Inverse Kinematics Analysis Of a Five-Degree Of Freedom Educated Type Of Alpha Robot Using Genetic Algorithm

Israa Rafie Shareef
Mechatronics Engineering Dept.
Al Khawarzmi College of Engineering
University of Baghdad \Iraq

Abstract

Inverse Kinematic is considered as one of the complicated issues in robots domain according to the nonlinearity , multiple solutions , the need for complete knowledge about robotic environment besides to other factors.

Whereas a great development has appeared in using the artificial intelligence methods to replace the classical analyzing methods.

In this paper , an approach for using the Genetic Algorithm is suggested to deal with the Inverse Kinematics problem of a five degree of freedom robot .

Firstly the forward and inverse kinematics of the robot is introduced , then the inverse kinematic equations are solved by the Genetic Algorithm with the aid of MATLAB programming , the results were compared using three ways , firstly by the classic analytical equations , secondly by the robot simulation program ROBCIM , and lastly by the MATLAB Genetic Algorithm Toolbox .

Genetic Algorithm had accomplished good results , and got rid of many limitations which may face other analytical methods .

Key-Words : Robotic Manipulator , Inverse Kinematics (I. K.) , Genetic Algorithm (G. A) .

1-Introduction

According to the definition presented by the International Standard Organization (ISO) , the robotic manipulator (named as industrial robot or as robotic arm) is an automated, programmed ,multi-purposes machine which deals with several domains to handle work parts.

It contains chain of links , with an end –effector or gripper carrying the needed tool (like welding gun or drill) , these links are attached by joints (either prismatic or revolute) .while , saying 5-axes means having five independently moving degrees of freedom (DOF) [16.[

The robotic kinematic structure deals with the calculation of the end effector position (without outer effected force) , and to make the robotic manipulator adopt the wanted configuration [22].

The manipulator has non-linearity kinematic equations to be solved [1]. The robot kinematic analysis is handled in two ways ; forward or direct (F. K.) , and inverse kinematic (I. K.) , where the second one is the most important because it gives the robotic joints' values to reach any spatial point . Thus , the I.K. is the important factor in following paths , and controlling robotic motion [9].

In (F. K.),we have the position of the end-effector (x,y,z) and its orientation (θx , θy ,θz) = f(q) Function of (q) the joints' variables either revolute or prismatic.

While in I.K. , the opposite case is occurred where , $q=f^{-1}$(x,y,z) and (θx , θy ,θz).

For its importance ,I.K. had been studied for years , for its roll in manipulator's controlling , as the robot actuators deal with joint space , while the end-effector deals with cartesian space , thus , the I.K. is the way to transform from the cartesian to the joints space [11] .

Good development had been achieved in I.K. for the recent years and many approaches were presented [14].

I.K. may has multiple solutions due to the non linearity in equations and the robot configuration's dependence [23].

I.K. is considered difficult because of : many unknowns equations variables , multiple solutions , some solution are undefined [12].

Many methods were used to solve the I.K. problems like jacobian inversion , optimization , numerical and lastly the Artificial Intelligence methods used like Neural Networks , Fuzzy , and Genetic Algorithms [14].

In this paper , we will analyze the I.K. of a 5-DOF robotic manipulator, using one of the most important Artificial Intelligence methods , which is the Genetic Algorithm. Reaching to this goal , we will start by defining the used robot , knowing its physical characteristics , D-H (Denavit – Hartenberg) representation , transformation matrix and analyzing the F.K. of this robot , besides to finding the I.K. equations using the classical approach .

Estimating the I.K. equations of this robot using the G.A. approach will be simulated using the MATLAB Genetic Algorithm Toolbox [5].

The robot which will be discussed for its I.K. , is a scale model for the famous robot (ALPHA) robot as mentioned in [20] , that's because of the high cost of the original robot [8] , we substitute by this model of laboratory robot for the sake of reach approximately to its physical information and limitations , Figure1 below show the microbot ALPHA in its laboratory type .

Figure (1-a) the ALPHA robot

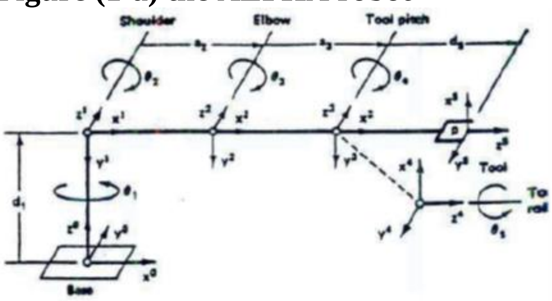

Figure (1-b) Link coordinates of the ALPHA robot

It's a 5-DOF robotic manipulator , articulated arm , delivered with a specific software that allows it to be simulated online and offline by computer .

This robot has Base , Shoulder , Elbow , Tool-roll and Tool pitch joints' motions providing the five degrees of freedom , besides to a grip movement by its two fingers gripper[24].

Figure1-b clarifies the link coordinates and D-H parameters of the robot .

The most important step in analyzing any robot kinematics , is to find the transformation matrix which relates between the base and the end-effector joints , in order to reach this matrix , we have to establish the Denavit-Hartenberg parameters which were defined in detail in [20&6] , as follows in **Table1**:

Table1 the robot parameters

Joint	θ	*d* *in millimeter*	*a* *in millimeter*	α *in degree*
Base	q 1	255	0	- 90
Shoulder	q 2	0	190	0
Elbow	q 3	0	190	0

Too roll	q 4	0	0	- 90
Tool pitch	q 5	115	0	0

Applying these parameters in the standard transformation matrix from base to tool , we'll get :

T_{base}^{tool} =

[column 1   column2   column 3   column 4]

where , **Column1**=

$$\mathbf{C}os\ q1\ Cos\ q234\ Cos\ q5 + Sin\ q1\ Sin\ q5$$

$$\mathbf{S}in\ q1\ Cos\ q234\ Cos\ q5 - Cos\ q1\ Sin\ q5$$

$$-\mathbf{S}in\ q234\ Cos\ q5$$

$$\mathbf{0}$$

Column2= $-\mathbf{C}os\ q1\ Cos\ q234\ Sin\ q5 + Sin\ q1\ Cos\ q5$

$$-\mathbf{S}in\ q1\ Cos\ q234\ Sin\ q5 - Cos\ q1\ Cos\ q5$$

$$\mathbf{S}in\ q234\ Sin\ q5$$

$$\mathbf{0}$$

Column 3= $-\mathbf{C}os\ q1\ Sin\ q234$

$$-\mathbf{S}in\ q1\ Sin\ q234$$

$$-\mathbf{C}os\ q234$$

$$\mathbf{0}$$

Column 4= $\mathbf{C}os\ q1\ (190\ Cos\ q2 + 190\ Cos\ q23 - 115\ Sin\ q234)$

$$\mathbf{S}in\ q1\ (190\ Cos\ q2 + 190\ Cos\ q23 - 115\ Sin\ q234)$$

$$\mathbf{2}55 - 190\ Sin\ q2 - 190\ Sin\ q23 - 115\ Cos\ q234$$

$$\mathbf{1}$$

Thus the robot –tool-configuration will be :

$$\mathbf{C}os\ q1\ (190\ Cos\ q2 + 190\ Cos\ q23 - 115\ Sin\ q234) \quad ..\ 1$$

$$\mathbf{S}in\ q1\ (190\ Cos\ q2 + 190\ Cos\ q23 - 115\ Sin\ q234) \quad ..\ 2$$

$$255 - 190\ Sin\ q2 - 190\ Sin\ q23 - 115\ Cos\ q234 \quad ..\ 3$$

$$-\exp(q5)\ Cos\ q1\ Sin\ q234 \quad \dots\ 4$$

$$-\exp(q5)\ Sin\ q1\ Sin\ q234 \quad \dots\ 5 \quad ;$$

$$-\exp(q5)\ Cos\ q234 \dots\ 6$$

Now , the Inverse kinematics of thi robot is : 4

q 1(the base joint) =atan (w2 , w1)

b 1=cos q1 w1+sin q1 w2-a4 cos q234 +d5 sin q234

b2 = d1 – a4 sin q234 – d5 cos q234 – w3

taking the expressions of w from tool configuration and substitute in b1 , and b2 ;

b1 = a2 cos q2 + a3 cos q23

$\boldsymbol{b1}^2 = a2^2 \cos q2^2$ +2 a2 a3 cos q2 cos q23 $+a3^2 \cos q23^2$

b2=a2 sin q2 + a3 sin q23

$\boldsymbol{b2}^2 =$

$a2^2 \sin q2^2$ +2 a2 a3 sin q2 sin q23 $+a3^2 \sin q23^2$

$b^2 = b1^2 + b2^2$

$= a2^2(cos\ q2^2 + sin\ q2^2)$ $+a3^2(\cos q23^2 + \sin q23^2)$+ 2 a2 a3 (cos q2 cos q23 + sin q2 sin q23)

$=a2^2 + a3^2$+2 a2 a3 cos q3

q 3 = ± arcos $\frac{b^2 - a2^2 - a3^2}{2\ a2\ a3}$

b1 = a2 cos q2 + a3 (cos q2 cos q3 – sin q2 sin q3)

b2 = a2 sin q2 + a3 (sin q2 cos q3 + cos q2 sin q3)

b1 = (a2 + a3 cos q3) cos q2 – (a3 sin q3) sin q2

b2 = (a2 + a3 cos q3) sin q2 + (a3 sin q3) cos q2

solving the two simultaneous linear equations :

cos q2 $= \frac{(a2 + a3\cos q3)b1 + (a3 \sin q3)b2}{b^2}$

q 2 =± arcos $\frac{(a2 + a3\cos q3)b1 + (a3 \sin q3)b2}{b^2}$ 5

or using sin q2 $= \frac{(a2 + a3\cos q3)b2 - (a3 \sin q3)b1}{b^2}$

q 2 = atan [(a2 +a3 cos q3) b2 – (q3 sin q3) b1 , (a2 + a3 cos q3) b1 + (a3 sin q3) b2]

q 234 =atan (-(- cos q1 w4 + cos 1 w5) , – w6)

q 4 = q234 – q2 – q3

q 5 = π ln $((w4)^2+(w5)^2+(w6)^2)$^(1/2)

Genetic Algorithm (GA)

GA is a special category of evolutionary algorithms [19] , it is an optimization method which mimic the behavior of biological evolution [7], its first appearance was in nineties [15]. In this paper , a GA is proposed to deal with the solving of robot's I.K. equations because of reasons , some of them :

GA as an A.I.(Artificial Intelligence) method acts as the robot's mind , making decisions based on environment's data , besides its ability to deal with the non linear functions which the robot has[4].

Also GA is considered as a technique to find the mostly best of approximated solutions for difficult and complex problems like the I.K. of robots [2].

By using GA to solve the I.K. , we're trying to get rid of the classical (like numerical) methods disadvantage of requiring iterative processes which may fail to reach a solution .

GA is selected because of its fast convergence , besides of providing the real-time response , comparing with the classical methods which are very slow with redundant iterations [17].

With GA , there's no need for function's derivatives to choose the optimum [19], and it does not stuck in local minimum or optimum [15 , 3 , 21].

In GA , processes of "initialization , fitness , selection , cross-over , and mutation " would be noticed. The first generation's parents or individuals will be prepared , then those with best fitness's function will be combined to deliver new generation of children or individuals having better fitness values than their parents . This process will continue till the convergence of the population around some individual's values with the best fitness value ever [18].

The next steps will discuss the main procedures of the G.A. with its application for our robot ; 1. Initialization and encoding 2. Fitness 3.Selection 4.Cross over 5. Mutation

1.Initialization: Firstly, randomly values were generated close to the I.K. results for each of joints variables (ө' s) , these values would represent the parents in the first generation . In fact , the initialization of joints' values (chromosomes) must be after the angles encoding step , representing in real numbers or coded as strings.

The next steps will discuss for one joint angle in a parallel time , for example ө1 and this will be applicable on other joint angles .

2. Fitness (or Evaluation) : Each chromosome (joint angle) will be evaluated by the fitness function to decide which of them will resolve the value of the end-effector position P_x , P_y , P_z which are described by the equations *1,2 & 3* respectively , those equations will be the indicator to the

fitness of the studied chromosome as they'll be compared for the ability to give the tool position, with the original position , where for the i^{th} individual ;

X=$P_x - P_x(i)$

Y=$P_y - P_y(i)$

Z=$P_z - P_z(i)$

Evaluation = $\sqrt{X^2 + Y^2 + Z^2}$

Now, the chromosomes of the current generation will be arranged in descending order according to its fitness function value , then the best ones will be selected by the next step.

3. selection : from all the individuals , those of suitable fitness will complete to the next generation ,
here another constrains have to be checked like the individuals occur within the robot's design limits , singularity phenomenon , and obstacles avoidance if existed .
4. cross over : (reproduction process) , where each two chosen chromosomes will give new offspring , many methods are available , the arithmetic crossover is used here .
5.Mutation: where a change is inserted to the individuals , in this paper, the uniform mutation is used lightly, to give the opportunity to get rid of the local minimum as it may takes place , and to ensure convergence through generations . [10,13]
This work algorithm is clarified by **Figure2**:
The MATLAB 2013 Genetic Algorithm GUI (Graphical User Interface) which is used in this work appears as shown in figure (3) , where the optimization choice is taken from the artificial intelligence applications' list ,

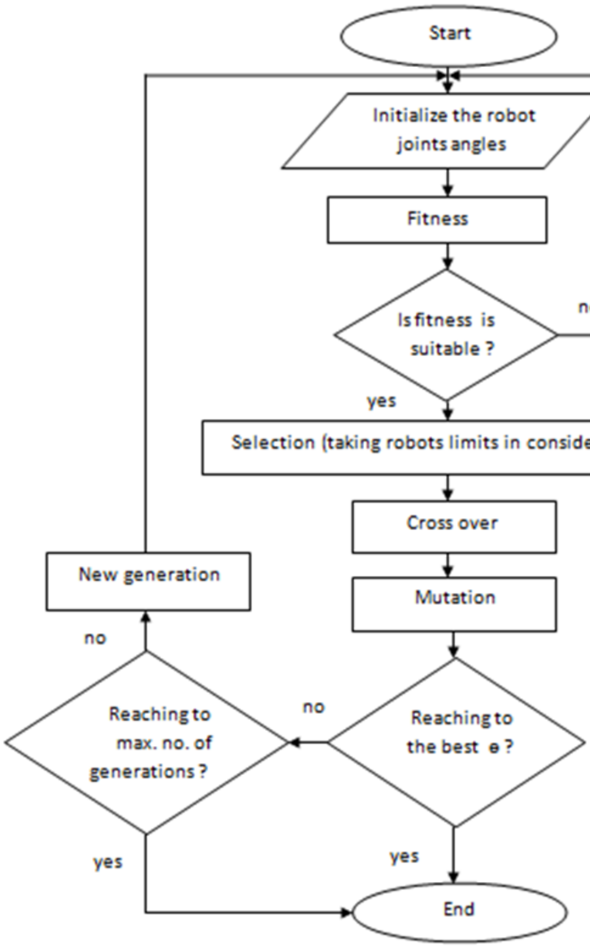

Figure 2 The work approach flow chart

The MATLAB 2013 Genetic Algorithm GUI (Graphical User Interface) which is used in this work appears as shown in figure (3) , where the optimization choice is taken from the artificial intelligence applications' list , Then , the function is assigned as it was set previously using the m-file (here the function is the inverse kinematics equations for each robotic joint variable and will be called respectively) , also the number of variables would be selected which was chosen to be one for having each joint variable individually , thus its physical constraints limits would be enrolled as the variable bounds .

Values are entered according to the user data like the population size , fitness limits and stopping criteria , changes are also possible in each step of the genetic algorithm like the crossover and mutation functions .

9

What is really going on inside the GA will start by the initialization process where a randomly set of joint angle values are delivered by the program to be the solutions set pool for the I.K. problem (these suggested or randomly delivered values can be set manually to alter the solution from a classical one to an optimized one and that's what had been done in this work) , the set values depend on the I.K. equations for each robotic joint .

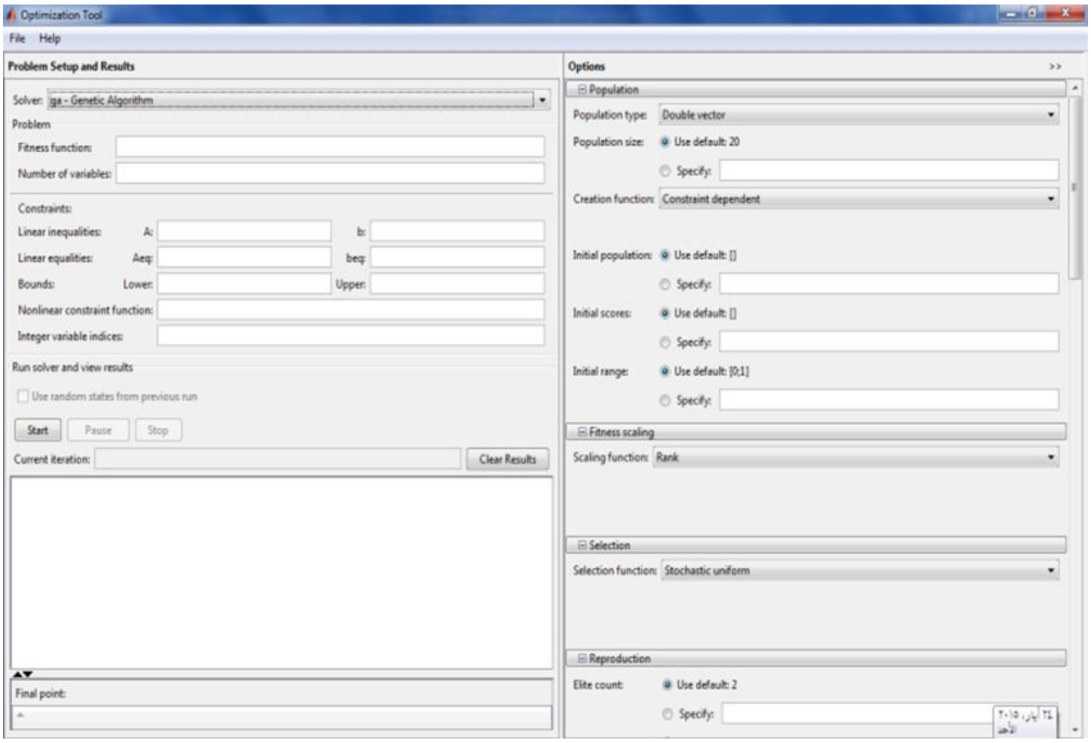

Figure (3) the MATLAB genetic interface

Then the Coding process will change the decimal joints values into binary , this will be followed by the fitness function which is constructed to give the inverse equations , the selection step now is taken to choose the best to survive for the next generation (the step has many choices to accomplish with either depending on uniform manner which is used or randomly like roulette method).

The crossover step will be between the best from all to be the parents for the new offspring (various ways are possible like single point , two points which are both used , or arithmetic)

If needed , the mutation process is available also (uniform , Gaussian and other methods) , in this work, the mutation is used in a limited way.

These processes continue till reaching to the best result , or till reaching to the maximum number of iterations which is delivered previously , or when no noticed change in results will take place (in fact even the stopping criteria could be limited in the program).

Next is a detailed example clarifying the way of solving the GA for this work ;

For the end-effector given position (337.5 , 129.5 , 162)mm , the base joint variable q1 is found by table 2:

Table (2)/A: an example of GA details solving (1st generation)

First Generation						
Chromosome	X	F(X) Fitness function (atan2)	G(X) = F(X) / ∑ F(X)	4 G(X)	Account	No. of Set
100101100	300	23.428	0.254	1.016	1	2
101000000	320	22.109	0.240	0.96	0	0
100011000	280	24.904	0.270	1.08	1	2
101001010	330	21.501	0.233	0.932	0	0
		∑ F(X) = 91.942			∑ Account = 2	

Table (2)/B: continue of an example of GA details solving (1st generation)

The remainder chromosomes	X	Chromosomes prepared to crossover	Crossover type	Crossover result	X
100101100	300	100 101 100	Two point	100 011 100	284
100101100	300	100 011 000		100 101 000	296
100011000	280	1001 01100	One point	1001 11000	312
100011000	280	1000 11000		1000 01100	268

Table (3) an example of GA details solving (2nd generation)

Second Generation						
Chromosome	X	F(X)	G(X)	4 G(X)	Account	No. of Set
100011100	284	24.595	0.254	1.016	1	2
100101000	296	23.710	0.244	0.976	0	0
100111000	312	22.619	0.233	0.932	0	0
100001100	268	25.876	0.267	1.068	1	2
		∑ F(X) = 96.8			∑ Account = 2	

And so on , with the possibility for mutation in a reasonable rate within generations in order to accelerate the reach to the fittest values , in this case of study Genetic Algorithm will go on till reaching to X= 338 , this value with a considering constant y=129.5 value , q1 = 21 degree which is so near from the actual value obtaining from the laboratory readings , besides it's within the range of motion ability of this robotic joint (-185 to + 153) , this case study is shown in **figure 4**

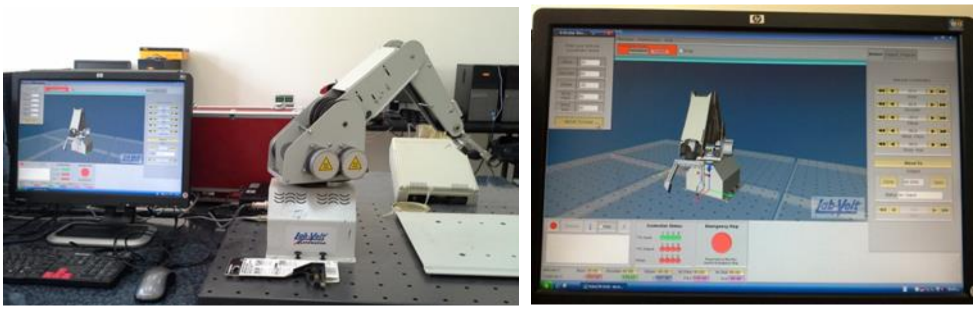

Figure4 the robot case study configuration

Next in table (4) is some of the joints values (selected the first three joints as they have mostly the responsibility of motion) which were obtained using GA for known tool positions .

For the same case (the second one in **Table4**) , some plots were selected to show the fitness , the range , and the stopping relations , where the first three joints from this case were picked to be showed as examples. The GUI for studying the base joint is shown in **Figure5** :

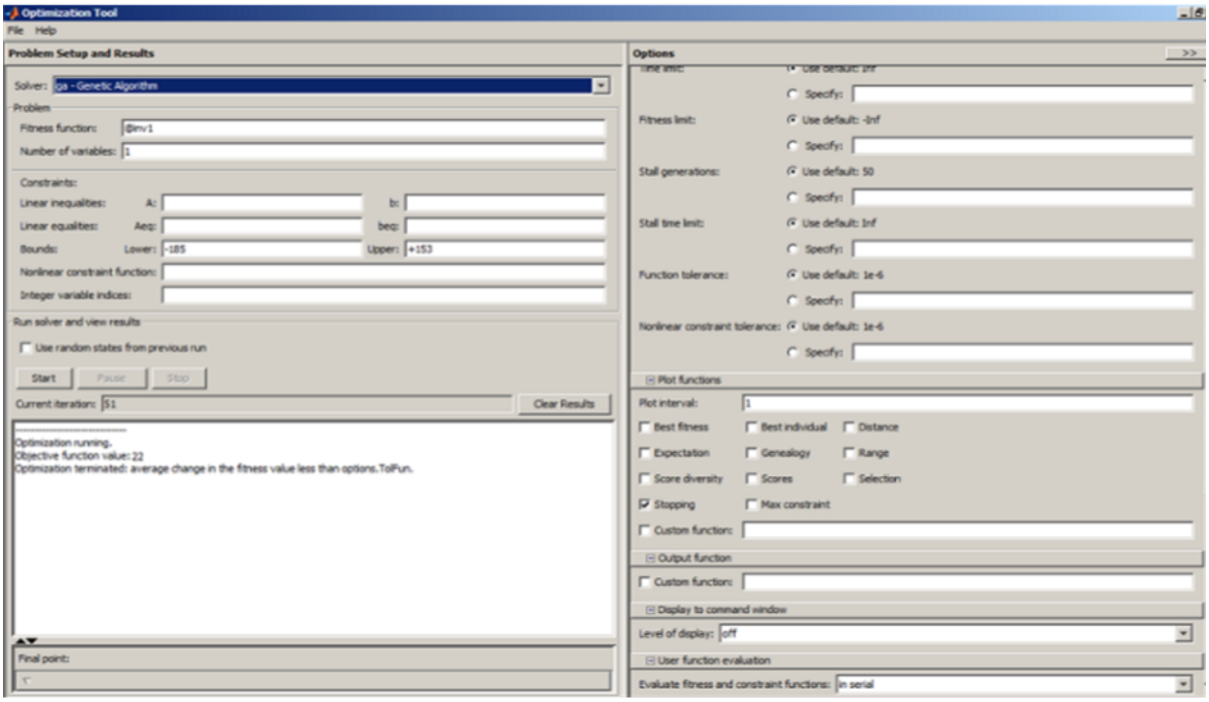

Figure5 the MATLAB interface for the base joint

Its best Fitness is shown in **Figure6** which has the relation between the generation (horizontal axis)& the fitness value (vertical axis) , finding that best value is 22 while mean value is 12.78

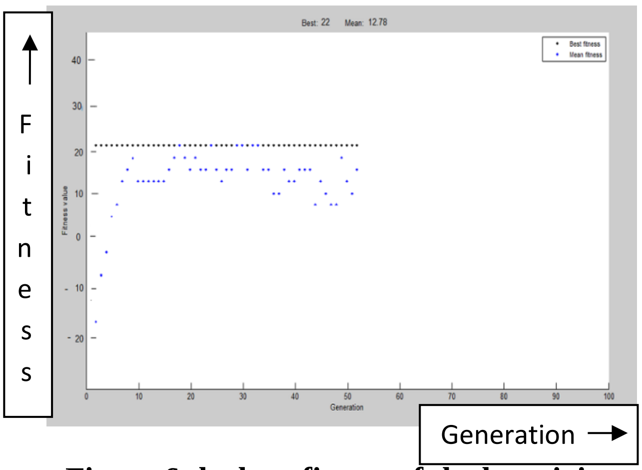

Figure 6 the best fitness of the base joint along generations

While its range is shown in **Figure7** which clarifies the best , mean , and worst values for q1(vertical axis) along generation (horizontal axis)

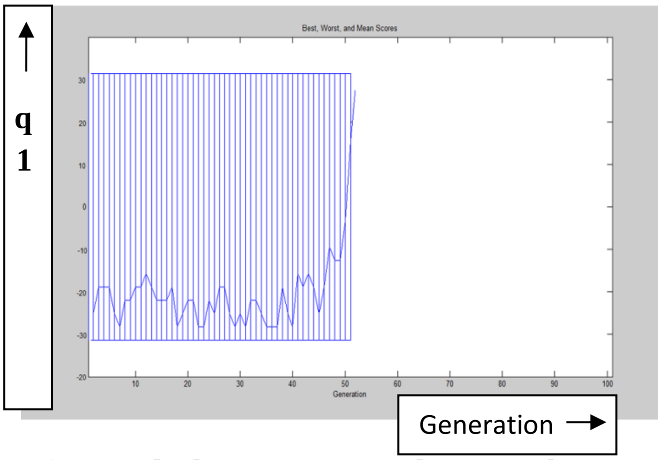

Figure7 the best , worst , and mean values of q 1 along generations

And the stopping criteria is shown in **Figure8** below where the stall and generation (the vertical axis) and how percent it met the stopping criteria (the horizontal axis)

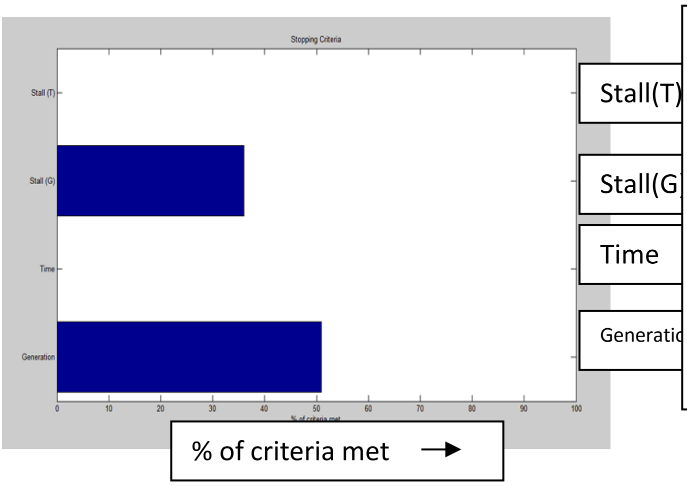

Figure8 the stopping criteria (Stall & no. of generations) of q 1

The GUI for studying the shoulder joint is shown in **Figure9** below

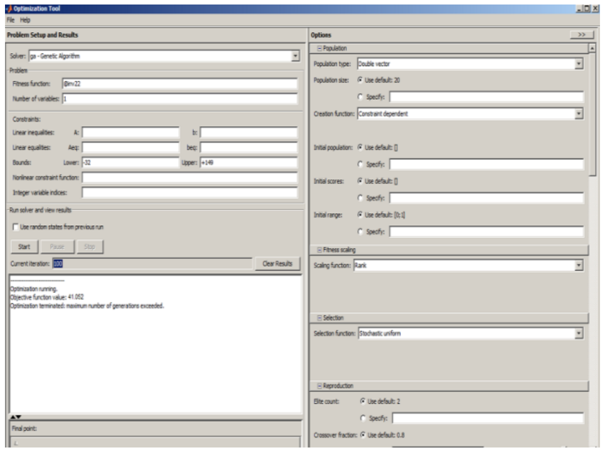

Figure9 the MATLAB interface for the base joint

its best fitness is clarified in figure (10) which has the relation between the generation (horizontal axis)& the fitness value (vertical axis) , finding that best value is 41.062 while mean value is 43.173

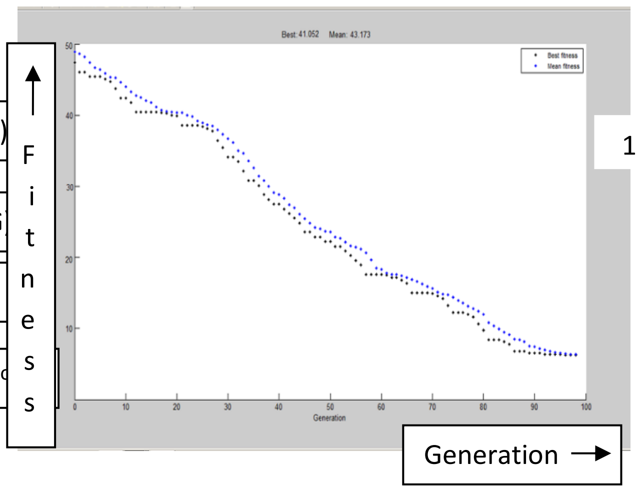

Figure10 the best fitness of the shoulder joint along generations

Its range is shown in **Figure11** which clarifies the best , mean , and worst values for q1(vertical axis) along generation (horizontal axis) as follows

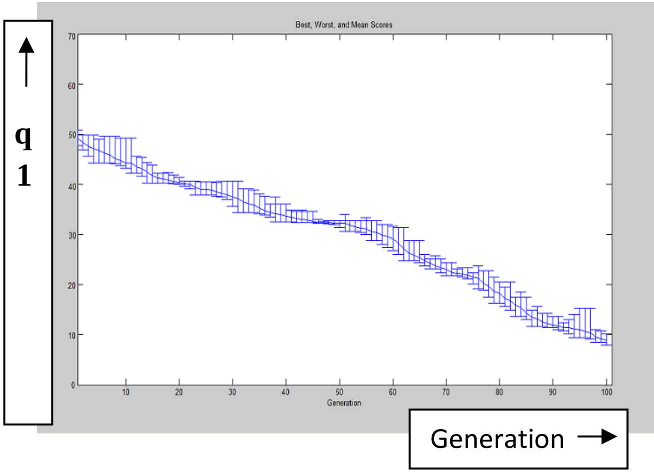

Figure 11 the best , worst , and mean values of q 1 along generations

The stopping criteria of the shoulder joint is shown in **Figure12** where the stall and generation (the vertical axis) and how percent it met the stopping criteria (the horizontal axis)

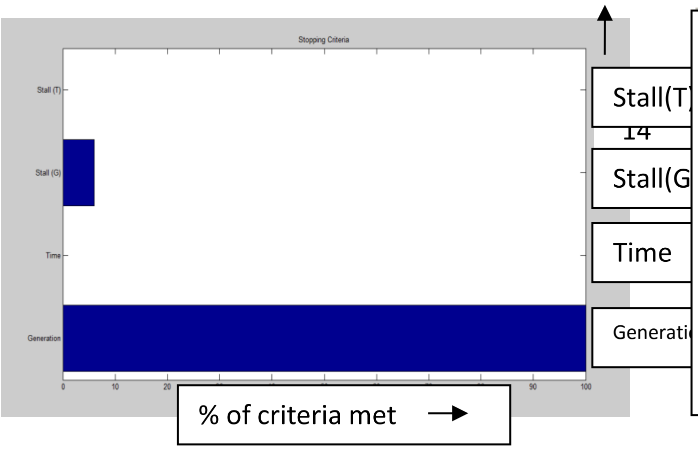

Figure 12the stopping criteria (Stall & no. of generations) of q 2

The elbow joint interface is shown in **Figure13**

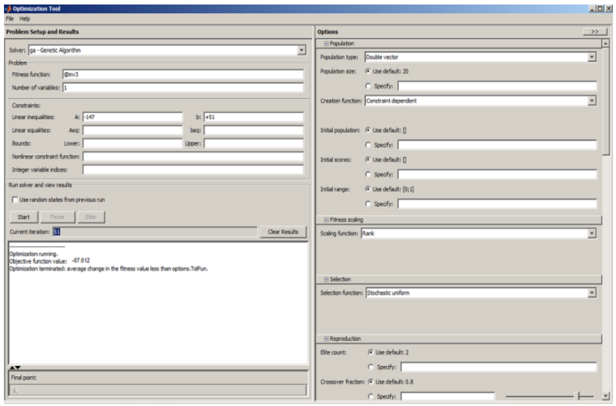

Figure 13 the MATLAB interface for the base joint

its best fitness is clarified in figure (14) which has the relation between the generation (horizontal axis)& the fitness value (vertical axis) , finding that best value is -87.612 while mean value is -79.2

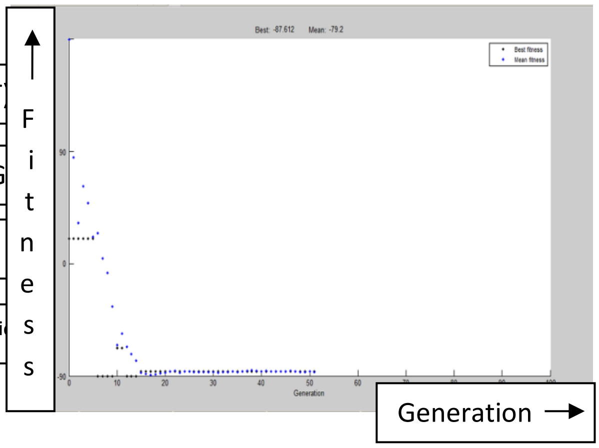

Figure 14the best fitness of the elbow joint along generations

Then the range of q3 is obvious in **Figure 15** below , which clarifies the best , mean , and worst values for q1(vertical axis) along generation (horizontal axis)

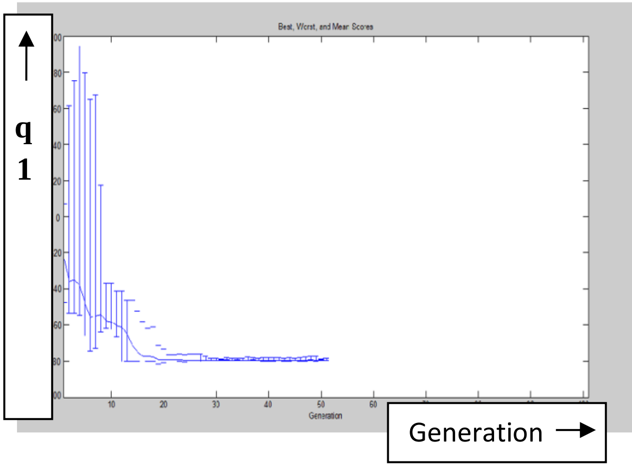

Figure 15 the best , worst , and mean values of q 3 along generations

The stopping criteria of q3 is shown in Figure16 where the stall and generation (the vertical axis) and how percent it met the stopping criteria (the horizontal axis)

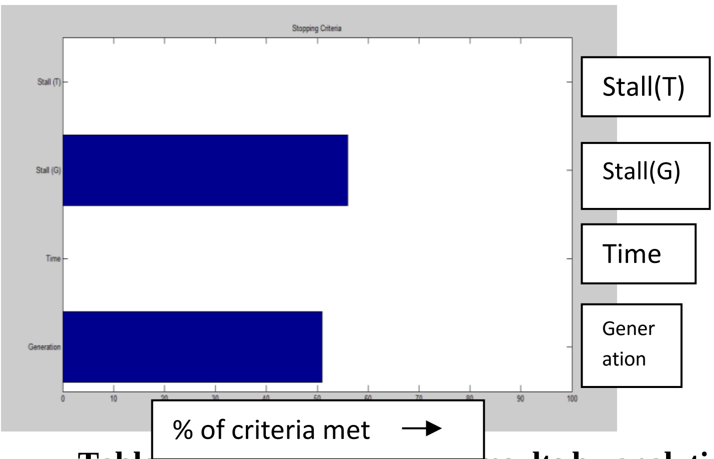

Figure16 the stopping criteria (Stall & no. of generations) of q 3

And below , **Table**4 that contains some of the best results which had been gained during work

In fact , to reach to the most acceptable results , many changes had been made in the available options of the GA characteristics like ; controlling the population type and size , the created function , the initial population , scores and range ,besides the fitness scaling . The crossover , the mutation and the stopping criteria were controlled also.

The crossover probability was chosen initially to be 0.85 , the mutation probability 0.05

Table (4) Inverse kinematics results by analytical , simulated , and GA approaches

	Fitness

End-effector position (x,y,z)	Analytical approach q_1	q_2	q_3	q_4	q_5	Simulated Approach q_1	q_2	q_3	q_4	q_5	GA approach q_1	q_2	q_3	q_4	q_5
182.5 , 0 , 401.5	0	127.31	-131.57	87	87	0	130	-130	90	90	0	128.21	-132.05	88.71	87.94
337.5 , 129.5 , 162	21.14	40.05	-80	87	87	21.14	40	-83	90	90	21	38	-78.96	88.71	87.94
270.7 , 0 , 449.7	0	107.03	-90	76.22	0	0	100	-95	80	0	3.12	105.64	-91.37	77	0
208.8 , 208.7 , 316.8	45	77.8	-94	50	0	45	79	-90	50	0	39.88	75.2	-93.5	48.16	0
195.8 , 113.05, 215.3	33.6	58.43	-90	11.27	15	30	60	-90	13	10	36.03	56	-93.5	10.05	12.45
169.6 , 36.05 , 140.64	16.15	40	-95	14.07	28.88	12	40	-95	10	25	14.85	38.02	-91.4	12.48	29.03
339.65 , 123.62 , 161.98	24	40	-81.9	87	87	20	40	-85	90	90	23.4	38	-78.72	88.71	87.94
219.4 , 0 , 273.9	0	105.04	-130	76.3	87	0	100	-130	80	90	0	104.03	-132.04	77.03	87.94

Discussion

As known , the I.K. problems have nonlinear equations , besides they depend on robot configurations , thus , multiple solutions may delivered ; also , there's the important GA characteristic of presenting un limited solutions for un limited random input parents or seeds . Then , the results mentioned in this work were selected from the best of multiple tries.

From a sight to the figures above , we notice for the stopping criteria plots shown in figures (8) , (12) and (16) ,which was controlled either according to a maximum number of generations (selected randomly by MATLAB or set by user) , a limited time of calculating , or according to the fitness closeness reaching , or as a result of the stall in the algorithm execution (breaking down or collapsing just like what happen when fitness values become so far from ideal values , giving a high rate of error). In this paper , the no. of generations and the stall were depended .

From figures (6) , (10) and (14) which shows the fitness of the first three joints respectively and which is the same is subjected for the last two joints , different numbers of population were used to get the good values that as near as possible from the ideal values (that known previously from real configurations that taken by the robotic arm with the aid of its simulation program where the values of joints' angles and end-effector position are available) , so , the fitness values along the generation's propagation is clear.

Figures (7) , (11) and (15) of the range where the best , worst and mean scores of the fitness individuals during all generations are obvious , and as gradually happened
17 will be better for results than abrupt changes , meaning that the enhancement had taken place along generations starting from randomly parents individuals and ending by the best children individuals and that point was satisfied in these figures .

Conclusion

16

- The Robot Inverse Kinematic has a growing importance , its classical solutions face some problems like complexity , fail , and slowness , thus , the Genetic Algorithm represents a good alternator .
- GA doesn't require a complete knowledge about the robot work environment .
- This study had been applied on the famous Alpha robot using its mini scale model , analyzing its complete

forward and inverse kinematics classically then optimized the I.K. results using the Genetic Algorithm approach .

- GA minimizes the local minima problem that appears usually in classical methods especially with some alterations in the real time parameters and with adding mutation if needed.
- It's better for the cross over rate to be high (80% to 95%) , the mutation rate to be low (1% to 5%) , the population size must be even no. (20 to 30) but also depending on the encoding string size.
- Future work may include another evolutionary algorithms to deal with the robotic kinematics .

References

1. Abed Issa Ahmed , 2012 , "Comparison Between Genetic Algorithm and Electromagnetism - Like Algorithm for Solving Inverse Kinematics" , World Applied Science Journal 20 (7) , 946-954.

2. Albert F. Y.C. , 2011 , " Optimizing Joint Angles of Robotic Manipulator using genetic algorithm " , International Conference on computer engineering and applications , Singapore , Vol. 2 , p.134 .

3. Anderrson J. , 2000 ," A survey of multi objective optimization in engineering design" , Technical Report : LiTH-IKP-R-1097 , department of mechanical engineering , Linkping university .

18 . Ata Atef A. , 2005 , " Optimal Point to Point Trajectory Tracking of Redundant Manipulators using Generalized Pattern Search" , International Journal of Advanced Robotic Systems , Vol. 2 , No. 3 , pp. 239-244.

5. Chipperfield A. J. , 2007 , " The MATLAB Genetic Algorithm Toolbox " , IEEE Colloquium on Applied Control Techniques using MATLAB.

6. Craig ,1989, J.J. , “ Introduction to ROBOTICS : MECHANICS AND CONTROL “ , Addison Wesley Longman .

7. Holland J. , 1975 , " Adatation in Natural and Artificial Systems" , The University of Michigan Press , Ann Arbor .

8. Janhavi B.,Avinash More , 2007 , " Simulation of spray painting

operation using microbot alpha II" , Institute of technology Mumbai , India.

9. Junfeng Shan , 2010 , " Analysis of Motion Control with Trajectory Planning for HP20 Manipulator " , Third International Conference of Measuring Technology and Mechatronics Automation .

10. Koksoy , 2008 , " Robust design using Pareto type optimization" , Computers & Industrial Engineering , 55 , 208-218.

11. Kucuk Serdar , 2006 , "Robot Kinematics : Forward and Inverse Kinematics", Germany , December.

12. Lindelce R. R. , 2011 , " Robotics and Controls " ,

13. Maia da Graca Marcos , 2009 , " Trajectory planning of redundant manipulators using genetic algorithms" , Commun Nonlinear Sci Numer Simulat , 14 , pp. 2858-2869.

14. Manoch D. , J. F. , 1994 , "Efficient inverse kinematics for general 6r manipulators" , IEEE Transactions on robotics and automation .

15. Pajaziti A. , 2014 , " Robotic Arm Control with Neural Network Using Genetic Algorithm Optimization Approach " , International Journal of Mechanical , Aerospace , Industrial and Mechatronics Engineering , Vol.8 , No. 8 .

16 . Rajput R.K. , 2009 , " A Text Book of MECHATRONICS" , S. CHAND.

17. Ramirez J. , 2011 , " Optimization of Inverse Kinematics of a 3R Robotic Manipulator using Genetic Algorithms" , World Academy of Science , Engineering and Technology , Vol. 5 , p. 850.

18. Sareni B. , 1998 , "Fitness sharing and niching methods revisited" , IEEE Trans. On Evolutionary Computation , Vol. 2 , No.3 , Sep.

19. Saud Laith Jasim , 2014 , " Investigating the Guidance Feature of Searching in the Genetic Algorithm " , IJCCCE , Vol. 14 , No. 1.

Computation , Canada , July 16-21.

20. Schilling J. ,1996," Fundamentals of Robotics Analysis and Control ", Prentice Hall of India , 1996 .

21. Sehitoglu O.T. , 2001 , " A building block favoring reordering method for Gene positions in Genetic Algorithms" , Proceedings of the genetic and evolutionary computation conference , San Francisco , USA.

22. Spong Mark W. , 2005 , "Robot Modeling and Control" , 1st edition , John Wiley & Sons.

23. Tabandeh Saleh , 2006 , " A Genetic Algorithm Approach to solve for Multiple Solutions of Inverse Kinematics using Adaptive Niching and Clustering " , IEEE Congress on Evolutionary

24. The staff of Lab-Volt Ltd. , 2007 , "Automation and Robotics : INTRODUCTION TO ROBOTICS " , Lab-Volt Ltd. Canada .

تحليل الحركة المعكوسة لإنسان آلي الفا بنموذجه التعليمي ذو خمس درجات لحرية الحركة باستخدام الخوارزمية الوراثية

إسراء رافع شريف
قسم هندسة الميكاترونيكس
كلية الهندسة-الخوارزمي
جامعة بغداد / العراق

الخلاصة

تعتبر الحركة العكسية كواحدة من القضايا المعقدة في مجال عمل الإنسان الآلي و ذلك نتيجة اللاخطية و تعدد الحلول و الحاجة للمعرفة الكاملة عن بيئة عمل الإنسان الآلي إلى جوانب عوامل أخرى .
و حيث ظهر تطور كبير في استخدام طرق الذكاء الصناعي لتستبدل طرق التحليل التقليدية , ففي هذا البحث تم اقتراح منهاج عمل يستخدم الخوارزمية الوراثية للتعامل مع مشكلة التحليل العكسي للإنسان الآلي ذو الخمس درجات لحرية الحركة .
أولا تم تقديم التحليل الأمامي و العكسي لهذا الإنسان الآلي , ثم تم حل معادلات التحليل العكسي باستخدام الخوارزمية الوراثية و برمجتها بواسطة برنامج MATLAB .
النتائج قورنت بثلاث طرق : الأولى بواسطة معادلات التحليل التقليدية , الثانية بواسطة برنامج محاكاة الإنسان الآلي ROBOCIM , و الأخيرة بواسطة صندوق الأدوات للخوارزمية الوراثية في برنامج MATLAB .
الخوارزمية الوراثية حققت نتائج جيدة و تخلصت من العديد من المعوقات التي تواجه طرق التحليل الأخرى .