IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 30, 2019, accepted November 10, 2019, date of publication November 14, 2019,
date of current version November 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2953553

Analytical Solution for Inverse Kinematics Using Dual Quaternions

PING-FENG LIN, MING-BAO HUANG, AND HAN-PANG HUANG, (Member, IEEE)
Department of Mechanical Engineering, National Taiwan University, Taipei 10617, Taiwan
Corresponding author: Han-Pang Huang (hanpang@ntu.edu.tw)

ABSTRACT This paper presents a new solution for inverse kinematics (IK) by using dual quaternions (DQ) as operators and combining them with the Paden–Kahan subproblem to determine the analytical solution. Kinematics under the structure of screw theory makes the process mathematically efficient and gives geometric meaning. Both derived results and simulation outgrowths, when compared with the generally used numerical solution and matrix-based analytical solution, show that our method is faster and does not suffer from the numerical instability caused by being near to a singular configuration. The method has been proven to have the advantages of reducing the computational load and promoting precision, and it can provide multiple choices for IK.

INDEX TERMS Analytical solution, dual quaternions (DQ), inverse kinematics (IK), Paden–Kahan subproblems, screw theory.

I. INTRODUCTION

Robot kinematics, the fundamental element of robotics, describes the relationship between joint displacements and end-effector motion. Multiple methods are used to deal with this discipline. Conventionally, the most famous method, Denavit–Hartenberg (D–H) [1], is popular for its concise description of the kinematic relationship of joint motion and links state, and among the current methods, it requires the fewest parameters for identification. On the other hand, a method based on screw theory [2] is another important approach. The screw-based method associates with physical meaning to a purely geometric entity by using a screw motion to replace any rigid body motion in three-dimensional (3D) space. Its flexibility in giving free choice of reference frame and a global description of all elements make it suitable for multiple coordinate cooperating systems. However, as technology has become advanced and machines are able to catch up with mathematically descriptive robot commands, more precise orders are eagerly demanded in the fields of medical surgery and aerospace. As a result, we should further develop a more accurate and faster methodologies. The inverse kinematics (IK) is the procedure to get joint space angles from Cartesian coordinates and is usually difficult to solve analytically. To accomplish this goal, a mathematical operator called dual quaternions (DQ) is proposed instead of the traditionally used matrix and is combined with an analytical solution using Paden–Kahan subproblems. Dual quaternions, as an extension of quaternions, takes into account the translation its predecessor lacked by using dual numbers, and it has several advantages. First, it is a compact and geometrically meaningful method of representing all kinds of motion that can be operated unanimously and simultaneously. Second, there are no singularity issues when representing 3D rotations, in contrast to local coordinates such as Euler angles. Third, quaternion-based methods have shown to have better computational efficiency than the other methods, including homogeneous coordinates [3], [4]. Moreover, DQ can treat point and line transformations in the same manner [5].

The study [6] did not use DQ as the basis of kinematics to derive a new analytical Jacobian matrix. In addition, a large number of computation resources were used while many operations only took the real part and the next seven bases were not used. It is a numerical method and cannot be used for comparison. Although the use of DQ to describe the kinematics reduced the computation time compared to the matrix approach, it still used the resultant principle [7], [8]. It may be used on all configured robots, however, it is not suitable for implementation. On the contrary, the proposed DQ approach lies between the resultant principle and the geometric analytical solution for the robot. It neither completely re-analyzes

The associate editor coordinating the review of this manuscript and approving it for publication was Yangmin Li.

the robot's geometrical configuration nor requires a lot of computation. In particular, it can be easily applied to many robots.

The study conducted in [9] for solving redundant robots used arm-angle to define additional degrees of freedom, derived an augmented Jacobian matrix, and proposed a kinematic analysis. Finding the singularity of the robot itself and the singularity of the algorithm in the so-called special case actually has global significance. The study in [10] aimed to improve the singularity of the algorithm used in [9], and a double reference plane was proposed. Since the elbow cannot be located on two reference planes at the same time, the reference plane can be changed appropriately to avoid the algorithm singularity. The pure geometric analytical kinematics used in that paper are included in our comparison.

Finally, since DQ allow the elegant representation of a screwing motion, they are the most suitable operators to apply screw theory. There have already been some researches conducted into combining DQ with screw theory in robotics [11]. However, the analytical solution of using Paden–Kahan subproblems also based on the screw theory. Inverse kinematics can have global geometric meaning only when the kinematic system is described under the structure of a screw. Then this technique can be implemented to decompose the coupled joint displacements into independent geometric problems, giving an intuitive and stable solution. Therefore, DQ are suitable for an analytical method based on Paden–Kahan subproblems. The following section shows the basic knowledge and formulations of DQ and the Paden–Kahan subproblems. In Section III, the kinematics structured by DQ are shown step by step. Section IV presents the main idea of this paper, the analytical solution based on DQ. Section V compares the computation complexities and simulation results, and finally, the conclusion is presented in Section VI.

II. PRELIMINARY KNOWLEDGE

A. DUAL QUATERNIONS (DQ)

Dual quaternions can be represented in the following form:

$$\hat{q} = (\hat{q}_S, \hat{\mathrm{q}}_\mathrm{V}) \text{ or } \hat{q} = q + \varepsilon q^O. \tag{1}$$

In the left expression, $\hat{q}_S = q_S + \varepsilon q_S^O$ is a dual scalar and $\hat{\boldsymbol{q}}_\mathbf{V} = \mathbf{q}_\mathbf{V} + \varepsilon \mathbf{q}_\mathbf{V}^O$ is a dual vector, while in the right expression, q and q^O are both quaternions, and ε is the dual factor [12], [13] with a property of $\varepsilon^2 = 0$. The sum and the multiplication of two DQ are shown as follows:

$$\begin{aligned} \hat{q}_a + \hat{q}_b &= (q_a + q_b) + \varepsilon(q_a^o + q_b^o), \\ \hat{q}_a \hat{q}_b &= (q_a q_b) + \varepsilon(q_a q_b^o + q_a^o q_b). \end{aligned} \tag{2}$$

There are numerous conjugate forms that exist for DQ, and which form to use is determined by the operation needed [14], [15]. The first type of conjugate is formulated by taking a conjugate to all the quaternions terms that exist in the DQ, resulting in the following:

$$\hat{q}^* = q^* + \varepsilon q^{o*}. \tag{3}$$

The second kind of conjugate is formulated by making the coefficients in the dual part change their sign as shown in (4),

$$\bar{\hat{q}} = q - \varepsilon q^o, \tag{4}$$

while the third kind of conjugate is the combination of the previous two conjugates:

$$\bar{\hat{q}}^* = q^* - \varepsilon q^{o*}. \tag{5}$$

The norm of a DQ is also a dual number [16] and is computed using the following, (6):

$$\left\|\hat{q}\right\| = \|q\| + \varepsilon\left(\frac{q^* q^o + q^{o*} q}{2\,\|q\|}\right). \tag{6}$$

In addition, there exists one kind of DQ defined as a unit DQ [17], [18]. The condition for a unit DQ is that their norm should be with length 1. Thus, based on the above definition, the conditions can be separated into two parts:

$$\text{Real part: } \|q\| = 1, \tag{7}$$

$$\text{Dual part: } q^* q^O + q^{O^*} q = 0. \tag{8}$$

The next section introduces DQ that represent the operation of movements, as well as points and lines, and are all unitary. Therefore, they have two additional constraints that reduce their dimensions from 8 to 6.

The matrix operation is only used to calculate the relative pose in terms of two lines and only acts as a linear algebraic tool to solve the equations generated by the comparison coefficients, rather than an algorithm of the DQ kinematics. As a complex number with multiplication equal to zero, the dual number can be used to retain the real part and the dual part with the spiral theorem. The special operation principle performs both the rotation and translation simultaneously. Further, it uses a small amount of memory and consumes less computational power.

B. PADEN–KAHAN SUBPROBLEMS

Paden–Kahan subproblems, originally presented by Paden [19], using geometric algorithms to simplify the encountered problems. The three Paden–Kahan subproblems used in this study will be described later. In the structure of screw-based kinematics, it can be imagined that the end-effector is being spirally moved along each joint axis sequentially and finally reaches the current position and orientation. Therefore, it is necessary to make the screw motions independent of each other. The IK can be solved via the following steps based on the three Paden–Kahan subproblems:

1) Find the points that are the intersections of two or more axes.
2) Apply motions to these points and ignore the influence of axes that go through the selected points (motion has no effect on the point that is on the axis.) If subproblems 1 and 2 can be formulated here, proceed to step 4; otherwise, go to step 3.
3) Subtract another point on the rest of the axes' intersections to form subproblem 3.

4) Use the solution described in the following three subproblems to find the joint angles, substitute the known angles back into the original equation, and rearrange it. Then, restart from step 1 and repeat until there are no unknowns.

The three Paden–Kahan subproblems used here are the following.

Subproblem 1: A point p is rotated along a screw axis ξ (direction ω) for angle θ until it is coincident with another point q, as shown in Fig. 1.

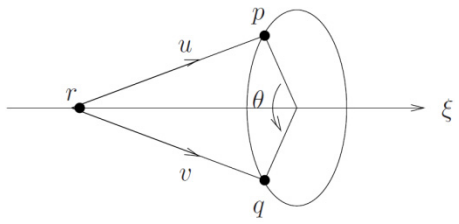


FIGURE 1. Geometric representation of subproblem 1.

The following equations can be used to solve this problem. First, we define

$$u = p - r \text{ and } v = q - r, \tag{9}$$

where r is an arbitrary point on the axis, and their projections on the direction ω are

$$u' = u - \omega\omega^T u \text{ and } v' = v - \omega\omega^T v. \tag{10}$$

Then (11) is used to obtain the unknown angle θ. It returns back one solution, no solution, or an infinite number of solutions.

$$\theta = \text{a}\tan 2(\omega^T(u' \times v'), u'^T v'). \tag{11}$$

Subproblem 2: A point p is rotated along a screw axis ξ_2 (direction ω_2) for angle θ_2 and then rotated along another screw axis ξ_1 (direction ω_1) for angle θ_1 until it is coincident with some point q, as shown in Fig. 2.

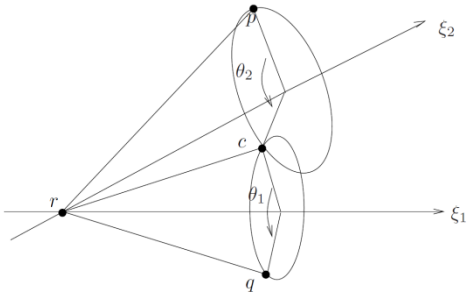


FIGURE 2. Illustration of subproblem 2.

This can be solved by using the following equations. First, we define

$$u = p - r, \quad v = q - r, \text{ and } z = c - r, \tag{12}$$

where r is an arbitrary point on the axis, and c is the relay point whose information is unavailable. Since ω_1, ω_2, and $\omega_1 \times \omega_2$ are linearly independent,

$$z = \alpha\omega_1 + \beta\omega_2 + \gamma(\omega_1 \times \omega_2). \tag{13}$$

Furthermore, by using

$$\alpha = \frac{(\omega_1^T\omega_2)\omega_2^T u - \omega_1^T v}{(\omega_1^T\omega_2)^2 - 1} \text{ and } \beta = \frac{(\omega_1^T\omega_2)\omega_1^T v - \omega_2^T u}{(\omega_1^T\omega_2)^2 - 1}, \tag{14}$$

we can obtain (15),

$$\gamma^2 = \frac{\|u\|^2 - \alpha^2 - \beta^2 - 2\alpha\beta\omega_1^T\omega_2}{\|\omega_1 \times \omega_2\|^2}. \tag{15}$$

Once γ is determined, z, and hence c, can also be determined. Then the problem can be decomposed into subproblem 1 for each. They return one group of answers, two groups of answers, an infinite number of groups of answers, or no answer, and each group of answers contains two angles.

Subproblem 3: A point p is rotated along a screw axis ξ (direction ω) for angle θ until it has a specific distance δ with some point q, as shown in Fig. 3.

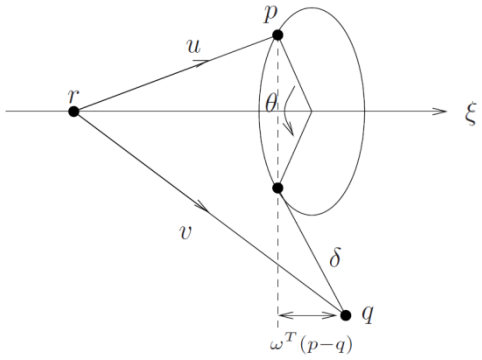


FIGURE 3. Illustration of subproblem 3.

First, we define

$$u = p - r \text{ and } v = q - r, \tag{16}$$

where r is an arbitrary point on the axis. Then, their projections are:

$$u' = u - \omega\omega^T u, \quad v' = v - \omega\omega^T v, \text{ and } \delta'^2 = \delta^2 - |\omega^T(p-q)|^2. \tag{17}$$

Let θ_0 be the angle between the vectors u' and v'. Thus,

$$\theta_0 = \text{a}\tan 2(\omega^T(u' \times v'), u'^T v'). \tag{18}$$

The unknown angle θ_2 can then be derived from the following:

$$\theta = \theta_0 \pm \cos^{-1}\left(\frac{\|u'\|^2 + \|v'\|^2 - \delta'^2}{2\|u'\|\|v'\|}\right), \tag{19}$$

which returns one answer, two answers, or no answer.

Note that subproblems sometimes provide more than one answer, so multiple combinations of solutions can be recorded until they are fit under a suitable condition.

Points and lines are the basis for geometric objects in Clifford algebra. To understand its application to robotics, the operation can be found in the Clifford algebra of points, lines and planes, and in the Points, Lines, Screws and Planes

in DQ kinematics. However, a problem arises in the description of a posture with an upward movement. A posture can be determined by two perpendicular lines. This information can be further combined with the Paden–Kahan subproblems to provide a DQ analytical solution for IK. Based on that information, a more suitable representation to describe the pose and properties of the rigid body motion to formulate the original DQ analytical solution is proposed, and hence to reduce the computation load so that it can replace the matrix approach.

III. FORWARD KINEMATICS USING DUAL QUATERNIONS

In this section, the formula for transforming screw motion into the form of DQ is introduced first. Later, we present how these motions can be applied to both points and lines. Last, the process to derive forward kinematics (FK) using DQ as operators is presented. The FK must be understood first in order to comprehend the subsequent IK.

A. SCREW MOTION

Screw motion consists of rotation of a part [20] around a straight line, followed by a translation parallel to that line. According to screw theory [21], all movements in 3D space can be equated to a screw motion, and screw motion can be realized with DQ by using the following components: a unitary rotation axis $\mathbf{n}$ that does not necessarily pass through the origin, a rotation angle θ with a translation distance d along $\mathbf{n}$, and a vector that denotes that there is a displacement between $\mathbf{n}$ and the origin, which is represented as $\mathbf{a}$ (an arbitrary vector point from origin to $\mathbf{n}$). This third component is an important characteristic because the screw motion formalism can be used to describe rotations about eccentric points, while quaternions and rotation matrices cannot be used in this manner. A screw motion can be represented by the following DQ:

$$\hat{q}_s = (\cos\frac{\theta}{2} + \mathbf{n}\sin\frac{\theta}{2}) + \varepsilon(-\frac{d}{2}\sin\frac{\theta}{2} + \mathbf{n}\frac{d}{2}\cos\frac{\theta}{2} + (\mathbf{a}\times\mathbf{n})\sin\frac{\theta}{2}). \quad (20)$$

B. APPLICATION TO POINTS AND LINES

Before operating on points and lines, a suitable representation of them under a DQ structure must be determined. According to the research performed by [22], a 3D physical point p with coordinates $\mathbf{x}$ is represented by a DQ $\mathrm{P} = 1 + \varepsilon\mathbf{x}$. Lines are known to be able to be represented in Plücker coordinates by specifying the line orientation $\mathbf{n}$ and a moment term $\mathbf{m}$ that comes from the cross product of $\mathbf{n}$ with an arbitrary vector $\mathbf{a}$ of a direction from the origin point to $\mathbf{n}$. A DQ line L is written as follows:

$$L = \mathbf{n} + \varepsilon(\mathbf{a}\times\mathbf{n}) = \mathbf{n} + \varepsilon\mathbf{m}. \quad (21)$$

After representation has been shown, the operations can be introduced. DQ operate points and lines in two ways. The first is by left-multiplying the conjugate of the operator and then right-multiplying the operator. This is the case if the new position and orientation of points and lines under the same reference frame are examined, which in this case is called *active movement*. The second way is the opposite, by left-multiplying the operator and then right-multiplying the conjugate of the operator, where the effect equals a frame transformation (i.e., the position and orientation of the same points and lines are under a different reference frames), which in this case is called *passive movement*. The two movements are explained through two examples also showing the slight difference between operating on lines and on points. The first objective is to screw a point and find its coordinates after moving in the original reference frame. The active movement can be realized through the following equations (for a point operation, a mixed conjugate is used):

$$P' = \bar{\hat{q}}_s^* P \hat{q}_s, \quad (22)$$

where $\hat{q}_s$ denotes the screw motion that is applied to the point, and the capital word means it is a DQ.

Next, if a line described from a different reference frame is examined, passive movement should be used. In a similar manner, it can be realized through the following operation (for a line operation, an ordinary conjugate should be used).

$$L' = \hat{q}_s L \hat{q}_s^*. \quad (23)$$

Again, all DQ, including screw motion, points, and lines, are unitary. In other words, they should follow the constraints mentioned in (7) and (8), which help verify the correctness of applications.

C. FORWARD KINEMATICS (FK)

In this section, the derivation for FK using DQ is presented. It is an extension of the knowledge in [23], which presents a treatment of the theory of screws.

To formulate the FK of a serial robot, the first step is to determine the joint axis vectors $\mathbf{n}$ and the moment vectors $\mathbf{m}$ (or $\mathbf{a}\times\mathbf{n}$). The second step is to generate screw motion operators by substituting $\mathbf{n}$, $\mathbf{m}$, d, and θ into (20) with respect to each joint number. Finally, all operators are multiplied sequentially to obtain the "total transformation operator." The following general case may help readers understand the entire process, supposing the joint number is notated as i.

1) For each i, define the joint axis as $\mathbf{n}_i$ and then arbitrarily choose a point $\mathbf{a}_i$ on the axis $\mathbf{n}_i$.
2) Use (20) to formulate the motion operator for each joint and obtain the equations shown below.

$$\hat{q}_{si} = (\cos\frac{\theta}{2} + \mathbf{n}_i\sin\frac{\theta}{2}) + \varepsilon(-\frac{d}{2}\sin\frac{\theta}{2} + \mathbf{n}_i\frac{d}{2}\cos\frac{\theta}{2} + (\mathbf{a}_i\times\mathbf{n}_i)\sin\frac{\theta}{2}). \quad (24)$$

3) Left-multiply operators sequentially, starting from step 1, resulting in a total motion operator $\hat{q}_{tot}$.

$$\hat{q}_{tot} = \hat{q}_{si}\cdots\hat{q}_{s3}\hat{q}_{s2}\hat{q}_{s1}. \quad (25)$$

After the total transformation operator has been obtained, the end-effector position can be found by using (21).

$$P' = \overline{\hat{q}_{tot}^*} P \hat{q}_{tot}, \quad (26)$$

where p is the initial position of the end-effector. For orientation, two lines that are perpendicular to each other and intersect at the end-effector initial position p must be well-defined. Equation (23) is then applied to obtain the following equations, assuming initial lines are parallel with the x- and y-axes, respectively:

$$L'_x = \hat{q}_{tot} L_x \hat{q}^*_{tot} \text{ and } L'_y = \hat{q}_{tot} L_y \hat{q}^*_{tot}. \tag{27}$$

Background knowledge and application of DQ to FK have been shown in this section. The next section will present the analytical solution of IK using DQ.

IV. ANALYTICAL SOLUTION FOR INVERSE KINEMATICS USING DUAL QUATERNIONS

This section will describe the line-based analytical solution and the configuration-based analytical solution for IK using DQ.

A. LINE-BASED ANALYTICAL SOLUTION

According to the precedent knowledge, the relationship between two configurations, also known as the total motion operator, can be found through the following steps. First, the relationship of one line to the total motion operator is examined, which can be described in (28) and then rearranged in (29) by right-multiplying a conjugate of the total motion operator.

$$\hat{q}^*_{n1} L \hat{q}_{n1} = L', \tag{28}$$
$$\hat{q}^*_{n1} L = L' \hat{q}^*_{n1}. \tag{29}$$

Now, assume $\hat{q}_{n1} = [a, b, c, d, e, f, g, h]$, $L = [0, u_1, v_1, w_1, 0, x_1, y_1, z_1]$, and $L' = [0, u_2, v_2, w_2, 0, x_2, y_2, z_2]$. Equation (29) can then be expanded and arrayed into (30), where $\bar{u}_1 = u_1 - u_2$ and $\bar{u}_2 = -u_1 - u_2$, which is true for the other symbols. The null space of the matrix on the left side in (30) is the solution of $\hat{q}_{n1}$.

$$\begin{bmatrix} 0 & \bar{u}_1 & \bar{v}_1 & \bar{w}_1 & 0 & 0 & 0 & 0 \\ \bar{u}_1 & 0 & -\bar{w}_2 & \bar{v}_2 & 0 & 0 & 0 & 0 \\ \bar{v}_1 & \bar{w}_2 & 0 & -\bar{u}_2 & 0 & 0 & 0 & 0 \\ \bar{w}_1 & -\bar{v}_2 & \bar{u}_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \bar{x}_1 & \bar{y}_1 & \bar{z}_1 & 0 & \bar{u}_1 & \bar{v}_1 & \bar{w}_1 \\ \bar{x}_1 & 0 & -\bar{z}_2 & \bar{y}_2 & \bar{u}_1 & 0 & -\bar{w}_2 & \bar{v}_2 \\ \bar{y}_1 & \bar{z}_2 & 0 & -\bar{x}_2 & \bar{v}_1 & \bar{w}_2 & 0 & -\bar{u}_2 \\ \bar{z}_1 & -\bar{y}_2 & \bar{x}_2 & 0 & \bar{w}_1 & -\bar{v}_2 & \bar{u}_2 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{30}$$

However, the number of degrees of freedom (DOFs) is greater than needed. This situation has been expected, as previously said, and the information from another line should be used. After doing the same thing on the second line, another equation similar to (30) is obtained, but the subscripts under the symbols are changed from 1 to 3, and 2 to 4. For convenience, the left-hand side is called *matrix x*, and the resemble matrix is called *matrix y* (since they are originally lines parallel to the x- and y-axes, respectively). The combination of *matrix x* and *matrix y* provides an augmented matrix, but such a large matrix is not necessary, so Gaussian elimination is employed, and the two constraints are shown in (31) that come from (7) and (8) are applied to *matrix x* and *matrix y*.

$$q^T q = 1 \text{ and } q^T q^O = 0. \tag{31}$$

Equation (32) shows *matrix x* after going through the reduction. As for matrix y, only the subscript needs to be modified.

$$\begin{bmatrix} \bar{w}_1 & -\bar{v}_2 & \bar{u}_2 & 0 & 0 & 0 & 0 & 0 \\ \bar{x}_1 & 0 & -\bar{z}_2 & \bar{y}_2 & \bar{u}_1 & 0 & -\bar{w}_2 & \bar{v}_2 \\ \bar{y}_1 & \bar{z}_2 & 0 & -\bar{x}_2 & \bar{v}_1 & \bar{w}_2 & 0 & -\bar{u}_2 \\ \bar{z}_1 & -\bar{y}_2 & \bar{x}_2 & 0 & \bar{w}_1 & -\bar{v}_2 & \bar{u}_2 & 0 \end{bmatrix}. \tag{32}$$

This matrix has a maximum rank equal to 4 (6 minus 2 for rotation and translation). In addition, its first row can be changed according to the configuration used. In a more detailed explanation, (32) was generated since the x- and y-directions were chosen to build the initial frame. If y and z directions are picked instead, then the second row in *matrix x* (30) should be used as the first row to form a new equation. In the next step, (33) is built by combining the reduced *matrix x* and *matrix y*.

$$\begin{bmatrix} \bar{w}_1 & -\bar{v}_2 & \bar{u}_2 & 0 & 0 & 0 & 0 & 0 \\ \bar{x}_1 & 0 & -\bar{z}_2 & \bar{y}_2 & \bar{u}_1 & 0 & -\bar{w}_2 & \bar{v}_2 \\ \bar{y}_1 & \bar{z}_2 & 0 & -\bar{x}_2 & \bar{v}_1 & \bar{w}_2 & 0 & -\bar{u}_2 \\ \bar{z}_1 & -\bar{y}_2 & \bar{x}_2 & 0 & \bar{w}_1 & -\bar{v}_2 & \bar{u}_2 & 0 \\ \bar{w}_3 & -\bar{v}_4 & \bar{u}_4 & 0 & 0 & 0 & 0 & 0 \\ \bar{x}_3 & 0 & -\bar{z}_4 & \bar{y}_4 & \bar{u}_3 & 0 & -\bar{w}_4 & \bar{v}_4 \\ \bar{y}_3 & \bar{z}_4 & 0 & -\bar{x}_4 & \bar{v}_3 & \bar{w}_4 & 0 & -\bar{u}_4 \\ \bar{z}_3 & -\bar{y}_4 & \bar{x}_4 & 0 & \bar{w}_3 & -\bar{v}_4 & \bar{u}_4 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{33}$$

This matrix (33) has the rank of 6, because the solution q_{n1} already exists, as well as an additional orthogonal solution $[0, 0, 0, 0, a, b, c, d]$. These solutions span a 2-dimensional null space. Using a singular-value decomposition (SVD) on the left-hand side in (33), two bases of null space are obtained. Finally, two free variables are verified according to Daniilidis [11]. Two free variables, λ_1 and λ_2, are set, and two bases of null space are assumed as $\hat{n}_1 = (n_1, n_1^O)$ and

TABLE 1. Kinematic parameters of 6-Axis robot manipulator.

Joint	1	2	3	4	5	6	Unit
DH-a	0	0	10	−10	0	131	*mm*
DH-α	90	90	90	−90	90	90	*deg*
DH-d	0	0	371	0	280	0	*mm*
DH-θ	0	180	180	−90	180	90	*deg*

$\hat{n}_2 = (n_2, n_2^O)$, respectively, so that $\hat{q}_{n1} = \lambda_1\hat{n}_1 + \lambda_2\hat{n}_2$. Using the constraint of (31), we have

$$\lambda_1^2 n_1^T n_1 + 2\lambda_1\lambda_2(n_1^T n_2) + \lambda_2^2 n_2^T n_2 = 1, \quad (34)$$

$$\lambda_1^2 n_1^T n_1^O + \lambda_1\lambda_2(n_1^T n_2^O + n_2^T n_1^O) + \lambda_2^2 n_2^T n_2^O = 0. \quad (35)$$

Without loss of generality, $n_1^T n_1^O \neq 0$. Consequently, $\lambda_2 \neq 0$. Setting $c = \lambda_1/\lambda_2$ and substituting into (35) yields

$$\lambda_2^2(n_1^T n_1^O c^2 + (n_1^T n_2^O + n_2^T n_1^O)c + n_2^T n_2^O) = 0. \quad (36)$$

Furthermore, by solving this equation with $\lambda_2 \neq 0$, two values of c are obtained. Now, considering (34), substituting λ_1 into $c\lambda_2$ gives:

$$\lambda_2^2(n_1^T n_1 c^2 + 2n_1^T n_2 c + n_2^T n_2) = 1. \quad (37)$$

For stability, the value of c that provides the greatest value of the inner term of (37) is used. Then, all the unknowns become known, and $\hat{q}_{n1}$ can then be clearly determined. After we have the total motion operator, the solving process can thus be started.

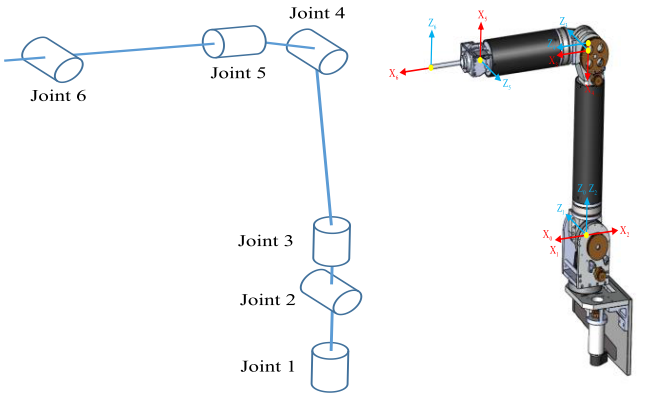


FIGURE 4. Joint configurations and standard D-H coordinates of the National Taiwan University (NTU) articulated 6-DOF robot arm.

An example is provided to help explain the analytical IK solving process. Consider the self-fabricated NTU-articulated robot arm [24]–[28] designed by our laboratory, as shown in Fig. 4. It is a 6-DOF all-revolute-joint robot arm, with a robot configuration of SRRR (S-spherical, R-revolute), with kinematic parameters of (a, α, d, θ), as shown in Table 1.

q_{61} is defined as the total motion operator, and its value can be obtained through the method mentioned earlier in this section. The beginning equation can be written by applying q_{61} on a point P_{56} (the subscript is the axis that has passed through it). This operation generates a new coordinate P'_{56} and provides (38).

$$\bar{\hat{q}}_{61}^* P_{56}\hat{q}_{61} = P'_{56}. \quad (38)$$

Because rigid body motions produced by revolute joint 5 and 6 have no effect on P_{56}, their motion operators can be canceled from each side, leaving only $\hat{q}_{41}$. Consequently, (38) can also be represented by (39).

$$\bar{\hat{q}}_{41}^* P_{56}\hat{q}_{41} = P'_{56}. \quad (39)$$

A point P_{123} is then subtracted at each side of (40). This allows the equation to be rearranged by pulling out the first three motion operators and sorting out an identical formulation to subproblem 3. The formulas are as follows:

$$\begin{aligned} &\bar{\hat{q}}_{41}^* P_{56}\hat{q}_{41} - \bar{\hat{q}}_{31}^* P_{123}\hat{q}_{31} \\ &\quad = P'_{56} - P_{123}, \\ &\Rightarrow \left\| \bar{\hat{q}}_4^* P_{56}\hat{q}_4 - P_{123} \right\| = \left\| P'_{56} - P_{123} \right\|. \end{aligned} \quad (40)$$

After solving subproblem 3, θ_4 and $\hat{q}_4$ become known. Next, P_{123} is operated. This time, the conjugate of the total motion operator is used, allowing the effects of the first three joints to be canceled. New coordinates and equations can be shown by (41):

$$\bar{\hat{q}}_{61} P_{123}\hat{q}_{61} = P'_{123}. \quad (41)$$

As mentioned earlier, this can also be shown as

$$\bar{\hat{q}}_{64} P_{123}\hat{q}_{64}^* = P'_{123}. \quad (42)$$

If the motion operators of axis 5 and 6 are multiplied to both sides of (42), the following is obtained:

$$\bar{\hat{q}}_4 P_{123}\hat{q}_4^* = \bar{\hat{q}}_{65}^* P'_{123}\hat{q}_{65}. \quad (43)$$

This is the standard form of subproblem 2, allowing the values of θ_5 and θ_6 to be obtained. At the present stage, θ_4, θ_5, and θ_6 are determined. For the next operation, the total operator should be transferred into q_{31} using the following equation.

$$\hat{q}_{31} = \hat{q}_4^*\hat{q}_5^*\hat{q}_6^*\hat{q}_{61}. \quad (44)$$

This equation can be applied to P_3, which is a point on axis 3 but does not go through axis 2, giving

$$\bar{\hat{q}}_{31}^* P_3\hat{q}_{31} = P'_3. \quad (45)$$

For the same reason, (45) can be represented as

$$\bar{\hat{q}}_{21}^* P_3\hat{q}_{21} = P'_3. \quad (46)$$

Equation (46) is the standard form of subproblem 2, and it can be used to find θ_1, θ_2. The last operator is the motion caused by joint 3, and its value can be derived by using the information already obtained from other angles.

$$\hat{q}_3 = \hat{q}_{31}\hat{q}_1^*\hat{q}_2^*. \quad (47)$$

There are two methods for determining θ_3, and both have the same result. One is using subproblem 1, while the other

uses (24) to compare the coefficients to verify θ_3. At this point, all the unknowns are solved. Next, we introduce a more optimized method that presents operators and configurations simultaneously.

B. CONFIGURATION-BASED ANALYTICAL SOLUTION

The previously discussed traditional method where DQ are used to manage points and lines is very inefficient. Therefore, presented in this section is another way to represent a frame configuration, which can be achieved by using (48).

$$\hat{q}_g = [q_R, \frac{1}{2} q_R q_t], \tag{48}$$

where $\hat{q}_g$ is the DQ representation of a configuration, q_R is a quaternion representation of the orientation, while q_t is a quaternion representation of the translation, and q_R and q_t can be found by

$$\hat{q}_R = \cos(\frac{\theta}{2}) + \mathbf{n}\sin(\frac{\theta}{2}), \tag{49}$$

$$\hat{q}_t = 1 + \varepsilon \frac{d}{2}\mathbf{t}, \tag{50}$$

where $\mathbf{n}$ is a three-dimensional vector that denotes the unitary rotation axis, θ denotes the rotation angle, d represents the distance, and $\mathbf{t}$ represents the direction. The configuration was formulated by combining a single rotation and a single translation DQ. Since this representation is a configuration and also a motion operator, using it to derive FK only needs right-multiplication, such as

$$\hat{q}'_g = \hat{q}_g \hat{q}_{tot}, \tag{51}$$

where $\hat{q}'_g$ is the configuration of the end-effector after the operation. Furthermore, orientation and position can be obtained by reversing the establishing process.

By employing this technique, less effort is required than the first method. The relationship between two configurations can easily be derived by left-multiplying only the conjugate of the initial configuration DQ, and $\hat{q}_{n1}$, the total motion operator, can then be found by

$$\hat{q}_{n1} = \hat{q}^*_{ginitial} \hat{q}_{gfinal}. \tag{52}$$

Also, for the solving process, this expression can help reduce computation loading. Notice that the deriving process is used to determine which point should be substituted into these subproblems. Most of the computational resource is used to find points and solve subproblems, while the subproblems are already fixed in the procedure. Therefore, the computational speed of how fast these treated points can be found is the most influential factor in favor of this method. The configuration-based method utilizes the dual property of its representation, which can be freely switched from a configuration to an operator to form a new process and successfully reduces the computational load. The new process shares nearly the same architecture with the previous process, but there is quite an improved version in computational speed.

Now let an operation $(\hat{q}_g)_p$ be defined to represent the motion of extracting position information from a DQ-represented configuration $\hat{q}_g = [q, q^O]$.

$$(\hat{q}_g)_p = \text{take the last three elements of}[2q^*(q^O)] = p, \tag{53}$$

where p is the coordinate or the origin of this configuration. By using this concept, the work presented in the previous method can be simplified. For example, (38) can be replaced by

$$(P_{g56}\hat{q}_{61})_p = p'_{56}, \tag{54}$$

which reduces about half of the computation and results in a faster calculation speed. The complete process using a configuration-based solution is summarized in Table 2.

TABLE 2. Complete process using a process configuration-based solution algorithm.

1:	function;
2:	$(P_{g56}\hat{q}_{61})_p = p'_{56}$;
3:	$\left\|(P_{g56}\hat{q}_4)_p - p_{123}\right\| = \left\|p'_{56} - p_{123}\right\|$;
4:	*return* subproblem 3; to get θ_4, $(P_{g123}\hat{q}^*_{61})_p = p'_{123}$,
5:	$(P_{g123}\hat{q}^*_4)_p = (P'_{g123}\hat{q}_{65})_p$;
6:	*return* subproblem 2; to get θ_5 and θ_6, $\hat{q}_{31} = \hat{q}_4{}^*\hat{q}_5{}^*\hat{q}_6{}^*\hat{q}_{61}$,
7:	$(P_{g3}\hat{q}^*_{31})_p = p'_3$, $(P_{g3}\hat{q}^*_{21})_p = p'_3$;
8:	*return* subproblem 2; to get θ_1 and θ_2, $\hat{q}_3 = \hat{q}_{31}\hat{q}_1{}^*\hat{q}_2{}^*$,
9:	end function;

Using subproblem 1 or comparing coefficients with (24) gives the final value of unknowns, and the entire IK has been done.

This section has shown two different ways of solving analytical solutions using DQ. Clearly, the configuration-based analytical solution is more efficient. For the following sections, the analytical method based on DQ refers to the configuration-based analytical solution.

V. SIMULATIONS AND EXPERIMENTS

To prove the efficiency and compactness of the proposed method, the computational load has been considered, and several methods are compared for both FK and IK. Although this paper proposes the inverse analytical solution using DQ, FK is still an important part because a complete kinematic will be better if operating under the same mathematical structure. The results have shown that the method of using DQ to represent configuration is the most efficient way to describe kinematics. The D–H convention method, exponential mapping method [23], and the proposed DQ method are compared, and the results are summarized in Table 3, where n is the number of joints. Clearly, the proposed method has the least computational load.

TABLE 3. Forward kinematics comparison.

Forward Kinematics	D–H Convention	Exponential Mapping	Dual Quaternions
Addition & Subtraction	48 n	95 n	65 n
Multiplication & Division	72 n	123 n	46 n
Trigonometric Function	14 n	18 n	5 n

D–H convention:

$$
{}^{i-1}_{i}A = \begin{bmatrix} c\theta_i & -c\alpha_i s\theta_i & s\alpha_i s\theta_i & a_i c\theta_i \\ s\theta_i & c\alpha_i c\theta_i & -s\alpha_i c\theta_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (55)
$$

$$
{}^{0}_{n}A = {}^{0}_{1}A^{1}_{2}A \cdots {}^{n-1}_{n}A
$$

Exponential mapping:

$$
e^{\xi\theta} = \begin{bmatrix} e^{\hat{\omega}\theta} & \left(I - e^{\hat{\omega}\theta}\right)(\omega \times v) + \omega\omega^T v\theta \\ 0 & 1 \end{bmatrix}
$$

$$
g(\theta) = e^{\xi_1\theta_1} e^{\xi_2\theta_2} \ldots e^{\xi_n\theta_n} g_{st}(0). \quad (56)
$$

Dual quaternions (DQ):

$$
\begin{aligned}
\hat{q}_{si} &= (\cos\frac{\theta}{2} + \mathbf{n}_i \sin\frac{\theta}{2}) + \varepsilon(-\frac{d}{2}\sin\frac{\theta}{2} + \mathbf{n}_i\frac{d}{2}\cos\frac{\theta}{2} \\
&\quad + (\mathbf{a}_i \times \mathbf{n}_i)\sin\frac{\theta}{2}), \\
\hat{q}_{tot} &= \hat{q}_{si} \ldots \hat{q}_{s3}\hat{q}_{s2}\hat{q}_{s1}, \\
\hat{q}'_g &= \hat{q}_g \hat{q}_{tot}. \quad (57)
\end{aligned}
$$

The equations show that in FK, the D–H convention uses the fewest number of parameters for identification but requires 16 memory units to record a matrix, and it spends moderate resources on calculation. As for exponential mapping, it has the largest loading on both memory size and computation, but it has flexibility in reference choice and represents the actual joint displacements. Equation (57) shows the DQ calculation. Although it requires more parameters than D–H to identify a joint motion, only eight parameters are used to record a rigid body motion. Moreover, the DQ calculation shows the least computational complexity, and it has the same advantage as exponential mapping due to their identical theoretical origin: screw theory. In addition, it is competitive in a situation with additional DOFs.

For IK, there is no general comparison because every analytical solution has its uniqueness. Consequently, one case is used to demonstrate the resources they will consume. Also, the most commonly used numerical solution method was assigned to be one of the selections for this comparison. In the comparison, the precision of the numerical solution was adjusted to 0.0001 (m), and the robot model is the same as the example mentioned in the previous section. Three methods (D–H Convention, Exponential Mapping, and Dual Quaternions) were used to solve a T-shaped trajectory lying parallel to the x–y plane segregated into 601 segments. To avoid singularity, the following procedure was executed ten times to generate the final result.

- ALGORITHM:

1) Given a set of Cartesian space missions.
2) Each of the group of target DQ analytical solutions and a solution with answers may be stored in the joint space.
3) Use the level of joint activity to find the best (minimum) combination.

$$
\int \dot{\theta}^T \dot{\theta} dt. \quad (58)
$$

4) For a singular point and the case corresponding to an infinite number of solutions, the average value between this step and the next step will be used for the joint space to ensure that equation (58) has a minimum value.

- FEATURES:
  1) Global optimal solution.
  2) Small number of calculations, and fast.
  3) High precision.
  4) Can deal with singular points without losing precision.
  5) It has geometric meaning without loss of generality.

In the comparison of different calculations, the exponential mapping analytical solution was used for comparison to reflect its advantages as an advanced version. The optimal solution at the singular point is guaranteed by the algorithm. The results of the experiment (compared with the numerical method) include the calculation time, the angle of the joint movement, and the angular velocity. The behavior produced by the singular point can prove the merits and effectiveness of the proposed method.

As for the experimental configuration, a Windows application with multi-thread programming was developed as the software for both simulation and real-world experimental purposes. The software development and testing environment were Microsoft® Visual Studio 2012 IDE under Microsoft® Windows 7, the programing language used was C/C++, and the GUI was built using the MFC Library. For the 3-D display, OpenGL [29] and OpenCV [30] were used for real-time computer vision and image processing. The computations related to linear algebra were consigned to the Eigen C++ library [31]. Multiple functions can be controlled through the program, including the robot control panel, trajectory planning, force/torque sensor oscilloscope, and Kinect sensor monitor. Among them, the robot control panel provided joint space and Cartesian space control, and the trajectory planning made the addition of time, position, velocity, and acceleration of a via point available and, further, was able to hybridize joint space and Cartesian space trajectories. Further, it also had various kinds of trajectory interpolations such as cubic/quantic polynomial, spline, and so on. Fig. 5 and Fig. 6 depict the main dialogue of the program interface of the multi-robot control.

Table 4 shows the derived computational load for two methods, where one is the proposed method, and the other is

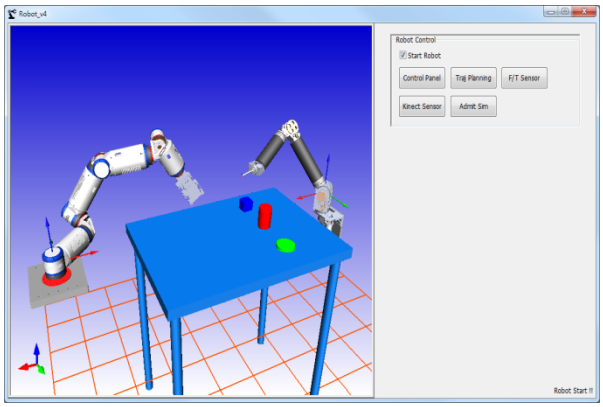


FIGURE 5. The main dialogue of the GUI application.

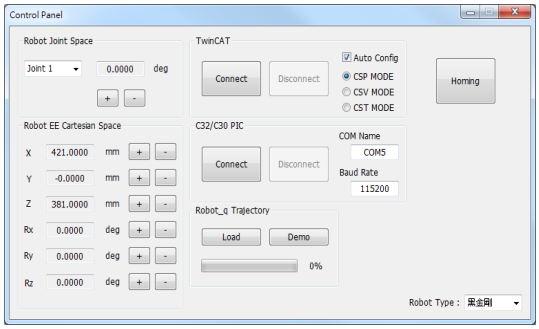


FIGURE 6. The dialogue of the robot control panel.

TABLE 4. Inverse Kinematics Comparison.

Inverse Kinematics	Exponential Mapping	Dual Quaternions
Addition & Subtraction	519	358
Multiplication & Division	663	507
Trigonometric Function	90	25

TABLE 5. Time consumption of inverse kinematics.

	Numerical Solution	Exponential Mapping	Dual Quaternions
Time consumed for solving total of 601 configurations (unit: seconds)	2.08	0.75	0.64

its predecessor (exponential mapping, an analytical solution based on the matrix). It gives the theoretical evidence to prove the high speed of the proposed method. Table 5 shows the experimental time consumed in each method, including the proposed method, its predecessor, and the generally used numerical method. Among them, the proposed method has the least time-consuming. Table 6 shows the complete comparison methodology framework. Fig. 7 demonstrates three

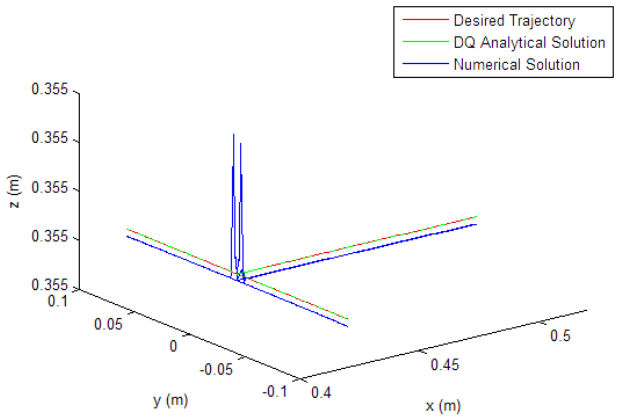


FIGURE 7. Cartesian coordinate trajectory comparison (notice that due to the z direction, variations are below 0.0001. While the z label seems to remain constant, virtually, it is varying).

trajectories in 3D space: the desired T-shaped trajectory (red), trajectories obtained by substituting joint angles calculated from the DQ analytical method (green), and the numerical method (blue) back into the FK. The conspicuous protuberance of the trajectory from the numerical method can be found near the intersection point. It is engendered since the intersection point is a singularity of the model robot. While the numerical method suffers from the singularity, the DQ analytical method visibly overlaps the desired trajectory. As a special configuration of a robot, the singular point will cause the Jacobian matrix to lose full rank and numerical instability. It can also be interpreted as the disappearance of a certain degrees of freedom in this configuration. Its importance is that when the singularity occurs, the robot is then unpredictable, and the joint may require a large angle of rotation at that instant. In real-world environments, people do not expect this behavior. The singular point in our experiment is found due to three collinear axes if the robot has to describe an Euler wrist [32], which is a singular wrist (however, we do not have the Euler wrist). Since the robots with an Euler wrist can be easily and directly analyzed geometrically, and the end-effector and body position information can be decoupled, there is no need to use the DQ approach.

Figure 8 is the animation screenshot of a robot operating the desired trajectory. The hand is mounted on the robot arm to simulate a manipulation task, although in this case, the hand [33] performs no motion. As can be seen, once the robot backed to initial condition and started the next direction motion, rotations of the first joint and the third joint were particularly large, approximately 90 degrees, for an instant. This obviously displays the situation of the robot encountering singularity. Fig. 9 simultaneously shows six simulated dimension errors (error along the x, y, and z directions and orientation errors around the x, y, and z directions) of the numerical method (blue), the analytical method based on the matrix (red), and the analytical method based on DQ (green), taken from simulation. In the figures, the green line and the red line are almost overlapping, and the abruptly ascending

TABLE 6. Methodology comparison framework.

Type	Advantages	Disadvantages
(D–H Convention) Numerical method	High versatility and workability, not affected by robot configuration.	The convergence direction depends on the initial conditions and is difficult to control. The accuracy is positively correlated with the calculation time. To avoid the singular point needs to sacrifice precision. It is often trapped in the local optimal solution.
Geometric analysis method	Extremely fast, accurate, and all possible solutions.	There is no one-to-one commonality. Highly geometric analysis demands the ability to have multi-robot configuration condition (e.g., enabling the end-effector position and orientation of the decoupling Euler wrist).
Resultant principle	Advantages of analytical solutions, robots that can correspond to all configurations.	Complicated calculation process, the use of algebraic numerical equations, high need for multi-variable linear equations, and multiple solutions.
Exponential mapping analytical solution	Using matrix operations, more familiar.	Compared with a DQ, it takes up more memory, much more computation, and slightly lower precision (at a small scale).
The proposed method (Dual Quaternions)	Advantages of using twists and wrenches: • They are the global description of the motions, in contrast to local coordinates, such as Euler angles, which may suffer from singularities due to the definition of the coordinates. • It provides a very geometric description of rigid motion which greatly simplifies the analysis of mechanisms. • It is a global description of (linear and rotational) velocities and forces, and therefore it is more convenient to use twist and wrench when considering the interactions involved in multiple coordinates, e.g., grasping, constrained motions, coordination controls, etc., compared to the conventions that treats linear and rotation terms separately or using local coordinates. • It is mathematically elegant and symmetric, e.g., the reciprocal property of twists and wrenches.	About the dual quaternions inverse kinematics, the analytical solution for a higher degree of freedom robot is expected, also, a general principle is wanted in order to ease the solving process when dealing with different types of robots. About the human-like motion allocation, it lacks a measurement to demonstrate the likeness between robot behavior under the guide of these principles and a real human's action, different types of combinations should be compared using some human likeness measurement to see which one is the best algorithm.

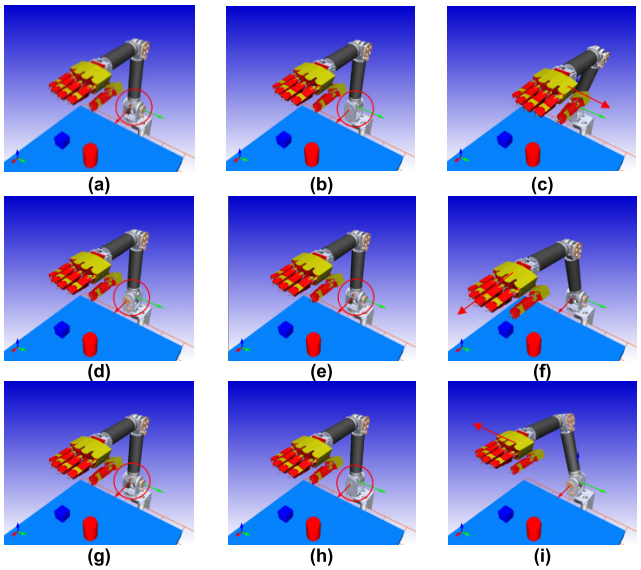
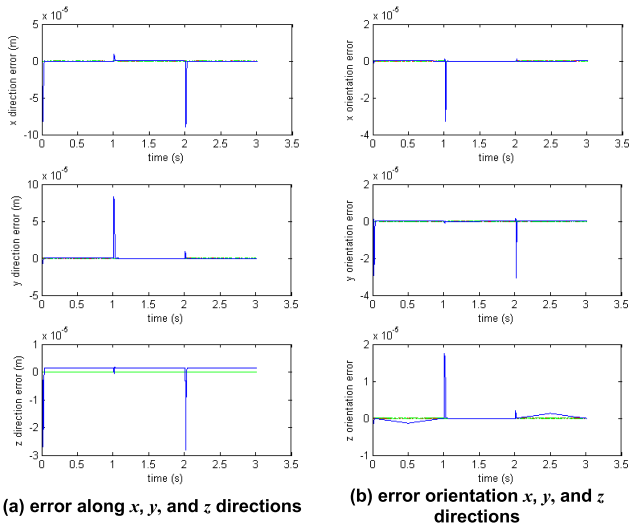


FIGURE 8. 3D display of robot demonstrating trajectory. Reading order is from left to right, top to bottom. a) Initial condition (first). b) Started to move to its left side (joint 1 and joint 3 significantly varied). c) Arrived at trajectory boundary and started to move back to initial position. d) Initial condition (second). e) Started to move forward (joint 1 and joint 3 significantly varied). f), g), h), i) Repeat above motion (to its right side).

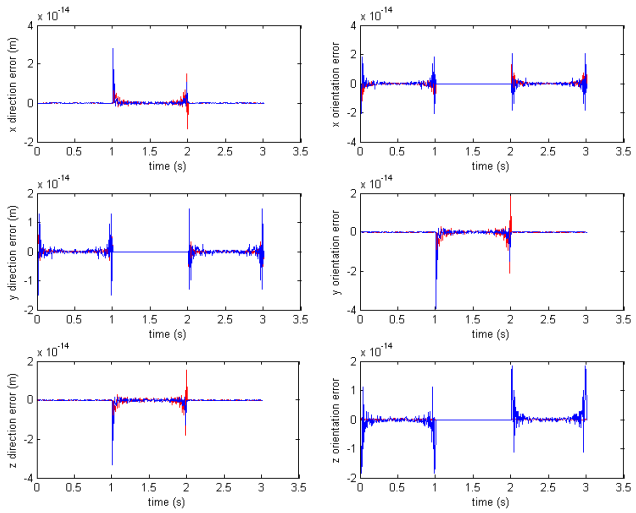


(a) error along x, y, and z directions

(b) error orientation x, y, and z directions

FIGURE 9. Error comparison of numerical method (blue), analytical method based on matrix (red), and analytical method based on dual quaternions (green).

error is due to singularity. Notice that the large discrepancies in error scale between the numerical method and other two methods can be due to the user. The precision of the numerical method can be freely adjusted, which means equivalent precision can be achieved at the expense of more computation time. In contrast, the proposed method achieves the same precision with less computation time. The difference between the numerical method and the DQ method is obvious. It may be possible to set the accuracy of the numerical method to



(a) errors along x, y, and z directions (b) error orientation x, y, and z directions

FIGURE 10. Error comparison of analytical method based on matrix (blue) and analytical method based on dual quaternions (red).

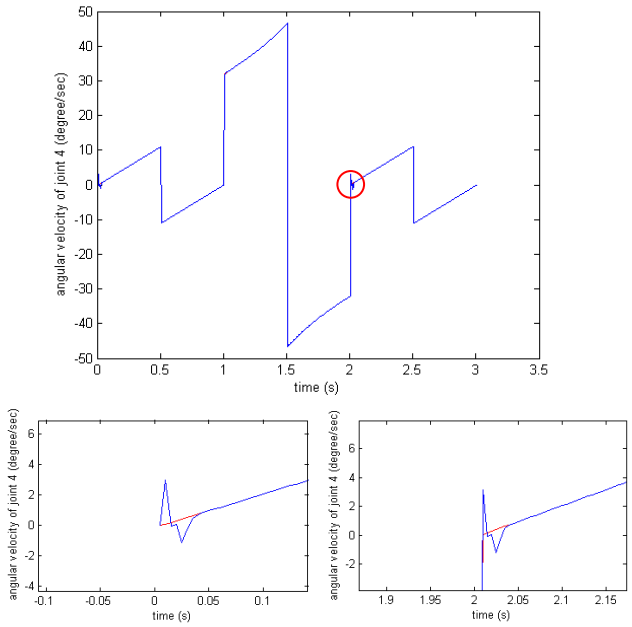


FIGURE 11. Joint 4 velocity comparison of numerical method (blue) and analytical method based on dual quaternions (red).

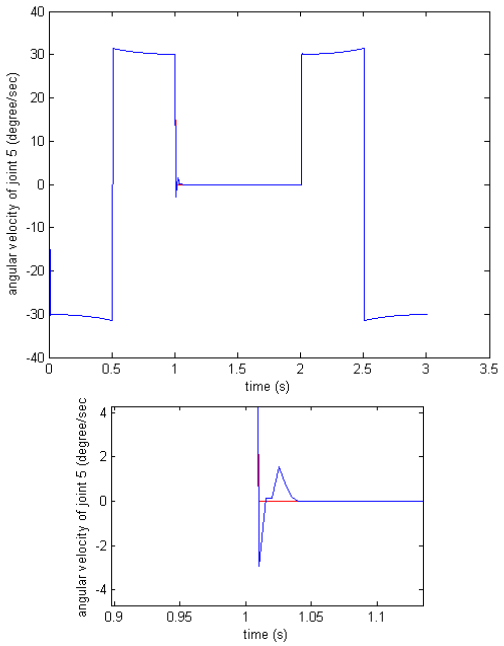


FIGURE 12. Joint 5 velocity comparison of numerical method (blue) and analytical method based on dual quaternions (red).

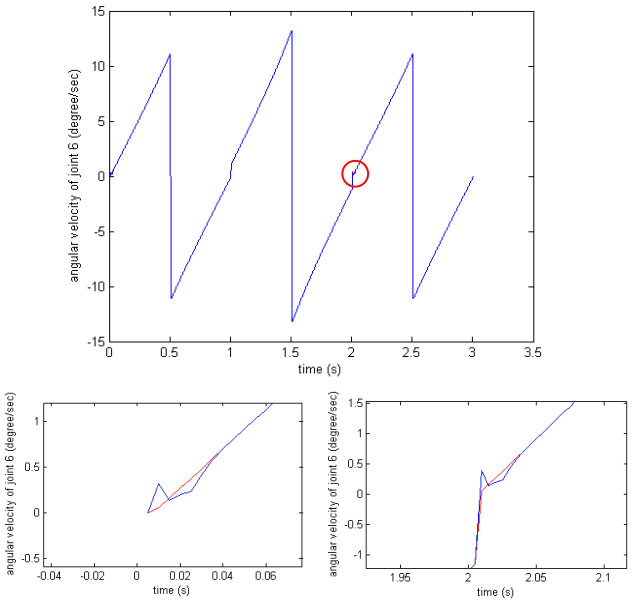


FIGURE 13. Joint 6 velocity comparison of numerical method (blue) and analytical method based on dual quaternions (red).

a relatively comparable level or to take a logarithm on the Y-axis of the chart to compare. Nonetheless, the pros and cons of accuracy and computation time are already obvious.

Fig. 10 gives us the infinitesimal error comparison between the analytical method based on a matrix (blue) and the analytical method based on DQ (red). Although errors of DQ are greater, it is still safe to say that the errors of the DQ method have less magnitude than those of the matrix methods. Fig. 11, Fig. 12, and Fig. 13 show the anomalous oscillations (encompassed by red circle) that occurred at singularity in the velocity aspect of the numerical method at joints 4, 5, and 6. Here, velocities are the average velocities in each sampling time interval (5*ms*). Again, the proposed method avoided an unstable situation with a stable solving process.

VI. CONCLUSION

This paper proposes a combination of DQ and Paden–Kahan subproblems to form an analytical solution. The process started with background knowledge of DQ and Paden–Kahan

subproblems, continued with DQ-based kinematics, and then introduced two kinds of operations under the frame of DQ for solving an NTU-articulated robot arm IK problem, and the proposed configuration method proved to be better. Finally, the computational loads of our proposed method were compared with other methods to support our viewpoint. A trajectory simulation showed that our method using DQ was better than other traditional methods in constructing kinematics for robots. Specifically, the proposed method had the fastest speed, the best precision, the fewest singularity influences, and the advantage of the availability of recording every possible joint angle as the answer of IK for one configuration. Conversely, the numerical solution only provided one answer. However, even if the above results look excellent, there are still some shortcomings in this method. One is that analytical solutions do not always exist if the target robot does not have enough axis intersection, which limits the extensive use of this method. Another problem corresponds to the solving process, which is too specific to have the skill easily applied to its robot (every robot's different analytical solution process requires users to derive these themselves). However, once the above two constraints are no longer valid in an application, an analytical solution based on DQ is a worthy technique.

Since the initial tests were conducted, the breadth and depth of this method have been improved and more algorithms have been applied. The method has proved to be more robust and effective, addressing the singular importance of rising global demand for freedom of choice for computing speed.

REFERENCES

[1] J. Denavit and R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," *J. Appl. Mech.*, vol. 22, no. 2, pp. 215–221, Jun. 1955.

[2] L.-W. Tsai, *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*. New York, NY, USA: Wiley, Feb. 1999.

[3] J. Funda, R. H. Taylor, and R. P. Paul, "On homogeneous transforms, quaternions, and computational efficiency," *IEEE Trans. Robot. Autom.*, vol. 6, no. 3, pp. 382–388, Jun. 1990.

[4] J. Funda and R. P. Paul, "A computational analysis of screw transformations in robotics," *IEEE Trans. Robot. Autom.*, vol. 6, no. 3, pp. 348–356, Jun. 1990.

[5] N. A. Aspragathos and J. K. Dimitros, "A comparative study of three methods for robot kinematics," *IEEE Trans. Syst., Man, Cybern., Syst. B, (Cybern.)*, vol. 28, no. 2, pp. 135–145, Apr. 1998.

[6] A. S. De Oliveira, E. R. De Pieri, and U. F. Moreno, "A new method of applying differential kinematics through dual quaternions," *Robotica*, vol. 35, no. 4, pp. 907–921, Apr. 2017.

[7] D. Gan, Q. Liao, S. Wei, J. S. Dai, and S. Qiao, "Dual quaternion-based inverse kinematics of the general spatial 7R mechanism," *Proc. Inst. Mech. Eng. C, J. Mech. Eng. Sci.*, vol. 222, no. 8, pp. 1593–1598, Aug. 2008.

[8] S. Qiao, Q. Liao, S. Wei, and H.-J. Su, "Inverse kinematic analysis of the general 6R serial manipulators based on double quaternions," *Mechanism Mach. Theory*, vol. 45, no. 2, pp. 193–199, Feb. 2010.

[9] K. Kreutz-Delgado, M. Long, and H. Seraji, "Kinematic analysis of 7-DOF manipulators," *Int. J. Robot. Res.*, vol. 11, no. 5, pp. 469–481, Oct. 1992.

[10] L. Yan, Z. Mu, and W. Xu, "Analytical inverse kinematics of a class of redundant manipulator based on dual arm-angle parameterization," in *Proc. IEEE Int. Conf. Syst. Man Cybern. (SMC)*, San Diego, CA, USA, Oct. 2014, pp. 3744–3749.

[11] K. Daniilidis, "Hand-eye calibration using dual quaternions," *Int. J. Robot. Res.*, vol. 18, no. 3, pp. 286–298, Mar. 1999.

[12] D. Kim, "Dual quaternion application to kinematic calibration of wrist-mounted camera," *J. Robotic Syst.*, vol. 13, no. 3, pp. 153–162, Mar. 1996.

[13] R. Mukundan, "Quaternions: From classical mechanics to computer graphics, and beyond," in *Proc. 7th Asian Technol. Conf. Math.*, Malaysia, Dec. 2002, pp. 97–106.

[14] Sir William Rowan Hamilton L L. D. V. P. R. I. A. F. R. A. S, "XI. On quaternions; or on a new system of imaginaries in algebra," *London, Edinburgh, Dublin Philos. Mag. J. Sci. Ser.*, vol. 33, no. 219, pp. 58–60, 1848.

[15] B. A. Rosenfeld, *A History of Non-Euclidean Geometry: Evolution of the Concept of a Geometric Space (Studies in the History of Mathematics and Physical Sciences)*, 1st ed. New York, NY, USA: Springer, 1988.

[16] L. Kavan, S. Collins, J. Žára, and C. O'Sullivan, "Geometric skinning with approximate dual quaternion blending," *ACM Trans. Graph.*, vol. 27, no. 4, p. 105, Oct. 2008.

[17] W. R. Hamilton, *Elements of Quaternions*, 3rd ed. New York, NY, USA: Chelsea, Mar. 1869.

[18] J. C. K. Chou, "Quaternion kinematic and dynamic differential equations," *IEEE Trans. Robot. Autom.*, vol. 8, no. 1, pp. 53–64, Feb. 1992.

[19] B. Paden, "Kinematics and control robot manipulators," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Univ. California, Oakland, CA, USA, 1986.

[20] J. C. Hart, G. K. Francis, and L. H. Kauffman, "Visualizing quaternion rotation," *ACM Trans. Graph.*, vol. 13, no. 3, pp. 256–276, Jul. 1994.

[21] R. S. Ball, *A Treatise on the Theory of Screws (Cambridge Mathematical Library)*. New York, NY, USA: Cambridge Univ. Press, Sep. 1998.

[22] E. Bayro-Corrochano, "Modeling the 3D kinematics of the eye in the geometric algebra framework," *Pattern Recognit.*, vol. 36, pp. 2993–3012, Dec. 2003.

[23] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, 1st ed. New York, NY, USA: CRC Press, Mar. 1994.

[24] M.-B. Huang and H.-P. Huang, "Innovative human-like dual robotic hand Mechatronic design and its chess-playing experiment," *IEEE Access*, vol. 7, pp. 7872–7888, 2019.

[25] C.-A. Cheng and H.-P. Huang, "Learn the Lagrangian: A vector-valued RKHS approach to identifying Lagrangian systems," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 3247–3258, Dec. 2016.

[26] C.-A. Cheng, H.-P. Huang, H.-K. Hsu, W.-Z. Lai, and C.-C. Cheng, "Learning the inverse dynamics of robotic manipulators in structured reproducing Kernel Hilbert space," *IEEE Trans. Cybern.*, vol. 46, no. 7, pp. 1691–1703, Jul. 2016.

[27] Y.-R. Liu, M.-B. Huang, and H.-P. Huang, "Automated grasp planning and path planning for a robot hand-arm system," in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, Paris, France, Jan. 2019, pp. 92–97.

[28] W.-Y. Lee, M.-B. Huang, and H.-P. Huang, "Learning robot tactile sensing of object for shape recognition using multi-fingered robot hands," in *Proc. 26th IEEE Int. Symp. Robot Hum. Interact. Commun. (RO-MAN)*, Lisbon, Portugal, Aug./Sep. 2017, pp. 1311–1316.

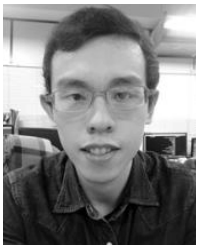
[29] OpenGL Coding Resources. (2018). *OpenGL—The Industry's Foundation for High Performance Graphics*. [Online]. Available: https://www.opengl.org/resources/

[30] (2018). *OpenCV Documentation Index Team*. [Online]. Available: https://docs.opencv.org/

[31] B. Jacob and G. Guennebaud. (May 10, 2017). *Eigen Library Official Website*. [Online]. Available: http://eigen.tuxfamily.org

[32] Y. Aydin and S. Kucuk, "Quaternion based inverse kinematics for industrial robot manipulators with Euler wrist," in *Proc. IEEE Int. Conf. Mechatronics*, Budapest, Hungary, Jul. 2006, pp. 581–586.

[33] S.-Q. Ji, M.-B. Huang, and H.-P. Huang, "Robot intelligent grasp of unknown objects based on multi-sensor information," *Sensors*, vol. 19, no. 7, p. 1595, Apr. 2019.



PING-FENG LIN received the B.S. degree in mechanical engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2016, and the M.S. degree in mechanical engineering from National Taiwan University, Taipei, Taiwan, in 2018.

His current research interests include grasping and motion control.

MING-BAO HUANG received the B.S. degree in mechanical engineering from Chinese Culture University, Taipei, Taiwan, in 2009, and the M.S. degree in engineering science from National Cheng Kung University, Tainan, Taiwan, in 2011. He is currently pursuing the Ph.D. degree in mechanical engineering with the NTU Robotics Laboratory, Department of Mechanical Engineering, National Taiwan University, Taipei.

His current research interests include human and robotic grasping and dexterous manipulation, mechanisms and machine design, anthropomorphic robotic hand design, sensors for artificial hands, human–robot interaction, and machine learning and pattern recognition. He was a recipient of the 2018 IEEE Access Best Multimedia Award Part 2 and the Best Paper Award from Conference on Automation Technology, in 2014 and 2016. He is a Reviewer of IEEE Access.

HAN-PANG HUANG (S'83–M'86) received the M.S. and Ph.D. degrees in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 1982 and 1986, respectively.

Since 1986, he has been with National Taiwan University, Taipei, Taiwan, where he is currently a Professor with the Department of Mechanical Engineering and the Graduate Institute of Industrial Engineering and serves as the Director of the NTU Robotics Laboratory. He has been a Distinguished Professor of the National Taiwan University, since 2006, and held the position of the Z. Zhuo-Zhang Chair Professor, since 2009. He served as the Director of the Manufacturing Automation Technology Research Center, the Associate Dean with the College of Engineering, the Director of the Graduate Institute of Industrial Engineering, and the Chairperson of Mechanical Engineering with National Taiwan University. His current research interests include machine intelligence, humanoid robots, intelligent robotic systems, prosthetic hands, manufacturing automation, nanomanipulation, and nonlinear systems. He has authored more than 380 articles on these topics that have been published in refereed technical journals and conference proceedings.

Prof. Huang was a member of the Publication Management Committee of IEEE/ASME journals and an Editorial Board Member of the *International Journal of Advanced Robotics*, from 2004 to 2008, and the *International Journal of Service Robots*. He was also a Fellow of the Chinese Institute of Automation Engineers, in 2010, the Chinese Society of Mechanical Engineers, in 2011, and the Robotics Society of Taiwan, in 2018. He was a recipient of the Ford University Research Award, from 1996 to 1998, the Distinguished Research Award thrice, from 1996 to 2002, the Research Fellow Award twice, from 2002 to 2008, the Distinguished Research Fellow Award from the National Science Council, Taiwan, in 2009, the Distinguished Education Award on Radio Frequency Identification from EPCglobal, in 2010, the TECO Outstanding Science and Technology Research Achievement Award, in 2012, the HIWIN Technologies Corporation Distinguished Industry Creation Award, in 2014, and the Automation Engineering Medal, Chinese Institute of Automation Engineers in, 2018. He was the Editor-in-Chief of the *Journal of the Chinese Fuzzy Systems Association*, from 1997 to 1999, and the *International Journal of Fuzzy System*, from 1999 to 2002, an Associate Editor of the IEEE Transactions on Automation Science and Engineering, from 2003 to 2005, the *International Journal of Advanced Robotics Systems*, and the IEEE/ASME Transactions on Mechatronics. He is currently on the editorial board of the *International Journal of Electronic Business Management*, the *International Journal of Fuzzy Systems*, an Editor of the *International Journal of Automation and Smart Technology*, and iRobotics. He was named in Who's Who in the World 2001 and 2002 and in Who's Who in the China, in 2002. He has served as the committee chair/co-chair and a member of several national, international and the IEEE conferences, and the *Journal of Healthcare Engineering* as a Guest Editor.