2009

Solution of inverse kinematic problem for serial robot using dual quaternions and plucker coordinates

Emre Sariyildiz  
*University of Wollongong, emre@uow.edu.au*

Hakan Temeltas  
*Istanbul Technical University*

Publication Details
Solution of inverse kinematic problem for serial robot using dual quaternions and plucker coordinates

Abstract
In this paper we present a new formulation method to solve kinematic problem of serial robot manipulators. In this method our major aims are to formulize inverse kinematic problem in a compact closed form and to avoid singularity problem. This formulation is based on screw theory with dual - quaternion. Compared with other methods, screw theory methods just establish two coordinates, and its geometrical meaning is obvious. We used dual-quaternion in plucker coordinates as a screw operator for compactness. 6R-DOF industrial robot manipulators forward and inverse kinematic equations are derived using this new formulation and simulation results are given.

Keywords
solution, robot, inverse, kinematic, dual, problem, serial, quaternions, coordinates, plucker

Disciplines
Engineering | Science and Technology Studies

Publication Details

This conference paper is available at Research Online: http://ro.uow.edu.au/eispapers/6727
Solution of Inverse Kinematic Problem for Serial Robot Using Dual Quaternions and Plücker Coordinates

E. Sariyildiz, H. Temeltas

Abstract—In this paper we present a new formulation method to solve kinematic problem of serial robot manipulators. In this method our major aims are to formulize inverse kinematic problem in a compact closed form and to avoid singularity problem. This formulation is based on screw theory with dual-quaternion. Compared with other methods, screw theory methods just establish two coordinates, and its geometrical meaning is obvious. We used dual-quaternion in plücker coordinates as a screw operator for compactness. 6R-DOF industrial robot manipulators forward and inverse kinematic equations are derived using this new formulation and simulation results are given.

I. INTRODUCTION

The problem of kinematic is to describe the motion of the manipulator without consideration of the forces and torques causing the motion. There are two main kinematic problems. First one is forward kinematic problem, which is to determine the position and orientation of the end effector given the values for the joint variables of the robot. The second one is inverse kinematic problem is to determine the values of the joint variables given the end effector’s position and orientation. [1] At least 6 parameters are needed to describe the motion of the rigid body. Three parameters are for position, three parameters are for orientation.

Several methods are used in robot kinematic. The most common method is Denavit and Hartenberg notation for definition of special mechanism [2]. This method is based on point transformation approach and it is used 4 x 4 homogeneous transformation matrix which is introduced by Maxwell [3]. Maxwell used homogeneous coordinate systems to represent points and homogeneous transformation matrices to represent the transformation of points. The coordinate systems are described with respect to previous one. For the base point an arbitrary base coordinate system is used. Hence some singularity problems may occur because of this coordinate systems description. And also in this method 16 parameters are used to represent the transformation of rigid body while just 6 parameters are needed.

Another main method in robot kinematic is screw theory which is based on line transformations approach. The elements of screw theory can be traced to the work of Chasles and Poinset in the early 1800s. Using the theorems of Chasles and Poinset as a starting point, Robert S. Ball developed a complete theory of screws which he published in 1900. [4] In screw theory every transformation of a rigid body or a coordinate system with respect to a reference coordinate system can be expressed by a screw displacement, which is a translation by along a λ axis with a rotation by a θ angle about the same axis [4]. This description of transformation is the basis of the screw theory. There are two main advantages of using screw, theory for describing rigid body kinematics. The first one is that it allows a global description of rigid body motion that does not suffer from singularities due to the use of local coordinates. The second one is that the screw theory provides a geometric description of rigid motion which greatly simplifies the analysis of mechanisms [12].

Several application of screw theory has been introduced in kinematic. Yang and Freudenstein were the first to apply line transformation operator mechanism by using the dual-quaternion as the transformation operator. Yang also investigated the kinematics of special five bar linkages dual 3 x 3 matrices [7]. Kumar and Kim obtained kinematic equations of 6-DOF robot manipulator by using dual-quaternion and DH parameters [7]. Dual – quaternion parameters are obtained from DH parameters. This method has singularity problem. In inverse kinematic they used geometrical solution approach with DH parameters. Hence they couldn’t avoid singularity problem. M. Murray solved 3-DOF and 6-DOF robot manipulator kinematics by using screw theory with 4x4 matrix operator [15]. In this method non-singular inverse kinematic solutions are obtained but 4x4 matrix operator is used. This operator needs 16 parameters while just 6 parameters are needed for description of rigid body motion.

In this paper we present a new formulation method to solve kinematic problem of serial robot manipulators. Our method plücker coordinates are used for line transformation and dual-quaternion is used as a transformation operator. Plücker Coordinate is a way to represent a line in homogeneous geometry. Body velocities and forces can be easily represented by using plücker coordinates. Singularity avoiding solution is obtained for serial robot manipulator and also just eight parameters are used to describe rigid body motion. In this method we used just two coordinates.
which are at the base and at the end effectors. For the other joints just axis are used to describe joint motions. All joint axis and end effectors coordinate are transformed with respect to base coordinate. Hence we avoid singularity problem. Its geometric meaning is obvious and it is very easy to implement to the robot manipulators. Dual-quaternion is used as a screw operator. Dual operators are the best way to describe screw motion and also the dual-quaternion is the most compact and efficient dual operator to express screw displacement [5] [6]. 6R-DOF robot manipulator forward and inverse kinematic problems are solved and also simulation results are given.

II. LINE GEOMETRY AND DUAL NUMBER

A line can be completely defined by the ordered set of two vectors. First one is point vector \( p \) which indicates the position of an arbitrary point on line, and the other vector is free direction vector \( d \) which gives the line direction. A line can be expressed as:

\[ L(p,d) \]  

Fig. 1. A line in Cartesian coordinate-system

The representation \( L(p,d) \) is not minimal, because it uses six parameters for only four degrees of freedom. With respect to a world reference frame, the line’s coordinates are given by a six-vector [8].

A. Plücker Coordinates

An alternative line representation was introduced by A. Cayley and J. Plücker. Finally this representation named after Plücker [8]. Plücker coordinates can be expressed as:

\[ L_p(m,d) \]  

\[ m = p \times d \]  

Both \( d \) and \( m \) are free vectors; \( d \) and \( p \) have the same meaning as before (they represent the direction of the line and the position of an arbitrary point on the line respectively) and \( m \) is the moment of \( d \) about the chosen reference origin. Note that \( m \) is independent of which point \( p \) on the line is chosen and the two three-vectors \( d \) and \( m \) are always orthogonal.

\[ p \times d = (p + td) \times d \]  

\[ d \cdot m = 0 \]  

Plücker coordinates representation is also not minimal representation. The advantage of plücker coordinate representation is that it is homogeneous: \( L_p(m,d) \) represents same line as \( L_p(km,kd) \), where \( k \in \mathbb{R} \).

B. Intersection of Two Lines

If we write two lines in plücker coordinates as:

\[ L_a = (ma, da) \quad \text{and} \quad L_b = (mb, db) \]  

The intersection point of two intersection lines can be formulated as:

\[ r = db \times mb + (da \times ma \cdot db) \cdot da \quad \text{or} \quad r = da \times ma + (db \times mb \cdot da) \cdot da \]  

where \( r \) indicates the intersection point.

C. Dual Numbers

The dual number was originally introduced by Clifford in 1873 [9]. In analogy with a complex number a dual number can be defined as:

\[ \hat{u} = u + \epsilon u^0 \]  

where \( u \) and \( u^0 \) are real number and \( \epsilon^2 = 0 \). We can use dual numbers to express plücker coordinates. We can write orientation vector \( (u) \) and moment vector \( (u^0) \) as:

\[ u = (x_2 - x_1, y_2 - y_1, z_2 - z_1) \quad \text{and} \quad u^0 = p \times u \]  

Hence a line can be expressed in plücker coordinates by using dual number as:

\[ u + \epsilon u^0 \]  

III. QUATERNION & DUAL QUATERNION

A. Quaternion Definition & Unit Quaternion

Quaternions are hyper-complex numbers of rank 4, constituting a four dimensional vector space over the field of real numbers [10] [11]. A quaternion can be represented as:

\[ q = (q_0, v) \]  

where \( q_0 \) is a scalar and \( q_v = (q_1, q_2, q_3) \) is a vector. A quaternion with \( q_0 = 0 \), is called as a real quaternion, and a quaternion with \( q_0 = 0 \), is called as a pure quaternion (or vector quaternion). Addition and multiplication of two quaternions can be expressed as:

\[ q_a + q_b = (q_{a0} + q_{b0}), (q_{av} + q_{bv}) \]  

\[ q_a \otimes q_b = q_{a0} q_{b0} - q_{av} q_{bv} \cdot q_{a0} q_{bv} + q_{b0} q_{av} + q_{av} \times q_{bv} \]  

where “\( \otimes \)””, “\( \cdot \)”, “\( \times \)” denotes quaternion product, dot product and cross product respectively. Conjugate and norm of the quaternion can be expressed as:

\[ q^* = (q_0, -v) = (q_0, -q_1, -q_2, -q_3) \]  

\[ ||q||^2 = q \otimes q^* = q_0^2 + q_1^2 + q_2^2 + q_3^2 \]  

339
When \(|q|\) = 1, we get a unit quaternion. The inverse of a quaternion can be expressed as:

\[ q^{-1} = \frac{1}{||q||^2} q^* \quad \text{and} \quad ||q|| \neq 0 \tag{15} \]

that satisfies the relation \(q^{-1} \otimes q = q \otimes q^{-1} = 1\). For a unit-quatination we have \(q^{-1} = q^*\). Unit quaternion can be defined as a rotation operator. Rotation about a unit axis \(n\) with an angle \(\theta\) is expressed as:

\[ q = (\cos(\frac{\theta}{2}), \sin(\frac{\theta}{2})n) \tag{16} \]

B. Dual Quaternion

A dual-quaternion can be defined as:

\[ \hat{q} = (\hat{q}_v, \hat{q}_o) \quad \text{or} \quad \hat{q} = q + \epsilon q^0 \tag{17} \]

where \(\hat{q}_o = q_o + \epsilon q^0_o\), is a dual scalar, \(\hat{q}_v = q_v + \epsilon q^0_v\), is a dual vector, and \(q^0\) are both dual quaternions, \(\epsilon\) is the dual factor. Addition and multiplication of two dual-quaternions can be expressed as:

\[
\begin{align*}
\hat{q}_a + \hat{q}_b &= (q_a + q_b) + \epsilon(q^0_a + q^0_b) \\
\hat{q}_a \otimes \hat{q}_b &= (q_a \otimes q_b) + \epsilon(q^0_a \otimes q^0_b + q_a \otimes q^0_b + q^0_a \otimes q_b) \tag{18}
\end{align*}
\]

where “\(\otimes\)” and “\(\oplus\)” denote quaternion and dual-quaternion product respectively. Conjugate, norm and inverse of the dual-quaternion is similar with quaternion. They can be expressed as:

\[
\begin{align*}
\hat{q}^* &= q^* + \epsilon(q^0)^* \tag{19} \\
||\hat{q}||^2 &= \hat{q} \otimes \hat{q}^* \tag{20} \\
\hat{q}^{-1} &= \frac{1}{||q||^2} \hat{q}^* \tag{21}
\end{align*}
\]

When \(|\hat{q}|^2 = 1\), we get a unit dual-quaternion. For unit dual-quaternion these equations can be written.

\[
\begin{align*}
||\hat{q}||^2 &= \hat{q} \otimes \hat{q}^* = 1 \quad \text{and} \quad \\
q \otimes q^* &= 1, \quad q^* \otimes q^0 + (q^0)^* \otimes q = 0 \tag{22}
\end{align*}
\]

C. Line Transformation by Using Dual-Quaternions

A unit-quaternion can be used as a rotation operator. A point \(p_b\) can be transformed to a point \(p_a\) by using unit quaternions as follow:

\[ p_a = q \otimes p_b \otimes q^* \tag{23} \]

where \(q\) is unit-quaternion. Unit-quaternions can be used for transformation of a point but general rigid transformation can’t be implemented by using unit-quaternions. A general rigid transformation has 6 DOF. Hence we need a transformation operator which has at least six parameters. We can use dual-quaternion for general rigid transformation [13]. Although it has eight parameters and it is not minimal, it is the most compact and efficient dual operator [5] [6].

Now, we explain how dual-quaternion allows a rigid-transformation. This transformation is very similar with pure rotation; however, not for a point but for a line. A line in plücker coordinates \((l_p(m, d))\) can be expressed by using dual quaternions as:

\[ \hat{l}_a = l_a + \epsilon m_a \tag{24} \]

After transformation of \(\hat{l}_a\) \((R:\text{rotation and } t:\text{translation})\) we obtain a transformed line \(\hat{l}_b\) \((\hat{l}_b = l_b + \epsilon m_b)\). Transformation of line can be expressed as:

\[ l_b = q \otimes l_a \otimes q^* \]

\[ m_b = q \otimes m_a \otimes q^* + \frac{1}{2} (q \otimes l_a \otimes q^* \otimes t^* + t \otimes q \otimes l_a \otimes q^*) \tag{25} \]

If we define a new quaternion \(q^* = \frac{1}{2} t \otimes q\) \((t \text{ is translation})\) and a new dual quaternion \(\hat{q} = q + \epsilon q^*\), we can formulize the transformation of a line given by

\[ l_b + \epsilon m_b = (q + \epsilon q^*) \otimes (l_a + \epsilon m_a) \otimes (q^* + \epsilon q^*) \tag{26} \]

IV. SCREW THEORY WITH DUAL-QUATERNION

The elements of screw theory can be traced to the work of Chasles and Poinsot in the early 1800s [4]. According to Chasles all proper rigid body motions in 3-dimensional space, with the exception of pure translation, are equivalent to a screw motion, that is, a rotation about a line together with a translation along the line [14]. A screw motion can be formulized using dual-quaternion as:

\[ \hat{q} = \begin{pmatrix} q_o^0_t \varepsilon \end{pmatrix} + \varepsilon \begin{pmatrix} - \frac{1}{2} \begin{pmatrix} q \times t \end{pmatrix} \\
\frac{1}{2} q_o^0 t + t \times q_o \end{pmatrix} \tag{27} \]

This representation is very compact and also it uses algebraically separates the angle (\(\theta\)) and pitch (\(k\)) information. Thus it is very powerful [12]. Moreover if we write \(\hat{q} = \theta + ek\) and \(\hat{d} = d + \epsilon m\) equation (27) becomes:

\[ \hat{q} = \cos(\frac{\theta}{2}) + \sin(\frac{\theta}{2})\hat{d} \tag{28} \]

V. MANIPULATOR KINEMATIC

A. Forward Kinematic

To find forward kinematic of serial robot manipulator we followed these steps:

**Notation:**

1. Label the joints and the links: Joints are numbered from number 1 to n, starting at the base, and the links are numbered from number 0 to n. The joints connect link i-1 to link i.
2. **Configuration of joint spaces**: For revolute joint we describe rotational motion about an axis and we measure all joint angles by using a right-handed coordinate system. For prismatic joint we describe a linear displacement along the direction of the axis.

3. **Attaching coordinate frames (Base and Tool Frames)**: Two coordinate frames are needed for n degree of freedom open-chain robot manipulator. The base frame can be attached arbitrary but in general it is attached directly to link 0 and the tool frame is attached to the end effector of robot manipulator. This notation is given for 6-DOF serial robot manipulator in figure 2.

**Formulation:**

1. **Determining joint axis vector and moment vector**: First we attach an axis vector which describes the motion of the joint. Then the moment vector of this axis is obtained for revolute joints by using (3). Hence we obtain the plucker notation of this axis.

2. **Obtaining transformation operator**: For all joints we obtain dual-quaternion for transformation operator as follow:

   \[
   \tilde{q}_i = (\tilde{q}_{oi}, \tilde{q}_{vi}) \quad \tilde{q}_i = q_i + \epsilon q_i^0 \quad (29)
   \]

3. **Formulation of rigid motion**: Using (18) transformation of serial robot manipulator can be obtained as:

   \[
   q_{tn} = q_t \otimes q_2 \otimes \ldots \otimes q_n \\
   q_{tn}^0 = q_{tn-1}^0 \otimes q^n_0 + q^n_{tn-1} \otimes q^n_0 \\
   (30) \quad (31)
   \]

Using (7), the position of the end effector can be written as:

\[
 t = (q_{vn} \times q_{vn}^0) + (q_{vn-1} \times q_{vn-1}^0), q_{vn} * q_{vn} \\
 (32)
\]

**B. Inverse Kinematic**

We will use Paden-Kahan subproblems to obtain the inverse kinematic solution of serial robot manipulators. There are some Paden-Kahan subproblems and also new extended subproblems [15] [16] [17]. We will use just three of them which occur frequently in inverse solutions for common manipulator design. To solve the inverse kinematics problem, we reduce the full inverse kinematics problem into appropriate subproblems. Here are some subproblems [15], [16].

1. Rotation about a single axis.
2. Rotation about two subsequent axes.
3. Rotation to a given distance

**VI. 6R SERIAL MANIPULATOR KINEMATIC MODEL**

**A. Forward Kinematic**

First we must determine the axes for all joints. Then we will find the moment vector for all axes. The axes and the moment vectors can be written as follow

\[
\begin{align*}
\mathbf{d}_1 &= [0 \ 0 \ 1] & \mathbf{d}_2 &= [0 \ 1 \ 0] & \mathbf{d}_3 &= [0 \ 1 \ 0] \\
\mathbf{d}_4 &= [0 \ 0 \ 1] & \mathbf{d}_5 &= [0 \ 1 \ 0] & \mathbf{d}_6 &= [1 \ 0 \ 0] \\
\end{align*}
\]

Any point on these axes can be written as:

\[
\begin{align*}
\mathbf{p}_1 &= [0 \ 0 \ l_0] & \mathbf{p}_2 &= [0 \ 0 \ l_0] \\
\mathbf{p}_3 &= [l_1 \ 0 \ l_0] & \mathbf{p}_4 &= [l_1 + l_2 \ 0 \ l_0] \\
\mathbf{p}_5 &= [l_1 + l_2 \ 0 \ l_0] & \mathbf{p}_6 &= [l_1 \ 0 \ l_0] \\
\end{align*}
\]

The moment vectors of these axes are obtained by using (3). Now we can write dual-quaternion by using axes and moment vectors. Dual-quaternion can be obtained from equation (29) where i = 1, 2…6. Finally we can find forward kinematic equation by using equations (30), (31) and (32) where n is equal to 6. Our general forward kinematic equation is:

\[
\hat{q} = \hat{q}_{16} \odot (d_6 + \epsilon m_6) \odot \hat{q}^*_1 \\
\]

where

\[
\hat{q}_{16} = \hat{q}_1 \odot \hat{q}_2 \odot \hat{q}_3 \odot \hat{q}_4 \odot \hat{q}_5 \odot \hat{q}_6 \\
\]

![Fig. 2. 6-dof serial arm robot manipulator in its reference configuration](image)

**B. Inverse Kinematic**

In the inverse kinematic problem of the serial manipulator, we have orientation and position knowledge of the end effector, \((\hat{q}_{tn} = q_{tn}, \hat{q}_{tn}^0)\). We will calculate all joint angles by using the orientation and the position knowledge of the end effector. We must convert the complete inverse kinematic problem into the appropriate subproblems. First we put two points at the intersection of the axes. First one is \(\mathbf{p}_w\) which is at the intersection of the wrist axes and the second one is \(\mathbf{p}_b\) which is at the intersection of the first two axes. The last three joints angles do not affect the point \(\mathbf{p}_w\).

Hence we can say the position of point \(\mathbf{p}_w\) is free from the wrist angles and we can formulate it as:

\[
(q_{v5} \times q_{v5}^0) + (q_{v5} \times q_{v5}^0), q_{v6} * q_{v6} = (q_{v3} \times q_{v3}^0) + (q_{v2} \times q_{v2}^0), q_{v3} * q_{v3} = \mathbf{p}_w \\
\]

We can write same equation for the point \(\mathbf{p}_b\) as:

\[
(q_{v2} \times q_{v2}^0) + (q_{v1} \times q_{v1}^0), q_{v2} * q_{v2} = \mathbf{p}_b \\
\]
The position of the point \( p_b \) is free from the angles of first two joints. If we subtract equation (36) from both of side of the equation (37) we obtain
\[
(q_{v6} \times q_{v6}^0) + (q_{v5} \times q_{v5}^0). q_{v6} \times q_{v6} - p_b = (q_{v3} \times q_{v3}^0) + (q_{v2} \times q_{v2}^0). q_{v3} \times q_{v3} - p_b
\]  
If we take the end effector position \( q_{in}^0 = (q_{v0}, q_{v1}, q_{v2}, q_{v3}) \) we get \( p_e = q_{in}^0 \). Hence we can write
\[
(q_{v3} \times q_{v3}^0) + (q_{v2} \times q_{v2}^0). q_{v3} \times q_{v3} - (q_{v2} \times q_{v2}^0) - (q_{v1} \times q_{v1}^0). q_{v2} \times q_{v2} = q_{in} - p_b
\]  
Using the property that distance between points is preserved by rigid motions, take the magnitude of both sides of equation (39) we obtain subproblem 3. \( \theta_3 \) can be found by using subproblem 3 as:
\[
\theta_3 = \theta_0 \pm \cos^{-1} \left( \frac{|[\mathbf{v}']_1 + |[\mathbf{v}']_1| - \delta^2}{2|[\mathbf{v}']_1|} \right)
\]  
where \( \theta_0 = \arctan 2(d^T_2(u' \times v')u'^Tv') \)

\[
u' = (p_b - r) - d_3 d_1^T(p_b - r)
\]
\[
u'^2 = (p_b - r) - d_3 d_1^T(p_b - r)
\]
\[
\delta^2 = \left( \frac{\sqrt{(q_{in} - p_b)}(q_{in} - p_b)}{(q_{in} - p_b)} - |d_3^T(p_b - p_b)| \right)^2
\]
\( \mathbf{r} \) is any point on the axis of \( d_3 \). If we translate \( p_w \) by using known \( \theta_3 \) we obtain a new point \( p \) and subproblem 2.
\[
p = d_3 \times p^o
\]

\[
p^o = q_3 \otimes m_\omega \otimes q_3 + q_3 \otimes d_3 \otimes q_3^o + q_3^o \otimes d_3 \otimes q_3
\]

Now we can find \( \theta_1 \) and \( \theta_2 \) as follow:
\[
\theta_2 = \arctan 2(d^T_2(u' \times v')u'^Tv')
\]
\[
u' = (p_b - r) - d_3 d_1^T(v - r)
\]
\[
\nu'^2 = (p_b - r) - d_3 d_1^T(v - r)
\]
\[
\alpha = \frac{(d_1^Td_2)(p - r) - d_1^T(q_{in} - r)}{(d_1^Td_2)^2 - 1}
\]
\[
\beta = \frac{(d_1^Td_2)(d_1^T(q_{in} - r) - d_1^T(p - r))}{(d_1^Td_2)^2 - 1}
\]
\[
\gamma = \frac{|p - r|^2 - \alpha^2 - \beta^2 - 2\alpha\beta d_1^Td_2}{|d_1 \times d_2|^2}
\]
\[
\theta_1 = \arctan 2(-d_1^T(u' \times v')u'^Tv')
\]

where \( \mathbf{v} \) is same with equation (45) and \( \mathbf{r} \) is the intersection point of the axis one and axis two.

To find wrist angles we put a point \( p_t \) (initial point) which is on the \( d_4 \) axis and it does not intersect with \( d_4 \) and \( d_5 \) axes. Two imaginer axes are used to find \( p_e \) (end point), that is, the position of the point \( p_t \) after rotation by \( \theta_4 \) and \( \theta_5 \). It can be found by using equation (7). These two imaginer axes intersect on \( d_6 \) axis and the point \( p_t \) is the intersection point of these imaginer axes. Hence it is not affected from the last joint angle. Fourth and fifth joints angles determine the position of the point \( p_e \). This gives us a subproblem 2. \( \theta_4 \) and \( \theta_5 \) can be solved as follow:
\[
\theta_5 = \arctan 2(d_3^T(u' \times v')u'^Tv')
\]
\[
u' = (p_e - r) - d_5 d_1^T(p_e - r)
\]
\[
\nu'^2 = (p_e - r) - d_5 d_1^T(v - r)
\]
\[
\alpha = \frac{(d_3^Td_4) d_5^T(p_e - r) - d_5^T(p_e - r)}{(d_4^Td_5)^2 - 1}
\]
\[
\beta = \frac{(d_4^Td_5)d_5^T(p_e - r) - d_5^T(p_e - r)}{(d_4^Td_5)^2 - 1}
\]
\[
\gamma = \frac{|p_e - r|^2 - \alpha^2 - \beta^2 - 2\alpha\beta d_3^Td_5}{||d_4 \times d_5||^2}
\]
\[
\theta_4 = \arctan 2(-d_4^T(u' \times v')u'^Tv')
\]

where \( \mathbf{v} \) is same with equation (51) and \( \mathbf{r} \) is the intersection point of the wrist axes. Thus first five joints angles are obtained. Only the last joint angle is unknown. To find last joint angle we need a point which is not on the last joint axis. We call it \( p_d \). The position of the point \( p_d \) after rotation by \( \theta_6 \) can be found by using two imaginer axes and equation (7). This gives us a subproblem 1. \( \theta_6 \) can be found as follow:
\[
\theta_6 = \arctan 2(d_3^T(u' \times v'), u'^Tv')
\]
\[
u' = (p_d - r) - d_5 d_1^T(p_d - r)
\]
\[
\nu'^2 = (p_d - r) - d_5 d_1^T(v - r)
\]

VII. SIMULATION RESULTS

6R-robot manipulator forward and inverse kinematic problems are solved and simulation results are obtained by
using screw theory with dual-quaternion and D-H convention. These two methods are compared in terms of computational efficiency, singularity avoiding and accuracy. Some solution results are given below. As we can see from table I and table II screw theory is a singularity avoiding method and it is more accurate than D-H convention. In singular case, however we can find finite inverse kinematic solutions when we use screw theory, we can’t find finite (or real) solutions when we use D-H convention. Screw theory solutions errors are smaller than D-H convention solutions. And also screw theory solutions are computed faster than D-H convention as shown in figure 3. Forward kinematic solutions are obtained in about 0.1 seconds for both methods. Inverse kinematic solutions are obtained in about 0.21 seconds for screw theory and 0.40 seconds for D-H convention. Running environment is as table III.

Table I

<table>
<thead>
<tr>
<th>Real Angle</th>
<th>Screw Solutions</th>
<th>Screw Error</th>
<th>D-H Solution</th>
<th>D-H Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>01=0.6283</td>
<td>01=0.6283</td>
<td>0</td>
<td>01=0.6283</td>
<td>0</td>
</tr>
<tr>
<td>02=0.5236</td>
<td>02=0.5236</td>
<td>0</td>
<td>02=0.5236</td>
<td>0</td>
</tr>
<tr>
<td>03=0.4488</td>
<td>03=0.4488</td>
<td>0</td>
<td>03=0.4488</td>
<td>0</td>
</tr>
<tr>
<td>04=0.5236</td>
<td>04=0.5236</td>
<td>0</td>
<td>04=0.5236</td>
<td>0.0019</td>
</tr>
<tr>
<td>05=0.2856</td>
<td>05=0.2856</td>
<td>0</td>
<td>05=0.2855</td>
<td>0.0001</td>
</tr>
<tr>
<td>06=1.0472</td>
<td>06=1.0471</td>
<td>0.0001</td>
<td>06=1.0474</td>
<td>0.0001</td>
</tr>
</tbody>
</table>

Table II

<table>
<thead>
<tr>
<th>Real Angle</th>
<th>Screw Solutions</th>
<th>D-H Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>01=0.6283</td>
<td>01=0.6283</td>
<td>Unreal</td>
</tr>
<tr>
<td>02=0.5236</td>
<td>02=0.5236</td>
<td>Unreal</td>
</tr>
<tr>
<td>03=1.5708</td>
<td>03=0</td>
<td>Unreal</td>
</tr>
<tr>
<td>04=0.5236</td>
<td>04=0.6434</td>
<td>Unreal</td>
</tr>
<tr>
<td>05=0.2856</td>
<td>05=1.5555</td>
<td>Unreal</td>
</tr>
<tr>
<td>06=1.0472</td>
<td>06=1.1669</td>
<td>Unreal</td>
</tr>
</tbody>
</table>

Note: In singular case some solutions are not same with real angle, because there are infinite solutions in singular case and one solution is found from infinite solutions.

VIII. CONCLUSION

We presented a new inverse kinematic solution by using screw theory. In this method dual-quaternion is used as a screw operator. And also screw theory and homogeneous transformation approaches are investigated and compared with respect to singularity, computational efficiency and accuracy. Screw theory is a singularity avoiding method but homogeneous transformation is not. And also screw theory is more efficiency and more accurate than homogeneous transformation. Nevertheless homogenous transformations with DH convention applications are more common than screw theory. Because point transformation can be understood easier than line transformation, its mathematical substructure is simpler than screw theory and also it is well defined method.

REFERENCES