Evolutionary Algorithms in Stabilization of Inverted Pendulum

S.Suganthi Amudhan, Dwivedi Vedvyas J, Bhavin Sedani

Abstract— A novel optimization algorithm of fuzzy logic controller (FLC) using genetic algorithms is used to characterize the major design parameters of an FLC known as characteristic parameters. The characteristic parameters simplify the design of FLC which are encoded into a chromosome as an integer string. These are optimized by maximizing the evaluated fitness through genetic operations to achieve the optimized FLC .An effective Genetic Algorithm (GA) is proposed using linkage learning, or building block identification. The genes arranged to have a fitness enhancement is the essence of linkage learning. A perfect and faster extended GA is suggested using an effective method to learn distributions and then by linking them. Stabilization of Inverted pendulum pole angle is taken as test bench.

Keywords : Genetic Algorithm (GA), Extended Compact Genetic Algorithm (eCGA), Marginal Product Model (MPM), ,Fuzzy Logic Controller(FLC).

I. INTRODUCTION

Genetic Algorithms

Genetic Algorithm (GA) is a search technique which imitates the method of natural selection and genetics. They utilize operations like reproduction, crossover and mutation to optimize a given fitness value. The parameters to be optimized are coded as a string known as chromosome. They can take real values or binary values, wherein finding a suitable way to encode the parameters of the problem is the main task. A set of prospective solutions is created which is called a population. The parameter values of the chromosomes are thus decoded to see if the optimization is done when applied to the problem. Fitness can be defined as the score achieved by performing the task.

Selection process involves selection of best individuals for the next generation using the probability of selection with the fitness calculated for each individual in the previous step. These selected individuals are then subjected to crossover process. New individuals are generated by mixing of two strings which have acquired characteristics from different successful strings. Lastly, the next operation of mutation is done with low probability in order to change the bits of the chromosome in a random manner. This assures that probability of searching is never zero. There are different types of operations applied with varied probabilities [1].

Scalable Genetic Algorithms

GA's have to follow a systematic approach to solve very difficult problems. The development of competent genetic algorithms that solve hard problems in a scalable way is proposed in last decades. This greatly reduces the problem of good coding and deciding a good genetic operation in various applications by the programmer. The user need not have to adjust coding and operators, if GA itself can do the same to the problem [2].

Decomposition in GA Design

Selectorecombinative GA's generally emphasizes on cross-fertilization and paves way for competent GA.By using building block (BB), a competent selectorecombinative GA is designed by Goldberg. These building blocks identification & recombining them for good performance is the main idea behind these GA's.

BB hard problems-It is very difficult to obtain BB's for complex problems as different BB's are hard to separate and some low order BB's may give wrong solutions.

BB growth and timing- BB's should allow the best ones to grow & the growth rate should be a steady one. Proper selection of the crossover probability p_c and selection pressure s which is the measure for selection leads to achieving the same.

$$p_c \leq 1\text{-}s^{-1}/\varepsilon \quad (1)$$

Three main approaches have been used in understanding time:

- The dynamics of the best individual is utilized in takeover time models.
- Selection-intensity models uses the dynamics of the average fitness of the population.
- The dynamics of average and higher-order simulants are taken in Higher-order models.

All BB's are of equal importance with the convergence time specified by

$$t\,c = \prod / 2I \sqrt{l} \quad (2)$$

where l gives the number of binary variables which determines the size of the problem. I is the selection intensity which depends on selection pressure for tournament type selection.

$$I = \sqrt{(2 (\log(s) - \log (\sqrt{4.14 \log (s)})))} \quad (3)$$

If the BB's are scaled exponentially, then the convergence time of a GA varies linearly with the size of the problem.

$$t_c = -\log 2/\log(1-I/\sqrt{3})l \quad (4)$$

BB supply and decision making

Adequate supply of the raw building blocks is to be done by the randomly generated populations. The population has increasing size, higher probability and more complex BBs. There is atleast one copy of all the raw building blocks given by

$$n = \chi k \log m + k \chi k \log \chi \quad (5)$$

Revised Manuscript Received on March 15, 2020.

***** Correspondence Author

S.Suganthi Amudhan*, Assistant Professor, Electronics and Communication Engineering, Babaria Institute of Technology, Varnama, Gujarat, India.

Dr. Dwivedi Vedvyas J, Pro Vice Chancellor, C U Shah University, Wadhwan City.Surendranagar Gujarat, India.

Dr.Bhavin Sedani, Professor, Electronics and Communication Engineering, L.D.College of Engineering, Ahmedabad, Gujarat, India.

where k alphabets of cardinality χ, the population size for a problem of m BB's.

Competing BB's is statistical and so the probability of making best possible outcomes is done by increasing the population size.

$$n = -\sqrt{\pi}/2\ \sigma BB\ /d\ 2k\ \sqrt{m}\ \log \alpha \quad (6)$$

where α is the probability of deciding incorrectly.
Population sizing consists of
• 2^k building blocks define the competition complexity.
• Same way, number of building blocks define subcomponent complexity.
• Decision making depends on d/ σBB, the signal-to-noise ratio.
• The coefficient −log α is the measure of probabilistic safety factor.
The population size is represented as

$$n = -co\ \sigma BB/d\ 2k\ m\ \log \alpha \quad (7)$$

if the building blocks are scaled exponentially where, c_o is a constant reliant on the drift affects.
Identification of BB's & exchange of the same pave way for the success in innovation. Achieving competence was the main challenge to advocate the BB exchange in first generation .Fixed operators like uniform crossover, have polynomial scaling on simple problems and exponential scaling with problem size for tough problems. A control map is a function of control parameters for selection and crossover.

Competent GA Designs

Competent genetic algorithms solves hard problems swiftly, reliably, perfectly and also successfully solves increasingly difficult problems in wide range of applications. These algorithms removes the problem of good coding & choosing good genetic operator by the programmer. Adding to this, it can also have GA which itself adapts to the changing conditions of the problem in consideration rather the user to take care of it.

Hard problems are those problems which cannot be divided into simpler solutions, not properly scalable sub solutions, have lot of local minima or maxima and generally have a high stochastic noise. This involves developing an algorithm for problems with bounded difficulty and display sub quadratic scalability with the problem size.

The competent GAs have varied mechanism but their success does not. Competent GA started with the messy genetic algorithm and the fast messy GA. Different mechanism are classified for the design of a number of competent GAs based on
1. Distinguishing good and bad linkage;
2. Representation of linkage;
3. Storage of linkage information.

Based on distinguishing good linkages from bad linkages, competent GAs can be classified into the following categories: Unimetric methods are those which depends on the fitness value. No other criterion is involved like in linkage learning GAs. Multimetric methods uses other extra criteria apart from the fitness function given by the problem for understanding the quality of individuals or models. Multimetric approaches also have certain bias which does not come from the problem e.g. Gene expression messy GA (gemGA), estimation of distribution algorithms (EDAs) and design structure matrix GA are examples of multimetric methods.

Based on representation of linkage, we can categorize scalable GA designs into two categories: Physical linkage physical locations of two or more genes on the chromosome decides the linkage. Physical linkage is biological inspired directly. Examples of competent GAs that use physical linkage are the messy GA, the fast messy GA, and the linkage learning GA. Virtual linkage, where linkage information is computer science based approach for using graphs, matrices, pointers, or other data structures that control linking of decision variables. Estimation of distribution algorithms, general linkage crossover and the dependency structure matrix GA are examples of competent GAs with virtual linkage.

Based on storing the linkage information, competent GAs can be classified into two categories: Distributed Model, is one in which no centralized storage of linkage information is seen but it is in a distributed manner. Centralized Model, is one in which where linkage information is stored in centralized manner, as a global probabilistic vector or dependency table.

The structures that GA exchanges and its association with the fitness is under argument. The main challenge is the complexity of structures and GA's success depends on two conflicting possibilities.
1) GA can only efficiently deal with single genes structures.
2) GA can also process complex structures consisting of several genes, called as building blocks.

Linkage learning aims at GA that processes complex structures than single genes. This suggests that genes shall exist in association to improve the fitness, if the relation of genes are not known. This recommends developing a computationally permissible algorithm that learns linkage and exploration of a set of probabilistic algorithms.

Probabilistic Optimization Algorithms

Each gene is represented as a fraction in the current search which had initially a value of 1. After successive crossovers, new population is generated. Depending on how well certain genes performed in the competitive environment, they emulate the effect of selection. The compact GA (cGA) [3] and Population Based Incremental Learning PBIL [4] are two examples of naive and effective algorithms.

In L-dimensional probability vector P[], 0.5 for each gene position is initialized in cGA. S solutions are produced by polling this vector in this random initialization phase. The fittest gene positions of S are rewarded in opposition with each of the less fit ones. If the fittest has a value 1 in the kth position, P[K] is increased by a specific probability. If the fittest has a value 0, P[K] is likewise decreased by the same probability and the less fit solution is left undisturbed. E is the amount of increase or decrease by a value.
Considering a problem of maximizing number of 1s ,total genes as 4,two solutions and increase or decrease rate as 0.25, gene arrangement of 0111 of fitness value 3 is fitter than 1010 of a fitness of 2. The initial probability vector is randomly chosen as all four genes having equal probability of 0.5.The probability of first gene is decreased by 0.25 as 0 is part of fitter chromosome. In a similar way, the next gene is increased by 0.25 as 1 is part of fitter chromosome and in the third gene being same in

both, it is unchanged. This forms a new probability vector of 0.25, 0.75, 0.5 and 0.75 respectively for four gene positions of the chromosome as given in Fig 1. This process proceeds till cGA has converged or when all its values are either zeros or ones.

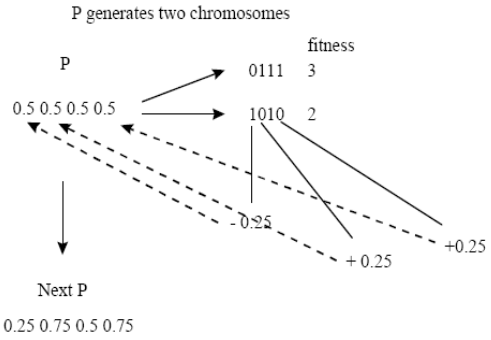

Fig 1: cGA evaluation with changes in probability distribution [6].

In cGA, the probability distribution of the set of future solutions of simple GA are given. Evolution chosen GA chromosomes and the distribution of the population recommends other good solutions. New chromosomes similar to the current population are generated using crossover techniques. This can also be attained by using cGA where direct representation of the distribution itself is done. Modification brought by selection process can be directly achieved by changing probability distribution. This probability distribution in turn leads to linkage learning.

Minimum Description Length Models

Current encoding and minimal distribution of population compressed are the deciding criteria for good distribution. This is a minimum description length prejudice on the distribution. The model size is minimized to penalize complex models. This is done because information theory says inaccurate models will not help in the compression of the population [5].

MDL Constraints on Marginal Product Models

Compression is done on the probability distribution of the space. Distribution is assessed based on the bits used to signify a given message in the population. Minimizing the combined model and population representation using distribution forms the optimization problem for the model.

The Combined Complexity Criterion

The subset of marginal product model is assumed to be of size S [I] of I^{th} partition. $2^{S[I]}$ frequency computations is the marginal distribution with N as the population size and log N is size of frequency counts. The model representation size is signified as

$$\text{Total Model Complexity} = \log N \sum I\ 2^{S[I]} \quad (8)$$

MI is the marginal distribution in this I^{th} subset. S[I] is always greater than entropy of distribution denoted as E (MI) . Thus the population is compressed, placing the I^{th} subset's genes after the I^{th} marginal. The aggregate compressed population complexity is:

$$\text{Compressed Population Complexity} = N \sum E(MI) \quad (9)$$

MPM structure gives the MPM partitioning, and not the definite probabilities of the marginal distributions. The probabilities are evaluated based on the optimal compression. MPM structure is formed by means of the population's frequency counts and then model complexity and compressed population complexity are added to get a combined complexity number.

Combined Complexity (MPM) = Model Complexity + Compressed Population Complexity

II. EXTENDED COMPACT GENETIC ALGORITHM (ECGA)

Good probability distribution leads to equivalent linkage learning in extended compact GA as proposed by [6]. Minimum description length MDL models concludes that, simpler distributions are better than more complex ones, the rest being same which is the gauge for good distribution.

MDL penalizes complex models in the optimization problem by providing a best probability distribution. MDL restriction improves the probability model and population representation to find a good distribution. Similar to those of the compact GA (cGA), MPMs are a class of probability formed as a product of marginal distributions on a partition of the genes. MPMs can signify probability distribution of more than one gene while in cGA, probability distribution of one gene can only be done. Direct linkage map with partition is done by MPMs to enable separation of tightly connected genes.

A MPM for a four-bit problem represents the 1st and 4th as linked genes and 2nd and 3rd as not linked genes.

Table I
Marginal probability model of four genes [6]

[0 , 3]		[1]		[2]	
00	0.5	0	0.5	0	0.6
01	0				
10	0	1	0.5	1	0.4
11	0.5				

Table I shows one possible linking of a four dimensional problem using MPM. The probability distribution of [0, 3] has a positive impact for the combination of 00 and 11.This combinational distribution is more influential than that of cGA.

4N bits of storage are actually required in the problem. The representation can be 0 for 0000 and 1 for 1111. This uses only one bit per structure, and hence only N bits for the whole population. Knowing these four bits are correlated, and representing their distribution in an MPM, there is a cut down in the storage space. The amount of space needed for storing the population is reduced to one fourth, when these four bits are found to be correlated and distributed in a MPM accordingly. The average number of bits for representation is characterized as E(P),the entropy of distribution. This is used for compression of the population by adjusting MPM's probabilities.

eCGA steps are as follows:

1. Initialization: Random individuals are initialized for the evaluation of the fitness values.
2. Selection: s-wise tournament selection is used [7]. Other techniques can also be tried.
3. Building the probabilistic model: In order to find optimal model, the structure and parameters are searched for using a greedy search heuristic.
4. Creation of new individuals: The

probabilistic model created is sampled to acquire new individuals.
5. The parent is replaced with offspring population.
6. The steps 2–5 are repeated until the convergence criteria are met.

An optimal or near-optimal probabilistic model is found using the greedy search heuristic that involves

1. Each variable has a variety of probabilities and so independent assumption has to be made.
2. Evaluation of the model and population complexity is done.
3. The potential 1/2l (l-1) merges of variables analyzed.
4. Then the model and compressed population complexity values are evaluated.
5. Then the merged model with lowest combined complexity is selected.
6. If the combined complexity of the best merged model is better than the combined complexity of the model, then the best merged model is replaced and step 2 is repeated.
7. There is no enhancement of model is possible, if the combined complexity of the best merged model is less than or equal to the combined complexity and we can conclude that the model evaluated in step 2 will be the probabilistic model of the current generation.

New population is formed of size n (1 − Pc), where Pc is the crossover probability that gives best individuals in the current population. nPc individuals are randomly chosen from the present individuals as shown in Fig 2.

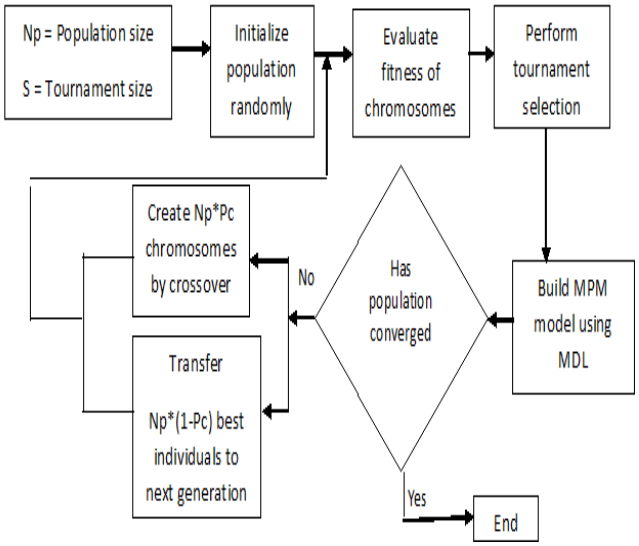

Fig 2: Flow Chart for eCGA [6].

III. INVERTED PENDULUM

In this paper inverted pendulum is used as a test bench problem. Using Simulink model of the system, the fitness function is evaluated to apply genetic algorithms for optimization of characteristic parameters. Then extended compact genetic algorithm is applied to study the linkage learning to arrive at the best fitness value for the application.

The design is implemented in MATLAB using functions like create_mfs.m to form the input and output membership functions and create_rules.m which yields the rule-base. Another function make_fis.m uses these two functions together to create the Fuzzy Inference System (FIS). Only triangular membership functions can be created.

create_mfs.m contains the number of membership functions and the spacing parameter . These decide the centers of the membership function. The centers of adjacent membership functions can be easily calculated as the base vertices are at the same positions in triangular membership functions, where only vertex coordinates are required These parameters are returned by the function.

In *create_rules.m* returns rule-base using the number of membership functions per variable, the spacing parameters for each variable and the characteristic angles for the seed line. The seed points are found and then the grid point co-ordinates. Measuring the distance to each seed point and finding the shortest one for each grid-point, gives the consequents. These antecedents and consequents forms the rule matrix in Fuzzy Logic Toolbox [8]. This leads to formation of Mamdani based FIS using only a few parameters. An optimal FLC can be found using Genetic Algorithm with these ideal parameters.

Evaluation functions

Evaluation of different designs are required to implement Genetic Algorithm to a Fuzzy Logic Controller. The GA should be able to perform the evaluation quickly as the GA has to process using various combinations of parameters.

The fitness function of a set of parameters are called Evaluation Functions [9]. A value is returned based on how well they optimize by passing the parameters to the evaluation function.

The error in the pole angle is continuously measured in the Simulink model formed and a record of the same is maintained. The time weight is multiplied with square of the error and the summation of this is inverted to have ITSE as the fitness value.

The simulation stops immediately if the pole angle reach 90°.Modelling controllers that will fail to balance the pole is not done. Here the fitness is calculated as the time taken before the pole failed to stabilize multiplied by $1x10^{9}$. This is a reward to keep the pole from failing a little bit longer.

In the SIMULINK model used in Fig 3, the desired angle of the pole is zero degrees. The error in the angle e and the change of error de are scaled by the suitable gains. They are trimmed to lie in the range of –1 to 1. These are the inputs to FLC and then the output is scaled by another gain. The force is the main input to the system and the outputs of model are the pendulum angle, angular speed, cart position and cart velocity.

The Simulink Model provides variables of time and error in angle of the pole after application of FLC and the same are utilized in the optimization using GA and linkage learning using eCGA model.

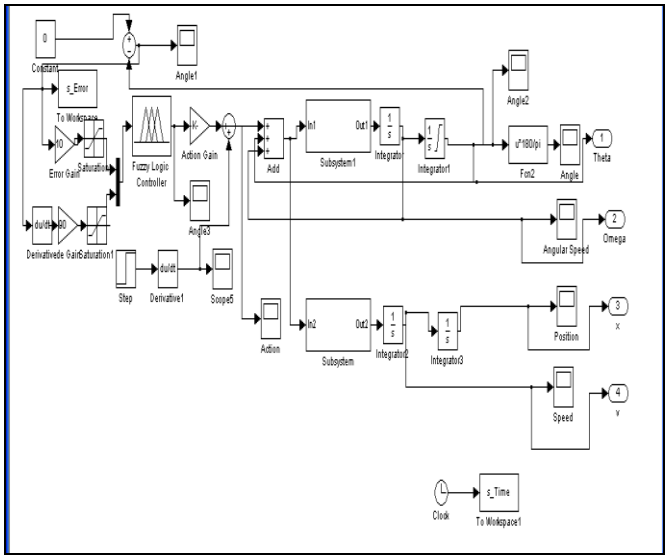

Fig 3: Simulink model FLC

IV. RESULTS

The simulation results arrived using SIMULINK are indicated in the figures(Fig 4 to Fig 9), which depicts the angle of the pole using standard manually-tuned FLC (Fig 4,Fig 6 & Fig 8) and also the pendulum angle using genetically tuned FLC (Fig 5,Fig 7 & Fig 9) for each set of impulse inputs.

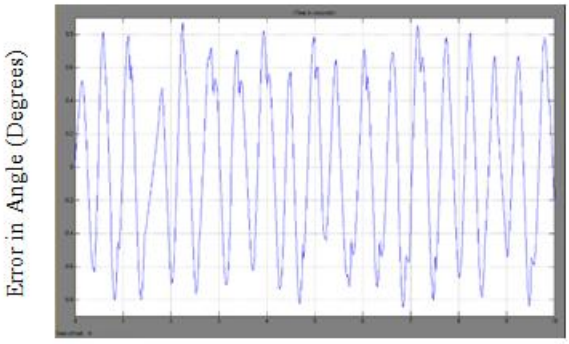

Fig 4: Performance with Impulse Input = 1 of a FLC without GA

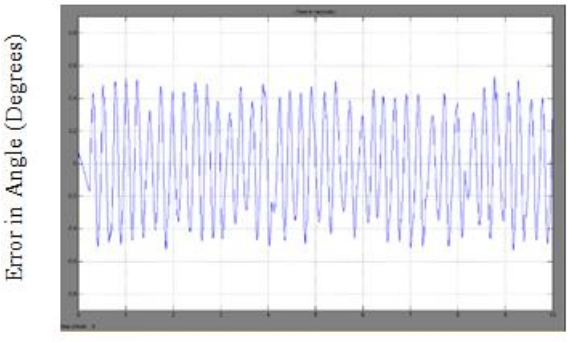

Fig 5: Performance with Impulse Input = 1 of a FLC with GA; No. Generations=15

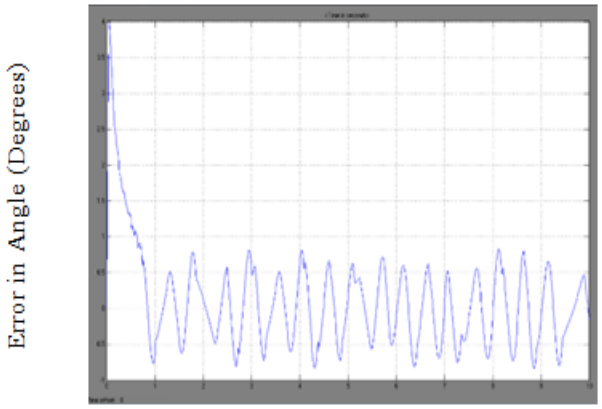

Fig 6: Performance with Impulse Input = 10 of a FLC without GA

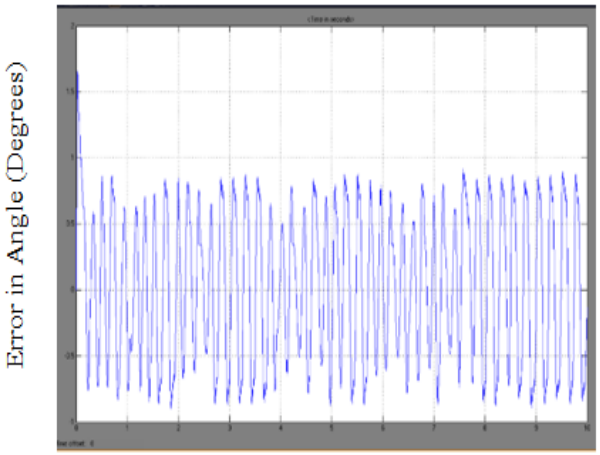

Fig 7: Performance with Impulse Input = 10 of a FLC with GA; No. Generations=15

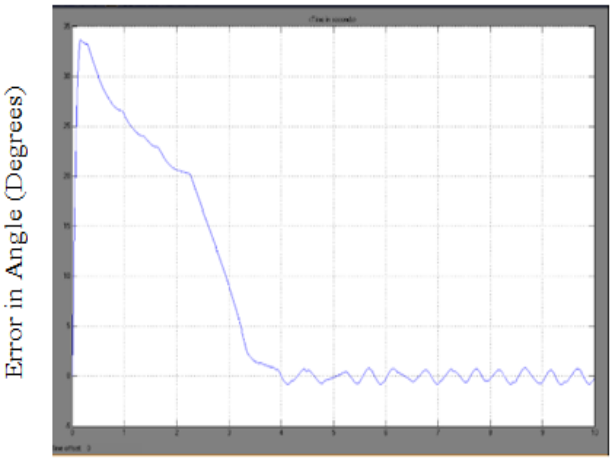

Fig 8: Performance with Impulse Input = 30 of a FLC without GA

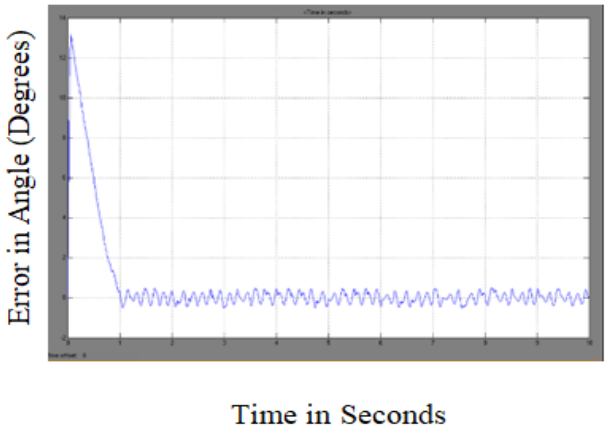

Fig 9: Performance with Impulse Input = 30 of a FLC with GA; No. Generations=15

The comparative results obtained by using FLC with and without GA are shown in Table II.

Table II
Comparative results using FLC without & with GA

S.No	Force applied (N)	Angle of the pole (deg)		Time taken to stabilize(sec)	
		Using FLC without GA	Using FLC with GA	Using FLC without GA	Using FLC with GA
1	1	0.85	0.41	1	1
2	10	4	1.8	1.5	1.2
3	30	33	13	5	5

The linkage structure generated using eCGA with maximum fitness.

GENERATION #1:
eCGA model:
{ [3 14] [13 15] [5 6 7 8] [9 10 11 12] [1 2 4] [17 18 19 20] [16] }
Maximum Fitness: 4.333333

GENERATION #2:
eCGA model: { [13 14 15 16] [1 2 3 4] [17 18 19 20] [5 6 7 8] [9 10 11 12] }
Maximum Fitness: 4.666667

GENERATION #3:
eCGA model: { [17 18 19 20] [1 2 3 4] [9 10 11 12] [5 6 7 8] [13 14 15 16] }
Maximum Fitness: 5.000000

V. CONCLUSION

It is very clear that as the force (impulse input) is increased the disturbance to the cart is more and hence angle of the pole as well as time taken for the pole to stabilize increases. When the force applied is beyond 60N it is very difficult to stabilize even if we increase the number of generations.

It can be seen that the fluctuations of the pole angle about the desired position are less with genetically tuned FLC, as compared with the manually tuned one. The fluctuations are also more smoothened for the genetically tuned FLC, which indicate improved relative stability.

These results are encouraging because using characteristic parameters is anyway a suboptimal approach, and we have only shown the result of an isolated random runs of genetic tuning of the FLC. With better reproduction and crossover methods, results can be much improved.

In the case of linkage learning, we can infer that structure changes in each generation and attempts to achieve a perfect linkage. In this problem of 20 bits, MPM structures that optimally compress the population using eCGA has been generated. This also concludes that concentrating on learning MPM, eCGA can solve difficult problems faster than simple GAs not using linkage information.

REFERENCES

1. D.E.Goldberg, Genetic algorithms in search, optimization,and machine learning, Addison-Wesley, 1989.
2. Sastry, Kumara, and Erick Cantú-Paz. Scalable optimization via probabilistic modeling. Edited by Martin Pelikan. Springer-Verlag Berlin Heidelberg, 2006.
3. Harik, G., Lobo, F. and Goldberg, D.E., The Compact Genetic Algorithm. Proceedings of the 1998 IEEE Conference on Evolutionary Computation, pp. 523-528, 1998.
4. Baluja, Shumeet, Population-based incremental learning, a method for integrating genetic search based function optimization and competitive learning. No. CMU-CS-94-163. Carnegie-Mellon Univ Pittsburgh Pa Dept Of Computer Science, 1994.
5. Goldberg, David E., The design of innovation: Lessons from and for competent genetic algorithms. Vol. 7. Springer Science & Business Media, 2013.
6. Harik, G., Linkage learning via probabilistic modeling in the ECGA (IlliGAL Report No. 99010). Urbana, IL: University of Illinois at Urbana-Champaign, January 1999.
7. Sastry, K., & Goldberg, D. E., Modeling tournament selection with replacement using apparent added noise. Intelligent Engineering Systems through Artificial Neural Networks, 11, 129–134, (Also IlliGAL Report No. 2001014) 2001.
8. Park, Y. J., Cho, H.S., Cha, D.H., "Genetic Algorithm-Based Optimization of FLC Using Characteristic Parameters", Proceedings of the 1995 IEEE International Conference on Evolutionary Computation, pp 831-836, 1995.
9. Bäck, Thomas, David B. Fogel, and Zbigniew Michalewicz, eds. Evolutionary computation 1: Basic algorithms and operators. CRC press, 2018.

AUTHORS PROFILE

Prof. S. Suganthi Amudhan is working as an Assistant Professor in Electronics and Communication Department at Babaria Institute of Technology, Vadodara. She has a teaching experience of 15 years. She is currently a research scholar of C.U Shah University, Surendranagar. She has received best paper presentation award once.

Dr. Dwivedi Vedvyas J. is working as Pro Vice Chancellor at C U Shah University, Wadhwan City, Surendranagar Gujarat, India. He is a Professor in Electronics and Communication Engineering at C.U Shah University and has total 23 years of experience including 5 years of industrial and 18 years of teaching experience. He has published/authored/co-authored several books, research/review articles/papers in refereed international/national journals/conference proceedings, successfully supervised 6 Ph.D. thesis (awarded), examined and reviewed many Indian Ph.D. thesis. He also has Indian patents; completed research and industry consultancy based projects, delivered many expert/keynotes talks.

Dr. Bhavin Sedani is working as a Professor in Electronics and Communication Department at L.D. College of Engineering (Government Engineering College),Ahmedabad. He has teaching experience of 17 years. He has presented more than 46 research papers in various international and national conferences. His 28 research papers are published in various international journals and Scopus indexed IEEE Xplore digital library. He has achieved best paper presentation awards 7 times for his research articles.