

Genetic Algorithms: Concepts, Design for Optimization of Process Controllers

Rahul Malhotra, Narinder Singh & Yaduvir Singh

Punjab Technical University, Jalandhar, Punjab, India

Tel: 91-94-6302-0007 E-mail: blessurahul@gmail.com

Abstract

Genetic Algorithm is a search heuristic that mimics the process of evaluation. Genetic Algorithms can be applied to process controllers for their optimization using natural operators. This paper discusses the concept and design procedure of Genetic Algorithm as an optimization tool. Further, this paper explores the well established methodologies of the literature to realize the workability and applicability of genetic algorithms for process control applications. Genetic Algorithms are applied to direct torque control of induction motor drive, speed control of gas turbine, speed control of DC servo motor for the optimization of control parameters in this work. The simulations were carried out in simulink package of MATLAB. The simulation results show better optimization of hybrid genetic algorithm controllers than fuzzy standalone and conventional controllers.

Keywords: Evolutionary Algorithms, Genetic algorithms, DTC Induction motor, Turbine compressor system, DC servo motor

1. Introduction

Evolutionary algorithms (EAs) are population-based meta heuristic optimization algorithms that use biology-inspired mechanisms and survival of the fittest theory in order to refine a set of solution iteratively. Genetic algorithms (GAs) are subclass of evolutionary algorithms (EAs) where the elements of the search space are binary strings or arrays of other elementary types. Genetic algorithms (GAs) are computer based search techniques patterned after the genetic mechanisms of biological organisms that have adapted and flourished in changing highly competitive environment. Last decade has witnessed many exciting advances in the use of genetic algorithms (GAs) to solve optimization problems in process control systems. Genetic algorithms (GAs) are the solution for optimization of hard problems quickly, reliably and accurately. As the complexity of the real-time controller increases, the genetic algorithms (GAs) applications have grown in more than equal measure.

One of the most fundamental principal in our world is the search for an optimal state. Optimization is the process of modifying the inputs or characteristics of a device, mathematical process to obtain minimum or maximum of the output. The input to the optimization process is the cost function, objective function or fitness function and the output is the fitness function of the system. Optimization is a primary tool, needed to tackle the unsolvable or hard problems. Optimization algorithms can be characterized into five categories. In trial and error optimization, the processes affect the output without knowing about the constraints, responsible to produce the output. A mathematical formula describes the objective function for optimization.

One dimensional optimization contains one variable and a problem having more than one variable requires multi-dimensional approach. As the number of dimensions increases, the process of optimization becomes difficult. Dynamic optimization is time dependent, whereas the static optimization is independent of time. The static problem is difficult to solve for finding the best solution but the added dimension of time increases the challenge of solving dynamic problems. Discrete variable optimization contains only a finite number of possible values, whereas continuous variables have an infinite number of possible values. Variables often have limits or constraints. Constrained optimization incorporates variable equalities and inequalities into the cost function, whereas unconstrained optimization allows the variable to take any value. A constrained optimization problem can be converted into unconstrained one through the transformation of variables.

Many optimization algorithms have been developed in their original form. The goal of global optimization is to find the global optima, that is, global maxima or minima of the objective function. Optimization problems are used to find good parameters or designs for components or plans to be put into action by the human beings or machines.

This paper discusses the various concepts and design of genetic algorithms for optimization of process controllers. Section I gives the basic introduction of genetic algorithms and optimisation procedure. Working procedure, algorithm and the flow chart representation of Genetic algorithm is explained in section II. Encoding and Selection techniques in genetic algorithms have been discussed with examples in section III and IV. Section

V discusses the various genetic algorithm operators i.e. mutation and crossover. Section VI discusses the various issues in genetic algorithms. Section VII shows the broad overview of workability and applicability of genetic algorithms to various process controllers. Section VIII shows the implementation of genetic algorithms optimization to control non-linear direct torque control of induction motor drive. Section IX shows the turbine compressor system optimization using genetic algorithms. Section X shows the applicability of genetic algorithms to control the speed of DC servo motor. Abridged, the superiority of genetic algorithms have been discussed in section XI.

2. Working Principle of Genetic Algorithms (GAs)

The workability of genetic algorithms (GAs) is based on Darwinian's theory of survival of the fittest. Genetic algorithms (GAs) may contain a chromosome, a gene, set of population, fitness, fitness function, breeding, mutation and selection. Genetic algorithms (GAs) begin with a set of solutions represented by chromosomes, called population. Solutions from one population are taken and used to form a new population, which is motivated by the possibility that the new population will be better than the old one. Further, solutions are selected according to their fitness to form new solutions, that is, offsprings. The above process is repeated until some condition is satisfied. Algorithmically, the basic genetic algorithm (GAs) is outlined as below:

- Step I [Start] Generate random population of chromosomes, that is, suitable solutions for the problem.
- Step II [Fitness] Evaluate the fitness of each chromosome in the population.
- Step III [New population] Create a new population by repeating following steps until the new population is complete.
 - a) [Selection] Select two parent chromosomes from a population according to their fitness. Better the fitness, the bigger chance to be selected to be the parent.
 - b) [Crossover] With a crossover probability, cross over the parents to form new offspring, that is, children. If no crossover was performed, offspring is the exact copy of parents.
 - c) [Mutation] With a mutation probability, mutate new offspring at each locus.
 - d) [Accepting] Place new offspring in the new population.
- Step IV [Replace] Use new generated population for a further run of the algorithm.
- Step V [Test] If the end condition is satisfied, stop, and return the best solution in current population.
- Step VI [Loop] Go to step 2.

The genetic algorithms performance is largely influenced by crossover and mutation operators. The block diagram representation of genetic algorithms (GAs) is shown in Fig.1.

3. Encoding Technique in Genetic Algorithms (GAs)

Encoding techniques in genetic algorithms (GAs) are problem specific, which transforms the problem solution into chromosomes. Various encoding techniques used in genetic algorithms (GAs) are binary encoding, permutation encoding, value encoding and tree encoding.

3.1 Binary encoding

It is the most common form of encoding in which the data value is converted into binary strings. Binary encoding gives many possible chromosomes with a small number of alleles. A chromosome is represented in binary encoding as shown in Figure 2 (a).

3.2 Permutation encoding

Permutation encoding is best suited for ordering or queuing problems. Travelling salesman is a challenging problem in optimization, where permutation encoding is used. In permutation encoding, every chromosome is a string of numbers in a sequence as shown in Figure 2 (b).

3.3 Value encoding

Value encoding can be form number, real number on characters to some complicated objects. Value encoding is technique in which every chromosome is a string of some values and is used where some more complicated values are required. It can be expressed as shown in Figure 2 (c).

3.4 Tree Encoding

It is best suited technique for evolving expressions or programs such as genetic programming. In tree encoding, every chromosome is a tree of some objects, functions or commands in programming languages.

Locator/identifier separation protocol (LISP) programming language is used for this purpose. Locator/identifier separation protocol (LISP) programs can be represented in tree structure for crossover and mutation. In tree encoding, the chromosomes are represented as shown in Figure 2 (d).

There are no specific directions for using the type of encoding scheme in the specified problem rather, it depends upon the applicability and the requirements of the problem.

4. Selection Techniques in Genetic Algorithms (GAs)

Selection is an important function in genetic algorithms (GAs), based on an evaluation criterion that returns a measurement of worth for any chromosome in the context of the problem. It is the stage of genetic algorithm in which individual genomes are chosen from the string of chromosomes. The commonly used techniques for selection of chromosomes are Roulette wheel, rank selection and steady state selection.

4.1 Roulette wheel selection

In this method the parents are selected according to their fitness. Better chromosomes, are having more chances to be selected as parents. It is the most common method for implementing fitness proportionate selection. Each individual is assigned a slice of circular Roulette wheel, and the size of slice is proportional to the individual fitness of chromosomes, that is, bigger the value, larger the size of slice is. The functioning of Roulette wheel algorithm is described below:

Step 1 [Sum] Find the sum of all chromosomes fitness in the population.

Step 2 [Select] Generate random number from the given population interval.

Step 3 [Loop] Go through the entire population and sum the fitness. When this sum is more than a fitness criteria value, stop and return this chromosome.

Figure 3 (a) shows Roulette wheel for six individuals having different fitness values. The Sixth individual has a higher fitness than any other, it is expected that the Roulette wheel selection will choose the sixth individual more than any other individual.

4.2 Rank selection method

The application of Roulette wheel selection method is not satisfactory in genetic algorithms (GAs), when the fitness value of chromosomes differs very much. It is a slower convergence technique, which ranks the population by certain criteria and then every chromosome receives fitness value determined by this ranking. This method prevents quick convergence and the individuals in a population are ranked according to the fitness and the expected value of each individual depends on its rank rather than its absolute fitness.

The rank selection method is shown in Figure 3 (b). For example, if the best chromosome fitness is 80 percent, its circumference occupies 80 percent of the roulette wheel and then other chromosomes will have minimum chances to be selected. On the other hand, the rank selection first ranks the population according to their fitness and then every chromosome receives ranking. The worst will have fitness 1, the second worst will have a fitness of 2, and the best one will have a fitness value n, where n is the number of chromosomes in the population.

4.3 Steady-state selection

This method replaces few individuals in each generation, and is not a particular method for selecting the parents. Only a small number of newly created offsprings are put in place of least fit individual. The main idea of steady-state selection is that bigger part of chromosome should retain to successive population.

5. Genetic Algorithms (GAs) Operators

Genetic algorithms (GAs) can be applied to any process control application for optimization of different parameters. Genetic algorithms (GAs) use various operators viz. the crossover, mutation for the proper selection of optimized value. Selection of proper crossover and mutation technique depends upon the encoding method and as per the requirement of the problem.

5.1 Crossover

It is the process in which genes are selected from the parent chromosomes and new offspring is created. Crossover can be performed with binary encoding, permutation encoding, value encoding and tree encoding.

5.1.1 Binary encoding crossover

In binary encoding, the chromosomes may crossover at single point, two point, uniformly or arithmetically. In single point crossover, a single crossover point is chosen and the data before this point are exactly copied from

first parent and the data after this point are exactly copied from the second parent to create new offsprings. Two parents in this method give two new offsprings. The single point crossover is illustrated in Figure 4 (a).

5.1.2 Uniform Crossover

In uniform crossover, data of the first parent chromosome and second parent chromosome are randomly copied, which is illustrated in Figure 4(b).

5.1.3 Arithmetic Crossover

In arithmetic crossover, crossover of chromosomes is performed by AND and OR operators to create new offsprings as illustrated in Figure 4(c).

5.1.4 Permutation encoding crossover

In permutation encoding crossover, one crossover point is selected. The permutation is copied from first parent chromosome upto the point of crossover and the other parent chromosome is exactly copied to ensure that no number is left to be put in the offspring. Further, if the number is not yet in the offspring, it is added to the offspring chromosome. Travelling salesman problems and task ordering problems can be easily solved by permutation encoding. Figure 4(d) illustrates the single point crossover with permutation encoding.

5.1.5 Value encoding crossover

It can be performed at single point, two point, uniform and arithmetic representation as in binary encoding technique. Figure 4(e) illustrates the single point crossover with value encoding.

5.1.6 Tree encoding crossover

In this type of crossover, one point of crossover is selected in both parent tree chromosomes, which are divided at a point. The parts of tree below crossover point are exactly exchanged to produce new offsprings, which is illustrated in Figure 4 (f).

The choice of the type of the crossover is strictly depends upon the problem.

5.2 Mutation

Premature convergence is a critical problem in most optimization techniques, consisting of populations, which occurs when highly fit parent chromosomes in the population breed many similar offsprings in early evolution time. Crossover operation of genetic algorithms (GAs) can not generate quite different offsprings from their parents because the acquired information is used to crossover the chromosomes. An alternate operator, mutation, can search new areas in contrast to the crossover. Crossover is referred as exploitation operator whereas the mutation is exploration one. Like crossover, mutation can also be performed for all types of encoding techniques.

5.2.1 Binary encoding mutation

In binary encoding mutation, the bits selected for creating new offsprings are inverted, which is illustrated in Figure 5 (a).

In binary encoding mutation, if the bit 1 is converted into bit 0, it decreases the numerical value of the chromosome, and is known as down mutation. Similarly, if the bit 0 is converted into bit 1, the numerical value of the chromosome increases and is referred as up mutation.

5.2.2 Permutation encoding mutation

In permutation encoding mutation, the order of the two numbers given in a sequence are exchanged which is illustrated in Figure 5(b).

5.2.3 Value encoding mutation

In value encoding mutation, a small numerical value is either added or subtracted from the selected values of chromosomes to create new offsprings, which is illustrated in Figure 5(c).

5.2.4 Tree encoding mutation

Tree encoding mutation, mutates the certain selected nodes of the tree to create new offspring, which is illustrated in Figure 5(d).

6. Genetic algorithms (GAs): Issues

Genetic algorithms (GAs) can be applied in complex non-linear process controllers for the optimization of parameters. Some issues are important to be considered for proper implementation of genetic algorithms (GAs) to a plant to be optimized.

Deciding of population size is an important issue while applying genetic algorithms (GAs). It is recommended by the researchers, that the population size should be of about 20 to 30 chromosomes. A very big population size consumes more time for finding optimum solution, which may deteriorate the performance of genetic algorithms (GAs).

Genetic algorithms (GAs) may suffer from the problem of premature convergence due to improper selection of crossover rates. Higher crossover rate of about 85 percent to 95 percent is recommended to minimize premature convergence problems.

Low mutation rate of about 0.5 percent to 1 percent is generally recommended to obtain optimized results from genetic algorithm. Mutation is an artificial and forced method of changing the numerical value of the chromosome. Mutation should be avoided as far as possible because it is totally adhoc and random in nature. Small mutation rates prevent genetic algorithms from falling into local maxima or minima.

Deciding of selection method for selecting good chromosomes is another important issue while applying genetic algorithms for process control applications. Rank selection method and roulette wheel selection methods have shown good results over other methods of selection.

Genetic algorithms (GAs) play an important role in process control applications for the optimization of parameters. Many researchers have contributed in this field. A broad review on genetic algorithm applications gives the directions to use this technique for the optimization of the process controllers.

7. A Review of Workability and applicability of Genetic Algorithms (GAs)

A computational technique based on a real coded genetic algorithm for microwave imaging purposes was discussed by Carsi. S. (2000). This study solves a non-linear inverse scattering problem for short range microwave imaging. Comparative analysis of the results obtained by approximate formulations and binary coded genetic algorithms (GAs) is made. Further, a hybrid version is presented and preliminary tested.

A dynamic routing control based on genetic algorithm can provide flexible real time management of the dynamic traffic changes in broadband networks. It was demonstrated through computer simulations using genetic algorithms by Shimamoto N. (2000). The proposed technique can generate the exact solution of path arrangement that keeps the traffic loss rate below the target value, even after changes in traffic.

A genetic algorithm for shortest path routing problem and the sizing of population was applied by Chang Wook et.al (2002). Variable length chromosomes and their genes have been used for encoding the problem. The proposed algorithm can cure all infeasible chromosomes with a simple repair function. A population sizing equation is emphasized using computer simulations.

The multiple destination routing algorithm was formulated for finding a minimal cost tree which contains designated source and multiple destination nodes to satisfy certain constraints in a given communication network. The simulation studies for sparse and dense network demonstrate the robustness and efficiency of the proposed algorithm in terms of yielding high quality solutions.

A novel approach to solve very large scale integration (VLSI) channel and switch box routing problems was discussed by Lienig et.al (1997). This approach is based upon genetic algorithms that run on a distributed network of workstations. An extensive investigation shows the qualitatively better results and significantly reduction in occurrence of cross talk.

Usefulness of heuristic algorithms as the search method for diverse optimization problems is examined by Jang Sung Chun et.al (1998). Immune algorithms, genetic algorithms, evolutionary algorithms were compared on diverse optimization problems and the results reveal the outperformance of genetic algorithms. Based on genetic algorithms, surface permanent magnet synchronous motor is designed.

The problem of premature converge in genetic algorithms optimization was discussed by Mori N. et.al (1996). A novel thermodynamical genetic algorithm approach was suggested, which adopts the concepts of temperature and entropy in the selection rule. Simulated annealing was used to maintain diversity of the population. A comparison of the thermodynamical genetic algorithm approach with the simple genetic algorithm is carried out taking a knapsack problem.

Takagi-Sugeno-Kang (TSK) type recurrent fuzzy network is proposed by Chia Feng Juang (2002), which develops from a series of recurrent fuzzy if-then rules with Takagi-Sugeno-Kang (TSK) type consequent parts. Takagi-Sugeno-Kang (TSK) type recurrent fuzzy network with supervised learning is suggested for the problems having online training data. To demonstrate the superiority of Takagi-Sugeno-Kang (TSK) type recurrent fuzzy

network, it is applied to dynamic system. By comparing the results the efficiency of Takagi-Sugeno-Kang (TSK) type recurrent fuzzy network is verified.

Optimization of antenna and scattering patterns using genetic algorithms was demonstrated by Haupt et.al (1995). The proposed algorithm encodes each parameter into binary sequences called a gene and a set of gene is called chromosome. Several examples have been taken and implemented in MATLAB. For optimal solution of antenna patterns and back scattering radar cross-section pattern.

An orthogonal genetic algorithm approach for multimedia multicast routing was suggested by Qing Fu Zhang et.al (1999). It can be investigated that the search space is stastically sound and is well suited for parallel implementation and execution. The implementation results reveal that the orthogonal genetic algorithms can find near optimal solution within moderate number of generations for practical problem sizes.

Dimeo R and Lee (1995) suggested a novel approach for boiler turbine control system using genetic algorithms. Proportional integral controller and a state feedback controller for non-linear multi-input-multi-output (MIMO) plant model is developed. Experimental results show the optimal control of boiler using genetic algorithms.

Channel assignment problem in hexagonal cellular network with two band buffering was discussed by Ghosh S. C. et.al (2003). An algorithm is presented for solving channel assignment problem using etilist model of genetic algorithm which shows the optimal results within a reasonable computational time.

A vehicular wire antenna was designed using genetic algorithms used for both global positioning system (GPS) and indium systems. The antenna was simulated using numerical electromagnetic codes and then fabricated and tested. The voltage standing wave ratio (VSWR) and circular polarization radiation patterns were compiled and measured. Altshuler et.al (2000) suggested a new approach, which uses genetic algorithms in conjunction with electromagnetic code, produces configurations that are unique and seem to outperform more conventional design.

Design of direct form of a finite word length, finite impulse response (FIR) low pass filter was proposed by Xu and Daley (1995). The results of the proposed design techniques are compared with an integer programming technique and it is inferred from the results that genetic algorithm based technique outperforms the traditional approach.

Design of optimal disturbance rejection using genetic algorithms was suggested by Krohling and Rey (2001). The method was proposed to design an optimal disturbance rejection proportional integral derivative (PID) controller. A condition for disturbance rejection of control system is described which is further formulated as a constrained optimization problem. A constraint optimization problem to optimize integral of time and absolute error (ITAE) was tested by proportional integral derivative (PID) controller as applied to servo motor system. A double genetic algorithm was applied for solving constraint optimization problem. Simulation results demonstrate the performance and validity of the methods.

Length of Yagi Uda antenna was optimized by Jones and Jones (1997) using genetic algorithms. To illustrate the capabilities of the method, the length and spacing of several Yagi-uda antennas are optimized for various performance characteristics.

Genetic algorithms were applied to pattern recognition problem by Raymer M. L. et.al (2000). A new approach is suggested to feature extraction in which feature selection and feature extraction was simultaneously done using genetic algorithms. The genetic algorithm optimizes a feature weight vector used to scale the individual features in the original pattern vectors.

Wei-Yen Wang and Li (2003) proposed a novel approach to adjust both the control points of B-spline membership functions and the weights of fuzzy neural network using reduced form genetic algorithms (RGA). Simulation results show the faster convergence of the evolution process and effectiveness of reduced form genetic algorithms.

Problem of finding robust or flexible solutions for scheduling problems for real world application was suggested by Jensen M. T, (2003). Experimentally, it is shown that using a genetic algorithm, it is possible to find robust and flexible schedules with a low makespan.

Scheduling of hydraulically coupled plants can be approximated by genetic algorithms. An effective approach was suggested by Po-Hung Chen and Chang (1996) to 24 hrs ahead generation scheduling of hydraulically coupled plants. Experimental results show that the genetic algorithm approach obtains a more highly optimal solution than the conventional dynamic programming method.

Design of finite impulse response filter using genetic algorithms was suggested by Suckley D. et.al (1991). Genetic algorithm was used for automatic rapid and minimal computational complexity to design a filter.

An early paper in terms of genetic algorithms and its applications was presented by Tang K. S et.al (1996) which elaborates the genetic algorithm technology and its comparison with other optimization techniques. The genetic algorithm procedures were discussed to implement signal processing applications for infinite impulse response (IIR) adaptive filtering, time delay estimation, active noise control and speech processing, which are being implemented and described.

Hong Y. (2002) applied genetic algorithms on economic dispatch for congregation units considering multiplant multibuyer wheeling, which transmits microwaves to design load buses via wheeling. Varying the weights coefficient for penalty functions and determination of gene variables using genetic algorithms were discussed. The IEEE 30 and IEEE 118 bus system were used as test systems to illustrate the applicability of the proposed method.

Engineering applications of genetic algorithms were introduced by Man & tang (1996). This study reveals how genetic algorithms can be integrated to form the framework of design tool for industrial engineers. An attempt has also been made to explain why, when and how to use genetic algorithms as an optimization tool for process controllers.

To find the shortest path, genetic algorithms can be used to encode a path in graph into a chromosome. The proposed approach have been tested by Gen M. et.al (1997) with different size from 6 nodes to 70 nodes and from 10 edges to 211 edges. The encouraging results using genetic algorithms can find the optimum very rapidly and with very high probability.

Genetic algorithm can be applied for system identification of both continuous and discrete time systems. They are effective in both domains for finding poles and zeroes. Simulations for minimum and non-minimum phase systems and a system with unmodeled dynamics were presented by Kristinsson and Dumont (1992).

Harmonic optimization of multilevel converter using genetic algorithm was proposed by Ozpineci et.al (2004). The optimization technique is applied to multilevel inverter to determine optimum switching angles for cascaded multilevel inverters for eliminating some higher order harmonics while maintaining the required fundamental voltage.

Simulations tuning of power system damping controller using genetic algorithms was done by Do Bomfim et.al (2000). Damping controller structures are assumed to be fixed consisting lead lag filter during the study. The study reveals the efficency of genetic algorithms for the proposed system.

It is inferred from the above literature review on genetic algorithms that they are successfully applied in many optimization problems in process control systems using genetic operators like selection, crossover and mutation. In this work Genetic Algorithms are applied to optimize the process controllers.

8. Direct Torque Control using Genetic algorithms (GAs)

Genetic algorithms (GAs) are the methods for solving optimization problems, based on natural selection. Genetic algorithms (GAs) consider individuals in a population called as strings or chromosomes. Genetic algorithm (GAs) repeatedly modifies the population of individual solutions. At each step, the genetic algorithms (GAs) select individuals at random from the current population to be parents and use them to produce the children for the next generation (Biswal B. et.al, 2007). They can be applied to solve a variety of optimization problems that are not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, non-differentiable, stochastic or highly non-linear.

The design of fuzzy sets and the rule base has been automated by the use of genetic algorithms (GAs) which are powerful search and optimization techniques based on the principles of natural evolution and population genetics. Genetic algorithms (GAs) does not rely on computing local derivatives to guide the search process, all they need is an objective function. There are several strategies that can be used by genetic algorithms for constrained optimization problems (Cheong F. and Lai, 2000) and existing strategies can be roughly classified as rejection, repairing, modifying and elitism of genetic operators. Several techniques (Habetler T. G. et.al, 1992) have been proposed to use genetic algorithms (GAs) for optimization, but there are no general guidelines for its formulation as it is often problem dependent. Genetic algorithms (GAs) is used with fuzzy logic (FL) model to enhance the reliability of the system. Genetic algorithms (GAs) based fuzzy rules increase the effectiveness and feasibility of the control system.

The various genetic algorithm parameters used for fuzzy logic rule base are as in Table 1.

Digital simulations have been carried out MATLAB/Simulink. During the simulation, using genetic algorithms (GAs), the population size of 20 chromosomes, run for 150 generations and the Roulette-Wheel method for selection with normalized fitness values is used. One point crossover is applied to select individuals and the mutation per generations was always applied. As the coding of chromosomes is realized directly with integers, uniform mutation is used.

The motor is subjected to a switching frequency of 10 Kiloherz, for a sampling time of 1ms. The torque and the flux reference values have been fixed as 2.5 Nm and 0.5 Wb respectively. An index error has been described by integral of time and absolute error (ITAE). It is inferred from the simulation results, that the genetic algorithm controller produces comparatively less torque error and flux error at different operating conditions. It is also important to note that the genetic algorithm shows the remarkable reduction of torque errors and flux errors at the initial or low speed operation of an induction motor.

9. Optimization of Turbine Compressor System using Genetic algorithms (GAs)

Genetic algorithms (GAs) are intelligent optimization technique that relies on the parallelism found in nature, in particular its searching procedures which are based on the mechanics of natural selection and genetics. Genetic algorithms (GAs) are used to solve difficult search, optimization, and machine-learning problems that have previously resisted automated solutions. They can be used to solve difficult problems quickly and reliably.

In genetic algorithms the transient response parameters are better optimized than fuzzy logic controller (FLC) and conventional proportional integral derivative (PID) controller. To verify the results of SIMULINK based model of turbine compressor system using optimum block set, m-file were generated using MATLAB commands. Through comparative analysis of flow control of turbine compressor system, genetic algorithms (GA) shows almost negligible overshoot of about 1% and the excellent improvement in settling time and peak time whose typical values are about 3.7 seconds and 4.1 seconds respectively, for controlling the speed of a typical turbine.

10. Optimization of DC Servo Motor using Genetic algorithms (GAs)

Genetic algorithms are useful approach to the problems requiring effective and efficient searching. This strategy is proposed to optimize the performance of the fuzzy logic controlled speed control of DC motor. Such strategies show robustness against parameter variation, better external disturbance rejection and stability. Due to its effectiveness in searching non-linear, multidimensional search spaces, genetic algorithms (GAs) can be applied to the tuning of proportional integral (PI) speed controller to cope up with the existing non-linearities in the motor. In this case, the fitness function to evaluate the individuals of each generation can be chosen to be the reciprocal of integral time of absolute error (ITAE). After simulation, it is investigated that the overshoot becomes 3.2 percent and settling time and peak time are noticed about 1.5 seconds and 1.7 seconds respectively which shows a significant improvement than fuzzy logic (FL) standalone controller and the conventional proportional integral (PI) controller. During the search process, the genetic algorithm looks for the optimum setting of proportional integral (PI) speed controller gains which minimizes the performance indices viz. integral of absolute error (IAE) and integral of time and absolute error (ITAE). The lowest performance indices are considered as the fittest. This function is used as the genetic algorithm evaluation criteria. The optimal blockset of MATLAB/Simulink shows the minimum error obtained by using genetic algorithm based controller than fuzzy logic standalone and conventional controllers.

11. Conclusion and Future Scope

Genetic algorithms can be applied to domains in which insufficient knowledge of the system and/or high complexity is there. Genetic algorithms can find optimal solutions among the search space with the operators like crossover and mutation. Genetic algorithms are very effective techniques of quickly finding a reasonable solution to a complex problem. They are not instantaneous, but can perform an excellent search. In this work, Genetic algorithm is tested to optimize the errors of Direct Torque Control of induction motor, turbine Compressor System and DC servo motor, which shows the superiority of Genetic Algorithm than standalone soft computing controllers and conventional controllers. In future work, these techniques may be implemented for other process controllers. Other Evolutionary techniques like may further be implemented for optimisation.

References

Atef Saleh Othman Al-Mashakbeh. (2009). PID Control of DC Motor, *European Journal of Scientific Research*, 35(2), pp. 198-203.

- Belkacem Mahdad, Tarek Bouktir and Kamel Srairi. (2008). Optimal power Flow of the Algerian Network using Genetic Algorithms/Fuzzy Rules, Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, pp. 1-8.
- Bhim Singh, B P Singh and (Ms) K Jain. (2003). Implementation of DSP Based Digital Speed Controller for Permanent Magnet Brushless dc Motor, *IE(I) Journal-EL*, 84, pp.16-21.
- B Mrozek, Z Mrozek. (2000). Modeling and Fuzzy Control of DC Drive, in Proceedings of 14th European Simulation Multi conference, pp 186-190.
- Chu-Kuei Tu and Tseng-Hsien Lin. (2000). Applying Genetic Algorithms On Fuzzy Logic System For Underwater Acoustic Signal Recognition, Proceedings of the 2000 International Symposium on Underwater Technology, pp. 405-410.
- Ching-Hung Wang, Tzung-Pei Hong, and Shian-Shyong Tseng. (1998). Integrating Fuzzy Knowledge by Genetic Algorithms, *IEEE Transactions On Evolutionary Computation*, 2(4), pp. 138-149.
- C.-K. Lee, and W.-H. Pang. (1994). A Brushless DC Motor Speed Control System Using Fuzzy Rules, *Power Electronics and Variable-Speed Drives Conference*, pp. 101-106.
- D. P. Kadam S. S. Wagh P. M. Patil. (2007). Thermal Unit Commitment Problem by using Genetic Algorithms, Fuzzy Logic and Priority List Method, International Conference on Computational Intelligence and Multimedia Applications, 1, pp.468-472.
- DePold, H., and Gass, F. D. (1999). The Application of Expert Systems and Neural Networks to Gas Turbine Prognostics and Diagnostics, *Journal of Engineering for Gas Turbine and Power*, 121(4), pp. 607–612.
- G.Madhusudhana Rao, B.V.SankerRam, B.Smapath Kumar and K.Vijay Kumar. (2010). Speed Control of BLDC Motor Using DSP, *International Journal of Engineering Science and Technology*, 2(3), pp.143-147.
- Hyun-Joon Cho, Kwang-Bo Cho and Bo-Hyeun Wang. (1996). Automatic rule generation using Genetic Algorithms for Fuzzy-PID hybrid control, Proceedings of the 1996 IEEE International Symposium on Intelligent Control, pp. 271-276.
- James M. Adams and Kuldip S. Rattan. (2001). Genetic multi-stage Fuzzy PID controller with a Fuzzy switch, *IEEE Transactions on Systems, Man, and Cybernetics*, 4, pp.2239-2244.
- K. B. Mohanty, A. Routray, and N. K. De. (2001). Design of a fuzzy sliding mode controller for a field oriented induction motor drive, *Journal of Systems Society of India, 'Paritantra'*, 6(1), pp. 8-16.
- K. B. Mohanty. (2001). Design and comparative study of sliding mode and fuzzy sliding mode controllers for an induction motor drive, Procc. of 25th National Systems Conf., Coimbatore, pp 66-71.
- Mehmet Akar and Ismail Temiz. (2007). Motion controller design for the speed control of dc servo motor, *International Journal of Applied Mathematics and Informatics*, 1(4), pp.131-137.
- P. K. Nandam, and P. C. Sen. (1986). A comparative study of proportional-integral (P-I) and integral-proportional (I-P) controllers for dc motor drives, *Int. Jour. of Control*, 44, pp. 283-297.
- Raj Subbu, Arthur C. Sanderson and Piero P. Bonissone. (1998). Fuzzy Logic Controlled Genetic Algorithms versus Tuned Genetic Algorithms: An Agile Manufacturing Application, Proceedings of the Intelligent Control (ISIC), IEEE Joint Conference, pp.434-440.
- S. H. Yakhchali and S. H. Ghodsspour. (2008). A Hybrid Genetic Algorithms for Computing the Float of an Activity in Networks with Imprecise Durations, Proceedings of the IEEE International Conference on Fuzzy Systems, pp.1789-1794.
- W.-S. Oh, Y.-T. Kim, C.-S. Kim, T.-S. Kwon, and H.-J. Kim. (1999). Speed Control of Induction Motor Using Genetic Algorithm Based Fuzzy Controller, *Industrial Electronics Society Conference of IEEE*, 3(2), pp. 625-629.
- Shimamoto, N., Hiramatsu, A. and Yamasaki, K. (2000). A dynamic routing control based on a genetic algorithm, *IEEE International Conference on Neural Networks, 1993*, vol.2, pp. 1123-1128.
- Chang Wook Ahn and Ramakrishna, R.S. (2002). A genetic algorithm for shortest path routing problem and the sizing of populations, *IEEE Transactions on Evolutionary Computation*, 6(6), pp. 566-579.
- Lienig, J. (1997). A parallel genetic algorithm for performance-driven VLSI routing, *IEEE Transactions on Evolutionary Computation*, 1(1), pp. 29-39.

- Jang-Sung Chun, Hyun-Kyo Jung and Song-Yop Hahn. (1998). A study on comparison of optimization performances between immune algorithm and other heuristic algorithms, *IEEE Transactions on Magnetics*, 34 (5), pp. 2972-2975.
- Mori, N., Yoshida, J., Tamaki, H. and Nishikawa, H.K. (1996). A thermodynamical selection rule for the genetic algorithm, *IEEE International Conference on Evolutionary Computation*, 1995, pp. 188-193.
- Chia-Feng Juang. (2002). A TSK-type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithms, *IEEE Transactions on Fuzzy Systems*, 10 (2), pp. 155-170.
- Haupt, R.L. (1995). An introduction to genetic algorithms for electromagnetics, *IEEE Antennas and Propagation Magazine*, 37(2), pp. 7-15.
- Qing-fu Zhang and Yiu-Wing Leung. (1999). An orthogonal genetic algorithm for multimedia multicast routing, *IEEE Transactions on Evolutionary Computation*, 3(1), pp. 53-63.
- Dimeo R., and Lee K.Y. (1995). Boiler-turbine control system design using a genetic algorithm, *IEEE Transactions on Energy Conservation*, 10(4), pp. 752-759.
- Ghosh S.C., Sinha B.P. and Das, N. (2003). Channel assignment using genetic algorithm based on geometric symmetry, *IEEE Transactions on Vehicular Technology*, 52(4), pp. 860-875.
- Altshuler E.E. (2000). Design of a vehicular antenna for GPS/Iridium using a genetic algorithm, *IEEE Transactions on Antennas and propagation*, 48 (6), pp. 968-972.
- Krohling R.A. and Rey J.P. (2001). Design of optimal disturbance rejection PID controllers using genetic algorithms, *IEEE Transactions on Evolutionary Computation*, 5(1), pp. 78-82.
- Jones E.A. and Joines W.T. (1997). Design of Yagi-Uda antennas using genetic algorithms. *IEEE Transactions on Antennas and Propagation*, 45(9), pp. 1386-1392.
- Raymer M.L., Punch W.F., Goodman E.D., Kuhn L.A., and Jain, A.K. (2000). Dimensionality reduction using genetic algorithms, *IEEE Transactions on Evolutionary Computation*, 4(2), pp. 164-171.
- Jensen M.T. (2003). Generating robust and flexible job shop schedules using genetic algorithms, *IEEE Transactions on Evolutionary Computation*, 7(3), pp. 275-288.
- Po-Hung Chen and Hong-Chan Chang. (1996). Genetic aided scheduling of hydraulically coupled plants in hydro-thermal coordination, *IEEE Transactions on Power Systems*, 11(2), pp. 975-981.
- Suckley D. (1991). Genetic algorithm in the design of FIR filters, *IEE Proceedings Circuits, Devices and Systems*, 138(2), pp. 234-238.
- Tang K.S., Man K.F., Kwong S. and He Q. (1996). Genetic algorithms and their applications, *IEEE Signal Processing Magazine*, 13(6), pp. 22-37.
- Hong Y. Y. and Li C. (2002). Genetic Algorithms Based Economic Dispatch for Cogeneration Units Considering Multiplant Multibuyer Wheeling, *IEEE Power Engineering Review*, 22(1), pp. 66-69.
- Gen M., Runwei Cheng and Dingwei Wang. (1997). Genetic algorithms for solving shortest path problems, *IEEE Transactions on Evolutionary Computation*, pp. 401-406.
- Kristinsson K. and Dumont G.A. (1992). System identification and control using genetic algorithms, *IEEE Transactions on Systems, Man and Cybernetics*, 22(5), pp. 1033-1046.
- Ozpineci B., Tolbert L.M. and Chiasson J.N. (2004). Harmonic optimization of multilevel converters using genetic algorithms, *IEEE 35th Annual Power Electronics Specialists Conference*, vol.5, pp. 3911-3916.
- Do Bomfim A.L.B., Taranto G.N. and Falcao D.M. (2000). Simultaneous tuning of power system damping controllers using genetic algorithms, *IEEE Transactions on Power Systems*, 15(1), pp. 163-169.

Table 1. Genetic Algorithms Parameters

Genetic Parameters	Magnitude
Number of Generations	150
Population size	100
Crossover size	0.5
Type of crossover	Single point
Mutation rate	0.1
Selection type	Roulette Wheel

Table 1 shows Various Genetic Algorithm parameters taken up for the optimization of process controllers.

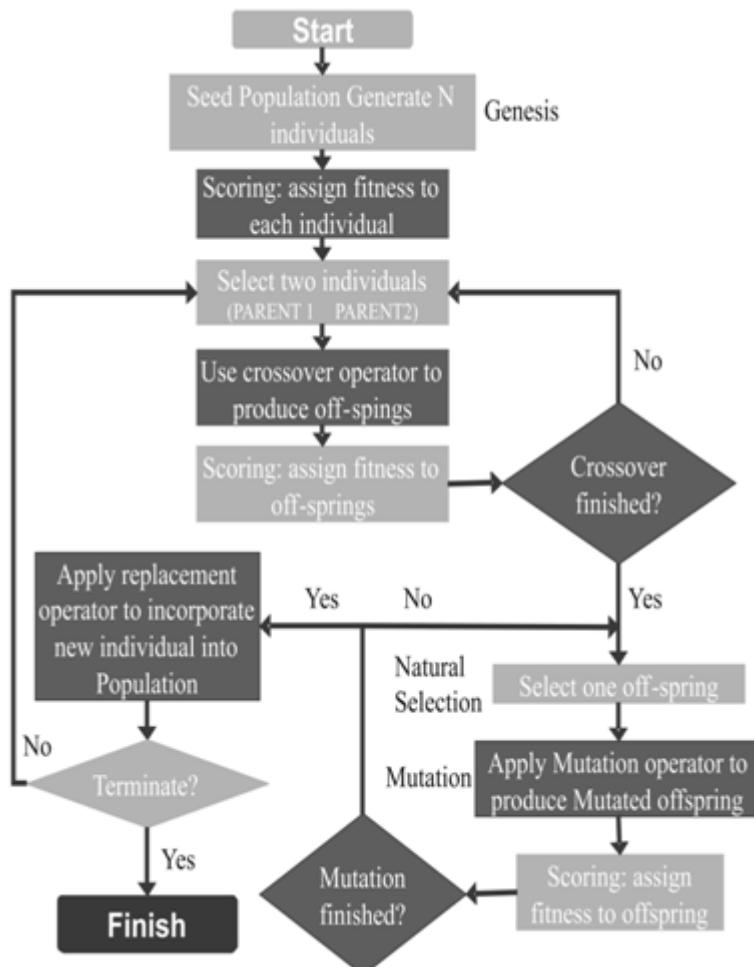


Figure 1. Block diagram representation of genetic algorithms (GAs).

Figure 1 shows block schematic of various stages to perform genetic algorithms (GAs) optimization

Chromosome 1 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1

Chromosome 2 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1

Figure 2(a). Binary encoding

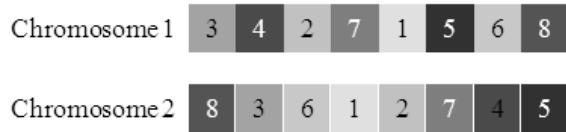


Figure 2 (b). Permutation encoding

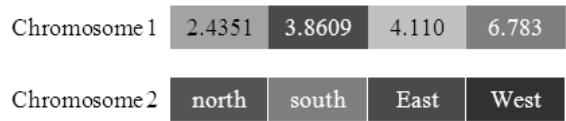


Figure 2 (c). Value encoding

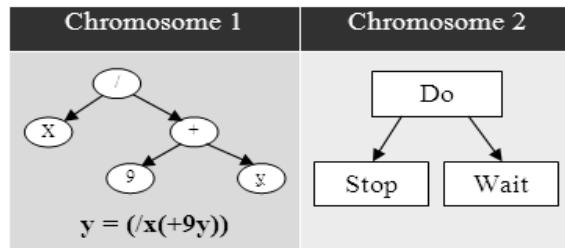


Figure 2(d). Tree Encoding

Figures 2 (a, b, c and d) show the different encoding techniques in Genetic Algorithms (GAs).

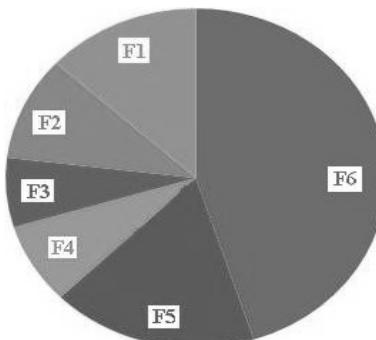


Figure 3 (a). Roulette wheel method

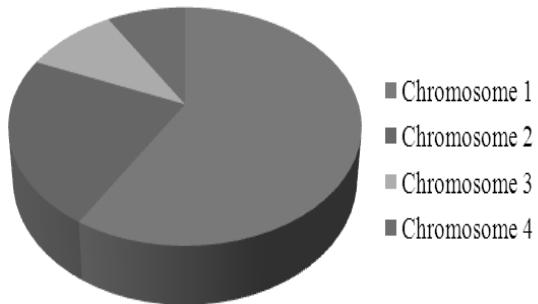
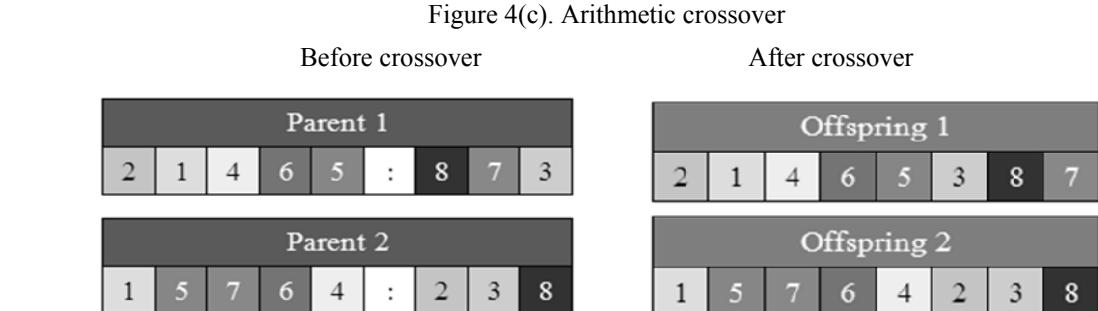
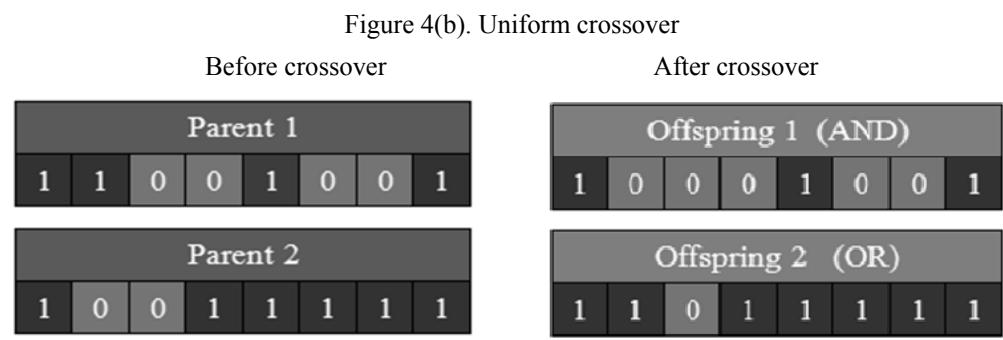
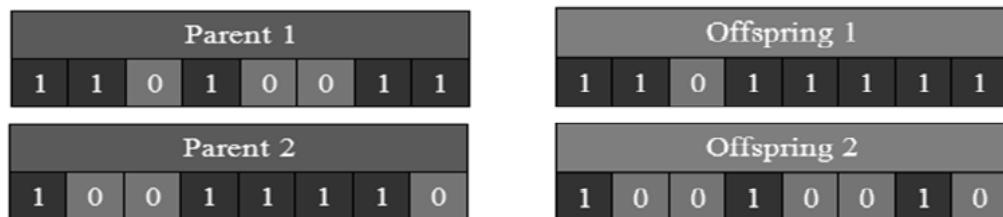
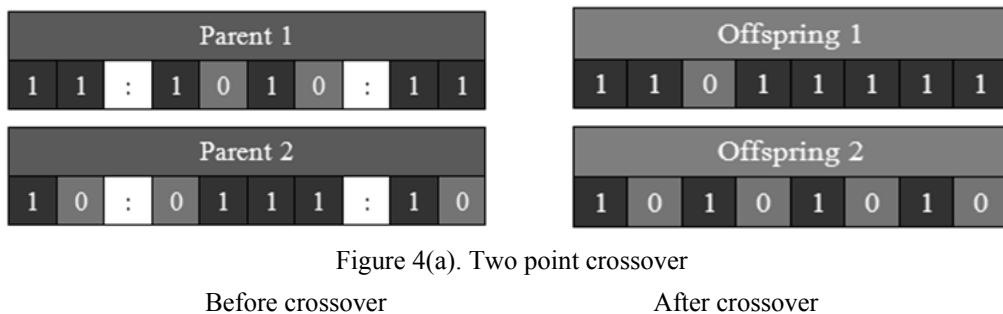


Figure 3(b). Rank selection methods

Figures 3 (a and b) show the selection techniques in Genetic Algorithms (GAs).

Before crossover

After crossover



: Denotes crossover point

Figure 4 (d). Permutation encoding crossover

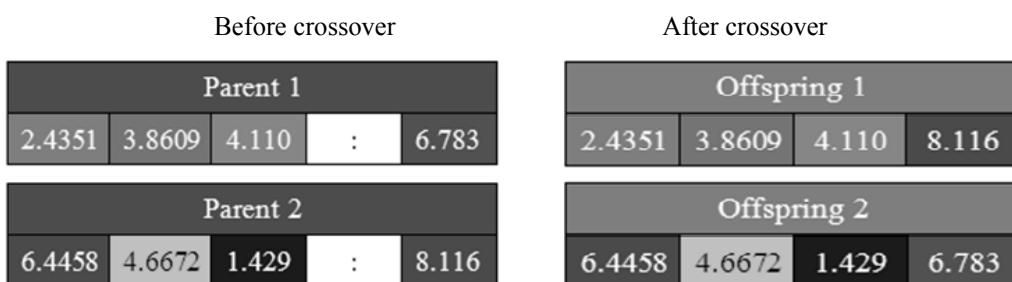


Figure 4 (e). Value encoding crossover

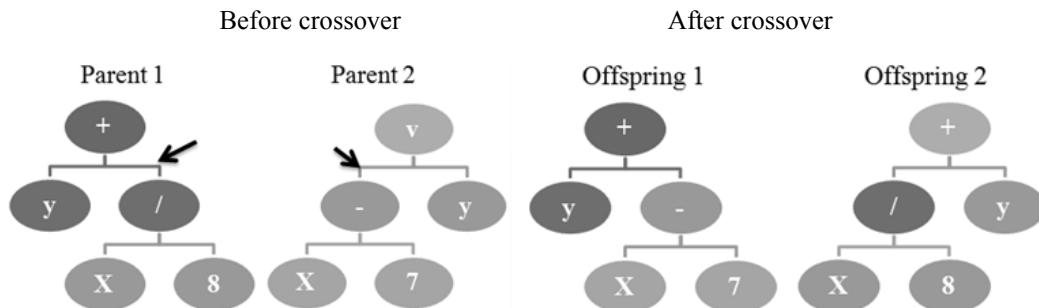


Figure 4 (f). Tree encoding crossover

Figures 4 (a, b, c, d, e and f) show the different techniques of crossover in Genetic Algorithms (GAs).

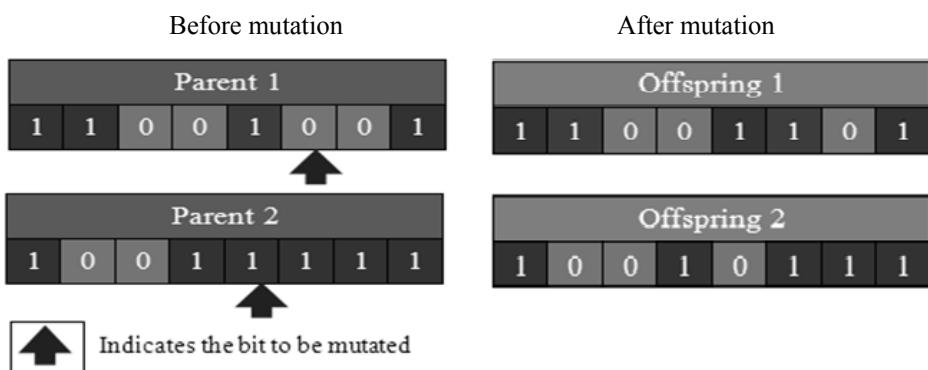


Figure 5 (a). Binary encoding mutation

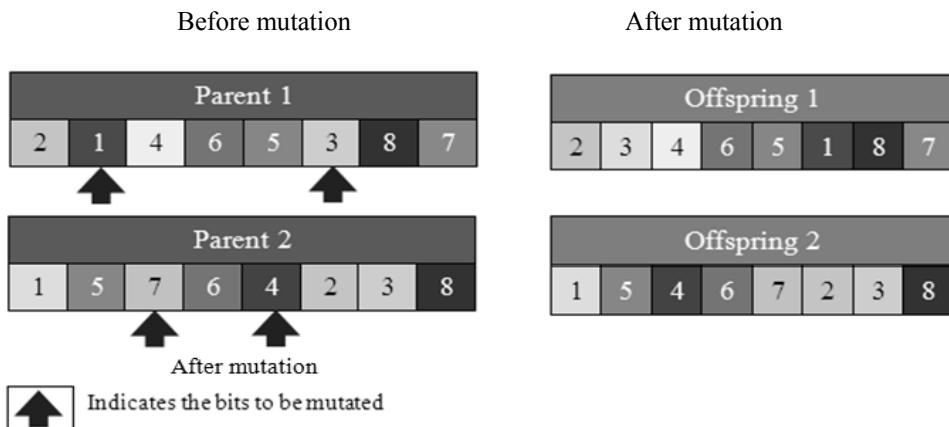


Figure 5 (b). Permutation encoding mutation

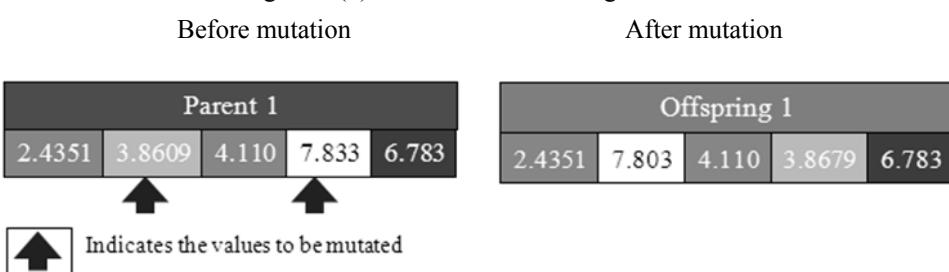


Figure 5 (c). Value encoding mutation

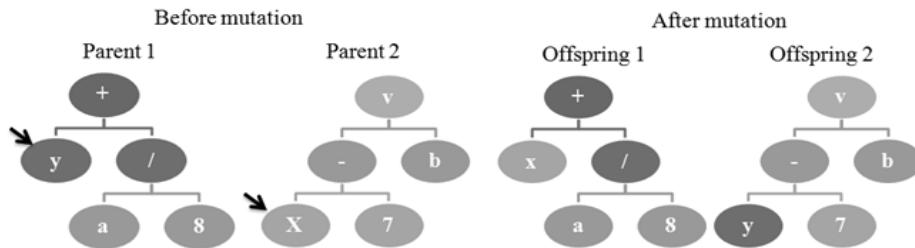


Figure 5 (d). Tree encoding mutation

Figures 5 (a, b, c and d) show the different techniques of mutation in Genetic Algorithms (GAs).

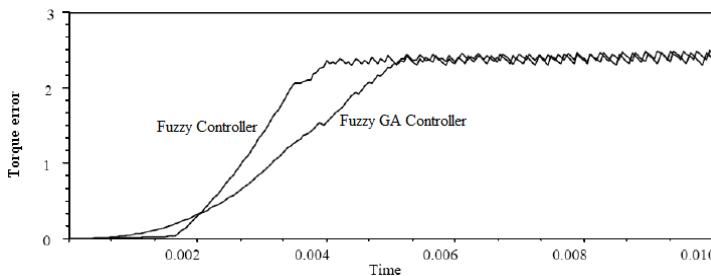


Figure 6. Comparison of electromagnetic torque error changes in fuzzy and hybrid fuzzy logic genetic algorithms controllers

Figure 6 shows the comparative reduction of electromagnetic torque error in hybrid fuzzy logic genetic algorithms (HFLGA) controller than stand alone fuzzy logic controller (FLC) in Direct Torque control strategy of Induction motor.

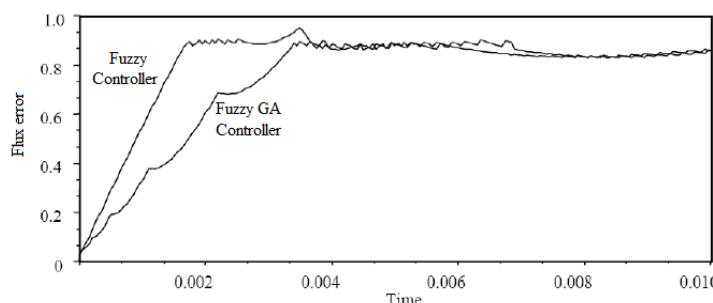


Figure 7. Comparison of flux error changes in fuzzy logic and hybrid fuzzy logic genetic algorithms controllers
Figure 7 shows the significant reduction of flux error in hybrid Fuzzy logic genetic algorithms (HFLGA) controller than stand alone fuzzy logic controller (FLC) in Direct Torque control strategy of Induction motor.

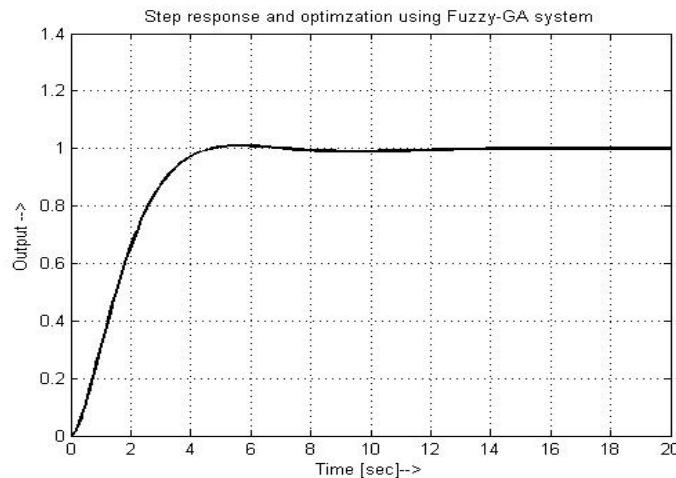


Figure 8. Step response of hybrid fuzzy genetic algorithms (HFLGA) system for turbine compressor system

Figure 8 shows flow control of turbine compressor system using hybrid fuzzy genetic algorithms (HFLGA). The response curve shows almost negligible overshoot of about 1%.

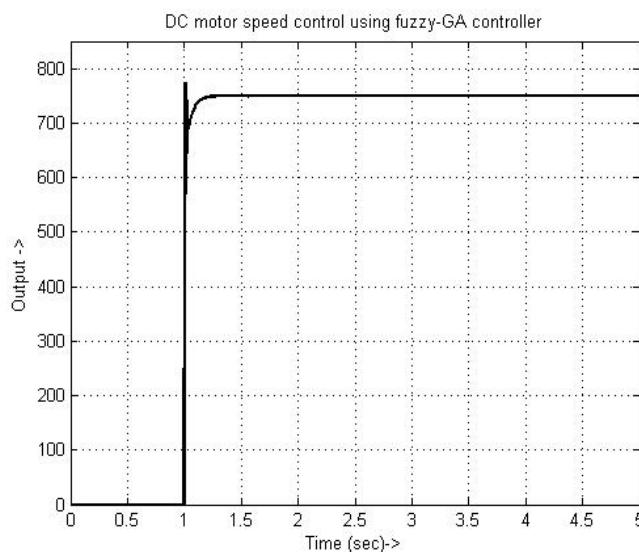


Figure 9. Step response of DC motor using hybrid fuzzy genetic algorithms

Figure 9 shows the step response of hybrid fuzzy logic genetic algorithm (HFLGA) system which is better optimized than fuzzy logic controller (FLC) and proportional integral (PI) controller.