Turk J Elec Eng & Comp Sci, Vol.20, No.4, 2012, © TÜBİTAK
doi:10.3906/elk-1011-971

A new formulation method for solving kinematic problems of multiarm robot systems using quaternion algebra in the screw theory framework

Emre SARIYILDIZ*, **Hakan TEMELTAŞ**
Department of Control Engineering, İstanbul Technical University, 34469 İstanbul-TURKEY
e-mails: esariyildiz@itu.edu.tr, hakan.temeltas@itu.edu.tr

Received: 28.11.2010

Abstract

We present a new formulation method to solve the kinematic problem of multiarm robot systems. Our major aims were to formulize the kinematic problem in a compact closed form and avoid singularity problems in the inverse kinematic solution. The new formulation method is based on screw theory and quaternion algebra. Screw theory is an effective way to establish a global description of a rigid body and avoids singularities due to the use of the local coordinates. The dual quaternion, the most compact and efficient dual operator to express screw displacement, was used as a screw motion operator to obtain the formulation in a compact closed form. Inverse kinematic solutions were obtained using Paden-Kahan subproblems. This new formulation method was implemented into the cooperative working of 2 Stäubli RX160 industrial robot-arm manipulators. Simulation and experimental results were derived.

Key Words: *Cooperative working of multiarm robot systems, dual quaternion, industrial robot application, screw theory, singularity-free inverse kinematic*

1. Introduction

Multiarm robot configurations offer the potential to overcome many difficulties by increased manipulation ability and versatility [1,2]. For instance, single-arm industrial robots cannot perform their roles in the many fields in which operators do the job with their 2 arms [3]. Multiarm robot systems can also manipulate bulky objects whose weight exceeds the working capacities of the individually cooperating participants [4]. For these reasons, the needs for multiarm robot manipulators are increasing in many industrial fields [3]. In addition to industrial robotics, the main application areas of the cooperative working of multiarm robot systems are medical robotics, telerobotics and humanoid robotics [5-7].

A fundamental research task of cooperative manipulation is to find the appropriate way to control the system of robots and objects in the work space at any stage of the cooperative work. This requires an exact

*Corresponding author: Department of Control Engineering, İstanbul Technical University, 34469 İstanbul-TURKEY

understanding of the physical nature of the cooperative system and derivation of the mathematical basis for its description. In the realization of this goal, the first crucial problem is kinematic uncertainty [8].

In the cooperative working of multiarm robot systems, a closed-chain mechanism comes into existence from the open-chain mechanisms [9]. Regarding the kind of object grasping, the closed-chain mechanism is usually redundant. The kinematic problem of robots working in cooperation is a highly complex problem because of redundancy. There are several studies based on differential kinematics to solve the kinematic problem of the closed-chain mechanism [10-12]. Differential kinematics-based solutions are used as Jacobian operators for velocity mapping. There are 2 main disadvantages of this method. The first is that joint angles are obtained by numerical integration of the joint velocities which suffer from errors due to both long-term numerical integration drift and incorrect initial joint angles. The second is that it needs inversion of the Jacobian matrix in the inverse kinematic solution. Taking the inverse of a matrix is computationally hard and even impossible at singular points. Since the closedchain mechanism is redundant the Jacobian matrix is always singular. There are, in general, 4 main techniques to cope with the kinematic singularity problems. These are avoidance of the singular configuration method, the robust inverse method, a normal form approach method and the extended Jacobian method [13-16]. However, the given techniques have some disadvantages which include computational load and errors [17]

The closed-chain kinematic problem of the cooperative working of multiarm robots can also be solved by reducing the full kinematic problem into the appropriate subopen-chain kinematic problems. If all of the robot arms' positions and orientations can be determined appropriately, the closed-chain robot kinematic problem can be reduced to serial robot-arm kinematic problems. Several methods are used to solve the kinematic problem of multiarm robot systems using this approach. Chiacchio et al. used differential kinematics to solve the subopen-chain kinematic problem [4]. This method also has some disadvantages, which are similar to those of the differential kinematics-based closed-chain solution; however, there are only single-arm singularities in this case. Hemami and Zheng used the Denavit-Hartenberg convention, which is the most common method in robot kinematics to solve the subopen-chain kinematic problem [18,19]. These methods use 4×4 homogeneous transformation matrices as a point transformation operator and suffer from singularity problems [20].

The subopen-chain kinematic problem of multiarm robot systems can also be solved by using screw theory. There are 2 main advantages of using screw theory for describing rigid body kinematics. The first is that it allows a global description of rigid body motion that does not suffer from singularities due to the use of local coordinates. The second is that the screw theory provides a geometric description of rigid motion, which greatly simplifies the analysis of mechanisms [21]. Several applications of screw theory have been introduced in robot kinematics. Among these studies, Funda and Paul and Funda et al. analyzed transformation operators of screw motion. They found that dual operators are the best way to describe screw motion and also that the dual quaternion is the most compact and efficient dual operator to express screw displacement [22,23].

In this paper, a new formulation method to solve the kinematic problem of multiarm robot systems is presented. This new formulation method is based on screw theory and it uses the dual quaternion, which is the most compact and efficient dual operator to express screw displacement, as a screw motion operator. Thus, the proposed method provides a singularity-free and computationally efficient inverse kinematic solution for multiarm robot manipulators. The kinematic solution of the cooperative working of a dual-arm robot manipulator using this new formulation method is given in Section 6. This paper also includes the mathematical preliminaries for quaternion algebra in Section 2, screw theory by using quaternion algebra in Section 3, the kinematic scheme of an n-degrees of freedom (DOF) serial robot manipulator in Section 4, the kinematic model of a 6-DOF serial robot manipulators in Section 5, cooperative working of serial robot arms in Section 6, and simulation and experimental results of forward and inverse kinematic solutions of the cooperative working of dual-arm robot manipulator in Sections 7 and 8, respectively. Conclusions and future works are presented in the final section.

2. Mathematical preliminaries

2.1. Quaternion

In mathematics, the quaternions are hypercomplex numbers of rank 4, constituting a 4-dimensional vector space over the field of real numbers [24]. The quaternion can be represented in the form:

$$q = (q_S, \mathbf{q}_V), \tag{1}$$

where q_S is a scalar and $\mathbf{q}_V = (q_1, q_2, q_3)$ is a vector. The sum of 2 quaternions is then:

$$q_a + q_b = (q_{aS} + q_{bS}), (\mathbf{q_a}_V + \mathbf{q_b}_V), \tag{2}$$

and the product of 2 quaternions is:

$$q_a \otimes q_b = (q_{aS}q_{bS} - \mathbf{q_a}_V \cdot \mathbf{q_b}_V), (q_{aS}\mathbf{q_b}_V + q_{bS}\mathbf{q_a}_V + \mathbf{q_a}_V \times \mathbf{q_b}_V), \tag{3}$$

where "$\otimes$", "$\cdot$", and "$\times$" denote quaternion products, dot products, and cross products, respectively. The conjugate, norm, and inverse of the quaternion can be expressed in the forms given below.

$$q^* = (q_s, -\mathbf{q_v}) = (q_s, -q_1, -q_2, -q_3) \tag{4}$$

$$||q||^2 = q \otimes q^* = q_s^2 + q_1^2 + q_2^2 + q_3^2 \tag{5}$$

$$q^{-1} = \frac{1}{||q||^2} q^* \text{ and } ||q|| \neq 0 \tag{6}$$

These satisfy the relation $q^{-1} \otimes q = q \otimes q^{-1} = 1$. When $||q||^2 = 1$, we get a unit quaternion. Any quaternion q can be normalized by dividing its norm. For the unit quaternion, we have:

$$q^{-1} = q^*. \tag{7}$$

A unit quaternion can be defined as a rotation operator [25-27]. Rotation about an axis of $\mathbf{n}$ by an angle of θ can be expressed by using the unit quaternion given by:

$$q = \cos\left(\frac{\theta}{2}\right), \sin\left(\frac{\theta}{2}\right)\mathbf{n}. \tag{8}$$

Definition 1: Let q_a and q_b be 2 pure quaternions and the product of these 2 quaternions be:

$$q_a \otimes q_b = (q_{aS}q_{bS} - \mathbf{q_a}_V \cdot \mathbf{q_b}_V), (q_{aS}\mathbf{q_b}_V + q_{bS}\mathbf{q_a}_V + \mathbf{q_a}_V \times \mathbf{q_b}_V) = (-\mathbf{q_a}_V \cdot \mathbf{q_b}_V), (\mathbf{q_a}_V \times \mathbf{q_b}_V). \tag{9}$$

Let us then define 2 new functions by using the product of 2 pure quaternions given by the following. Let:

- $V\{q_a \otimes q_b\} = \mathbf{q_a}_V \times \mathbf{q_b}_V$ be the vector part of the quaternion multiplication, and
- $S\{q_a \otimes q_b\} = -(\mathbf{q_a}_V \cdot \mathbf{q_b}_V)$ be the scalar part of the quaternion multiplication.

2.2. Dual quaternion

The dual quaternion can be represented in the form:

$$\hat{q} = (\hat{q}_S, \hat{\mathbf{q}}_V) \text{ or } \hat{q} = q + \varepsilon q^o, \tag{10}$$

where $\hat{q}_S = q_S + \varepsilon q_S^0$ is a dual scalar, $\hat{\mathbf{q}}_V = \mathbf{q}_V + \varepsilon \mathbf{q}_V^{\mathbf{o}}$ is a dual vector, q and q^o are both quaternions, and ε is the dual factor [28]. The sum of 2 dual quaternions is then:

$$\hat{q}_a + \hat{q}_b = (q_a + q_b) + \varepsilon(q_a^o + q_b^o), \tag{11}$$

and the product of 2 dual quaternions is:

$$\hat{q}_a \Theta \hat{q}_b = (q_a \otimes q_b) + \varepsilon(q_a \otimes q_b^o + q_a^o \otimes q_b), \tag{12}$$

where "$\otimes$" and "Θ" denote the quaternion and dual quaternion products, respectively. The conjugate, norm, and inverse of the dual quaternion are similar to the quaternion's conjugate, norm, and inverse, respectively.

$$\hat{q}^* = q^* + \varepsilon(q^o)^* \tag{13}$$

$$||\hat{q}||^2 = \hat{q}\Theta\hat{q}^* \tag{14}$$

$$\hat{q}^{-1} = \frac{1}{||\hat{q}||^2}\hat{q}^* \text{ and } ||\hat{q}|| \neq 0 \tag{15}$$

When $||\hat{q}||^2 = 1$, we get a unit dual quaternion. For the unit dual quaternion, we have:

$$||\hat{q}||^2 = \hat{q}\Theta\hat{q}^* = \hat{q}^*\Theta\hat{q} = 1, \tag{16}$$

$$q \otimes q^* = 1 \quad , \quad q^* \otimes q^o + (q^o)^* \otimes q = 0. \tag{17}$$

The unit dual quaternion can be used as a rigid body transformation operator [29]. Although it has 8 parameters and it is not minimal, it is the most compact and efficient dual operator [22,23]. This transformation is very similar to pure rotation, although not for a point but rather for a line. A line in Plücker coordinates $L_a(\mathbf{m}, \mathbf{d})(\hat{l}_a = l_a + \varepsilon m_a$ in dual quaternion form; see Appendix) can be transformed to $L_b(\mathbf{m}, \mathbf{d})$ by using unit dual quaternions as follows:

$$\hat{l}_b = \hat{q}\Theta\hat{l}_a\Theta\hat{q}^*, \tag{18}$$

where $\hat{q}$ is the unit dual quaternion [30].

Definition 2: Let $\hat{q}_a$ and $\hat{q}_b$ be 2 dual quaternions and let the product of 2 dual quaternions be

$$\hat{q}_{ab} = \hat{q}_a\Theta\hat{q}_b = q_{ab} + \varepsilon q_{ab}^o = (q_{abS}, \mathbf{q_{ab}}_V) + \varepsilon(q_{abS}^o, \mathbf{q^o_{ab}}_V). \tag{19}$$

Let us then define 4 new functions by using the product of 2 dual quaternions and definition 1. Let:

- $S\{R\{\hat{q}_a\Theta\hat{q}_b\}\} = q_{abS}$ be the scalar part of the real part of multiplication,
- $S\{D\{\hat{q}_a\Theta\hat{q}_b\}\} = q_{abS}^o$ be the scalar part of the dual part of multiplication,
- $V\{R\{\hat{q}_a\Theta\hat{q}_b\}\} = \mathbf{q_{ab}}_V$ be the vector part of the real part of multiplication, and
- $V\{D\{\hat{q}_a\Theta\hat{q}_b\}\} = \mathbf{q^o_{ab}}_V$ be the vector part of the dual part of multiplication.

3. Screw theory

The elements of screw theory can be traced to the work of Chasles and Poinsot in the early 1800s. According to Chasles, all proper rigid body motions in 3-dimensional space, with the exception of pure translation, are equivalent to a screw motion (see Figure 1), that is, a rotation about a line together with a translation along the line [31,32].

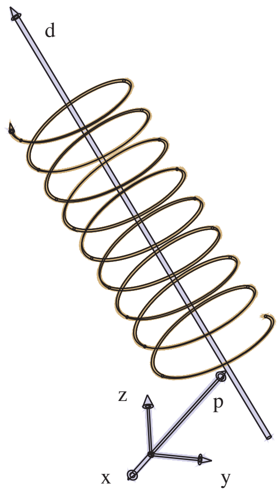


Figure 1. General screw motion.

The general screw motion operator can be represented by using a dual quaternion as follows:

$$\hat{q} = \cos\left(\frac{\hat{\theta}}{2}\right) + \sin\left(\frac{\hat{\theta}}{2}\right)\hat{d}, \tag{20}$$

where $\hat{\theta} = \theta + \varepsilon k$ and $\hat{d} = d + \varepsilon m$ are dual numbers. Here, θ and $d = [0, \mathbf{d}]$ indicate the rotation angle and the screw motion axis, respectively. $m = [0, \mathbf{p} \times \mathbf{d}]$ indicates the moment vector of the rotation axis, where $\mathbf{p}$ is any point on the direction vector of $\mathbf{d}$ and $k = \mathbf{d} \cdot \mathbf{t}$. Further details of general screw motion formulation using dual quaternions can be found in [30].

4. Manipulator kinematics

4.1. Forward kinematics

The forward kinematic problem is to determine the position and orientation of the end effector given the values for the joint variables of the robot. To find the forward kinematics of the serial robot manipulator, we followed these steps:

Step 1: Determine the joints' axis and moment vectors. First, the axis vectors that describe the motion of the joints are attached. The moment vectors of these axes are then obtained for revolute joints (see Appendix). Hence, the Plücker coordinate notations of these axes are obtained.

Step 2: Obtain transformation operators. For all joints, dual quaternion transformation operators can be obtained as follows:

$$\hat{q}_i = (\hat{q}_{Si}, \hat{\mathbf{q}}_{\mathbf{Vi}}) \text{ or } \hat{q}_i = q_i + \varepsilon q_i^o. \tag{21}$$

For prismatic joints:
$q_i = (1,0,0,0)$ and $q_i^o = (0, q_1^o, q_2^o, q_3^o)$ describe rotation and the amount of translation, respectively.

For revolute joints:
$q_i = \cos\left(\frac{\theta_i}{2}\right) + \sin\left(\frac{\theta_i}{2}\right)\mathbf{d_i}$ and $q_i^o = \frac{1}{2}(p_i - q_i \otimes p_i \otimes q_i^*) \otimes q_i$ or $q_i^o = [0, \sin\left(\frac{\theta_i}{2}\right)\mathbf{m_i}]$ describe rotation and the amount of translation, respectively.
Here, $i = 1, 2, ..., n$.

Step 3: Formulate rigid motion. Rigid motion formulation can be obtained by using Eq. (18). For an n-DOF robot manipulator, the general rigid body transformation operation is given by:

$$\hat{q}_{1n} = \hat{q}_1 \Theta \hat{q}_2 \Theta .....\Theta \hat{q}_n, \tag{22}$$

where $\hat{q}_{1n} = q_{1n} + \varepsilon q_{1n}^o$. The orientation and position of the end effector can be found as follows.

Let $\hat{l}_n = l_n + \varepsilon l_n^o$ and $\hat{l}_{n-1} = l_{n-1} + \varepsilon l_{n-1}^o$ be the n*th* and (n − 1)*th* joints' Plücker coordinate representations, respectively. Additionally, let $\hat{l}'_n = l'_n + \varepsilon l_n^{o\prime} = \hat{q}_{1n}\Theta\hat{l}_n\Theta\hat{q}_{1n}^*$ and $\hat{l}'_{n-1} = l'_{n-1} + \varepsilon l_{n-1}^{o\prime} = \hat{q}_{1n-1}\Theta\hat{l}_{n-1}\Theta\hat{q}_{1n-1}^*$ be the n*th* and (n − 1)*th* joints' Plücker coordinate representations after the transformation. The orientation of the end effector is $\hat{l}'_n$. The position of the end effector can be found using definitions 1 and 2 and Eq. (A.1) from the Appendix, given by:

$$\begin{aligned}\mathbf{p}_n = &(V\left\{R\left\{\hat{q}_{1n}\Theta\hat{l}_n\Theta\hat{q}_{1n}^*\right\}\right\} \times V\left\{D\left\{\hat{q}_{1n}\Theta\hat{l}_n\Theta\hat{q}_{1n}^*\right\}\right\})+\\ &(V\left\{R\left\{\hat{q}_{1n-1}\Theta\hat{l}_{n-1}\Theta\hat{q}_{1n-1}^*\right\}\right\}\times V\left\{D\left\{\hat{q}_{1n-1}\Theta\hat{l}_{n-1}\Theta\hat{q}_{1n-1}^*\right\}\right\}) \cdot V\left\{R\left\{\hat{q}_{1n}\Theta\hat{l}_n\Theta\hat{q}_{1n}^*\right\}\right\} * V\left\{R\left\{\hat{q}_{1n}\Theta\hat{l}_n\Theta\hat{q}_{1n}^*\right\}\right\}.\end{aligned} \tag{23}$$

4.2. Inverse kinematics

The inverse kinematic problem is to determine the values of the joint variables given the end effector's position and orientation. Paden-Kahan subproblems are used to obtain the inverse kinematic solution of the serial robot-arm manipulator [33-35]. The solution of the inverse kinematic problem of a 6-DOF serial robot-arm is given in the next section.

5. 6-DOF serial robot-arm kinematic model

In this section, the kinematic problem of a serial robot arm, which is shown in Figure 2, is solved by using the new formulation method.

5.1. Forward kinematics

Step 1: First, the axes of all joints should be determined.

$$\mathbf{d_1} = [0,0,1] \quad \mathbf{d_2} = [0,1,0] \quad \mathbf{d_3} = [0,1,0]$$

$$\mathbf{d_4} = [0,0,1] \quad \mathbf{d_5} = [0,1,0] \quad \mathbf{d_6} = [0,0,1] \tag{24}$$

The moment vectors of all axes must then be calculated.

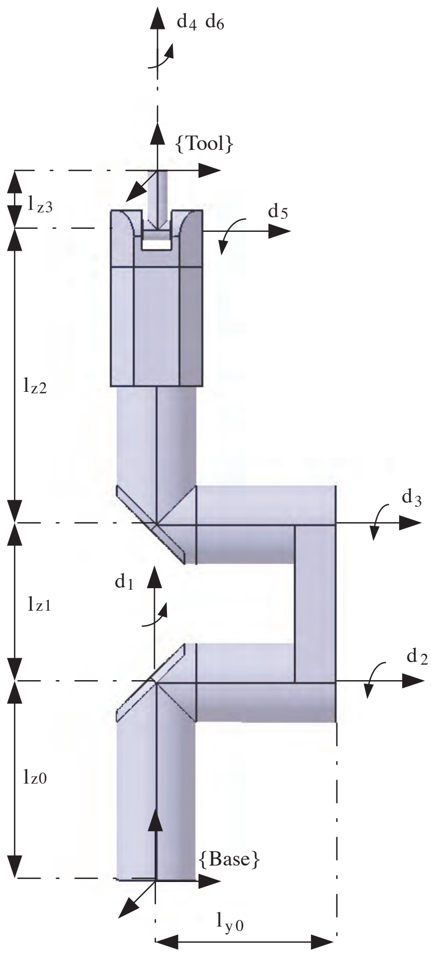


Figure 2. 6-DOF serial robot-arm manipulator in its reference configuration.

$$\mathbf{m_1} = \mathbf{p_1} \times \mathbf{d_1} \quad \mathbf{m_2} = \mathbf{p_2} \times \mathbf{d_2} \quad \mathbf{m_3} = \mathbf{p_3} \times \mathbf{d_3}$$

$$\mathbf{m_4} = \mathbf{p_4} \times \mathbf{d_4} \quad \mathbf{m_5} = \mathbf{p_5} \times \mathbf{d_5} \quad \mathbf{m_6} = \mathbf{p_6} \times \mathbf{d_6} \tag{25}$$

Here,

$$\mathbf{p_1} = [0, 0, lz_0], \mathbf{p_2} = [0, 0, lz_0], \mathbf{p_3} = [0, ly_0, lz_0 + lz_1],$$

$$\mathbf{p_4} = [0, 0, lz_0 + lz_1 + lz_2], \quad \mathbf{p_5} = [0, 0, lz_0 + lz_1 + lz_2], \quad \mathbf{p_6} = [0, 0, lz_0 + lz_1 + lz_2]. \tag{26}$$

Step 2: The transformation operator that is in dual quaternion form can be written using the axis and moment vectors and Eq. (21).

Step 3: Finally, the forward kinematic equation of serial robot manipulator can be obtained as follows:

$$\hat{l}_6' = l_6' + \varepsilon l_6^{o\prime} = \hat{q}_{16} \Theta \hat{l}_6 \Theta \hat{q}_{16}^* = \hat{q}_{16} \Theta (l_6 + \varepsilon l_6^o) \Theta \hat{q}_{16}^*,$$

$$\hat{l}_5' = l_5' + \varepsilon l_5^{o\prime} = \hat{q}_{15} \Theta \hat{l}_5 \Theta \hat{q}_{15}^* = \hat{q}_{15} \Theta (l_5 + \varepsilon l_5^o) \Theta \hat{q}_{15}^*, \tag{27}$$

where $\hat{q}_{16} = \hat{q}_1 \Theta \hat{q}_2 \Theta \hat{q}_3 \Theta \hat{q}_3 \Theta \hat{q}_5 \Theta \hat{q}_6$ and $\hat{q}_{15} = \hat{q}_1 \Theta \hat{q}_2 \Theta \hat{q}_3 \Theta \hat{q}_3 \Theta \hat{q}_5$. The orientation of the end effector is then $\hat{l}_6'$ and the position of the end effector is:

$$\begin{array}{l} \mathbf{p}_6 = (V\left\{R\left\{\hat{q}_{16}\Theta\hat{l}_6\Theta\hat{q}_{16}^*\right\}\right\} \times V\left\{D\left\{\hat{q}_{16}\Theta\hat{l}_6\Theta\hat{q}_{16}^*\right\}\right\})+ \\ (V\left\{R\left\{\hat{q}_{15}\Theta\hat{l}_5\Theta\hat{q}_{15}^*\right\}\right\} \times V\left\{D\left\{\hat{q}_{15}\Theta\hat{l}_5\Theta\hat{q}_{15}^*\right\}\right\}) \cdot V\left\{R\left\{\hat{q}_{16}\Theta\hat{l}_6\Theta\hat{q}_{16}^*\right\}\right\} * V\left\{R\left\{\hat{q}_{16}\Theta\hat{l}_6\Theta\hat{q}_{16}^*\right\}\right\} \end{array}. \tag{28}$$

5.2. Inverse kinematics

In the inverse kinematic problem of the serial robot manipulator, we have the position and orientation information of the end effector such that $\hat{q}_{in} = (q_{in}, q_{in}^o)$, where $q_{in} = (q_0, q_1, q_2, q_3)$; that is, the orientation of the end effector is the real part of the dual quaternion $\hat{q}_{in}$, and $q_{in}^o = (q_0^o, q_1^o, q_2^o, q_3^o)$, the position of the end effector, is the dual part of the of the dual quaternion $\hat{q}_{in}$. The general inverse kinematic problem should be converted into the appropriate Paden-Kahan subproblems (see Appendix) to obtain the inverse kinematic solution. This solution can be obtained as follows.

Step 1: First, we put 2 points at the intersection of the axes. The first is $\mathbf{p_w}$, which is at the intersection of the wrist axes, and the second is $\mathbf{p_b}$, which is at the intersection of the first 2 axes. The last 3 joints do not affect the position of point $\mathbf{p_w}$ and the first 2 joints do not affect the position of point $\mathbf{p_b}$. We can then easily write Eq. (21).

$$\left(\begin{array}{l} (V\left\{R\left\{\hat{q}_{13}\Theta\hat{l}_6\Theta\hat{q}_{13}^*\right\}\right\} \times V\left\{D\left\{\hat{q}_{13}\Theta\hat{l}_6\Theta\hat{q}_{13}^*\right\}\right\})+ \\ (V\left\{R\left\{\hat{q}_{13}\Theta\hat{l}_5\Theta\hat{q}_{13}^*\right\}\right\} \times V\left\{D\left\{\hat{q}_{13}\Theta\hat{l}_5\Theta\hat{q}_{13}^*\right\}\right\}) \cdot V\left\{R\left\{\hat{q}_{13}\Theta\hat{l}_6\Theta\hat{q}_{13}^*\right\}\right\} * V\left\{R\left\{\hat{q}_{13}\Theta\hat{l}_6\Theta\hat{q}_{13}^*\right\}\right\} \end{array}\right) - \left(\begin{array}{l} (V\left\{R\left\{\hat{q}_{12}\Theta\hat{l}_2\Theta\hat{q}_{12}^*\right\}\right\} \times V\left\{D\left\{\hat{q}_{12}\Theta\hat{l}_2\Theta\hat{q}_{12}^*\right\}\right\})+ \\ (V\left\{R\left\{\hat{q}_{12}\Theta\hat{l}_1\Theta\hat{q}_{12}^*\right\}\right\} \times V\left\{D\left\{\hat{q}_{12}\Theta\hat{l}_1\Theta\hat{q}_{12}^*\right\}\right\}) \cdot V\left\{R\left\{\hat{q}_{12}\Theta\hat{l}_2\Theta\hat{q}_{12}^*\right\}\right\} * V\left\{R\left\{\hat{q}_{12}\Theta\hat{l}_2\Theta\hat{q}_{12}^*\right\}\right\} \end{array}\right) = \mathbf{q_{in}^o} - \mathbf{p_b} \tag{29}$$

Using the property that the distance between the points is preserved by rigid motions and taking the magnitude of both sides of Eq. (21), we get:

$$\left\| \begin{array}{l} \left(\begin{array}{l} (V\left\{R\left\{\hat{q}_3\Theta\hat{l}_6\Theta\hat{q}_3^*\right\}\right\} \times V\left\{D\left\{\hat{q}_3\Theta\hat{l}_6\Theta\hat{q}_3^*\right\}\right\}) + (V\left\{R\left\{\hat{q}_3\Theta\hat{l}_5\Theta\hat{q}_3^*\right\}\right\} \times \\ V\left\{D\left\{\hat{q}_3\Theta\hat{l}_5\Theta\hat{q}_3^*\right\}\right\}) \cdot V\left\{R\left\{\hat{q}_3\Theta\hat{l}_6\Theta\hat{q}_3^*\right\}\right\} * V\left\{R\left\{\hat{q}_3\Theta\hat{l}_6\Theta\hat{q}_3^*\right\}\right\} \end{array}\right) - \\ \left((V\left\{R\left\{\hat{l}_2\right\}\right\} \times V\left\{D\left\{\hat{l}_2\right\}\right\}) + (V\left\{R\left\{\hat{l}_1\right\}\right\} \times V\left\{D\left\{\hat{l}_1\right\}\right\}) \cdot V\left\{R\left\{\hat{l}_2\right\}\right\} * V\left\{R\left\{\hat{l}_2\right\}\right\}\right) \end{array} \right\| = \left\|\mathbf{q_{in}^o} - \mathbf{p_b}\right\|. \tag{30}$$

Eq. (30) gives us subproblem 3 (see Appendix). The parameters of subproblem 3 are

$$\mathbf{a} = (V\left\{R\left\{\hat{l}_6\right\}\right\} \times V\left\{D\left\{\hat{l}_6\right\}\right\}) + (V\left\{R\left\{\hat{l}_5\right\}\right\} \times V\left\{D\left\{\hat{l}_5\right\}\right\}) \cdot V\left\{R\left\{\hat{l}_6\right\}\right\} * V\left\{R\left\{\hat{l}_6\right\}\right\},$$

$$\mathbf{b} = (V\left\{R\left\{\hat{l}_2\right\}\right\} \times V\left\{D\left\{\hat{l}_2\right\}\right\}) + (V\left\{R\left\{\hat{l}_1\right\}\right\} \times V\left\{D\left\{\hat{l}_1\right\}\right\}) \cdot V\left\{R\left\{\hat{l}_2\right\}\right\} * V\left\{R\left\{\hat{l}_2\right\}\right\}.$$

$\mathbf{l}$ is the axis of joint 3, that is, $\mathbf{d_3}$, and $\delta = \mathbf{q_{in}^o} - \mathbf{p_b}.\theta_3$ can be found by using subproblem 3.

Step 2: If we use the known θ_3 in Eq. (21), we then obtain:

$$(V\left\{R\left\{\hat{q}_{12}\Theta\hat{l}'_6\Theta\hat{q}^*_{12}\right\}\right\} \times V\left\{D\left\{\hat{q}_{12}\Theta\hat{l}'_6\Theta\hat{q}^*_{12}\right\}\right\})+ (V\left\{R\left\{\hat{q}_{12}\Theta\hat{l}'_5\Theta\hat{q}^*_{12}\right\}\right\} \times V\left\{D\left\{\hat{q}_{12}\Theta\hat{l}'_5\Theta\hat{q}^*_{12}\right\}\right\}) \cdot V\left\{R\left\{\hat{q}_{12}\Theta\hat{l}'_6\Theta\hat{q}^*_{12}\right\}\right\} * V\left\{R\left\{\hat{q}_{12}\Theta\hat{l}'_6\Theta\hat{q}^*_{12}\right\}\right\} = \mathbf{q^o_{in}}, \quad (31)$$

where $\hat{l}'_6 = \hat{q}_3\Theta\hat{l}_6\Theta\hat{q}^*_3$ and $\hat{l}'_5 = \hat{q}_3\Theta\hat{l}_5\Theta\hat{q}^*_3$.

Eq. (31) gives us subproblem 2. The parameters of subproblem 2 are:

$$\mathbf{a} = (V\left\{R\left\{\hat{l}'_6\right\}\right\} \times V\left\{D\left\{\hat{l}'_6\right\}\right\}) + (V\left\{R\left\{\hat{l}'_5\right\}\right\} \times V\left\{D\left\{\hat{l}'_5\right\}\right\}) \cdot V\left\{R\left\{\hat{l}'_6\right\}\right\} * V\left\{R\left\{\hat{l}'_6\right\}\right\}.$$

$\mathbf{l_1}$ is the axis of joint 1, or $\mathbf{d}_1$; $\mathbf{l_2}$ is the axis of joint 2, or $\mathbf{d}_2$; and $\mathbf{b} = \mathbf{q^o_{in}}$.$\theta_1$ and θ_2 can be found by using subproblem 2 (see Appendix).

Step 3: To find the wrist angles, let us consider point $\mathbf{p_i} = \mathbf{p}_6 + \lambda\mathbf{d_6}$ (initial point) on axis $\mathbf{d_6}$; it is not coincident with the $\mathbf{d_4}$ and $\mathbf{d_5}$ axes. Two imaginer axes are used to find $\mathbf{p_e}$ (end point), that is, the position of point $\mathbf{p_i}$ after rotation by θ_4 and θ_5 angles. Point $\mathbf{p_i}$ is the intersection point of the 2 imaginer axes. Let us define the 2 imaginer axes that are on the $\mathbf{d_6}$ axis and intersect at point $\mathbf{p_i}$, given by $\mathbf{d_7} = [0,1,0], \mathbf{d_8} = [0,0,1]$, and $\mathbf{p_{i=}p_{7=}p_8} = [\lambda d_{6x}, ly_0 + ly_1 + \lambda d_{6y}, lz_0 + lz_1 + lz_2 + \lambda d_{6z}]$. The moment vectors are $\mathbf{m_7} = \mathbf{p_i} \times \mathbf{d_7}$ and $\mathbf{m_8} = \mathbf{p_i} \times \mathbf{d_8}$. We can then easily write:

$$(V\left\{R\left\{\hat{q}_{13}\Theta\hat{q}_{45}\Theta\hat{l}_8\Theta\hat{q}^*_{45}\Theta\hat{q}^*_{13}\right\}\right\} \times V\left\{D\left\{\hat{q}_{13}\Theta\hat{q}_{45}\Theta\hat{l}_8\Theta\hat{q}^*_{45}\Theta\hat{q}^*_{13}\right\}\right\}) + (V\left\{R\left\{\hat{q}_{13}\Theta\hat{q}_{45}\Theta\hat{l}_7\Theta\hat{q}^*_{45}\Theta\hat{q}^*_{13}\right\}\right\} \times V\left\{D\left\{\hat{q}_{13}\Theta\hat{q}_{45}\Theta\hat{l}_7\Theta\hat{q}^*_{45}\Theta\hat{q}^*_{13}\right\}\right\}) \cdot V\left\{R\left\{\hat{q}_{13}\Theta\hat{q}_{45}\Theta\hat{l}_8\Theta\hat{q}^*_{45}\Theta\hat{q}^*_{13}\right\}\right\} * V\left\{R\left\{\hat{q}_{13}\Theta\hat{q}_{45}\Theta\hat{l}_8\Theta\hat{q}^*_{45}\Theta\hat{q}^*_{13}\right\}\right\} = \mathbf{q^o_{in}} + \lambda\mathbf{d_6}, \quad (32)$$

which is equal to:

$$(V\left\{R\left\{\hat{q}_{45}\Theta\hat{l}'_8\Theta\hat{q}^*_{45}\right\}\right\} \times V\left\{D\left\{\hat{q}_{45}\Theta\hat{l}'_8\Theta\hat{q}^*_{45}\right\}\right\}) + (V\left\{R\left\{\hat{q}_{45}\Theta\hat{l}'_7\Theta\hat{q}^*_{45}\right\}\right\} \times V\left\{D\left\{\hat{q}_{45}\Theta\hat{l}'_7\Theta\hat{q}^*_{45}\right\}\right\}) \cdot V\left\{R\left\{\hat{q}_{45}\Theta\hat{l}'_8\Theta\hat{q}^*_{45}\right\}\right\} * V\left\{R\left\{\hat{q}_{45}\Theta\hat{l}'_8\Theta\hat{q}^*_{45}\right\}\right\} = \mathbf{q^o_{in}} + \lambda\mathbf{d_6}. \quad (33)$$

Eq. (23) gives us subproblem 2. The parameters of subproblem 2 are:

$$\mathbf{a} = (V\left\{R\left\{\hat{l}'_8\right\}\right\} \times V\left\{D\left\{\hat{l}'_8\right\}\right\}) + (V\left\{R\left\{\hat{l}'_7\right\}\right\} \times V\left\{D\left\{\hat{l}'_7\right\}\right\}) \cdot V\left\{R\left\{\hat{l}'_8\right\}\right\} * V\left\{R\left\{\hat{l}'_8\right\}\right\}.$$

$\mathbf{l_1}$ is the imaginer axis $\mathbf{d_7}$, $\mathbf{l_2}$ is the imaginer axis $\mathbf{d}_8$, and $\mathbf{b} = \mathbf{q^o_{in}} + \lambda\mathbf{d_6}$. θ_4 and θ_5 can be found using subproblem 2 (see Appendix).

Step 4: To find the last joint angle, we need a point that is not on the last joint axis. We call it $\mathbf{p_d} = \mathbf{p}_5 + \lambda\mathbf{d_5}$. Two imaginer axes are used to find $\mathbf{p'_d}$, the position of point $\mathbf{p_d}$ after rotation by θ_6. Point $\mathbf{p_d}$ is the intersection point of the 2 imaginer axes. Let us define the 2 imaginer axes that are on the $\mathbf{d_5}$ axis and intersect at point $\mathbf{p_d}$, given by $\mathbf{d_9} = [0,1,0]$, $\mathbf{d_{10}} = [1,0,0]$, and $\mathbf{p_d} = \mathbf{p_9} = \mathbf{p_{10}} = [\lambda d_{5x}, ly_0 + ly_1 + \lambda d_{5y}, lz_0 + lz_1 + lz_2 + \lambda d_{5z}]$. The moment vectors are $\mathbf{m_9} = \mathbf{p_9} \times \mathbf{d_9}$ and $\mathbf{m_{10}} = \mathbf{p_{10}} \times \mathbf{d_{10}}$. We can then easily write Eq. (34).

$$(V\left\{R\left\{\hat{q}_{16}\Theta\hat{l}_{10}\Theta\hat{q}^*_{16}\right\}\right\} \times V\left\{D\left\{\hat{q}_{16}\Theta\hat{l}_{10}\Theta\hat{q}^*_{16}\right\}\right\})+ (V\left\{R\left\{\hat{q}_{16}\Theta\hat{l}_9\Theta\hat{q}^*_{16}\right\}\right\} \times V\left\{D\left\{\hat{q}_{16}\Theta\hat{l}_9\Theta\hat{q}^*_{16}\right\}\right\}) \cdot V\left\{R\left\{\hat{q}_{16}\Theta\hat{l}_{10}\Theta\hat{q}^*_{16}\right\}\right\} * V\left\{R\left\{\hat{q}_{16}\Theta\hat{l}_{10}\Theta\hat{q}^*_{16}\right\}\right\} = \mathbf{q^o_{in}} + \lambda\mathbf{d_5} \quad (34)$$

(34) This is equal to:

$$\begin{aligned}&(V\left\{R\left\{\hat{q}_6\Theta\hat{l}'_{10}\Theta\hat{q}_6^*\right\}\right\}\times V\left\{D\left\{\hat{q}_6\Theta\hat{l}'_{10}\Theta\hat{q}_6^*\right\}\right\})+\\&(V\left\{R\left\{\hat{q}_6\Theta\hat{l}'_9\Theta\hat{q}_6^*\right\}\right\}\times V\left\{D\left\{\hat{q}_6\Theta\hat{l}'_9\Theta\hat{q}_6^*\right\}\right\})\cdot V\left\{R\left\{\hat{q}_6\Theta\hat{l}'_{10}\Theta\hat{q}_6^*\right\}\right\}*V\left\{R\left\{\hat{q}_6\Theta\hat{l}'_{10}\Theta\hat{q}_6^*\right\}\right\}=\mathbf{q_{in}^o}+\lambda\mathbf{d_5}\end{aligned} \tag{35}$$

Eq. (35) gives us subproblem 1. The parameters of subproblem 1 are:

$$\mathbf{a}=(V\left\{R\left\{\hat{l}'_{10}\right\}\right\}\times V\left\{D\left\{\hat{l}'_{10}\right\}\right\})+(V\left\{R\left\{\hat{l}'_9\right\}\right\}\times V\left\{D\left\{\hat{l}'_9\right\}\right\})\cdot V\left\{R\left\{\hat{l}'_{10}\right\}\right\}*V\left\{R\left\{\hat{l}'_{10}\right\}\right\}.$$

$\mathbf{l}$ is the axis of joint 6, namely $\mathbf{d_6}$, and $\mathbf{b}=\mathbf{q_{in}^o}+\lambda\mathbf{d_5}$. θ_6 can be found by using subproblem 1 (see Appendix).

6. Cooperative working of serial robot arms

Figure 3 illustrates a possible arrangement of 2 robot arms. In the cooperative working of 2 robot arms, a closed-chain mechanism comes into existence from 2 open-chain mechanisms. As shown in Figure 3, there are 12 DOF in the closed-chain mechanism. Since the closed-chain mechanism is redundant, there are infinite solutions (or singularity) in the inverse kinematic problem. On this account, the kinematic problem of the closed-chain mechanisms is more difficult than that of the open-chain mechanisms. However, the kinematic problem of the closed-chain robot arms can be solved by reducing the full kinematic problem to the appropriate subopen-chain kinematic problem. If the position and orientation of both robot arms can be determined appropriately, the closed-chain robot kinematic problem can be reduced to the serial robot-arm kinematic problem. It can then be solved by using the proposed method. Two different methods are used in Section 7. In the first, a path is determined for an object (a ball). The appropriate positions and orientations of the 2 robot arms are then determined for each step of the cooperative work. In the second, a path is determined for the first robot arm. The second robot arm's path is then determined by using the position and orientation of the first robot arm. The point symmetry method is used to obtain the position and orientation of the second robot arm [18].

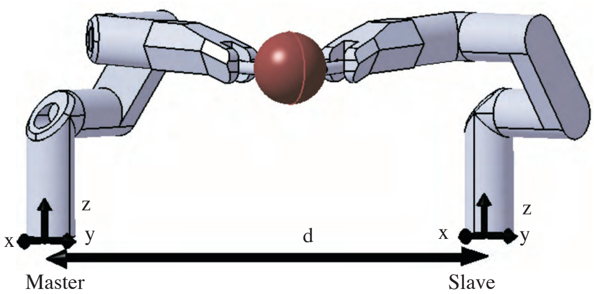


Figure 3. Configuration of the cooperative working of a dual-arm robot manipulator.

7. Simulation results

Stäubli RX160 industrial robot arms were used for the simulation studies. Stäubli RX160 robot-arm series features an articulated arm with 6 DOF for high flexibility. It covers a wide-ranging area in industrial robot applications. The kinematic simulation studies were done using MATLAB and the animation applications were done using the Virtual Reality Toolbox of MATLAB. Stäubli RX160 IGES files, which can be freely obtained from the Stäubli web page, were also used for the animation application. Two different animation applications, which are shown in Figures 4 and 5, were performed.

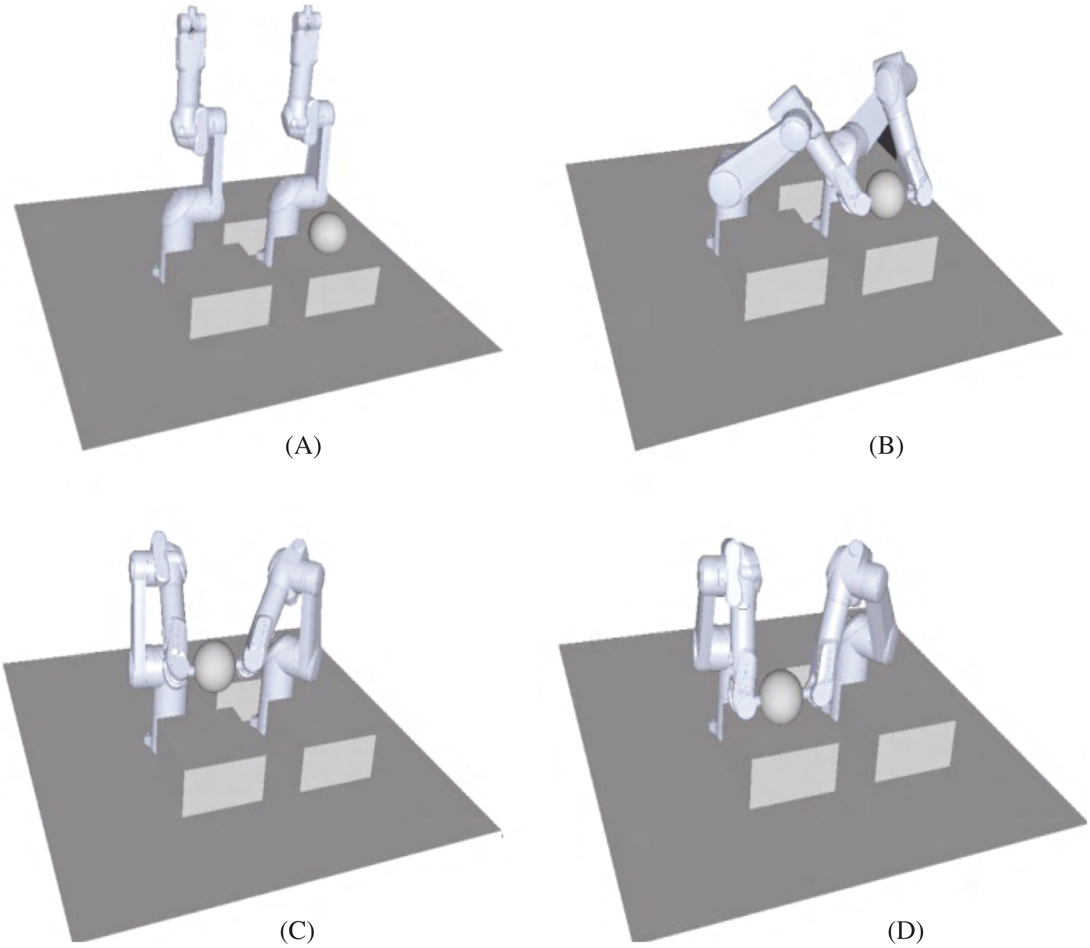
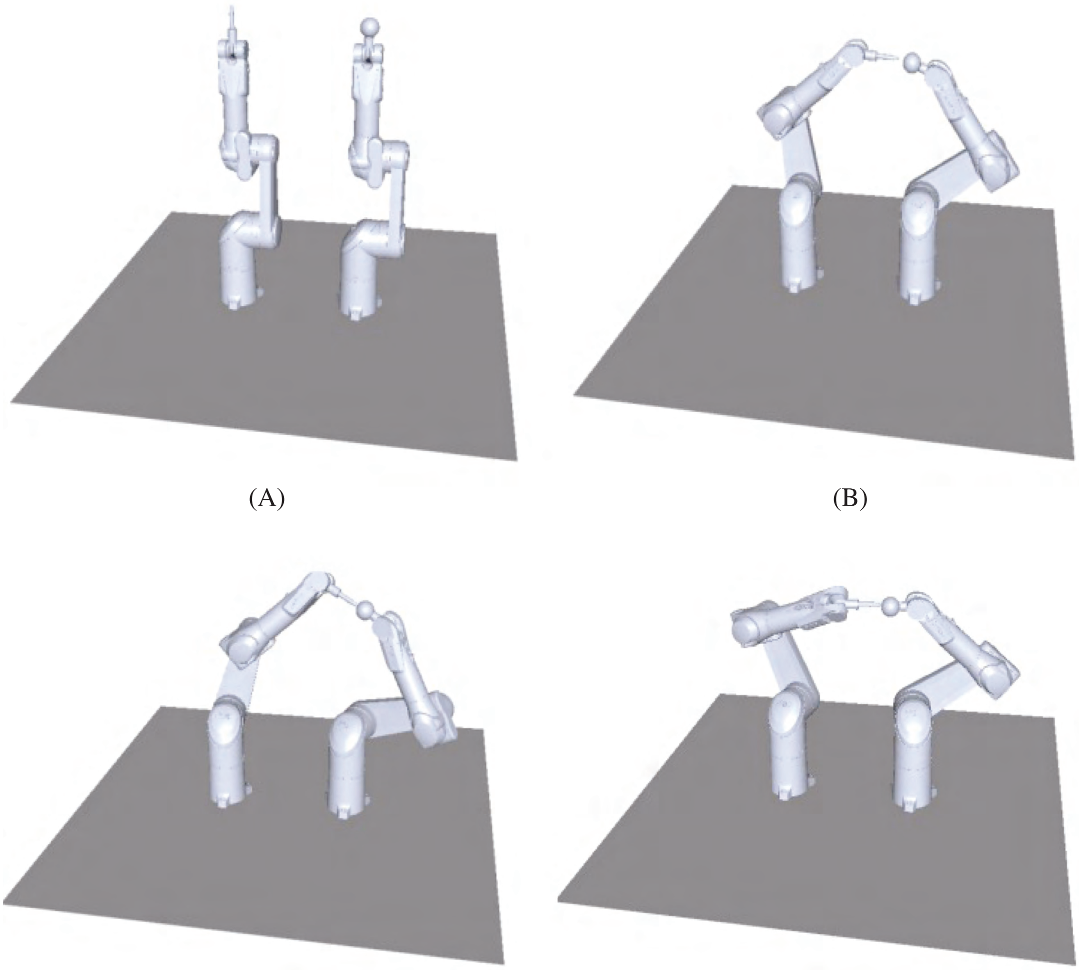


Figure 4. Cooperative working (ball-carrying experiments).

In the first case, the 2 robot arms worked together and carried a ball from its initial position to the desired target position, as shown consecutively in Figure 4. To implement this case, first a path was determined for the ball. The inverse kinematic problem of the serial robot arm was then solved by using this path for both of the robot arms. The orientations of the robot arms were chosen adversely to each other.

The second case involves work in the master-slave mode. In this case, the first robot arm, which has a ball at the end effector, moved by a given path, and the second robot arm followed the tip point of the first robot arm, as shown consecutively in Figure 5. To implement this case, first a path was determined for the first robot arm. The orientation and position information of the first robot arm was then sent to the second robot arm and the inverse kinematic problem of the second robot arm was solved using the orientation and position information. The first robot arm, which sends its position and orientation information, works as a master, and the second robot arm, which follows the tip point of the first robot arm, works as a slave.

Dual operators are the best way to describe screw motion, and the dual quaternion is the most compact and efficient dual operator to express screw displacement. A dual quaternion requires 8 memory locations for the definition of the rigid body motion, while a homogeneous transformation matrix requires 16 memory locations. The storage requirement affects the computational time because the cost of fetching an operand from the memory exceeds the cost of performing a basic arithmetic operation [36], and it is very important for the real-time implementation. Dual quaternion-based rigid transformation requires less computational load. The performance analysis of the transformation operators can be seen in Table 1.



(A) (B) (C) (D)

Figure 5. Cooperative working (working in master-slave mode).

Table 1. Performance comparison of rigid transformation operations.

Method	Storage	Multiplication	Add/subtract	Total
Hom. trans. matrix*	16	64	48	112
Dual quaternion	8	48	40	88

*Homogeneous transformation matrix

In order to obtain the rigid body transformation operator for an n-link serial robot manipulator:

- $64(n-1)$ multiplications and $48(n-1)$ additions must be done if the transformation operator is a homogeneous transformation matrix.
- $48(n-1)$ multiplications and $40(n-1)$ additions must be done if the transformation operator is a dual quaternion.

Figure 6 shows that as the degrees of freedom increase, the method that uses the dual quaternion as a rigid body transformation operator becomes more advantageous.

The computational efficiency of the dual quaternion-based and homogeneous transformation matrix-based solutions are given in Figures 7 and 8. The computation time was evaluated using MATLAB's tic-toc commands.

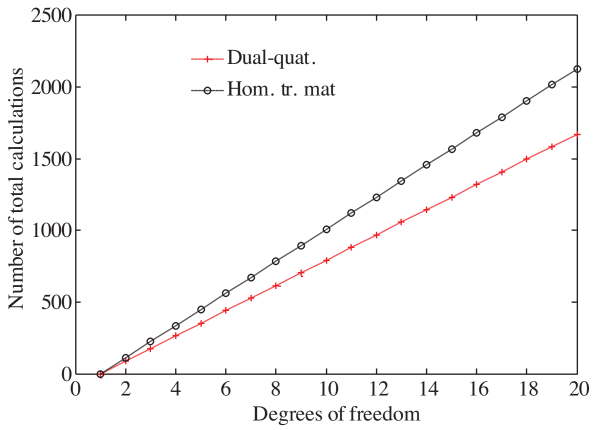


Figure 6. Performance comparison of the rigid body transformation chaining operations.

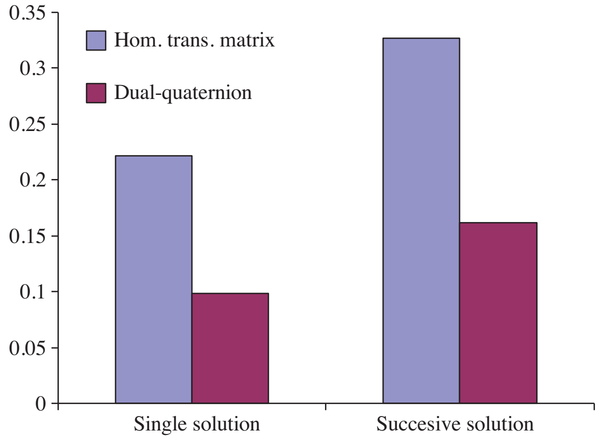


Figure 7. Simulation times of the forward kinematic solutions (s).

As can be seen from Figures 6, 7, and 8, the method that uses a dual quaternion as a screw motion operator is more computationally efficient than the homogenous transformation matrix-based method since the dual quaternion-based method describes screw motion using fewer parameters and has less computational load. The running environment is given in Table 2.

Table 2. Running environment.

CPU	CPU memory	Operating system	Simulation software
Intel Core 2 Duo 2.2 GHz	2 GB	Windows XP	MATLAB 7

8. Experimental results

In the experimental study, we used Stäubli RX 160 and RX 160L serial robot arms and a CS8 controller, which includes a low-level programming package to control the robot under a VxWorks® real-time operating system. The given kinematic algorithm was applied to the Stäubli RX 160 robots using the Stäubli Robotics LLI Programming Interface S6.4, which is a C programming interface for low-level robot control. LLI stands for low-level interface; it is a software package that includes the minimum functions required to construct a robot control mechanism via C/C++ API [37]. The algorithm was written in C++ language using the library functions of the LLI software package and embedded in the controller.

In order to verify the simulation results, an experiment was performed using the Stäubli RX 160 and RX 160L robot arms shown in Figure 9.

In the experimental study, a cubic trajectory that passed through the singular configurations of the robot arms was determined for the first robot arm, the RX 160L (master). The trajectory of the second robot arm, the RX 160 (slave), was determined using the point symmetry method explained in section 7. The trajectory tracking results for the position and orientation of dual-arm cooperative work are shown in Figures 10 and 11. Figure 10 shows the position of the end effectors of both robot arms on the x, y, and z coordinates. Figure 11 shows the orientation angle of the master robot. Ellipses in Figures 10 and 11 show that the robot arms pass through singular configurations at those trajectories.

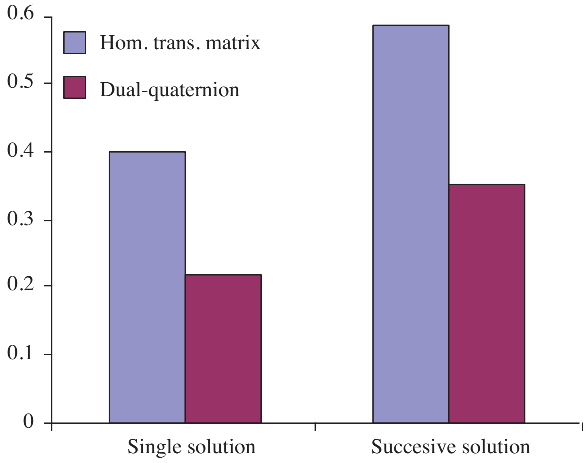


Figure 8. Simulation times of the inverse kinematic solutions (s).

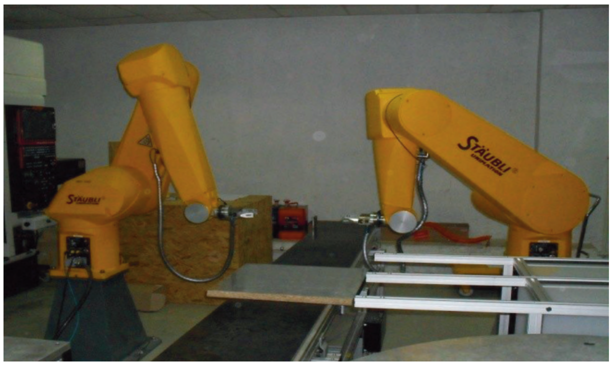


Figure 9. Stäubli RX 160 and RX 160L serial robot arms.

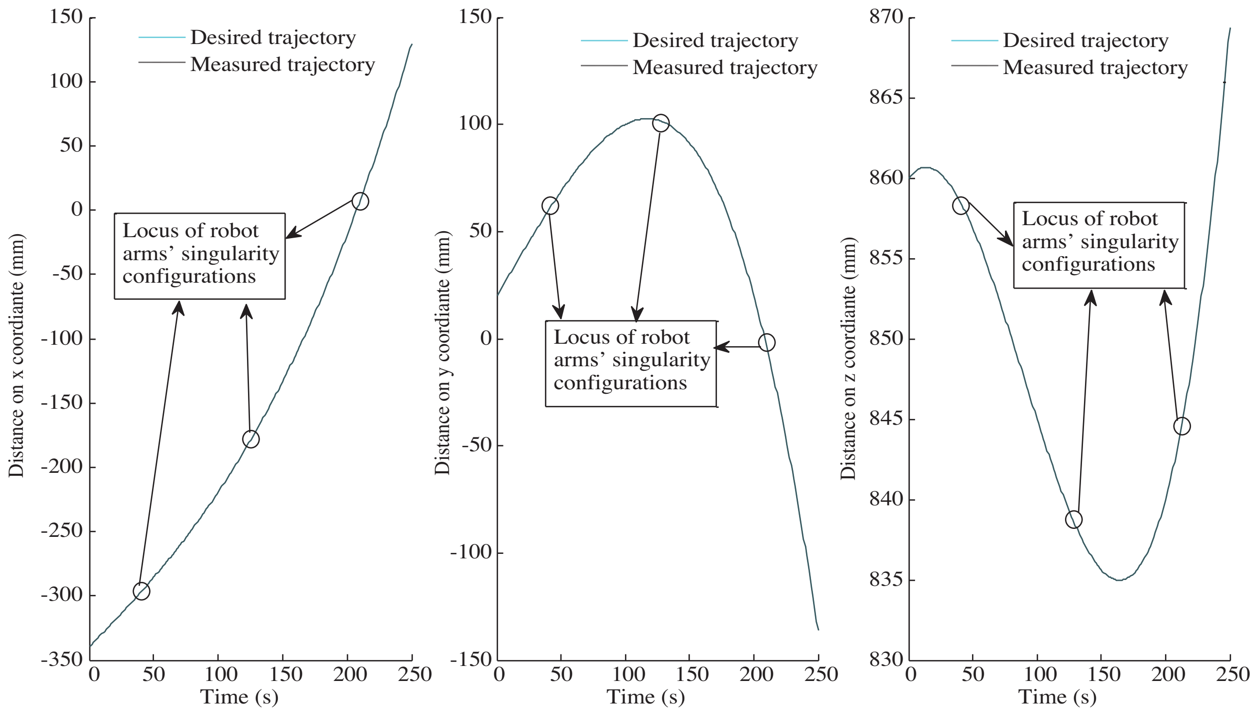


Figure 10. Trajectory tracking for the x, y, and z coordinates.

The trajectory tracking errors are given in Figure 12. As can be seen from Figures 10, 11, and 12, a satisfactory singularity-free trajectory tracking application was implemented.

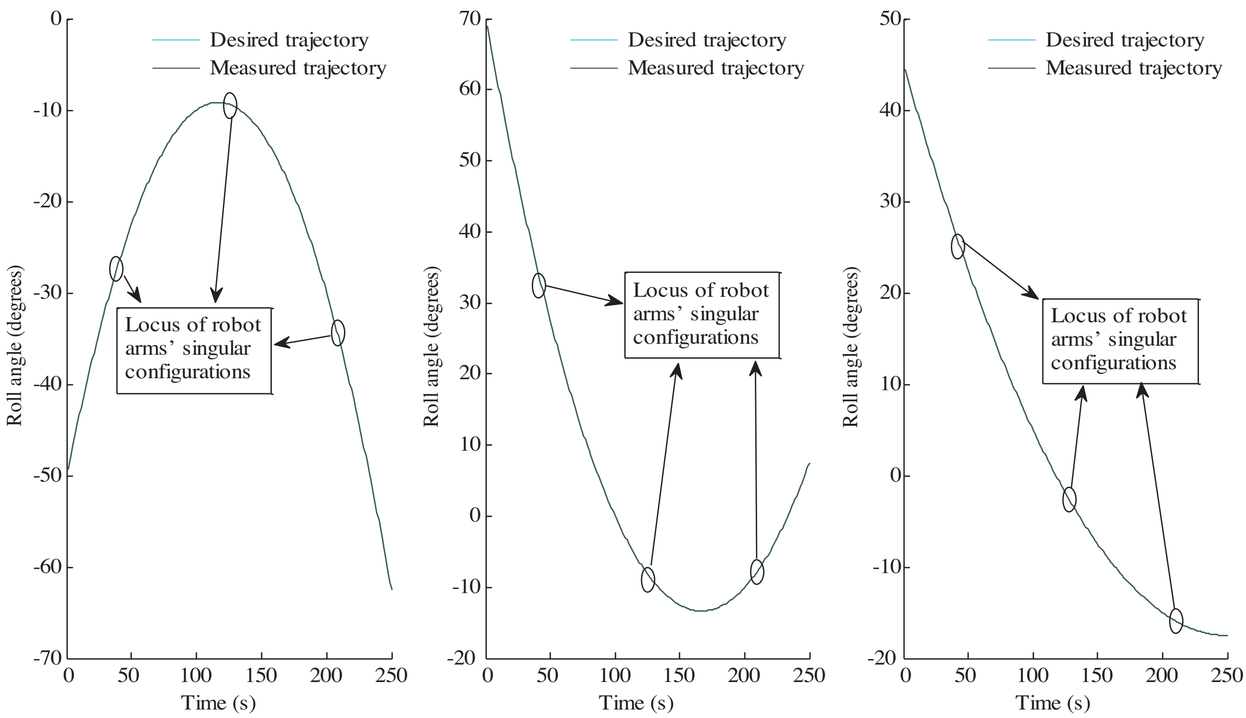


Figure 11. Trajectory tracking for the Roll, Pitch, and Yaw orientation angles of the master robot.

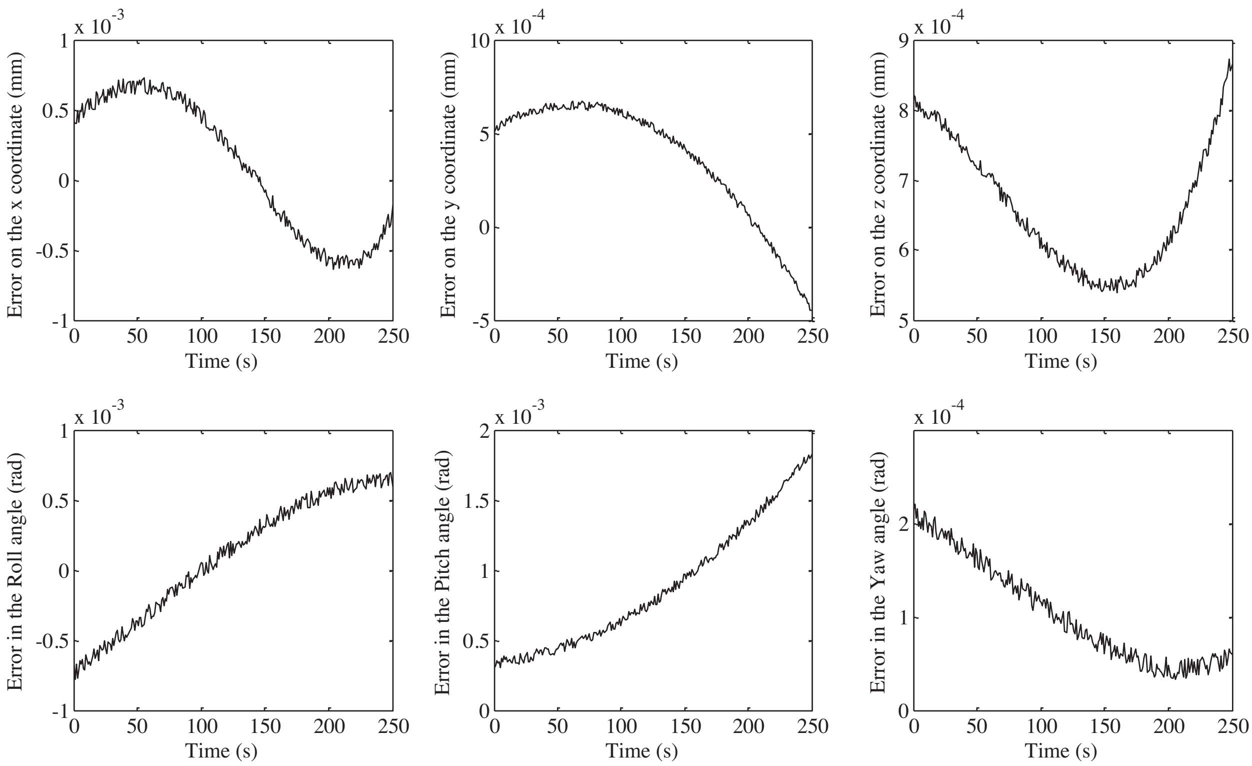
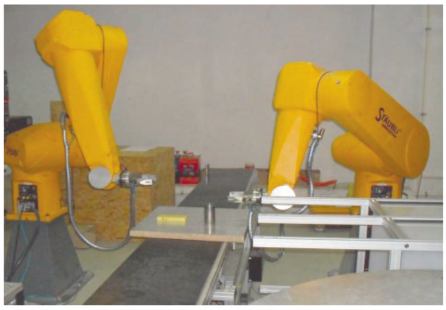


Figure 12. Trajectory tracking errors for position and orientation.

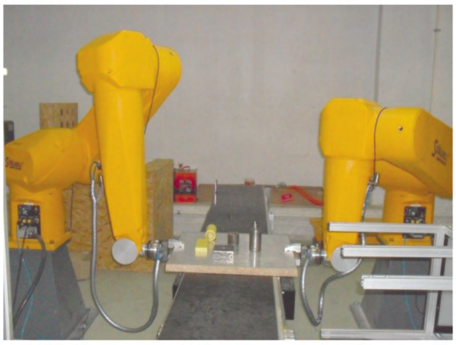
(A)

(IB)

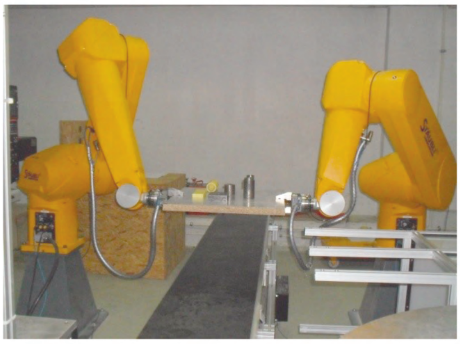
(C)



(D)



(E)



(F)

(G)

(H)

Figure 13. An industrial robot application (cooperative and independent working of dual-arm robot system).

An industrial robotics application was also done for the experimental study. This experimental study can be seen consecutively in Figure 13. In this study, both the cooperative and independent workings of the robot arms were implemented. First, 2 robots were worked independently. They took the products from the computer numerical control machines and gathered them on the tablet. Both of the robot arms then cooperatively carried the heavy tablet from the initial position to the target position. In the independent working of the robot arms, position kinematic control satisfies the desired task; however, velocity kinematic control is also needed for the synchronization of the robot arms in the cooperative work.

Conclusion

In this paper, a singularity-free inverse kinematic solution method of serial robot-arm manipulators was implemented into the cooperative working of the industrial robot-arm manipulators. This solution method is based on screw theory and quaternion algebra. Screw theory is an effective way to establish a global description of the rigid body and avoids singularities due to the use of local coordinates. Compared with other methods, screw theory methods establish just 2 coordinates, and screw theory's geometrical meaning is obvious. Screw theory with the dual quaternion method is the most compact and efficient way to express screw displacement. As the complexity and the degrees of freedom of the system increase, the methods based on screw theory and quaternion algebra give better results. This is because, as the degrees of the freedom of the systems increase, they have many more singularity points, more computational loads, and more complex geometrical structures. On these accounts, the wider use of screw theory-based methods and quaternion algebra in robot kinematic studies has to be considered by the robotics community.

In future work, collision-free path planning of multiarm robot systems should be studied by using this new formulation method. In addition, velocity and dynamic analysis based on screw theory with quaternion algebra should be studied.

Appendix

A.1 Plücker coordinates

Any line can be completely defined by using position ($\mathbf{p}$) and direction ($\mathbf{d}$) vectors. It can also be represented by using Plücker coordinates given by $L_p(\mathbf{m}, \mathbf{d})$, where $\mathbf{m} = \mathbf{p} \times \mathbf{d}$ is the moment vector of $\mathbf{d}$ about the chosen reference origin [38].

Note that $\mathbf{m}$ is independent of which point $\mathbf{p}$ on the line is chosen: $\mathbf{m} = \mathbf{p} \times \mathbf{d} = (\mathbf{p} + t\mathbf{d}) \times \mathbf{d}$.

The Plücker coordinate representation is not minimal since it uses 6 parameters for the line representation. The main advantage of Plücker coordinate representation is that it is homogeneous. $L_p(\mathbf{m}, \mathbf{d})$ represents the same line as $L_p(k\mathbf{m}, k\mathbf{d})$, where $k \in \Re$.

A.2 Dual numbers

In analogy with a complex number, a dual number can be defined by $\hat{u} = u + \varepsilon u^o$, where u and u^o are real numbers and $\varepsilon^2 = 0$ [39]. Dual numbers can be used to express Plücker coordinates given by $\hat{u} = \mathbf{d} + \varepsilon \mathbf{m}$, where $\mathbf{d}$ and $\mathbf{m} = \mathbf{p} \times \mathbf{d}$ are the orientation and moment vectors of the line, respectively.

A.3 Intersection of 2 orthogonal unit line vectors

The intersection of 2 orthogonal lines is shown in Figure A.1.

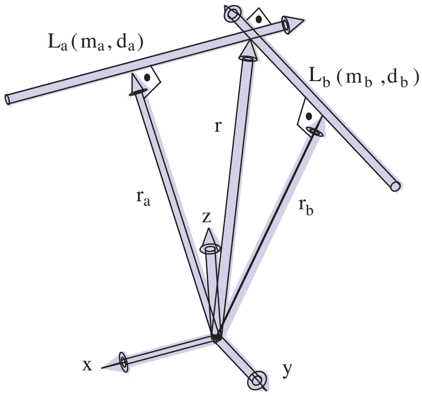


Figure A.1. Intersection of 2 lines.

The intersection point of 2 lines can be found given by [40]:

$$\mathbf{r} = \mathbf{d_b} \times \mathbf{m_b} + (\mathbf{d_a} \times \mathbf{m_a} \cdot \mathbf{d_b})\mathbf{d_b} \text{ or } \mathbf{r} = \mathbf{d_a} \times \mathbf{m_a} + (\mathbf{d_b} \times \mathbf{m_b} \cdot \mathbf{d_a})\mathbf{d_a} \tag{A.1}$$

A.4 Paden-Kahan subproblems using quaternion algebra

A.4.1 Subproblem 1: Rotation about a single axis

Point **a** rotates about the axis of **l** until point **a** is coincident with point **b**. This rotation is shown in Figure A.2.

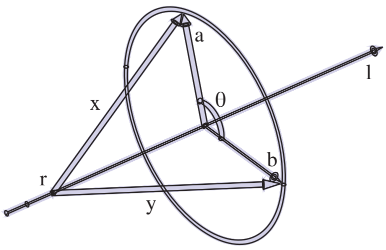


Figure A.2. Rotate **a** about the axis of **l** until it is coincident with **b**.

Let **r** be a point on the axis of **l**, and let $\mathbf{x} = \mathbf{a} - \mathbf{r}$ and $\mathbf{y} = \mathbf{b} - \mathbf{r}$ be 2 vectors. The rotation angle θ about the axis of **l** can be found as follows:

$$\theta = \arctan 2\left(S\left\{l \otimes x' \otimes y'\right\}, S\left\{x' \otimes y'\right\}\right), \tag{A.2}$$

where $x' = x - S\{l \otimes x\}\, l$ and $y' = q \otimes x \otimes q^* - S\{l \otimes q \otimes x \otimes q^*\}\, l$.

Here, $x = [0, \mathbf{x}]$, $y = [0, \mathbf{y}]$, and $l = [0, \mathbf{l}]$ is the pure quaternion form of vectors $\mathbf{x}, \mathbf{y}$, and $\mathbf{l}$, respectively, and $q = \left[\cos\left(\frac{\theta}{2}\right), \sin\left(\frac{\theta}{2}\right)\mathbf{l}\right]$.

A.4.2 Subproblem 2: Rotation about 2 subsequent axes

First, point **a** rotates about the axis of $\mathbf{l_1}$ by θ_1 and then about the axis of $\mathbf{l_2}$ by θ_2; hence, the final location of **a** is coincident with point **b**. This rotation is shown in Figure A.3.

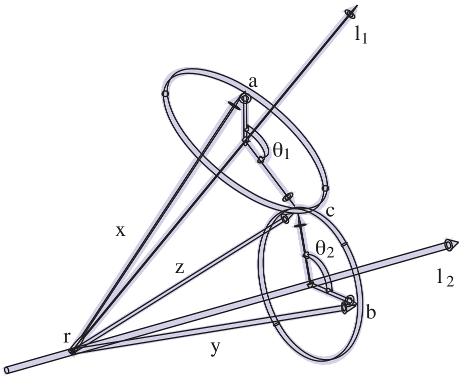


Figure A.3. Rotate **a** about the axis of $\mathbf{l_1}$, followed by a rotation around the axis of $\mathbf{l_2}$, until it is coincident with point **b**.

Let **r** be the intersection point of the 2 axes, and let $\mathbf{x} = \mathbf{a} - \mathbf{r}$ and $\mathbf{y} = \mathbf{b} - \mathbf{r}$ be 2 vectors. Let **c** be the intersection point of the rotations that is shown in Figure A.3 and let $\mathbf{z} = \mathbf{c} - \mathbf{r}$ be the vector that is defined between points **c** and **r**, with $z = [0, \mathbf{z}]$ the pure quaternion form of vector **z**. We can also define 2 rotations given by $q_1 \otimes x \otimes q_1^* = z = q_2 \otimes y \otimes q_2^*$, where $q_1 = [\cos\left(\frac{\theta_1}{2}\right), \sin\left(\frac{\theta_1}{2}\right)\mathbf{l}_1]$, $q_2 = [\cos\left(-\frac{\theta_2}{2}\right), \sin\left(-\frac{\theta_2}{2}\right)\mathbf{l}_2]$, and $x{=}[0,\mathbf{x}]$ and $y{=}[0,\mathbf{y}]$. Since $\mathbf{l_1}$, $\mathbf{l_2}$, and $\mathbf{l_1} \times \mathbf{l_2}$ are linearly independent, we can write $z = \alpha l_1 + \beta l_2 + \gamma\,[0, V\{l_1 \otimes l_2\}]$, where:

$\alpha = \frac{S\{l_1 \otimes l_2\} S\{l_2 \otimes x\} - S\{l_1 \otimes y\}}{(S\{l_1 \otimes l_2\})^2 - 1}, \beta = \frac{S\{l_1 \otimes l_2\} S\{l_1 \otimes y\} - S\{l_2 \otimes x\}}{(S\{l_1 \otimes l_2\})^2 - 1}, \gamma^2 = \frac{||x||^2 - \alpha^2 - \beta^2 - 2\alpha\beta S\{l_1 \otimes l_2\}}{||V\{l_1 \otimes l_2\}||^2}$. Thus, subproblem 2 is reduced to subproblem 1. The angles of rotation axes θ_1 and θ_2 can be solved using subproblem 1.

$$q_1 \otimes x \otimes q_1^* = z \text{ and } q_2 \otimes y \otimes q_2^* = z \tag{A.3}$$

A.4.3. Subproblem 3: Rotation to a given distance

Point **a** rotates about the axis of **l** until the point is at distance δ from **b**, as shown in Figure A.4.

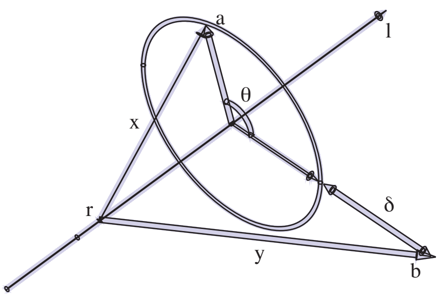


Figure A.4. Rotate **a** about the axis of **l** until it is at distance δ from **b**.

Let $\mathbf{r}$ be a point on the axis of $\mathbf{l}$ and let $x = [0, \mathbf{a} - \mathbf{r}]$ and $y = [0, \mathbf{b} - \mathbf{r}]$ be pure quaternion forms of vectors $\mathbf{x}$ and $\mathbf{y}$, respectively. Rotation angle θ about the axis of $\mathbf{l}$ can be found as follows:

$$\theta = \theta_0 \pm \cos^{-1}\left(\frac{||x'|| + ||y'|| - \delta'^2}{2||x'||||y'||}\right), \tag{A.4}$$

where $\theta_0 = \arctan 2\left(S\left\{l \otimes x' \otimes y'\right\}, S\left\{x' \otimes y'\right\}\right)$,

$$x' = [0, \mathbf{x}'] = x - S\left\{l \otimes x\right\} l, \quad y' = [0, \mathbf{y}'] = q \otimes x \otimes q^* - S\left\{l \otimes q \otimes x \otimes q^*\right\} l, \quad \delta'^2 = \delta^2 - \left|S\left\{l \otimes (a - b)\right\}\right|^2,$$

and $q = \left[\cos\left(\frac{\theta}{2}\right), \sin\left(\frac{\theta}{2}\right)\mathbf{l}\right]$.

References

[1] R. Buckingham, "Multi-arm robots", Industrial Robot, Vol. 23, pp. 16-20, 1996.

[2] A. Edsinger, C.C. Kemp, "Two arms are better than one: a behavior based control system for assistive bimanual manipulation", Lecture Notes in Control and Information Sciences, Vol. 370, pp. 345-355, 2008.

[3] C. Park, K. Park, "Design and kinematics analysis of dual arm robot manipulator for precision assembly", IEEE International Conference on Industrial Informatics, pp. 430-435, 2008.

[4] P. Chiacchio, S. Chiaverini, B. Siciliano, "Direct and inverse kinematics for coordinated motion tasks of a two-manipulator system", Journal of Dynamic Systems, Measurement, and Control, Vol. 118, pp. 691-697, 1996.

[5] R. Konietschke, T. Ortmaier, U. Hagn, G. Hirzinger, S. Frumento, "Kinematic design optimization of an actuated carrier for the DLR multi-arm surgical system", IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4381-4387, 2006.

[6] R. Zollner, T. Asfour, R. Dillmann, "Programming by demonstration: dual-arm manipulation tasks for humanoid robots", IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 1, pp. 479-484, 2004.

[7] U. Özbay, E. Zergeroğlu, İ. Kandemir, "A model based nonlinear adaptive controller for the passive bilateral telerobotic system", Turkish Journal of Electrical Engineering and Computer Sciences, Vol. 18, pp. 781-798, 2010

[8] D.M. Zivanovic, K.M. Vukobratovic, "Multi-arm cooperating robots, dynamics and control series", Intelligent Systems, Control and Automation: Science and Engineering, Vol. 30, pp. 1-297, 2006.

[9] A. Jazidie, T. Tsuji, M. Nagamachi, K. Ito, "Dynamic simulation of multi-arm robots using Appel's method", IFToMM-jc International Symposium on Theory of Machines and Mechanisms, pp. 199-204, 1992.

[10] J.T. Wen, L.S. Wilfinger, "Kinematic manipulability of general constrained rigid multibody systems", IEEE Transactions on Robotics and Automation, Vol. 15, pp. 1020-1025, 1999.

[11] W.S. Owen, E.A. Croft, B. Benhabib, "A multi-arm robotic system for optimal sculpting", Robotics and Computer-Integrated Manufacturing, Vol. 24, pp. 92-104, 2008.

[12] V. Kumar, J.F. Gardner, "Kinematics of redundantly actuated closed chains", IEEE Transactions on Robotics and Automation, Vol. 6, pp. 269-274, 1990.

[13] Z. Hu, Z. Fu, H. Fang, "Study of singularity robust inverse of Jacobian matrix for manipulator", International Conference on Machine Learning and Cybernetics, pp. 40610, 2002

[14] Y. Nakamura, H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control", Journal of Dynamic Systems, Measurement and Control, Vol. 108, pp. 163-171, 1986

[15] C.W. Wampler, "Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods", IEEE Transactions on Systems, Man and Cybernetics, Vol. 16, pp. 93101, 1986

[16] A. Balestrino, G. De Maria, L. Sciavicco, "Robust control of robotic manipulators", International Proceedings of the 9th IFAC World Congress Vol. 6, pp. 8085, 1984

[17] A.T. Hasan, N. Ismail, A.M.S Hamouda, I. Aris, M.H. Marhaban, H.M.A.A. Al-Assadi, "Artificial neural network-based kinematics Jacobian solution for serial manipulator passing through singular configurations", Advances in Engineering Software, Vol. 41, pp. 359367, 2010

[18] A. Hemami, "Kinematics of two-arm robots", IEEE Journal of Robotics and Automation, Vol. 2, pp. 225-228, 1986.

[19] Y.F. Zheng, "Kinematics and dynamics of two industrial robots in assembly", International Conference on Robotics and Automation, Vol. 3, pp. 1360-1365, 1989.

[20] E. Sarıyıldız, A New Approach to Inverse Kinematic Solutions of Serial Robot Arms Based on Quaternions in the Screw Theory Framework, Master Thesis, İstanbul Technical University, İstanbul, 2009.

[21] Z. Huang, Y.L.Yao, "Extension of usable workspace of rotational axes in robot planning", Robotica, Vol. 17, pp. 293-301, 1999.

[22] J. Funda, R.P. Paul, "A computational analysis of screw transformations in robotics", IEEE Transactions on Robotics and Automation, Vol. 6, pp. 348-356, 1990.

[23] J. Funda, R.H. Taylor, R.P. Paul, "On homogeneous transforms, quaternions, and computational efficiency", IEEE Transactions on Robotics and Automation, Vol. 6, pp. 382-388, 1990.

[24] W.R. Hamilton, Elements of Quaternions, Vols. I and II, New York, Chelsea, 1869.

[25] R. Mukundan, "Quaternions: from classical mechanics to computer graphics and beyond", Proceedings of the 7th Asian Technology Conference in Mathematics, 2002.

[26] J.C. Hart, G.K. Francis, L.H. Kauffman, "Visualizing quaternion rotation", ACM Transactions on Graphics, Vol. 13, pp. 256-276, 1994.

[27] Q. Tan, J.G. Balchen, "General quaternion transformation representation for robotic application", International Conference on Systems, Man, and Cybernetics, Vol. 3, pp. 319-324, 1993.

[28] B. Akyar, "Dual quaternions in spatial kinematics in an algebraic sense", Turkish Journal of Mathematics, Vol. 32, pp. 373-391, 2008.

[29] D. Han, Q. Wei, Z. Li, "Kinematic control of free rigid bodies using dual quaternions", International Journal of Automation and Computing, Vol. 5, pp. 319-324, 2008.

[30] K. Daniilidis, "Hand-eye calibration using dual quaternions", The International Journal of Robotics Research, Vol. 18, pp. 286-298, 1999.

[31] R.S. Ball, The Theory of Screws, Cambridge, Cambridge University Press, 1900.

[32] J.M. Selig, Geometric Fundamentals of Robotics, 2nd ed., New York, Springer, 2005.

[33] B. Paden, Kinematics and Control Robot Manipulators, PhD Thesis, Department of Electrical Engineering and Computer Science, University of California, Berkeley, 1986.

[34] M. Murray, Z. Li, S.S. Sastry, A Mathematical Introduction to Robotic Manipulation, Boca Raton, Florida, CRC Press, 1994.

[35] Y. Tan, A. Xiao, "Extension of the second Paden-Kahan sub-problem and its' application in the inverse kinematics of a manipulator", IEEE Conference on Robotics, Automation and Mechatronics, pp. 379-384, 2008.

[36] N.A. Aspragathos, J.K. Dimitros, "A comparative study of three methods for robot kinematics", IEEE Transactions on Systems, Man, and Cybernetics B, Vol. 28, pp. 135-145, 1998.

[37] F. Pertin, J.M.B. des Tuves, "Real time robot controller abstraction layer", Proceedings of International Symposium on Robots, 2004.

[38] H. Bruyninckx, J.D. Schutter, Introduction to Intelligent Robotics, Technical Report, Leuven, Katholieke Universiteit de Leuven, 2001.

[39] Y.L. Gu, J.Y.S. Luh, "Dual-number transformation and its application to robotics", IEEE Journal of Robotics and Automation, Vol. 3, pp. 615-623, 1987.

[40] J.H Kim, V.R. Kumar, "Kinematics of robot manipulators via line transformations", Journal of Robotic Systems, Vol. 7, pp. 649-674, 1990.