frequency-domain and time-domain simulations have been presented to show the effectiveness of the EA-based PSSs in damping low-frequency oscillations. It is shown that in the frequency domain, the performances of BGA-PSS, PBIL-PSS, and DE-PSS are comparable and better than that of the GA-PSS for all cases investigated. However, time-domain simulations show that DE-PSS performs better than BGA-PSS and PBIL-PSS in terms of undershoot and subsequent swings albeit with a relatively large 1st swing overshoot. This overshoot could be attributed to the high gain of the controller. One way to deal with this overshoot is to reduce the gain of the controller; however, this could also affect the damping. GA-PSS is shown to give the worst performance among the EA-based PSSs, but it performed better than the CPSS. In designing the PBIL-PSS, more generations were required compared to GA-PSS, BGA-PSS, and DE-PSS. Since PBIL works by learning from the previous best individual, it takes time for the algorithm to explore the search space. Compared to the EA-based PSS, the CPSS that was designed using the conventional method has been shown to perform poorly and is not robust. Further research will be done in the direction of improving the EAs algorithms by self-adapting the genetic parameters and using multi-objective functions in the optimization.

Acknowledgements

This research was funded in part by the National Research Foundation (NRF) of South Africa, Grant UID 118550.

Appendix

Generator and automatic voltage regulator (AVR) equations

$$\frac{d}{dt}E_{fd} = \frac{K_A}{T_A}(V_{ref} - V_t) - \frac{E_{fd}}{T_A}$$

where K_A and T_A are the gain and time constant of the AVR. V_t is the terminal voltage of the generator. In this work, $K_A = 200$ and $T_A = 0.05$ sec.

PSS block diagram

where K is the gain of the PSS, T_1 to T_4 are lead/lag time constants, and T_w is the washout time constant. T_1 and T_2 form the first lead/lag block, while T_3 and T_4 form the second lead/lag block of the PSS.

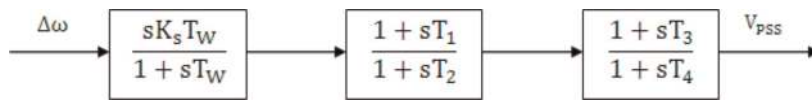



Figure A.1.
PSS block diagram.

Author details

Komla Agbenyo Folly*, Severus Panduleni Sheetekela and Tshina Fa Mulumba
University of Cape Town, South Africa

*Address all correspondence to: komla.folly@uct.ac.za

IntechOpen

© 2022 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Kundur P. Power System Stability and Control. USA: Prentice-Hall; 1994
- [2] Klein M, Rogers GJ, Kundur P. A fundamental study of inter-area oscillations in power systems. IEEE Transactions on Power Systems. 1991; **6**(3):914-921
- [3] Chen L. A Novel Method for Power System Stabilizer Design. Cape Town, South Africa: University of Cape Town; 2003
- [4] Du W, Dong W, Wang Y, Wang H. A method to design power system stabilizers in a multi-machine power system based on single-machine infinite-bus model. IEEE Transaction on Power Systems. 2021;**36**(4):3475-3486. DOI: 10.1109/TPWRS.2020.3041037
- [5] Chow JH, Sanchez-Gasca JJ. Power system stabilizers. In: Power System Modeling, Computation and Control. 2020. pp. 265-294. DOI: 10.1002/9781119546924.ch10
- [6] Folly KA, Yorino N, Sasaki H. Improving the robustness of H_{∞} -PSSs using the polynomial approach. IEEE Transactions on Power Systems. 1998; **13**(4):1359-1364
- [7] Holland JH. Adaptation in Nature and Artificial Systems. Ann Arbor: University of Michigan Press; 1975
- [8] Goldberg DE. Genetic Algorithms in Search, Optimization & Machine Learning. USA: Addison-Wesley; 1989
- [9] Mitchell M. An Introduction to Genetic Algorithms. Cambridge MA, United States: The MIT Press; 1996
- [10] Alkhatib H, Duveau J. Robust design of power system stabilizers using adaptive genetic algorithms. In: Proceeding of the World Academy of Science, Engineering, and Technology. 2010. pp. 267-272
- [11] Sheetekela S. Design of Power System Stabilizer using Evolutionary Algorithms. Cape Town, South Africa: University of Cape Town; 2010
- [12] Mulumba TF, Folly KA. Application of evolutionary algorithms to power system stabilizer design. In: Subair S, Thron C, editors. Implementation and Application of Machine Learning. Studies in Computational Intelligent (SC 782). 2020. pp. 29-62
- [13] Price K, Storn R, Lampinen J. Differential Evolution—A Practical Approach to Global Optimization. Berlin, Germany: Springer; 2005
- [14] Ahmad MF, Isa NAM, Lim WH, Ang KM. Differential evolution: A recent review based on state-of-the-art works. Alexandria England Journal. 2022;**61**: 3831-3872
- [15] Verdejo H, Pino V, Kliemann W, Becker C, Delpiano J. Implementation of particle swarm optimization (PSO) algorithm for tuning power system stabilizers in multi-machine electric power systems. Energies. 2020;**13**(8): 2093. DOI: 10.3390/en13082093
- [16] Folly KA. Performance of power system stabilizers based on population-based incremental learning (PBIL) algorithm. International Journal of Electrical Power and Energy System. 2011;**33**(7):1279-1287
- [17] Folly KA. Parallel PBIL applied to power system controller design. Journal of Artificial Intelligence and Soft

Computing Research. 2013;3(3):
215-223. DOI: 10.2478/jaiscr-
2014-0015

[18] Baluja S. Population-Based
Incremental Learning: A method for
integrating Genetic Search Based
Function Optimization and Competitive
Learning. Technical Report CMU-CS-
49-163, 1994

[19] Baluja S, Caruana R. Removing the
genetics from the standard genetic
algorithm. In: Proceedings of the
12th International Conference on
Machine Learning, Lake Tahoe,
CA; 1995

[20] Sheetekela S, Folly KA.
Multimachine power system stabilizer
design based on evolutionary algorithm.
In: Proceedings of the 44th International
Universities' Power Engineering
Conference. 2009

[21] Sheetekela S, Folly KA.: Breeder
genetic algorithm for power system
stabilizer design. In: Proceedings of 2010
IEEE Congress on Evolutionary
Computation (CEC), Barcelona, Spain;
2010

[22] Mühlenbein H, Schlierkamp-Voosen
D. Predictive models for the Breeder
Genetic Algorithm, I. continuous
parameter optimization. *Evolutionary
Computation*. 1993;1(1):25-49

[23] Greene J. The Basic Idea behind the
Breeder Genetic Algorithm. Cape Town,
South Africa: University of Cape Town;
2005

[24] Folly KA, Sheetekela SP. Optimal
design of power system controller using
breeder genetic algorithm. In: Gao S,
editor. *Bio-Inspired Computational
Algorithms and Their Applications*.
InTech-open science; 2012. pp. 303-316.
DOI: 10.5772/38447

[25] Das S, Suganthan PN. Differential
evolution: A survey of the state-of-the-
art. *IEEE Transaction on Evolutionary
Computation*. 2011;15(1):4-31