Applied Engineering Letters Vol.8, No.2, 52-59 (2023) e-ISSN: 2466-4847

FORWARD KINEMATICS ALGORITHM IN DUAL QUATERNION SPACE BASED ON DENAVIT-HARTENBERG CONVENTION

UDC:007.52:531.1
Original scientific paper https://doi.org/10.18485/aeletters.2023.8.2.2

Nikola LJ. Zivkovic[1*], Jelena Z. Vidakovic[1], Mihailo P. Lazarevic[2]

[1]Lola Institute Ltd., Kneza Viseslava 70a, 11030 Belgrade, Serbia
[2]Faculty of Mechanical Engineering, Kraljice Marije 16, 11060 Belgrade, Serbia

Abstract:
Forward kinematics is fundamental to robot design, control, and simulation. Different forward kinematics algorithms have been developed to deal with the complex geometry of a robot. This paper presents a robot forward kinematics algorithm in dual quaternion space. The presented method uses Denavit-Hartenberg (DH) convention for uniform definition of successive rotational and translational transformations in joints along the robot's kinematic chain. This research aims to utilize the advantages of dual quaternions and DH convection for forward kinematics computation and make the algorithm, which is compact, intuitive, numerically robust, and computationally efficient as it uses the minimal number of parameters required for the computation, suitable for implementation in ROS and similar software. The algorithm is verified on the 6DoF industrial robot RL15, with the symbolic equations and numerical simulation presented.

ARTICLE HISTORY
Received: 31 March 2023
Revised: 26 May 2023
Accepted: 5 June 2023
Published: 30 June 2023

KEYWORDS
Forward kinematics, kinematic modelling, DH convention, coordinate frame transformations, quaternions, dual quaternions, robotics

1. INTRODUCTION

Forward kinematics represents the foundation of robot design, control, and simulation. The solution to the robot forward kinematics problem is to determine the position and orientation of the robot end-effector for given joint positions and geometric parameters [1]. Solving the forward kinematics problem requires the development of a kinematic (geometric) model for a given robot. The kinematic model in robotics refers to the relationship between joint space and operational space coordinates [2]. This relationship is most often defined by the suitable placement of coordinate frames rigidly attached to robot links and by the determination of spatial transformations between successive coordinate frames.

The placement of coordinate frames is usually based on the adopted convention, which is advantageous for consistency and computational efficiency [1]. Some of the most used methods for kinematic modeling in robotics are the Denavit-Hartenberg convention [3], the Product of Exponentials (PoE) approach [4], and the Rodriguez approach [5-6]. DH convention minimizes the required parameters to locate coordinate frames attached to robot links relative to one another to four parameters per each coordinate frame [3]. Following the DH convention, a general transformation from one coordinate frame to another can be conveniently described using 4×4 matrices known as homogeneous transformation matrices (HTM). Rodriguez's approach [5-6] employs rotation matrices to describe transformation between coordinate frames and is also often used for modeling biomechanical systems [7]. The Product of Exponentials (PoE) approach uses matrix exponentials to represent transformations between coordinate frames [4]. DH convention is by far the most used method for kinematic modeling in robotics due to its minimal number of parameters required to represent robot configuration [8]. Still, HTM representation is highly redundant since in the transformation matrix, 16

*CONTACT: Nikola Zivkovic, e-mail: nikola.zivkovic@li.rs

numbers are used to represent six degrees of freedom of the rigid body (9 numbers are used for representing the orientation, 3 for the position, and four are trivial) [9-10]. This redundancy can introduce numerical problems in the calculation.

Successive rigid body transformations in robot kinematic chains can also be defined using dual quaternions [11-14]. Dual quaternions are the most compact mathematical construct for defining the screw motion [10,15]. Compared to HTM, they represent a numerically robust method for robot links' pose definition. Dual quaternion consists of eight elements, of which one is trivial. They can be defined as dual numbers with unit quaternions as components that enable them to represent rotational and translational rigid body transformation [16].

In this paper, we present a dual quaternion-based algorithm to solve the robot forward kinematics problem, which uses the DH convention for uniform definition of successive rotational and translational transformations in joints along the open kinematic chain. The motivation for the presented research is to employ the compactness and numerical robustness of dual quaternions with the practicality and intuitiveness of DH convection for forward kinematics computation. Ulterior motivation is to present the algorithm suitable for implementation in Robot Operating System (ROS), i.e., customized to the robot link's pose/movement representation in ROS [17]. The ROS is considered the leading free and open-source software framework for robotic applications.

DH convention assignment of the coordinate systems makes using the dual unit quaternions as rigid transformations straightforward. The algorithm presented in [18] also uses a combination of DH convention and dual unit quaternions for the solution of forward and inverse kinematic problems, as well for differential kinematics with the adopted convention that for each frame, rigid translational transformation is always carried out first, followed by rigid rotational transformation In this paper, we adopted the order of rigid transformations where transformations tied to the z(i-1) axis are carried out first, followed by transformations associated with the xi axis, which is based on the DH convention presented in [19].

The presented forward kinematics algorithm is implemented in the programming language C++ and used with Robot Operating System (ROS) and RViz [20] for visualization of robot motion [21]. The solution is demonstrated and verified with the 6 Degree of Freedom (DoF) industrial robot arm.

This paper is organized as follows. Section 2 presents significant arithmetic operations of quaternions, dual numbers, and dual quaternions. Section 3 summarizes a dual unit quaternion-based algorithm for robot forward kinematics computation with the presented procedure of the coordinate assignment according to the DH convention. In Section 4, the previously presented algorithm is applied to six degrees of freedom industrial robot arm RL15 and verified with a previously developed forward kinematics solution based on homogeneous matrices. Finally, in Conclusion, the advantages and disadvantages of the presented forward kinematics algorithm are summarized, and numerical and symbolic verification results are given.

2. PRELIMINARIES

2.1 Quaternions

Quaternions are four-dimensional vectors with distinct algebraic operations, introduced by Sir William Rowan Hamilton [22-23]. They have uses in the fields such as quantum mechanics, computer graphics, electrodynamics, robotics, etc. [24-25]. Quaternion is defined as:

$$\mathbf{q} = w + xi + yj + zk, \tag{1}$$

where w, x, y and z are real numbers and i, j and k are imaginary numbers [23].

A quaternion can also be represented in a two-component form consisting of a scalar and a vector part:

$$\mathbf{q} = [w, \mathbf{v}],\ \ \mathbf{v} = [x, y, z], \tag{2}$$

where w is the scalar and $\mathbf{v}$ is the vector part. This representation defines vector $\mathbf{v}$ with x, y, z coordinates along unit vectors $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$, respectively. Basic arithmetic operations with quaternions are given in Table 1. In Table 1, s is an arbitrary scalar, $\mathbf{q}_1$ and $\mathbf{q}_2$ are quaternions with their scalar w_1, w_2 and vector parts $\mathbf{v}_1$, $\mathbf{v}_2$ respectively.

Table 1. Arithmetic operations with quaternions: 1. Scalar-quaternion multiplication; 2. Quaternion addition; 3. Quaternion multiplication ($\odot$); 4. Conjugate quaternion; 5. Norm

Designated number	Arithmetic operation
1.	$s\mathbf{q}_1=[sw_1,\ s\mathbf{v}_1]$
2.	$\mathbf{q}_1+\mathbf{q}_2=[w_1+w_2,\ \mathbf{v}_1+\mathbf{v}_2]$
3.	$\mathbf{q}_1\odot\mathbf{q}_2=[w_1w_2\text{-}\mathbf{v}_1\mathbf{v}_2,\ w_1\mathbf{v}_2+w_2\mathbf{v}_1+\mathbf{v}_1\times\mathbf{v}_2]$
4.	$\bar{\mathbf{q}}_1=[w_1,\text{-}\mathbf{v}_1]$
5.	$\lVert\mathbf{q}_1\rVert=\sqrt{\mathbf{q}_1\bar{\mathbf{q}}_1}$

Quaternions are particularly useful in representing a rigid body orientation/rotational transformation. Singularity-free nature and unambiguousness make quaternions the preferred choice over some of the traditional methods for representing rigid transformation [25].

Unit quaternion, a quaternion whose norm equals one, is used to describe rotation about a unit axis $\mathbf{u} = [\mathbf{u}_x, \mathbf{u}_y, \mathbf{u}_z]$ for an angle ϑ using the following expression:

$$\mathbf{q}_{\mathrm{u}}^{\mathrm{rot}} = \left[\cos\left(\frac{\theta}{2}\right), \sin\left(\frac{\theta}{2}\right)\mathbf{u}\right], \tag{3}$$

Besides being used primarily as a tool to represent rotations, quaternions can also be used to represent rigid body translation. Translational rigid transformation, defined as a translation for scalar value m along a unit axis $\mathbf{t} = [t_x, t_y, t_z]$, can be written in the form of a quaternion:

$$\mathbf{q}_{\mathrm{t}}^{\mathrm{trans}} = [0, m\mathbf{t}] = [0, mt_x, mt_y, mt_z]. \tag{4}$$

2.2 Dual numbers

Dual numbers were introduced by an English polymath William Kingdon Clifford in the nineteenth century [26]. A dual number is considered a generalized type of complex number where the imaginary number i is replaced with the operator ε. Dual numbers are defined as:

$$a = p + d\varepsilon, \tag{5}$$

where $\varepsilon \neq 0$ and $\varepsilon^2 = 0$. Real numbers p and d in Eq. (5) are considered primary and dual parts of the dual number a, respectively. Dual numbers are essentially an ordered pair of primary and dual parts (p,d) with distinct arithmetic operations given in Table 2 [26-27].

Table 2. Operations with dual numbers: 1. Addition; 2. Multiplication, where a_1 and a_2 are dual numbers, (p_1, p_2) and (d_1, d_2) are their primary and dual parts, respectively.

Designated number	Arithmetic operation
1.	$a_1+a_2=(p_1+p_2)+\varepsilon(d_1+d_2)$
2.	$a_1a_2=p_1p_2+\varepsilon(d_1p_2+p_1d_2)$

2.3 Dual quaternions

Following the previous definitions of quaternions and dual numbers, dual quaternions can be defined as dual numbers with quaternions as components, i.e., the dual quaternion consists of the primary and the dual part in the form of a unit quaternions [16,25] written as:

$$\mathbf{h} = \mathbf{q}^{\mathrm{p}} + \varepsilon\mathbf{q}^{\mathrm{d}}, \tag{6}$$

where $\mathbf{h}$ is a dual quaternion, $\mathbf{q}^{\mathrm{p}}$, $\mathbf{q}^{\mathrm{d}}$ are quaternions with superscripts denoting primary and dual parts of the dual quaternion, respectively, and ε is the dual operator. The rigid transformation described by the relative position and orientation of two coordinate frames can be presented with a unit dual quaternion [11]. Unit dual quaternion has unit quaternions as components and represents a rigid transformation in the following manner:

$$\mathbf{h} = \mathbf{q}_{\mathrm{u}}^{\mathrm{rot}} + 0.5\varepsilon(\mathbf{q}_{\mathrm{u}}^{\mathrm{rot}} \odot \mathbf{q}_{\mathrm{t}}^{\mathrm{trans}}), \tag{7}$$

where quaternion $\mathbf{q}_{\mathrm{u}}^{\mathrm{rot}}$ represents rigid rotational transformation calculated using equation (3) and $\mathbf{q}_{\mathrm{t}}^{\mathrm{trans}}$ is a vector representing translation written in a quaternion form using equation (4). Based on equations (6) and (7) primary and dual parts of the unit dual quaternion are:

$$\mathbf{h}^{\mathrm{p}} = \mathbf{q}_{\mathrm{u}}^{\mathrm{rot}}, \tag{8}$$

$$\mathbf{h}^{\mathrm{d}} = 0.5(\mathbf{q}_{\mathrm{u}}^{\mathrm{rot}} \odot \mathbf{q}_{\mathrm{t}}^{\mathrm{trans}}). \tag{9}$$

Let two dual quaternions be defined as:

$$\mathbf{h}_1 = \mathbf{h}_1^{\mathrm{p}} + \varepsilon\mathbf{h}_1^{\mathrm{d}}, \tag{10}$$

$$\mathbf{h}_2 = \mathbf{h}_2^{\mathrm{p}} + \varepsilon\mathbf{h}_2^{\mathrm{d}}. \tag{11}$$

The formula for dual quaternion multiplication is:

$$\mathbf{h}_1 \otimes \mathbf{h}_2 = \mathbf{h}_1^{\mathrm{p}} \odot \mathbf{h}_2^{\mathrm{p}} + \varepsilon\left(\mathbf{h}_1^{\mathrm{p}} \odot \mathbf{h}_2^{\mathrm{d}} + \mathbf{h}_1^{\mathrm{d}} \odot \mathbf{h}_2^{\mathrm{p}}\right). \tag{12}$$

Herein the dual quaternion multiplication is denoted with the symbol ($\otimes$).

The operation of dual quaternion multiplication can be used to represent a composition of rigid transformations [11] in the same way as homogeneous transformation matrices multiplication. Dual quaternion multiplication forms a new unit dual quaternion that represents the combined transformation of the original two.

3. MATERIALS AND METHODS

3.1 Algorithm explanation

As was already stated in Section 2, the unit dual quaternion is capable of representing general rigid transformations. With a single unit dual quaternion, one can describe translation followed by a rotation or vice versa. This property enables employment of the DH convention. Using previously derived theory, we can represent rigid transformation from (i-1)-st to i-th coordinate system as a multiplication of two dual quaternions:

$$\mathbf{h}_{i-1}^{i} = \mathbf{h}_{z_{i-1}} \otimes \mathbf{h}_{x_i}, \tag{13}$$

Unit dual quaternions $\mathbf{h}_{z_{i-1}}$ and $\mathbf{h}_{x_i}$ represent rigid transformations tied to the respective axes z_{i-1} and x_i , according to the DH convention. Unit dual quaternion $\mathbf{h}_{z_{i-1}}$ is defined as a rotation about z_{i-1} axis for an angle ϑ_i followed by a translation along the same z_{i-1} axis by a value d_i. Analogously, we have a similar situation with unit dual quaternion $\mathbf{h}_{x_i}$ defined as a translation along a x_i axis followed by a rotation about the same x_i axis for a value of a_i and an angle α_i, respectively. Parameters ϑ_i, d_i, a_i and α_i are DH parameters of the i-th joint. Unit dual quaternions $\mathbf{h}_{z_{i-1}}$ and $\mathbf{h}_{x_i}$ can be written as:

$$\mathbf{h}_{z_{i-1}} = \mathbf{h}_{z_{i-1}}^{\mathrm{rot}} + 0.5\varepsilon(\mathbf{h}_{z_{i-1}}^{\mathrm{rot}} \odot \mathbf{h}_{z_{i-1}}^{\mathrm{trans}}), \tag{14}$$

$$\mathbf{h}_{x_i} = \mathbf{h}_{x_i}^{\mathrm{rot}} + 0.5\varepsilon(\mathbf{h}_{x_i}^{\mathrm{trans}} \odot \mathbf{h}_{x_i}^{\mathrm{rot}}), \tag{15}$$

After all unit dual quaternions that represent rigid transformations between each two successive coordinate frames denoted with indices i and i+1 are calculated based on DH parameters, the next step is to multiply them to obtain dual quaternion that represents rigid transformation between reference and end-effector coordinate frame:

$$\mathbf{h}_{\mathrm{end}} = \mathbf{h}_0^1 \otimes \mathbf{h}_1^2 \otimes \ldots \otimes \mathbf{h}_{n-1}^{n}, \tag{16}$$

The end effector position in quaternion form is extracted from the dual quaternion $\mathbf{h}_{\mathrm{end}}$ via the next formula:

$$\mathbf{p} = 2\left(\mathbf{h}_{\mathrm{end}}^{\mathrm{d}} \odot \bar{\mathbf{h}}_{\mathrm{end}}^{\mathrm{p}}\right), \tag{17}$$

The position of the end effector is the vector part of the quaternion $\mathbf{p}$. Orientation of the end effector is the primary part of dual quaternion $\mathbf{h}_{\mathrm{end}}$:

$$\mathbf{o} = \mathbf{h}_{\mathrm{end}}^{\mathrm{p}}, \tag{18}$$

3.2 Algorithm steps

Step 1. For i = 1,2,…,n, locate the reference frames attached to each link according to the DH convention presented in [19]. The base is denoted with subscript 0, and n is the number of DoF.

Step 2. For i = 1,2,…,n joints, assign DH parameters for the robot manipulator as presented in [19].

Step 3. For i = 1,2,…,n joints, define unit quaternions $\mathbf{q}_{z_{i-1}}^{\mathrm{rot}}$, $\mathbf{q}_{z_{i-1}}^{\mathrm{trans}}$, $\mathbf{q}_{x_i}^{\mathrm{rot}}$ and $\mathbf{q}_{x_i}^{\mathrm{trans}}$ according to their respective formulas and DH parameters assigned in Step 2.

Step 4. For i = 1,2,…,n joints calculate dual quaternions $\mathbf{h}_{z_{i-1}}$ and $\mathbf{h}_{x_i}$ according to equations (14) and (15), respectively.

Step 5. For i = 1,2,…,n perform successive dual quaternion multiplications according to equation (16) to calculate dual quaternion describing rigid transformation from the base frame to the coordinate frame attached to the end-effector.

Step 6. Calculate the position and orientation of the end-effector from $\mathbf{h}_{\mathrm{end}}$ using equations (17) and (18).

A summarized overview of the previous steps is given in Fig. 1.

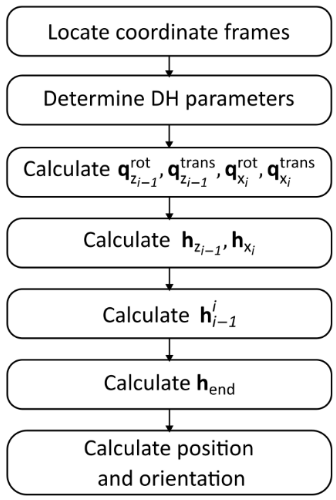

Fig. 1. Steps of the kinematic modelling and forward kinematics algorithm

4. RESULTS AND DISCUSSION

4.1 Symbolic example on 6DoF robot RL15

The presented forward kinematics algorithm is verified using the industrial six-degree-of-freedom robot arm RL15 with revolute joints. The kinematic model and forward kinematics algorithm with homogeneous matrices for robot RL15 used as the reference for our verification can be found in [28]. Following steps 1. and 2. of the algorithm in subsection 3.2. of this paper, assigned coordinate frames are shown in Fig. 2., and DH parameters for RL15 are populated in Table 3.

Table 3. DH parameters for robot RL15

joint	ϑ_i[°]	d_i[mm]	a_i[mm]	α_i[°]
1	q_1	0	200	90
2	q_2+90	0	600	0
3	q_3	0	115	90
4	q_4	825	0	-90
5	q_5	0	0	90
6	q_6	0	0	0

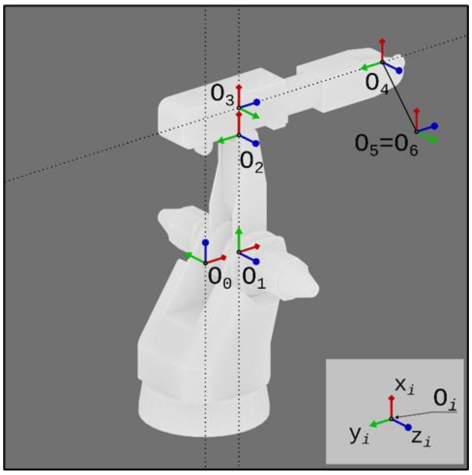

Fig. 2. Coordinate frames assignment according to DH convention for RL15

Now, with all coordinate frames assigned and all of the DH parameters determined, the next step is setting up the dual quaternions for rigid transformations. First, quaternions representing rotation and translation vectors are determined based on DH parameters as shown in Table 4 and Table 5. Parameters d_i and a_i are substituted as symbolic variables for clarity.

Table 4. Rotation and translation quaternions for z_{i-1} axis based on DH parameter for RL15

i	$\mathbf{q}^{rot}_{z_{i-1}}$	$\mathbf{q}^{trans}_{z_{i-1}}$
1	$\left[\cos\left(\frac{q_1}{2}\right),\ 0,\ 0,\ \sin\left(\frac{q_1}{2}\right)\right]$	[0, 0, 0, 0]
2	$\left[\cos\left(\frac{q_2}{2}+90\right),\ 0,\ 0,\ \sin\left(\frac{q_2}{2}+90\right)\right]$	[0, 0, 0, 0]
3	$\left[\cos\left(\frac{q_3}{2}\right),\ 0,\ 0,\ \sin\left(\frac{q_3}{2}\right)\right]$	[0, 0, 0, 0]
4	$\left[\cos\left(\frac{q_4}{2}\right),\ 0,\ 0,\ -\sin\left(\frac{q_4}{2}\right)\right]$	$[0, 0, 0, d_4]$
5	$\left[\cos\left(\frac{q_5}{2}\right),\ 0,\ 0,\ \sin\left(\frac{q_5}{2}\right)\right]$	[0, 0, 0, 0]
6	$\left[\cos\left(\frac{q_6}{2}\right),\ 0,\ 0,\ -\sin\left(\frac{q_6}{2}\right)\right]$	[0, 0, 0, 0]

Table 5. Rotation and translation quaternions for x_i axis based on DH parameter for RL15

i	$\mathbf{q}^{rot}_{x_i}$	$\mathbf{q}^{trans}_{x_i}$
1	$\left[\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0, 0\right]$	$[0, a_1, 0, 0]$
2	[1, 0, 0, 0]	$[0, a_2, 0, 0]$
3	$\left[\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0, 0\right]$	$[0, a_3, 0, 0]$
4	$\left[\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}, 0, 0\right]$	[0, 0, 0, 0]
5	$\left[\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0, 0\right]$	[0, 0, 0, 0]
6	[1, 0, 0, 0]	[0, 0, 0, 0]

Using steps 3. through 6. from the algorithm in subsection 3.2. we can calculate the pose of the end effector. The following substitutions are adopted for clarity:

$$\cos\beta_i = \mathrm{c}_{\beta_i}, \sin\beta_i = \mathrm{s}_{\beta_i}, \tag{19}$$

$$\cos(\beta_i \pm \beta_j) = \mathrm{c}_{(\beta_i \pm \beta_j)}, \sin(\beta_i \pm \beta_j) = \mathrm{s}_{(\beta_i \pm \beta_j)}. \tag{20}$$

Position of the end effector frame with projections onto *x*, *y* and *z* coordinate axes:

$$\mathbf{p}_x = \mathrm{c}_{q_1}\left(a_1 + d_4\mathrm{c}_{q_2+q_3} - a_3\mathrm{s}_{q_2+q_3} - a_2\mathrm{s}_{q_2}\right), \tag{21}$$

$$\mathbf{p}_y = \mathrm{s}_{q_1}\left(a_1 + d_4\mathrm{c}_{q_2+q_3} - a_3\mathrm{s}_{q_2+q_3} - a_2\mathrm{s}_{q_2}\right), \tag{22}$$

$$\mathbf{p}_z = a_3\mathrm{c}_{q_2+q_3} + d_4\mathrm{s}_{q_2+q_3} + a_2\mathrm{c}_{q_2}. \tag{23}$$

Orientation of the end effector frame in quaternion form:

$$\mathbf{o}_w = \frac{\sqrt{2}}{8}\left(\mathrm{U}^c_w + \mathrm{U}^s_w\right), \tag{24}$$

$$\mathbf{o}_x = \frac{\sqrt{2}}{8}\left(\mathrm{U}^c_x + \mathrm{U}^s_x\right), \tag{25}$$

$$\mathbf{o}_y = \frac{\sqrt{2}}{8}\left(\mathrm{U}^c_y + \mathrm{U}^s_y\right), \tag{26}$$

$$\mathbf{o}_z = \frac{\sqrt{2}}{8}\left(\mathrm{U}^c_z + \mathrm{U}^s_z\right), \tag{27}$$

where further substitutions are adopted and displayed in Table 6.

Table 6. Substitutions for the end-effector orientation equations (24), (25), (26) and (27)

Substitution	Substituted expression
U^c_w	$\mathrm{c}_{(V_1-V_2-V_3-V_4+V_5+V_6-V_7+V_8)}$
U^s_w	$\mathrm{s}_{(V_1-V_2+V_3+V_4-V_5-V_6+V_7-V_8)}$
U^c_x	$\mathrm{c}_{(V_1-V_9+V_{10}-V_6+V_{11}+V_{12}+V_{13}+V_{14})}$
U^s_x	$\mathrm{s}_{(V_1-V_9-V_{10}+V_6-V_{11}-V_{12}-V_{13}+V_{14})}$
U^c_y	$\mathrm{c}_{(V_9-V_{15}-V_{14}-V_{10}-V_6+V_{11}+V_{12}+V_{13})}$
U^s_y	$\mathrm{s}_{(V_{15}-V_9+V_{14}-V_{10}-V_6+V_{11}+V_{12}+V_{13})}$
U^c_z	$\mathrm{c}_{(V_5-V_1+V_2+V_3+V_4+V_6-V_7+V_8)}$
U^s_z	$\mathrm{s}_{(V_5+V_1-V_2+V_3+V_4+V_6-V_7+V_8)}$
V_1	E_1+E_4
V_2	E_1+E_5
V_3	$-E_1+E_6$
V_4	$-E_1+E_3$
V_5	E_2-E_3
V_6	E_2-E_4
V_7	E_2+E_5
V_8	E_2+E_4
V_9	E_1-E_5
V_{10}	$-E_1+E_4$
V_{11}	E_2-E_5
V_{12}	E_2+E_6
E_1	$\frac{1}{2}(q_1-q_2-q_3)$

Table 6. Substitutions for the end-effector orientation equations (24), (25), (26) and (27) - Continuation of the table from the previous page

Substitution	Substituted expression
E_2	$\frac{1}{2}(q_1+q_2+q_3)$
E_3	$\frac{1}{2}(q_4+q_5+q_6)$
E_4	$\frac{1}{2}(q_4+q_5-q_6)$
E_5	$\frac{1}{2}(q_4-q_5-q_6)$
E_6	$\frac{1}{2}(q_4-q_5+q_6)$

Comparing previously obtained results with the dual quaternion-based algorithm and results obtained with homogeneous transformation matrices [28], it can be verified that the dual quaternion algorithm yields the same results.

4.2 Simulation example using implementation in ROS and RViz

A simple library is developed in C++ to implement dual quaternion algebra for coordinate transformations with necessary data structures and operations [21]. Quaternion is defined as a composition of scalar and vector objects according to representation in (6). A dual quaternion is defined as a composition of two quaternions. The multiplication and addition operators are overloaded based on the operands that are using them. The dual quaternion library enables the dual quaternion computation of the forward kinematics in ROS. Fig. 3. shows the model of the RL15 in RViz.

The end-effector pose, denoted as ‘eeCarry_seg6’ in Fig. 4., can be examined in the left tab inside RViz under the options ‘Position’ and ‘Orientation’.

Fig. 3. RL15 model in RViz

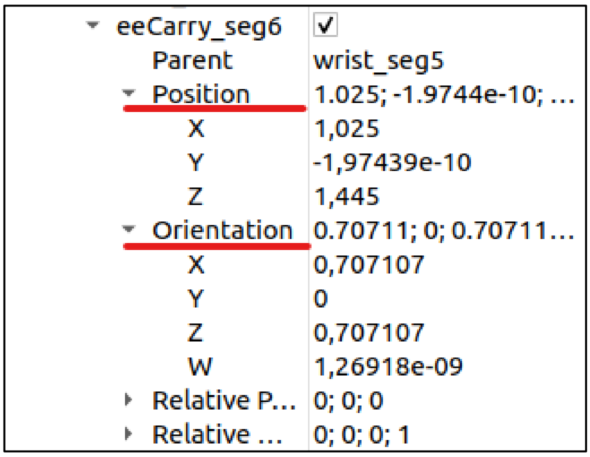

Fig. 4. Numeric values of a pose of the end-effector in RViz underlined with red line

The primary application of this library is to enable forward kinematics computation in ROS projects. It is used in conjunction with RViz to visualize robots for simulation and remote monitoring purposes easily.

5. CONCLUSION

In this paper, we presented and verified the forward kinematics algorithm based on dual quaternions using the DH convention and implemented it as a C++ library for applications in ROS. Dual quaternions are a viable option for representing rigid transformations in the kinematic analysis of robotic manipulators. Dual quaternions allow singularity-free equations, which is a significant advantage compared to Euler angles, and they can represent rotational and translational rigid transformations between frames combined into one object. The DH convention provides a consistent and systematic way to describe the kinematic structure of a manipulator, allowing for the use of simple algebraic equations to compute the forward and inverse kinematics and being able to handle manipulators with any number of degrees of freedom and any joint type. Using a combination of DH convention and dual quaternions, one can benefit from minimal parameter representation, systematic kinematic modeling, numerical robustness, and compact representation of screw motion.

ACKNOWLEDGEMENT

This research has been supported by the research grants of the Serbian Ministry of Science, Technological Development and Innovations, grant No. 451-03-68/2023-14/200066 and 451-03-47/2023-01/ 200105 from 03.02.2023.

REFERENCES

[1] B. Siciliano, O. Khatib, T. Kroger, Handbook of robotics. *Springer*, Berlin, 2008.

[2] J.J. Craig, Introduction to robotics: mechanics and control, 3rd ed. *Pearson Prentice Hall*, Upper Saddle River, USA, 2005.

[3] P. Corke, Robot Arm Kinematics. In: Robotics, Vision and Control. Springer Tracts in Advanced Robotics. *Springer*, 118, 2017: 193-228. https://doi.org/10.1007/978-3-319-54413-7_7

[4] K. M. Lynch, F.C. Park, Modern Robotics: Mechanics, Planning and Control. *Cambridge University Press*, 2017.

[5] P. Mandić, M. Lazarević, Z. Stokić, T. Šekara, Dynamic modelling and control design of seven degrees of freedom robotic arm. *6th International Congress of Serbian Society of Mechanics*, 19-21 June 2017, Tara, Serbia, pp.1-8.

[6] V. Covic, M. Lazarevic, Mehanika robota, second ed. *Faculty of Mechanical Engineering*, Belgrade, 2021. (in Serbian)

[7] M. Lazarevic, Mechanics of human locomotor system. *FME Transactions*, 34(2), 2006: 105-114.

[8] C. Faria, J.L. Vilaca, S. Monteiro, W. Erlhagen, E. Bicho, Automatic Denavit-Hartenberg parameter identification for serial manipulators. *45th Annual Conference of the IEEE Industrial Electronics Society (IECON)*, 1 October 2019, Lisbon, Portugal, pp.610-617. https://doi.org/10.1109/IECON.2019.8927455

[9] N.A. Aspragathos, J.K. Dimitros, A Comparative Study of Three Methods for Robot Kinematics. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 28(2), 1998: 135-145. https://doi.org/10.1109/3477.662755

[10] J. Funda, R.H. Taylor, R.P. Paul, On homogeneous transformations, quaternions and computational efficiency. *IEEE Transactions on Robotics and Automation*, 6(3), 1990: 382-388. https://doi.org/10.1109/70.56658

[11] J.Z. Vidakovic, M.P. Lazarevic, V.M. Kvrgic, Z.Z. Dancuo, G.Z. Ferenc, Advanced quaternion forward kinematics algorithm including overview of different methods for robot kinematics. *FME Transactions*, 42(3), 2014: 189-199. https://doi.org/10.5937/fmet1403189V

[12] B.V. Adorno, Two-arm Manipulation: From Manipulators to Enhanced Human-Robot Collaboration (Ph.D. Thesis). *Université Montpellier II - Sciences et Techniques du Languedoc,* Montpellier, 2011.

[13] N.T. Dantam, Robust and efficient forward, differential and inverse kinematics using dual quaternions. *International Journal of Robotics Research*, 40(10-11), 2021: 1087-1105. https://doi.org/10.1177/0278364920931948

[14] S. Kucuk, Z. Bingul, Robot Kinematics: Forward and Inverse Kinematics. *Industrial Robotics: Theory, Modelling and Control*, 2006: 117-148. https://doi.org/10.5772/5015

[15] N.T. Dantam, Practical Exponential Coordinates Using Implicit Dual Quaternions. In: Morales, M., Tapia, L., Sánchez-Ante, G., Hutchinson, S. (eds) Algorithmic Foundations of Robotics XIII. WAFR 2018. *Springer Proceedings in Advanced Robotics*, 14, 2020. https://doi.org/10.1007/978-3-030-44051-0_37

[16] E. Özgür, Y. Mezouar, Kinematic modeling and control of a robot arm using unit dual quaternions. *Robotics and Autonomous Systems*, 77, 2016: 66-73. https://doi.org/10.1016/j.robot.2015.12.005

[17] Stanford Artificial Intelligence Laboratory, Robotic Operating System, 2018. https://www.ros.org (Accessed 27 December 2022).

[18] A. Valverde, P. Tsiotras, Spacecraft Robot Kinematics Using Dual Quaternions. *Robotics*, 7(4), 2018: 64. https://doi.org/10.3390/robotics7040064

[19] M.W. Spong, S. Hutchinson, M. Vidyasagar, Forward Kinematics, in Robot Modelling and Control, 2nd ed. *Wiley*, 2020, p.79.

[20] Github, RViz, 2022. https://github.com/ros-visualization/rviz.git (Accessed 28 March 2023)

[21] N. Zivkovic, J. Vidakovic, S. Mitrovic, M. Lazarevic, Implementation of Dual Quaternion-based Robot Forward Kinematics Algorithm in ROS. *11th Mediterranean Conference on Embedded Computing (MECO)*, 21 June 2022, Budva, Montenegro, pp.1-4. https://doi.org/10.1109/MECO55406.2022.9797160

[22] W. R. Hamilton, On Quaternions, or on a new system of imaginaries in algebra.

https://www.maths.tcd.ie/pub/HistMath/People/Hamilton/OnQuat/ (Accessed 28 March 2023).

[23] W. R. Hamilton, Additional Applications of the theory of algebraic quaternions. https://www.maths.tcd.ie/pub/HistMath/People/Hamilton/QDynamics/ (Accessed 28 March 2023).

[24] G. Ron, Understanding quaternions. *Graphical models*, 73(2), 2011: 21-49. https://doi.org/10.1016/j.gmod.2010.10.004

[25] B. Kenwright, A Beginners Guide to Dual-Quaternions. *20th International Conference on Computer Graphics, Visualization and Computer Vision (WSCG)*, 26-28 June 2012, Plzen, Czech Republic, pp.1-10.

[26] J. Rooney, Wiliam Kingdon Clifford (1845-1879). In: M. Ceccarelli (Ed.), Distinguished Figures in Mechanism and Machine Science. *History of Mechanism and Machine Science*, 1, 2007. https://doi.org/10.1007/978-1-4020-6366-4_4

[27] V. Brodsky, M. Shoham, Dual numbers representation of rigid body dynamics. *Mechanism and Machine Theory*, 34(5), 1999: 693-718. https://doi.org/10.1016/S0094-114X(98)00049-4

[28] V. Kvrgic, J. Vidakovic, Efficient method for robot forward dynamics computation. *Mechanism and Machine Theory*, 145, 2020: 103680. https://doi.org/10.1016/j.mechmachtheory.2019.103680