A Hybrid Genetic Algorithm Approach To Solve Inverse Kinematics Of A Mechanical Manipulator

Masood Ahmad, Narender Kumar, Roopa Kumari

Abstract: The forward kinematics is the process of finding the position of end effector using joint angles and inverse kinematics is vice-versa of it. The forward kinematics is straight forward and gives a unique solution while inverse kinematics may lead to multiple solutions or sometimes even no solution. Traditional methods such as graphical, analytical and iterative are quite popular to solve the inverse kinematics but due to their computational complexity, alternative methods using neural network and genetic algorithm are being proposed. This paper proposes a hybridization algorithm to solve the inverse kinematics of a UR5 (six degrees of freedom) mechanical manipulator. The hybridization algorithm proposed in this paper uses global optimization of the genetic algorithm and fast local optimization of the Newton-Raphson method. This paper also presents a simulation of the proposed algorithm on a UR5 mechanical manipulator.

Index Terms: Hybrid Algorithm, Genetic Algorithm, Inverse Kinematics, Mechanical Manipulator, Newton Raphson method, Robotics.

1. INTRODUCTION

This era of automation is a global competition and the robotics industry is one of the drivers. The robotics industry is increasing faster than the predictor's calculation and this increasing industry is leading towards global competition. This competition is pushing the researcher and designer to create a better machine in terms of design, efficiency, and cost. The robotics is a collective field of kinematics, dynamics, trajectory planning, control system, computer vision system, and artificial intelligence. The solution of the inverse kinematics is one of the most fundamental examples in the discussion of improving the traditional method. Kinematics is dealing with the motion of a robot without concerning with the cause of that motion. The manipulator kinematic is mainly about the forward kinematics and the inverse kinematics. The forward kinematics is finding end-effector frame (T) using the given joint variables (θi) where i = 1, 2, 3, etc. whereas inverse kinematics is trying to find joint variables (θi) where i = 1, 2, 3, etc. using end-effector frame (T). The forward kinematics is quite straight forward as it as a unique solution whereas inverse kinematics form a non-linear continuous function which leads to multiple solutions or sometimes even no solution. The end-effector frame is usually denoted by the 4 x 4 matrix called transformation matrix and joint variables are usually denoted by joint angle for revolute joint and sliding plates distance for the prismatic joint.

The basic approach to solve inverse kinematic is either iterative which uses J-1 for finding inverse kinematics or closed form solution. Traditionally geometric [3][4] and algebraic [4] interpretation methods are used for finding inverse kinematics but due to their high analytic, iterative method comes into existence. The iterative methods have high computation and usually, fall for local optima. The iterative methods are replaced by artificial neural networks. A neural network uses layers and weight associated with it to find the best possible solutions [5][6][44]. Tejomurtula and Kak [6] uses structured neural networks to find the solution of inverse kinematics of a three-joint mechanical manipulator. R.V. Mayorga [7] presented inverse kinematics solution and singularity avoidance using novel neural networks. A neural network is quite fast but it lacks the power to converge at a global level. The genetic algorithm is quite popular for finding global optimization and hence many researchers have presented papers for solving the inverse kinematics using genetic algorithms [9][11][12][15][19][20]. J. Ramírez A. and A. Rubiano F. [16] presented a paper for solving inverse kinematics using a genetic algorithm for 3R mechanical manipulator using the classical homogeneous transformation matrices and the Denavit - Hartenberg parameters. B. Durmus, H.Temurtas and A. Gun [9] presented inverse kinematics solution by using particle swarm optimization. F.Y.C. Albert, S.P. Koh, and S.K. Tiong [11] use a genetic algorithm to solve the real-time computation of inverse kinematics. The genetic algorithms have a slow optimization which makes it hard to use in real time application such as finding the end-effector position of a mechanical manipulator. Many papers also present an inverse kinematic using other evolutionary algorithms such as Bee optimization algorithm [14]. D.T. Pham, M. Castellani, and A. A. Fahmy [14] uses the robustness of bee algorithm to optimize inverse kinematics in the neighbor. Many papers present method that combine the advantages of both neural and genetic algorithm, the fast convergence of artificial neural network and global optimization of the genetic algorithm. This method is commonly known as called hybridization technique [1][2]. Rasit Kolker [1] uses the Elman neural networks for finding the initial solution to reach close to a precise solution and then uses the evolutionary technique to reach a precise solution. The floating-point portion of the neural network result is used as the initial population of the genetic algorithm. Xiulan Wen, Danghong Sheng and Jiacai Huang [2] presented a hybridization algorithm which uses particle swarm optimization [PSO] and Genetic Algorithm. The particle swarm optimization is another popular evolutionary

- *Masood Ahamad, Student of B.Tech., HNB Garhwal University, Srinagar Garhwal Uttarakhand, India. E-mail: dudemessi1@gmail.com*
- *Narender Kumar, Assistant Professor, HNB Garhwal University, Srinagar Garhwal Uttarakhand, India. E-mail: narenrawal@gmail.com*
- *Roopa Kumari, Research scholar of Gurukul Kangri Vishvidhalaya Haridwar, Uttrakhand, India. E-mail:rooparawal@gmail.com*

algorithm which uses a swarm of the particle to search the optimal solution. Sebastian Starke, Norman Hendrich, Sven Magg, and Jianwei Zhang [8] presented a genetic algorithm with evolutionary optimization. Many papers also presented for parallel robotics which shows working in a distributed system. Sergiu-Dan Stan [10] presented a paper showing optimization of Bipod micro (2 DOF), uses the robust nature of the genetic algorithm and applies it to the multi-criteria optimization problem. Prashant Kumar Jamwal, Shengquan Xie, Kean C. Aw [13] presented a paper explaining a genetic algorithm for the rehabilitation of an ankle joint. The design uses soft parallel computing robot for rehabilitation which provides lightweight to the robot.

$$M = \begin{bmatrix} -1 & 0 & 0 & L_1 + L_2 \\ 0 & 0 & 1 & W_1 + W_2 \\ 0 & 1 & 0 & H_1 - H_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

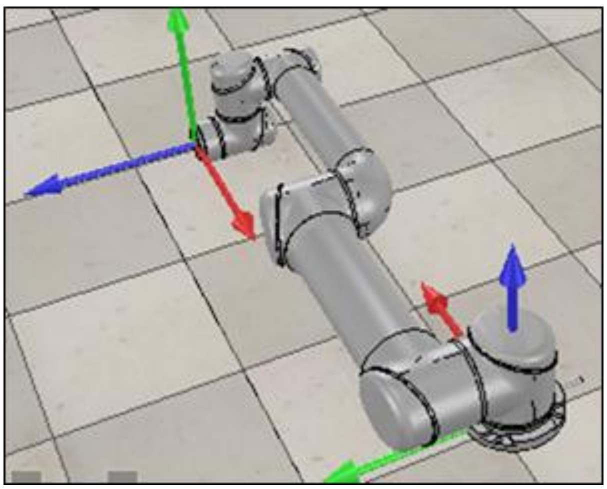

Figure 1 – Stationary reference frame

This paper proposes a hybrid algorithm to solve inverse kinematics and the proposed algorithm is simulated on UR5 mechanical manipulator. The algorithm uses fast optimization of a Newton-Rapshon method and global optimization of a genetic algorithm. The paper contains five sections. Section 2 contains a system model, section 3 contains hybrid algorithm, section 4 contains simulation results and at last section 5 conclude this paper.

2. System Model

2.1 UR5 Manipulator Kinematics

The UR5 is a six degree of freedom (6 DOF) mechanical manipulator designed by Universal Robot which is widely used in industries because it is easily programmable, fast setup and safe in use. The forward kinematics of UR5 mechanical manipulator referred to the computation of orientation and position of the end-effector frame using joint coordinates, θ. The classical method uses Denavit-Hartenberg parameters (D-H parameter) for finding forward kinematics. The D-H parameter uses an implicit parameter or a minimum number of parameters to describe the forward kinematics. This paper uses the product of exponentials (PoE) formula [7], which uses an explicit parameter for solving forward kinematics. The product of exponentials (PoE) formula uses a stationary reference frame M ∈ SE (3), a robot at its zero configuration. Each joint axis of a UR5 mechanical manipulator contributes in one screw axis. The rotation and translation from one frame to another frame about a screw axis is called twist about that axis. The screw axis [7] Bi = (ωi, vi) with respect to body frame is as follow-

Table 1 – A list of screw axes with respect to the body frame

i	*ωi*	*Vi*
1	(0,1,0)	(W1+W2, 0, L1+L2)
2	(0,0,1)	(H2, -L1-L2, 0)
3	(0,0,1)	(H2,-L2 , 0)
4	(0,0,1)	(H2, 0, 0)
5	(0,-1,0)	(-W2, 0, 0)
6	(0,0,1)	(0, 0, 0)

Given the stationary reference frame *M*, screw axes with respect to body frame $B_i = (\omega_i, v_i)$, the desired end-effector configuration with respect to body frame can be calculated using the given equation. The inverse kinematics of a UR5 manipulator referred to the computation of joint coordinates $\theta \in R^n$ using end-effector frame T ∈*SE (3).* The inverse kinematics form non-linear continuous function which leads to multiple solutions sometimes even no solution. The inverse kinematics can be solved with the help of using a finite number of ways. The popular methods are either using an analytical approach or an iterative approach. This paper uses the Newton-Raphson method with a genetic algorithm to optimize at a global level.

$$T(\theta) = M e^{[\mathcal{B}_1]\theta_1} e^{[\mathcal{B}_2]\theta_2} \cdots e^{[\mathcal{B}_6]\theta_6}$$

2.2 Newton-Raphson Method

The Newton-Raphson is a root-finding algorithm which produces the result by continuous convergence to root. The single-variable function f uses the Newton-Raphson method as

$$X_{n+1} = X_n - f(X_n) / f'(X_n) \qquad \text{.......... (1)}$$

The equation 1 is used to predict the next possible root Xn+1 using present root Xn of a single-variable function. The inverse kinematics produces a non-linear system of equations and the Newton-Raphson method for non-linear systems of equations having non-linear continuous function f: Rk→Rk form a k x k Jacobian matrix Jf (Xn). The solutions for non-linear differential equations are as follow

$$X_{n+1} = X_n - J_f(X_n)^{-1} f(X_n) \qquad \text{......... (2)}$$

2.3 Genetic Algorithm

A genetic algorithm is a metaheuristic approach inspired by the process of natural selection. It is a subclass of evolutionary algorithms and commonly used for global optimization and search problem. In 1960, John Holland introduces genetic algorithms on the basis of Darwin's theory evolution and later his student David E. Goldberg extended his theory to various applications. The genetic algorithm uses the processes inspired by natural behavior such as selection, mutation, crossover, and survival of the fittest. The initial population is selected from a pool of vast population and a fitness value is associated with each chromosome of the population. On the basis of their fitness values, best out of the population will be taken as parents to breed new generations. The parents' pass-through crossover, mutation, and other natural selection processes. This process will repeat until the optimal solution is achieved.

Initialization

The initial population depends on the nature of the problem and commonly selected from the pool of large population size. The process of initial population selection is through a uniform random distribution and each chromosome has fitness value associated with it.

Selection

The selection is the process of choosing chromosomes having the lowest fitness value from the population. The fitness function provides fitness value to each chromosome and on the basis of these fitness value, the best possible chromosome is selected for the breeding of the next population. These selected chromosomes are known as the parent for the next generation.

Genetic Operators

The next step in the production of the new generation is breeding. The breeding can be achieved by genetic operators and the most popular genetic operators are crossover and mutation. The crossover again is achieved by many possible ways such as taking half of the digit from one parent and another half from other parent or selecting alternative genes from each parent, etc. A crossover is followed by mutation and the mutation operator provides variation in the next generation. It also prevents the algorithm from converging at local minima. The mutation rate is defined as the ratio of randomly changing digit in a chromosome to the total digit in a chromosome. At the high percentage of mutation, the algorithm may work similarly brute force algorithm and sometimes even impossible to converge. The mutation rate is usually kept low. The new generation is replaced by the old generation and this process continues until the desired solution is achieved.

3. Proposed Hybrid Algorithm

This paper presents a hybridization approach which uses global optimization of the genetic algorithm and quick local optimization of the Newton-Raphson. The algorithm initializes the joint angle population through uniform random distribution between $-\pi$ *to* π. The initial population is passed through fitness function and fitness function assigns a fitness value to each chromosome. The fitness function in this algorithm is the difference between the desired configuration and configuration calculated using forward kinematics of a chromosome.

T_{sd} - $T_{sb}(\theta i)$ ……… (3)

The difference in the transformation matrix can't be calculated by using the difference operator. The transformation of a configuration from T_{sd} to $T_{sb}(\theta i)$ is used to calculate the desired difference.

$T_{bd}(\theta i) = T_{sb}\text{-}1(\theta i)\, T_{sd} = T_{bs}(\theta i)\, T_{sd}$ ……… (4)

The transformation matrix is converted into twist using matrix logarithms [17]

$[V_b] = \log T_{bd}(\theta i)$ ……… (5)

The equation - 5 gives twist which is required to change the configuration from T_{sd} to $T_{sb}(\theta i)$ and norm of this twist is the fitness value of the proposed algorithm. Half of the population has the best fitness value is taken as a parent for the next population and this selected chromosome passes through the Newton-Raphson method for fast convergence. The figure - 3 (flows chart) explains working of the Newton-Raphson method. The chromosome of a population is a list of joint angles. These joint angles can provide an end-effector frame of a manipulator using forward kinematics.

Using equation – 5, twist required to change the configuration

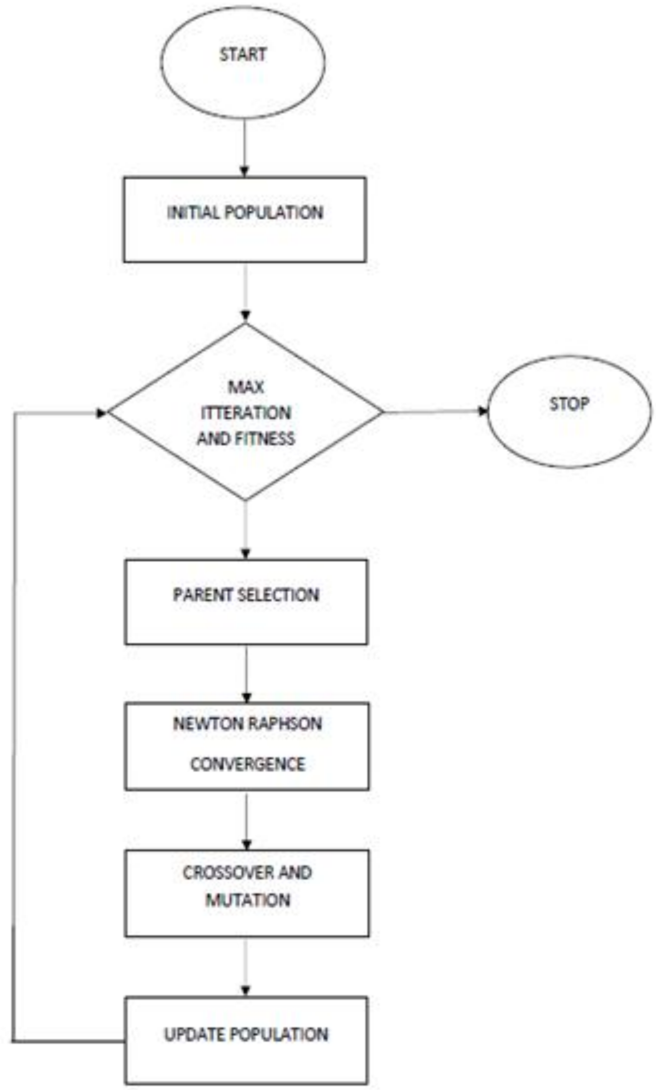

Figure 2: Hybrid Algorithm

is computed (shown in the flow chart). Putting twist into the Newton-Raphson method and new pair of twist joint angles are calculated. This process continues for a predefined number of times (maximum iteration) and converges each time. The number of iteration should not be too high as it may lead to local optimal or not too low as it may not affect joint angles at all. The Newton - Raphson method updates parent chromosomes closer to the solutions. The process of breeding a new offspring by selecting two random chromosomes out of the selected population is called crossover. The crossover is popularly known as recombination and it can be achieved by different processes. In this paper, the crossover point is selected at half of the length of joint angles and a new offspring is created by taking the first half from parent 1 and second half from parent 2. This process continues until the required population of offspring is generated. As flowchart shows new offsprings are created using

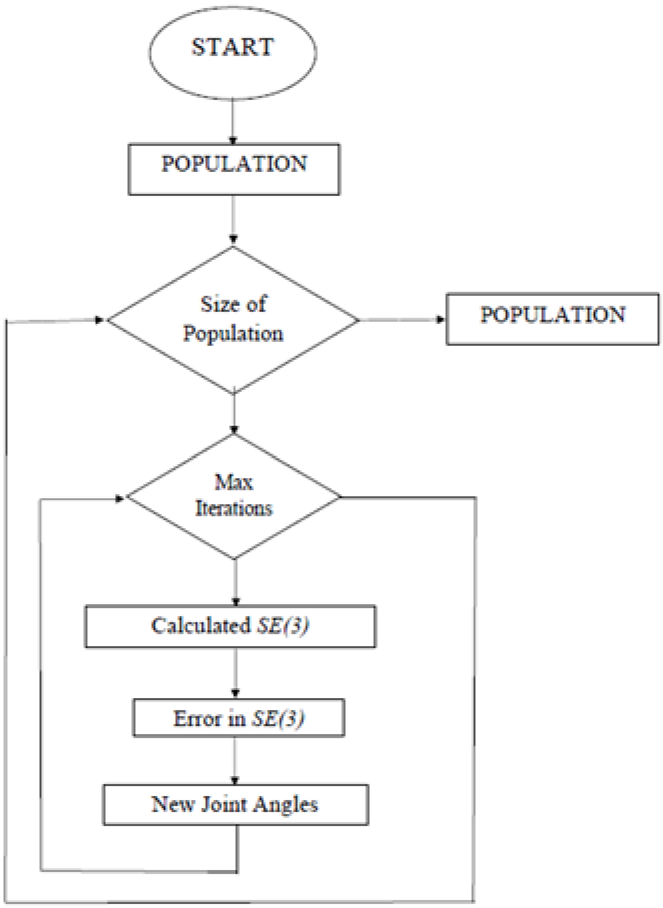

Figure 3: Newton Raphson method (SE(3)-transformation matrix)

parents. The crossover is followed by mutation and in the mutation, a couple of digits are randomly changed from a list of joint angles. After mutation, a new population is updated with an old population. The new population consists of parents and offsprings combined together. The whole process repeats until fitness value does not reach to the desired configuration.

4. Simulation Results

In this section, we evaluate the performance of the proposed hybrid algorithm. This paper uses V-REP PRO EDU simulator of a UR5 mechanical manipulator. Given W1 = 109 mm, W2 = 82 mm, L1 = 425 mm, L2 = 392 mm, H1 = 89 mm, H2 = 95 mm. The desired configuration of the manipulator is,

```
[[ 0.    1.    0.   -0.5]
 [ 0.    0.   -1.    0.1]
 [-1.    0.    0.    0.1]
 [ 0.    0.    0.    1. ]]
```

```
[[-1.     0.     0.     0.817]
 [ 0.     0.     1.     0.191]
 [ 0.     1.     0.    -0.006]
 [ 0.     0.     0.     1.   ]]
```

```
[[ 0.     1.     0.     0.191  0.     0.817]
 [ 0.     0.     1.     0.095 -0.817  0.   ]
 [ 0.     0.     1.     0.095 -0.392  0.   ]
 [ 0.     0.     1.     0.095  0.     0.   ]
 [ 0.    -1.     0.    -0.082  0.     0.   ]
 [ 0.     0.     1.     0.     0.     0.   ]]
```

Current SE(3):

T(θ) =

```
| -0.000 +1.000 +0.000 -0.500 |
| +0.000 +0.000 -1.000 +0.100 |
| -1.000 -0.000 -0.000 +0.100 |
| +0.000 +0.000 +0.000 +1.000 |
```

The stationary reference frame M ∈ SE (3), The screw axis of a UR5 manipulator with respect to body frame is given as, The initial random population of joint angles between ($-\pi$ *to* π) of size 10, 20 and 30 is discussed in this paper. The parents are selected using fitness values. These set of parents processed through Netwon-Rapshon iterations which lead to converging at a local optimum. This process is repeated until the new population is generated. Each joint angle of a chromosome undergoes through the crossover to generate a new joint angle of chromosome and hence new generation.

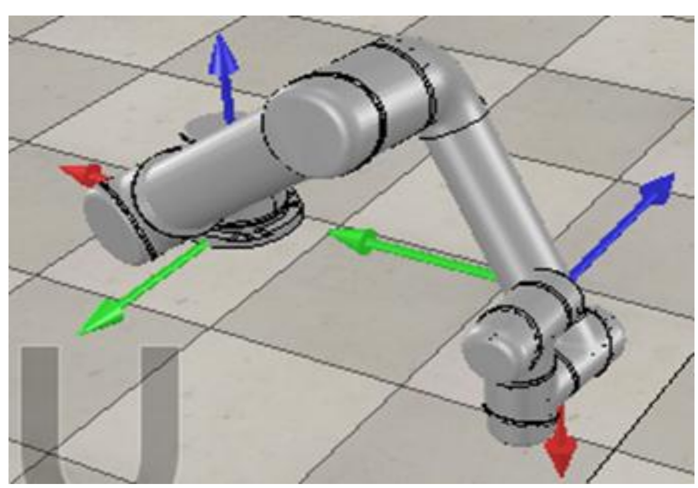

Figure 4: Crossover of a joint angle

The crossover is followed by mutation for each joint angle of the chromosome.

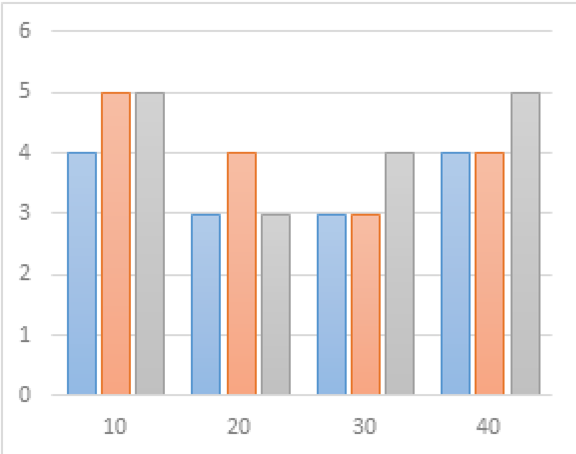

Figure 5 – Mutation on a joint angle

The process repeats until the new generation is generated and further until computed chromosome value does not reach to the desired configuration whole process repeats itself. The computed joint angles using the hybrid algorithm, This result is verified using forward kinematics of UR5 manipulator. The effector configuration using obtained angles is shown in figure 6. The desired configuration is the same as computed configuration. The result of this paper presents a comparison between a couple of mutation rates, fitness values on the population of size 10 and 20.

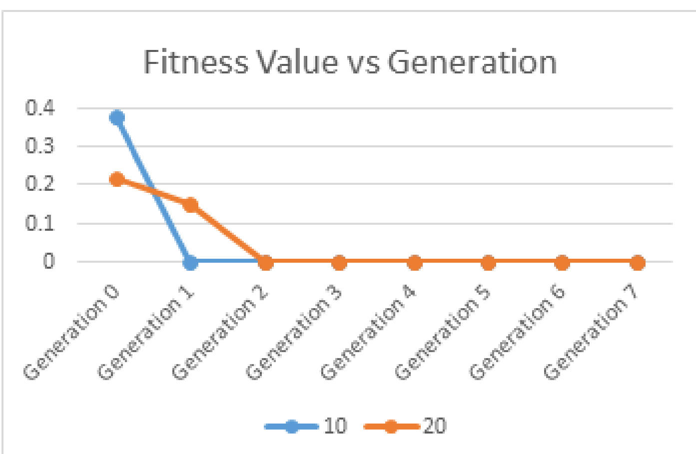

Figure 6: Computed Configuration (forward kinematics) and simulated result.

The figure 7 presents mutation of one digit in a joint angle and each mutation rate has three samples (Blue, Red, and Purple). The population size of 10 is fast but the population size of 20 optimizes before size of 10. To analysis, the performance between the population of size 10 and size 20 graphical comparison and tabular comparison is used.

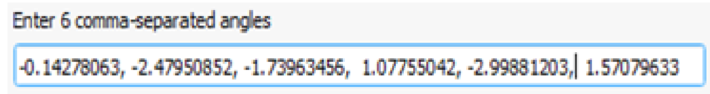

Figure 7: Mutation of one digit in a joint angle No. of generation Vs Population Size

The graphical representation of table 2 is shown in figure 8. It clearly shows that the population of size 10 converge faster as compared to the population of size 20. The population of size 20 does have low initial fitness value but the population of size 10 converges faster. This is quite obvious since a number of chromosomes in the population of size 20 is greater hence its initial search result is better while its computation and convergence are slow for preceding generations. On the other hand, the population of size 10 having fewer chromosomes hence its initial search fitness value is low but its convergence and computation is fast for preceding generations.

5. Conclusion and Future Work

This paper presents an algorithm which uses the Newton-Raphson method and genetic algorithm to solve inverse kinematics. Inverse kinematics is a non-linear continuous function which leads to multiple solutions or some time no solution at all. This paper presents a hybrid algorithm to solve this problem. This paper presents a simulation for computing joint angles of a UR5 manipulator using the proposed hybrid algorithm. The Hybrid algorithm converges to a solution within a couple of generations. This paper also provides a discussion on the effect of size of the population and mutation rate. This concludes the hybridization algorithm on a UR5 mechanical manipulator. This algorithm can also be used to find the intermediate configuration required by the manipulator to reach its final configuration. This can be used for obstacle avoidance. This work is for future work.

References

[1]. RasitKolker, "A genetic algorithm approach to neural-network-based inverse kinematics solution of robotic manipulators based on error minimization", Information Science 222 (2013) 528-543.

[2]. Xiulan Wen, Danghong Sheng, and Jiacai Huang, "A Hybrid Particle Swarm Optimization for Manipulator Inverse Kinematics Control", Springer-Verlag Berlin Heidelberg ICIC 2008, LNCS 5226, pp. 784–791, 2008.

[3]. R. Featherstone, "Position and velocity transformation between robot end-effector coordinate and joint angle", International Journal of Robotics Research 2 (2) (1983) 33–45.

[4]. John J. Craig, "Introduction to Robotics Mechanics and Control (3rd Edition)", Pearson Education, Inc. (2005).

[5]. Z. Bingul, H.M. Ertunc, Applying neural network to inverse kinematics problem for 6R robot manipulator with offset wrist, in Proceedings, 7th International Conference on Adaptive and Natural Computing Algorithms, Coimbra, Portugal, March 2005.

[6]. S. Tejomurtula, S. Kak, "Inverse kinematics in robotics using neural networks", Information Sciences 116 (1999) 147–164.

[7]. R.V. Mayorga, P. Sanongboon, 'Inverse kinematics and geometrically bounded singularities prevention of redundant manipulators: an artificial neural network approach", Robotics and Autonomous Systems 53 (2005) 164–176.

[8]. Sebastian Starke, Norman Hendrich, Sven Magg, Jianwei Zhang, "An Efficient Hybridization of Genetic Algorithms and Particle Swarm Optimization for Inverse Kinematics", 2016 IEEE International Conference on Robotics and Biomimetics Qingdao, China, December 3-7, 2016.

[9]. B. Durmus, H.Temurtas, and A. Gun, "An Inverse Kinematics Solution using Particle Swarm Optimization", 6th International Advanced Technologies Symposium (IATS'11), 16-18 May 2011, Elazığ, Turkey.

[10]. Sergiu-Dan Stan, "Genetic Algorithms Multiobjective Optimization of a 2 DOF Micro Parallel Robot", 2007 IEEE International Symposium on Computational Intelligence in Robotics and Automation Jacksonville, FL, USA, June 20-23, 2007.

[11]. F.Y.C. Albert, S.P. Koh, S.K. Tiong, C.P. Chen, F.W. Yap, "Inverse Kinematic Solution in Handling 3R Manipulator via Real-Time Genetic Algorithm", 978-1-4244-2328-6/08, 2008 IEEE.

[12]. Nizar Rokbani, Adel.MAlimi, "Inverse Kinematics Using Particle Swarm Optimization, A Statistical Analysis", ScienceDirect, Procedia Engineering 64 (2013) 1602 – 1611.

[13]. Prashant Kumar Jamwal, ShengquanXie, Kean C. Aw, "Kinematic design optimization of a parallel ankle rehabilitation robot using modified genetic algorithm", Robotics and Autonomous Systems 57 (2009) 1018-1027.

[14]. D.T. Pham, M. Castellani, and A. A. Fahmy, "Learning the Inverse Kinematics of a Robot Manipulator using the Bees Algorithm", IEEE International Conference on Industrial Informatics (INDIN 2008) DCC, Daejeon, Korea July 13-16, 2008.

[15]. BahaaIbraheemKazem, Ali Ibrahim Mahdi, Ali TalibOudah "Motion Planning for a Robot Arm by Using Genetic Algorithm", JJMIE Volume 2, Number 3, Sep. 2008 ISSN 1995-6665 Pages 131 – 136.

[16]. J. Ramírez A., and A. Rubiano F., "Optimization of Inverse Kinematics of a 3R Robotic Manipulator using Genetic Algorithms", World Academy of Science, Engineering and Technology International Journal of Mechanical, Aerospace, Industrial, Mechatronic, and Manufacturing Engineering Vol:5, No:11, 2011.

[17]. Kevin M. Lynch and Frank C. Park, "Modern Robotics Mechanics, Planning and Control", Cambridge U. Press, 2017.

[18]. V. K. Banga, Y. Singh, and R. Kumar, "Simulation of Robotic Arm using Genetic Algorithm & AHP", World Academy of Science, Engineering and Technology International Journal of Electrical and Computer Engineering Vol:1, No:1, 2007.

[19]. F.Y.C. Albert, S.P. Koh, C.P. Chen, S.K.Tiong, S.Y.S. Edwin, "Optimizing Joint Angles of Robotic Manipulator using Genetic Algorithm", 2009 International Conference on Computer Engineering and Applications IPCSIT vol.2 (2011) IACSIT Press, Singapore.

[20]. Shaher Momani, Zaer S. Abo-Hammour and Othman MK. Alsmadi, "Solution of Inverse Kinematics Problem using Genetic Algorithms", Appl. Math. Inf. Sci. 10, No. 1, 1-9 (2016).