International Journal of Electrical, Electronics and Computer Engineering **5**(2): 37-40(2016)

ISSN No. (Online): 2277-2626

Review of Several Optimization Techniques for Control of Induction Motor

Rashmi Tiwari* and Dr. Satish Kumar Yawale**

**M. Tech. Scholar, Department of Electrical and Electronics Engineering, NRI Institute of science & Technology, Bhopal, (Madhya Pradesh), India*

***Asst. Professor, Department of Electrical and Electronics Engineering, NRI Institute of Science & Technology, Bhopal, (Madhya Pradesh), India*

(Corresponding author: Rashmi Tiwari)

(Received 19 October, 2016 Accepted 02 December, 2016)

(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: Power system divided into three parts; generation; transmission and distribution. At the distribution end most of the industrial loads are induction motors. In manufacturing industries they are used in large numbers because of its robustness and ease to control. The inductive nature of induction motor has a tendency to consume reactive power. Hence it became important to assess the reactive power demand of the machines in order to avoid the situation of voltage instability and voltage collapse. Extensive research is been carried out on reactive power compensation. Therefore, this paper presents a detailed review of several optimization techniques of induction motor for operation, planning and control of industrial engineering processes.

Index Terms: Induction motor; Soft Computing Technique, Neural Network; Genetic algorithm, optimization

I. INTRODUCTION

An induction motor is an electrical machine that is the basic building block of most of modern engineering systems. It is the part of the modern industries that connects to the engineering processes. The induction motor is the most crucial part of the process which helps in sustained production. Most of the motor-driven equipment's also controlled by governing induction motor. Various techniques are being presented by several authors for the governing or controlling the induction motor although many companies are still met with unforeseen reactive power demand and system failure (like voltage sag, interruption, voltage collapse, etc.). Enhancement in sensors and algorithms will deliver the essential tools for emerging failure detection. In the most of active research work, researchers proposed few frameworks for the control and optimization of induction motor. Not surprisingly, soft computing based algorithms were quickly exploited for the controls of these machines. In order to improve results several hybrid techniques like neuro-fuzzy (a combination of ANN and fuzzy logic), etc. are being applied for optimum flux to an indirect vector-controlled induction motor drive. Mostly torque, voltage, frequency and speed are used as inputs for the neural network. The change of iron core loss resistance due to flux and frequency variation plays an important part in deciding fuzzy logic set and has been taken into consideration during this study. In this paper, a detailed review study on the various soft computing technique for optimization and control studies has been carried out. Thus, a detailed literature review is presented in the next section. In addition, special attention is paid to Neural Network and Fuzzy logic techniques, which form the basis of the various power and efficiency estimation methods that would be discussed in this paper. The rest of the paper is organized as follows. Section 2 presents a details literature review of induction motor on the basis of soft computing techniques. Section 3 gives the conclusion and the future work which can be done in this field.

II. LITERATURE REVIEW

A hybrid technique neuro-fuzzy (a combination of ANN and fuzzy logic) is one of the soft computing techniques being applied for control of induction motor. There are also different types of controls which are responsible for governing the machine but the neural network based controller is the most accurate way to predict the changes in the output with reference to change in the input. In this section, we study several soft computing techniques to provide a proper understanding of induction motor for industrial applications.

Arroyo and Conejo [1] studied the unit commitment problem for thermal units. The unit commitment problem is the task of minimizing cost of fulfilling customer demand while taking into account startup costs, shut down costs, start up and shut down times, operating costs, and crew costs. The solution will give information about which units to activate when and in what order. Examples of complications are the fact that startup costs are modeled as a nonlinear function of how long the unit has been off, and that operating costs are nonlinear, non-differentiable functions of power output. Some constraints on this problem are that feasible solutions must be within start-up and shut down time limits, each unit has a minimum and maximum down time, a minimum and maximum output, and the desired performance must be possible with the amount of crew available.

Arroyo and Conejo investigate the use of a repair algorithm to deal with the constraints. There are several advantages for example all proposed strings are guaranteed to be feasible, and there is no need to come up with penalty functions to make sure that unfeasible strings have a low fitness function value. The disadvantage of the repair algorithm is that it takes a substantial amount of time to repair strings that are unfeasible to convert them into ones that are feasible which could be overcome by implementing a parallel structure. Park *et al.*, [2] address the generation expansion planning problem, which is the problem of how to minimize the cost of addition a plant to an existing structure, accounting for the type and number of plants, and still meet forecasted demand with a specified reliability. This problem is highly constrained, nonlinear, discrete, and dynamic. The authors cite previous works on similar problems that have used GAs but have displayed problem within the GAs such as premature convergence and duplications among strings. Premature convergence is a common problem with simple GAs in which there is one string in the initial random population that is so much more fit than all of the others that it duplicates over and over and takes over the whole population without ever giving the algorithm a chance to search for better possibilities. Three proposed improvements are a stochastic crossover scheme, elitism, and artificial initialization of the population.

They investigate the merits of three different crossover methods: one-point crossover, two-point crossover, and one-point substring crossover. One-point crossover switches the bits sequence between two parent strings at a randomly chosen point on the strings. Two-point crossover essentially does the same thing except that two points on the string are chosen at random so that a segment in the middle of each parent string is exchanged with the other parent. In one-point substring crossover, the string is divided at given intervals into substrings and each substring undergoes one-point crossover. The one-point substring crossover has the advantage of being able to mix together the parents' sequences well and promote diversity, but it has a high chance of destroying good bit structures that already exist. The one-point and two-point crossovers, on the other hand, are not very good at mixing up the bits but they are fairly harmless to already existing bit structures. This way, the fit string is always preserved. Variations would include preserving several of the fit strings instead of just one. Further, the case studies, stochastic crossover was more successful than artificial initial population, but hybridizing the two techniques yields even more impressive results.

Burke and Smith [3] worked with yet another scheduling problem, this time in thermal generator maintenance. They claim that using an integer representation instead of a binary representation for encoding possible solutions reduces the execution time of the GA since the strings are more compact. Unlike Park et al., they use penalty functions for unfeasible solutions. The primary purpose of the paper is to discuss different hybrid techniques of GAs and other search methods. The search methods were first tried individually on the problem described. The simple GA performed relatively badly, with simulated annealing performing better, and tabu-search performing the best. The hybrid tabu and simulated annealing worked better than simulated annealing on its own but still worse than the simple tabu. Hybrid hill-climbing and GA improved speed but produced relatively poor quality. The hybrid simulated annealing and GA algorithm was the worst as it both increased execution time and produced poor results. Finally, the most effective combination was found to be the GA with the tabu-search operator. This combination increased speed significantly. The tabu search was performed in two stages. In the first stage, penalty factors were made high to enforce feasibility. In the second stage, lower penalties were used to allow expanding searching.

These authors also used artificial population initialization, which used knowledge about the problem to seed the population with solutions that would have lower penalty factors than a randomly generation population. However, they discovered that the initial population fitness was not as significant in the hybrid methods as in the simple GA. They also found that penalty factors induced large differences in fitness values so that probabilistic reproduction caused premature convergence.

For this reason, a tournament selection was used, in which only the individuals fitness ranking in its population is taken into account, not the actual fitness value. Damousis *et al*., [4] investigated the network-constrained economic dispatch problem, which schedules online generating unit outputs to meet demand while at the same time operating at a minimum cost and within safety limits. Like Burke and Smith, they claimed that the binary representation could be improved upon. They went even further than using an integer representation and used a real-coded GA instead. Each place held a real number instead of a simple binary digit. This formulation allowed more precision and was found to produce more accurate results and complete faster than the traditional GA. Also, the quality did not decrease much with decreases in population size.

Milosevic and Begovic [5] investigated a Phasor Measurement Unit (PMU) problem. They used a non-dominated sorting algorithm to balance the two main objectives of minimizing the number of PMUs and maximizing the redundancy. These objectives conflicted with each other since improving one decreased the quality of the other. Non-dominated algorithms are used when there are two objectives that conflict and it is not clear how to weight them against each other. The algorithm suggests several optimal solutions with different weightings and the user decides between these solutions.

In this algorithm, there is a random initial population of feasible individuals. Any infeasible solutions will be repaired. The individuals are evaluated based on how non-dominant (diverse) they are. The population is divided into groups, called fronts that are close to each other in non-dominant characteristics. Diversity is maintained by using a sharing function. Sharing functions are based on the idea that there is only a limited quantity of resources for similar individuals. The fitness evaluation for an individual will be lower if there are many copies of that individual present. The case study results showed that the algorithm gave good performance for high population sizes and high values of crossover. Also, if crossover was high, the mutation rate needed to be inversely correlated with the size of the system.

Bakirtzis *et al.,* [6] attack another network constrained economic dispatch problem, this time using the terminology “optimal power flow”. This is a nonlinear, non-convex, large-scale, static problem with both continuous and discrete variables. Several enhancements were made. Three general enhancements were fitness scaling, elitism, and hill climbing. The first enhancement was fitness scaling by linear transformation. This is a scheme which artificially evaluated fitness values of different individuals as closer to each other than they were at the beginning of the run, while evaluating fitness values to be relatively further apart at the end of the run. This is often desirable because fitness values are usually very different in value in the initial population and may cause premature convergence. On the other hand, fitness values at the end of the run tend to be very close together so if the differences are not magnified there may be lack of convergence. The elitism is carried out in its simplest form by just preserving the most fit string from generation to generation. Hill climbing, due to its time consuming nature, is applied only to the best of each generation. Each bit of the individual is toggled one by one and the result is retained if it yields a higher fitness.

There were several problem specific gene (subsection) operators introduced. The first was a gene-swap, in which similar sections a given individual could be randomly exchanged. For example, if there were two subsections of the string that represented voltages, the voltage value at the first and second occurrence could be interchanged. Switching between different types of genes was not allowed. The second operator was a gene cross-swap, which was similar to the gene-swap except the genes were exchanged between different parents. The third operator was the gene copy, in which a gene was replaced by a copy of the predecessor or successor of that gene, on the same individual, that was the same type. The fourth operator was the gene inverse, which selected a random gene and reversed the order of all the bits. The final operator was the gene max-min, which selected a random gene and set (with equal probability) all bits to zeros or all bits to ones.

Tippayachai *et al.,* [7] provided yet another attack on the constrained economic dispatch problem. The authors suggested a parallel algorithm in which subpopulations were allowed to evolve and exchange solutions every epoch (a specified number of generations). Each subpopulation was randomly initialized with a certain bias to increase diversity. Elitism was used. Also, there was convergence checking with re-initialization. In this scheme, if the best string in a generation had a total number of bit differences with each other individual that was less than 5% of the total number of bits in the population, the population was said to have converged and the best individual was maintained while the rest were re-initialized. This also removed the need for a mutation operator.

Wu *et al.,* [8] used a diploid GA to solve a short-term scheduling problem. The concept of a diploid GA was based upon natural genetic structure.

A human, for example, carries not just one description of its gene structure, but two (one from each parent). Each description is called a chromosome. Therefore, for each gene, there are two opposing commands (alleles) on what form the gene should express. The decision of which one to actually implement is decided by dominance, for example, if a child inherits a blue eye genes from one parent and brown eye genes from another parent, it will express brown eyes because brown eyes are dominant over blue eyes.

This section presents the detailed overview of optimization study for the control and optimization of induction motor based on the existing techniques. The study provides the direction of the future work associated with the optimization and control of induction motor in the field of soft computing techniques. Furthermore, the changes in the value of various parameters can also be predicted with the help of discussed soft computing techniques.

III. CONCLUSION

The study provides the track of the future work associated with the optimization and control of induction motor in the field of soft computing techniques. We studied several soft computing techniques to provide a proper understanding of induction motor for industrial applications. Further based on the study and analysis from various researchers, we have noticed that the fuzzy logic controller and neural network controller provides a new direction for the optimization and control of induction motor. Furthermore, these techniques will help the system for planning of adequate reactive power support in electrical system for the increase in capacity of existing process industries.

REFERENCES

[1]. H. R. Andersen and J. K. Pedersen, "Low cost energy optimized control strategy for a variable speed three-phase induction motor," in *Proc. IEEE PESC'96*, Baveno, Italy, 1996, pp. 920–924.

[2]. G. C. D. Sousa, B. K. Bose, and J. G. Cleland, "Fuzzy logic based online efficiency optimization control of an indirect vector-controlled induction motor drive," *IEEE Trans. Ind. Electron*., vol. **42**, pp. 192– 198, Apr. 1995.

[3]. G. O. Garcia, J. C. Mendes Luis, R. M. Stephan, and E. H. Watanabe, "An Efficient Controller for an Adjustable Speed Induction Motor Drive", *IEEE Trans. on Industrial Electronics*, Vol. **41**, No.5, pp 533 -539, Oct 1994.

[4]. P. Famouri and J. L. Cathey, "Loss minimization control of an induction motor drive," *IEEE Trans. Ind. Applicat.*, vol. **27**, pp. 32–37, Jan./Feb. 1991.

[5]. S. K. Sul and M. H. Park, "A novel technique for optimal efficiency control of a current source inverter fed induction motor," *IEEE Trans. Power Electron.*, vol. **3**, pp. 192–199, Mar. 1988

[6]. D. S. Kirschen, D. W. Novotny, and T. A. Lipo, "Optimal efficiency control of an induction motor drive," *IEEE Transhtc cu u. Energy Conversion*, vol. **EC-2**, pp. 70–76, Mar. 1987.

[7]. D. S. Kirschen, D. W. Novotny, and T. A. Lipo, "On line efficiency optimization of a variable frequency induction-motor drive," *IEEE Trans. Ind. Applicat*., vol. **IA-21**, pp. 610–616, July/Aug. 1985.

[8]. D. S. Kirschen, D. W. Novotny, and W. Suwanwisoot, "Minimizing induction motor losses by excitation control in variable frequency drives," *IEEE Trans. Ind. Applicat.*, vol. **IA-20**, pp. 1244–1250, Sept./Oct. 1984.