Solution of Inverse Kinematics for 6R Robot Manipulators With Offset Wrist Based on Geometric Algebra

Article *in* Journal of Mechanisms and Robotics · August 2013

DOI: 10.1115/1.4024239 · Source: PubMed

Zhongtao Fu
Wuhan Institute of Technology

Wenyu Yang
Huazhong University of Science and Technology

Solution of Inverse Kinematics for 6R Robot Manipulators With Offset Wrist Based on Geometric Algebra

Zhongtao Fu
e-mail: hustfzt@gmail.com

Wenyu Yang[1]
e-mail: mewyang@mail.hust.edu.cn

Zhen Yang
e-mail: yangzhen0607@126.com

State Key Laboratory of Digital Manufacturing Equipment and Technology,
Mechanical School of Science and Engineering,
Huazhong University of Science and Technology,
Wuhan 430074, China

In this paper, we present an efficient method based on geometric algebra for computing the solutions to the inverse kinematics problem (IKP) of the 6R robot manipulators with offset wrist. Due to the fact that there exist some difficulties to solve the inverse kinematics problem when the kinematics equations are complex, highly nonlinear, coupled and multiple solutions in terms of these robot manipulators stated mathematically, we apply the theory of Geometric Algebra to the kinematic modeling of 6R robot manipulators simply and generate closed-form kinematics equations, reformulate the problem as a generalized eigenvalue problem with symbolic elimination technique, and then yield 16 solutions. Finally, a spray painting robot, which conforms to the type of robot manipulators, is used as an example of implementation for the effectiveness and real-time of this method. The experimental results show that this method has a large advantage over the classical methods on geometric intuition, computation and real-time, and can be directly extended to all serial robot manipulators and completely automatized, which provides a new tool on the analysis and application of general robot manipulators.
[DOI: 10.1115/1.4024239]

Keywords: inverse kinematics, geometric algebra, symbolic elimination technique, 6R robot manipulator, offset wrist

1 Introduction

The IKP for 6R robot manipulator is to determine the joint values given the position and orientation of the end-effector relative to the base and the values of all link parameters, which has been recognized as a more important problem for robot workspace analysis, trajectory planning, motion control and off-line programming [1]. For 6R robot manipulators with offset wrist, it requires dealing with the highly nonlinear mapping from Cartesian space to joint space, computing the solutions of a set of polynomials obtained from the kinematics equations and yielding multiple poses for the manipulators. However, this type of robot manipulators owns a special geometric structure in practical applications, which does not allow having closed-form solutions for IKP. Therefore, efficient numerical methods should be provided to solve IKP for this type of robot manipulators.

In case of any difficulties to solve IKP when the kinematics equations are complex, highly nonlinear, coupled and have multiple solutions [2], many researchers tried various methods to solve IKP effectively based on concepts introduced by mechanicians and mathematicians. Raghavan and Roth [3] used dialytic elimination to reduce the IKP to a 16 deg polynomial in one unknown and found all possible solutions, and Manocha and Canny [4] formulated this problem as an eigenvalue problem and improved the stability and accuracy of the overall algorithm. Aspragathos et al. [5] compared three methods, i.e., homogeneous transform matrix, Lie algebra, and screw theory used in the kinematics analysis of robot manipulators for the computing time and storage requirements, and the latter two methods are more cost effective as the number of robot DOFs increases. Husty et al. [6] made use of classical multidimensional geometry to structure the IKP and to use the geometric information before the elimination process. Then, the 6R chain was broken up in the middle to form two open 3R chains and so they obtained 16 solutions. Qiao et al. [7] transformed the 4×4 homogeneous matrix to the form of double quaternion and leaded to double kinematic equations. Then, 16 solutions were obtained from resultant matrix via linear algebra and Dixion resultant elimination. In addition, detailed descriptions of other methods have been given [8–12]. However, such methods suffer from complex matrix calculations, high computational cost and severe singularity as the number of robot DOFs increases and may not guarantee the requirement for real-time and position accuracy.

In terms of the drawbacks of conventional methods, a few attempts were made to apply Geometric Algebra to solve IKP of robot manipulators. Geometric algebra [13,14], as introduced by William Clifford, puts vector, quaternion, Riemann algebra, Lie algebra, complex, screw, etc., algebra system into a unified framework for many mathematical systems, thus avoids the complex transformation between different algebraic languages on the same problem. Moreover, it has a direct calculation on geometric elements, owns the characteristics of intuitive geometric meaning, simple symbols representation, coordinate independence, multidimensional space calculation, and provides a convenient mathematical notation for representing basic entities and transformation of Euclidean space. These powerful advantages of Geometric Algebra make it very popular for applications in robotics and attract many excellent researchers. Zamora and Bayro-Corrochano [15] first applied Geometric Algebra to visually guided robotics and computed the inverse kinematics of a robot arm to solve a problem of visually guided grasping, which demonstrated the power and simplicity of Geometric Algebra for solving IKP. Hildenbrand [16] adopted conformal geometric algebra to solve the IKP for a simple robot visually and showed that

[1]Corresponding author.
Contributed by the Mechanisms and Robotics Committee of ASME for publication in the Journal of Mechanisms and Robotics. Manuscript received October 25, 2012; final manuscript received March 19, 2013; published online June 24, 2013. Assoc. Editor: J. M. Selig.

Geometric Algebra was a promising mathematical tool for computer science and animations, then later, [17] presented an approach for the IKP of the arm of a virtual human efficiently using reconfigurable hardware for real-time and make sure that this approach can be used very advantageously in the applications of computer animation and graphics. Aristidou and Lasenby[18] employed Geometric Algebra for incorporation of IKP, and described a novel iterative inverse kinematics solver [19], FABRIK, that was implemented using Geometric Algebra, thus solved the IKP of a human hand for pose tracking. This application proved that Geometric Algebra was a useful mathematical tool which can be successfully used for applications in robotics. These related researches, however, are nearly focused on simplified manipulators, anthropomorphical arms and robots, few for complex multijoint manipulators.

In this paper, we apply the powerful advantages of Geometric Algebra and dual angle to the kinematic modeling of 6R robot manipulator and generate closed-form kinematic equations simply, reformulate the problem as a generalized eigenvalue problem with symbolic elimination technique, and then yield 16 solutions. Finally, a spray painting robot with offset wrist is used as an example of implementation for the effectiveness and real-time of this method. Moreover, this method facilitates a novel, elegant, simple representation and computation to the analysis and application of general robot manipulators.

The rest of this paper is structured in the following manner. After introducing the overview of Geometric Algebra abut representing basic elements and transformations of 3D Euclidean space in Sec. 2, we describe in Sec. 3 the mathematical modeling and solutions of the inverse kinematics for 6R robot manipulator based on Geometric Algebra. A numerical example of spray painting robot with offset wrist to verify this method is presented in Sec. 4, and the set of all 16 solutions obtained are listed in Table 2. Finally, conclusion of this study is given in Sec. 5.

2 Introduction to Geometric Algebra

In this section, we will give a brief overview on Geometric Algebra abut representing basic elements and transformation of 3D Euclidean space in a very elegant and geometrically intuitive way. More detailed introduction to Geometric Algebra should refer to Refs. [13,14,20,22–26].

2.1 The Basic Elements of Geometric Algebra. Consider the 3D Euclidean space $\boldsymbol{R}^3$ with orthonormal basis $\{e_1, e_2, e_3\}$, a set of linearly independent combinations of these basis elements using the geometric product is given by

$$\underbrace{1}_{\text{scalar}}, \underbrace{\{e_1, e_2, e_3\}}_{\text{vectors}}, \underbrace{\{e_{23}, e_{31}, e_{12}\}}_{\text{bivectors}}, \underbrace{e_1 e_2 e_3 \equiv I}_{\text{trivector=pseudoscalar}} \tag{1}$$

These $2^3 = 8$ elements are named the basic elements of Geometric Algebra $\boldsymbol{G}_{3,0,0}$, and own the following properties:

$$\text{vectors: } e_1^2 = e_2^2 = e_3^2 = 1 \tag{2}$$

$$\text{bivetors: } e_{23}^2 = e_{31}^2 = e_{12}^2 = -1 \tag{3}$$

In the conformal model, we extend the space $\boldsymbol{G}_{3,0,0}$ by adding two additional orthogonal basis elements $\{e_+, e_-\}$, and form the Conformal space $\boldsymbol{G}_{4,1,0}$

$$e_+^2 = 1 \quad e_-^2 = 1 \quad e_+ \cdot e_- = 0 \tag{4}$$

Another basis $\{e_0, e_\infty\}$ with the following geometric meaning:

$$e_0\text{: representing the 3D origin}$$

$$e_\infty\text{: representing the 3D infinity}$$

can be defined with the relations

$$e_0 = \frac{1}{2}(e_- - e_+) \quad e_\infty = e_- + e_+ \tag{5}$$

These new basic elements are null vectors

$$e_0^2 = e_\infty^2 = 0 \quad e_\infty \cdot e_0 = -1 \tag{6}$$

In addition, a dual unit ε is defined by

$$\varepsilon = e_\infty I^{-1} \quad \text{where} \quad \varepsilon^2 = 0 \tag{7}$$

2.2 Dual Angle. In differential geometry and motion analysis of spatial mechanisms, a dual number [21] can be commonly referred to as *dual angle*

$$\hat{\theta} = \theta + \varepsilon d \tag{8}$$

between two lines S_1 and S_2 in space as shown in Fig. 1, where the real part θ is the projected angle and the dual part d is the length of the common perpendicular between lines S_1 and S_2, respectively. In general, the dual angle between skew lines is a proper dual number, can also be represented by

$$\hat{\theta} = \theta(1 + \varepsilon p) \tag{9}$$

where the ratio $p = d/\theta$ is referred to as the pitch of the dual angle $\hat{\theta}$. If p is zero, it is a pure rotation; if p is infinity, it is a pure translation.

Furthermore, a function of dual number $f(a + \varepsilon b)$ can be expanded into a formal Taylor series with the definition $\varepsilon^2 = 0$

$$f(a + \varepsilon b) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a) b^n \varepsilon^n}{n!} = f(a) + \varepsilon b f'(a) \tag{10}$$

where $f^{(n)}(a)$ denotes the nth-order derivative of f evaluated at the point a. According to Eq. (10), one obtains

$$\begin{aligned} \cos(\hat{\theta}) &= \cos(\theta + \varepsilon d) = \cos(\theta) - \varepsilon d \sin(\theta) \\ \sin(\hat{\theta}) &= \sin(\theta + \varepsilon d) = \sin(\theta) + \varepsilon d \cos(\theta) \end{aligned} \tag{11}$$

2.3 Rigid Transformation. In the Conformal space $\boldsymbol{G}_{4,1,0}$, the Euclidean transformation between rigid bodies can be decomposed into a rotation (*rotor*) followed by a translation (*translator*) or vice versa.

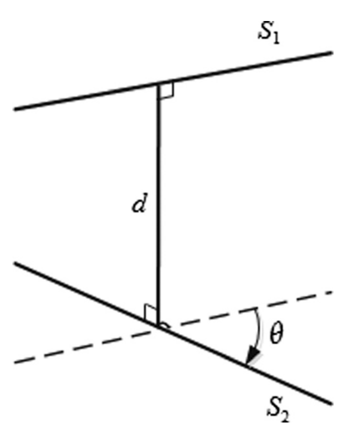

Fig. 1 Dual angle $\hat{\theta} = \theta + \varepsilon d$ between lines S_1 and S_2

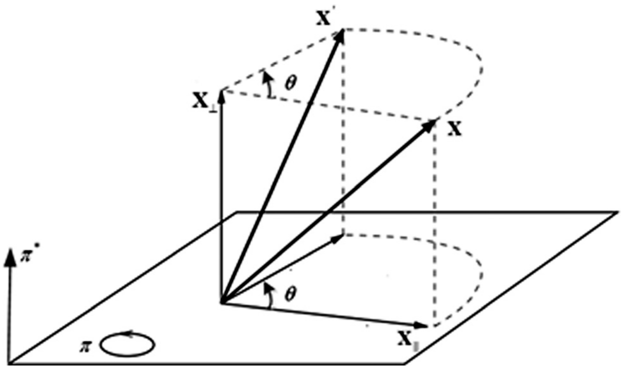

Fig. 2 Rotation of a vector X

A rotor, $\boldsymbol{R}$, is an even-grade element of the algebra which satisfies $\boldsymbol{R}\tilde{\boldsymbol{R}} = 1$, where $\tilde{\boldsymbol{R}}$ stands for the conjugate of $\boldsymbol{R}$. By using the Euler representation of a rotor, we can write it as follows:

$$\boldsymbol{R} = \exp\left(-\frac{\theta}{2}\boldsymbol{\pi}\right) = \cos\left(\frac{\theta}{2}\right) - \boldsymbol{\pi}\sin\left(\frac{\theta}{2}\right) \tag{12}$$

And the rotation of a vector $\boldsymbol{X}$ over the rotor $\boldsymbol{R}$ is given by

$$\begin{aligned} \boldsymbol{X}' = \boldsymbol{R}\boldsymbol{X}\tilde{\boldsymbol{R}} &= \left[\cos\left(\frac{\theta}{2}\right) - \boldsymbol{\pi}\sin\left(\frac{\theta}{2}\right)\right](\boldsymbol{X}_\perp + \boldsymbol{X}_\parallel) \\ &\cdot \left[\cos\left(\frac{\theta}{2}\right) + \boldsymbol{\pi}\sin\left(\frac{\theta}{2}\right)\right] \\ &= \boldsymbol{X}_\perp + \boldsymbol{X}_\parallel \exp(\boldsymbol{\pi}\theta) \end{aligned} \tag{13}$$

where the rotation axis is an orthogonal to the plane $\boldsymbol{\pi} = u_1 e_{23} + u_2 e_{31} + u_3 e_{12}$ that is spanned by the normalized bivector and θ is the rotation angle around this axis. That is illustrated in Fig. 2. Besides, rotor combines in a straightforward manner, i.e., a rotor $\boldsymbol{R}_1$ followed by a rotor $\boldsymbol{R}_2$ is equivalent to a total rotor $\boldsymbol{R}$ where $\boldsymbol{R} = \boldsymbol{R}_2\boldsymbol{R}_1$.

A *translator*, $\boldsymbol{T}$, can be regarded as special rotation acting at infinity by using the null vector e_∞ and defined by

$$\boldsymbol{T} = \exp\left(-\frac{\boldsymbol{d}}{2}e_\infty\right) = 1 - \frac{\boldsymbol{d}}{2}e_\infty \tag{14}$$

where $\boldsymbol{d}$ is a vector: $\boldsymbol{d} = d_1 e_1 + d_2 e_2 + d_3 e_3$.

A *motor*, $\boldsymbol{M}$, can describe rigid transformation, and as a composition of a rotation and a translation, both related to the same rotation axis (Fig. 3). Therefore, the *motor* can be written

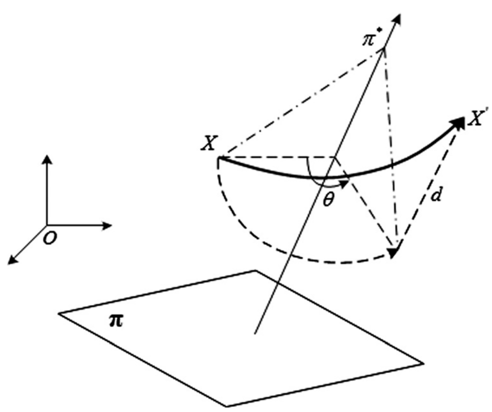

Fig. 3 Transformation of a vector X

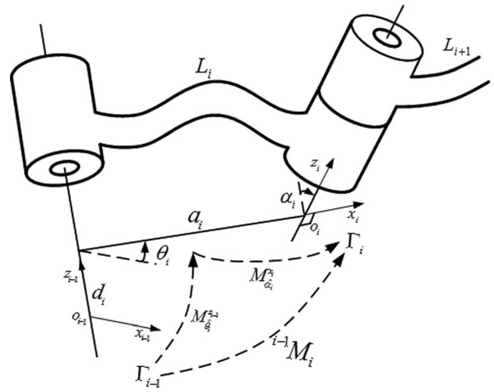

Fig. 4 D_H parameters and frames between relative links and the transformation ${}^{i-1}\boldsymbol{M}_i$ from Γ_{i-1} to Γ_i

$$\boldsymbol{M} = \boldsymbol{R}\boldsymbol{T} = \exp\left(-\frac{\theta}{2}\boldsymbol{\pi}\right)\exp\left(-\frac{\boldsymbol{d}}{2}e_\infty\right) \tag{15}$$

And the transformation of a vector $\boldsymbol{X}$ over the *motor* $\boldsymbol{M}$ (also called screw motion) is given by

$$\boldsymbol{X}' = \boldsymbol{M}\boldsymbol{X}\tilde{\boldsymbol{M}} = \boldsymbol{R}\boldsymbol{T}\boldsymbol{X}\tilde{\boldsymbol{T}}\tilde{\boldsymbol{R}} \tag{16}$$

In addition, by introducing the concept of *dual angle* and combining Eq. (8), we can rewrite the *motor* as follows:

$$\boldsymbol{M} = \exp\left(-\frac{\hat{\theta}}{2}\boldsymbol{\pi}\right) = \exp\left(-\frac{\theta + \varepsilon d}{2}\boldsymbol{\pi}\right) \tag{17}$$

Rigid transformation of an object $\boldsymbol{Q}$, such as points, lines, planes, circles and spheres, can be carried out by multiplying the *motor* $\boldsymbol{M}$ from the left and its reverse $\tilde{\boldsymbol{M}}$ from the right as follows [23,24]:

$$\boldsymbol{Q}' = \boldsymbol{M}\boldsymbol{Q}\tilde{\boldsymbol{M}} \tag{18}$$

3 Inverse Kinematics Analysis of 6R Robot Manipulators

3.1 Representation of Rigid Transformation. A robot manipulator consists of a number of rigid bodies connected in succession by kinematic joints to form an open serial mechanism. In terms of D_H convention [1], every link can be presented with the attached frame, and the four parameters $\theta_i, d_i, \alpha_i, a_i$ are generally named joint angle, link offset, link twist angle and link length, respectively. The transformation ${}^{i-1}\boldsymbol{M}_i$ from Frame Γ_{i-1} to Γ_i consists of a sequence of two *motors*, one variable, i.e., $\boldsymbol{M}_{\hat{\theta}}^{Z_{i-1}}$, and another constant, i.e., $\boldsymbol{M}_{\hat{\alpha}}^{X_{i-1}}$, see Fig. 4. Note that we use dual angles $\hat{\theta}_i = \theta_i + \varepsilon d_i$ and $\hat{\alpha}_i = \alpha_i + \varepsilon a_i$. The angle θ_i is the joint variable for 6R robot manipulator. The transformation reads

$${}^{i-1}\boldsymbol{M}_i = \boldsymbol{M}_{\hat{\theta}_i}^{Z_{i-1}}\boldsymbol{M}_{\hat{\alpha}_i}^{X_i} = \left(\cos\frac{\hat{\theta}_i}{2} - e_{12}\sin\frac{\hat{\theta}_i}{2}\right)\left(\cos\frac{\hat{\alpha}_i}{2} - e_{23}\sin\frac{\hat{\alpha}_i}{2}\right) \tag{19}$$

and since ${}^{i-1}\boldsymbol{M}_i{}^{i-1}\tilde{\boldsymbol{M}}_i = 1$, we obtain the inverse transformation ${}^{i}\boldsymbol{M}_{i-1}$

$${}^{i}\boldsymbol{M}_{i-1} = \tilde{\boldsymbol{M}}_{\hat{\alpha}_i}^{X_i}\tilde{\boldsymbol{M}}_{\hat{\theta}_i}^{Z_{i-1}} = \left(\cos\frac{\hat{\alpha}_i}{2} + e_{23}\sin\frac{\hat{\alpha}_i}{2}\right)\left(\cos\frac{\hat{\theta}_i}{2} + e_{12}\sin\frac{\hat{\theta}_i}{2}\right) \tag{20}$$

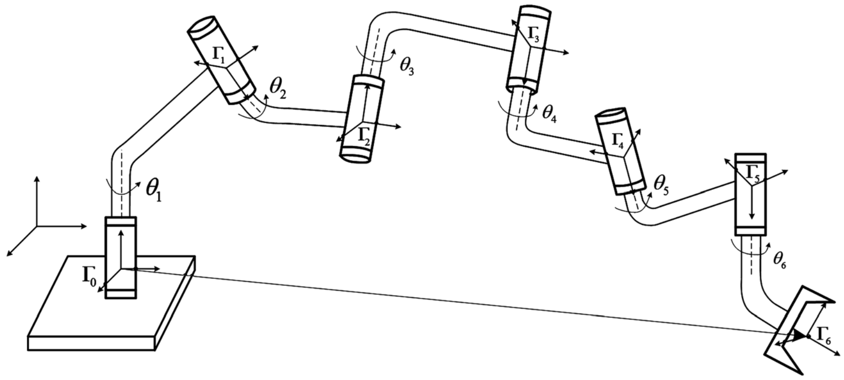

Fig. 5 Schematic diagram of a general serial 6R robot manipulator

3.2 Kinematics Modeling. Given a general serial 6R robot manipulator shown in Fig. 5, we can write the kinematics equation in conformal space $\boldsymbol{G}_{4,1,0}$ using a succession of *motors* and the dual angle as follows:

$$
{}^0\boldsymbol{M}_6 = {}^0\boldsymbol{M}_1(\hat{\theta}_1,\hat{\alpha}_1)\,{}^1\boldsymbol{M}_2(\hat{\theta}_2,\hat{\alpha}_2)\,{}^2\boldsymbol{M}_3(\hat{\theta}_3,\hat{\alpha}_3)\,{}^3\boldsymbol{M}_4(\hat{\theta}_4,\hat{\alpha}_4) \cdot {}^4\boldsymbol{M}_5(\hat{\theta}_5,\hat{\alpha}_5)\,{}^5\boldsymbol{M}_6(\hat{\theta}_6,\hat{\alpha}_6) \tag{21}
$$

where $\hat{\theta}_i = \theta_i + \varepsilon d_i, \hat{\alpha}_i = \alpha_i + \varepsilon a_i \ (i = 1, 2, ..., 6)$
θ_i is the joint variable.
and combing Eqs. (19) and (21), we can get the result of ${}^0\boldsymbol{M}_6$ in the following form:

$$
{}^0\boldsymbol{M}_6 = m_0 + m_1 e_{23} + m_2 e_{31} + m_3 e_{12} + \varepsilon(n_0 + n_1 e_{23} + n_2 e_{31} + n_3 e_{12}) \tag{22}
$$

where m_i and $n_i (i = 0, 1, 2, 3)$ are the functions of D_H parameters of 6R manipulator structure and satisfy the following two constraints Eq. (23), i.e., rigid body motions are parameterized by eight parameters subject to two constraints in Conformal space $\boldsymbol{G}_{4,1,0}$ [26]

$$
\begin{aligned}
m_0^2 + m_1^2 + m_2^2 + m_3^2 &= 1 \\
m_0 n_0 + m_1 n_1 + m_2 n_2 + m_3 n_3 &= 0
\end{aligned} \tag{23}
$$

Furthermore, the position and orientation of the end effector with respect to the base frame can be expressed by 4×4 homogeneous transformation matrix ${}^0\boldsymbol{T}_6$ in 3D Euclidean space $\boldsymbol{E}^3$ while motor ${}^0\boldsymbol{M}_6$ in conformal space $\boldsymbol{G}_{4,1,0}$. The transformation from ${}^0\boldsymbol{M}_6$ to ${}^0\boldsymbol{T}_6$ using Eqs. (18) and (22) can be formulated as follows:

$$
{}^0\boldsymbol{P}_6 = -2\begin{bmatrix} -m_1 & m_0 & m_3 & -m_2 \\ -m_2 & -m_3 & m_0 & m_1 \\ -m_3 & m_2 & -m_1 & m_0 \end{bmatrix}\begin{bmatrix} n_0 \\ n_1 \\ n_2 \\ n_3 \end{bmatrix}
$$

$$
{}^0\boldsymbol{R}_6 = \begin{bmatrix} 1 - 2(m_2^2 + m_3^2) & 2(m_1 m_2 + m_0 m_3) & 2(m_1 m_3 - m_0 m_2) \\ 2(m_1 m_2 - m_0 m_3) & 1 - 2(m_1^2 + m_3^2) & 2(m_2 m_3 + m_0 m_1) \\ 2(m_1 m_3 + m_0 m_2) & 2(m_2 m_3 - m_0 m_1) & 1 - 2(m_1^2 + m_2^2) \end{bmatrix} \tag{24}
$$

According to Eq. (24), we can compute the corresponding transformation from ${}^0\boldsymbol{T}_6$ to ${}^0\boldsymbol{M}_6$ very easily.

For the IKP, both the position and orientation of the end effector are given, i.e., ${}^0\boldsymbol{M}_6$ known. We can rewrite Eq. (21) as follows:

$$
{}^0\boldsymbol{M}_1(\hat{\theta}_1,\hat{\alpha}_1)\,{}^1\boldsymbol{M}_2(\hat{\theta}_2,\hat{\alpha}_2)\,{}^2\boldsymbol{M}_3(\hat{\theta}_3,\hat{\alpha}_3)\,{}^3\boldsymbol{M}_4(\hat{\theta}_4,\hat{\alpha}_4) = {}^0\boldsymbol{M}_6\,{}^5\tilde{\boldsymbol{M}}_6(\hat{\theta}_6,\hat{\alpha}_6)\,{}^4\tilde{\boldsymbol{M}}_5(\hat{\theta}_5,\hat{\alpha}_5) \tag{25}
$$

Let the left hand and right hand of Eq. (25) be expressed as $\boldsymbol{M}_L$ and $\boldsymbol{M}_R$, respectively, we will get the same form with Eq. (22)

$$
\boldsymbol{M}_L = \lambda_{l0} + \lambda_{l1} e_{23} + \lambda_{l2} e_{31} + \lambda_{l3} e_{12} + \varepsilon(\mu_{l0} + \mu_{l1} e_{23} + \mu_{l2} e_{31} + \mu_{l3} e_{12}) \tag{26}
$$

$$
\boldsymbol{M}_R = \lambda_{r0} + \lambda_{r1} e_{23} + \lambda_{r2} e_{31} + \lambda_{r3} e_{12} + \varepsilon(\mu_{r0} + \mu_{r1} e_{23} + \mu_{r2} e_{31} + \mu_{r3} e_{12}) \tag{27}
$$

where $\lambda_{li}, \mu_{li}, \lambda_{ri}, \mu_{ri} \ (i = 0, 1, 2, 3)$ are also the functions of D_H parameters of 6R manipulator structure.

3.3 Solution Process. Substituting D_H parameters of a general robot manipulator into Eqs. (26) and (27), and in terms of $\lambda_{li} = \lambda_{ri}, \mu_{li} = \mu_{ri} \ (i = 0, 1, 2, 3)$, we will get

$$
\begin{aligned}
\boldsymbol{L}_{8\times16}\boldsymbol{C}_{1234} &= \boldsymbol{R}_{8\times4}\begin{bmatrix} c_5 & c_6 \\ c_5 & s_6 \\ s_5 & c_6 \\ s_5 & s_6 \end{bmatrix} \\
\begin{bmatrix} \boldsymbol{L}_1 \\ \boldsymbol{L}_2 \end{bmatrix}\boldsymbol{C}_{1234} &= \begin{bmatrix} \boldsymbol{R}_1 \\ \boldsymbol{R}_2 \end{bmatrix}\begin{bmatrix} c_5 & c_6 \\ c_5 & s_6 \\ s_5 & c_6 \\ s_5 & s_6 \end{bmatrix}
\end{aligned} \tag{28}
$$

where $c_k = \cos(\theta_k/2), s_k = \sin(\theta_k/2), (k = 1, 2, \ldots 6)$

$$
\boldsymbol{C}_{1234} = [c_1c_2c_3c_4, c_1c_2c_3s_4, c_1c_2s_3c_4, c_1c_2s_3s_4, c_1s_2c_3c_4, c_1s_2c_3s_4, c_1s_2s_3c_4, c_1s_2s_3s_4, s_1c_2c_3c_4, s_1c_2c_3s_4, s_1c_2s_3c_4, s_1c_2s_3s_4, s_1s_2c_3c_4, s_1s_2c_3s_4, s_1s_2s_3c_4, s_1s_2s_3s_4]^T
$$

The elements of $\boldsymbol{L}_j \in \boldsymbol{R}^{4\times16}, \boldsymbol{R}_j \in \boldsymbol{R}^{4\times4} (j = 1, 2)$ are relied on D_H parameters of manipulator structure.

Further, Eq. (28) can be written as follows:

$$
(\boldsymbol{R}_1^{-1}\boldsymbol{L}_1 - \boldsymbol{R}_2^{-1}\boldsymbol{L}_2)\boldsymbol{C}_{1234} = \boldsymbol{0} \tag{29}
$$

Let $\boldsymbol{C}_{1234}$ be divided by $c_1c_2c_3c_4$ and then introduce $t_i = s_i/c_i = \tan(\theta_i/2), (i = 1, 2, 3, 4)$, the equivalent form of Eq. (29) is given by

$$
(\boldsymbol{R}_1^{-1}\boldsymbol{L}_1 - \boldsymbol{R}_2^{-1}\boldsymbol{L}_2)\boldsymbol{\Omega}_{1234} = \boldsymbol{0} \tag{30}
$$

where

$$\boldsymbol{\Omega}_{1234} = [1, t_4, t_3, t_3t_4, t_2, t_2t_4, t_2t_3, t_2t_3t_4, t_1, t_1t_4, t_1t_3, t_1t_3t_4, t_1t_2, t_1t_2t_4, t_1t_2t_3, t_1t_2t_3t_4]^T.$$

Moreover, we extract the variable t_1 from Eq. (30) and give another equivalent form of Eq. (29)

$$\boldsymbol{F}(t_1, t_2, t_3, t_4) = \boldsymbol{N}(t_1)\boldsymbol{\Delta}_{234} = \boldsymbol{0} \tag{31}$$

where $\boldsymbol{N}(t_1) \in \boldsymbol{R}^{4\times 8}$, $\boldsymbol{\Delta}_{234} = [1, t_2, t_3, t_4, t_2t_3, t_3t_4, t_2t_3t_4]^T$.

3.3.1 Solving θ_1, θ_3, and θ_4. According to the principle of resultant elimination [7], we can construct the following determinant:

$$H(t_2, t_3, t_4, \alpha, \beta, \gamma) = \begin{vmatrix} F_1(t_2, t_3, t_4) & F_2(t_2, t_3, t_4) & F_3(t_2, t_3, t_4) & F_4(t_2, t_3, t_4) \\ F_1(\alpha, t_3, t_4) & F_2(\alpha, t_3, t_4) & F_3(\alpha, t_3, t_4) & F_4(\alpha, t_3, t_4) \\ F_1(\alpha, \beta, t_4) & F_2(\alpha, \beta, t_4) & F_3(\alpha, \beta, t_4) & F_4(\alpha, \beta, t_4) \\ F_1(\alpha, \beta, \gamma) & F_2(\alpha, \beta, \gamma) & F_3(\alpha, \beta, \gamma) & F_4(\alpha, \beta, \gamma) \end{vmatrix} \tag{32}$$

Further, an equation can be written as follows:

$$F(t_2, t_3, t_4, \alpha, \beta, \gamma) = \frac{H(t_2, t_3, t_4, \alpha, \beta, \gamma)}{(t_2 - \alpha)(t_3 - \beta)(t_4 - \gamma)} = \boldsymbol{U}\boldsymbol{\Sigma}(t_1)\boldsymbol{V}^T = 0 \tag{33}$$

where $\alpha, \beta, \gamma \in \boldsymbol{R}$ are new variables.

$$\boldsymbol{U} = [1, \alpha, \beta, \alpha\beta, \alpha^2, \alpha^2\beta], \quad \boldsymbol{V} = [1, t_3, t_4, t_3t_4, t_4^2, t_3t_4^2]$$

Each element of $\boldsymbol{\Sigma} \in \boldsymbol{R}^{6\times 6}$ is only the fourth degree polynomial of variable t_1. By the corresponding column transformation and extracting the common factor $(1 + t_1^2)$, we get a 16th degree equation of variable t_1. However, ill-condition problem will appear when solving the 16th degree equation in some cases, and the requirement for accuracy cannot be guaranteed. Therefore, we can reduce the problem of roots to the eigenvalue algorithm of the resulting matrix [4].

We express the matrix $\boldsymbol{\Sigma}$ as

$$\boldsymbol{\Sigma} = \boldsymbol{K}_0t_1^4 + \boldsymbol{K}_1t_1^3 + \boldsymbol{K}_2t_1^2 + \boldsymbol{K}_3t_1 + \boldsymbol{K}_4 \tag{34}$$

where $\boldsymbol{K}_0, \boldsymbol{K}_1, \boldsymbol{K}_2, \boldsymbol{K}_3$, and $\boldsymbol{K}_4$ are 6×6 constant matrices.

Then, when the matrix $\boldsymbol{K}_0$ is well-conditioned, we use Eqs. (33) and (34) to construct 24×24 matrix $\boldsymbol{E}$ of the following form:

$$\boldsymbol{E} = \begin{bmatrix} \boldsymbol{0} & I & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & I & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & I \\ -\boldsymbol{K}_0^{-1}\boldsymbol{K}_4 & -\boldsymbol{K}_0^{-1}\boldsymbol{K}_3 & -\boldsymbol{K}_0^{-1}\boldsymbol{K}_2 & -\boldsymbol{K}_0^{-1}\boldsymbol{K}_1 \end{bmatrix} \tag{35}$$

where $\boldsymbol{0}, \boldsymbol{I}$ are 6×6 null and identity matrices, respectively. The eigenvalues of $\boldsymbol{E}$ correspond exactly to the solutions of t_1 (including eight imaginary extraneous roots) [27]. Furthermore, the eigenvectors of $\boldsymbol{E}$ corresponding to the eigenvalues t_1 have the form

$$\boldsymbol{P} = \left[\boldsymbol{p}, t_1\boldsymbol{p}, t_1^2\boldsymbol{p}, t_1^3\boldsymbol{p}\right]^T \tag{36}$$

where $\boldsymbol{p}$ is the 6×1 vector whose elements are not equal but proportional to the corresponding ones of vector $\boldsymbol{V}$ in Eq. (33). Thus, the eigenvectors of $\boldsymbol{E}$ can be used to computing t_3 and t_4. Finally, we can get the values of θ_1, θ_3, and θ_4 using the formula $\theta_i = 2a\tan(t_i), i = 1, 3, 4$.

However, the matrix $\boldsymbol{K}_0$ may be ill-conditioned, so we use Eqs. (33) and (34) and reduce it to a generalized eigenvalue problem by constructing two 24×24 matrices, $\boldsymbol{E}_1$ and $\boldsymbol{E}_2$ as follows:

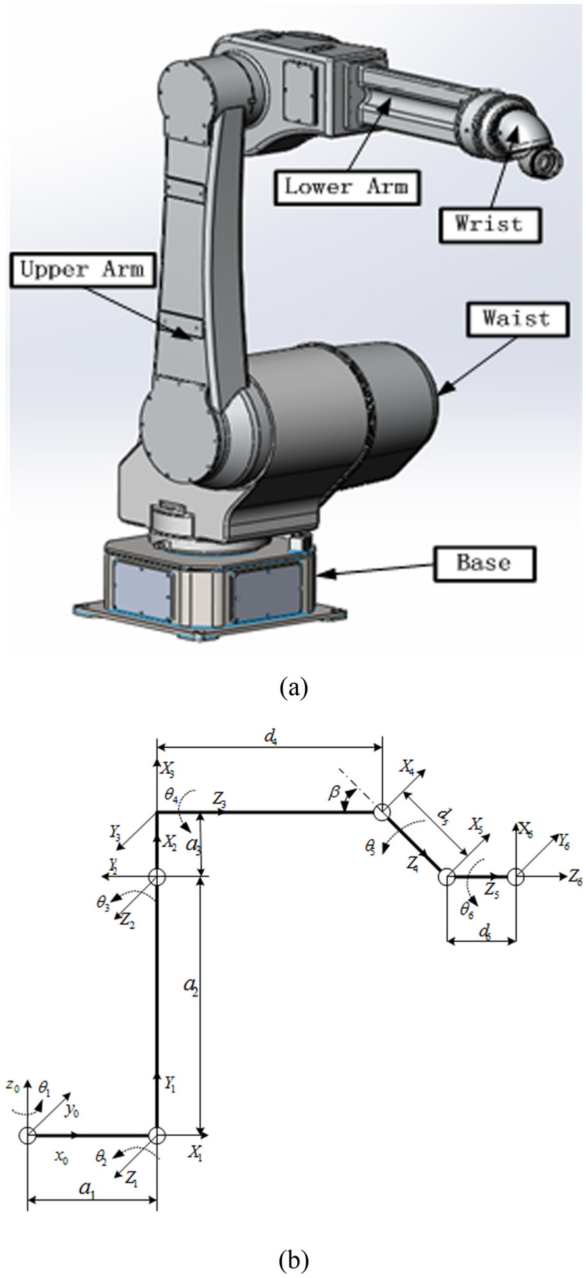

Fig. 6 The spray painting robot

Table 1 D_H parameters of the spray painting robot

Link i	$a_i(m)$	α_i(deg)	$d_i(m)$	θ_i(deg)
1	0.270	90	0	θ_1
2	1.300	0	0	θ_2
3	0.0425	90	0	θ_3
4	0	70	1.300	θ_4
5	0	−70	0.1089	θ_5
6	0	0	0.082	θ_6

Table 2 Sixteen solutions of IKP

	$\theta_1(^0)$	$\theta_2(^0)$	$\theta_3(^0)$	$\theta_4(^0)$	$\theta_5(^0)$	$\theta_6(^0)$
Solution 1	13.0291	−36.0910	146.7909	−124.7637	−101.2028	−44.5777
Solution 2	9.5166	−36.2019	142.4923	11.7055	99.0468	85.7735
Solution 3	160.5708	155.6891	−153.1752 + 46.4686i	−134.1849 + 221.6922i	1.4921 − 1.2080i	26.6476
Solution 4	160.5708	155.6891	−153.1752 − 46.4686i	−134.1849 − 221.6922i	1.4921 + 1.2080i	26.6476
Solution 5	68.4694	−50.1330	−104.7204 + 62.8655i	−167.3949 + 219.4235i	−179.3736 − 41.0370i	−5.3605
Solution 6	68.4694	−50.1330	−104.7204 − 62.8655i	−167.3949 − 219.4235i	−179.3736 + 41.0370i	−5.3605
Solution 7	−96.6069	−128.0163	−76.1061 + 63.5234i	7.4316 − 227.0604i	−179.4279 + 40.6667i	2.9813
Solution 8	−96.6069	−128.0163	−76.1061 − 63.5234i	7.4316 + 227.0604i	−179.4279 − 40.6667i	2.9813
Solution 9	−12.5370	26.1929	−26.5834 + 48.0091i	37.0611 − 190.2111i	1.7105 + 2.2096i	42.5233
Solution 10	−12.5370	26.1929	−26.5834 − 48.0091i	37.0611 + 190.2111i	1.7105 − 2.2096i	42.5233
Solution 11	−170.4507	−173.3506	89.5168 + 48.9314i	−162.0267 + 9.6638i	77.2939 − 22.3301i	81.8041
Solution 12	−170.4507	−173.3506	89.5168 − 48.9314i	−162.0267 − 9.6638i	77.2939 + 22.3301i	81.8041
Solution 13	−166.7017	−173.3615	84.9895 + 47.8752i	46.5011 − 2.3285i	−79.1824 + 21.8929i	−62.9339
Solution 14	−166.7017	−173.3615	84.9895 − 47.8752i	46.5011 + 2.3285i	−79.1824 − 21.8929i	−62.9339
Solution 15	13.0447	19.7764	36.7486	−125.2456	−52.8692	−92.6375
Solution 16	10.0	20.0	30.0	40.0	50.0	60.0

$$E_1 = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{K}_0 \end{bmatrix} \quad E_2 = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{K}_4 & -\boldsymbol{K}_3 & -\boldsymbol{K}_2 & -\boldsymbol{K}_1 \end{bmatrix} \tag{37}$$

where $\boldsymbol{0}, \boldsymbol{I}$ are 6×6 null and identity matrices, respectively. The solutions of t_1 correspond exactly to the eigenvalues of the generalized eigenvalue problem $\boldsymbol{E}_1 - t_1\boldsymbol{E}_2$ and the eigenvectors have the same structure as Eq. (33). Finally, the solutions of θ_1, θ_3, and θ_4 are obtained with the same procedure.

3.3.2 Solving θ_2, θ_5, and θ_6. Substituting each solution of t_1, t_3, and t_4 into Eq. (30) correspondingly, we can work out the corresponding 16 solutions of variable t_2. Then, the values of θ_2 can be worked out using the formula $\theta_2 = 2a\tan(t_2)$.

Similarly, substituting the solutions of $\theta_1, \theta_2, \theta_3$, and θ_4 into Eq. (28), we can work out the corresponding 16 solutions of θ_5 and θ_6 linearly.

4 Numerical Example

In this section, we present a numerical example to implement this new method efficiently on the IKP of the Spray Painting robot with offset wrist that is researched and developed independently by Kunshan Industrial Robot Research Centre Co. Ltd, see Fig. 6(*a*). Coordinate systems and related parameters of the robot kinematic are attached to the Spray Painting robot by means of D_H Convention as shown in Fig. 6(*b*). Table 1 was the D_H parameters of the Spray Painting robot.

Given a set of joint values $\theta = \{\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6\} = \{10\,\text{deg}, 20\,\text{deg}, 30\,\text{deg}, 40\,\text{deg}, 50\,\text{deg}, 60\,\text{deg}\}$, the position and orientation of the end effector $\boldsymbol{M}$ in Conformal space $\boldsymbol{G}_{4,1,0}$ is computed by Eq. (21) as follows:

$$\begin{aligned}\boldsymbol{M} = [&-0.2556, -0.6554, 0.6881, 0.1780,\\ &-0.6712, 0.5130, -0.0225, 1.0118]\end{aligned}$$

And we can further compute the corresponding position and orientation $\boldsymbol{T}_{\text{end}}$ in 3D Euclidean space $\boldsymbol{E}^3$ explicitly by Eq. (24)

$$T_{\text{end}} = \begin{bmatrix} -0.0103 & -0.9929 & 0.1185 & 2.5424 \\ -0.8109 & 0.0776 & 0.5800 & 0.5737 \\ -0.5851 & -0.0901 & -0.8059 & -0.3918 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Then, we use the method described in Sec. 3 to calculate the inverse kinematics solutions of the Spray Painting robot. A computer program is developed to implement the method and it takes about 6.933 ms on the platform of Intel i5-2320@3.0 GHz, RAM 4GB@1066 MHz to obtain the set of all 16 solutions which are listed in Table 2.

In Table 2, there are four real solutions and the 16th solution is equal to the given joint angles. In order to verify the quality of these solutions, each set of solutions was substituted into Eq. (21) to compute the errors relative to the corresponding position and orientation $\boldsymbol{M}$ of the end effector. The calculation for the maximum error can be written as follows:

$$\begin{aligned}\text{err} &= \max\left\{\sum_{j=1}^{8}\left|\boldsymbol{M}_j - \left({}^0\boldsymbol{M}_{1i}{}^1\boldsymbol{M}_{2i}{}^2\boldsymbol{M}_{3i}{}^3\boldsymbol{M}_{4i}{}^4\boldsymbol{M}_{5i}{}^5\boldsymbol{M}_{6i}\right)_j\right|\right\}\\ &= 7.7299 \times 10^{-8} \quad (i = 1, 2, \ldots, 16)\end{aligned} \tag{38}$$

The results show that the method meets the accuracy and real-time requirements for spray painting, and achieves the aim to the solution of inverse kinematics for 6R robot manipulator with offset wrist.

5 Conclusion

The paper provided an efficient method based on Geometric Algebra for computing the solutions to the inverse kinematics problem of the 6R robot manipulators with offset wrist, and yielded a set of all 16 solutions with symbolic elimination technique successfully. Further, a spray painting robot with offset wrist verified the effectiveness, and real-time of this method in detail. In addition, this method facilitates an elegant, simple representation and computation for the solutions to the IKP of 6R robot manipulators. It is believed that this method can be directly extended to general robot manipulators and completely automatized. However, the singularity, joint limitations and optimum solution of robot manipulator needs further research in future work.

Acknowledgment

This work was supported by the National Science and Technology Major Project 15 kg spraying robot equipment' (No. 2010zx04008-041) in CHINA.

References

[1] Siciliano, B., and Khatib, O., 2008, *Springer Handbook of Robotics*, Springer-Verlag, New York Inc.

[2] Bingul, Z., Ertunc, H. M., and Oysu, C., 2005, "Comparison of Inverse Kinematics Solutions Using Neural Network for 6R Robot Manipulator With Offset," Proceedings of the ICSC Congress on Computational Intelligence Methods and Applications, pp. 1–5.

[3] Raghavan, M., and Roth, B., 1993, "Inverse Kinematics of the General 6R Manipulator and Related Linkages," ASME J. Mech. Des., **115**(3), pp. 502–508.
[4] Manocha, D., and Canny, J. F., 1994, "Efficient Inverse Kinematics for General 6R Manipulators," IEEE Trans. Rob. Autom., **10**(5), pp. 648–657.
[5] Aspragathos, N. A., and Dimitros, J. K., 1998, "A Comparative Study of Three Methods for Robot Kinematics," IEEE Trans. Syst., Man, Cybern., Part B: Cybern., **28**(2), pp. 135–145.
[6] Husty, M. L., Pfurner, M., and Shrocker, H. P., 2007, "A New and Efficient Algorithm for the Inverse Kinematics of a General Serial 6R Manipulator," Mech. Mach. Theory, **42**(1), pp. 66–81.
[7] Qiao, S., Liao, Q., Wei, S., and Su, H. J., 2010, "Inverse Kinematic Analysis of the General 6R Serial Manipulators Based on Double Quaternions," Mech. Mach. Theory, **45**(2), pp. 193–199.
[8] Cheng, H., and Gupta, K. C., 1991, "A Study of Robot Inverse Kinematics BasedUupon the Solution of Differential Equations," J. Rob. Syst., **8**(2), pp. 159–175.
[9] Olsen, A. L., and Petersen, H. G., 2011, "Inverse Kinematics by Numerical and Analytical Cyclic Coordinate Descent," Robotica, **29**(3), pp. 619–626.
[10] Zhang, X., and Nelson, C. A., 2011, "Multiple-Criteria Kinematic Optimization for the Design of Spherical Serial Mechanisms Using Genetic Algorithms," ASME J. Mech. Des., **133**, pp. 1–11.
[11] Olaru, A., Olaru, S., and Paune, D., 2011, "Assisted Research and Optimization of the Proper Neural Network Solving the Inverse Kinematics Problem," Proceedings of 2011 International Conference on Optimization of the Robots and Manipulators, Romania, pp. 26–28.
[12] Feng, Y., Yao-nan, W., and Yi-min, Y., 2012, "Inverse Kinematics Solution for Robot Manipulator Based on Neural Network Under Joint Subspace," Int. of Comput. Commun., **7**(3), pp. 459–472.
[13] Hestens, D., and Sobczyk, G., 1987, *Clifford Algebra to Geometric Calculus: A Unified Language for Mathematics and Physics*, Springer-Verlag, Berlin, Heidelberg, New York.
[14] Hestens, D., 2001, "Old Wine in New Bottles: A New Algebraic Framework for Computational Geometry," *Advances in Geometric Algebra With Applications in Science and Engineering*, E. Bayro-Corrochano, and G. Sobczyk, eds., Birkauser, Boston, pp. 1–14.
[15] Zamora, J., and Bayro-Corrochano, E., 2004, "Inverse Kinematics, Fixation and Grasping Using Conformal Geometric Algebra," IROS 2004, Sendai, Japan.
[16] Hildenbrand, D., 2006, "Geometric Computing in Computer Graphics and Robotics using Conformal Geometric Algebra," Ph.D. thesis, Darmstadt University of Technology, Darmstadt.
[17] Hildenbrand, D., Lange, H., and Stock, F., 2008, "Efficient Inverse Kinematics Algorithm Based on Conformal Geometric Algebra," Proceedings of the 3rd International Conference on Computer Graphics Theory and Applications, Medeira, Portugal.
[18] Aristidou, A., and Lasenby, J., 2011, "Inverse Kinematics Solutions Using Conformal Geometric Algebra," Guide to Geometric Algebra in Practice, Vol. 1, Springer Verlag, pp. 47–62.
[19] Aristidou, A., and Lasenby, J., 2011, "FABRIK: A Fast, Iterative Solver for the Inverse Kinematics Problem," Graphical Models, **73**(5), pp. 243–260.
[20] Dorst, L., Fontijne, D., and Mann, S., 2007, *Geometric Algebra for Computer Science: An Object-Oriented Approach to Geometry*, Morgan Kaufmann Publishers/Elsevier, San Francisco, CA.
[21] Cheng, H. H., 1994, Programming With Dual Numbers and its Applications in Mechanism Design. Engineering With Computers, **10**(4), pp. 212–229.
[22] Bayro-Corrochano, E., and Scheuermann, G., 2009, *Geometric Algebra Computing for Engineering and Computer Science*, Springer–Verlag, New York.
[23] Hestenes, D., 2010, "New Tools for Computational Geometry and Rejuvenation of Screw Theory," *Geometric Algebra Computing in Engineering and Computer Science*, E. Bayro-Corrochano, and G. Scheuermann, eds., Springer-Verlag, pp. 3–35.
[24] Bayro-Corrochano, E., 2010, *Geometric Computing for Wavelet Transforms, Robot Vision, Learning, Control and Action*, Spring-Verlag, London, pp. 169–203.
[25] Selig, J. M., 2005, *Geometric Fundamentals of Robotics, Monographs in Computer Science*, Springer, New York, pp. 206–278.
[26] Li, H., Hestenes, D., and Rockwood, A., 2001, "Generalized Homogeneous Coordinates for Computational Geometry," *Geometric Computing With Clifford Algebra*, G. Sommer, ed., Springer-Verlag, pp. 25–58.
[27] Gohberg, I., Lancaster, P., and Rodman, L., 1982, *Matrix Polynomials*, Academic Press, New York.