Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Damping Controller Design Using Self-Adaptive DE

Komla A. Folly*, T. Mulumba*

**Department of Electrical Engineering, University of Cape Town, Private Bag, Rondebosch 7701 South Africa (e-mail: Komla.Folly@uct.ac.za, Famutshi@gmail.com).*

Abstract: Differential Evolution (DE) is one of the most powerful stochastic real parameter, population-based optimization algorithms. It is similar to Genetic Algorithms (GAs), except that it uses differential mutation technique as the main operator to arrive at the best trial solution. The performance of DE is sensitive to the choice of the mutation and crossover strategies and their associated control parameters such as the amplification factor *F* and the crossover rate *CR*. Inappropriate choice of control parameter values may result in significant deterioration of the performance of the algorithm. In this paper, a self-adaptive DE is applied to design a controller for damping low frequency oscillations in a power system. In self-adaptive DE, the control parameters such as the mutation scale factor *F* and crossover rate *CR* are adapted as the population evolves. The performance of the proposed self-adaptive DE-PSS (jDE-PSS) is compared with that of the classical DE-PSS (CDE-PSS). Simulation results show that for damping low-frequency oscillations, jDE-PSS gives the best performance.

1. INTRODUCTION

In the last three decades, there has been a growing interest in applying Evolutionary Algorithm (EA) to solve optimization problems. Until now, Genetic Algorithms has been the most used EA (Abdel *et al*, 1999), (Bomfim *et al*, 2000), (Davis, 1996), (Goldberg,1989), (Michalewicz, 1996). Recently, Differential Evolution (DE) has received increasing attention due to its simplicity and its straightforward strategy (Mulumba *et al*, 2011), (Price *et al*, 2005), (Shayeghi *et al*, 2008). This algorithm was first proposed by Price and Storn (Price, and Storn, 1997). It is a stochastic population-based optimization that uses differential mutation technique as the main operator to arrive at the best results *(Ali et al, 2009), (Price et al, 2005)*. However, the performance of DE is sensitive to the choice of the mutation and crossover strategies and their associated control parameters such as the amplification factor *F* and the crossover rate *CR* (Suganthan and Quin, 2005), (Tang, *et al*, 2008). Inappropriate choice of control parameter values may result in significant deterioration of the performance of the algorithm (Brest *et al*, 2006). In general, DE users select the initial parameter settings for the problem at hand from previous experience or from the literature. Then trial-and-error method is used to fine tune these parameters. However, with the trial and error approach there is no guaranty that the best parameters will be obtained (Zhang and Sanderson, 2009); moreover, in some cases, the time for finding the parameters is unacceptably long. In the last few years, several researchers have proposed methods to make the control parameters of DE adaptive (Suganthan and Quin, 2005), (Tang, *et al*, 2008), (Zhang and Sanderson, 2009).

In this paper, we explore the idea of self-adaptive DE and applied it to design a power system controller (also known as Power System Stabilizer-PSS) to damp low frequency oscillations in a power system. Low frequency oscillations arise because of heavy transfer of power over long distance. In the self-adaptive DE used in this paper, the control parameters such as the mutation scale factor *F* and crossover rate *CR* are adapted as the population evolves (Brest *et al*, 2006).

The effectiveness of the proposed self-adaptive DE-PSS (jDE-PSS) is assessed by comparing its performance with that of the classical DE-PSS (CDE-PSS). Simulation results show that jDE-PSS is more effective than CDE-PSS in damping the low-frequency oscillations.

2. BACKGROUND OF DE

2.1 Overview

DE is a parallel direct search method that uses a population of points to search for a global optima of a function over a wide search space (Price *et al*, 2005). Like GAs, DE is a population based algorithm that uses operators such as crossover, mutation and selection to generate successive populations that we hope will improve over time (Davis, 1996), (Ali *et al*, 2009). The main differences between the two search methods are briefly summarized below (Mulumba *et al*, 2011), (Price *et al*, 2005):

- GAs rely on the crossover to be able to explore the search space and escape from local optima. DE on the other hand, relies on the mutation parameters (i.e., *F*) as a search mechanism and the selection operation to direct the search toward the prospective regions in the search space.
- In DE, all solutions have the same chance of being selected as parents regardless of their fitness values.

Some of the features of DE are: ease of use, efficient memory utilization, lower computational complexity.

2.2 DE Operator

In DE, the population is constituted of N_p candidates solutions. Each candidate is a D dimensional real–valued vector, where D is the number of parameters.

The summary of DE's operation is as follows *(Price et al, 2005), (Mulumba et al, 2011)*

- *Step 1(Initialization)*: DE generates N_p vectors candidates $x_{i,g}$, where *"i"* represents the vector and *"g"* the generation. The i[th] trial solution can be written as $x_{i,g}=[z_{j,i,g}]$ where $j=1,2,...,D$. The vector's parameters are initialized within the specified upper and lower bounds of each parameter $Z_j^L \leq z_{j,i,g} \leq Z_j^U$.
- *Step 2 (Mutation)*: In this process, four vectors are randomly selected from the initial population,where one is chosen as the *target vector*, and another is selected as the *base vector*. The difference of the remaining two vectors, is scaled by a factor F and is added to the base vector to form the mutant vector. This is the most popular strategy called DE/rand/1/. For a given parameter vector $x_{i,g}$, we randomly select three vectors $x_{r0,g}$, $x_{r1,g}$, and $x_{r2,g}$, such that the indices i, r_0, r_1, and r_2 are distinct. The equation below shows how mutant vectors are created.

$$v_{i,g} = x_{r0,g} + F.(x_{r_1,g} - x_{r_2,g}) \quad (1)$$

where $v_{i,g}$ is the mutant vector (or donor vector) and the base vector is denoted by $x_{r0,g}$. The indices r_0, r_1, and r_2 are mutually exclusive integers randomly chosen from the range [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 (Price, and Storn, 1997).

- *Step 3 (Recombination or crossover):* In this stage DE crosses each vector with a mutant vector, as in (2), to form a trial population. This is a binomial crossover. The purpose is to enhance the potential diversity of the population.

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

where

$CR \in [0, 1]$ is the crossover probability defined by the user. $j_{rand} \in [1, 2, \ldots, D]$ is randomly chosen index, which ensures that $\boldsymbol{u}_{i,g}$ gets at least one component from $\boldsymbol{v}_{i,g}$. If the randomly generated value between 0 and 1 is less than CR, the parameters of the trial solutions are copied from the mutants $v_{j,i,g}$, otherwise, they are copied from the target vector.

- *Step 4 (Selection):* The selection of vectors to populate the next generation is accomplished by comparing each vector $\boldsymbol{u}_{i,g}$ of the trial population to its target vector $x_{i,g}$ from which it inherits parameters. The values of the vectors are obtained using the function in (3)

$$\mathbf{x}_{\mathbf{i,g+1}} = \begin{cases} \mathbf{u}_{\mathbf{i,g}} & \text{if } f(\mathbf{u}_{\mathbf{i,g}}) \leq f(\mathbf{x}_{\mathbf{i,g}}) \\ \mathbf{x}_{\mathbf{i,g}} & otherwise \end{cases} \quad (3)$$

In the above, we assume the minimization of a function. As soon as the new population is installed, the cycle from step 2 to step 4 is repeated until the optimum is located or the termination criterion is satisfied.

It should be mentioned that the values of DE control parameters F, CR can have a significant impact on the performance of the algorithm. In general, the selection of the parameters is done using trial-and-error method, which in many cases is time consuming. The best way to deal with this problem would be to make the control parameters of DE adaptive. That is, the values of the parameters F, CR are changed during the run (Suganthan and Quin, 2005), (Tang, *et al*, 2008), (Zhang and Sanderson, 2009)

One of the most attractive approaches is to make the parameters self-adaptive by encoding them into the chromosome (individuals) so that they to undergo the actions of genetic operators and evolve with the individuals (Brest *et al*, 2006). The best of these parameters will lead to better individuals which in turn are more likely to survive and produce better offspring solutions. Below, we will discuss the self-adaptive DE approach used in this paper.

3. BACKGROUND OF SELF-ADAPTIVE DE

DE's ability to find the global maximum is mainly dependent on the mutation and crossover process. The differential mutation allows DE to explore the search space for the global maximum or minimum. This process is controlled by the mutation scale factor $F \in [0\ 2]$. 'F' controls the rate at which the population evolves. On the other hand, the crossover ensures that the diversity of population is maintained so as to avoid premature convergence. Hence this process is directly dependent on the crossover constant 'CR'.

A modified version of the self-adaptive DE (jDE) proposed by (Brest *et al*, 2006) which uses a strategy based on DE/rand/1/bin is used in this paper. The strategy proposed here is the DE/rand/2/ bin instead of the DE/rand/1/bin. The population size is fixed during the optimization whilst adapting the control parameters F_i and CR_i associated with each individual. Each individual in the population is extended with parameter values as shown in Fig.1. In other words, the control parameters that are adjusted by means of evolution are F and CR. The initialization process sets $Fi = 0.5$ and $CRi = 0.9$ for each individual. jDE regenerates (with probabilities $\tau_1 = \tau_2 = 0.1$ at each generation) new values for Fi and CRi

according to uniform distributions on [0.1,1] and [0,1], respectively. The following (4)-(5) are used to update F and CR:

$$F_{i,g+1} = \begin{cases} F_l + \text{rand}_1 * \text{F}_\text{u} & \text{if } rand_2 < \tau_1 \\ F_{i,g} & otherwise \end{cases} \quad (4)$$

$$CR_{i,g+1} = \begin{cases} \text{rand}_3 & \text{if } rand_4 < \tau_2 \\ CR_{i,g} & otherwise \end{cases} \quad (5)$$

where rand$j$, j = 1, 2, 3, 4, are uniform random values on [0, 1], and $\tau_1 = \tau_2 = 0.1$ represent the probabilities to adjust the control parameters. The newly generated parameter values are used in the mutation and crossover operations to create the corresponding offspring vectors that will replace the previous parameter values if the offspring survive in the selection. It is believed that better parameter values tend to generate individuals which are more likely to survive, and thus the newly generated better values are able to go into the next generation.

The DE/rand/2 mutation strategy adopted in this paper is given below

$$v_{i,g} = x_{r0,g} + F.(x_{r_1,g} - x_{r_2,g}) + F.(x_{r_3,g} - x_{r4,g}) \quad (6)$$

where, for a given parameter vector $x_{i,g}$, we randomly select five vectors $x_{r0,g}$, $x_{r1,g}$, $x_{r2,g}$, $x_{r3,g}$, and $x_{r4,g}$, such that the indices i, r_0, r_1, r_2, r_3, and r_4 are distinct.

$\vec{x}_{1,G}$	$F^1_{1,G}$	$CR^1_{1,G}$	$F^2_{1,G}$	$CR^2_{1,G}$
$\vec{x}_{2,G}$	$F^1_{2,G}$	$CR^1_{2,G}$	$F^2_{2,G}$	$CR^2_{2,G}$
...	...	...	...	...
$\vec{x}_{NP,G}$	$F^1_{NP,G}$	$CR^1_{NP,G}$	$F^2_{NP,G}$	$CR^2_{NP,G}$

Fig. 1. Encoding of the chromosomes of self-adaptive DE.

4. SYSTEM MODEL AND OPERATING CONDITIONS

The power system considered in this paper is a single machine infinite bus (SMIB). The generator is connected to the infinite bus through a double transmission line. The non-linear differential equations of the system are linearized around the nominal operating condition to form a set of linear equations (Mulumba *et al*, 2011). The generator is modeled using a 6th order machine model, whereas the AVR was represented by a simple exciter of first order differential equation (Kundur,1994).

The system is represented by a set of linear equations as follows:

$$\begin{cases} \dfrac{dx}{dt} = Ax + Bu \\ y = Cx + Du \end{cases} \quad (7)$$

where:
A is the system state matrix
B is the system input matrix
C is the system output matrix
D is the feed forward matrix
x is the vector of the system states
u is the vector of the system inputs
y is the vector of the system outputs

For the design of the controller, several operating conditions were considered. These operating conditions were obtained by varying the active power output and the reactive power of the generator as well as the line reactance to represent the uncertainties due to varying operating conditions in the system model. For simplicity, four operating conditions are presented in this paper as listed in the Table 1. This table shows the operating conditions with the open loop eigenvalues and their respective damping ratios in brackets.

Table 1. Selected Operating Conditions

Case	Active Power P_e (p.u)	Line Reactance X_e (p.u)	Eigenvalues (ζ)
1	1.000	0.300	-0.268 ± j4.457 (0.060)
2	1.000	0.500	-0.118 ± 3.781i (0.048)
3	1.000	1.700	-0.133 ± 3.311i (0.040)
4	1.000	0.900	-0.0997 + 2.947 (0.034)

5. FITNESS FUNCTION

The fitness function is used to provide the measure of how individuals performed. In this instance, the problem is to tune the PSS parameters such that the PSS is able to stabilize a set of plant modes simultaneously over a certain range of specified operating conditions. Therefore, the PSS parameters are optimized simultaneously. The block diagram of the PSS has the form given in the Appendix (Kundur, 1994). There are five parameters that need to be optimized: the PSS gain gain K, and the time constants T_1, T_2, T_3 and T_4. Since the washout value is not critical, it was not optimized and was set to 5 sec.

The fitness function that was used is given as follows:

$$J = \max(\min(\zeta_{ij})) \quad (8)$$

$$i = 1,2,3, \ldots, n \quad eigenvalues$$

j=1,2,3, …, *m* *operating conditions*

Where $\zeta_{ij} = \dfrac{-\sigma_{ij}}{\sqrt{\sigma_{ij}^2 + \omega_{ij}^2}}$ is the damping ratio of the *ith* closed – loop eigenvalue of the *jth* operating condition. σ_{ij} is the real part of the eigenvalue and ω_{ij} is the frequency.

6. CONTROLLER DESIGN

6.1 Application of the Conventional DE

The configuration that was used for CDE is as follows:

Population: 30
Generation: 100
Mutation scale factor *F*: 0.9
Crossover *CR*: 0.9

Note that the values of parameters *CR* and *F* as given above were obtained using trial and error method. These values give the best possible performance of the DE algorithm for the problem at hand.

6.2 Application of Self-adaptive DE

The configuration that was used for *jDE* is as follows

Population: 30
Generation: 100
Mutation scale factor *F*: Adaptive
Crossover *CR*: Adaptive

The parameter domain for both CDE and *j*DE are:

$$5 \le K_p \le 20$$
$$0 \le T_1,\ T_3 \le 1$$
$$0.015 \le T_2,\ T_4 \le 0.5$$

The parameters of the designed controllers are shown in Table 2. It should be mentioned that the algorithm selects automatically the optimal gain and time constants of the controllers based on the best objective function. It can be seen from table 2 that the gain of jDE is slightly higher than that of CDE. Furthermore, the time constants T_1-T_4 are higher for *j*DE than for CDE.

Table 2. Parameters of the designed controllers

Controller	K	T_1	T_2	T_3	T_4
CDE-PSS	17.20	0.010	0.15	4.84	0.27
jDE-PSS	18.90	4.640	1.67	3.21	1.50

Also the ratio T_1/T_2 is approximately 0.065 for CDE but for jDE it is approximately 2.78, whereas the ratio T_3/T_4 is about 17.79 for CDE and approximately 2.14 for jDE. Therefore the overall control effort needed for CDE is higher than that needed for jDE to achieve the same performance.

7. SIMULATION RESULTS

7.1 Eigenvalue Analysis

Table 3 shows the closed-loop eigenvalues and damping ratios in brackets. It can be seen that jDE-PSS gives the best damping ratio under all operating conditions considered. It is also observed that as the system becomes weaker (i.e., line reactance bigger), the performance of CDE is deteriorating. On the other hand, the damping provided by j*DE* is consistent under all operating conditions. Therefore, jDE can be considered to be more robust than CDE.

Table 3. Closed-loop eigenvalues and damping ratios

Case	CDE-PSS	jDE-PSS	No PSS
1	-1.52 ± j3.41 (0.410)	-1.92±j3.98 (0.430)	-0.268 ± *j*4.457 (0.060)
2	-1.13±j2.74 (0.380)	-1.57±j3.23 (0.440)	-0.118 ± 3.781i (0.048)
3	-0.83±j2.32 (0.330)	-1.34±j2.69 (0.450)	-0.133 ± 3.311i (0.040)
4	-0.49±j1.69 (0.280)	-1.16±j2.25 (0.460)	-0.0997 + 2.947 (0.034)

7.2 Small Disturbance

A small disturbance was simulated by applying a 10% step change in the reference voltage. The step responses for speed deviation of the generator are presented in Figs. 2-5.

Fig. 2 shows the responses of the rotor speed deviations for case 1. It can be seen that all controllers are able to damp the oscillations and improve the stability of the system. However, jDE-PSS has a slightly higher overshoot and undershoot but settles within 2.5 sec. as compared to CDE-PSS which settled in about 3 sec.

Figure 3 shows the responses for case 2. It is observed that the speed deviation of jDE-PSS displays a better performance in terms of settling time than that of CDE-PSS; however, the overshoot is slightly higher for jDE-PSS than for CDE-PSS.

Figs. 4-5 show the speed responses of the system for cases 3 to 4, respectively. In both cases, jDE-PSS gives relatively large undershoots than CDE-PSS, this can be due to the slightly higher gain of jDE-PSS as discussed previously. However, jDE-PSS provides the best performance in terms of settling time. In particular in case 4, where the system is weaker than the previous cases, jDE-PSS settled quicker (in about 5.5 sec) compared to CDE which settled in about 10 sec.

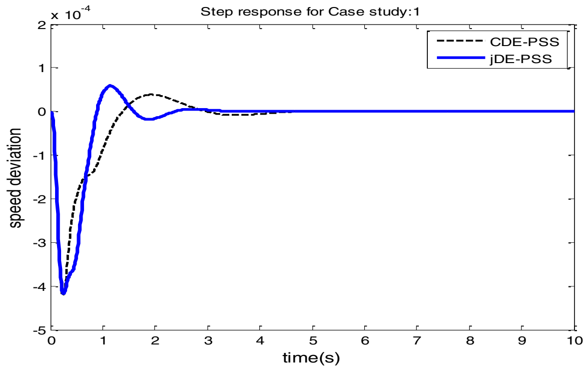

Fig.2 Speed deviations for a step response (case 1)

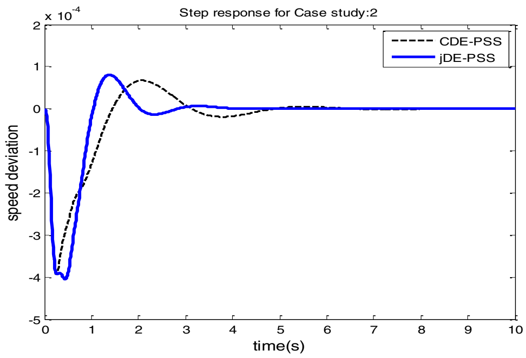

Fig.3 Speed deviations for a step response (case 2)

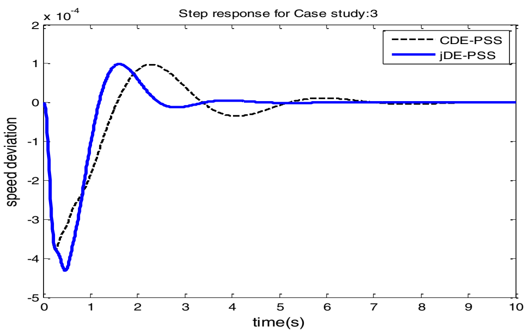

Fig.4 Speed deviations for a step response (case 3)

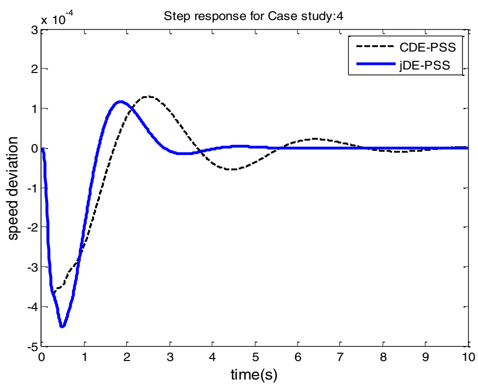

Fig.5 Speed deviations for a step response (case 4)

7.3 Nonlinear Simulations under Large Disturbance

A large disturbance was simulated by applying a three-phase fault on one of the transmission lines. The fault was cleared by disconnecting the line after 5 cycles. Because of the severity of the fault, case 4 was not considered in the simulations. For simplicity, only some of the simulations related to cases 2-3 will be shown here.

Fig. 6-9 show the rotor angle, rotor speed, terminal voltage of the generator, and the exciter field voltage responses, respectively for case 2. It can be seen that although both the CDE-PSS and the jDE-PSS are able to stabilize the system, overall, *j*DE-PSS gives the best performance in terms of overshoots, undershoots and settling time. The large control effort needed by CDE-PSS is evident in the field voltage as depicted in Fig. 9.

Fig. 10-13 show the rotor angle, rotor speed, terminal voltage of the generator, and the exciter field voltage responses, respectively for case 3. Again, jDE-PSS is seen to give a better performance in terms of overshoots, undershoots and settling time. CDE-PSS required a large control effort in terms of the field voltage to provide similar performance as as depicted in Fig. 13.

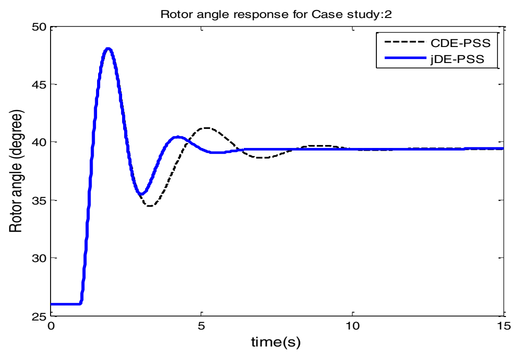

Fig. 6 Rotor angle responses following a three-phase fault (case 2)

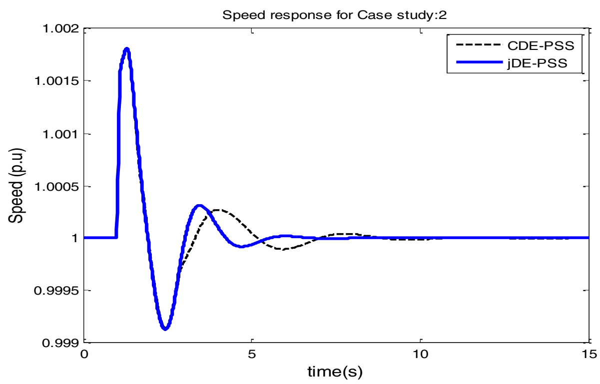

Fig. 7 Rotor speed responses under three-phase fault (case 2)

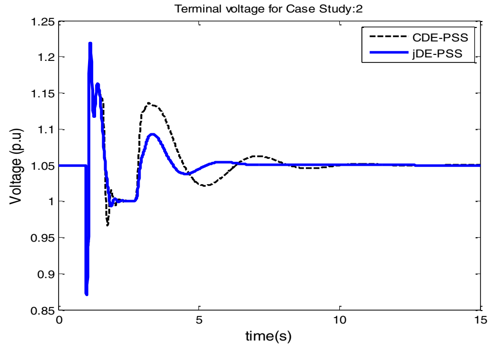

Fig. 8 Terminal voltage responses under three-phase fault (case 2)

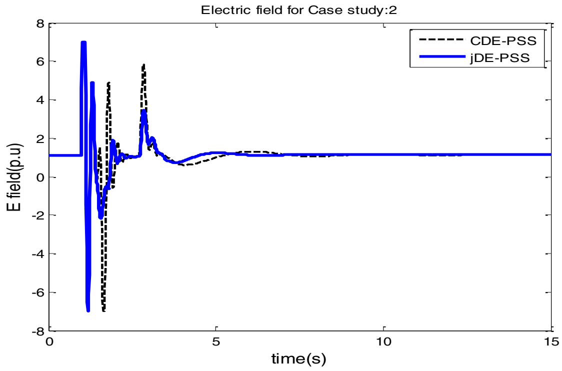

Fig. 9 Exciter field voltage responses under three-phase fault (case 2)

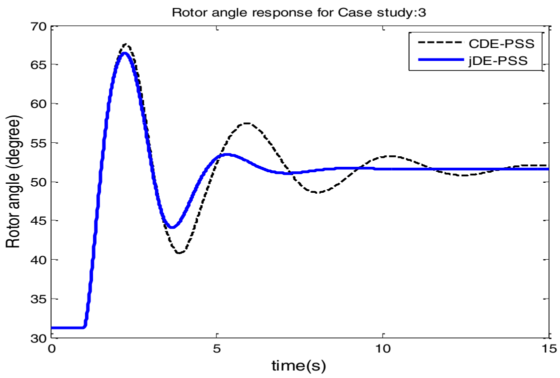

Fig. 10 Rotor angle responses following a three-phase fault (case 3)

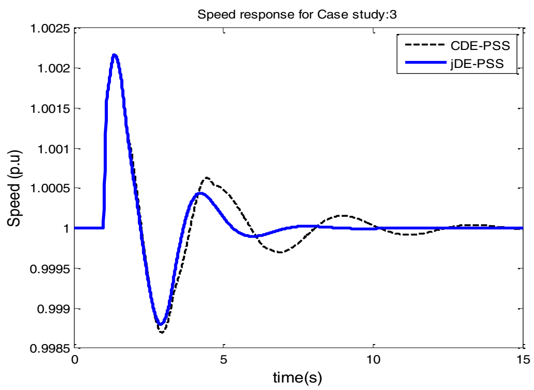

Fig. 11 Rotor speed responses under three-phase fault (case 3)

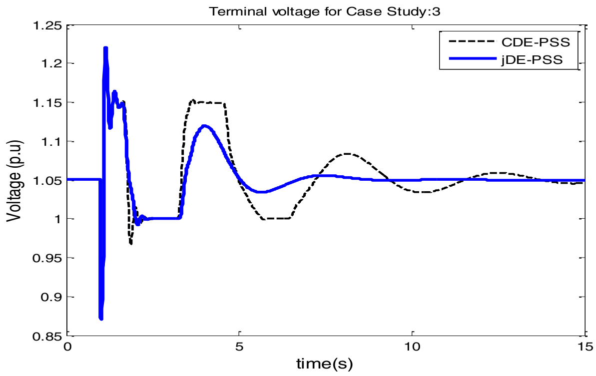

Fig. 12 Terminal voltage responses under three-phase fault (case 3)

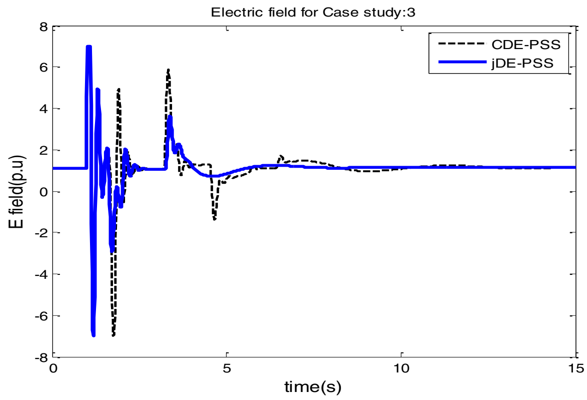

Fig. 13 Exciter field voltage responses under three-phase fault (case 3)

8. CONCLUSIONS

In this paper, self-adaptive DE is used to design controller for damping low frequency oscillations in a power system. It is shown that there are clear advantages in using self-adaptive DE as compared to the conventional DE. Firstly, the time consuming trial-and-error approach is removed and secondly, there is a high possibility that the algorithm will converge to optimal values. Results based on eigenvalue analysis and time domain simulations show that under small disturbance, the self-adaptive DE performs better than the classical DE. These results were confirmed by nonlinear simulations based on large disturbance. Work is in progress to extend the self-adaptive DE approach to controller design in multi-machine power system in the future.

ACKNOWLEDGEMENT

This work is based on research supported in part by the National Research Foundation of South Africa UID 71967.

REFERENCES

Abdel, Y.L. Abido, M. A. and H. Mantawy (1999). Simultaneous Stabilization of Multimachine Power Systems via Genetic Algorithms. *IEEE Trans on Power Systems*, **Vol. 14**, No. 4, pp. 1428-1438.

Baluja, S. (1994). Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Carnigie Mellon University, Technical report CMU-CS-94-163.

Bomfim, L.B. Taranto, G. N. and Falcao, D. M. (2000). Simultaneous Tuning of Power System Damping Controllers using Genetic Algorithm. *IEEE Trans. on Power Systems*, **Vol. 15**, No.1, pp. 163-169.

J. Brest, J.S. Greiner, B. Boskovic, M. Mernik, and V. Zumer (2006). Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. *IEEE Trans. on Computation Evol.*, **Vol. 10**, No. 6, pp. 646-657.

Davis L. (1996). *Handbook of Genetic Algorithms,* International Thomson Computer Press.

Goldberg, D. E. (1989). *Genetic Algorithms: in Search, Optimization & Machine Learning*, Addition Wesley Inc.

Kundur,P. (1994). *Power System Stability and Control*, McGraw-, Hill, Inc.

Lui, J. and J. Lampinen, J. (2005). A Fuzzy adaptive differential evolution algorithm. *Soft Compu. A Fusion Founda Methodol. Applicat.*, No. 6, **Vol. 9**, pp. 448-462.

Michalewicz Z. (1996). *Genetic Algorithms+Data Structure=Evolution Programs*, 3rd ed., Springer-Velag.

Mulumba, T., Folly, K.A., and Malik, O. P. (2011). Tuning of PSS parameters using Differential Evolution. In: *Proc. of the 20th Southern African Universities' Power Engineering Conference- SAUPEC 2011*, Cape Town, South Africa

Price K.V., and R. Storn R. (1997), "Differential evolution: A simple evolution strategy for fast optimization," *Dr. Dobb's J.*, **Vol. 22**, no. 4, pp. 18–24.

Price, K., Storn, R. and Lampinem, J. (2005). *Differential Evolution: A practical Approach to Global Optimization*, Berlin Heldelberg: Springer-Verlag.

Shayeghi, H., Safari, A., and Shayanfar, H.A. (2008). Multimachine Power System Stabilizers Design using PSO Algorithm. *International Journal of Elec. Power and Energy Syst. Eng*. **Vol. 1**, pp. 226-233.

Suganthan, P. N., and Quin, A. K. (2005). Self-Adaptive Differential Evolution Algorithm for numerical optimization. In: *Congres on Evolutionary Computation*, pp. 1785-1791

Tang, K., Yao, X. and Yang, Z. (2008). Self-Adaptive with neighborhood search. In: *Congress on Evolutionary Computation*, pp. 1110-1116.

Zhang, J and Sanderson, A.C. (2009). *Adaptive Differential Evolution: A robust Approach to Multimodal Problem Optimization*. Berlin Heidelberg: Springer.

Ali, M., Plant, M. and Abraham, A. (2009). Simplex Differential Evolution. *Acta Polytechnica Hungarica*, **Vol. 6**, No. 5, pp. 95-115.

Appendix A. BLOCK DIAGRAM OF THE PSS

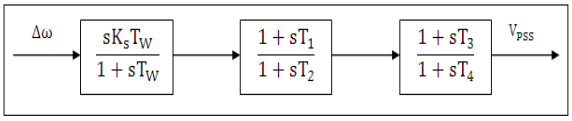

Fig. A1 PSS block diagram