Volume minimization in small scale wind turbine PMSG using Dynamic PSO and DE algorithms

[1]N.A. Prashanth, [1]Research Scholar, JNTUACE, Ananthapuramu
[2]Dr. P. Sujatha [2]Professor, JNTUACE, Ananthapuramu.

Abstract: The wind turbine generators employed in modern wind power plants should have the characteristics such as high efficiency, power density and reliability, low weight and volume. Among different generators, permanent magnet synchronous generator adequately satisfied these requirements. In this paper permanent magnet synchronous generator has been designed to meet the desired output power for direct driven small scale wind turbine. Design care has also been taken to minimize the overall volume of generator to reduce the cost. A minimization optimization with constraint has been proposed to model the problem. Particle swarm optimization and Differential evolution algorithms have been considered separately as methods to estimate the optimal values of design parameters. In this comparative study we evaluate the performance of Differential Evolution (DE) and Particle Swarm Optimization (PSO) algorithms in design optimization of PMSG for minimizing volume and at the same time maintain the generated power at the rated level.

Keywords : Permanent magnet synchronous generators (PMSG), Particle Swarm Optimization (PSO), Dynamic Particle Swarm Optimization (DPSO), Differential Evolution (DE) Algorithm

1. INTRODUCTION

Due to increasing demand for clean, emission-free energy and also due to the foreseeable depletion of fossil fuel sources in the future, wind energy among other renewable energy sources has been receiving considerable attention in recently years as an important low cost alternative to traditional fossil fuel energy sources. In past several decades, there has been increased effort on wind energy research to make it cost effective and efficient.

With the considerable growth of installed wind power capacity through the world and the consequent development of wind power technologies, various types of wind turbine generators have been developed. These various types of wind generators have been developed with the objective of maximizing the energy capture with minimum costs involved. Permanent magnet synchronous generators (PMSG) are most widely used for generation of wind power since they have high efficiency, high power factor and increased power density. In permanent magnet synchronous generators instead of field winding excited by dc supply, the excitation is provided by permanent magnets. Permanent magnet generators require less maintenance when compared with induction generators and synchronous generators with electrical excitation.

For wind power applications, permanent magnet synchronous generators with multiple poles have become very attractive especially in small ratings. Among the different wind generator topologies, Permanent magnet synchronous generators (PMSG) are one of the best solutions for small-scale wind power plants. Low-speed multi-pole PM generators are maintenance-free and may be used in different climate conditions. Potentially, permanent magnet generators offer a high efficiency in operational and are simple and robust in construction. The availability of modern high energy density magnet materials such as NdFeB has made it possible to design special topologies. Most of the wind turbine generators currently installed in small scale levels are in directly driven system, variable speed, and partially rated power electronic converter. Choice of such system is to avoid failures in gearbox and leading to long downtimes. Gearless full variable speed PM generators connected to the end user via a full rated power electronic converter are considered in several new installations. In most of the tropical countries, due to low wind speed, synchronous generators with smaller or medium speed PM generator designs found to be important and given high consideration. Hence, in this paper permanent magnet synchronous generator has been designed to meet the desired output power for direct driven small scale wind turbine. Design care has also been taken to minimize the overall volume of generator to reduce the cost using dynamic PSO and DE algorithms separately\.

2. LITERATURE SURVEY

Permanent magnet synchronous generators (PMSG) are widely employed for generation of wind power since they have high efficiency, high power factor and increased power density [1-2]. In permanent magnet synchronous generators instead of field winding the excitation is provided by permanent magnets. Permanent magnet generators require less maintenance when compared with induction generators and synchronous generators with electrical excitation [3].

In PMSG, the excitation is provided by permanent magnets instead of field winding. Permanent magnet machines are characterized as having large air gaps, which reduce flux linkage even in machines with multi magnetic poles [4]-[5]. As the result of previous study [6]-[7], low rotational speed generators can be manufactured with relatively small sizes with respect to its power rating. Moreover, gearbox can be omitted due to low rotational speed in PMSG wind generation system, resulting in low cost. In a recent survey, gearbox is found to be the most critical component, since its downtime per failure is high in comparison to other components in the wind turbine system.

[8] and [9] have reported the design aspect of PMSG to achieve better performance of PMSG at the cost of minimum permanent magnet material. This paper deals with design optimization of PMSGs used in small wind turbines. In fact, goal oriented design optimization of PMSGs is not well covered in literature. Therefore, in this paper a required generated power oriented auto design for PMSG parameters has been presented using the Dynamic PSO and DE algorithms separately. The proposed method also focuses on reducing the cost of generator by minimizing its volume.

DE is a simple evolutionary algorithm introduced by Storn and Price [10]. The main objective of the DE algorithm is to generate a new position for an individual, based on the calculation of the vector differences, between the members of the population [12-14]. PSO is a population based iterative optimization algorithm inspired by the collective behavior of the bird flocks and fish schools, firstly developed by Eberhart and Kennedy [11-14].

3. PMSG DESIGN MODELING

The generator output in terms of fundamental parameters is given by

$$P_{out} = \frac{D^2 L\, n_s \sigma_p}{\varepsilon} \tag{1}$$

Where

$$\sigma_p = \frac{\Pi^2 k_w A_m B_{mg1} cos\phi}{2}$$

$$B_{mg1} = \frac{4}{\Pi} B_{mg} sin\left(\frac{\Pi}{2}\alpha_i\right)$$

Hence final expression for P_{out} can be written as follows

$$P_{out} = \frac{D^2 L\, n_s\, \Pi^2 k_w A_m \frac{4}{\Pi} B_{mg} sin\left(\frac{\Pi}{2}\alpha_i\right) cos\phi}{2\,\varepsilon} \tag{2}$$

$$P_{out} = \frac{2\, D^2\, L\;\, n_s\, \Pi\; k_w\; A_m\; B_{mg}\; sin\left(\frac{\Pi}{2}\alpha_i\right) cos\phi}{\varepsilon} \tag{3}$$

Where:

Pout is the generator power output. ε is the no-load to rated load terminal phase voltage ratio. σp is the output coefficient, cosϕ is the load power factor, D is the diameter of the airgap (inner diameter of the stator), L is the stack length of the stator, ns is speed of rotation in rev/sec , kw1 is the fundamental winding factor , Bmg1 is the maximum value of the first harmonic component of the magnetic flux density in the air gap, Bmg is the peak value of the air gap magnetic flux density and αi is the ratio of the pole-shoe arc to pole pitch and Am is the maximum value of the linear current density of the stator respectively.

The total volume of generator can be represented as function of inner diameter of stator (D), axial length of generator (L), slot depth(hs) and stator back-iron(hbis) as given in Eq3.

$$Vol_{tot} = \Pi L \left[\frac{D}{2} + 2(h_s + h_{bis})\right]^2 \tag{4}$$

The definition of objectives can be stated as: for a desired value of Generator power, find the design parameters D, L,Am, Bmg and α_i so that generated power could meet the objective of desired one and at the same time care has been taken to reduce the cost by minimizing the volume of generator as much as possible.

4. OBJECTIVE FUNCTION

To achieve the objectives, problem is modeled as a problem of minimization with constraint, where obtained optimal values of five parameters D,L,Am, Bmg and α_i could deliver the Pout at rated level as it is considered as constraint and reduction of volume by minimizing the Eq.4 with removal of h_s and h_{bis}.

$$Vol = min\left\{\Pi\; L \left[\frac{D}{2}\right]^2 + 1000 * abs(P_{out} - P_{rated}\,)\right. \tag{5}$$

Equation (5) is the objective function of this study. The aim of optimization is to minimize it.

Table.1. Minimum and Maximum ranges of design variables

SI No.	Variable	Minimum	Maximum
1.	Stator inner diameter(D)(m)	0	1
2.	Stator stack length(L)(m)	0	1
3.	Maximum value of the magnetic flux density in air gap(Bmg)(T)	0	1
4.	Maximum value of the stator linear current density(Am)(A/m)	0	100000

A penalty concept has been applied if there is violation in the constraint by increasing the objective function value by large number in terms of 1000x$|P_{out} - P_{rated}|$, where P_{out} is the generated output with obtained values of parameters. The population size is selected as 200. The considered wind turbine generator has power rating Pout=3500w, Rated shaft speed [rpm] =250, Kw/ε= 0.7432 and power factor is taken as 1.Wind speed is assumed to be around 10m/s.

DESIGN OPTIMIZATION

5.1. Particle Swarm Optimization (PSO) Algorithm

In this section, firstly, Particle swarm optimization (PSO) algorithm is discussed. In addition, Differential Evolution Algorithm (DE) is briefly explained. Then, optimization process and selected case study are surveyed. Then optimization results are presented and discussed by these two algorithms.
To minimize the volume of PMSG and at the same time maintain the generated power output at rated level, a novel multipopulation based strategy of PSO is applied. PSO and Evolutionary Computation techniques are similar in that, a population of individual solutions to the problem considered is used to search the promising regions of the solution space or search space. On the other hand in PSO, every individual of the entire population has an adaptable velocity (change in position), and it travels in the solution space accordingly. In addition, each individual of the population has a memory, which helps to remember the best position of the solution space it has visited so far. Therefore the movement of each individual member of the population is gravitated towards its best position visited previously and towards the best individual of a topologically interacting neighborhood companions. PSO algorithm was developed with two variants. The first variant with a global neighborhood and second variant was with a local neighborhood. As per the global variant type, each individual travels towards its best previously visited position and towards the best performing individual in the entire swarm. In contrast, as per the local variant of PSO, each individual in the swarm moves towards its best previously visited position and towards the best performing individual in its restricted neighborhood. The global variant of PSO is exposed in the following paragraphs. In global variant PSO technique, by assuming that the current searching space is having D dimensions, then the position of the i-th particle of the swarm is denoted by a vector, Xi =[xi1, xi2,…… xiD] having D dimensions. The flying velocity (change in position) of this particle can be denoted by another vector Vi = [vi1, vi2, . . . viD] having D dimensions. The previous best position of the i-th particle in the swarm is represented as Pi =[pi1, pi2, . . . piD]. Let 'g' denote the index of the best performing particle in the entire swarm (i.e., best is the g-th particle),'n' is the best seen by that particular particle and let the superscripts represent the iteration number, then in the global variant of PSO, the swarm is manipulated using the following two equations

$$V\,id\,(n+1) = \chi\,[w\,Vnid + C1\,r1\,(Pidn - Xnid) + C\,2\,r2\,(Pngd - Xnid)] \quad (6)$$

$$X\,(n+1)\,id = X\,nid + V\,(n+1)\,id \;\ldots\ldots.. (7)$$

Where w is the inertia weight; c1is called cognitive constant and c2 social constant respectively; and constriction factor is denoted by χ. In the next section, the role of these parameters on the performance of PSO is discussed. Each particle in the swarm travels towards the best performing particle of its neighborhood in the local variant of PSO. Space calculations are done by the swarm in PSO concept over series of time steps. The population responds to the quality factors implied by each particle's personal best position and the global best, thereby allocates the responses in such a way that ensures diversity. In case of global variant of PSO, the swarm changes its behavior or state only when the best performing particle in the entire swarm changes. In case of local variant of PSO, the swarm changes its behavior or state when the best performing particle in the neighborhood changes. Therefore, it is stable as well as adaptive.

In Equation (6), the inertia weight w plays a most important role in the convergence characteristics of PSO. The impact of the past history of velocities of particles on the current velocity of particles is controlled by inertia weight w. Therefore, the inertia weight w balances between global search and local search abilities of the entire swarm. A large value of inertia weight promotes global exploration while a small value of inertia weight tends to promote local exploration, i.e., fine-tuning of the current search area. An appropriate value of inertia weight w usually provides a better balance between global exploration and local exploration abilities of the swarm and therefore results in a decrease in the number of iterations needed to locate the optimal solution in the search space. The inertia weight w was set to a constant value initially. But, results from experiments indicated that initially it is better to set the inertia weight w to a large value, to promote global exploration of the problem space, and steadily decrease the value of inertia weight w in order to get solutions which are more refined. Therefore, initially setting the inertia weight to an approximate value of 1.2 and a steady decline towards 0 can be treated as an appropriate choice for inertia weight w. In Equation (6), the constants c1 and c2 are not significant for convergence of PSO. In order to preserve the diversity of the population in PSO parameters r1 and r2 are used and they are distributed uniformly in the range [0, 1]. The constriction factor χ plays a major role in controlling the magnitude of the velocities of flying particles, in a manner very much similar to the Vmax parameter which improves the convergence speed of the particles.

5.2. Dynamic Particle Swarm Optimization (DPSO) Algorithm

Dynamic particle swarm optimization (DYPSO) was originally proposed by Shi and Eberhart in which they introduced a linearly decreasing inertia weight factor into the velocity of the updated equation from the original PSO. In the Eq.6, weight factor *w* plays the central role in the convergence characteristics of PSO. High value of *w* makes PSO under the exploration stage. Low

value will make the move towards the exploitation. It is very logical that at the beginning of iteration there is need of high level of exploration and as the iterations are increasing, level of exploration has to reduce. Mathematically in this work, this has been achieved by providing a reduced value of inertia weight w as a function of iterations as given by Eq.8.
The DYPSO balances out the global and local search abilities of the swarm effectively and therefore an improvement in the performance cab be expected from LDIWPSO compared to the original version of PSO. In DYPSO, the inertia weight 'w' is linearly decreased from 1.2 to 0.1 through the search process with iterations.

$$w = w_{max} - \frac{(w_{max} - w_{min})}{iter_{max}} \times iter \qquad (8)$$

(4.2)

Where w_{max}: initial weight. wmin: final weight. itermax : maximum iteration number. iter: current iteration number.

5.3. Differential Evolution (DE) Algorithm

In the evolutionary computation algorithms, fundamentally GA evolutionary programming and evolutionary strategy play the central role. Functionally there are two approaches to generate the offspring. Through single parent or through bi-parental system. As like the GA, there are cross-over and mutation operations for creation of new population members. DE is also carrying both the operators but it is a single parental system. The offspring is created through differences existing between two or more population members. This difference can provide the facility to exploit the solution more efficiently. There is all point cross over, which makes the solution move towards the maximum level of exploration and exploitation. The high level of exploration and exploitation makes the DE more valuable in terms of finding the global solution. There is less chance to converge the solution sub-optimally.

The population of the DE algorithm contains NP individuals and each has D-dimensional vector as according to D dimensions available in the problem. During one generation for each vector, DE employs mutations to produce a donor vector of dimension D. Mutation implies the addition of scaled, randomly sampled, vector difference taken from the previous population randomly shuffled to a third vector of the previous population There are various strategies exist to define the donor vector. In this research, a strategy called DE/rand/1 as defined in Eq.9 has been taken. The crossover operator under probabilistic environment has been applied to develop the trial vector as shown in Eq.10. CR is a crossover control parameter or factor within the range [0, 1] and represents the probability of creating parameters for a trial vector from the mutant vector. Index jrand is a randomly chosen integer within the range [1, NP]. Then a greedy selection operation selects between the target and corresponding trial vectors to choose vectors for the next generation as according to Eq.11 where F represents a type of mutation factor in terms of comparing the fitness value through fitness function f. The mutation and crossover factor have taken as 0.4 and 0.5.The population size has considered as 200.

$$V_i^{(G)} = X_{r1}^{(G)} + F * (X_{r2}^{(G)} - X_{r3}^{(G)}) \qquad (9)$$

$$\boldsymbol{u_{ij}^{(G)}} = \begin{cases} \boldsymbol{v_{ij}^{(G)}} & \boldsymbol{if\ rand\ (0,1) \le CR\ or\ j = j_{rand}} \\ \boldsymbol{x_{ij}^{(G)}} & \boldsymbol{otherwise} \end{cases} \qquad (10)$$

$$\boldsymbol{x_{ij}^{(G)}} = \begin{cases} \boldsymbol{u_i^{(G)}} & \boldsymbol{if\ f(u_i^{(G)}) \le f(x_i^{(G)})} \\ \boldsymbol{x_i^{(G)}} & \boldsymbol{otherwise} \end{cases} \qquad (11)$$

Optimization Results

The algorithms DE and DYPSO were independently run for 5 trials to obtain the optimal design parameters and minimum value of generator volume along with satisfying the required rated power.The performance to minimize the volume for given power rating using the DE and DYPSO algorithms under 5 different trials have been shown in Table1 and Table2. During each trial, the required computation time is also recorded independently so that comparative benefit can be measured. The convergence characteristics have shown for 200 iteration number and shown in Fig.1 and in Fig.2. It can observe that there is not only minimal volume has achived by the DYPSO but also computational cost is also very less.

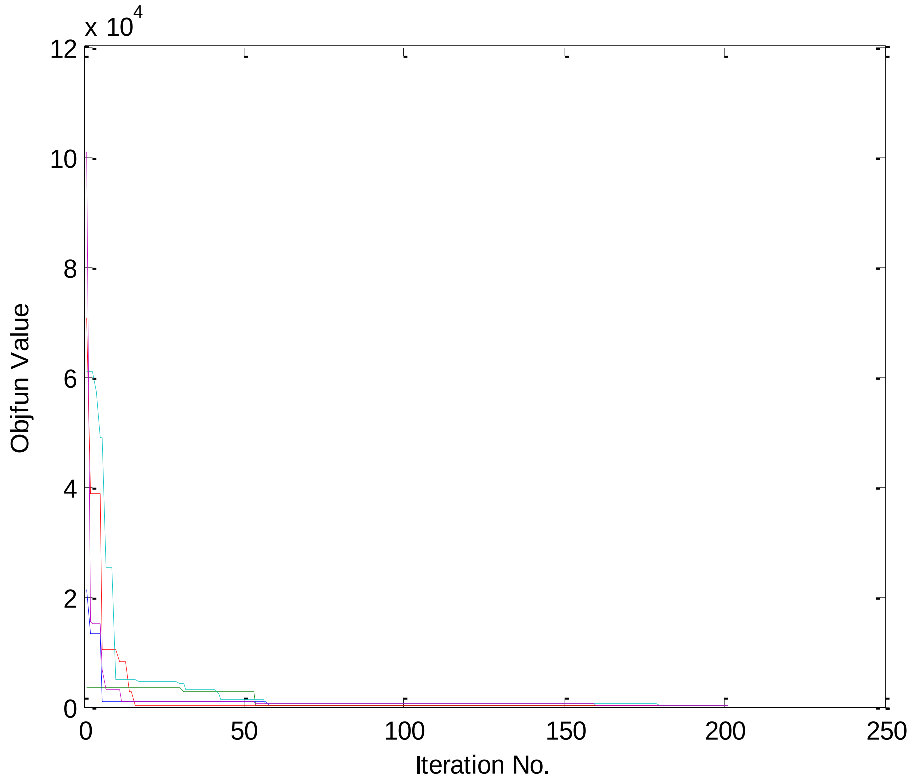

Fig.1. Objective function value versus iterations by DE in 5 trials

Table2. Performance delivered by DE

DE	D (mm)	L(mm)	A_m (A/m)	B_{mg} (T)	α_i
1	7.064084579396871e-001	9.875530244941426e-001	5.375115777340901e-002	2.633552088232887e-001	1.660334139114483e-001
2	1.504794401579739e-001	2.358710440371731e-001	5.894234524870124e-001	6.048419375184960e-001	7.874155497132513e-001
3	4.213285528098437e-001	6.110890010994354e-001	1.173777373036334e-001	1.478608841624796e-001	8.091067025963716e-001
4	4.464607335938265e-001	4.244305401698904e-001	3.768970212019203e-001	5.004914248116970e-001	7.190235090150277e-002
5	3.472242000908485e-001	7.407199512760287e-001	3.233118986731451e-001	6.230057188045853e-002	9.931176083090473e-001

Table3. Corresponding performance by DE

DE	Pout	D= abs (Pout –Prated	Volume	Vol+1000*D	Comput Cost
1	3.499911366074078e+003	8.863392592184027e-002	3.870455563726744e-001	8.902097147821294e+001	80.2
2	3.500219503507081e+003	2.195035070808444e-001	4.194873263322744e-003	2.195077019541077e+002	79.6
3	3.499737887581416e+003	2.621124185839108e-001	8.519932059499027e-002	2.621976179045058e+002	81.3
4	3.499500636331573e+003	4.993636684271223e-001	6.644511305548538e-002	4.994301135401778e+002	80.7
5	3.499747275975812e+003	2.527240241879554e-001	7.013969871941095e-002	2.527941638866748e+002	82.3
mean	**3.499823333893992e+003**	**2.644675088403347e-001**	**1.226049124011767e-001**	**2.645901137527358e+002**	**80.82**
std	**2.654805186391363e-001**	**1.485419738716137e-001**	**1.510426194491817e-001**	**1.484517106186258e+002**	

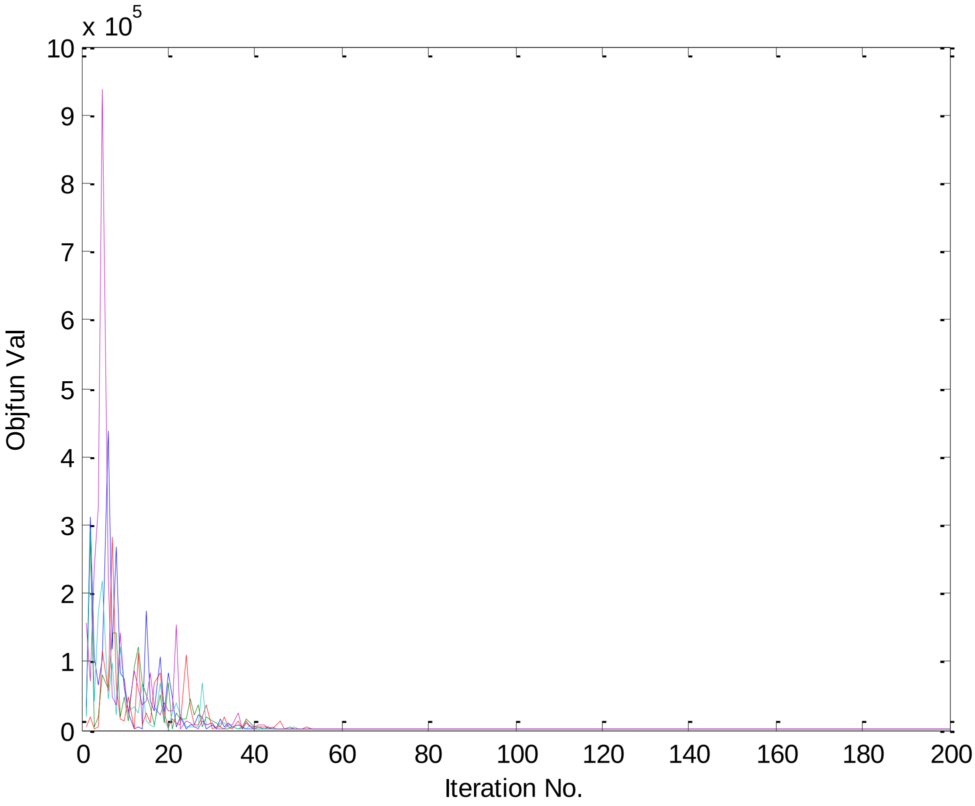

Fig.2. Objective function value versus iterations by DYPSO in 5 trials

Table4. Performance delivered by DYPSO

	D (mm)	L(mm)	A_m (A/m)	B_{mg} (T)	α_i
1	1.345285411511799e-001	6.617521583961445e-001	9.897717734698972e-001	2.289507417855571e-001	4.612768895971676e-001
2	2.412334279646222e-001	1.211213019925436e-001	6.445114429817274e-001	7.443981155123712e-001	3.570733204729930e-001
3	1.699038797968262e-001	6.043473043752980e-001	2.287550751820752e-001	8.612154783733141e-001	3.506590911197546e-001
4	2.582610240103065e-001	8.768683244311251e-002	5.607893334214712e-001	7.651528416829061e-001	5.087813561966702e-001
5	1.928823544947054e-001	2.891304129307820e-001	5.967016893299151e-001	3.206436143729218e-001	6.770273101119589e-001

Table5. Corresponding Performance by DYPSO

DE	Pout	D= abs (Pout –Prated	Volume	Vol+1000*D	Comput Cost
1	3.499999999999991e+003	9.549694368615747e-012	9.406197930552905e-003	9.406207480247274e-003	0.92
2	3.500000000000006e+003	6.366462912410498e-012	5.535863697936979e-003	5.535870064399892e-003	0.87
3	3.500000000000001e+003	4.547473508864641e-013	1.370197160233783e-002	1.370197205708518e-002	0.86
4	3.499999999999991e+003	9.549694368615747e-012	4.593481809112219e-003	4.593491358806588e-003	0.87
5	3.499999999999971e+003	2.910383045673370e-011	8.448286929616761e-003	8.448316033447218e-003	0.94
mean	**3.499999999999991e+003**	**1.100488589145243e-011**	**8.337160393911339e-003**	**8.337171398797231e-003**	**0.8920**
std	**1.343622598387279e-011**	**1.077760815384188e-011**	**3.598568541410093e-003**	**3.598565922153609e-003**	

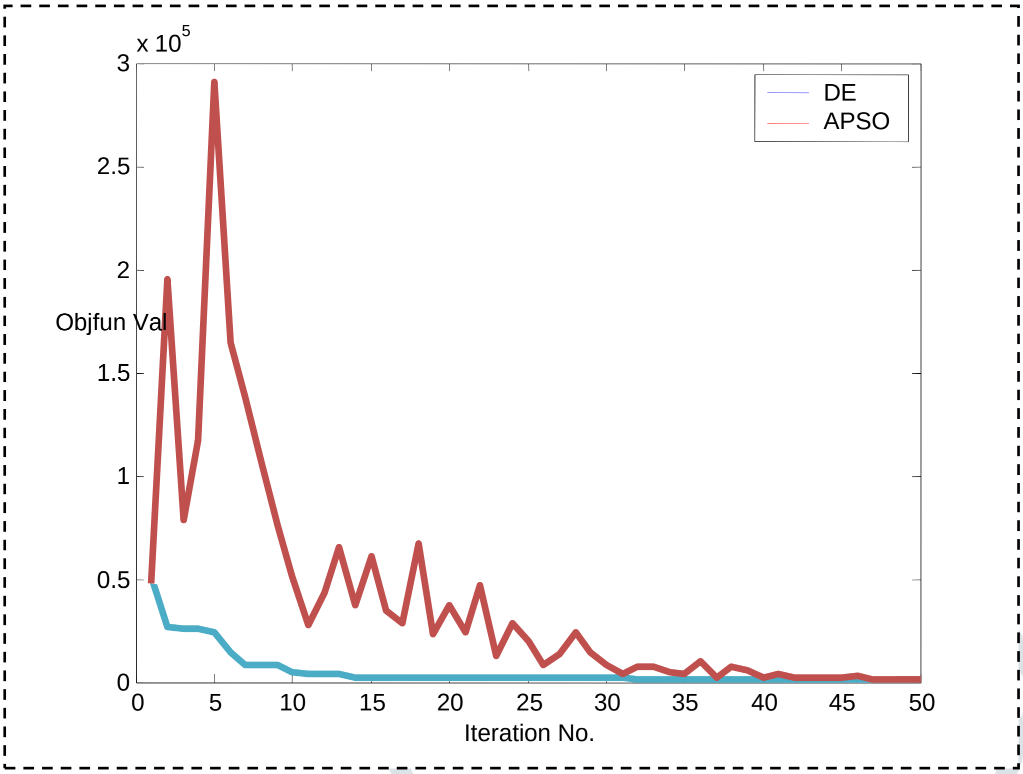

Fig.3. Mean convergence in first 50 iteration between DE and APSO

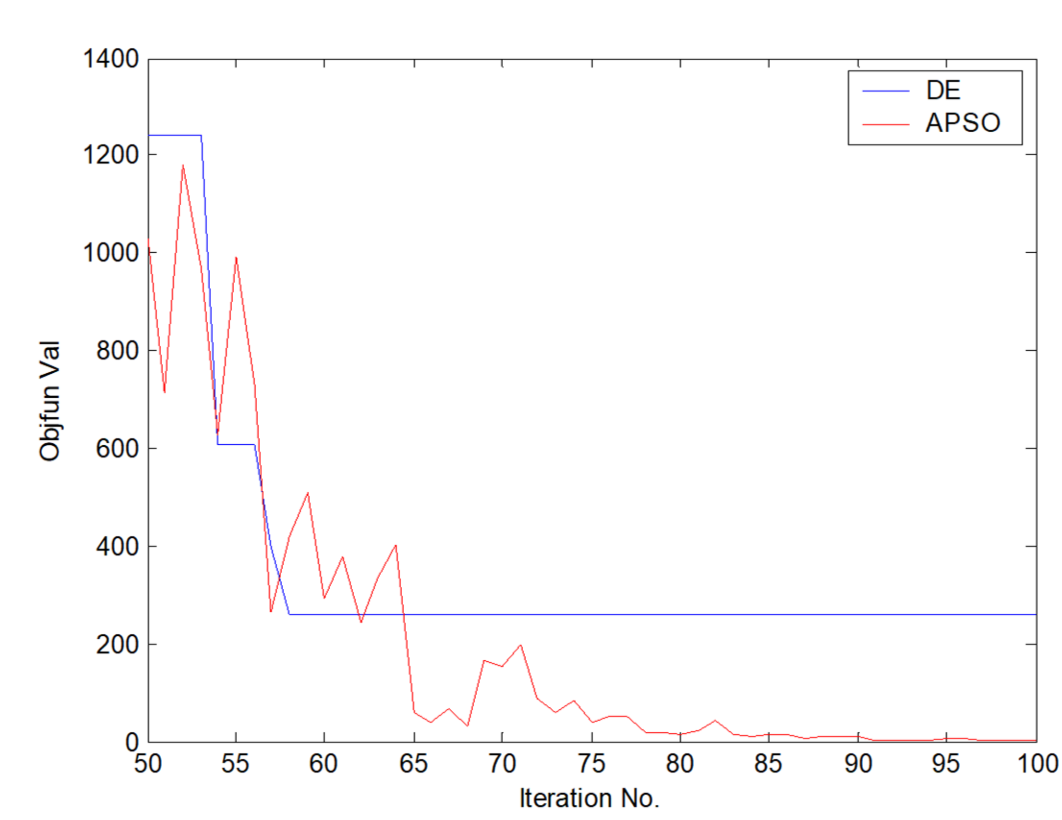

Fig.4. Mean convergence in 50-100th iteration between DE and APSO

The comparisons between mean convergence in DE and DYPSO have shown in two different sections to get the clarity in progress. In Fig.3. for the first 50 iterations and in Fig.4. for 50-100th iteration performances have shown .It can observe that there was better convergence by DE at the beginning but later it has converge sub optimally. DE algorithm shows excellent performance during the initial iterations but shows stagnation when the iteration increases. Hence, DE algorithm settles in local optima and gives inaccurate results.

It can be observed from the simulation results in Table1. and Table2. that the objective function in case of PSO algorithm is minimized to a larger extent as compared to the DE algorithm. This result indicates that PSO outperforms DE in terms of minimizing the volume of PMSG and at the same time maintaining the generated power at the rated level. It can be noticed from table that the execution time for the PSO algorithm is less compared with the execution time of the DE algorithm for different trials.

CONCLUSION

In this paper, we present a comprehensive performance comparison of the DE and DYPSO algorithms in minimizing the volume of PMSG and at the same time maintain the generated power at the rated level. From the simulation results it is clear that DYPSO is better than DE in terms of solution quality, running time and chance of reaching the best solutions for different number of trials. The better performance of PSO in terms of complexity of computation and solution convergence may be due to its simple structure and minimal parameters. It is also interesting to note that DYPSO is computationally efficient than DE.

Reference

[1] Y. Chen, P. Pillay, and A. Khan, “ PM wind generator topologies”, IEEE Trans. Ind. Appl.41, 1619 (2005).

[2] L. Soderlund and J.-T. Eriksson, “A permanent-magnet generator for wind power applications”, IEEE Trans. Magn. 32, 2389 (1996).

[3] J. Chen, C. V. Nayar, and L. Xu, “Design and finite-element analysis of an outer-rotor permanent-magnet generator for directly coupled wind turbines”, IEEE Trans. Magn. 36, 3802 (2000).

[4] Peter Vas, Electrical machines and Drives- A Space Vector Theory Approach, New York: Oxford University Press, 1992.

[5] T. J. E. Miller, Brushless Permanent-Magnet and Reluctance Motor Drives, New York: Oxford University Press, 1989.

[6] J. Ribrant and L. M. Bertling, Survey of Failures in Wind Power Systems With Focus on Swedish Wind Power Plants During 1997- 2005.

[7] G. Böhmeke, “Development and operational experience of the wind energy converter WWD-1,” in Proc. Europ. Wind Energy Conf., 2003.

[8] Sahu A., Gupta S., Singh V.K., Bhoi A.K., Garg A., Sherpa K.S. (2018) “Design of Permanent Magnet Synchronous Generator for Wind Energy Conversion System”, in: SenGupta S., Zobaa A., Sherpa K., Bhoi A. (eds) Advances in Smart Grid and Renewable Energy. Lecture Notes in Electrical Engineering, vol 435. Springer, Singapore

[9]Fang, H. & Wang, D., “Design of permanent magnet synchronous generators for wave power generation”, Transactions of Tianjin University, October 2016, Volume 22, Issue 5, pp 396–402

[10] R. Storn, and K. Price, “Differential evolution: A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces,” Technical report, TR-95-012, International Computer Science Institute, Germany, Mar. 1995

[11] R. Eberhart, and J. Kennedy, "Particle swarm optimization," In Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948, Perth, WA, Dec. 1995.

[12] M. Zhang, W. Zhangi, and Y. Sun, "Chaotic Co-evolutionary Algorithm Based on Differential Evolution and Particle Swarm Optimization," IEEE International Conference on Automation and Logistics, pp. 885-889, Shenyang, Aug. 2009.

[13] I. Ahmed, S. Sadeque, and S. Pervin, "Margin Adaptive Resource Allocation for Multiuser OFDM Systems by Modified Particle Swarm Optimization and Differential Evolution," IEEE 21st International Conference on Electrical Communications and Computers, pp. 227-231, San Andres Cholula, Mar. 2011.

[14] J. Zhang, S. Chen, X. Mu, and L. Hanzo,"Evolutionary-Algorithm-Assisted Joint Channel Estimation and Turbo Multiuser Detection/Decoding for OFDM/SDMA," IEEE Transactions on Vehicular Technology, vol. 63, no. 3, pp. 1204-1222, Mar. 2014.