A LITERATURE SURVEY ON PLACEMENT AND SIZING OF DGs AND SHUNT CAPACITORSIN DISTRIBUTION FOR POWER SYSTEM

Shishir Dixit
Department of Electrical Engineering
Madhav Institute of Technology and Science,
Gwalior, India.

Abstract : - In the present work, a hybrid approach has been proposed for optimal placement of multiple DGs of multiple types. The analytical approaches may not be appropriate for optimal placements of multiple DGs alone. In this work, hybridization of analytical method and search for the optimal placement of multiple DGs in power distribution network for reduction of power loss has been proposed. In this approach, the sizes of DGs are evaluated at each bus by analytical method while the locations are determined by PSO based technique. The objective function has been minimized under operating constraints. The improvements in bus voltage profile and optimal power factor of the DGs have also been observed. To validate the proposed hybrid approach, results have been compared with particle swarm optimization (PSO) technique.

Keywords- PID Controllers; Tuning; Classical Techniques; Intelligent Computational Technique.

I. INTRODUCTION

The increasing number of consumers and how to supply the loads are the most important challenges in the power system. Since the cost of construction or upgrading transmission lines and distribution networks is very high, the proper utilization of the low cost DGs has been a solution to eliminate or to delay such investments [1].Moreover, among different sections of the power system, distribution network has the largest portion of the power loss because of its low level voltage with having a high current [1]. In this regard, it has been shown that one of the most cost-effective and an economical solution to solve this problem is to use DG resources [2].In this regard, the optimal operation and planning of distribution networks, considering power system uncertainties, especially in the modern smart distribution networks are also very important. Refs. [3–6] highlight the importance of energy storage in combination with distributed generation for these purposes. More information about the application of DGs in implementing smart distribution network functions such as self-healing ability can be found in [7]. Several papers in which different optimization techniques such as genetic algorithms, continuous power flow, ant colony, particle swarm optimization have been used [8–11]. Analytical methods for finding the optimal size of different types of DGs are also suggested in [12]. In [13] an analytical method and in [14–17] numerical techniques are applied to find the optimal locations and sizes of multiple DGs. A fuzzy GA is employed to solve a weighted multi objective optimal DG placement model [18–19].

Furthermore, it is very common to use reactive power sources such as parallel capacitors to improve the voltage profile as well as reducing the power losses in the lines. Refs. [20–23] determine the locations and sizes of shunt capacitors with different goals and algorithms.

Considering the advantages of using both DGs and capacitors in distribution networks many researchers have recently proposed different techniques to simultaneously determine the locations and sizes of both to improve the voltage stabilization, system capacity release, energy loss minimization and reliability enhancement. Ref. [24] uses the PSO algorithm to find the optimal location and size of shunt capacitor and DG in 12, 30, 33 and 69-bus IEEE standard networks in order to minimize losses. The IEEE 33-busnetwork is employed as a test system in [25] to show the advantage of using an improved genetic algorithm for locating DG and capacitor. The same purpose was followed in [26] by using BFOA in the 33-bus network, and results are compared for three different cases of when; 1- only DG, 2- both DG and capacitor, and 3- none of them, are used in the test system. In [27] the problem of locating both DG and capacitor is solved by using BPSO algorithm, wherein addition to the main objectives of loss reduction and voltage improvement, the network reliability indices are used. In [28] both artificial bee colony and artificial immune system algorithms are combined to locate and to determine the size of capacitors and DGs in distribution networks. The proposed method was tested on IEEE 33-bus test system for several cases. The simulation results show that the proposed approach provides better power loss reduction and voltage profile enhancement when compared with different methods. DGs and capacitors are optimally located and sized in [29] by using DPSO algorithm. Ref. [30] employs TLBO technique to maximize the ratio of the profit to cost when both capacitor and DG are used. Other algorithms such as BGSA [31], simple genetic algorithm [32, 33], genetic and ICA techniques [34] have been also used in this field.

This paper presents a new algorithm, named IMDE [35] to optimally locate DGs and shunt capacitors as well as determining their sizes in radial distribution networks. This algorithm not only has a higher convergence speed, but also gives a better performance compared to earlier works in this field. The results clearly show the highest level of loss reduction as well as keeping the voltages of buses within their limits.

A. Classical Techniques

Classical techniques make certain assumptions about the plant and the desired output and try to obtain analytically, or graphically some feature of the process that is then used to decide the controller settings. These techniques are computationally very fast and simple to implement, and are good as a first iteration. But due to the assumptions made, the controller settings usually do not give the desired results directly and further tuning is required. A few classical techniques have been reviewed in this paper.

B. Computational or Optimization Techniques

These are techniques which are usually used for data modeling and optimization of a cost function, and have been used in PID tuning. Few examples are neural networks (computational models to simulate complex systems), genetic algorithm and differential evolution. The optimization techniques require a cost function they try to minimize. There are four types of cost functions used commonly. Computational models are used for self tuning or auto tuning of PID controllers. Self tuning of PID controllers essentially sets the PID parameters and also models the process by using some computational model and compares the outputs to see if there are any process variations, in which case the PID parameters are reset to give the desired response.

The existent types of adaptive techniques are classified based on the fact that if the process dynamics are varying [5], then the controller should compensate these variations by adapting its parameters. There are two types of process dynamics variations, predictable and unpredictable. The predictable ones are typically caused by nonlinearities and can be handled using a gain schedule, which means that the controller parameters are found for different operating conditions with an auto-tuning procedure that is employed thereafter to build a schedule. Different techniques have been used to replace the gain schedule mentioned above. In the discussion of various techniques its usage in self tuning is also mentioned.

II. CLASSICAL TECHNIQUES

A. Ziegler Nichols Method

This is by far the most popular tuning method in use. It was proposed by John Ziegler and Nathaniel Nichols [6] in 1942 and is still a simple, fairly effective PID tuning method. There are two methods proposed by Ziegler and Nichols. This method was used to tune PID controllers for spindle motor systems [8]. The second method is based on knowledge of the response to specific frequencies. The idea is that the controller settings can be based on the most critical frequency points for stability. This method is based on experimentally determining the point of marginal stability. This frequency can be found by increasing the proportional gain of the controller, until the process becomes marginally stable. The Ziegler and Nichols method is the first PID tuning techniques made and they are made based on certain controller assumptions. Hence, there is always a requirement of further tuning; because the controller settings derived are rather aggressive and thus result in excessive overshoot and oscillatory response. Also for the first method the parameters are rather difficult to estimate in noisy environment. In the second method, as the system is driven towards instability for determining the parameters, practically this can be quite detrimental to the system.

III. COMPUTATIONAL AND INTELLIGENT OPTIMIZATIONTECHNIQUES

The various intelligent optimization techniques are discussed below.

A. Immune Algorithm

Artificial Immune Systems (AIS) are computational systems inspired by the principles and processes of the vertebrate immune system, which learns about the foreign substances to defend the body against them. The immune system has two types of responses, primary and secondary. The former is the response when it first encounters the antigen. In this period, the system learns about the antigen, creating a memory of it. The later occurs when the antigen is encountered for the second time, which is a more rapid and larger response. The cells primarily involved in this system are B cells. Against the antigen, the level to which a B cell is stimulated relates partly to how well its antibody binds the antigen The immune system has two types of responses, primary and secondary. The former is the response when it first encounters the antigen. In this period, the system learns about the antigen, creating a memory of it. The later occurs when the antigen is encountered for the second time, which is a more rapid and larger response. The cells primarily involved in this system are B cells. Against the antigen, the level to which a B cell is stimulated relates partly to how well its antibody binds the antigen.

B. Ant colony Optimization

Ant Colony Optimization (ACO) [13-14] is a recently developed meta-heuristic approach to solving optimization problems based on working of an ant colony. More precisely, it is based on the ant colony finding the shortest path to the food. Each ant tries to find the food through some random path leaving behind a trail of pheromones. The pheromone trail weakens with reducing no. of ants passing through that path and strengthens with increasing no. of ants passing through it. So basically it is a search algorithm which depends on a number of ants acting together moving towards the optimal solution.

Ant colony optimization was used for PID tuning in [15]. It was used to minimize a multi-objective function and its results were found to be better than genetic algorithm and Ziegler Nichols method. In [16] authors have demonstrated the use of bees algorithm to tune a PID controller and solving complex systems. The results of ACO, PSO and bees algorithm are compared and presented in [17].

C. Bacteria Forage Technique

Since the selection behavior of bacteria tends to eliminate entities with poor foraging strategies and favor the propagation of genes of those that have successful foraging strategies, they are applied to find an optimal solution through methods for locating, capturing, and ingesting food [18]. Foraging theory is discussed in [19]. All papers on PID tuning with bacteria foraging technique [18], [20], [21] study the foraging behavior of E. Coli, a common bacteria [22]-[23]. The behavior of E.Coli is described in [18] as,

- If in a neutral medium, it alternates between tumbles and runs and searches the environment.
- If swimming up a nutrient gradient (or out of noxious substances) or swimming longer (up a nutrient gradient or down a noxious gradient) it seeks an increasingly favorable environment.
- If swimming down a nutrient gradient (or up a noxious substance gradient), it searches to avoid unfavorable environments.

D. Coli occasionally engage in a conjugation that affects the characteristics of a population of bacteria. There are attractants that bacteria like, attraction to oxygen (aero taxis),light (photo taxis), temperature (thermo taxis), and magneto taxis (it is affected by magnetic lines of flux). Some bacteria change their shape and number of flagella based on the medium to reconfigure and ensure efficient foraging in a variety of media. The main goal based on bacterial forgingis to find the best position of the bacteria with respect to the attractant and repellent profile. A hybrid approach consisting of genetic algorithm and bacteria forage fortuning of PID controller for AVR system is proposed in [21].

E. Genetic Algorithm

Genetic algorithm (GA) is a search algorithm that explores the search space in a manner analogous to evolution in nature [24]. It uses probabilistic rules to search for and change the potential solutions in the search space, using a cost function to analyze the fitness of solutions. GA requires the solution to be represented in a way that is analogous to genes so that the processes that bring about a change in the genes (like mutation) can be used. Usually this is done by representing the solutions in a binary format.

- Initialization, firstly initial solutions are randomly selected from the search space.
- Selection, during each iteration, a proportion of solutions is selected, based on the fitness function (fitter solutions are more likely to get selected), for breeding the next generation of solutions. The selection is done in a probabilistic manner.
- Reproduction, selected solutions are paired up and crossover and mutation operation are performed to get the next generation of solutions.
- Termination, the iterations are terminated when the termination condition (time or accuracy) is reached

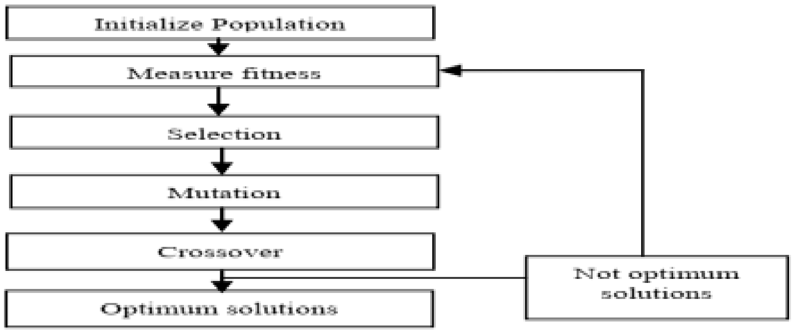

Fig. 2 Flowchart of genetic algorithm based tuning

GA is very popular in PID tuning, and has gained wide applications in control systems [25]. Girishrajet. al [25] used GA for improving performance of a PID controller used in bioreactor and compared the performance with Ziegler Nichols, Skogestad modification [26] and IMC rule [27] and found that GA outperformed both in terms of overshoot, disturbance rejection, gain margin and phase margin. The one of the limitations of GA is its tuning a multivariable. GA has been used in position and speed control of a DC motor.

Lot of work has been done in using GA along with other computational techniques. In [21], Kim et al. use bacteria forage along with GA for PID controller tuning of AVR systems. GA was used with NN in [34] and with fuzzy logic in [35] for developing self tuning methods

F. Differential Evolution

Differential Evolution (DE) is a method for doing numerical optimization without explicit knowledge of the gradient of the problem to be optimized. The DE method is originally due to Storn and Price and works on multidimensional real-valued functions which are not necessarily continuous or differentiable. DE optimizes a problem by maintaining a population of candidate solutions and creating new candidate solutions by combining existing ones according to its simple formulae of vector-crossover and -mutation, and then keeping whichever candidate solution has the best score or fitness on the optimization problem at hand. In this way the optimization problem is treated as a black box that merely provides a measure of quality given a candidate solution and the gradient is therefore not needed. Differential evolution is used for online PID tuning.

G. Evolutionary Programming

Generally, the EP algorithm for global optimization contains four parts, initialization, mutation, competition, and reproduction. Mutation is based on the current values and a Gaussian random variable. Furthermore, a quasi-random sequence (QRS) is used to generate an initial population for EP [16] to avoid causing clustering around an arbitrary local optimum [20]. Evolutionary programming was used in [21] for PID tuning using IAE and

H. PARTICLE SWARM OPTIMIZATION (PSO)

It is an algorithm, very simple and easy to implement. The algorithm keeps track of 3 global variables:

1. Target value or condition
2. Global best value indicating which that which
3. particle's data is currently closest to the target. Stopping value indicating, when the algorithm

4. should stop if the target is not found.
5. Each particle consists of:A velocity value indicating how much the data canbe changed.
6. A personal best (pbest) value indicating the closest

the particle's data has ever come to the target.

PSO learned from the scenario and it used to solve the optimization problem in PSO, each single solution is a "bird" in the search space. It is known as "particle". All particles have fitness values which are evaluated by the fitness function to be optimized. And have the velocities which direct the flying of the particles.

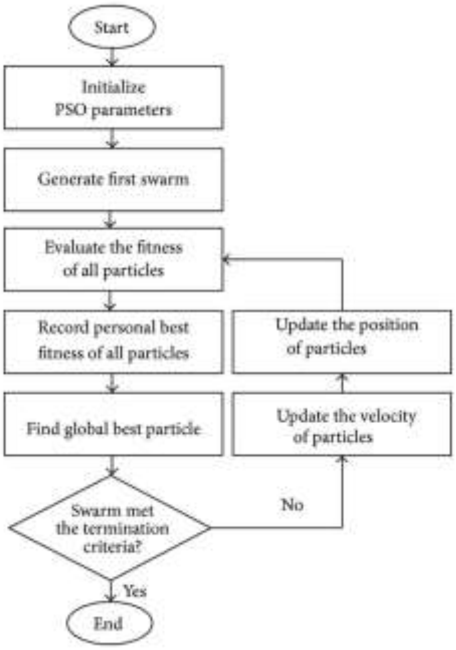

Fig. 3 Flowchart of PSO based tuning

The particles fly thorough the problem space by following the current optimum particles. PSO initialized with a group of random particles and then searches for optima by updating generations. In every iteration, each particle is updated by following two "best" values. The best solution it has achieved so far, this value is called pbest. Another best value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is known as "gbest".
When a particle takes part of the population as its topological neighbors, the best value is a local best and is Called p-best. After finding the two best values, the particle updates its velocity and positions with following equations.

- The parameter V_{max} determines the resolution, or fitness, with which regions are to be searched between the present position and the target position.
- If V_{max} is too high, particles may fly past good solutions. If V_{min} is too small, particles may not explore sufficiently beyond local solutions.
- In many experiences with PSO, Vmax was often set at 10-20% of the dynamic range on each dimension.
- The constants C 1 and C 2 pull each particle towards pbest and gbest positions.
- Low values allow particles to roam far from the target regions before being tugged back. On the other hand, high values result in abrupt movement towards, or past, target regions.
- The acceleration constants C 1 and C 2 are often set to be 2.0 according to past experiences.
- Suitable selection of inertia weight "ω" provides a balance between global and local explorations, thus requiring less iteration on average to find a sufficiently optimal solution.
- In general, the inertia weight w is set according to the following equation,

VI. CONCLUSION

All the algorithms discussed in the above section are optimization problems. These are compared with each other and every algorithm has its own advantages. Genetic Algorithm is much more popular because of its parallel computation. One more advantage of Genetic Algorithm is that it may handle both continuous and discrete variable without any gradient information where as all other techniques may give best performance for continuous problems but need slight.

We examined various optimization techniques by a comparative analysis based on evolution, methodology, performance and applications. In this paper, we find that these algorithms may be applied in various domains whether using as a direct approach or as any modified version. In future, new improved algorithms can be found for different scope areas.

REFERENCES

[1] S. Bennett, A History of Control Engineering 1930-1955, IEE Control Engineering Series 47, 1993.
[2] S. Bennett, "Nicholas Minorsky and the Automatic Steering of Ships," IEEE Control Systems Magazine, vol. 4, no .4, pp. 10–15, 1984.

[3] M. Araki, PID Control in Control systems, Robotics and Automation, vol II, edited by Heinz Unbehauen, Encyclopedia of Life Support Systems (EOLSS), Developed under the Auspices of the UNESCO, Eolss Publishers, Oxford ,UK.

[4] K. J. Astrom and T. Hagglund, PID controllers, Theory, design and tuning, 2nd edition, Instrument Society of America, 1995.

[5] T. Hagglund and K. J Astrom, "Industrial Adaptive Controllers Based on Frequency Response Techniques, "Automatica, vol. 27, no. 4, pp. 599-609, 1991.

[6] J. G. Ziegler and N. B Nichols, "Optimum Settings for Automatic Controller," Transaction of ASME, vol. 64, pp. 759-768, 1942.

[7] J. C. Cool, F. J. Schijff, T. J. Viersma, and Regeltechniek, Control Engineering, Delta Press, Amerongen, 1991.

[8] C. S. Soh, C. Bi and K. C. Chua, "Direct PID Tuning For Spindle Motor Systems," Asia-Pacific Magnetic Recording Conference, Singapore, pp. 1-2, Nov-Dec.2006.

[9] G. H. Cohen, and G. A. Coon, "Theoretical considerations of retarded control," Transactions of ASME, vol. 75, pp. 827-834, 1953.

[10] Dong Hwa Kim, Tuning of PID Controller Using Gain/Phase Margin and Immune Algorithm, IEEE Mid-Summer Workshop on Soft Computing in Industrial Applications, Helsinki University of Technology, Espoo, Finland, pp. 6-74, 2005.

[11] Dong Hwa Kim and Jae Hoon Cho, "Intelligent Tuning of PID Controller with Disturbance Function Using Immune Algorithm," International Conference on Computational Intelligence for Measurements and Applications, vol. 1, pp. 286-291, 2004.

[12] Shi Zhongzhi, Shi Zhongzhi, Shi Zhongzhi, "Auto-tuning of reference model based PID controller using immune algorithm," Proceedings of the World on Congress on Computational Intelligence, pp. 483-488, 2002.

[13] M. Dorigo and G. Dicaro, The Ant Colony Optimization Meta-heuristic, In New Ideas in Optimization, McGraw Hill, London, UK, pp. 11–32, 1999.

[14] M. Dorigo, G. Di Caro, and L. M. Gambardella, "Ant Algorithms for Discrete Optimization," Artificial Life, vol. 5, no. 2, pp. 137–172, 1999.

[15] IbtissemChiha, NoureddineLiouane and Pierre Borne, "Multi-Objective Ant Colony Optimization to tuning PID controller," International Journal of Engineering, vol. 3, no. 2, pp. 11-16, 2009.

[16] D. T. Pham, A. Ghanbarzadeh, E. Koç, S. Otri, S. Rahim, and M. Zaidi, "The Bees Algorithm, A Novel Tool for Complex Optimization Problems," Proc. 2nd Virtual International Conference on Intelligent Production Machines and Systems,
Elsevier (Oxford), pp. 454-459, 2006.

[17] Karl O. Jones, andréBouffe, Comparison of bees algorithm, ant colony optimization and particle swarm optimization for PID controller tuning, Proceedings of the 9th international Conference on Computer Systems and Technologies and Workshop For PhD Students in Computing, CompSysTech 08,374, ACM, New York, IIIA. 9-1 – IIIA. 9-6.

[18] Dong Hwa Kim, and Jae Hoon Cho, "Robust Tuning of PID Controller Using Bacterial-Foraging- Based Optimization," Journal of Advanced Computational Intelligence and Intelligent Informatics, vol. 9, no. 6, pp. 669-676, 2005.

[19] Stephens, D.W. and Krebs J. R., Foraging Theory, Princeton University Press, Princeton, New Jersey, 1986.

[20] Dong Hwa Kim1 and Jae Hoon Cho, "Adaptive Tuning ofPID Controller for Multivariable System Using Bacterial Foraging Based Optimization," Atlantic Web Intelligence Conference, pp. 231-235, 2005.

[21] Dong Hwa Kim, Jae Hoon Cho, "A Biologically Inspired Intelligent PID Controller Tuning for AVR Systems," International Journal of Control, Automation, and Systems, vol. 4, no. 5, pp. 624-636, 2006.

[22] K. M. Passino, Biomimicry of Bacterial Foraging for Distributed Optimization, University Press, Princeton, New Jersey, 2001.

[23] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," IEEE Control Systems Magazine, vol. 22, no. 3, pp. 52-67, 2002.

[24] M. Salami and G. Cain, "An Adaptive PID Controller Based on Genetic Algorithm Processor," IEE Genetic Algorithms in Engineering Systems, Innovations and Applications Conference, Publication No. 414, 12-14 September, 1995.

[25] S. M. G. Kumar, R. Jain, and N. Anantharaman, "Genetic Algorithm Based PID Controller Tuning for a Model Bioreactor," Indian Chemical Engineer, vol. 50, no. 3, pp. 214-226, 2008.

[26] S. Skogestad, "Simple Analytic Rules for Model Reduction and PID Controller tuning," Journal of Process Control, vol.13, pp. 291-309, 2003.

[27] B. W. Bequette, Process Control, Modeling, Design, and Simulation, Prentice Hall, Inc., New Jersey, 2003.

[28] Karl O. Jones and WilfriedHengue, Limitations of multivariable controller tuning using genetic algorithms, International Conference on Computer Systems and Technologies – CompSysTech 2009, IIIA.20-1 – IIIa.20.5.

[29] Neenu Thomas, P. Poongodi, "Position Control of DC Motor Using Genetic Algorithm Based PID Controller," Proceedings of the World Congress on Engineering, London, U.K., vol. 2 , pp. 1618-1622, 2009.

[30] M. B. B. Sharifian, R. Rahnavard and H. Delavari, "Velocity Control of DC Motor Based Intelligent methods and Optimal.