International Journal of Innovation in Science and Mathematics
Volume 1, Issue 1, ISSN (Online): 2347–9051

Robot's Controller Design with Genetic Alghorithm

Alireza Rezaee
Assistant Professor of Department of Mechatronics Engineering,
Faculty of New Sciences and Technologies, University of Tehran, Tehran, Iran
Address: Amir Abad, Tehran, Iran. Email: arrezaee@ut.ac.ir

***Abstract* – In this project we use the ILC control method to manipulate the robotic arms of a robot with two degrees of freedom. First we implement the dynamic equations of robot according to the Schillings book of robotic. The aforementioned implementation was done in MATLAB SIMULINK environment. The Genetic Algorithm was used for tuning the coefficients of PD Controllers (proportional and derivative gains). Also we use Multiobjective genetic Algorithm to attain the coefficients of ILC PD Controllers.**

***Keywords* – Algorithm, Controller, Genetic, Robotic, Tuning.**

I. Introduction

We begin with a brief introduction about ILC control method in section 2. After that we will discuss about our simulated [1,2] model in MATLAB ,and then the using of MATLAB Genetic toolbox is described (two using toolbox at sections ΠI to V). Also we have attached some pictures of our MATLAB model diagram and results that have been achieved from running the model. The results are as some graphs, achieving from simulated model.

II. ILC Control Method

ILC method is a new approach in control theory. This method is a technique for machines, equipments, processes and systems which iterate one particular motion or action. Essentially, ILC is a short brief of the Iterative learning control phrase, and ILC controllers are added to compensate the shortage of the other controllers in the simulated model. We use ILC controller and tune its parameters to learn during a sequence of iterations to correct the error or behavior of robot manipulator, for following a particular trajectory. There are some problems with systems that use ILC [1], [3], [4]. Those are classified at two general groups: stability, performance.

The problem of system performance is to lead an out put response of a dynamic system to a particular path, with the minimum of predefined error. Stability means the declining of error continue with the increasing of iterations. We don't get a good stability, if a good fitness is not defined. In the other hand the parameters that the GA yields, cause the model reach to a 0.057 amount of error from initial value of 7.684 error value in 10 iteration, but if we continue the running of model the amount of error don't stay constant with out oscillation. . Consequently for overcoming the problem of stability we ran the model 10 times to evaluate the fitness of each chromosome.The bellow diagrams attached according to the [1], [3], [4] references.

Introduction to MOEA Multi-objective Evolutionary Algorithm (MOEA) is a stochastic search technique inspired by the principles of natural selection and natural genetics. It has attracted significant attention from researchers and technologists in various fields because of its ability to search for a set of Pareto optimal solutions for multi-objective optimization. The MOEA first begins with a population of possible solutions, called strings. Each string is fed into a model. This model is usually a computer program representation of the problem. The model returns the answer in the form of a cost function. Based on these cost functions, strings are selected for evolution to create the next generation of strings.

Multi-objective simply means that there are more than one objective involved. For each string, each objective returns a separate cost. The manner in which a string is deemed superior or inferior to other strings shall be discussed elsewhere. The selected strings undergo the evolutionary process where the traits of the selected strings (which may or may not be good) are selected and combined to form new strings for the next generation. In theory, with each generation, the strings in the population should return better and better cost functions. In practice, there is a limit to the cost functions that strings can achieve, depending on the objective functions and the limits imposed on the model parameters. Further, there is the possibility that the best traits are not found [5-13]. We use a toolbox with MOEA name that is written in National University of Singapore.

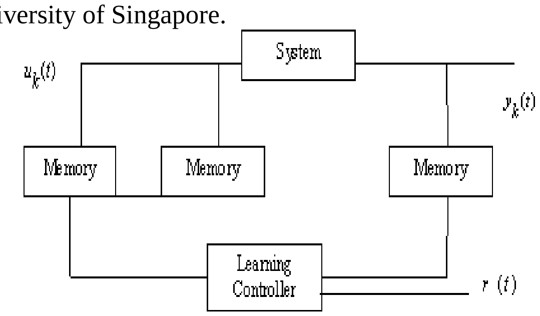

Fig.1. A general block diagram of ILC.

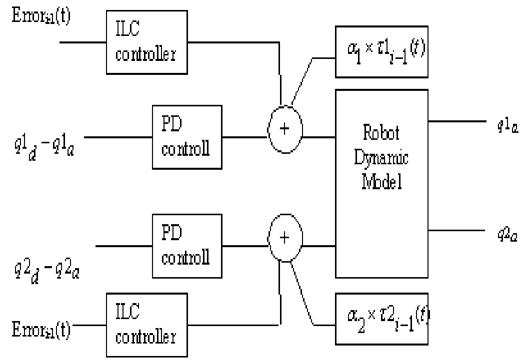

Fig.2. our Model in Simulink Environment of Matlab

International Journal of Innovation in Science and Mathematics
Volume 1, Issue 1, ISSN (Online): 2347–9051

III. Parameters and Objectives

We choose the proportional and derivative coefficients (P, D) of ILC controllers and α_1, α_2 such as genes in one chromosome. These aforementioned genes form the Genetic parameters that were adjusted according to the some objectives. Thus we have six gene or parameters which participate at forming a chromosome and two objectives for one of such chromosome.

P1	D1	P2	D2	α_1	α_2

Fig.3. structure of one chromosome

In this implementation seven (7) bits was assigned for each gene. Two objectives were defined F1 and F2, that we discuss them bellow, before the definition of F1 and F2 are described.

$$F1 = Error1 + Error2$$
$$F2 = (Error1 > Errorp1) + (Error2 > Errorp2) \qquad (1)$$

Error1 means the total error in this iteration for the first arm joint. Errorp1 means the total error of the previous iteration for the first arm. And Error2, Errorp2 are such as predefined parameters for the second degree of freedom.
F1 means that the total error of the first and second joints must be minimized.
F2 consists of two inequalities:

The first inequality means if the total error of this iteration for arm #1 is greater than the total error of the previous iteration then the result of it becomes zero (0). If (Error1<=Errorp1) then the outcome of inequality is one (1). The second inequality has the same description.

Therefore the minimum of F2 occurs when that (Error1<=Errorp1) and (Error2<=Errorp2) that the amount of F2 is zero.

Maximum value of F2 occurs when, the both of above inequality occurs simultaneously. For each chromosome the model was ran 10 times thus F2_total equals the sum of above F2's and F1_total equals the sum of F1's.

IV. GAOT (Another Matlab Toolbox)

It is a genetic toolbox in MATLAB which is used to optimize complex nonlinear functions. Generally Genetic algorithms are good tools for searching in the space of problem solutions. This toolbox is used to optimize one fitness function and we can not use it, to optimize further than one fitness function. But, we can combine some fitness and collect them to one well defined fitness function and then use it for optimizing. In this way, we can obtain the aim of implementing multi objective problems. The combination of these fitness functions can be defined as a weighting average of goals that must be optimized.
We use GAOT to optimize the parameters but in this method, our parameters set were (P,D) gains of controllers. α1 and α2, is assumed to be fixed with 0.95 and 0.9 values.

The values for this parameters is obtained manually with trail and error.

Also empirical experiments exhibit that amount of these parameters must be close to 1 but not exactly 1.

The outcome of GAOT for tuning ILC parameters is as well as MOEA, but they seem to be memories with longer time of remember than the results of MOEA. Because they give us average of error less than those of MOEA in iteration numbers further than 10.

We compare the result of MOEA and those of GAOT, both of them be well for 10 iteration, but GAOT outcomes, give better average of error than MOEA. Because of applying a constraint such as Error (k+1)<=Error(k) to objectives is not possible in MOEA, it is possible to impose it to goals in GAOT. The fitness that, we use in this toolbox is:

$$Val = (1/(Error1 + Error2))*(1/count) + (1/Error) \qquad (2)$$

V. The Results

The bellow figures illustrates the total error of the 1st and 2nd joints versus number of iterations:

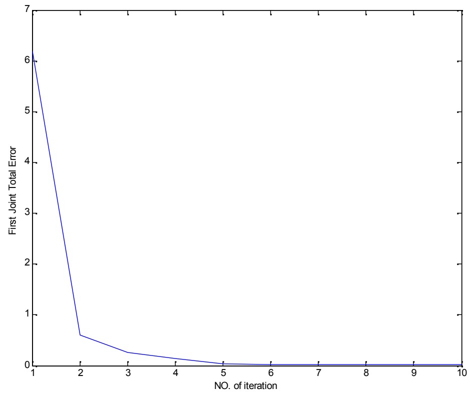

Fig.4. total error of the 1st joint

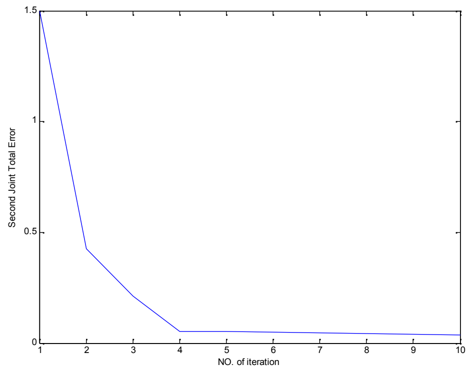

Fig.5. Total error of the 2st joint

The bellow figures indicate desired value of 1st and 2nd joint angle and the actual value of it:

International Journal of Innovation in Science and Mathematics
Volume 1, Issue 1, ISSN (Online): 2347–9051

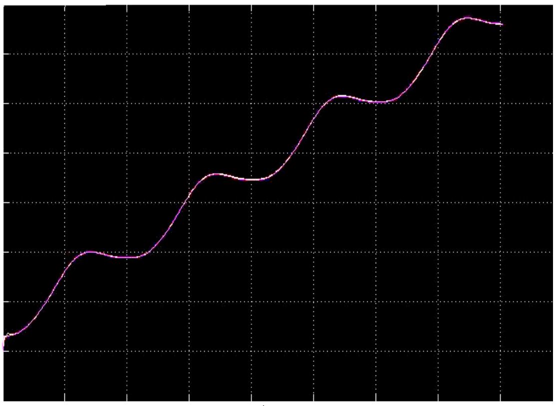

Fig.6. 1st jont angle

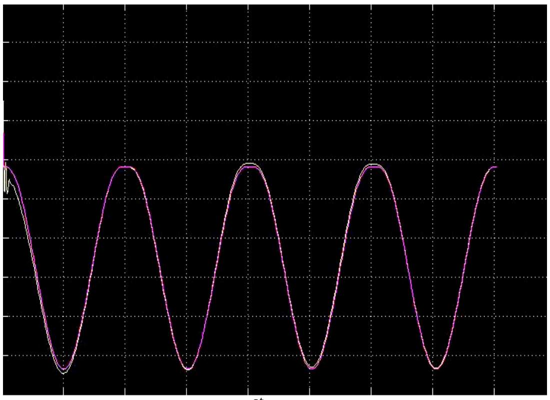

Fig.7. 1st jont angle

These results illustrate increasing in total error during 10 iteration after the tuning phase.

Iteration: 1 Total Error 7.6864
Iteration: 2 Total Error 1.0291
Iteration: 3 Total Error 0.46992
Iteration: 4 Total Error 0.19265
Iteration: 5 Total Error 0.09609
Iteration: 6 Total Error 0.07436
Iteration: 7 Total Error 0.064329
Iteration: 8 Total Error 0.066136
Iteration: 9 Total Error 0.061094
Iteration: 10 Total Error 0.055988

The pictures of trajectory in Tool space is as bellow: :(each picture is for one iteration) These pictures are plotted with XY graph in Matlab Simulink Environment.

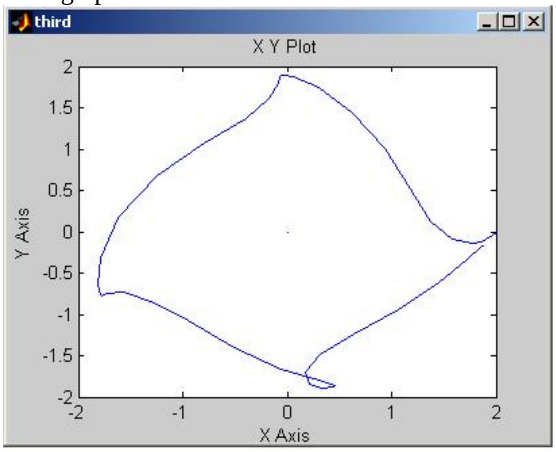

Fig.8. Tool space trajectory in iteration NO.1.

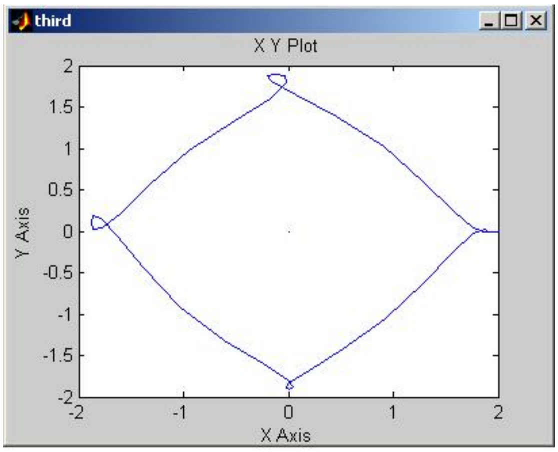

Fig.9. Tool space trajectory in iteration NO.2

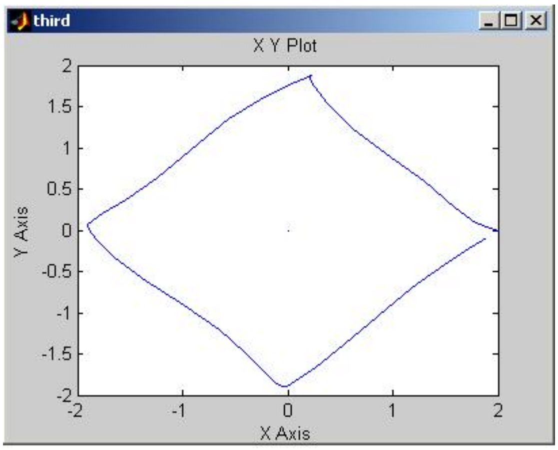

Fig.10. Tool space trajectory in iteration NO.3

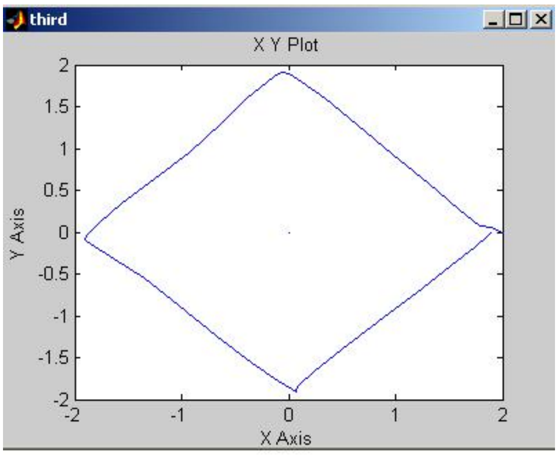

Fig.11. Tool space trajectory in iteration NO.4

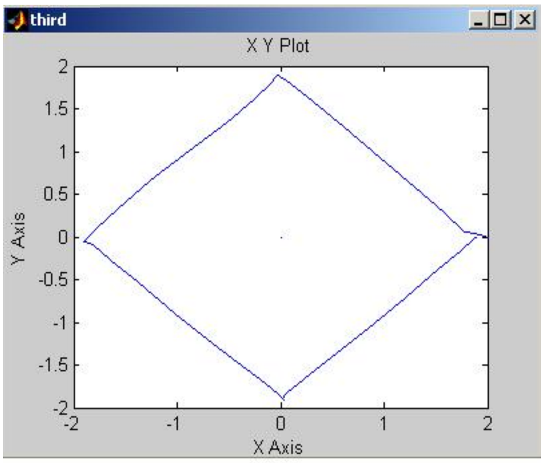

Fig.12. Tool space trajectory in iteration NO.5

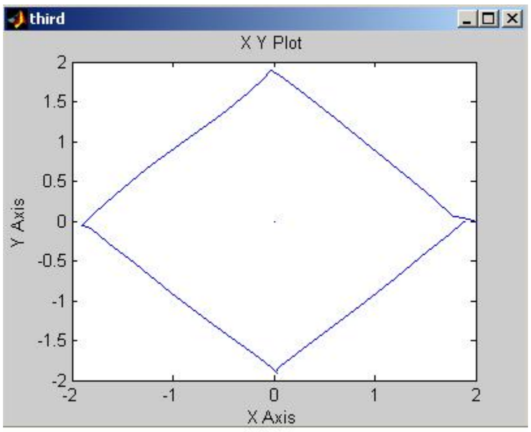

Fig.13. Tool space trajectory in iteration NO.6.

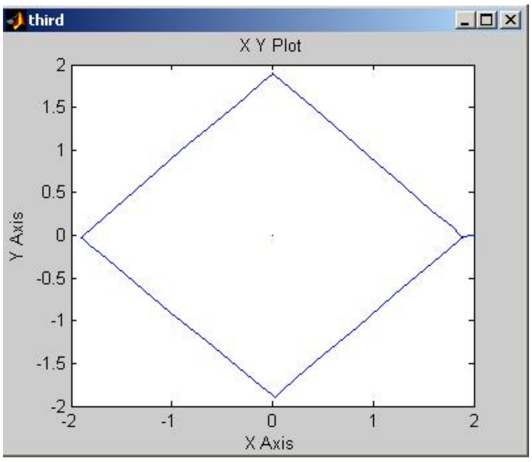

Fig.14. Tool space trajectory in iteration NO.7.

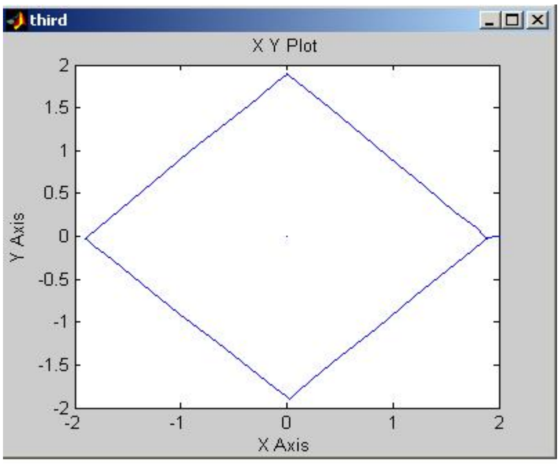

Fig.15. Tool space trajectory in iteration NO.8.

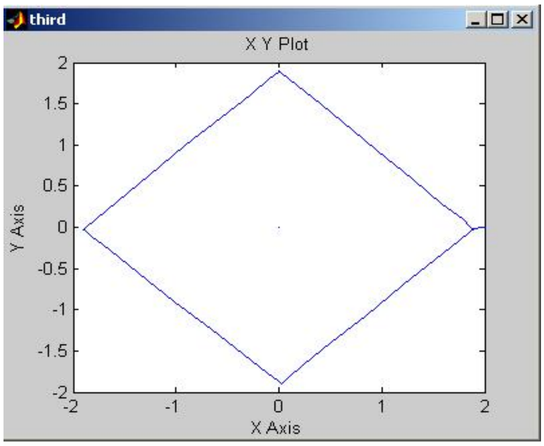

Fig.16. Tool space trajectory in iteration NO.9.

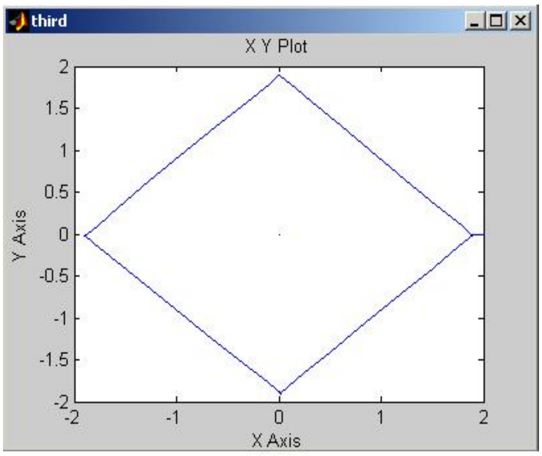

Fig.17. Tool space trajectory in iteration NO.10.

By using more iteration, error is decreased too.

VI. Conclusions and Recommendations

In this section we would be discussed the Conclusions and some recommendations for future works. Genetic algorithms which have a single objective [6], it seems that yield better results, than generating multi-objective [5] ones. But we think that preference multi-objective genetic algorithms can yield better results than both of single and generating multi-objective algorithms with respect to some objectives such as a measure of performance, a measure of stability, total error and the number of times that we get a run with smaller error than the previous run. These objectives could be given distinct weights and then be accumulated to get a single objective. The aforementioned approach is called weighted sum method and is a very powerful type of multi-objective preference methods. The weights can be fixed or variable according to the type of algorithm that be used. Generating multi-objective algorithms based on Pareto approach don't have certainty because they wouldn't yield a response without sacrificing any objective. Future works can be done with swarm intelligence and other method of machine learning such as reinforcement learning. And also a good variety of works could be done in the performance and stability aspects.

REFERENCES

[1] Siciavicco and B.Siciliano, Modeling and control of robot Manipulators, McGraw-Hill, 1996.
[2] R.J. Schiling, Fundemental of Robotics, prentice Hall, 1990.
[3] Z.Bien and J.-X. XU, Iterative learning control, Analysis, Design, Integration and Application, Kluwer Academic Publisher, 1998.
[4] S.Boming, "On Iterative Learning", PhD thesis, National University of Singapore, 1996.
[5] Fonseca, C. M. and P. J. Fleming. An overview of evolutionary algorithms in multiobjective optimization. Evolutionary Computation 3(1),1998,pp 1–16
[6] G. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley, Reading, MA, 1989.
[7] M. Setnes, R. Babuska, and H.B. Verbruggen. "Rule-based modeling: Precision and transparency." IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews, 28(1): 165-169, 1998.
[8] R. R. Yager and D. P. Filev, Essentials of Fuzzy Modeling and Control, John Wiley & Sons, Inc, 1994.
[9] D. Dubois, H. Prade, Fuzzy Sets and Systems: Theory and Applications. Academic Press: New York, 1980.
[10] X. Yao, "Evolving artificial neural networks," Proceedings of the IEEE, 87(9):
[11] 1423-1447, September 1999.
[12] J.-S. R. Jang and N. Gulley, "The Fuzzy Logic Toolbox for Use with MATLAB," The MathWorks, Inc., Natick, Massachusetts, 1995.
[13] www.mathworks.com

AUTHOR'S PROFILE

Alireza Rezaee
received his BS degree in Control Engineering from Sharif University of Technology, Iran (2002) and the MS and PhD degrees in Electrical Engineering from Amirkabir University of Technology,Iran (2005 and 2011, respectively). From 2013 till now, he is a member of mechatronics department of faculty of New Sciences and Technologies at University of Tehran, Iran. His field of research is machine learning, Bayesian networks, sensor fusion and mobile robot navigation.