*International Journal of Engineering & Technology*, 7 (4.6) (2018) 250-256

International Journal of Engineering & Technology

*Website: www.sciencepubco.com/index.php/IJET*

*Research paper*

# Generalized solution for inverse kinematics problem of a robot using hybrid genetic algorithms

**Pavan Kumar K [1]*, Murali Mohan J[2], Srikanth D[3]**

[1]*Assistant Professor, Vignan Institute Of Technology And Science, Hyderabad, INDIA*
[2] *Assistant Professor, St Marys Group of Institutions, Hyderabad, INDIA*
[3] *Assistant Professor, St Marys Group of Institutions, Hyderabad, INDIA*
**Corresponding Author Email: pavan.ms2617@gmail.com*

## Abstract

The robot control consists of kinematic control and dynamic control. Control methods of the robot involve forward kinematics and inverse kinematics (IK). In Inverse kinematics the joint angles are found for a given position and orientation of the end effector. Inverse kinematics is a nonlinear problem and has multiple solutions. This computation is required to control the robot arms. A Genetic Algorithm (GA) and Hybrid genetic algorithm (HGA) (Genetic Algorithm in conjunction with Nelder-Mead technique) are proposed for solving the inverse kinematics of a robotic arm. HGA introduces two concepts *exploration, exploitation.* In an exploration phase, the GA identifies the good areas in entire search space and then exploitation phase is performed inside these areas by using Nelder- mead technique Binary Simulated Crossover and niching strategy for binary tournament selection operator is used. Proposed algorithms can be used on any type of manipulator and the only requirement is the forward kinematic equations, which are easily obtained. As a case study inverse kinematics of a Two Link Elbow Manipulator and PUMA manipulator are solved using GA and HGA in MATLAB. The algorithm is able to find all solutions without any error

***Keywords***: *Binary Simulated Crossover, Inverse Kinematics, Hybrid Genetic Algorithm (HGA), Nelder-Mead Technique, Niching Strategies*

## 1. Introduction

In designing a robotic system the important step is solving the the inverse kinematics problem .It is a nonlinear problem and has multiple solutions. There are different procedures available for solving the inverse kinematics problem. These include the geometric, algebraic and numerical iterative methods. The analytical methods have advantage of high computing speed but these solutions exists only for a few robot configurations. Analytical expressions are possible only if robots's configuration satisfies one of following conditions

- three adjacent joint axes intersect in one point.
- three adjacent joint axes are parallel to each other.

Another method to solve the IK problem is to use numerical methods. The numerical method is used in the case where analytical expressions are hard to obtain. Most of numerical methods are divergence based so they converge to a solution which is closest to the starting point. To overcome this evolutionary algorithms like genetic algorithms have been used.

GA's are not vulnerable to local minima and they can find multiple solutions of IK problem. A real coded GA and fitness sharing niching method was used for a two Degree Of Freedom (DOF) robot to find multiple solutions.[1].In this work real coded GA in combination with tournament selection is used. An adaptive niching method is used to solve the IK problem of **a** three DOF robot [2]. With the help of niching methods the algorithm found all the possible solutions of the IK problem. .A modified genetic algorithm was used to solve the IK problem of redundant industrial robots[8]. Simulated Binary crossover and a two level binary coded GA was used The IK problem was solved by minimizing total joint displacement. High level GA was used to detect good areas and a low level GA was used to further optimize these good areas. It has been shown for continuous search spaces real coded GA's are more suitable than binary coded GA. The fitness function used was a weighted sum of joints displacement and the positional error of the end effector of robot. Genetic algorithms do not use gradient information but they are susceptible to distortion effect that is caused because of penalty term in fitness functions.

In this paper as a case study IK problem of a two link elbow manipulator and Puma manipulator is solved using Genetic Algorithm and Hybrid Genetic Algorithm. HGA introduces two concepts *exploration, exploitation.* In an exploration phase, the GA identifies the good areas in entire search space and then exploitation phase is performed inside these areas by using Nelder- mead technique Binary Simulated Crossover and niching strategy for binary tournament selection operator is used. Proposed algorithms can be used on any type of manipulator and the only requirement is the forward kinematic equations, which are easily obtained from DH parameters.

## 2. Mathematical Modeling

The objective is to find the inverse kinematic solutions for a robotic manipulator using Genetic (GA) and Hybrid Genetic Algorithm (HGA). As a test case study a two DOF planar elbow manipulator and Puma manipulator were taken and the Inverse

Kinematics solutions are found by converting the Inverse Kinematics problem into a minimization problem. Using optimization techniques like GA and HGA, the optimal Inverse Kinematic solutions were found.

For a robot with *'m'* joint variables, the position vector $\{P_o^h\}$ of the end-effector of robot relative to the base coordinate frame can be obtained from the last column of the matrix $[T_o^h]$ , The $[T_o^h]$ matrix relates the end-effector coordinate frame to the base coordinate frame and can be obtained by DH matrices[14]. The number of degrees of freedom of robot can be decreased to 'm-3' .The position vector of the robot wrist $\{P_o^w\}$ relative to base coordinate frame is obtained as

$$\{P_o^w\} = \{P_o^h\} - [R_o^{m-2}]\{P_{m-2}^h\} \quad (1)$$

where $[R_o^{m-2}]$ relates the axes of coordinate frame *'m-2'* to the base coordinate frame and $\{P_o^w\}$ is a function of joint variables $\{\theta_1\ \theta_2 \dots \theta_{(m-3)}\}^T$

For a robot working in an obstacle free environment, the minimization of the total joint displacement is taken as performance criterion

$$\Delta\theta = \|\{\theta\}-\{\theta_{in}\}\| \quad (2)$$

Where ||. || denotes the Euclidean distance, $\{\theta_{in}\}=\{\theta_{1i}\ \theta_{2i} \dots \theta_n\}^T$ represents joint variables of the robot at initial position and $\{\theta\}=\{\theta_1\ \theta_2 \dots \theta_n\}^T$ represents the joint variables at the desired location of the robotic manipulator. In order to attain a desired position $\{P_{o,des}^w\}$ of the robot wrist, eqn. (1) is used to form a equation

$$\{P_o^w\} - \{P_{o,des}^w\} = \{0\} \quad (3)$$

The limits on the joint variable values is expressed as

$$\theta_k^L \le \theta_k \le \theta_k^U \quad k=1,2,\dots m-3 \quad (4)$$

The IK problem can be solved by minimizing the following equation

$$\text{Min} \quad \|\{P_o^w\} - \{P_{o,des}^w\}\|$$

$$\text{subject to} \quad \theta_k^L \le \theta_k \le \theta_k^U \qquad k=1,2,\dots m-3$$
(5)

## 3. Solution methodology

**step 1.** Derive equations for position of end effector of manipulator

**step 2.** Define a objective function describing the error between the required and obtained end effector positions of the manipulator

**step 3.** Minimization of this objective function using GA

**step 4.** Minimization of this fitness function using Hybrid GA (real coded GA+ Nelder mead(a local search technique))

### 3.1 Minimization of this fitness function using GA

#### 3.1.1 Mechanics of genetic algorithm

The GA include following operations initialization, iterative selection of individuals made on the basis of fitness of individuals , recombination and mutation. A flow chart of a real coded GA is shown, genetic Algorithm contains following steps

- Initialization
- Selection
- Recombination
- Mutation

#### 3.1.1.1 Initialization

An initial population is generated by random sampling from the variable search space. The goal is individuals should cover entire search space, respecting the joint limits.

#### 3.1.1.2 Evaluation and selection

In the current work, the binary tournament selection operator is used fitness value of individuals are evaluated from equations defined later .These equations can be used to carry out a multimodal optimization with the objective function having zero value at the optimal points

A niching strategy for the binary tournament selection operator suggested in [10] is used to obtain simultaneously the multiple solutions of inverse kinematics function. In this method, a niche size parameter *n** is defined. For each of the two solutions participating in a tournament, the niche count of the $i^{th}$ individual, $nc_i$, is evaluated from below equation

$$nc_i = \sum_{j=1}^{N} Sh(d_{ij}) \quad (6)$$

$N$ represents the total number of individuals, $d_{ij}$ represents the Euclidean distance between the $i^{th}$ and $j^{th}$ individual and $Sh(d_{ij})$ represents the sharing function values for the $i^{th}$ individual. The latter is obtained as

$$Sh(d_{ij}) = 1-\left(\frac{d_{ij}}{\sigma_{share}}\right)^{\alpha}, \text{ if } d_{ij} \le \sigma_{share}$$
$$= 0, \quad \text{otherwise} \quad (7)$$

The $\alpha$ in above equation is taken **as** 1 and the $\sigma_{share}$ is given by

$$\sigma_{share} = \frac{\sqrt{\sum_{k=1}^{n}(q_k^L - q_k^U)^2}}{2\sqrt[n]{r}} \quad (8)$$

In *eqn. (8)*, n is the number of variables, r is the number of niches. The tournament selection is carried out by following procedure. If the niche count of both individuals is less than niche size parameter, the one with the better fitness value wins otherwise, the one with the smaller niche count wins. Thus, if one solution is overly crowded, and the other is not, the individual at second solution is selected.

#### 3.1.1.3 Recombination and mutation

In this work, a SBX (Simulated Binary Crossover) operator suggested in [13] with probability distribution same as that of the single point crossover operator used in binary-coded GA'S is used. Two children solutions $(q_{k,c}^{(1)}, q_{k,c}^{(2)})$ from two parent solutions $(q_{k,p}^{(1)}, q_{k,p}^{(2)})$ are created as follows

- A random number $u$ is created between 0 and 1.
- spread factor parameter β is calculated as follows

$$\beta = (2u)^{\frac{1}{n+1}} \quad , u \le 0.5$$
$$= \left(\frac{1}{2-2u}\right)^{\frac{1}{n+1}} \quad , \text{otherwise} \quad (9)$$

- Two children are created by following equations

$$q_{k,c}^{(1)}=0.5\left[\left(q_{k,p}^{(1)}+q_{k,p}^{(2)}\right)-\beta\left|q_{k,p}^{(2)}-q_{k,p}^{(1)}\right|\right]$$

$$q^{(2)}_{k,c}= 0.5\left[\left(q^{(1)}_{k,p}+q^{(2)}_{k,p}\right)+\beta\left|q^{(2)}_{k,p}-q^{(1)}_{k,p}\right|\ \right] \quad (10)$$

#### 3.1.1.4 Mutation

Mutation is a process of changing the variable in a chromosome with a random value
The probability of mutation $p_m$, is evaluated as

$$p_m= \frac{1}{n}+\left(\frac{t}{t_{max}}\right)\left(1-\frac{1}{n}\right) \quad (11)$$

Here $n$ is the number of variables and $t$ is the current generation number and $t_{max}$ is the maximum number of generation allowed . Thus, in the initial generation, one variable is mutated on an average with an expected 1% perturbance and as generations proceed more variables are mutated with less perturbance.

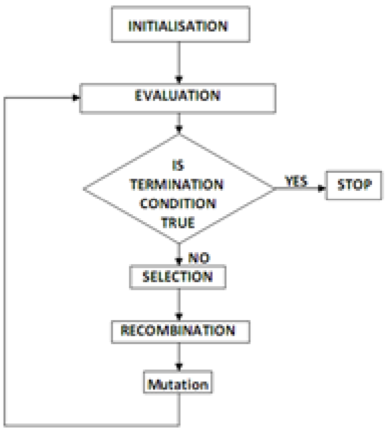


Figure 1 Flow chart of Genetic Algorithm

### 3.2 Minimisation of this fitness function using HGA

Hybrid Genetic Algorithm is genetic algorithm combined with a local search technique. This paper presents a real-coded genetic algorithm that applies a Nelder-Mead technique to solutions produced by the genetic operators The Nelder–Mead is a powerful local descent algorithm, which makes no use of the objective function derivatives. The algorithm uses a geometric construct, called simplex. A ‘‘simplex’’ is a geometrical figure with n-dimensions, of (n +1) points. In this paper we use the idea, where the GA detects good areas in solution search space then the Nelder-Mead technique is used to improve these areas. The Nelder-Mead technique generates a new space around the best point obtained from the GA, and search within this space about a better point. This approach introduces two concepts: *exploration, exploitation* In an exploration phase, the GA identifies the good areas in entire search space and then exploitation phase is performed inside these areas by using Nelder- mead technique

**Algorithm**

**step 1.** Order the points according to the fitness values at the vertices: $f(x_1)\le f(x_2)\le\ldots f(x_{n+1})$

**step 2.** The center of gravity $x_0$ is calculated for all points except $x_{n+1}$

**step 3.** *Reflection:* reflected point is calculated from eqn. 12

$$x_r=x_0+\alpha(x_0-x_{n+1}) \quad (12)$$

If the reflected point is not better than the best but better than the second worst, i.e.: $f(x_1)\le f(x_r)\le f(x_n)$ then a new simplex is obtained by replacing the worst point $x_{n+1}$ with the reflected point $\boldsymbol{x_r}$, and go to step 1.

**step 4.** *Expansion* If the reflected point is the best point of all points i.e. $(x_r)\le f(x_1)$ then the expanded point is computed from eqn.13

$$x_e=x_0+\beta(x_0-x_{n+1}) \quad (13)$$

If the expanded point is better than the reflected point i.e. $f(x_e)\le f(x_r)$ then a new simplex is obtained by replacing the worst point $x_{n+1}$ with the expanded point $x_e$, and go to step 1.
Else a new simplex is obtained by replacing the worst point $x_{n+1}$ with the reflected point $\boldsymbol{x_r}$, and go to step 1.
Else (i.e. second worst is better than reflected point) continue at step 5.

**step 5.** *Contraction* Here, it is obvious that $f(x_r)\ge f(x_n)$

contracted point is computed from eqn. 14

$$x_c=x_0+\rho(x_0-x_{n+1}) \quad (14)$$

If the contracted point is better than the worst point, i.e. $f(x_r)<f(x_{n+1})$ then a new simplex is obtained by replacing the worst point $x_{n+1}$ with the contracted point $x_c$ , and go to step 1.
Else go to step 6

**step 6.** *Reduction* For all but the best point, replace the point with eqn.15

$$x_i=x_1+\sigma\left(x_i-x_1\right) \text{ for all } \mathbf{i} \in\{2,\ldots,n+1\} \quad (15)$$

go to step 1.

Below are the values of coefficients used in Algorithm

Reflection coefficient ($\alpha$) is taken as1
Expansion coefficient ( $\beta$) is taken as 2
Contraction coefficient ($\rho$) is taken as -1/2
Shrink coefficient ($\sigma$) is taken as ½

In the reflection, since $x_{n+1}$ is the vertex with the higher associated value among the vertices, a lower value can be expected at the reflection of $x_{n+1}$ in the opposite face formed by all point***s*** $x_i$ except $x_{n+1}$.

In the expansion, if the reflection point $x_r$ is the new minimum among the vertices interesting values can be expected along the direction from $x_0$ to $x_r$

For the contraction i.e. reflected point is not better than second worst we can expect that a better value will be inside the simplex formed by all the vertices $x_i$.

### 3.3 Case Study 1: Two Link Planar Elbow Manipulator

We performed experiments on the two link elbow manipulator shown in *fig.3* It is a two degree of freedom planar manipulator. The link lengths of manipulator ($a_1$, $a_2$) are taken as 25cm and 16cm limits on the joint variables are taken as $-180°\le\theta_1\le180°$ and $--180°\le\theta_2\le180°$ .The multiple solutions of the inverse kinematics function for this robotic manipulator is because of the existence of multiple configurations as shown in *fig.4* which result in the same wrist position of robot

$$p_x = a_1c_1+ a_2c_{12} \quad (16)$$

$$p_y = a_2s_2+ a_2s_{12} \quad (17)$$

equation (16) & (17) are the forward kinematic equations of elbow manipulator

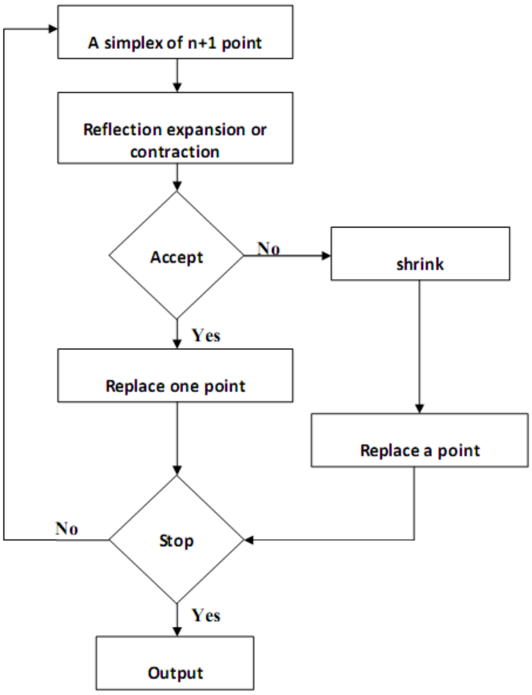


**Figure 2** Flow chart of Nelder-Mead Technique

### 3.3. 1. Fitness function for two Link Manipulator

Let ($P_{x,des}$,$P_{y,des}$) represent the position vector of end effector in the desired point. Position vector of individual ($P_{x,ind}$,$P_{y,ind}$) is given by the equations (16) & (17).The fitness function is calculated as

Fitness

$$= \sqrt{(P_{x,des} - a_1 c_1 + a_2 c_{12})^2 + (P_{y,des} - a_2 s_2 + a_2 s_{12})^2} \qquad (18)$$

**Table 1**: parameters for inverse kinematic solution of two link articulated manipulator

| | |
|---|---|
| Population size (n) | 750 |
| SBX parameter (η) | 5 |
| Number of niches (r) | 2 |
| No of generations (tmax) | 12 |
| Niche size parameter (n*) | 200 |
| Expansion coefficient | 2 |

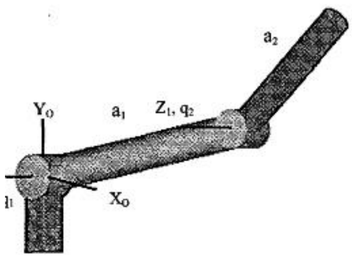

**Figure 3** Two Link planar elbow manipulator

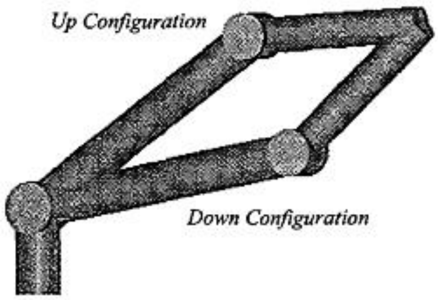


**Figure 4** Existence of multiple configurations for same wrist position

## 3.4 Case Study 2: Puma Manipulator

We performed experiments on the Puma manipulator shown in *fig. 5*. It is a three degree of freedom robot used in many industrial applications. The link lengths ($a_2$,$d_2$, $a_3$, $d_4$) are taken as 43.18 cm ,14.90cm ,2.302cm and 43.30cm respectively and the joint variable limits are taken as $-180° \le \theta_1 \le 180°$ , $-180° \le \theta_2 \le 180°$ and, $-180° \le \theta_3 \le 180°$ .The multiple solutions of the inverse kinematics function for this robotic manipulator is because multiple configurations exist as shown in *fig. 6* which result in the same wrist position .

$$p_X = C_1[d_6(C_{23}C_4S_5+S_{23}C_5)+S_{23}d_4+a_3C_{23}+a_2C_2]-S_1(d_6S_4S_5+d_2)$$

$$p_Y = S_1[d_6(C_{23}C_4S_5+S_{23}C_5)+S_{23}d_4+a_3C_{23}+ a_2C_2]+C_1(d_6S_4S_5+d_2)$$

$$p_Z = d_6(C_{23}C_5-S_{23}C_4S_5)+C_{23}d_4- a_3S_{23}+a_2S_2 \qquad (19)$$

Equation (19) gives the forward kinematic equations of the puma robot

### 3.4.1 Fitness function for PUMA Manipulator

Let ($P_{x,des}$,$P_{y,des}$, $P_{z,des}$) represent the position vector of end effector in the desired point.The fitness function is defined as

$$\text{Fitness}=(\sqrt{((P_{x,des}-(C_1[S_{23}d_4+a_3C_{23}+a_2C_2]- d_2S_1))^2 + (\mathbf{P_{y,des}} - [\mathbf{S_{23}d_4} + \mathbf{a_3C_{23}} + \mathbf{a_2C_2}] + \mathbf{d_2C_1}))^2 + (P_{z,des} - (d_4 - a_3S_{23} - a_2S_2))^2)} \qquad (20)$$

\

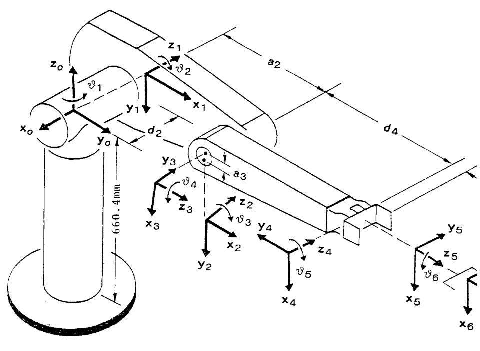

**Figure 5** PUMA Manipulator

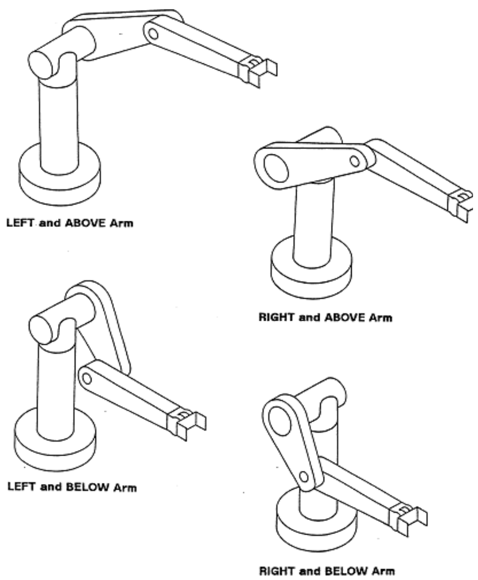


**Figure 6** Existence of multiple configurations of PUMA Manipulator

**Table 2** parameters for inverse kinematic solution of two link PUMA manipulator

| | |
|---|---|
| Link length ($a_2$) | 43.18cm |
| Link length ($d_2$) | 14.90 cm |
| Link length($a_3$) | 2.03 cm |
| Link length($d_4$) | 43.30 |
| Joint limits ($\theta_1$) | $-180° \le \theta_1 \le 180°$ |
| Joint limits ($\theta_2$) | $-180° \le \theta_2 \le 180°$ |
| Joint limits ($\theta_3$) | $-180° \le \theta_3 \le 180°$ |
| Population size (n) | 1000 |
| SBX parameter (η) | 5 |
| Number of niches (r) | 4 |
| No of generations ($t_{max}$) | 12 |
| Niche size parameter (n*) | 200 |
| Expansion coefficient | 1 |
| Contraction coefficient | 0.5 |
| Link length ($a_2$) | 43.18cm |

## 4. Results and discussions

### 4.1 Two Link Planar Elbow Manipulator

Fitness Function obtained in *eqn.18* is minimized using GA and HGA in MATLAB. In HGA good solutions from GA are taken and further optimized using Nelder mead technique. Termination condition for algorithm is twelve numbers of generations for GA and fitness function value approximately equal to zero for HGA.

**Table 3** Results of simulation experiments performed on articulated robot using GA

| S.NO. | Final position of wrist | Results obtained analytically | | Obtained values with GA | |
|---|---|---|---|---|---|
| | | $\theta_1$(in deg) | $\theta_2$ (in deg) | $\theta_1$(in deg) | $\theta_2$ (in deg) |
| 1 | (20,20) | 10.7 | 95.8 | 12.6 | 93.8 |
| | | 79.2 | -95.8 | 76.6 | -94.6 |
| 2 | (20,25) | 21.8 | 79.6 | 21.4 | 77.0 |
| | | 80.7 | -79.6 | 81.3 | -80.6 |
| 3 | (15,30) | 36.4 | 72.2 | 35.9 | 70.3 |
| | | 90.4 | -72.2 | 91.8 | -70.5 |
| 4 | (38,0) | -17.4 | 45.2 | -16.4 | 48.8 |
| | | 17.4 | -45.2 | 18.3 | -43.6 |

From the *Table. 3* and *Table. 4* we can observe that GA was able to find near optimal solutions with some error and HGA was able to find accurate values without any error.

**Table 4** Results of simulation experiments performed on articulated robot using HGA

| S.NO. | Final position of wrist | Results obtained analytically | | Obtained values with HGA | |
|---|---|---|---|---|---|
| | | $\theta_1$(in deg) | $\theta_2$ (in deg) | $\theta_1$(in deg) | $\theta_2$ (in deg) |
| 1 | (20,20) | 10.7 | 95.8 | 10.7 | 95.8 |
| | | 79.2 | -95.8 | 79.2 | -95.8 |
| 2 | (20,25) | 21.8 | 79.6 | 21.8 | 79.6 |
| | | 80.7 | -79.6 | 80.7 | -79.6 |
| 3 | (15,30) | 36.4 | 72.2 | 36.4 | 72.2 |
| | | 90.4 | -72.2 | 90.4 | -72.2 |
| 4 | (38,0) | -17.4 | 45.2 | -17.4 | 45.2 |
| | | 17.4 | -45.2 | 17.4 | -45.2 |

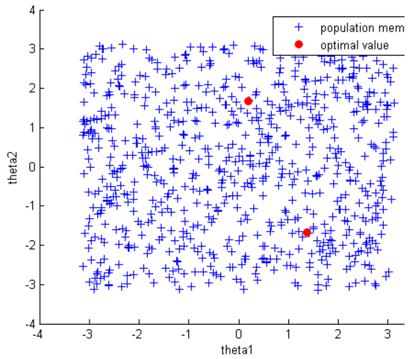


**Figure 7** Population histories at initial generation (simulation experiment no. 1)

Figures 7 through 9 show how the individuals are distributed around optimal points at the initial, an intermediate and final generation of the GA for the simulation experiment one The figures show that the GA population is converged around the optimal points.

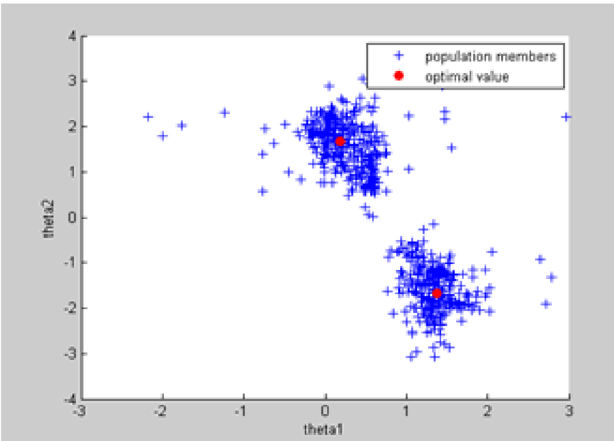


**Figure 8** Population histories at generation 4 (simulation experiment no. 1)

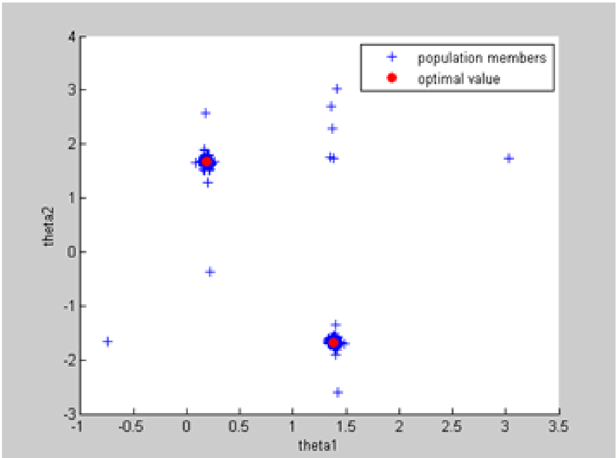


**Figure 9** Population histories at 12th or final generation (simulation experiment no. 1)

## 4.2 Puma manipulator (RRR)

Fitness Function obtained in *eqn.20* is minimized using GA and HGA in MATLAB. In HGA good solutions from GA are taken and further optimized using Nelder mead technique. Control parameters used are GA Population size = 1000, Cross-over probability = 0.9, SBX parameter $\eta$ = 5, Fifty number of experiments were conducted, for different final configuration of the robotic manipulator. The individuals are specified through the x-coordinate y-coordinate and z-coordinate in figures. Termination condition for algorithm is twelve numbers of generations for GA and fitness function value approximately equal to zero for HGA.

The results of a experiments are given in Table 6

From Table 5 and Table 6 we can observe that HGA was able to find values without any error Figures 10and 11 show the distribution of individuals in the population at the initial, final generation of the GA for the simulation experiment one of PUMA robot. The above figures show that a good convergence of the GA population around the optimal points is achieved

**Table 5** Analytical Results of experiments of PUMA robot

| S.NO. | Final position of wrist | Expected values $\theta_1$ | $\theta_2$ | $\theta_3$ |
|---|---|---|---|---|
| | | -23.7 | 167.8 | 45 |
| | | -23.8 | 119.9 | 140.3 |
| | | 130 | 12.1 | 140.3 |
| 1 | (-52.5, 39.3,-46.5) | 130.0 | 60.0 | 44.9 |
| | | -170.7 | -104.0 | -164.8 |
| | | -170.7 | -1.1 | -9.8 |
| | | 58.0 | -178.8 | -164.8 |
| 2 | (-30, -20, 43) | 58.0 | -75.9 | -9.8 |
| | | -68.6 | 155. | 140.3 |
| | | -68.6 | -157 | 45.01 |
| | | 90 | -22.82 | 140.3 |
| 3 | (-15,79, -1.5) | 90 | 24.9 | 45.01 |
| | | -150 | -2.8 | 140.3 |
| | | -150 | 45.0 | 44.9 |
| | | 52.8 | -177.1 | 44.9 |
| 4 | (-56.4, 49.7,-28.4) | 52.8 | 134.9 | 140.3 |

**Table 6** Results of experiments performed on PUMA robot using HGA

| S.NO | Final position of wrist | Obtained values $\theta_1$ | $\theta_2$ | $\theta_3$ |
|---|---|---|---|---|
| | | -23.7 | 167.8 | 45 |
| | | -23.8 | 119.9 | 140.3 |
| | | 130 | 12.1 | 140.3 |
| 1 | (-52.5, 39.3,-46.5) | 130.0 | 60.0 | 44.9 |
| | | -170.7 | -104.0 | -164.8 |
| | | -170.7 | -1.1 | -9.8 |
| | | 58.0 | -178.8 | -164.8 |
| 2 | (-30, -20, 43) | 58.0 | -75.9 | -9.8 |
| | | -68.6 | 155. | 140.3 |
| | | -68.6 | -157 | 45.01 |
| | | 90 | -22.82 | 140.3 |
| 3 | (-15,79, -1.5) | 90 | 24.9 | 45.01 |
| | | -150 | -2.8 | 140.3 |
| | | -150 | 45.0 | 44.9 |
| | | 52.8 | -177.1 | 44.9 |
| 4 | (-56.4, 49.7,-28.4) | 52.8 | 134.9 | 140.3 |

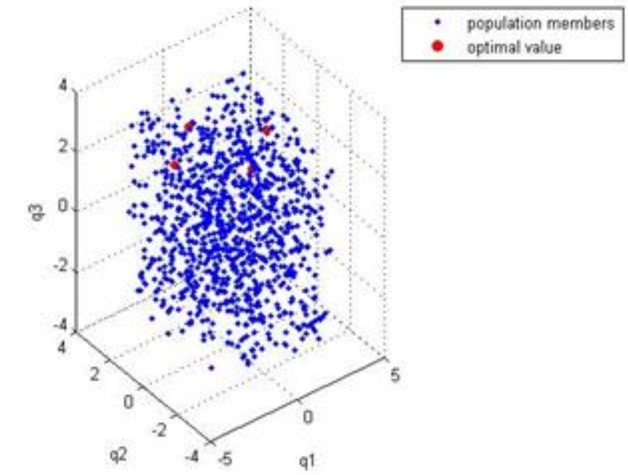


Figure. 10 Population histories at initial generation (simulation experiment no. 1)

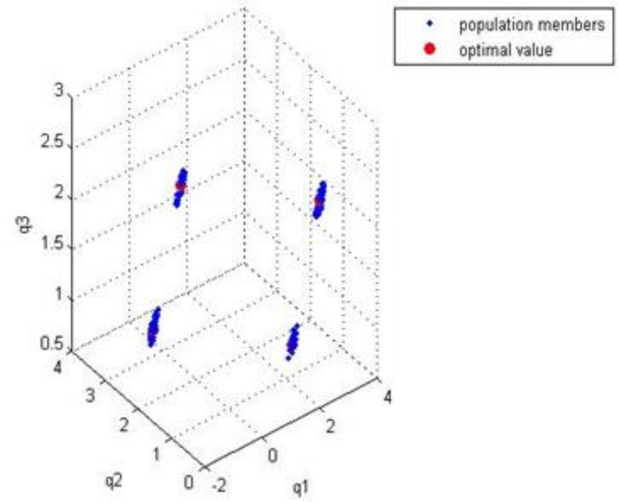


**Figure.11** Population histories at final generation (simulation experiment no. 1)

## 5. Conclusions

A real-coded HGA along with a niching strategy for the binary tournament selection operator was used to evaluate the multiple inverse kinematics solutions of an articulated robotic manipulator and a Puma manipulator. GA is used to explore the whole search space to find the optimal regions and Nelder-Mead algorithm is used to improve solutions obtained from GA. A good le distribution of population converged around the multiple inverse kinematics solutions. Therefore proposed algorithms can be used for obtaining the multiple inverse kinematics solutions of any robotic manipulator

By using GA the relative errors ranged from minimum of 0.439 degrees to a maximum of 3.602 degrees. By using HGA the relative error ranged from minimum of 0 degrees to maximum of 0.001 degrees. So relative error obtained using HGA is less compared to GA. Hybrid GA's helps in fast convergence and to get accurate solutions

To incorporate a spherical wrist in the problem, the positioning part of inverse kinematic problem can be solved with the proposed algorithm. Then the joint variables of the spherical wrist can be solved by using analytical methods , to set the orientation of the end-effector to the desired values.

## 6. Suggestions for future work

Although the proposed approach has been verified for a two link Elbow manipulator and PUMA manipulator it should be investigated further for some more complex configurations such as greater than three DOF manipulators.

The presented algorithm is not able to find multiple solutions if population size is decreased by more than a specific value So, the Algorithm can be made faster by decreasing the population size.

The proposed algorithm can also be developed to use in conjunction of optimization measures like obstacle avoidance, joint singularity avoidance, minimum joint displacement etc

The Algorithm can be made faster by integrating the solution obtained in GA with other surrogate techniques like FUZZY LOGIC, Neural networks.

## References

[1] P.Karla , P.B.Mahapatra and D.K.Aggarwal, “On the solution of Multimodal Robot Inverse Kinematic Function using Real-coded GeneticAlgorithms”, IEEE Int. Conf.on Systems, Man and Cybernetics, Vol.2,pp.1840–1845,2003
[2] Saleh Tabandeh,Christopher Clark,and William Melek, “A Genetic Algorithm Approach to solve for Multiple Solutions of Inverse Kinematics using Adaptive Niching and Clustering”
[3] A.A.Goldenberg, B.Benhabiband G.Fenton, “A Complete Generalized Solution to the Inverse Kinematics of Robots”, IEEEJ of Robotics and Automation, Vol.RA-1,No.1,March1985
[4] C. S. G. Lee and M. Ziegler, "A Geometric Approach in Solving the Inverse Kinematics of PUMA Robots," IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-20, No. 6, pp. 695-706, November 1984
[5] Hua Cai, Sheng Li, Weili Hu,“ A Subsection Algorithm for the Inverse Kinematics of the Manipulators based on the Simplex Method,” IEEE Int. Conf.on Intelligent Systems Design and Applications, Vol.2,pp.72-77,2006
[6] L.Tsai and A.Morgan,“Solving the kinematics of the most general six-and five-degree-of-freedom manipulators by continuation methods”,Journal of Mechanisms,Transmissions and Automation in Design,Vol.107,pp.189–200,1985
[7] J.K.Parker, A.R.Khoogar, D.E.Goldberg, “Inverse kinematics of redundant robots using genetic algorithm”, IEEE Int.Conf.on Robotics and Cybernetics,Vol.1,pp.271–276,1989.
[8] A.C. Nearchou, “Solving the inverse kinematics problem of redundant robots operating in complex environments via a modified genetic algorithm”, Mechanism and Machine Theory, Vol. 33, No. 3, pp. 273-292, 1998
[9] Rachid Chelouah, Patrick Siarry, “Genetic and Nelder-Mead algorithms hybridized for a more accurate global optimization of continuous multi-minima functions”, European Journal of Operational Research 148 ,2003, pp. 335-348.
[10] C.K. Oei, D.E. Goldberg and S.J. Chang, “Tournament selection, niching and the preservation of diversity”, IlliGAL report no. 91011. Urbana, IL: Universiv of IUinois 01 Urbana-Champoign, 1991.
[11] A.A.Goldenberg and D.L.Lawrence, “A generalized solution to the inverse kinematics of robotics manipulators”, ASMEJ. Dynamic Syst.,Meas.,Contr.,Vol.107, pp.103–106, Mar.1985.
[12] P.Karla, P.B.Mahapatra and D.K.Aggarwal,“On the Comparison of Niching Strategies for finding the Solution of Multimodal Robot InverseKinematics”, IEEE Int.Conf.on Systems, Man and Cybernetics,Vol.6,pp.5356–5361,2004.
[13] K.Deb and R.B.Agrawal,“Simulated binary crossover for continuous search space”, Complex- systems,Vol.9,No.2,pp.115–148,1995.
[14] Deb S.R. “Robotics technology and flexible automation”, Tata McGrow-Hill Publishing Company Limited. New-Delhi, 2008.
[15] Clarke, R. “Asimov’s Laws of Robotics: Implications for Information Technology- part II”,
[16] Computer, 27(1), September (1994): pp.57–66.
[17] K. S. Fu, R. C. Gonzalez and C. S. G. Lee, “Robotics – Control, Sensing, Vision and Intelligence.” McGraw Hill, 1987
[18] W. Khalil and E. Dombre, “Modelling, Identification & Control of Robots.” ,Hermes Penton Ltd., 2002.