International Journal of the Physical Sciences Vol. 6(2), pp. 204-212, 18 January, 2011
Available online at http://www.academicjournals.org/IJPS
DOI: 10.5897/IJPS10.491
ISSN 1992 - 1950 ©2011 Academic Journals

Full Length Research Paper

Fuzzy-genetic optimal control for robotic systems

V. K Banga[1]*, R. Kumar[2] and Y. Singh[2]

[1]Amritsar College of Engineering and Technology, Amritsar (143001), Punjab, India.
[2]Thapar University, Patiala (147004), Punjab, India.

Accepted 12 January, 2011

In this paper, we present optimal control for movement and trajectory planning of a various degrees-of-freedom robots using fuzzy logic (FL) and genetic algorithms (GAs). We have evaluated and shown comparative analysis for three degree-of-freedom (3 DOF) and four degree-of-freedom (4 DOF) robotics arm to compensate the uncertainties like; Movement, friction and settling time in robotic arm movement. This paper describes genetic algorithms, which is designed to optimize robot movement and trajectory. Though the model represents is a general model for redundant structures and could represent any n-link structures. Results shows optimal angular movement of joints, it converges too quickly even if the population is very large. The result also shows the complete trajectory planning with FL and GAs and also demonstrating the flexibility of this technique.

Key words: Inverse kinematics, genetic algorithms (GAs), fuzzy logic (FL), robotic arm.

INTRODUCTION

Automation has become an extremely fast growing phenomenon, impacting all engineering applications. The robots and robotic arms have become the major part of this trend. Autonomous navigating robots have become increasingly important. Motion planning is one of the important tasks in intelligent control of an autonomous mobile robot. Optimal movement is critical for efficient autonomous mobile robot. Many proposed approaches used fuzzy logic (FL) or genetic algorithms (GAs) or neural networks. Path conditions can be modeled using fuzzy linguistic variables so as to allow for imprecision and uncertainties of path data. Many new methods have been proposed that are appropriate for dynamic environment or provide response in real-time.

The kinematics of two collaborating robot arms handling an object, such a task is much more difficult, both kinematically and dynamically (Hemami and Cheng, 1992). The obstacles have always been a source of malfunctioning of the robot and robotic arm, various efforts have been made to develop efficient arm movement trajectories for eluding obstacles. Probability goes along with the real time process and their control for better performance (Olson et al., 2000) have developed model and techniques for probabilistic self-localization for mobile robot. A basic and general framework for robot control has been developed (Gillespie et al., 2001).

Genetic algorithms (GAs) have been used as the optimization techniques for energy minimization in robotics. The GA identifies the optimal trajectory based on minimum joint torque requirements (Garg and Manish, 2002). In the path planning problem, without obstacles for closed kinematics chains with n- links connected by spherical joints in space or revolute joints in the plane. The configuration space of such systems is a real algebraic variety whose structure is fully determined using techniques from algebraic geometry and differential topology (Trinkle and Milgram, 2002).

Uncertainties in robotic arm movement have been compensated using genetic algorithms, the nature of these parameters is not to be deterministic in nature. Optimal control is concerned with control policies that can be deduced using optimization algorithms. It deals with the problem of finding a control law for a given system such that a certain optimality criterion is achieved. A Genetic algorithm based path-planning software for mobile robot systems focusing on energy consumption. This algorithm is executed within two different phases of

*Corresponding author. E-mail: vijaykumar.banga@gmail.com. Tel: +919417800072.

the optimization process. For each obstacle within the environment regarded a path circumventing it is computed in a preparation phase. In the execution phase of the GA itself, the results of the preparation phase are used to find optimum paths. Genetic algorithms are often viewed as function optimizers, although the range of problems to which GAs have been applied is quite broad (Gemeinder and Gerke, 2003; Nguyen and Morris, 2007). Industrial robots should perform complex tasks in the minimum possible cycle time in order to obtain high productivity. The problem of determining the optimum route of a manipulator's end effector reaching a number of task points is similar (Zacharia and Aspragathos, 2005). The difficulty in a robotic system is that the desired features, that is, the speed, energy efficiency and accuracy are often contradictory. If robotic arm moves very quickly it also requires a great deal of energy and if it moves too quickly then this could have a negative impact on stability and accuracy. The multiple solutions of the inverse kinematics problem should be taken into consideration to meet the objective.

Ho et al. (2007) have developed a stable adaptive fuzzy-based tracking control for robot systems with parameter uncertainties and external disturbance. Fuzzy logic system is introduced to approximate the unknown robotic dynamics by using adaptive algorithm. Hybrid fuzzy adaptive robust controller is developed for trajectory tracking problem.

Machine learning techniques, such as evolutionary algorithms or artificial neural networks for the learning of fuzzy controllers to incorporate behaviours in mobile robots are widely used (Mucientes et al., 2007). Many recent contributions on robotic arms to solve path tracking and vibration damping problems are reported. To improve the system performance by applying GAs to tune the membership function parameters of a FL controller for the robotic manipulator.

Intelligent methods can also be used in optimization of movement and trajectory planning of manipulators (Ho et al., 2007). These methods can be used for solving redundancy resolution problems. Genetic algorithms are viewed as function optimizers. The range of problems to which GAs can be applied is quite broad. GAs are tools on probabilistic and causality, not necessarily they will have the same type of evolution when applied to the same problem. ANN and FL techniques required more information regarding system and more mathematics as compare to GA. The great advantage observed in this work is that the GAs are tools of easy application and in robotics they could be thoroughly used to do several tasks, needing for that only small description of the problem. Another advantage is that present algorithm, without any much alteration could be used for more degree-of-freedom robots.

GAs tuned fuzzy logic controllers were successfully implemented for a robot arm movement. Many recent contributions on flexible link and elastic joint robotic arms focus on how to solve path tracking and vibration damping problems both in slow and fast mode control. As a result, system performances are often tiresome and intractable. The system with the new controller is simulated and its behaviour is compared with that provided by conventional and expert-designed fuzzy logic controllers (Nguyen and Morris, 2007; Alam and Tokhi, 2008). In order to solve the simultaneous localization and mapping (SLAM) problem of mobile robots, using FL and GAs. The core of the proposed SLAM algorithm is based on an island model GA (IGA) that searches for the most probable map(s), which provide robot with the best localization information. Prior knowledge about the problem domain is transferred to GA in order to speed up the convergence. FL is employed to serve this purpose and allows the IGA to conduct the search starting from a potential region of the pose space (Momotaz et al., 2008).

Fuzzy logic (FL) and genetic algorithms (GAs) have been successfully implemented on mobile robots navigation. In the genetic algorithms, selection of the fitness function parameters are task specified and the results dependent on fitness function (Doitsidis et al., 2009). FL was also implemented on robot manipulator actuated by pneumatic artificial muscles to address the position and velocity control problem. The difficulty in the design of controllers due to modeling uncertainty and disturbances of unknown origin can be reduced significantly by using FL (Boudoua et al., 2009).

In this paper, genetic optimization is employed to find optimum joint angles for various degree-of-freedom robotic systems. The genetic optimization replaces the tedious process of trial and error for a better combination of joint angles, which is valid as per inverse kinematics for robotic arm movement. The cost function in genetic algorithm as implemented in this case is augmented by three attributes viz. joint movement, fiction and least settling time. At any time the values of these three attributes is found with the help of FL. In a given case of cost function the weightages for these three attributes are determined through fuzzy reasoning. FL models have been developed for the above said three attributes as its input and the weightages as required for these three attributes in the cost function as three outputs. The developed fuzzy genetic optimal control architectures have been implemented on three and four degree-of-freedom (DOF) robotic systems. The results for optimized joint angles are presented and discussed and a new paradigm of fuzzy-GA control architecture has been contemplated. The method proposed in this paper is robust, allowing optimal robotic arm movement and adaptation to be dynamic conditions in the environment.

Robotic systems

Robotic systems are characterized by their degree-of freedom (DOF). A very simple robotic system may have two degree-of-freedom, whereas a complex a robotic system may have more degree-of-freedoms (DOFs). Robotic arm movement is effected by various joint movement parameters like friction, settling time and orientation etc. In this research work, we have considered only three major parameters that is, friction, settling time and joint movement (minimum energy).

The robotic arm movement depends upon the angular movement of the joint. Joint movement determines the power required. The joint movement must be adjusted to stay within the power available on the robotic system to be used. Friction must also be considered in relation to robotic arm movement. The actual angular arm movement is defined as theoretical angular movement, which is provided by the controller minus the movement lost due to friction. Settling time is the most important factor in the case of any real time system. It refers to the transient response, which contains dam pings (vibrations) for a given change in the input (step function). High-speed robots must have least settling time thus exhibiting minimum physical vibrations in the movement of robotic arm.

Mathematical model of three degree-of-freedom (3 DOF) robotic system

To calculate movements in dynamic systems made up of several parts, the main approach is to calculate possible movements with the aid of mathematical models. At the same time it is necessary to understand both the mechanics and the physical aspects. A vertical articulated robotic arm with 3 links (Figure 1) having length l_1, l_2 and l_3 respectively, is considered which has a three degree-of- freedom.

In three degree-of-freedom robotic arm the inverse kinematics equations are as below:

$$x = l_1 \cos \theta_1 + l_2 \cos (\theta_1 + \theta_2) + l_3 \cos (\theta_1 + \theta_2 + \theta_3) \quad (1)$$

$$y = l_1 \sin \theta_1 + l_2 \sin (\theta_1 + \theta_2) + l_3 \sin (\theta_1 + \theta_2 + \theta_3) \quad (2)$$

$$\phi = \theta_1 + \theta_2 + \theta_3 \quad (3)$$

Knowing the arm link lengths l_1, l_2 and l_3 for position (x, y) we had calculated the values of joint angles θ_1, θ_2 and θ_3.

Mathematical model of four degree-of-freedom (4 DOF) robotic system

In Four degree-of-freedom of the robotic arm the inverse kinematics equations are as below (Figure 2):

$$x=\cos \theta \,(L \cos \phi + L_4 \cos \psi) \quad (4)$$

$$y = \sin \theta \,(L \cos \phi + L_4 \cos \psi) \quad (5)$$

$$z = L_1 + L \sin \phi + L_4 \sin \psi \quad (6)$$

where ψ: Pitch angle

Let the position of fourth joint "P_4" be (x_4, y_4, z_4). Also,

$$x_4 = x - \cos \theta \,(L_4 \cos \psi) \quad (7)$$

$$y_4 = y - \sin \theta \,(L_4 \cos \psi) \quad (8)$$

$$z_4 = z - L_4 \sin \psi \quad (9)$$

The manipulator has four degree-of-freedom: joint 1 (J_1) allows rotation about the z-axis; joint 2 (J_2) allows rotation about an axis that is perpendicular to the z-axis; joint 3 (J_3) is a linear joint which is capable of sliding over a certain angle; and joint 4 (J_4) which allows rotation about an axis that is parallel to the joint 2 (J_2) axis. Rotation along joint 1 (J_1) to the base rotation θ; the angle of rotation of joint 2 (J_2), elevation angle ϕ; the length of linear joint 3 (J_3), extension L (L represents a combination of link 2 and 3); and the angle that joint 4 (J_4) makes with x-y plane, pitch angle ψ.

Knowing the arm link lengths L_1, L and L_4 for position (x, y, z) we had calculated the values of joint angles θ, ϕ and ψ.

Problem formulation

Conventional methods of optimization require an accurate mathematical model. In robot manipulator any mathematical modeling inaccuracy will hamper the mathematical optimization process. Also, as the configuration is changed, the optimization needs to be redefined. GAs is an intelligent optimization method (Gemeinder and Gerke, 2003; Nguyen and Morris, 2007; Mucientes et al., 2007). Here in this work, genetic algorithms are proposed to search the optimal angular displacement of robot arms.

The genetic algorithm for generating the population of chromosomes having optimized values. The proposed algorithm is as follows:

[Start] Generate random population of *n* chromosomes (suitable solutions for the problem).
[Fitness] Evaluate the fitness *f(x)* of each chromosome *x* in the population.
[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 (the better fitness, the bigger chance to be selected).
b. Crossover: Crossover the parents to form new offspring (children), with a crossover probability. If no crossover was performed, offspring is the exact copy of parents.
c. Mutation: With a mutation probability, mutate new offspring at each locus (position in chromosome).
d. Accepting: Place new offspring in the new population.

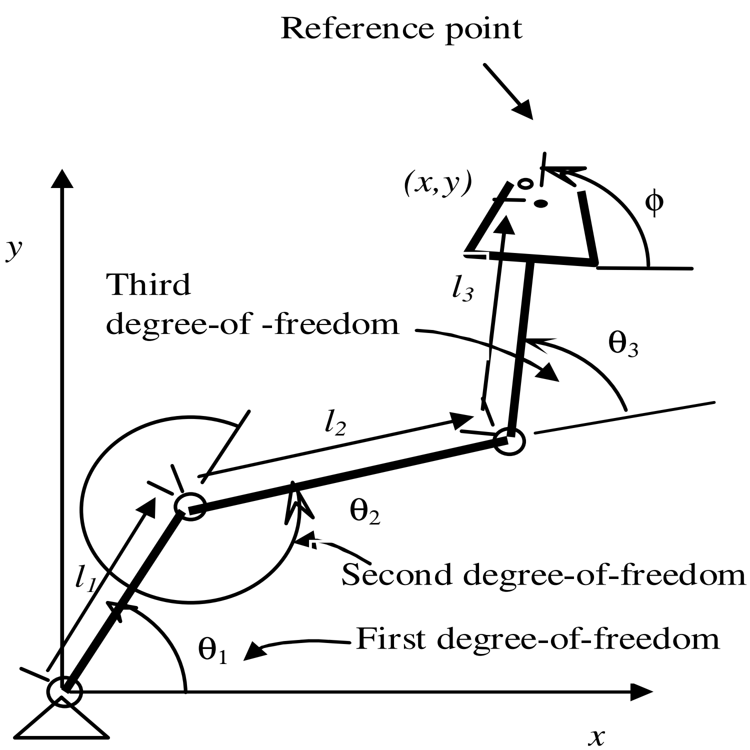

Figure 1. Three degree-of -freedom manipulator.

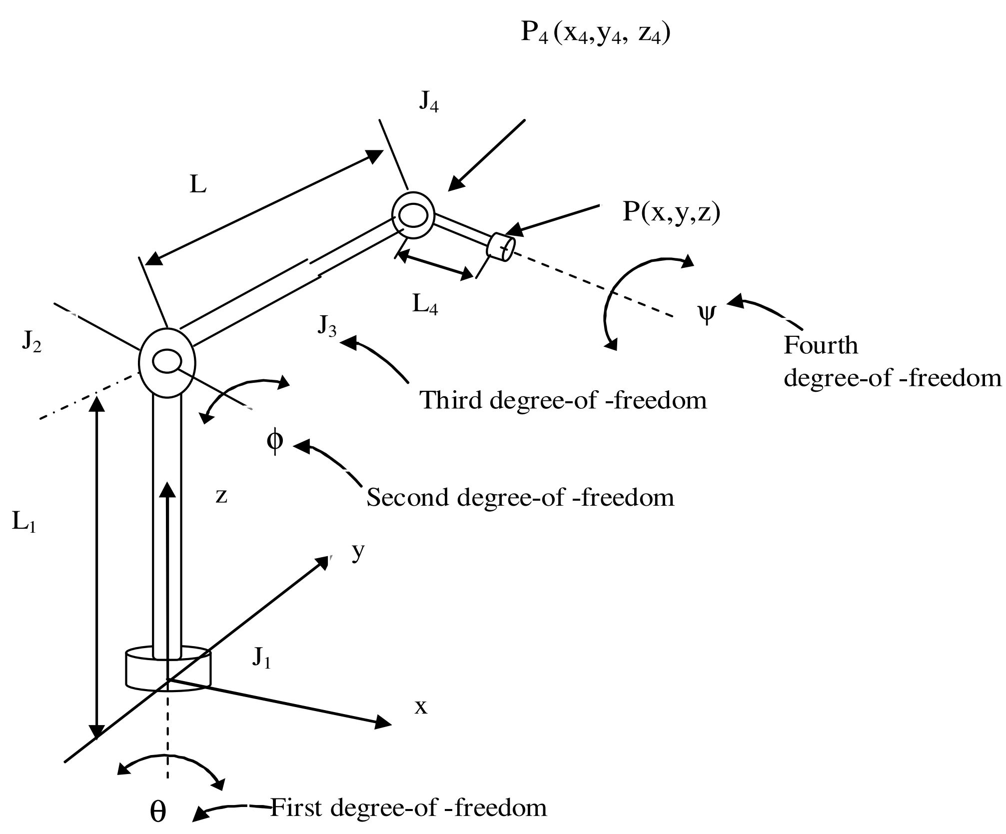

Figure 2. Four degree-of –freedom manipulator.

Table 1. Input fuzzy expressions.

1st input (joint movement) abbreviation expression		2nd input (friction) abbreviation expression		3rd input (settling time) abbreviation expression		Index representation for all three inputs
VS	Very small	VS	Very small	VS	Very small	0.00
S	Small	S	Small	S	Small	0.25
M	Medium	M	Medium	M	Medium	0.50
L	Large	L	Large	L	Large	0.75
VL	Very Large	VL	Very Large	VL	Very Large	1.00

Table 2. Output fuzzy expressions.

1st output (λ1) for abbreviation expression		2nd output (λ2) abbreviation expression		3rd Output (λ3) abbreviation expression		Index representation all three outputs
EVS	Extremely very small	EVS	Extremely very small	EVS	Extremely very small	0.00
ES	Extremely small	ES	Extremely small	ES	Extremely small	0.10
VVS	Very, very small	VVS	Very, very small	VVS	Very, very small	0.20
VS	Very small	VS	Very small	VS	Very small	0.30
S	Small	S	Small	S	Small	0.40
M	Medium	M	Medium	M	Medium	0.50
L	Large	L	Large	L	Large	0.60
VL	Very large	VL	Very large	VL	Very large	0.70
VVL	Very, very large	VVL	Very, very large	VVL	Very, very large	0.80
EL	Extremely large	EL	Extremely large	EL	Extremely large	0.90
EVL	Extremely very large	EVL	Extremely very large	EVL	Extremely very large	1.00

[Replace] Use new generated population for a further run of the algorithm.

[Test] If the end condition is satisfied, stop, and return the best solution in current population.

[Loop] go to [Fitness]

Solutions obtained from inverse kinematics are fed to the genetic algorithm for generating the population of chromosomes to be optimized.

In our case fitness of each chromosome depends upon many factors. We will consider three mains factors on which the fitness function will be calculated by applying fuzzy logic. These three main factors are:

1. Joint movement (A1)
2. Friction (A2)
3. Least settling time (Min. vibration) (A3)

First we will decide the importance and value of these three attributes for the each angle separately.

The corresponding cost function ($ƒc$) is given below by Equation 10.

$$ƒc = A1 \times \lambda 1 + A2 \times \lambda 2 + A3 \times \lambda 2 \qquad (10)$$

In case of 3 DOF robotic arm manipulator, inverse kinematics is applied on these specifications and two solutions are obtained for each link angle. Three links each having two solutions in total gives six angles. These six angles are arranged in a way that eight combinations are obtained. These eight combinations / solutions are fed into a genetic algorithm which generates the new population with the help of fuzzy logic.

If the new population is matched with the desired results, the population is stored in the search space, otherwise the inverse kinematics is again applied to the newly obtained population and the whole procedure is repeated till the required or desired results are obtained. Similarly for 4 DOF robotic arm manipulator, whole process has to be repeated as in the case of 3 DOF.

Attributes joint movement (A1), friction (A2) and settling time (A3) are inputs and weights λ1, λ2 and λ3 are outputs of fuzzy models. Tables 1 and 2 shows the inputs fuzzy expressions and output fuzzy expressions respectively.

The ranges of fuzzy input membership functions and output membership functions are from 0 to 1 (per unit basis). Table 3 shows fuzzy rules considered in this case.

Simulation and testing

Three degree-of-freedom (3 DOF)

A case study has been considered with the following specifications for 3 DOF manipulator.

Maximum reach of the robot arm: 915 mm
Length of first link (l_1): 330 mm
Length of second link (l_2): 320 mm
Length of third link (l_3): 265 mm

Origin or reference
Point (O) coordinates: (0, 0, 0)
Destination point (P) coordinates: (x, y, ø)
x: 50 mm
y: 25 mm

Table 3. Fuzzy rules.

If A1 is VS and A2 is VS and A3 is VS then λ1 is EVS and λ2 is EVS and λ3 is EVS.
If A1 is S and A2 is VS and A3 is VS then λ1 is ES and λ2 is EVS and λ3 is EVS.
If A1 is M and A2 is VS and A3 is VS then λ1 is VVS and λ2 is EVS and λ3 is EVS.
If A1 is L and A2 is VS and A3 is VS then λ1 is VS and λ2 is ES and λ3 is ES.
If A1 is VL and A2 is VS and A3 is VS then λ1 is S and λ2 is ES and λ3 is ES.
If A1 is VS and A2 is S and A3 is VS then λ1 is EVS and λ2 is ES and λ3 is EVS.
If A1 is S and A2 is S and A3 is VS then λ1 is ES and λ2 is ES and λ3 is EVS.
If A1 is M and A2 is S and A3 is VS then λ1 is VVS and λ2 is ES and λ3 is EVS.
If A1 is L and A2 is S and A3 is VS then λ1 is VS and λ2 is ES and λ3 is EVS.
If A1 is VL and A2 is S and A3 is VS then λ1 is S and λ2 is ES and λ3 is EVS.
If A1 is VS and A2 is M and A3 is VL then λ1 is EVS and λ2 is VVS and λ3 is S.
If A1 is S and A2 is M and A3 is VL then λ1 is ES and λ2 is VVS and λ3 is S.
If A1 is M and A2 is M and A3 is VL then λ1 is VVS and λ2 is VVS and λ3 is S.
If A1 is L and A2 is M and A3 is VL then λ1 is VS and λ2 is VVS and λ3 is S.
If A1 is VL and A2 is M and A3 is VL then λ1 is S and λ2 is VVS and λ3 is S.
……………………………………………………………………………………………
……………………………………………………………………………………………
……………………………………………………………………………………………
……………………………………………………………………………………………
If A1 is VL and A2 is L and A3 is VL then λ1 is EVL and λ2 is VL and λ3 is EVL.
If A1 is VS and A2 is VL and A3 is VL then λ1 is L and λ2 is EVL and λ3 is EVL.
If A1 is S and A2 is VL and A3 is VL then λ1 is VL and λ2 is EVL and λ3 is EVL.
If A1 is M and A2 is V L and A3 is VL then λ1 is VVL and λ2 is EVL and λ3 is EVL.
If A1 is L and A2 is VL and A3 is VL then λ1 is EL and λ2 is EVL and λ3 is EVL.
If A1 is VL and A2 is VL and A3 is VL then λ1 is EVL and λ2 is EVL and λ3 is EVL.

Table 4. Population from inverse kinematics.

No.	Chromosomes	Fitness value
1.	{137.91°, 217.49°, 106.14°}	1377.84
2.	{137.91°, 217.49°, 73.85°}	1281.28
3.	{137.91°, 37.49°, 106.14°}	841.44
4.	{137.91°, 37.49°, 73.85°}	744.88
5.	{84.78°, 37.49°, 106.14°}	682.59
6.	{84.78°, 37.49°, 73.85°}	586.03
7.	(84.78°, 217.49°, 106.14°}	1218.99
8.	{84.78°, 217.49°, 73.85°}	1122.43

Table 5. Population from first run.

No.	Chromosomes	Fitness value
9.	{20.84°, 36.95°, 103.84°}	482.90
10.	{20.84°, 36.95°, 76.15°}	400.11
11.	{20.84°, 28.75°, 103.84°}	458.47
12.	{20.84°, 28.75°, 76.15°}	375.68
13.	{40.33°, 36.95°, 103.84°}	541.18
14.	{40.33°, 36.95°, 76.15°}	458.39
15.	{40.33°, 28.75°, 103.84°}	516.75
16.	{40.33°, 28.75°, 76.15°}	433.96

For developing the software code using GA and FL for 3-link robotic model.

Solving these equations we get the following values for the angles of the links:

θ_1 = 137.9°, -84.78°
θ_2 = -217.49°, 37.491°
θ_3 = 106.14°, 73.85°

By applying the inverse kinematics initially and then from the three runs performed during the design and development for the optimization process, we obtain the following population as illustrated in Tables 4, 5, 6 and 7.

Table 6. Population from second run.

No.	*Chromosomes*	Fitness value
17.	{137.91°, 26.95°, 84.40°}	742.30
18.	{137.91°, 26.95°, 150.10°}	938.74
19.	{137.91°, 38.75°, 84.40°}	777.47
20.	{137.91°, 38.75°, 150.10°}	973.91
21.	{84.78°, 26.95°, 84.40°}	586.16
22.	{84.78°, 26.95°, 150.10°}	782.6°
23.	{84.78°, 38.75°, 84.40°}	621.31
24.	{84.78°, 38.75°, 150.10°}	817.77

Table 7. Population from third run.

No.	Chromosomes	Fitness value
25.	{137.91°, 16.75°, 84.40°}	713.43
26.	{137.91°, 16.75°, 172.30°}	977.44
27.	{137.91°, 60.95°, 84.40°}	845.14
28.	{137.91°, 60.95°, 172.30°}	1109.16
29.	{84.78°, 16.75°, 84.40°}	555.57
30.	{84.78°, 16.75°, 172.30°}	818.58
31.	{84.78°, 60.95°, 84.40°}	686.28
32.	{84.78°, 60.95°, 172.30°}	950.30

Table 8. Population from inverse kinematics.

No.	Chromosomes	Fitness value
1.	{17.36°, 30.09°, 84.69°}	394.83
2.	{17.36°, 30.09°, 5.26°}	157.35
3.	{17.36°, 26.26°, 84.69°}	383.41
4.	{17.36°, 26.26°, 5.26°}	145.93

From the above simulation, we obtained optimized result for various joints having lowest fitness value:

θ_1 = - 20.84°, - 40.33°
θ_2 = - 36.95°, - 28.75°
θ_3 = 103.84°, 76.15°

Four degree-of-freedom (4 Dof)

A case study has been considered with the following specifications for 4 DOF manipulator:

Maximum reach of the robot arm: 915 mm
Length of first link (l_1): 305 mm
Length of second link (L): 434 mm
Length of third link (l4): 51 mm

Origin or reference
Point (O) coordinates: (0, 0, 0)
Destination Point (P) coordinates: (x, y, z)
x: 406 mm
y: 127 mm
z: 533 mm

The system has been considered for developing the software code using GA and FL. Solving these equations we get the following values for the angles of the links:

θ =17.37°
ϕ =30.09°, 26.26°
ψ =84.69°, 5.26°

By applying the inverse kinematics initially and then from the two runs performed during the design and development for the optimization process, we obtain the following population as illustrated in Tables 8, 9 and 10.

From the above simulation, we obtained optimized result for various joints having lowest fitness value.

θ =17.36°
ϕ =20.06° , 36.29°
ψ =84.69° , 5.26°

Table 9. Population from first run.

No.	Chromosomes	Fitness value
5	{17.36°, 20.06°, 84.69°}	364.93
6	{17.36°, 20.06°, 5.26°}	127.45
7	{17.36°, 36.29°, 84.69°}	413.29
8	{17.36°, 36.29°, 5.26°}	175.82

Table 10. Population from second run.

No.	Chromosomes	Fitness value
9.	{17.36°, 26.26°, 84.69°}	383.40
10.	{17.36°, 26.26°, 5.26°}	145.93
11.	{17.36°, 30.09°, 84.69°}	394.82
12.	{17.36°, 30.09°, 5.26°}	157.34

RESULTS AND DISCUSSION

In the developed genetic algorithms, in order to obtain the optimal angular displacements for the robotic arms in the whole workspace, elitism has been retained from the previous generation to the next. Simulations, testing and comparisons have been carried out. GA does not need complete knowledge of system. Evolutionary process converges too quickly even if the population is very large. Figure 3 illustrates percentage fitness versus generation graph for 3 DOF and 4 DOF systems.

Figure 4 shows the convergence of best of each generation for 3 DOF robotic system. It can be seen that there is rapid convergence within 30 generations to an almost perfect solutions. Where as in the case of only GA there is rapid convergence within 50 generations. The remaining generation produced minor generations as the algorithm continues to optimize over the test cases.

Figure 5 shows the convergence of best of each generation for 4 DOF robotic system. It can be seen that there is rapid convergence within 20 generations to an almost perfect solutions. Where as in the case of only GA there is rapid convergence within 40 generations. The performance is good, the robotic arm moves and reaches the target position within the simulation time. The performance is optimal is over all possible input values as the evolution function exhaustively test the possible input spaces. There is an improvement from initial average fitness of more than 25% error to less than 5% error in the first 30 generations. When the solution is near the optimum point, only small improvement some time in significant is achieved in each later generation. Figure 5 shows the convergence of best of each generation for 4 DOF robotic system.

It is concluded that GA and FL is practical and effective method for achieving optimization of robotic arm angular displacements. On the other hand it also requires very

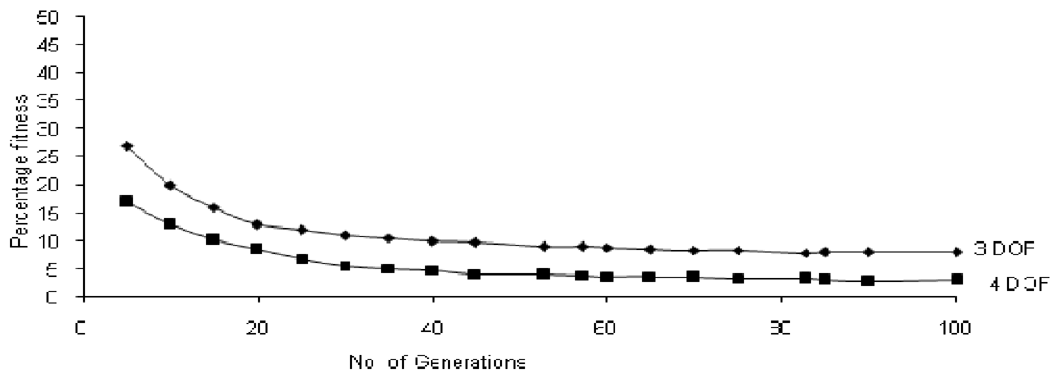

Figure 3. Percentage fitness versus no. of generations.

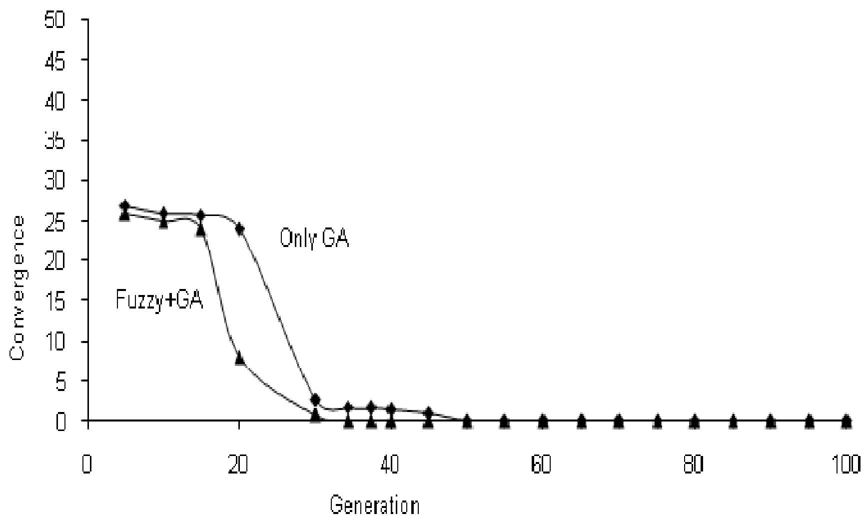

Figure 4. Convergences versus generations for 3 DOF robotic system.

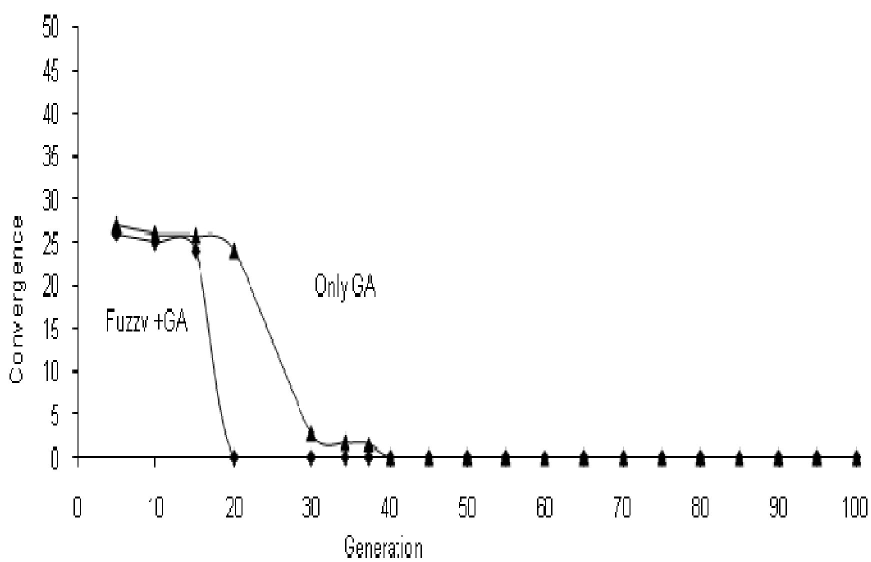

Figure 5. Convergence versus generation for 4 DOF robotic system.

less knowledge of the system description to apply this technique over other artificial intelligence technique like ANN. This algorithm can be extended to more degree-of-freedom robots with very little modifications.

REFERENCES

Alam MS, Tokhi MO (2008). Hybrid fuzzy logic control with genetic optimisation for a single-link flexible manipulator. Elsevier Eng. Appl. Artif. Intell., 21(6): 858-873.

Boudoua S, Rezoug A, Hamerlain F (2009). Fuzzy Logic Control for Manipulator Robot actuated by pneumatic Artificial muscles. J. Electrical Syst., 1: 1-6.

Doitsidis L, Tsourveloudis NC, Piperidis S (2009). Evolution of Fuzzy controllers for Robotic vehicles: The role of fitness function selection. Springer J. Intell. Robot Syst., 56: 469–484.

Garg DP, Manish K (2002). Optimization techniques applied to multiple manipulators for path planning and torque minimization. Eng. Appl. Artif. Intell., 15(3-4): 241-252.

Gemeinder M, Gerke M (2003). GA-based Path Planning for mobile Robot systems employing an active search algorithm. Elsevier Appl. Soft Comput., 3(2): 149-158.

Gillespie RB, Colgate JE, Peshkin MA (2001). A general framework for robot control. IEEE Trans. Robot. Automation, 17(4): 391-401.

Hemami A, Cheng RMH (1992). A preliminary step for path tracking of coordinated Robot arms based on kinematics. SAGE Int. J. Robot. Res., 11(3): 185-195.

Ho HF, Wong YK, Rad AB (2007). Robust fuzzy tracking control for robotic manipulators. Simul. Model. Pract. Theory, 15(7): 801-816.

Momotaz B, George KIM, Raymond GG (2008). Integrated fuzzy logic and genetic algorithmic approach for simultaneous localization and mapping of mobile robots. Elsevier Appl. Soft Comput., 8(1): 50–165.

Mucientes M, Moreno DL, Bugarin A, Barro S (2007). Design of a fuzzy controller in mobile robotics using genetic algorithms. Elsevier Appl. Soft Comput., 7(2): 540-546.

Nguyen VB, Morris AS (2007). Genetic algorithm tuned fuzzy logic controller for a Robot arm with two-link flexibility and two-joint elasticity. Springer J. Intell. Robot Syst., 49: 3–18.

Olson CF, Banga VK, Kumar R, Singh Y (2000). Probabilistic self-localization for mobile robots. IEEE Trans. Robot. Automation, 16(1): 55-66.

Trinkle JC, Milgram RJ (2002). Complete path planning for closed kinematics chains with spherical joints. SAGE Int. J. Robot. Res., 21(9): 773-789.

Zacharia PTh, Aspragathos NA (2005). Optimal robot task scheduling based on Genetic algorithms. Robot. Comput. Integrated Manuf., 21: 67–79.