We are IntechOpen, the world’s leading publisher of Open Access books
Built by scientists, for scientists

4,300 Open access books available
116,000 International authors and editors
130M Downloads

154 Countries delivered to
top 1% Our authors are among the most cited scientists
12.2% Contributors from top 500 universities

WEB OF SCIENCE™
Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com
Abstract

Inverse kinematics of serial or parallel manipulators can be computed from given Cartesian position and orientation of end effector and reverse of this would yield forward kinematics. Which is nothing but finding out end effector coordinates and angles from given joint angles. Forward kinematics of serial manipulators gives exact solution while inverse kinematics yields number of solutions. The complexity of inverse kinematic solution arises with the increment of degrees of freedom. Therefore it would be desired to adopt optimization techniques. Although the optimization techniques gives number of solution for inverse kinematics problem but it converges the best solution for the minimum function value. The selection of suitable optimization method will provides the global optimization solution, therefore, in this paper proposes quaternion derivation for 5R manipulator inverse kinematic solution which is later compared with teachers learner based optimization (TLBO) and genetic algorithm (GA) for the optimum convergence rate of inverse kinematic solution. An investigation has been made on the accuracies of adopted techniques and total computational time for inverse kinematic evaluations. It is found that TLBO is performing better as compared GA on the basis of fitness function and quaternion algebra gives better computational cost.

Keywords: TLBO, GA, quaternion, kinematics

1. Introduction

Kinematic chain may consist of rigid/flexible links which are connected with joints or kinematic pair permitting relative motion of the connected bodies. In case of manipulator kinematics it can be categorized into forward and inverse kinematics. Forward kinematics for any serial manipulator is easy and mathematically simple to resolve but in case of inverse kinematics there is no unique solution, generally inverse kinematics gives multiple solutions. Hence, inverse kinematics solution is very much problematic and computationally expensive. For real time control of any configuration manipulator will be expensive and generally it takes long
time. Forward kinematics of any manipulator can be understood with translation of position and orientation of end effector from joint space to Cartesian space and opposite of this is known as inverse kinematics. It is essential to calculate preferred joint angles so that the end effector can reach to the desired position and also for designing of the manipulator. Various industrial applications are based on inverse kinematics solutions. In real time environment it is obvious to have joint variables for fast transformation of end effector. For any configuration of industrial robot manipulator for \( n \) number of joints the forward kinematics will be given by,

\[
y(t) = f(\theta(t))
\]

where \( \theta_i = \theta(t) \), \( i = 1, 2, 3, \ldots, n \) and position variables by \( y_j = y(t) \), \( j = 1, 2, 3, \ldots, m \).

Inverse kinematics for \( n \) number of joints can be computed as,

\[
\theta(t) = f^{-1}(y(t))
\]

Inverse kinematics solution of robot manipulators has been considered and developed different solution scheme in last recent year because of their multiple, nonlinear and uncertain solutions. There are different methodologies for solving inverse kinematics for example iterative, algebraic and geometric etc. [8] proposed inverse kinematic solution on the basis of quaternion transformation. [20–36] have proposed application of quaternion algebra for the solution of inverse kinematics problem of different configurations of robot manipulator. [35] presented a quaternion method for the demonstrating kinematics and dynamics of rigid multi-body systems. [34] presented analytical solution of 5-dof manipulator considering singularity analysis. [11] presented quaternion based kinematics and dynamics solution of flexible manipulator. [14] proposed detailed derivation of inverse kinematics using exponential rotational matrices. On the other hand, after numerous surveys on conventional analytical and other Jacobian based inverse kinematics are quite complex as well as computationally exhaustive those are not exactly well suitable for the real time applications. Because of the aforementioned reasons, various authors adopted optimization based inverse kinematic solution.

Optimization techniques are fruitful to solve inverse kinematics problem for different configurations of manipulator as well as spatial mechanisms. Conventional approaches such as Newton-Raphson can be used for nonlinear kinematic problems and predictor corrector type methods can compute differential problem of manipulator. But major drawback of these methods are Singularity or ill condition which converse to local solutions. Moreover, when initial guessing is not accurate then the method becomes unstable and does not converge to optimum solution. Therefore, recently developed metaheuristic techniques can be used to overcome the conventional optimization drawbacks. Literature survey shows the efficiency of these metaheuristic algorithms or bi-inspired optimization techniques are more convenient to achieve global optimum solutions. The major issue with these nature inspired algorithms is framing of objective function. Even these algorithms are direct search algorithms which do not require any gradient or differentiation of objective function. The comparison of the metaheuristic algorithm with heuristic algorithms is based on the convergence rate as it has been proved that the convergence of heuristic-based techniques is slower. Therefore, to adopt metaheuristic
techniques such as GA, BBO, teachers learner based optimization (TLBO), ABC, ACO etc. will be suitable for enhancing the convergence rate and yielding global solution. From literature survey the teaching learning based optimization (TLBO) is similar to swarm based optimization in which the impact of learning methods from teacher to student and student to student has been highlighted. Wherein, the population or swarm is represented by group of students and these students gain knowledge from either teacher or students. If these student gain knowledge from teacher then it is called as teachers phase similarly when students learns form student then it is student phase. The output is considered as result or grades of students. Therefore, number for number of subjects resembles the variables of the function and grades or results gives fitness value, [5, 6]. There are numerous other population centered methods which have been effectively applied and shown efficiency [33]. However, all algorithms are not suitable for complex problem as proved by Wolpert and Macready. On the other hand, evolutionary strategy (ES) based methods such as GA, BBO etc. gives better results for various problems and these methods are also population based metaheuristic [16, 28]. Moreover [22] proposed inverse kinematic solution of redundant manipulator using modified genetic algorithm considering joint displacement (\(\Delta \theta\)) error minimization and the positional error of end effector. [32] proposed inverse kinematic solution of PUMA 560 robot using cyclic coordinate descent (CCD) and Broyden-Fletcher-Shanno (BFS) technique. [23] proposed IK solution of 4-dof PUMA manipulator using genetic algorithm. This paper uses two different objective functions which are based on end-effector displacement and joint variable rotations. [18] proposed inverse kinematic solution of 3-dof manipulator using cyclic coordinate descent (CCD) and Broyden-Fletcher-Shanno (BFS) technique. [21] proposed inverse kinematic solution of 6R revolute manipulator using real time optimization algorithm. [24] proposed inverse kinematic solution of 3-dof redundant manipulator using deterministic global optimization method. [17] proposed optimization of design parameter i.e. link length using for 2-dof manipulator. [15] proposed inverse kinematic solution of 2-dof articulated robot manipulator using real coded genetic algorithm. [19] proposed inverse kinematic solution scheme of 3-dof redundant manipulator based on reach hierarchy method. [30] proposed inverse kinematic solution of 3-dof PUMA manipulator for the major displacement propose. In this work they have adopted genetic algorithm with adaptive niching and clustering. [12] proposed inverse kinematic solution of 6-dof MOTOMAN robot manipulator for positioning of the end-effector. In this work they have adopted adaptive genetic algorithm for optimum placement of the end effector. [26] proposed inverse kinematic and trajectory generation of humanoid arm manipulator using forward recursion with backward cycle computation method. [21] proposed inverse kinematic solution for 6R revolute manipulator using real time optimization algorithm. [24] proposed kinematic solution using three different methods such as bee algorithm, neural network which is later optimized by bee algorithm and evolutionary algorithm. [2] proposed kinematic solution of 3-dof serial robot manipulator using real time genetic algorithm. [13] proposed inverse kinematic solution of 6-dof robot manipulator using immune genetic algorithm. [9] proposed conventional approach i.e. penalty function based optimization method for solving IK. Even though few methods can solve hard NP problems, but it requires high-performance computing system and intricate computer programming.
On the other hand, the use of optimization algorithms is not new in the field of multi-objective and NP-hard problem to arrive at a very reasonable optimized solution, the TLBO algorithm have not been tried to solve an inverse kinematics problems and trajectory of joint variables for robot manipulator. Moreover, computational cost for yielding the inverse kinematics solution with adopted algorithms has been compared without any specialized tuning of concern parameters. Therefore, the key purpose of this work is focused on minimizing the Euclidian distance of end effector position based resolution of inverse kinematics problem with comparison of GA and TLBO obtained solution for 5R robot manipulator. The results of all algorithm are computed from inverse kinematics equations and obtained resultant error for data statistics. In other words, end-effector coordinates utilized as an input for joint angle calculations. At the end 4th order spline formula is considered for generation of end effector trajectory and analogous joint angles of robotic arm using TLBO, GA and quaternion. The sectional organization of the paper henceforth is as follows: Section 2 pertains to the mathematical modeling of the 5R robot manipulator and detail derivation of forward and inverse kinematics of 5R manipulator using quaternion algebra. In Section 3 discuss about the inverse kinematic objective function formulation for 5R manipulator. The experimental results as obtained from simulations are discussed elaborately in Section 5.

2. Quaternion vector approach for mathematical modeling

This section deals with mathematical modeling of quaternion vector algebra and application for the derivation of inverse kinematics equations. Quaternion vector methods are fruitful for both rotation and translation of a point, line, etc. with references to origin coordinate system irrespective of homogeneous transformation matrix. The Interpolation of series of rotations and translations are quite complex using Euler’s angle method. In other words, the variables lies in isotropic space which is nothing but sphere surface topology and complex in nature. A brief formulation of quaternion mathematics is given in this section for assessment of references and to create background for mathematical derivation of inverse kinematic.

2.1. Rotation and translation from quaternion

The above discussions gives the importance of quaternions and the necessity of it. The quaternion rotation and translation are lies in four dimensional space therefore it is quite difficult to represent here or to imagine. Figure 1 represents the rotation through quaternion and Eq. (3) describes rotation of a point in a space mathematically.

\[ h = \cos \left( \frac{\theta}{2} \right) + i^* \sin \left( \frac{\theta}{2} \right) + j^* \sin \left( \frac{\theta}{2} \right) + k^* \sin \left( \frac{\theta}{2} \right) \]  (3)

The 4-dimensional space, imagination of fourth axis is quite complex. Therefore, in Figure 1 a unit distance point around axis (X, Y, and Z) is given and which traces a circle. When this rotation circle is projected on a plane then the point \( P_1 \) can be seen rotated through angle \( \theta \) to point \( P_3 \) which crosses the mid-point \( P_2 \). Therefore \( P_1 \) point is transforming to \( P_3 \) following by straight line makes \( \cos(\theta/2) \) and \( \sin(\theta/2) \).
Now two quaternions can be represented on the basis of above discussed concept. If there is subsequent rotation of two quaternion $h_1$ and $h_2$ then the composite rotations $h_1^{-1}h_2$ can be given by From Figure 1, $p^1$ is the point vector representing initial position and $p^3$ is the point vector final condition to be transformed. Therefore,

$$p^3 = h_2^{-1}(h_1^* p^1 h_1^{-1})^* h_2^{-1}$$
$$= (h_2^* h_1^*)p^1 (h_1^{-1}h_2^{-1})$$
$$= (h_2^* h_1^*)p^1 (h_2^{-1} h_1^{-1})$$

(4)

Now pure translations $t_r$ can be done by quaternion operator that is given below,

$$t_r = h + p^1$$

(5)

Quaternion transform can be given by,

$$p^2 = h^* p^1 h^{-1}$$

(6)

Therefore, an expression for the inverse of a quaternion can be given as,
\[
H^{-1} = [< h^{-1} >, < -h^{-1} \otimes P \otimes h >]
\]  
(7)

where, \(-h^{-1} \otimes P \otimes h = P + [-2k(j \times (-P)) + 2j \times (j \times (-P))]\) quaternion product, which is defined in the most general form for two quaternions \(h_1 = (r_1, j_1)\), and \(h_2 = (r_2, j_2)\) as

\[
h_1 \otimes h_2 = [n_1 r_2 - j_1 \bullet j_2, n_1 r_2 j_1 + r_2 j_1 + j_2 \times j_1]
\]  
(8)

where \(j_1 \bullet j_2\) and \(j_1 \times j_2\) denote dot and cross products, between the 3-dimensional vectors \(j_1\) and \(j_2\). Clearly, quaternion multiplication is not commutative. The set of elements \([\pm 1, \pm i, \pm j, \pm k]\) is known as the quaternion group of order 8 in multiplication.

Similarly vector transformation multiplication can be given as

\[
H_1 \otimes H_2 = (h_1, P^1) \otimes (h_2, P^2) = h_1^* P^2 \ast h_1^{-1} + P^1
\]  
(9)

where, \(h_1^* P^2 \ast h_1^{-1} = P^2 + 2r_1 (j_1 \times P^2) + 2j_1 \times (j_1 \times P^2)\)

### 2.2. Quaternion derivation for 5R manipulator kinematics

The configuration and base coordinate frame attachment of 5R manipulator is given in Figure 2 (a) and MATLAB plot of 5R manipulator is presented in (b). Where \(\theta_1, \theta_2, \theta_3, \theta_4\) and \(\theta_5\) joint angles for are articulated arm and \(d_1, d_2\) and \(d_3\) are the link offset. \(a_1\) and \(a_2\) represents link lengths.

Now quaternion for successive transformation of each joint can be calculated from the Eq. (3) as follows,

\[
H_1 = [< C_1 + S_1 \hat{k} >, < a_1 C_1 \hat{i} + a_1 S_1 \hat{j} + d_1 \hat{k} >]
\]  
(10)

\[
H_2 = [< C_2 + S_2 \hat{j} >, < -a_2 S_2 \hat{i} - a_2 C_2 \hat{k} >]
\]  
(11)

\[
H_3 = [< C_3 + S_3 \hat{j} >, < -d_3 S_3 \hat{i} - d_3 C_3 \hat{k} >]
\]  
(12)

\[
H_4 = [< C_4 + S_4 \hat{i} >, < d_4 \hat{i} >]
\]  
(13)

\[
H_5 = [< C_5 + S_5 \hat{j} >, < -d_5 S_5 \hat{i} - d_5 C_5 \hat{k} >]
\]  
(14)

Inverse of a dual quaternion can be calculated by Eq. (8),

\[
H_1^{-1} = [< C_1 - S_1 \hat{k} >, < -a_1 \hat{i} >]
\]  
(15)

\[
H_2^{-1} = [< C_2 - S_2 \hat{j} >, < a_2 \hat{k} >]
\]  
(16)

\[
H_3^{-1} = [< C_3 - S_3 \hat{j} >, < d_4 \hat{k} >]
\]  
(17)
\[ H_4^{-1} = \left[ \begin{array}{c} <C_4 - S_3i>, < - d_4i > \end{array} \right] \] (18)

\[ H_5^{-1} = \left[ \begin{array}{c} <C_5 - S_5j>, < d_5k > \end{array} \right] \] (19)

\[ Q_i = H_i \otimes H_{i+1} \ldots H_n \] (20)

Where in case of 5R manipulator arm \( n = 5 \). Now calculating quaternion vector products using Eq. (20)

\[ Q_5 = H_5 \left[ \begin{array}{c} <C_5 - S_5j>, < - d_5S_3i - d_5C_5k > \end{array} \right] \]

\[ Q_4 = H_4 \otimes Q_5 = \left[ \begin{array}{c} <C_4 + S_4i>, < d_4i > \end{array} \right] \otimes \left[ \begin{array}{c} <C_5 + S_5j >, < - d_6S_3i - d_6C_5k > \end{array} \right] \] (21)

\[ Q_4 = \left[ \begin{array}{c} <C_4C_5 + S_4S_3i + C_4S_5j + S_4C_5k >, < (d_4 - d_5S_3i) + d_6C_5S_4j - d_6C_4C_5k > \end{array} \right] \] (22)

\[ Q_3 = H_3 \otimes Q_4 = \left[ \begin{array}{c} <C_3 + S_3j >, < - d_4S_3i - d_4C_3k > \end{array} \right] \otimes \left[ \begin{array}{c} <C_4C_5 + S_4S_3i + C_4S_5j + S_4C_5k >, < (d_4 - d_5S_3i) + d_6C_5S_4j - d_6C_4C_5k > \end{array} \right] \] (23)

\[ Q_3 = \left[ \begin{array}{c} <C_4C_3 + S_4S_3S_5i + C_4S_3S_5j - S_4S_3S_5k >, < (d_4 + d_4 - d_5C_4C_5S_3 - d_4C_3 - d_6S_3C_3 - d_4S_3)i + (d_6C_5S_4j + (-d_4S_3 + d_6S_3S_3 - d_6C_4C_3C_3 - d_4C_3)k > \end{array} \right] \] (24)

Figure 2. (a) Base frame and model of 5R manipulator; (b) configuration of 5R manipulator.
Therefore, 

\[
Q_2 = \left[ < (C_2 S_4 C_{3,5} - S_2 S_4 S_{3,5}) + (C_2 S_4 C_{3,5} - S_2 S_4 S_{3,5}) i >, \right.

\]

\[
+ (C_2 S_4 C_{3,5} + S_2 S_4 C_{3,5}) j + (-C_2 S_4 S_{3,5} - S_2 S_4 S_{3,5}) k >, \right.

\]

\[
< (-a_2 S_j + a_1 C_i + a_4 S_i + d_4 S_i + d_6 S_2 S_{2,3} + d_4 C_{2,3} - d_4 S_{2,3}) k > \right] \right] 

\[
Q_1 = H_1 \otimes Q_2 = \left[ < C_1 + S_1 j >, < a_1 C_i + a_1 S_j + d_1 k > \right] \otimes 

\[
\left[ \begin{array}{c}
(C_1 C_4 C_{2,3,5} + S_1 S_4 S_{2,3,5}) + (C_1 S_4 C_{2,3,5} - S_1 C_4 S_{2,3,5}) i \\
+ (C_1 S_4 S_{2,3,5} + S_1 C_4 C_{2,3,5}) j + (S_1 C_4 S_{2,3,5} - C_1 S_4 C_{2,3,5}) k \\
\end{array} \right] >, 

\]

\[
< (-a_2 S_j + a_1 C_i + a_4 S_i + d_4 C_{2,3} - d_4 S_{2,3} + d_6 S_5 C_{2,3} - d_4 C_{2,3} - d_4 S_{2,3}) k > \right] 

\]

Then calculating vector pair of quaternion using Eq. (29), to solve the inverse kinematics problem, the transformation quaternion of end effector of robot manipulator can be defined as

\[
O_{j+1} = H_j^{-1} \otimes O_j
\]
\[
[R_{1v}, T_{1v}] = O_1 = \left[< w + \hat{a}i + \hat{b}j + \hat{c}k >, < X\hat{i} + Y\hat{j} + Z\hat{k} > \right]
\]

(30)

Now using Eq. (29), \(O_2\) will be given by,

\[
O_2 = H_1^{-1} \otimes O_1
\]

\[
O_2 = \left[< \overline{c_1} - S\overline{k} >, < -a_1\hat{i} > \right] \otimes \left[< w + \hat{a}i + \hat{b}j + \hat{c}k >, < X\hat{i} + Y\hat{j} + Z\hat{k} > \right]
\]

(31)

where,

\[
O_{21} = w\overline{c_1} + c\overline{s_1}, \quad O_{22} = a\overline{c_1} + b\overline{s_1}, \quad O_{23} = b\overline{c_1} - a\overline{s_1}, \quad O_{24} = c\overline{c_1} - w\overline{s_1}, \quad O_{25} = X\overline{c_1} - a_1 + Y\overline{s_1}, \quad O_{26} = Y\overline{c_1} - X\overline{s_1}, \quad O_{27} = Z.
\]

Now,

\[
O_3 = H_2^{-1} \otimes O_2
\]

\[
O_3 = \left[< o_{31} + o_{32}\hat{i} + o_{33}\hat{j} + o_{34}\hat{k} >, < o_{35}\hat{i} + o_{36}\hat{j} + o_{37}\hat{k} > \right]
\]

(32)

where,

\[
o_{31} = \overline{c_2}\overline{c_{21}} + c(\overline{c_2}\overline{s_2}) - w(\overline{s_2}\overline{s_1}), \quad o_{32} = \overline{c_2}\overline{s_{21}} + \overline{s_2}\overline{c_{23}} - \overline{s_2}\overline{s_{22}}, \quad o_{33} = \overline{s_2}\overline{c_{23}} - \overline{s_2}\overline{c_{22}}, \quad o_{34} = \overline{s_2}\overline{s_{24}} - \overline{s_2}\overline{s_{21}}, \quad o_{35} = -ZS_2 + XC_1C_2 + YS_1C_2 - a_1C_2, \quad o_{36} = YC_1 - XS_1, \quad o_{37} = -ZC_2 + XC_1S_2 + YS_1S_2 - a_1S_2.
\]

\[
O_4 = \left[< o_{41} + o_{42}\hat{i} + o_{43}\hat{j} + o_{44}\hat{k} >, < o_{45}\hat{i} + o_{46}\hat{j} + o_{47}\hat{k} > \right]
\]

(33)

where,

\[
o_{41} = \overline{s_2}\overline{s_{21}} + \overline{S_2}\overline{c_{23}} + \overline{S_2}\overline{s_{25}},
\]

\[
o_{42} = \overline{s_2}\overline{s_{21}} - \overline{S_2}\overline{c_{23}} - \overline{S_2}\overline{s_{24}}
\]

\[
o_{43} = \overline{s_2}\overline{c_{23}} - \overline{s_2}\overline{c_{22}} - \overline{s_2}\overline{s_{25}}
\]

\[
o_{44} = \overline{s_2}\overline{s_{24}} - \overline{s_2}\overline{s_{21}}
\]

\[
o_{45} = -ZS_2 - a_1S_3 - XC_1S_2S_3 + a_1S_2S_3 - ZC_2S_3 + Z - ZC_3 + XC_1C_2C_3 + XS_1C_2C_3 - a_1C_2C_3
\]

\[
o_{46} = YC_1 - XS_1
\]

\[
o_{47} = -ZS_2S_3 + a_1C_3 + XC_1S_2S_3 + YS_1C_2S_3 - a_1C_3S_3 + ZC_2C_3 + XC_1S_2C_3 + YS_1S_2C_3 - a_1S_2C_3
\]

Therefore, all the joint variables can be calculated by equating quaternion vector products and quaternion vector pairs i.e. \(Q_1, Q_2\) and \(Q_3\) to \(O_1, O_2\) and \(O_3\) respectively.
\[ \Rightarrow \theta_1 = \tan \left( \frac{Y - u_y}{X - u_x} \right) \times \frac{(a_1 + d_4C_2 + d_4C_2 - d_4C_2 - a_6S_5C_2 + d_6S_5S_2S_3 + d_4S_5S_2S_3 - d_4S_2S_3 + d_6S_5C_2S_3)}{(a_2S_2 + d_4C_2 - d_4C_2 - d_4C_2 - d_6S_5C_2S_3 + d_6S_5S_2S_3 - d_4S_2S_3 - d_6S_5C_2S_3)} \]  

(34)

\[ \theta_2 = a \tan 2 \left( \frac{Z + v_y}{d_4} \right) \sqrt{1 - \left( \frac{Z + v_y}{d_4} \right)^2} \]  

(35)

where, \( d_4S_2 + d_4S_2 - v_y = Z, d_4S_2 + d_4S_2 = Z + v_y, S_2 = \left( \frac{Z + v_y}{d_4} \right) \)

\[ \theta_3 = a \tan 2 \left[ \frac{m \sqrt{1 - \left( \frac{-XS_2 + XC_1C_2 + YS_1C_2 - a_1C_2 - v_y}{-d_4 - d_6S_3} \right)^2}}{-ZS_2 + XC_1C_2 + YS_1C_2 - a_1C_2 - v_y} \right] \]  

(36)

\[ \theta_4 = a \tan 2 \left[ \frac{m \sqrt{1 - \left( \frac{-ZS_2S_1 + a_2C_3 + XCS_1C_2 + a_1C_2S_1 + a_3C_3 + ZC_2C_3 + XC_1S_1C_3 + YS_1S_2C_3 - a_2S_3}{-d_6S_3} \right)^2}}{-d_6S_3} \right] \]  

(37)

\[ \theta_5 = a \tan 2 \left[ \frac{m \sqrt{1 - \left( \frac{-ZS_2 - a_3S_1 - XC_1S_1S_2 - a_1S_1S_2 + ZCS_2 + ZC_1S_1C_3 + YS_1C_3C_3 - a_4C_3}{-d_4} \right)^2}}{-d_4} \right] \]  

(38)

3. Inverse kinematic solution scheme

In this section optimization algorithms are selected for computation of inverse kinematics solution of 5R manipulator. However, there are various types of optimization algorithms existed and can produce the desired IK solution, the major necessity is to achieve global optimum solution with fast convergence rate. Therefore, selection of appropriate optimization algorithm is important for fitness evaluations and GA is so far best known tool, but on the other hand TLBO has also proven its efficiency and performance. Finally selection of optimization algorithms has been made on global searching point, computational cost and quality of the result.

3.1. Optimization approach to solve inverse kinematics

Any Optimization algorithms which are capable of solving various multimodal functions can be implemented to find out the inverse kinematic solutions. The fitness function is given by the Eq. (46) fitness function \( F(x) \). Each individual represents a joint variable solution of the inverse kinematic problem for adopted population based metaheuristic algorithm. All individuals moving in D-dimensional search space and sharing the information to find out best fitness
value of the function. Each individual contains set of joint angles (θ₁, θ₂, θ₃, θ₄ and θ₅) of 5R manipulator. The optimum set of joint angle can be find by using appropriate optimization algorithm from given desired position of end effector (X, Y, Z). In case of inverse kinematics of 5R manipulator multiple solutions exist for the single position of the end effector so it is required to find out the best set of joint angle in order to minimize whole movement of manipulator.

For the optimization of joint angle rotation of robot manipulator, one can define objective function or fitness function from joint angle rotation difference and other can be defined from end effector position displacement. These are known as joint angle error and positional function method [3, 7, 10, 29].

3.1.1. Position based function

The current position of the manipulator is described by (39):

\[ P_c = [X_c, Y_c, Z_c] \]  

(39)

Desired position of end effector can be denoted by (40):

\[ P_d = [X_d, Y_d, Z_d] \]  

(40)

Current position of end effector will be compared with the desired position \( P_d \). General equation for the fitness function is given in Eq. (41) that is based on the distance norm of homogeneous Euclidian distance between the current positions to the desired position of end effector \( P_d \) evaluated by number of iterations.

\[ P_{\text{min}} = \| P_d - P_c(i) \|^2 \]  

(41)

Current position \( P_c \) can be evaluated from Eqs. (34) through (38). Now putting the value of \( P_c \) on Eq. (42)

\[ P_{\text{min}} = \sqrt{(X_d - X_c(i))^2 + (Y_d - Y_c(i))^2 + (Z_d - Z_c(i))^2} \]  

(42)

3.1.2. Joint angle error

Corresponding joint error can be given by the difference between current set of joint variables to the final required angles.

\[ \theta_c = (\theta_{c1}, \theta_{c2}, \theta_{c3}, \theta_{c4}, \theta_{c5}) \]  

(43)

\[ \theta_d = (\theta_{d1}, \theta_{d2}, \theta_{d3}, \theta_{d4}, \theta_{d5}) \]  

(44)

Therefore using square norm the objective function can be given as

\[ \theta_{\text{min}} = \| \theta_d - \theta_c \|^2 \]  

(45)

Subjected to joint limits
\[ \theta_1 \in [\theta_{1\text{min}}, \theta_{1\text{max}}] \]
\[ \theta_2 \in [\theta_{2\text{min}}, \theta_{2\text{max}}] \]
\[ \theta_3 \in [\theta_{3\text{min}}, \theta_{3\text{max}}] \]
\[ \theta_4 \in [\theta_{4\text{min}}, \theta_{4\text{max}}] \]
\[ \theta_5 \in [\theta_{5\text{min}}, \theta_{5\text{max}}] \]

Now overall error minimization can be given by using Eqs. (42) and (45),
\[
F(x)_{\text{min}} = \lambda \left\{ \sqrt{(X_d - X_c(i))^2 + (Y_d - Y_c(i))^2 + (Z_d - Z_c(i))^2} \right\} + \left\| \theta_d - \theta_c \right\|^2 \tag{46}
\]

where \( \lambda \) is proportional weight factor for the minimization of the problem and calculation of the entire joint angles base on constraint can be achieved using fitness function (46). The performance of considered algorithm is checked with the parameters: \( a_1 = 60 \) mm, \( a_2 = 145 \) mm, \( d_1 = 150 \) mm, \( d_2 = 125 \) mm, \( d_3 = 130 \) mm. Upper and lower limit of five joint angles are: \( \theta_1 = [0, 180] \); \( \theta_2 = [0, 150] \); \( \theta_3 = [0, 150] \); \( \theta_4 = [0, 85] \) and \( \theta_5 = [15, 45] \).

4. Results and discussions

TLBO and GA has been used to compute the inverse kinematics of 5-R manipulator and comparison of obtained results has been made on the basis of quality and performance. Table 1 gives the five random position of end effector and respective inverse kinematics solutions. Current work is performed in MATLAB R2013a. The data sets are obtained from Eq. (34) through (38). The data sets are generated using quaternion vector based inverse kinematics equations as given in Table 2. These generated data sets are used to compare the IK solution through adopted GA and TLBO. In Table 3, comparative evaluations of fitness function and obtained joint variables through TLBO and GA is presented.

<table>
<thead>
<tr>
<th>Positions</th>
<th>Joint angles</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_1(-76.09, 54.36, -61.94) )</td>
<td>84.559, 77.318, 101.74, 30.616, 38.697</td>
</tr>
<tr>
<td>( P_2(89.69, 192.55, 90.87) )</td>
<td>84.791, 97.25, 130.44, 50.771, 36.428</td>
</tr>
<tr>
<td>( P_3(-4.24, 94.08, 97.55) )</td>
<td>18.384, 78.688, 35.234, 77.708, 34.889</td>
</tr>
<tr>
<td>( P_4(29.10, 154.02, 31.52) )</td>
<td>104.43, 115.47, 124.11, 7.3372, 33.774</td>
</tr>
<tr>
<td>( P_5(-184.33, 43.21, 8.27) )</td>
<td>39.177, 107.13, 97.052, 65.672, 15.374</td>
</tr>
</tbody>
</table>

Table 1. Five different positions and joint variables.
This work does not use special tuning of various parameters of GA and TLBO algorithm. In future research the sensitivity analysis can be performed to achieve better results. From Table 3, TLBO generated solutions for the position 4 is better as compared to GA in account of fitness function evaluation. There are different distance based norms, one of them is Euclidean distance

\[
\begin{array}{ccccccc}
\theta_1 & \theta_2 & \theta_3 & \theta_4 & \theta_5 & X & Y & Z \\
4 & 57.6885 & 119.6396 & 104.5818 & 71.1946 & 33.8225 & -10.7684 & 111.7435 & 77.4862 \\
5 & 31.4308 & 2.9242 & 71.5757 & 63.0358 & 39.8749 & 64.7966 & 172.0372 & 151.5714 \\
7 & 89.1765 & 13.1827 & 101.6747 & 80.3340 & 29.4704 & -76.9533 & 96.2813 & 121.3505 \\
10 & 32.8579 & 102.3539 & 138.8770 & 26.4141 & 33.7466 & 146.4984 & 48.7416 & 54.4041 \\
11 & 134.1878 & 70.4224 & 43.5151 & 82.2471 & 44.0566 & -188.7864 & 15.4108 & -53.9823 \\
\end{array}
\]

Table 2. Desired joint variables determined through quaternion algebra.

This work does not use special tuning of various parameters of GA and TLBO algorithm. In future research the sensitivity analysis can be performed to achieve better results. From Table 3, TLBO generated solutions for the position 4 is better as compared to GA in account of fitness function evaluation. There are different distance based norms, one of them is Euclidean distance

\[
\begin{array}{ccccccc}
\theta_1 & \theta_2 & \theta_3 & \theta_4 & \theta_5 & X & Y & Z \\
P1(-76.09, 54.36, -61.94) & 86.598 & 72.165 & 72.165 & 40.894 & 30.459 & 0 \\
P2(89.69, 192.55, 90.87) & 83.874 & 69.895 & 69.895 & 39.607 & 30.76 & 0 \\
P3(-4.24, 94.08, 97.55) & 84.512 & 70.427 & 70.427 & 39.909 & 30.686 & 0 \\
P4(29.10, 154.02, 31.52) & 85.566 & 71.305 & 71.305 & 40.406 & 30.53 & 0 \\
P5(-184.33, -43.21, 8.27) & 87.818 & 73.181 & 73.181 & 41.469 & 30.364 & 0 \\
\end{array}
\]

Table 3. TLBO results for joint variable and function value.
norm and which is used here for minimum distance between the end effector positions. If the
distance between two points reached to 0 or less than 0.001 than the evolutions of fitness function
can be reached best or global minimum value. It is clear that the obtained fitness value is less
than the defined distance norm so adopting these algorithms are fruitful and qualitative.

Figures 3–7 signify the best fitness function value and analogous joint variables for position 1.
These figures show efficiency of adopted algorithms for IK solution of 5-R manipulator. The
convergence of objective function evaluation lies to zero error for GA and TLBO algorithms
while for position 4, GA yields 0.013 error. It means that GA is less performing as compared to
TLBO. From Figures 8–12, the results obtained through GA shows in terms of convergence and
histogram graph and the obtained joint angles are in radian which is later converted into
degree and given in Table 3. The GA results are obtained through MATLAB toolbox and that
shows the zero convergence in single run. Figures 8–12, it can be seen that the generated
solutions for joint angles are multiple for single position and similarly there are multiple fitness
function evaluations. The best fitness function achieved here using the termination criteria and
the corresponding joint variables has taken for comparison.

The proposed work is performed in dual core system with 4 GB RAM computer. It has been
observe that the convergence of the solution for GA is taking less computation time as com-
pared to TLBO and quaternion algebra. Corresponding joint angles trajectory using 4th order
cubic spline is presented in Figure 13. Using inverse kinematic solution joint variables are used

![Figure 3](image3.png)
**Figure 3.** Joint variables and fitness function value for position P1.

![Figure 4](image4.png)
**Figure 4.** Joint variables and fitness function value for position P2.
Figure 5. Joint variables and fitness function value for position P3.

Figure 6. Joint variables and fitness function value for position P4.

Figure 7. Joint variables and fitness function value for position P5.

Figure 8. Joint variables and fitness function value for position P1.
Figure 9. Joint variables and fitness function value for position P2.

Figure 10. Joint variables and fitness function value for position P3.

Figure 11. Joint variables and fitness function value for position P4.

Figure 12. Joint variables and fitness function value for position P5.
to calculate the joint space trajectory for TLBO and GA as presented in Figures 14 and 15. Final time has been taken $t_f=6$ second for trajectory generation but to complete this trajectory overall computational time is 5.674 seconds. The computation time for TLBO is 15.671 seconds which is more than the GA i.e. 7.932 seconds. Therefore, on the basis of computational cost GA is performing better than TLBO while quaternion algebra taken least time (Table 4).
5. Conclusions

In this paper, the work discourses the problem associated to the optimization of positional and angular error of end effector using TLBO and GA for 5R robot manipulator. Metaheuristic algorithms like PSO, GA, ABC, etc. have been used in various field of industrial robotics but the most critical issue is to solve inverse kinematic problem for any configuration of robot manipulators. Most of the optimization approach are being used for numerical solution but it has been observed that the numerical solutions does not yield solution when the manipulator is in ill-conditioned besides this it has also been observed that classical optimization methods converge in local minima. Therefore in this work global optimization method like TLBO and GA is adopted and after analyzing the results it can be concluded that adopted optimization algorithms convergence rate is higher and complexity does not increase with the manipulator configuration. Although many researchers are tried to obtain global solution but the computations cost are more in the problem henceforth overcoming the problem of computational cost with quaternion objective function.

The adopted algorithms are very much appropriate for constrained and unconstrained problems. To estimate the effectiveness of considered algorithms, comparison has been made with

<table>
<thead>
<tr>
<th>SN</th>
<th>Method</th>
<th>Computational time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TLBO</td>
<td>15.671 s</td>
</tr>
<tr>
<td>2</td>
<td>GA</td>
<td>7.932 s</td>
</tr>
<tr>
<td>3</td>
<td>Quaternion</td>
<td>0.993 s</td>
</tr>
</tbody>
</table>

Table 4. Computational time for inverse kinematic evaluations.

Figure 15. Trajectory of joint angle for GA.
quaternion algebra. **Table 3** gives the comparative results of adopted algorithm and proposed quaternion solutions of 5-R manipulator. This work considered forward and inverse kinematic equations for preparing the objective function for TLBO and GA. These adopted algorithms has shown the potential of getting faster convergence and yielding global optimum solution for the stated problem. In future the tuning of various parameters of GA and TLBO can be considered so as to avoid trapping in local minimum point. Even the hybridization of these algorithms may be proposed and adopt for the IK problems.

**Author details**

Panchanand Jha¹ and Bibhuti Bhusan Biswal²

*Address all correspondence to: jha_ip007@hotmail.com
¹ Raghu Engineering College, Visakhapatnam, India
² National Institute of Technology, Meghalaya, India

**References**


[36] Zu D. Efficient inverse kinematic solution for redundant manipulators. JME. 2005;41:71. DOI: 10.3901/jme.2005.06.071