Finding the Position and Orientation of a Sensor on a Robot Manipulator Using Quaternions

Jack C. K. Chou and M. Kamel

The International Journal of Robotics Research 1991 10: 240
DOI: 10.1177/027836499101000305

Jack C. K. Chou
Computer Aided Design and Systems Analysis Laboratory
Erik Jonsson School of Engineering and Computer Science
University of Texas at Dallas
Richardson, Texas 75083-0688

M. Kamel
Pattern Analysis and Machine Intelligence Laboratory
Department of Systems Design Engineering
Institute for Computer Research
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1

Finding the Position and Orientation of a Sensor on a Robot Manipulator Using Quaternions

Abstract

The problem of finding the relative position and orientation between the reference frames of a link-mounted sensor and the link has been formulated as a kinematic equation of the form $\mathbf{H}_l\mathbf{H}_x = \mathbf{H}_x\mathbf{H}_c$ *in terms of homogeneous transformation matrices by Shiu and Ahmad (1987). In this article, normalized quaternions (Euler parameters) are used to transform the kinematic equation into two simple and structured linear systems with rank-deficient coefficient matrices. Closed-form solutions to these systems are derived using the generalized inverse method with singular-value decomposition analysis. To obtain a unique solution, two distinct robot movements are required. This leads to an overdetermined system of equations. A criterion for selecting the independent set of equations is developed. A set of closed-form formulae for the solution of these equations are derived. The selection criterion and the solution formulae can be easily incorporated in application programs that require the calculation of the relative position and orientation of the sensor.*

1. Introduction

To locate an object in the base coordinate frame of a robot manipulator, the relative position and orientation between the coordinate frames of the link on which a sensor (such as a camera that is able to find the three-dimensional orientation and position of an object in space) is installed and the coordinate frame of the sensor have to be known by the controller. A direct measurement of the relative position for an accurate result is not feasible, because the measurement path may be blocked, and the origins of the coordinate systems may not be reachable. It is even more difficult to measure the relative orientation directly, as two coordinate systems may be oriented arbitrarily, and the directions of the Cartesian coordinate axes may not be identified physically (Fig. 1).

Shiu and Ahmad (1987) proposed a mathematical approach to solve the problem by formulating it as an equation of the form $H_lH_x = H_xH_c$. It is important to be specific about the location of the sensor with respect to the link on which the sensor is mounted. For a six-link robot, the wrist[1] subassembly unit usually consists of the last three links (links 4, 5, and 6) and possesses three rotational degrees of freedom (Fu et al. 1987). For instance, if a camera is mounted on link 6 (robot hand), as shown in Figure 1, the position and orientation of the camera frame relative to the coordinate frames l_4 and l_5 (attached to the links 4 and 5, respectively) are variant. That is, the transformation H_x is not constant; in this case, the underlying problem will never be solved. The problem can be formulated by defining the transformation matrices as follows:

- H_l: describes the position and orientation of the coordinate system l_i (attached to link i) relative to l_i after an arbitrary movement of the arm (see fig 2).
- H_c: describes the position and orientation of the coordinate system c of the camera (mounted on link i) relative to the system c after a movement (see figure 2).

The International Journal of Robotics Research,
Vol. 10, No. 3, June 1991,

1. Note that some robot arms cannot be decomposed into an arm and a wrist.

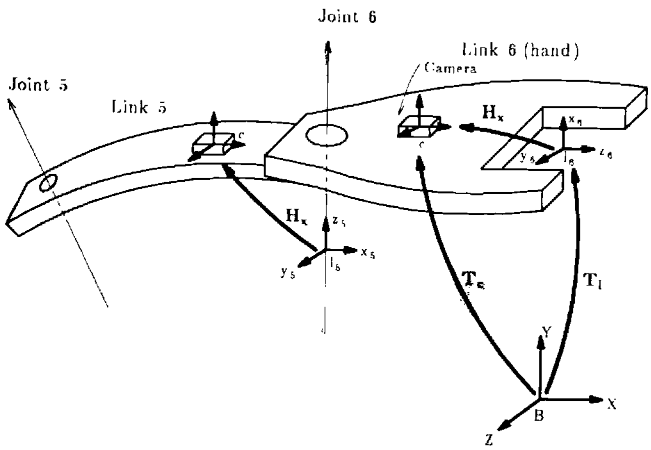

Fig. 1. Configuration of a camera mounted on a robot arm.

H_x: describes the position and orientation of the camera frame c relative to the frame l_i when the camera is mounted on link i.

In this manner, the algorithm presented later is a general solver for H_x where the camera can be mounted on any link of the robot arm.

The general solution of an equation of the form $AX = XB$, where A and B are square matrices and X is a rectangular matrix, has been discussed by Gantmacher (1959). However, the direct application of Gantmacher's method to the problem in question is not appropriate, because the homogeneous transformation matrices possess special geometric structures and have well-known physical meanings. A simpler method that exploits the characteristics of these matrices is desirable.

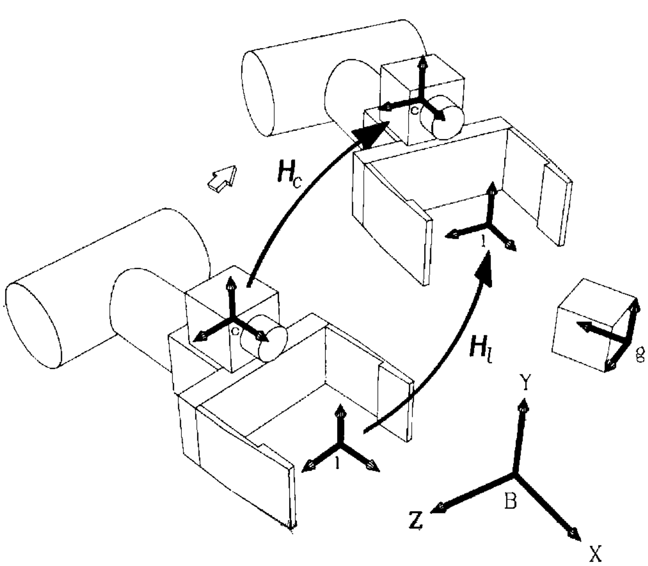

Fig. 2. Movement of a robot arm.

The kinematic equation, $H_l H_x = H_x H_c$, can be divided into two separate sets of equations (detailed in section 4). One is for rotation, and the other is for translation. An outline of solution methods was given by Shiu and Ahmad (1987). In this article, the Euler parameters (normalized quaternions) are used as orientation parameters. Using quaternions to describe relative orientations between coordinate systems can be traced back to Hamilton (1853) and Cayley (1885a,b; 1889). It did not become popular until the recent applications to kinematics (Sandor 1968), rigid-body dynamics (Robinson 1958; Yang and Freudenstein 1964; Koshlyakov 1983; Chelnokov 1984; Nikravesh and Chung 1982; Nikravesh 1984; Chou, Kesavan, and Singhal 1986; Chou, Singhal, and Kesavan 1986), control (Ickes 1970; Hendley 1971), robot dynamics simulation (Chou et al. 1987a,b), and photogrammetry (Horn 1987). Euler parameters and quaternion algebra provide a simple, unique, and elegant representation for describing finite rotations in space. Using quaternions to formulate the equation of rotation was first proposed by Chou and Kamel (1988), where a system of four nonlinear equations were chosen by a simple criterion. The equations were solved simultaneously using the Newton-Raphson iterative method.

In this article, we discuss the general solutions of both translation and rotation of the same problem solved by Shiu and Ahmad (1987). However, we use a totally different approach that is based on quaternions and the singular-value decomposition (SVD) analysis. First, the equations of rotation and translation are reformulated to their equivalent forms in terms of quaternions in order to arrive at two simple and well-structured linear systems. Then the generalized inverse method incorporating the SVD is applied to develop the general solutions, as the coefficient matrices of the equations of rotation and translation are both rank deficient. Although the equations are solved by analysis of the SVD, the actual computation of the SVD is avoided by deriving a set of closed-form formulae for the SVD analytically. This results in two general solutions for rotation and translation in closed form.

To obtain unique solutions for rotation and translation, two distinct robot movements have to be made to obtain two sets of data. The robot arm moves twice, one after another consecutively, to provide three position points that can be used to specify $H_l^{(1)}$ and $H_l^{(2)}$. At the same time, the camera views the object three times to determine $H_c^{(1)}$ and $H_c^{(2)}$. This produces a system of equations in which the number of equations is greater than the number of unknowns. Typically this overdetermined system can be solved by least-squares methods as proposed by Shiu and Ahmad

(1987). However, we develop simple criteria for choosing an independent set of equations automatically. For rotation, this leads to a mixed system of three linear equations and one nonlinear equation. Because the nonlinear equation is very simple, the system of four equations can be solved analytically to obtain a set of closed-form formulae for calculating the rotation. For translation, a set of two linear equations is derived from three equations to obtain a set of closed-form solution formulae.

2. *Homogeneous Transformations*

In three-dimensional euclidean space, a homogeneous transformation $\boldsymbol{H}$ is a 4×4 matrix:

$$\boldsymbol{H} = \begin{bmatrix} A_{(3\times3)} & \mathbf{r}_{(3\times1)} \\ \mathbf{v}_{(1\times3)} & s_{(1\times1)} \end{bmatrix} \tag{1}$$

which maps a position vector, expressed in homogeneous coordinates, from one coordinate system to another coordinate system with rotation (A), translation ($\mathbf{r}$), perspection ($\mathbf{v}$), and scaling (s). In the context of robot kinematics, a homogeneous transformation geometrically represents the location of a moving coordinate system with respect to a reference coordinate system; therefore a homogeneous transformation is also referred to as a coordinate transformation, and the entries $\boldsymbol{H}$ involve only the two significant parts of rotation and translation:

$$\boldsymbol{H} = \begin{bmatrix} A & \mathbf{r} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{2}$$

The inverse of this homogeneous transformation is

$$\boldsymbol{H}^{-1} = \begin{bmatrix} A^T & -A^T\mathbf{r} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{3}$$

The rotational transformation A is a 3×3 orthonormal matrix that may be specified by various orientation parameters such as direction cosines (three dextral orthonormal vectors—nine parameters), Euler angles (three angles), Euler axis-angle parameters (one unit vector and one angle—four parameters) (Shiu and Ahmad 1987; Wang and Ravani 1985; Paul 1981), or Euler parameters (one normalized quaternion—four parameters) (Nikravesh 1984; Wehage 1984).

The Euler parameters are a normalized quaternion, denoted by a column vector

$$\mathbf{p} = [e_0, e_1, e_2, e_3]^T = [e_0, \mathbf{e}^T]^T. \tag{4}$$

They are equivalent to the vector of Euler axis-angle parameters, $[\mathbf{u}^T, \theta]^T$, when we define

$$e_0 = \cos\frac{\theta}{2}; \qquad \mathbf{e} = \mathbf{u}\sin\frac{\theta}{2}. \tag{5}$$

Using Euler parameters or Euler axis-angle parameters to describe the orientation of a reference frame with respect to another reference frame has the advantage that they do not pose any difficulty in arriving at explicit formulae for the inverse problem; that is, no singular cases are encountered in determining Euler parameters (Klumpp 1976; Horn 1987) and Euler axis-angle parameters (Paul 1981) from a given rotational transformation matrix. A brief summary of quaternions, Euler parameters, Euler axis-angle parameters, and finite rotation is given in appendix A.

3. The Physical Problem

The relative positions and orientations among the base, link, camera, and object (or goal) can be represented as a graph (Fig. 3). The nodes B, 1, c, and g are the origins of the reference frames of the base, link, camera, and object, respectively. A directed edge represents a relative relationship between two reference frames and is associated with a homogeneous transformation. For instance, $\boldsymbol{H}_x$ is the relative transformation of the frame c with respect to the frame l. With this graph, the physical problem can be visualized easily.

To locate an object g in the base frame B, the transformation $\boldsymbol{H}_x$ has to be known at any instant of time to complete the path (B, l, c, g), as shown in Figure 3. The problem of finding $\boldsymbol{H}_x$ can be formulated mathematically by equating the homogeneous transformations in the middle loop of the graph as

$$\boldsymbol{H}_l\boldsymbol{H}_x = \boldsymbol{H}_x\boldsymbol{H}_c. \tag{6}$$

When the link of a robot moves, the relative transformation $\boldsymbol{H}_l$, which describes the relative position and orientation of the link frame *after* the motion with

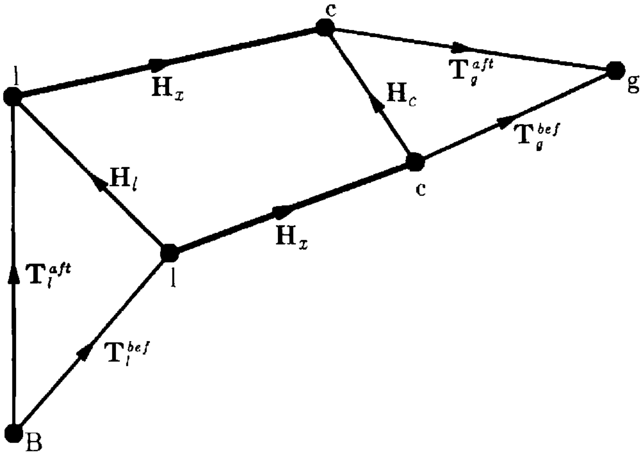

Fig. 3. Relative coordinate transformations represented by a graph.

respect to the link frame *before* the motion, can be obtained by equating the homogeneous transformations in the leftmost loop of Figure 3; that is, $T_l^{aft} = T_l^{bef} H_l$, where T_l specifies the absolute position and orientation of the link with respect to the base frame and is calculated by the robot controller from the joint encoder values. The superscripts "aft" and "bef" denote "after" and "before", respectively. The subscript "l" indicates that the transformations pertain to the *link*. Expressing H_l explicitly gives

$$H_l = (T_l^{bef})^{-1}(T_l^{aft}). \tag{7}$$

With the same movement, the transformation H_c, which describes the relative position and orientation of the camera *after* the motion with respect to the camera *before* the motion, can be derived by equating the homogeneous transformations in the rightmost loop (Fig. 3) as $T_g^{bef} = H_c T_g^{aft}$, where T_g specifies the relative position and orientation of an object in the goal frame (Craig 1986) with respect to the camera frame. The transformation T_g is given by any sensor capable of finding the three-dimensional position and orientation of an object. Expressing H_c explicitly gives

$$H_c = (T_g^{bef})(T_g^{aft})^{-1}. \tag{8}$$

As we can see from the graph, in order to locate an object in the base frame the fixed transformation H_x has to be known so that the position and orientation of the object can be calculated by $T_l H_x T_g$.

4. General Solutions

Expressing the aforementioned kinematic equation (6) in terms of rotational and translational transformation matrices, we have

$$\begin{bmatrix} A_l & \mathbf{r}_l \\ 0 & 1 \end{bmatrix}\begin{bmatrix} A_x & \mathbf{r}_x \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} A_x & \mathbf{r}_x \\ 0 & 1 \end{bmatrix}\begin{bmatrix} A_c & \mathbf{r}_c \\ 0 & 1 \end{bmatrix}. \tag{9}$$

Expanding (9) to

$$\begin{bmatrix} A_l A_x & A_l \mathbf{r}_x + \mathbf{r}_l \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} A_x A_c & A_x \mathbf{r}_c + \mathbf{r}_x \\ 0 & 1 \end{bmatrix}$$

and equating the first row yields two sets of equations as follows:

$$A_l A_x = A_x A_c \tag{10}$$

$$(A_l - I)\mathbf{r}_x = A_x \mathbf{r}_c - \mathbf{r}_l \tag{11}$$

where I is a 3×3 identity matrix. Equation (10), referred to as the *equation of rotation*, will be solved for the rotational quantity A_x, and the translational quantity $\mathbf{r}_x$ will be obtained by solving the *equation of translation* (11).

4.1. Solving the Equation of Rotation

Using Euler parameters (a normalized quaternion) to define the relative orientation between two coordinate systems provides a simple and elegant way to formulate successive rotations. As shown in (A21)–(A26) in appendix A, a sequence of rotations can be formulated as an equation without involving rotational transformation matrices. As a result, the problem of solving $A_l A_x = A_x A_c$ can be transformed into an equivalent problem involving the corresponding Euler parameters as follows:

$$\mathbf{p}_l \otimes \mathbf{p}_x = \mathbf{p}_x \otimes \mathbf{p}_c \tag{12}$$

where $\mathbf{p}_l$, $\mathbf{p}_c$ and $\mathbf{p}_x$ are Euler parameters that specify the rotational transformation matrices A_l, A_c, and A_x, respectively, and the operation $\otimes$ denotes quaternion multiplication. Expressing quaternion multiplication in matrix form and applying its special commutative property (see (A4)–(A8) in appendix A), we can further write (12) in matrix form as $\overset{+}{\mathbf{p}}_l \mathbf{p}_x = \bar{\mathbf{p}}_c \mathbf{p}_x$, or

$$(\overset{+}{\mathbf{p}}_l - \bar{\mathbf{p}}_c)\mathbf{p}_x = \mathbf{0} \tag{13}$$

where $\overset{+}{\mathbf{p}}_l$ and $\bar{\mathbf{p}}_c$ are orthonormal[2] matrices constructed with the Euler parameters $\mathbf{p}_l$ and $\mathbf{p}_c$, respectively, and $\mathbf{p}_x$ is a 4×1 column vector of unknown Euler parameters. Let $E = \overset{+}{\mathbf{p}}_l - \bar{\mathbf{p}}_c$. Then the problem of solving $A_l A_x = A_x A_c$ is transformed to the problem of solving a homogeneous linear system:

$$E\mathbf{p}_x = \mathbf{0}. \tag{14}$$

This homogeneous system of linear equations has a nontrivial solution if and only if the rank of the coefficient matrix E is less than 4.

Let $\mathbf{p}_l = [e_{0_l}, e_{1_l}, e_{2_l}, e_{3_l}]^T = [e_{0_l}, \mathbf{e}_l^T]^T$, and $\mathbf{p}_c = [e_{0_c}, e_{1_c}, e_{2_c}, e_{3_c}]^T = [e_{0_c}, \mathbf{e}_c^T]^T$; then

$$E = \begin{bmatrix} (e_{0_l} - e_{0_c}) & -(\mathbf{e}_l - \mathbf{e}_c)^T \\ (\mathbf{e}_l - \mathbf{e}_c) & (e_{0_l} - e_{0_c})I + (\tilde{\mathbf{e}}_l + \tilde{\mathbf{e}}_c) \end{bmatrix}.$$

The matrices $\tilde{\mathbf{e}}_l$ and $\tilde{\mathbf{e}}_c$ are constructed with the components of $\mathbf{e}_l$ and $\mathbf{e}_c$, respectively, and are skew-symmetric (detailed in A5). The matrix E can be further written as

$$E = \sin(\theta/2)\begin{bmatrix} 0 & -(\mathbf{u}_l - \mathbf{u}_c)^T \\ (\mathbf{u}_l - \mathbf{u}_c) & (\tilde{\mathbf{u}}_l + \tilde{\mathbf{u}}c) \end{bmatrix}$$

where $e_{0_l} \equiv \cos(\theta_l/2)$, $e_{0_c} \equiv \cos(\theta_c/2)$, $\mathbf{e}_l \equiv \sin(\theta_l/2)\mathbf{u}_l$, and $\mathbf{e}_c \equiv \sin(\theta_c/2)\mathbf{u}_c$, as given in (5). Also, $\theta = \theta_l = \theta_c$,

2. We distinguish an orthonormal matrix M from an orthogonal matrix N by the fact that the former has the property: $M^T M = MM^T = I$, and the latter the property: $N^T N = NN^T = D$ where D is a diagonal matrix, and I is an identity matrix.

which comes from the fact that $A_l = A_x A_c A_x^T$. Because the matrices A_l and A_c are similar, the angles of rotation defined by A_l and A_c are identical. This has been shown by the proof of lemma 4 in Shiu and Ahmad (1987).

Eliminating sin $(\theta/2)$ from both sides, we can write (13) as:

$$(\overset{+}{\mathbf{u}}'_l - \bar{\mathbf{u}}'_c)\mathbf{p}_x = \mathbf{0}, \text{ or } \mathbf{u}'_l = (\bar{\mathbf{p}}_x^T \overset{+}{\mathbf{p}}_x)\mathbf{u}'_c = A_x \mathbf{u}'_c, \qquad (15)$$

where $\mathbf{u}'_l = [0, \mathbf{u}_l^T]^T$ and $\mathbf{u}'_c = [0, \mathbf{u}_c^T]^T$ are *vector quaternions.* Equation (15) naturally shows that any rotational transformation that rotates $\mathbf{u}_c$ into $\mathbf{u}_l$ is a solution to (10). The same result has been obtained by in theorem 3 Shiu and Ahmad (1987). Equation (15), in fact, provides an alternative form of the problem described in (10). In addition, the method works when sin $(\theta/2)$ is not zero; that is, $\theta \neq 360°n$ $(n = 0, 1, \ldots)$ or $A_l \neq A_c \neq I$.

Now, let us define

$$B \equiv \overset{+}{\mathbf{u}}'_l - \bar{\mathbf{u}}'_c = \begin{bmatrix} 0 & -(\mathbf{u}_l - \mathbf{u}_c)^T \\ (\mathbf{u}_l - \mathbf{u}_c) & (\tilde{\mathbf{u}}_l + \tilde{\mathbf{u}}_c) \end{bmatrix}. \qquad (16)$$

Because B is a 4×4 skew-symmetric matrix, the rank of B is either 2 or 4 (Bradley 1975). To solve $B\mathbf{p}_x = \mathbf{0}$, we use the SVD (Stewart 1973; Golub and Van Loan 1983) to diagonalize the system by decomposing B into

$$B = U\Sigma V^T,$$

where $\Sigma = diag[\sigma_1, \sigma_2, \sigma_3, \sigma_4]$. The numbers σ_i $(i = 1, \ldots, 4)$ are called the *singular values of* B, and they are the positive square roots of the eigenvalues of B^TB. The columns of U are called the *left singular vectors* of B (or the orthonormal eigenvectors of BB^T), and the columns of V are called the *right singular vectors* of B (or the orthonormal eigenvectors of B^TB). Because $\sigma_1 = \sigma_2 = 2$, and $\sigma_3 = \sigma_4 = 0$ (see appendix B), the matrix $B = U\Sigma V^T$, where

$$\Sigma = \begin{bmatrix} D & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

and $D = diag[\sigma_1, \sigma_2]$, is a matrix of rank 2.

After applying the SVD, the problem of solving $B\mathbf{p}_x = \mathbf{0}$ becomes the problem of solving

$$\begin{cases} (U\Sigma)\mathbf{y} = \mathbf{0} \\ V^T\mathbf{p}_x = \mathbf{y} \end{cases}. \qquad (17)$$

Partitioning $\mathbf{y} = [\mathbf{y}_1^T, \mathbf{y}_2^T]^T$, $U = [U_1, U_2]$, and $V = [V_1, V_2]$ according to the partition of Σ, we can solve the first equation of (17) to obtain

$$\begin{cases} \mathbf{y}_1 = D^{-1}\mathbf{U}_1^T\mathbf{0} = \mathbf{0} \\ \mathbf{y}_2 = arbitrary \end{cases} \qquad (18)$$

where the columns of U_1 span the range of the matrix B. Because the right side of the first equation in (17) is $\mathbf{0}$, U_1 is insignificant. The generalized inverse solution can be obtained from the second equation of (17)

$$\mathbf{p}_x = V_1\mathbf{y}_1 + V_2\mathbf{y}_2, \qquad (19)$$

where the columns of V_2 span the null-space (kernel) of B. Substituting (18) into (19), we obtain the complete solution as follows:

$$\mathbf{p}_x = V_2\mathbf{y}_2; \qquad \mathbf{y}_2 = arbitrary. \qquad (20)$$

The solution given by (20) has one degree of freedom, because the Euler parameters have to satisfy the normality constraint $\mathbf{p}_x^T\mathbf{p}_x = 1$, which is $\mathbf{y}_2^T V_2^T V_2 \mathbf{y}_2 = \mathbf{y}_2^T\mathbf{y}_2 = 1$. As we can see, there are an infinite number of solutions for $\mathbf{p}_x$.

Because the matrix B is very simple and possesses a special structure, the right singular vectors of V_2 that span the null-space of B can be constructed symbolically such that the implementation of the SVD can be avoided completely. In appendix B, the closed-form formula of V_2 is derived, giving the following result:

$$V_2 = \begin{bmatrix} -\frac{1}{2}m_2 & 0 \\ \frac{\tilde{\mathbf{u}}_l\mathbf{u}_c}{m_2} & \frac{(\mathbf{u}_l + \mathbf{u}_c)}{m_2} \end{bmatrix}; \quad m_2 = \sqrt{2 + 2\mathbf{u}_l^T\mathbf{u}_c}. \qquad (21)$$

For the special case when $\mathbf{u}_l = -\mathbf{u}_c$, we use (B11).

This gives the general solution for rotation. Although it is obtained using analysis of the SVD, the final solution formula is in a closed form, and *the actual implementation of the SVD is not necessary.*

4.2. Solving the Equation of Translation

Now let us consider the solution of

$$(A_l - I)\mathbf{r}_x = \mathbf{c} \qquad (22)$$

where

$$\mathbf{c} = A_x\mathbf{r}_c - \mathbf{r}_l. \qquad (23)$$

Applying the quaternion algebra again, we can transform (22) to a simpler form.

First, let us view $\mathbf{r}_x$ and $\mathbf{c}$ as vector quaternions: $\mathbf{r}'_x = [0, \mathbf{r}_x^T]^T$, and $\mathbf{c}' = [0, \mathbf{c}^T]^T$, and rewrite (22) in quaternion space

$$(\overset{+}{\mathbf{p}}_l\bar{\mathbf{p}}_l^T - \bar{\mathbf{p}}_l\bar{\mathbf{p}}_l^T)\mathbf{r}'_x = \mathbf{c}',$$

where A_l is a 4×4 matrix, as shown in (A12). The matrix I is a 4×4 identity matrix that can be redefined as $I = \bar{\mathbf{p}}_l^T$, as $\bar{\mathbf{p}}_l$ is orthonormal. Simplifying again by taking $\bar{\mathbf{p}}_l^T$ out and associating it with $\mathbf{r}'_x$, we get $(\overset{+}{\mathbf{p}}_l - \bar{\mathbf{p}}_l)(\bar{\mathbf{p}}_l^T\mathbf{r}'_x) = \mathbf{c}'$, or in matrix form

$$\begin{bmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & 2\tilde{\mathbf{e}}_l \end{bmatrix} \begin{bmatrix} r_0 \\ \mathbf{r} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{c} \end{bmatrix} \tag{24}$$

where $\bar{\mathbf{p}}_l^T \mathbf{r}_x' \equiv \mathbf{r}' = [r_0, \mathbf{r}^T]^T$, or

$$\mathbf{r}_x' = \bar{\mathbf{p}}_l \mathbf{r}'. \tag{25}$$

From (24), we can return to 3-space and reach a simplified system that is equivalent to the system in (22) as follows: $r_0 = arbitrary$, and $\tilde{\mathbf{e}}_l \mathbf{r} = \frac{1}{2}\mathbf{c}$, or further written as

$$\begin{cases} r_0 = arbitrary \\ \tilde{\mathbf{u}}_l \mathbf{r} = k\mathbf{c}; \quad k = \dfrac{1}{2 \sin \dfrac{\theta_l}{2}} \end{cases} \tag{26}$$

as $\tilde{\mathbf{e}}_l = (\sin \theta_l/2)\tilde{\mathbf{u}}_l$. At this stage, the problem of solving (22) is completely transformed into the problem of solving (26), which is a linear system of three equations with a well-structured coefficient matrix $\tilde{\mathbf{u}}_l$. From a mathematical point of view, we have transformed the augmented unknown vector $\mathbf{r}_x'$ by an orthonormal matrix $\bar{\mathbf{p}}_l^T$. From the physical point of view we in fact rotated the vector $\mathbf{r}_x$ about $\mathbf{u}_l$ to be on the plane that is perpendicular to the vector $\mathbf{c}$. However, we can see that if $\theta_l = 360°n$ ($n = 0, 1, \ldots$), the problem is not defined.

Decomposing $\tilde{\mathbf{u}}_l$ by the SVD gives

$$\tilde{\mathbf{u}}_l = U\Sigma V^T.$$

Because $\Sigma = diag[\sigma_1, \sigma_2, \sigma_3]$, $\sigma_1 = \sigma_2 = \sqrt{\mathbf{u}_l^T\mathbf{u}_l} = 1 \neq 0$, and $\sigma_3 = 0$ (see appendix C), the rank of $\tilde{\mathbf{u}}_l$ is 2. The problem of solving (26) becomes the problem of solving

$$\begin{cases} \Sigma \mathbf{z} = kU^T\mathbf{c} \\ V^T\mathbf{r} = \mathbf{z}. \end{cases} \tag{27}$$

Partitioning U, V, and Σ according to the nonzero singular values and zero singular values, we rewrite the first equation of (27) in matrix form as

$$\begin{bmatrix} D & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{z}_1 \\ z_2 \end{bmatrix} = k \begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix} \mathbf{c}$$

and solve for $\mathbf{z}$ as

$$\begin{cases} \mathbf{z}_1 = kD^{-1}U_1^T\mathbf{c} \\ z_2 = arbitrary \end{cases} \tag{28}$$

where $D = diag[\sigma_1, \sigma_2]$. From the second equation of (27), we have the generalized inverse solution

$$\mathbf{r} = \mathbf{V}_1\mathbf{z}_1 + V_2 z_2. \tag{29}$$

Substituting $\mathbf{z}_1$ into (28) gives $\mathbf{r} = kV_1D^{-1}U_1^T\mathbf{c} + V_2z_2$, or simply $\mathbf{r} = -k\tilde{\mathbf{u}}_l\mathbf{c} + V_2z_2$, because $D^{-1} = D^T = D = I$, and $(U_1DV_1^T)^T = \tilde{\mathbf{u}}_l^T = -\tilde{\mathbf{u}}_l$. Furthermore, the right singular vectors of V_2 that span the null-space of $\tilde{\mathbf{u}}_l$ can be derived as (see appendix C)

$$V_2 = \mathbf{u}_l,$$

and therefore

$$\mathbf{r} = -k\tilde{\mathbf{u}}_l\mathbf{c} + \mathbf{u}_l z_2. \tag{30}$$

To find the original solution of (22), we have to scale the vector $\mathbf{r}'$ back using (25). In matrix form, (25) is written as

$$\begin{bmatrix} 0 \\ \mathbf{r}_x \end{bmatrix} = \begin{bmatrix} e_{0l} & -\mathbf{e}_l^T \\ \mathbf{e}_l & e_{0l}I - \tilde{\mathbf{e}}_l \end{bmatrix} \begin{bmatrix} r_0 \\ \mathbf{r} \end{bmatrix}. \tag{31}$$

From the first equation in (31), we solve for $r_0 = (1/e_{0_l})\mathbf{e}_l^T\mathbf{r}$. Substituting r_0 and (30) into the second equation of (31) and applying some simple manipulations, we get the complete solution of $\mathbf{r}_x$ as

$$\mathbf{r}_x = \left(\frac{1}{e_{0_l}}\right)\mathbf{u}_l z_2 - k(e_{0_l}I - \tilde{\mathbf{e}}_l)(\tilde{\mathbf{u}}_l\mathbf{c}); \quad z_2 = arbitrary. \tag{32}$$

The quantities pertaining to Euler parameters in (32) can be transferred to $\mathbf{u}$ and θ using (5) to reach the following formula:

$$\mathbf{r}_x = \mathbf{u}_l z_3 - \frac{1}{2}\left[\cot\left(\frac{\theta_l}{2}\right)\tilde{\mathbf{u}}_l + I\right]\mathbf{c}; \quad z_3 = arbitrary \tag{33}$$

where $z_3 = (1/e_{0_l})z_2$. In reaching (33), the identity $\mathbf{u}_l^T\mathbf{c} = \mathbf{0}$ has been applied. This results from substituting $(A_l - I)\mathbf{r}_x$ for $\mathbf{c}$ and identifying $\mathbf{u}_l = A_l\mathbf{u}_l$.

Equation (33) gives the general solution of translation, which has one degree of freedom. As in the solution of rotation, the final solution formula is in a closed form, and the implementation of the SVD is eliminated.

5. Finding Unique Solutions

The solutions of rotation and translation presented in the previous section have two degrees of freedom and are not unique. To obtain a unique solution, two distinct sets of data $H_l^{(1)}$, $H_c^{(1)}$, $H_l^{(2)}$, and $H_c^{(2)}$ should be collected from two movements of the robot arm. This idea was also introduced by Shiu and Ahmad (1987).

Because the solution formulae are in terms of Euler parameters or Euler axis-angle parameters, their extraction from the given A_l and A_c is necessary. The inverse formulae for this purpose can be found in Klumpp (1976), Paul (1981), and Horn (1987).

5.1. A Unique Solution For Rotation

Given two sets of data (superscribed by 1 and 2, respectively), the solutions of the equation of rotation can be written as

$$\begin{cases} \mathbf{p}_x = V_2^{(1)}\mathbf{y}_2^{(1)} \\ \mathbf{p}_x = V_2^{(2)}\mathbf{y}_2^{(2)} \end{cases} \tag{34}$$

which are subject to two scalar constraints

$$\begin{cases} \mathbf{y}_2^{(1)T}\mathbf{y}_2^{(1)} = 1 \\ \mathbf{y}_2^{(2)T}\mathbf{y}_2^{(2)} = 1 \end{cases} \tag{35}$$

where $V_2^{(1)}$ and $V_2^{(2)}$ are the right singular vectors spanning the null-space of $\boldsymbol{B}^{(1)}$ and $\boldsymbol{B}^{(2)}$, respectively.

Equating (34) gives $V_2^{(1)}\mathbf{y}_2^{(1)} - V_2^{(2)}\mathbf{y}_2^{(2)} = \mathbf{0}$, or

$$[V_2^{(1)} - V_2^{(2)}]\begin{bmatrix} \mathbf{y}_2^{(1)} \\ \mathbf{y}_2^{(2)} \end{bmatrix} = \mathbf{0}. \tag{36}$$

Equations (35) and (36) are a total of six scalar equations in four unknowns. Three of (36) and one of (35) may be chosen to form a system of four simultaneous equations. However, equations in (35) cannot be used at the same time, as one implies the other. This can be shown by applying the first equation in (35) to (36) and using the identity, $V_2^{(1)T}V_2^{(1)} = V_2^{(2)T}V_2^{(2)} = \boldsymbol{I}$; the result gives the second equation of (35).

Forming the normal equation of (36) and identifying $V_2^{(1)T}V_2^{(1)} = V_2^{(2)T}V_2^{(2)} = \boldsymbol{I}$, we have

$$\begin{bmatrix} \boldsymbol{I} & -\boldsymbol{W} \\ -\boldsymbol{W}^T & \boldsymbol{I} \end{bmatrix}\begin{bmatrix} \mathbf{y}_2^{(1)} \\ \mathbf{y}_2^{(2)} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{37}$$

where

$$\boldsymbol{W} = V_2^{(1)T}V_2^{(2)} = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix}.$$

If the normal matrix has full rank, we can choose any three out of four from (37). However, we can see that the first two equations have full row rank of 2; we can always choose them. The other two equations may impose dependency on the first two. A criterion that is designed to choose one equation from the second set of (37) is developed below.

Let's perform Gaussian elimination (GE) on the normal matrix:

$$\begin{bmatrix} 1 & 0 & -w_{11} & -w_{12} \\ 0 & 1 & -w_{21} & -w_{22} \\ 0 & 0 & \alpha & \gamma \\ 0 & 0 & \gamma & \beta \end{bmatrix} \tag{38}$$

where

$$\alpha = 1 - w_{11}^2 - \omega_{21}^2 \tag{39}$$

$$\beta = 1 - w_{12}^2 - w_{22}^2 \tag{40}$$

$$\gamma = -w_{11}w_{12} - w_{21}w_{22}. \tag{41}$$

In (38), if the last two rows are all zeroes, the corresponding two equations are dependent; we do not have a sufficient set of equations to solve simultaneously. If one of the last two rows is zero, we can choose the equation corresponding to the other row to solve simultaneously. When the last two rows are not zero, we can perform GE again using q as the pivoting element to get a matrix

$$\begin{bmatrix} 1 & 0 & \times & \times \\ 0 & 1 & \times & \times \\ 0 & 0 & q & \times \\ 0 & 0 & 0 & \dfrac{\alpha\beta - \gamma^2}{\pm q} \end{bmatrix}$$

where q is α, β, or γ that has the largest absolute value. If $\alpha\beta - \gamma^2 \neq 0$, we can choose either one of the last two equations. If $\alpha\beta - \gamma^2 = 0$, we can choose the equation corresponding to the third row. From the previous discussion, we can conclude that an equation corresponding to the row with the largest absolute value of q can be chosen for a simultaneous solution. However, it is not necessary to use γ, as it appears in both rows; we can use α and β only to choose the equation.

Case 1

If $\alpha = \beta = \gamma = 0$, no unique solution exists. This case is equivalent to having $\boldsymbol{I} - \boldsymbol{W}^T\boldsymbol{W} = \mathbf{0}$. This condition can be obtained when $\mathbf{u}_l^{(1)}$ and $\mathbf{u}_l^{(2)}$ are parallel or antiparallel (similar for $\mathbf{u}_c^{(1)}$ and $\mathbf{u}_c^{(2)}$) by substituting $\mathbf{u}_l^{(1)T}\mathbf{u}_l^{(2)} = \mathbf{u}_c^{(1)T}\mathbf{u}_c^{(2)} = 1$ or -1 into the equation of $\boldsymbol{W}$. This conclusion has also been obtained by Shiu and Ahmad (1987; theorem 4).

Case 2

If $|\alpha| \geq |\beta|$, we choose the first equation of $-\boldsymbol{W}^T\mathbf{y}_2^{(1)} + \mathbf{y}_2^{(2)} = \mathbf{0}$; therefore we have a system of three linear equations and one nonlinear equation as follows:

$$\begin{cases} y_{21}^{(1)} - w_{11}y_{21}^{(2)} - w_{12}y_{22}^{(2)} = 0 \\ y_{22}^{(1)} - w_{21}y_{21}^{(2)} - w_{22}y_{22}^{(2)} = 0 \\ -w_{11}y_{21}^{(1)} - w_{21}y_{22}^{(1)} + y_{21}^{(2)} = 0 \\ y_{21}^{(2)2} + y_{22}^{(2)2} = 1 \end{cases} \tag{42}$$

where $\mathbf{y}_2^{(1)} = [y_{21}^{(1)}, y_{22}^{(1)}]^T$ and $y_2^{(2)} = [y_{21}^{(2)}, y_{22}^{(2)}]^T$. From (38), we get

$$\alpha y_{21}^{(2)} + \gamma y_{22}^{(2)} = 0. \tag{43}$$

Solving (43) with the fourth equation of (42) gives a set of closed-form formulae:

$$\begin{cases} y_{21}^{(2)} = \pm \dfrac{\gamma}{\sqrt{\alpha^2 + \gamma^2}} \\ y_{22}^{(2)} = \mp \dfrac{\alpha}{\sqrt{\alpha^2 + \gamma^2}} \end{cases} \tag{44}$$

The values of $y_{21}^{(1)}$ and $y_{22}^{(1)}$ can be calculated using the first two equations of (42), if necessary.

Case 3

If $|\alpha| < |\beta|$, we choose the second equation of $-W^T \mathbf{y}_2^{(1)} + \mathbf{y}_2^{(2)} = 0$. Similarly, we can derive another set of closed-form formulae:

$$\begin{cases} y_{21}^{(2)} = \pm \dfrac{\beta}{\sqrt{\beta^2 + \gamma^2}} \\ y_{22}^{(2)} = \mp \dfrac{\gamma}{\sqrt{\beta^2 + \gamma^2}} \end{cases} \tag{45}$$

Again, $y_{21}^{(1)}$ and $y_{22}^{(1)}$ can be calculated if needed. Cases 2 and 3 include the cases when $\beta = \gamma = 0$ and $\alpha = \gamma = 0$, respectively.

There are two choices for selecting the signs in each of sets (44) and (45). One choice may give a positive $\mathbf{p}_x$ and the other result in a negative $\mathbf{p}_x$. However, because the entries of a rotational matrix defined by Euler parameters are quadratic functions, $\mathbf{p}_x$ and $-\mathbf{p}_x$ will arrive at the same matrix A_x. The choice of signs in (44) and (45) is arbitrary. Now, $\mathbf{p}_x$ can be evaluated using the equation $\mathbf{p}_x = V_2^{(2)} \mathbf{y}_2^{(2)}$.

A similar set of closed-form formulae for $y_{21}^{(1)}$ and $y_{22}^{(1)}$ can also be derived, if the previous derivations are to be based on the following equations: $\mathbf{y}_2^{(2)} - W^T \mathbf{y}_2^{(1)} = \mathbf{0}$, one of $\mathbf{y}_2^{(1)} - W \mathbf{y}_2^{(2)} = \mathbf{0}$, and $y_{21}^{(1)2} + y_{22}^{(1)2} = 1$.

5.2. *A Unique Solution For Translation*

Given two sets of data (superscribed by 1 and 2, respectively), equations (32) can be written as

$$\begin{cases} \mathbf{r}_x = \mathbf{u}_l^{(1)} z_3^{(1)} + \mathbf{d}^{(1)} \\ \mathbf{r}_x = \mathbf{u}_l^{(2)} z_3^{(2)} + \mathbf{d}^{(2)} \end{cases} \tag{46}$$

where

$$\mathbf{d}^{(i)} = -\frac{1}{2}\left[\cot\left(\frac{\theta_l^{(i)}}{2}\right) \tilde{\mathbf{u}}_l^{(i)} + I\right] \mathbf{c}^{(i)}; \quad i = 1, 2. \tag{47}$$

Equating (46), we have

$$\mathbf{u}_l^{(1)} z_3^{(1)} - \mathbf{u}_l^{(2)} z_3^{(2)} = \mathbf{d}^{(2)} - \mathbf{d}^{(1)}. \tag{48}$$

There are three equations in two unknowns; two equations may be chosen to solve the two unknowns, if the coefficient matrix has full column rank of 2.

Using $\mathbf{u}_l^{(1)T}\mathbf{u}_l^{(1)} = \mathbf{u}_l^{(2)T}\mathbf{u}_l^{(2)} = 1$, we can form the normal equation of (48) as follows:

$$\begin{bmatrix} 1 & -\mathbf{u}_l^{(1)T}\mathbf{u}_l^{(2)} \\ -\mathbf{u}_l^{(2)T}\mathbf{u}_l^{(1)} & 1 \end{bmatrix} \begin{bmatrix} z_3^{(1)} \\ z_3^{(2)} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_l^{(1)T}\mathbf{d}^{(2)} \\ \mathbf{u}_l^{(2)T}\mathbf{d}^{(1)} \end{bmatrix}$$

where $\mathbf{u}_l^{(1)T}(\mathbf{d}^{(2)} - \mathbf{d}^{(1)}) = \mathbf{u}_l^{(1)T}\mathbf{d}^{(2)}$ and $\mathbf{u}_l^{(2)T}(\mathbf{d}^{(2)} - \mathbf{d}^{(1)}) = -\mathbf{u}_l^{(2)T}\mathbf{d}^{(1)}$ can be verified easily. If the normal matrix has a rank of 2, we can find $z_3^{(1)}$ and $z_3^{(2)}$ uniquely; otherwise, if $1 - (\mathbf{u}_l^{(1)T}\mathbf{u}_l^{(2)})^2 = 0$, we cannot find a unique solution.

Solving the normal equations, we obtain $z_3^{(1)}$ and $z_3^{(2)}$ in closed-form:

$$\begin{cases} z_3^{(1)} = \dfrac{\mathbf{u}_l^{(1)T}\mathbf{d}^{(2)} + m\mathbf{u}_l^{(2)T}\mathbf{d}^{(1)}}{1 - m^2} \\ z_3^{(2)} = \dfrac{\mathbf{u}_l^{(2)T}\mathbf{d}^{(1)} + m\mathbf{u}_l^{(1)T}\mathbf{d}^{(2)}}{1 - m^2}; \quad m = \mathbf{u}_l^{(1)T}\mathbf{u}_l^{(2)}. \end{cases} \tag{49}$$

Substituting $z_3^{(1)}$ and $z_3^{(2)}$ into (46), we can calculate $\mathbf{r}_x$ from either of the following two closed-form formulae:

$$\begin{cases} \mathbf{r}_x = \left(\dfrac{\mathbf{u}_l^{(1)T}\mathbf{d}^{(2)} + m\mathbf{u}_l^{(2)T}\mathbf{d}^{(1)}}{1 - m^2}\right) \mathbf{u}_l^{(1)} + \mathbf{d}^{(1)} \\ \mathbf{r}_x = \left(\dfrac{\mathbf{u}_l^{(2)T}\mathbf{d}^{(1)} + m\mathbf{u}_l^{(1)T}\mathbf{d}^{(2)}}{1 - m^2}\right) \mathbf{u}_l^{(2)} + \mathbf{d}^{(2)}; \quad m = \mathbf{u}_l^{(1)T}\mathbf{u}_l^{(2)}. \end{cases} \tag{50}$$

Again, for the case when $1 - m^2 = 0$, we cannot find a unique solution. The constraint indicates that $\mathbf{u}_l^{(1)T}\mathbf{u}_l^{(2)} \neq 1$ or -1; that is, $\mathbf{u}_l^{(1)}$ and $\mathbf{u}_l^{(2)}$ cannot be parallel or antiparallel (theorem 4 in Shiu and Ahmad [1987]).

6. Algorithm

The discussion presented in the previous sections can be summarized into algorithmic steps for computer implementation as follows:

Input and Output

- Given two distinct sets of input data: homogeneous transformations $H_l^{(i)}$ and $H_c^{(i)}$ $(i = 1, 2)$ are calculated using (7) and (8).
- The algorithm finds A_x and $\mathbf{r}_x$ as the relative orientation and position between the reference frames of a link-mounted sensor and the link of a robot arm.

Initialization

- For $i = 1, 2$, calculate $\theta_l^{(i)}$, $\theta_c^{(i)}$, $\mathbf{u}_l^{(i)}$, and $\mathbf{u}_c^{(i)}$.
- Calculate $\mathbf{p}_l^{(i)}$ and $\mathbf{p}_c^{(i)}$ using (5), if needed.
- If $(\theta_l^{(i)} \neq \theta_c^{(i)})$, output "Incorrect Data: The angles θ_l and θ_c must be equal."; stop.

- If ($\theta_l^{(i)} = 360°n$, $n = 0, 1, \ldots$), output "Improper data: The angle θ_1 must not be a multiple of 360°."; stop.
- If ($\mathbf{u}_l^{(1)T}\mathbf{u}_l^{(2)} = 1$ or -1), output "No Unique Solution"; stop.

Rotation

- For $i = 1, 2$, compute $V_2^{(i)}$. If $\mathbf{u}_l^{(i)} \neq \mathbf{u}_c^{(i)}$, use (21); otherwise, use (B11).
- Compute $W = V_2^{(1)T}V_2^{(2)}$.
- Compute α, β, and γ using (39), (40), and (41).
- If $|\alpha| \geqslant |\beta|$, compute $y_{21}^{(2)}$ and $y_{22}^{(2)}$ using (44); else, compute $y_{21}^{(2)}$ and $y_{22}^{(2)}$ using (45).
- Compute $\mathbf{p}_x$ and A_x using (34) and (A14), respectively.

Translation

- For $i = 1, 2$, compute $\mathbf{c}^{(i)}$ and $\mathbf{d}^{(i)}$ using (23) and (47), respectively.
- Compute $\mathbf{r}_x$ using (50).

7. Example

A program called WHEREC (WHERE is the Camera?) has been coded to test the algorithm. The program was written in FORTRAN and was run on an IBM 4341. Consider a camera that is mounted on the hand of a robot arm, as shown in Figure 1. The camera is mounted with the position and orientation relative to the hand coordinate frame 1_6 as follows:

$$H_x = \begin{bmatrix} -0.88405797 & -0.40579710 & -0.23188406 & 11.00000000 \\ -0.40579710 & 0.42028986 & 0.81159420 & 21.00000000 \\ -0.23188406 & 0.81159420 & -0.53623188 & -18.00000000 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (51)$$

Input

Moving the arm twice, we can find $H_l^{(i)}$ and $H_c^{(i)}$ ($i = 1, 2$) by $T_c^{(j)}$ and $T_l^{(j)}$ ($j = 1, 2, 3$) using the following equations:

$$\begin{cases} H_l^{(i)} = [T_l^{(i)}]^{-1}[T_l^{(i+1)}] \\ H_c^{(i)} = [T_c^{(i)}]^{-1}[T_c^{(i+1)}]; \quad i = 1, 2 \end{cases} \quad (52)$$

They are found numerically as

$$H_l^{(1)} = \begin{bmatrix} -0.87179487 & 0.48717949 & -0.05128205 & 5.00000000 \\ 0.33333333 & 0.66666667 & 0.66666667 & -4.00000000 \\ 0.35897436 & 0.56410256 & -0.74358974 & 3.00000000 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (53)$$

$$H_l^{(2)} = \begin{bmatrix} -0.70114943 & 0.02298850 & -0.71264368 & 2.00000000 \\ 0.66666667 & -0.33333333 & -0.66666667 & -3.00000000 \\ -0.25287356 & -0.94252874 & 0.21839080 & 9.00000000 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (54)$$

$$H_c^{(1)} = \begin{bmatrix} -0.13831397 & -0.61660716 & -0.77502572 & 0.1311780 \\ -0.84328869 & -0.33704404 & 0.41864724 & 34.399851 \\ -0.51935868 & 0.71147518 & -0.47335994 & -41.570048 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (55)$$

$$H_c^{(2)} = \begin{bmatrix} -0.69307617 & 0.66439727 & -0.27968142 & 7.6275196 \\ 0.01005777 & -0.37903029 & -0.92532961 & -3.1216059 \\ -0.72079419 & -0.64413687 & 0.25601450 & -8.9446943 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (56)$$

Initialization

The axes of rotation are found to be

$$\begin{cases} \mathbf{u}_l^{(1)} = [0.22792115, 0.91168461, 0.34188173]^T \\ \mathbf{u}_l^{(2)} = [-0.32929278, -0.54882130, 0.76834982]^T \\ \mathbf{u}_c^{(1)} = [-0.65073141, 0.56815128, 0.50373878]^T \\ \mathbf{u}_c^{(2)} = [0.33565593, 0.52655029, -0.78107611]^T \end{cases} \quad (57)$$

and the angles of rotation are

$$\begin{cases} \theta_l^{(1)} = \theta_c^{(1)} = 193.002824° \\ \theta_l^{(2)} = \theta_c^{(2)} = 155.236701° \end{cases}. \quad (58)$$

Because $\theta_l^{(1)} = \theta_c^{(1)}$, $\theta_l^{(2)} = \theta_c^{(2)}$, and $\mathbf{u}_l^{(1)T}\mathbf{u}_l^{(2)} = \mathbf{u}_c^{(1)T}\mathbf{u}_c^{(2)} = -0.31271995 \neq 1$ or -1, a unique solution exists.

Rotation

The singular vectors that span the null-space of $B^{(1)}$ and $B^{(2)}$ are calculated using (21) as

$$V_2^{(1)} = \begin{bmatrix} 0.87803144 & 0.00000000 \\ -0.15091165 & 0.24077171 \\ 0.19206938 & -0.84270097 \\ -0.41157724 & -0.48154341 \end{bmatrix} \quad (59)$$

$$V_2^{(2)} = \begin{bmatrix} -0.01321406 & 0.00000000 \\ 0.91177042 & -0.24077171 \\ 0.02642813 & 0.84270097 \\ 0.40963599 & 0.48154341 \end{bmatrix}. \quad (60)$$

Now the matrix W can be calculated as

$$W = V_2^{(1)T}V_2^{(2)} = \begin{bmatrix} -0.31271996 & -0.00000000 \\ 0.00000000 & -1.00000000 \end{bmatrix} \quad (61)$$

and accordingly

$$\begin{cases} \alpha = 0.90220622 \\ \beta = 0.00000000 \\ \gamma = 0.00000000. \end{cases} \quad (62)$$

Because $|\alpha| \geqslant |\beta|$, we use (44) to calculate $\mathbf{y}_2^{(2)}$

$$\mathbf{y}_2^{(2)} = \begin{bmatrix} y_{21}^{(2)} \\ y_{22}^{(2)} \end{bmatrix} = \begin{bmatrix} -0.00000000 \\ 1.00000000 \end{bmatrix}. \quad (63)$$

Using the first two equations of (42), we can find $\mathbf{y}_2^{(1)}$ as

$$\mathbf{y}_2^{(1)} = W\mathbf{y}_2^{(2)} = \begin{bmatrix} y_{21}^{(1)} \\ y_{22}^{(1)} \end{bmatrix} = \begin{bmatrix} -0.00000000 \\ -0.99999999 \end{bmatrix}. \quad (64)$$

Now $\mathbf{p}_x$ can be found by either of the following formulae:

$$\begin{cases} \mathbf{p}_x = V_2^{(1)}\mathbf{y}_2^{(1)} = [0.00000000, -0.24077171, 0.84270097, 0.48154341]^T \\ \mathbf{p}_x = V_2^{(2)}\mathbf{y}_2^{(2)} = [0.00000000, -0.24077171, 0.84270097, 0.48154341]^T \end{cases}. \quad (65)$$

The rotational matrix is computed using (A.14) as

$$A_x = \begin{bmatrix} -0.88405797 & -0.40579710 & -0.23188406 \\ -0.40579710 & 0.42028986 & 0.81159420 \\ -0.23188406 & 0.81159420 & -0.53623188 \end{bmatrix}, \quad (66)$$

which is identical to the rotational matrix given in (51).

Translation

The $\mathbf{c}$ vectors can be calculated using (23) as

$$\begin{cases} \mathbf{c}^{(1)} = [-9.4358974, -15.333333, 47.179487]^T \\ \mathbf{c}^{(2)} = [-5.4022989, -8.6666667, -8.5057471]^T \end{cases} \quad (67)$$

and the $\mathbf{d}$ vectors are calculated using (47)

$$\begin{cases} \mathbf{d}^{(1)} = [7.4675325, 6.8701299, -23.298701]^T \\ \mathbf{d}^{(2)} = [1.4578313, 5.0963855, 4.2650602]^T \end{cases}. \quad (68)$$

The unknown $\mathbf{r}_x$ can be calculated directly using (50) as

$$\begin{cases} \mathbf{r}_x = \mathbf{u}_l^{(1)}z_3^{(1)} + \mathbf{d}^{(1)} = [11.00000000, 21.00000000, -18.00000000]^T \\ \mathbf{r}_x = \mathbf{u}_l^{(2)}z_3^{(2)} + \mathbf{d}^{(2)} = [11.00000000, 21.00000000, -18.00000000]^T \end{cases} \quad (69)$$

where

$$\begin{cases} z_3^{(1)} = -1.7548752 \\ z_3^{(2)} = -6.2134849 \end{cases}. \quad (70)$$

The $\mathbf{r}_x$ is identical to the $\mathbf{r}_x$ given in (51).

8. Conclusions

A direct application of quaternion algebra, incorporating analysis of the SVD, to the solution of the kinematic equations $H_lH_x = H_xH_c$ for the position and orientation of a link-mounted sensor frame relative to the link frame has been presented.

Using quaternions to manipulate the kinematic equations in terms of rotational matrices provides an alternative way to reformulate the equations in terms of quaternions. Instead of multiplying 3×3 rotational matrices in a sequence, the corresponding quaternions that are expressed by 4×1 column vectors are equated using quaternion multiplications. This simplifies the original equations greatly, so that simple and well-structured linear systems can then be derived.

The general solutions of rotation and translation were both obtained using analysis of the singular-value decomposition. Because the computation of the SVD is costly, a set of closed-form formulae for the SVD has been derived analytically to avoid computing the SVD numerically.

Because the general solutions are not unique, two distinct movements of the robot manipulator have to be made to obtain two sets of data. A simple criterion is derived to select the independent set of simultaneous equations for rotation and translation. Then the equations are solved analytically to obtain a set of closed-form formulae for unique solutions. The selection criterion and the solution formulae can be easily incorporated in application programs that require the calculation of the relative position and orientation of the sensor. This is particularly useful for real-time applications and when different sensor configurations and setups may be required.

Appendix A. Quaternions

A quaternion, $\boldsymbol{\alpha}$, is defined as a complex number, $\boldsymbol{\alpha} = a_0 + a_1\mathbf{i} + a_2\mathbf{j} + a_3\mathbf{k}$, formed from four different units $(1, \mathbf{i}, \mathbf{j}, \mathbf{k})$ by means of the real parameters a_i $(i = 0, 1, 2, 3)$ (Hamilton 1853; Klein 1925; Brand 1957; Bottema and Roth 1979; Wehage 1984). Incorporating ideas from both vectors and matrices, the quaternion $\boldsymbol{\alpha}$ may be viewed as a linear combination of a scalar $\boldsymbol{\alpha}_0$ and a spatial vector $\mathbf{a}$:

$$\boldsymbol{\alpha} = a_0 + \mathbf{a}. \tag{A1}$$

If $a_0 = 0$, $\boldsymbol{\alpha}$ is called a *vector quaternion;* when $\mathbf{a} = \mathbf{0}$, $\boldsymbol{\alpha}$ is a *scalar quaternion.* As we can see, scalars and spatial vectors are quaternions, and they are in the subspace of quaternions. The *conjugate* of a quaternion, denoted by $\boldsymbol{\alpha}^*$, is defined by negating its vector part; that is, $\boldsymbol{\alpha}^* = a_0 - \mathbf{a}$. The matrix (column vector) representation of an arbitrary quaternion is merely the collection of its parameters:

$$\boldsymbol{\alpha} = [a_0, a_1, a_2, a_3]^T = [a_0, \mathbf{a}^T]^T \tag{A2}$$

where "^{T}" indicates the transpose of a matrix.

Because scalars and spatial vectors are in the subspace of quaternions, the rules of scalar and vector algebra also apply to quaternions. Let us consider the following three quaternions: $\boldsymbol{\alpha} = [a_0, \mathbf{a}^T]^T$, $\boldsymbol{\beta} = [b_0, \mathbf{b}^T]^T$, and $\gamma = [c_0, \mathbf{c}^T]^T$. The *addition* and *subtraction,* $\pm$, of two quaternions $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are defined as

$$\boldsymbol{\alpha} \pm \boldsymbol{\beta} = [a_0 \pm b_0, a_1 \pm b_1, a_2 \pm b_2, a_3 \pm b_3]^T. \tag{A3}$$

In terms of matrices, the *Quaternion multiplication,* $\otimes$, can be written as

$$\boldsymbol{\alpha} \otimes \boldsymbol{\beta} = \begin{bmatrix} a_0 b_0 - \mathbf{a}^T\mathbf{b} \\ \mathbf{a}b_0 + (a_0\boldsymbol{I} + \tilde{\mathbf{a}})\mathbf{b} \end{bmatrix} = \begin{bmatrix} b_0 a_0 - \mathbf{b}^T\mathbf{a} \\ \mathbf{b}a_0 + (b_0\boldsymbol{I} - \tilde{\mathbf{b}})\mathbf{a} \end{bmatrix} \tag{A4}$$

where

$$\tilde{\mathbf{a}} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}. \tag{A5}$$

The matrix $\tilde{\mathbf{b}}$ is similar to $\tilde{\mathbf{a}}$, and $\boldsymbol{I}$ is a 3×3 identity matrix.

Letting $\gamma = \boldsymbol{\alpha} \otimes \boldsymbol{\beta}$ and factoring (A4) into a product of two matrices pertaining to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, we get

$$\begin{bmatrix} c_0 \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} a_0 & -\mathbf{a}^T \\ \mathbf{a} & a_0\boldsymbol{I} + \tilde{\mathbf{a}} \end{bmatrix} \begin{bmatrix} b_0 \\ \mathbf{b} \end{bmatrix} = \begin{bmatrix} b_0 & -\mathbf{b}^T \\ \mathbf{b} & b_0\boldsymbol{I} - \tilde{\mathbf{b}} \end{bmatrix} \begin{bmatrix} a_0 \\ \mathbf{a} \end{bmatrix}. \tag{A6}$$

Quaternion multiplication is associative and distributive with respect to addition and subtraction, but the commutative law does *not* hold in general (Brand 1957; Brainerd et al. 1967; Foulis 1969; Bottema and Roth 1979; Wehage 1984). However, from (A6) we can observe that $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ can be commutated simply with a single sign changed. This property is very useful; therefore, two compact notations are designed for the leading matrices (Wehage 1984):

$$\overset{+}{\boldsymbol{\alpha}} = \begin{bmatrix} a_0 & -\mathbf{a}^T \\ \mathbf{a} & a_0\boldsymbol{I} + \tilde{\mathbf{a}} \end{bmatrix}; \quad \bar{\boldsymbol{\beta}} = \begin{bmatrix} b_0 & -\mathbf{b}^T \\ \mathbf{b} & b_0\boldsymbol{I} - \tilde{\mathbf{b}} \end{bmatrix} \tag{A7}$$

where the hats "$^+$" and "$^-$" used in $\overset{+}{\boldsymbol{\alpha}}$ and $\bar{\boldsymbol{\beta}}$ correspond to the "+" and "−" signs attached to the matrices $\tilde{\mathbf{a}}$ and $\tilde{\mathbf{b}}$ in (A6), respectively. Now equation (A6) can be expressed in a compact form as

$$\gamma = \overset{+}{\boldsymbol{\alpha}}\boldsymbol{\beta} = \bar{\boldsymbol{\beta}}\boldsymbol{\alpha}. \tag{A8}$$

The matrices $\overset{+}{\boldsymbol{\alpha}}$ and $\bar{\boldsymbol{\beta}}$ are orthogonal. They are orthonormal matrices if $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are normalized quaternions.

Unlike spatial vectors, the set of quaternions forms a *division algebra* (Brand 1957; Brainerd et al. 1967; Foulis 1969), as for each nonzero quaternion $\boldsymbol{\alpha}$, there is an inverse $\boldsymbol{\alpha}^{-1}$ such that $\boldsymbol{\alpha} \otimes \boldsymbol{\alpha}^{-1} = \boldsymbol{\alpha}^{-1} \otimes \boldsymbol{\alpha} = 1$. Consider two nonzero quaternions $\boldsymbol{\alpha}$ and $\boldsymbol{\beta} = \boldsymbol{\alpha}^*/N(\boldsymbol{\alpha})$, where $N(\boldsymbol{\alpha}) \equiv \boldsymbol{\alpha}^* \otimes \boldsymbol{\alpha} = \alpha \otimes \boldsymbol{\alpha}^*$ is a scalar quaternion and is defined as the *norm* of $\boldsymbol{\alpha}$. Because $\boldsymbol{\alpha} \otimes \boldsymbol{\beta} = (\boldsymbol{\alpha} \otimes \boldsymbol{\alpha}^*)/[N(\boldsymbol{\alpha}) = 1$, we find the *inverse* of $\boldsymbol{\alpha}$ to be

$$\boldsymbol{\alpha}^{-1} = \frac{\boldsymbol{\alpha}^*}{N(\boldsymbol{\alpha})}. \tag{A9}$$

If $\mathrm{N}(\boldsymbol{\alpha}) = 1$, $\boldsymbol{\alpha}$ is called a *unit quaternion;* in this case, the inverse of $\boldsymbol{\alpha}$ is $\boldsymbol{\alpha}^*$.

Euler Parameters and Finite Rotations

Euler parameters, denoted by $\mathbf{p} = [e_0, e_1, e_2, e_3]^T = [e_0, \mathbf{e}^T]^T$, are unit quaternions. They can be expressed in the form (Brand 1957)

$$\mathbf{p}\cos(\theta/2) + \mathbf{u}\sin(\theta/2), \quad 0 \leq \theta \leq 2\pi \tag{A10}$$

where $\cos(\theta/2) = e_0$, $\sin(\theta/2) = \pm\sqrt{\mathbf{e}^T\mathbf{e}}$, and $\mathbf{u} = \pm\mathbf{e}/\sqrt{\mathbf{e}^T\mathbf{e}}$. The vector $\mathbf{u}$ is a unit vector when $\sqrt{\mathbf{e}^T\mathbf{e}}$ is not zero. The Euler parameters are required to satisfy the normality constraint

$$\mathbf{p}^T\mathbf{p} = 1. \tag{A11}$$

Let $\mathbf{p}$ be a unit quaternion and $\boldsymbol{\alpha}$ be an arbitrary quaternion. The operation, $\mathbf{p} \otimes \boldsymbol{\alpha} \otimes \mathbf{p}^*$, transforms $\boldsymbol{\alpha}$ into another quaternion $\boldsymbol{\alpha}'$ without changing its norm (Klein 1925; Brand 1957; Bottema and Roth 1979). Expressing $\boldsymbol{\alpha}' = \mathbf{p} \otimes \boldsymbol{\alpha} \otimes \mathbf{p}^*$ in matrix form, we have

$$\boldsymbol{\alpha}' = \overset{+}{\mathbf{p}}\,\bar{\mathbf{p}}^*\ \boldsymbol{\alpha} = \overset{+}{\mathbf{p}}\,\bar{\mathbf{p}}^T\ \boldsymbol{\alpha} = \boldsymbol{A}\boldsymbol{\alpha} \tag{A12}$$

where $\bar{\mathbf{p}}^* = \bar{\mathbf{p}}^T$ can be verified directly by expanding their entries. The matrix $\boldsymbol{A}$ is a quaternion transformation in 4-space and is a 4×4 orthonormal matrix.[3]

With the aid of quaternion algebra, finite rotations in space may be dealt with in a simple and elegant manner. If $\boldsymbol{\alpha}$ is a vector quaternion, the formula (Hamilton 1853; Cayley 1889)

$$\boldsymbol{\alpha}' = \mathbf{p} \otimes \boldsymbol{\alpha} \otimes \mathbf{p}^* \tag{A13}$$

is, in fact, an alternative statement of the Euler theorem that a general rotation in space can be achieved by a single rotation θ about an axis $\mathbf{u}$. The *rotational transformation matrix* $\boldsymbol{A}$ for a spatial vector can be obtained by taking the lower right 3×3 submatrix of (A12) directly:

$$\boldsymbol{A} = (e_0^2 - \mathbf{e}^T\mathbf{e})\boldsymbol{I} + 2(\mathbf{e}\mathbf{e}^T + e_0\tilde{\mathbf{e}}). \tag{A14}$$

Directly from Euler's theorem, the matrix $\boldsymbol{A}$ can be derived as

$$\boldsymbol{A} = (\cos\theta)\boldsymbol{I} + (1 - \cos\theta)\mathbf{u}\mathbf{u}^T + (\sin\theta)\tilde{\mathbf{u}}. \tag{A15}$$

The matrix $\boldsymbol{A}$ has been given in Rodrigues (1840), Gibbs (1960), Wittenburg (1977), and Suh and Radcliffe (1978) and has been used by many researchers (Paul 1981; Shiu and Ahmad 1987). Defining the quantities as given in (A10): $e_0 = \cos(\theta/2)$, and $\mathbf{e} = \sin(\theta/2)\mathbf{u}$, and applying some trigonometric identities, we can find that (A14) and (A15) are identical.

3. $\boldsymbol{A}\boldsymbol{A}^T = (\overset{+}{\mathbf{p}}\bar{\mathbf{p}}^T)(\overset{+}{\mathbf{p}}\bar{\mathbf{p}}^T)^T = \overset{+}{\mathbf{p}}(\bar{\mathbf{p}}^T\bar{\mathbf{p}})\overset{+}{\mathbf{p}}^T = \boldsymbol{I}$, as $\overset{+}{\mathbf{p}}$ and $\bar{\mathbf{p}}$ are orthonormal. Similarly, it can be shown that $\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}$.

Successive Rotations

Euler parameters are employed to define the relative orientation between any two coordinates systems. Consider a spatial vector $\mathbf{r}$ that is resolved in four coordinate systems $o_0(x_0, y_0, z_0)$, $o_i(x_i, y_i, z_i)$, $o_j(x_j, y_j, z_j)$, and $o_k(x_k, y_k, z_k)$. The projections of $\mathbf{r}$ onto these systems are denoted by $\mathbf{r}^0$, $\mathbf{r}^i$, $\mathbf{r}^j$, and $\mathbf{r}^k$, respectively. A set of Euler parameters is defined for specifying each relative orientation between any two coordinate systems; for example, $\mathbf{p}_{ji}$ specifies the relative orientation of system o_j relative to system o_i.

From (A13), we can relate $\mathbf{r}^0$, $\mathbf{r}^i$, $\mathbf{r}^j$, and $\mathbf{r}^k$ by the following equations:

$$\mathbf{r}^0 = \mathbf{p}_{i0} \otimes \mathbf{r}^i \otimes \mathbf{p}_{i0}^* \tag{A16}$$

$$\mathbf{r}^0 = \mathbf{p}_{j0} \otimes \mathbf{r}^j \otimes \mathbf{p}_{j0}^* \tag{A17}$$

$$\mathbf{r}^0 = \mathbf{p}_{k0} \otimes \mathbf{r}^k \otimes \mathbf{p}_{k0}^* \tag{A18}$$

$$\mathbf{r}^i = \mathbf{p}_{ji} \otimes \mathbf{r}^j \otimes \mathbf{p}_{ji}^* \tag{A19}$$

$$\mathbf{r}^j = \mathbf{p}_{kj} \otimes \mathbf{r}^k \otimes \mathbf{p}_{kj}^* \tag{A20}$$

Substituting (A19) into (A16) and equating with (A17) give

$$\mathbf{p}_{j0} = \mathbf{p}_{i0} \otimes \mathbf{p}_{ji} \tag{A21}$$

Similarly, substituting (A20) into (A17) and equating with (A18) give

$$\mathbf{p}_{k0} = \mathbf{p}_{j0} \otimes \mathbf{p}_{kj} \tag{A22}$$

Now, substituting (A21) into (A22), we get

$$\mathbf{p}_{k0} = \mathbf{p}_{i0} \otimes \mathbf{p}_{ji} \otimes \mathbf{p}_{kj}. \tag{A23}$$

If we express eqs. (A16)–(A20) in matrix form and follow the same substitution procedures, we can also find

$$\boldsymbol{A}_{j0} = \boldsymbol{A}_{i0}\boldsymbol{A}_{ji} \tag{A24}$$

$$\boldsymbol{A}_{k0} = \boldsymbol{A}_{j0}\boldsymbol{A}_{kj} \tag{A25}$$

$$\boldsymbol{A}_{k0} = \boldsymbol{A}_{i0}\boldsymbol{A}_{ji}\boldsymbol{A}_{kj} \tag{A26}$$

Equations (A21)–(A26) give a set of recursive formulae that are useful in the development of the kinematics of a rigid-body chain such as a robotic manipulator. From these equations, we can observe that the equations of successive rotation matrices can be expressed alternatively in terms of quaternion multiplications in a similar sequence.

Appendix B. Singular Values and Vectors of B

The singular values of B can be derived by solving the characteristic polynomial $det(B^TB - \lambda I) = 0$ where λ is the eigenvalue of B^TB. Because B is a 4×4 matrix, the characteristic equation has the form

$$\lambda^4 + K_3\lambda^3 + K_2\lambda^2 + K_1\lambda + K_0 = 0 \qquad \text{(B1)}$$

where $K_3 = -8$, $K_2 = 16$, and $K_1 = K_0 = 0$. The quantities $K_i (i = 0, 1, 2, 3)$ can be obtained using the property: $\mathbf{u}_l^T\mathbf{u}_l = \mathbf{u}_c^T\mathbf{u}_c = 1$. The roots of equation (B1) are

$$\lambda = (4, 4, 0, 0). \qquad \text{(B2)}$$

The singular values of B are

$$\sigma(B) = (2, 2, 0, 0), \qquad \text{(B3)}$$

as they are the positive square roots of the eigenvalues of B^TB.

Consider the matrix B that has been decomposed into $B = U\Sigma V^T$. Partitioning U, V, and Σ according to the nonzero and zero singular values of B, we have

$$B = [U_1 \quad U_2]\begin{bmatrix} D & 0 \\ \mathbf{0} & \mathbf{0}\end{bmatrix}\begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}$$

where U and V have a collection of eigenvectors of BB^T and B^TB, respectively. Because B is a simple and well-structured matrix, the eigenvectors of BB^T and B^TB can be derived by solving $(BB^T - \lambda I)\mathbf{X} = \mathbf{0}$ and $(B^TB - \lambda I)\mathbf{X} = \mathbf{0}$ directly. For U_1, U_2, V_1, and V_2, we solve the following four equations: $(BB^T - 4I)\mathbf{X} = \mathbf{0}$, $(BB^T)\mathbf{X} = \mathbf{0}$, $(B^TB - 4I)\mathbf{X} = \mathbf{0}$, and $(B^TB)\mathbf{X} = \mathbf{0}$, respectively. Because B, BB^T, and B^TB are rank deficient and $B^TB = BB^T$, we can obtain two eigenspaces for U_1, U_2, V_1, and V_2 as follows:

$$\mathbf{X}_{U_1} = \mathbf{X}_{V_1} = \begin{bmatrix} (u_{l_z} - u_{c_z}) & -(u_{l_y} - u_{c_y}) \\ -(u_{l_y} + u_{c_y}) & -(u_{l_z} + u_{c_z}) \\ (u_{l_x} + u_{c_x}) & 0 \\ 0 & (u_{l_x} + u_{c_x}) \end{bmatrix}\begin{bmatrix} s \\ t \end{bmatrix} \qquad \text{(B4)}$$

$$\mathbf{X}_{U_2} = \mathbf{X}_{V_2} = \begin{bmatrix} (u_{l_z} - u_{c_z}) & -(u_{l_y} - u_{c_y}) \\ -(u_{l_y} + u_{c_y}) & -(u_{l_z} + u_{c_z}) \\ (u_{l_x} + u_{c_x}) & 0 \\ 0 & (u_{l_x} + u_{c_x}) \end{bmatrix}\begin{bmatrix} s \\ t \end{bmatrix} \qquad \text{(B5)}$$

where s and t are arbitrary. There are many choices for selecting a proper set of orthonormal eigenvectors for our purposes. One of them which is simple is given as follows:

$$U_1 = \left(\frac{1}{m_1}\right)\begin{bmatrix} 0 & -\mathbf{u}_l^T(\mathbf{u}_l - \mathbf{u}_c) \\ (\mathbf{u}_l - \mathbf{u}_c) & -\tilde{\mathbf{u}}_l\mathbf{u}_c \end{bmatrix}; \quad m_1 = \sqrt{2 - 2\mathbf{u}_l^T\mathbf{u}_c} \qquad \text{(B6)}$$

$$U_2 = \left(\frac{1}{m_2}\right)\begin{bmatrix} 0 & -\mathbf{u}_l^T(\mathbf{u}_l + \mathbf{u}_c) \\ (\mathbf{u}_l + \mathbf{u}_c) & \tilde{\mathbf{u}}_l\mathbf{u}_c \end{bmatrix}; \quad m_2 = \sqrt{2 + 2\mathbf{u}_l^T\mathbf{u}_c} \qquad \text{(B7)}$$

$$V_1 = \left(\frac{1}{m_1}\right)\begin{bmatrix} \mathbf{u}_l^T(\mathbf{u}_l - \mathbf{u}_c) & 0 \\ \tilde{\mathbf{u}}_l\mathbf{u}_c & (\mathbf{u}_l - \mathbf{u}_c) \end{bmatrix} \qquad \text{(B8)}$$

$$V_2 = \left(\frac{1}{m_2}\right)\begin{bmatrix} -\mathbf{u}_l^T(\mathbf{u}_l + \mathbf{u}_c) & 0 \\ \tilde{\mathbf{u}}_l\mathbf{u}_c & (\mathbf{u}_l + \mathbf{u}_c) \end{bmatrix} \qquad \text{(B9)}$$

where $U_1^TU_1 = U_2^TU_2 = V_1^TV_1 = V_2^TV_2 = I$, $U_1^TU_2 = V_1^TV_2 = \mathbf{0}$, and $U_1DV_1^T = B$ can be verified easily.

To solve our problem, V_2 is of interest. There are many V_2s that can be chosen from the eigenspace $\mathbf{X}_{V_2}$; each of them can be used to solve the problem as long as its two column vectors are in the eigenspace and mutually orthogonal. The particular one chosen in this article is (B9).

For the particular case when $\mathbf{u}_l = -\mathbf{u}_c = \mathbf{u} = [u_x, u_y, u_z]^T$, (B9) cannot be used. The eigenspace $\mathbf{X}_{V_2}$ becomes

$$\mathbf{X}_{V_2} = \begin{bmatrix} 0 & 0 \\ -u_y & -u_z \\ u_x & 0 \\ 0 & u_x \end{bmatrix}\begin{bmatrix} s \\ t \end{bmatrix}. \qquad \text{(B10)}$$

We can choose a set of particular eigenvectors

$$V_2 = \left(\frac{1}{m_3}\right)\begin{bmatrix} 0 & 0 \\ -(u_y + u_z) & u_x(u_y - u_z) \\ u_x & u_y(u_y - u_z) - 1 \\ u_x & u_z(u_y - u_z) + 1 \end{bmatrix};$$

$$m_3 = \sqrt{2 - (u_y - u_z)^2} \qquad \text{(B11)}$$

such that the two singular vectors in V_2 are mutually orthogonal and will never be zero.

Appendix C. Singular Values and Vectors of $\tilde{\mathbf{u}}$

The singular values and vectors of $\tilde{\mathbf{u}}$ must be known when solving the equation of translation. Solving the characteristic polynomial $det(\tilde{\mathbf{u}}^T\tilde{\mathbf{u}} - \lambda I) = 0$, we get

$$\lambda^3 + K_2\lambda^2 + K_1\lambda + K_0 = 0 \qquad \text{(C1)}$$

where $K_2 = -2\mathbf{u}^T\mathbf{u} = -2$, $K_1 = (\mathbf{u}^T\mathbf{u})^2 = 1$, and $K_0 = 0$, as $\mathbf{u}^T\mathbf{u} = 1$. The roots of (C1) are

$$\lambda = (1, 1, 0). \qquad \text{(C2)}$$

The singular values of $\tilde{\mathbf{u}}$ are $+\sqrt{\lambda_i}$ $(i = 1, 2, 3)$; therefore

$$\sigma(\tilde{\mathbf{u}}) = (1, 1, 0). \qquad \text{(C3)}$$

The right singular vectors of V_2 that span the null-space of $\tilde{\mathbf{u}}$ are of interest. They can be obtained by solving $(\tilde{\mathbf{u}}^T\tilde{\mathbf{u}})\mathbf{X} = \mathbf{0}$ directly. The solution gives an eigenspace:

$$\mathbf{X}_{V_2} = \mathbf{u}t. \qquad \text{(C4)}$$

As we can see, the simplest eigenvector in (C4) that we can have is

$$V_2 = \boldsymbol{u}. \tag{C5}$$

Acknowledgments

This research was supported by the Pattern Analysis and Machine Intelligence Laboratory at the Department of Systems Design Engineering, University of Waterloo, and was funded by the Natural Science and Engineering Research Council of Canada. We are indebted to Mr. Gregory Aicklen of the Erik Jonsson School of Engineering and Computer Science at the University of Texas at Dallas for his work in creating the robot arm diagram.

References

Bottema, O., and Roth, B. 1979. *Theoretical Kinematics.* Amsterdam: North-Holland.

Bradley, G. L. 1975. *A Primer of Linear Algebra.* Englewood Cliffs, N.J.: Prentice-Hall.

Brainerd, B., Clarke, M. J., Leibovitz, R. A., et al. 1967. *Vector and Matrices.* Toronto: The Ryerson Press.

Brand, L. 1957. *Vector and Tensor Analysis.* New York: John Wiley and Sons.

Cayley, A. 1885a. *On the matrical equation $qQ - Qq' = 0$. Messenger of Mathematics* XIV:176–178.

Cayley, A. 1885b. On the quaternion equation $qQ - Qq' = 0$. *Messenger of Mathematics* XIV:108–112.

Cayley, A. 1889. On certain results relating to quaternions. *The Collected Mathematical Papers of Arthur Cayley.* England Cambridge Univ. Press, vol. 1. pp. 123–126.

Chelnokov, Yu. N. 1984. Use of quaternions in the precessional theory of gyroscopes. *Izv. An SSSR. Mekhanika Tverdogo Tela.* 19(6):3–10.

Chou, J. C. K., Baciu, G., and Kesavan, H. K. 1987a (March). Graph-theoretic models for simulating robot manipulators. *Proc. 1987 IEEE International Conference on Robotics and Automation.* Raleigh, N.C.: IEEE, 2:953–959.

Chou, J. C. K., Baciu, G., and Kesavan, H. K. 1987b (March). Computational scheme for simulating robot manipulators. *Proc. 1987 IEEE International Conference on Robotics and Automation.* Raleigh, N.C.: IEEE, 2:961–966.

Chou, Jack C. K., and Kamel, M. 1988 (April). Quaternions approach to solve the kinematic equation of rotation, $A_a A_x = A_x A_b$, of a sensor-mounted robotic manipulator. *Proc. 1988 IEEE International Conference on Robotics and Automation.* Philadelphia, P.A.: IEEE, 2:656–662.

Chou, J. C. K., Kesavan, H. K., and Singhal, K. 1986a. Dynamics of 3-D isolated rigid-body systems: graph-theoretic models. *Mechanism Machine Theory* 21(3):261–272.

Chou, Jack C. K., Kesavan, H. K., and Singhal, Kishore. 1986b. A systems approach to 3-D multi-body systems using graph-theoretic models. *IEEE Trans. Sys. Man. Cybernet.* SMC-16(2):219–230.

Chou, J. C. K., Singhal, K., and Kesavan, H. K. 1986. Multi-body systems with open chains: Graph-theoretic model. *Mechanism Machine Theory.* 21(3):273–284.

Craig, J. J. 1986. *Introduction to Robotics—Mechanics and Control.* Reading, Mass.: Addison-Wesley.

Foulis, D. J. 1969. *Fundamental Concepts of Mathematics.* Prindle. Boston, Mass.: Weber and Schmidt.

Fu, K. S., Gonzalez, R. C., and Lee, C. S. G. 1987. *Robotics: Control, Sensing, Vision, and Intelligence.* New York: McGraw-Hill.

Gantmacher, F. R. 1959. *The Theory of Matrices, Volume One.* New York: Chelsea Publishing Company.

Gibbs, J. W. 1960. *Vector Analysis.* New York: Dover.

Goldstein, H. 1981 (July). *Classical Mechanics,* 2nd ed. Reading, Mass.: Addison-Wesley.

Golub, G., and Van Loan, C. 1983. *Matrix Computations.* Baltimore: The Johns Hopkins University Press.

Hamilton, W. R. 1853. *Lectures on Quaternions.* Dublin: Hodges and Smith.

Hendley, A. C. 1971 (February). Quaternions for control of space vehicles. *Proc. the Institute of Navigation, National Space Meeting on Space Shuttle-Space Station-Nuclear Shuttle Navigation.* Huntsville, Ala.: The Institute of Navigation.

Horn, Berthold K. P. 1986. *Robot Vision.* New York: McGraw-Hill.

Horn, Berthold K. P. 1987. Closed-form Solution of Absolute Orientation Using Unit Quaternions. *J. Optical Soc. Amer.* 4(4):629–642.

Ickes, B. P. 1970. A new method for performing digital control system amplitude computations using quaternions. *AIAA J.* 8(1):13–17.

Klein, F. 1925. *Elementary Mathematics From the Advanced Standpoint.* New York: Dover.

Klumpp, A. R. 1976. Singularity-free extraction of a quaternion from a direction-cosine matrix. *J. Spacecraft Rockets.* 13(12):754–755.

Koshlyakov, V. N. 1983. Equations of a heavy rigid body that rotates about a fixed point in the Rodrigues-Hamilton parameters. *Izv. An SSSR. Mekhanika Tverdogo Tela.* 18(4):16–25.

Nikravesh, P. E. 1984. Spatial kinematic and dynamic analysis with Euler parameters. In Haug, E. J. (ed.): *Computer Aided Analysis and Optimization of Mechanical System Dynamics.* Heidelberg: Springer-Verlag, pp. 261–281.

Nikravesh, P. E., and Chung, I. S. 1982. Applications of Euler parameters to the dynamic analysis of three-dimensional constrained mechanical systems. *ASME J. Mech. Des.* 104(4):785–791.

Paul, R. P. 1981. *Robot Manipulators: Mathematics, Programming, and Control.* Cambridge, Mass.: The MIT Press.

Robinson, A. C. 1958. On the use of quaternions in simulation of rigid-body motion. Technical report 58-17. Dayton, Ohio: Aeronautical Research Lab., Wright Air Development Center.

Rodrigues, O. 1840. Des lois géometriques qui régissent les déplacements d'un système solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire. *J. Mathematiques Pures Appl.* 5:380–440.

Sandor, G. N. 1968. Principles of general quaternion-operator method of spatial kinematic synthesis. *ASME J. Appl. Mechanics* Series E, 35(1):40–46.

Shiu, Y. C., and Ahmad, S. 1987 (March). Finding the mounting position of a sensor by solving a homogeneous transform equation of the form $\boldsymbol{AX} = \boldsymbol{XB}$. *Proc. 1987 IEEE International Conference on Robotics and Automation,* Vol. 3. Raleigh, N.C.: IEEE, pp. 1667–1671.

Stewart, G. W. 1973. *Introduction to Matrix Computations.* New York: Academic Press.

Suh, C. H., and Radcliffe, C. W. 1978. *Kinematics and Mechanisms Design.* New York: John Wiley and Sons.

Wang, L. T., and Ravani, B. 1985. Recursive computations of kinematic and dynamic equations for mechanical manipulators. *IEEE J. Robot. Automat.* RA-1(3):124–131.

Wehage, R. A. 1984. *Quaternions and Euler Parameters—A Brief Exposition.* Computer Aided Analysis and Optimization of Mechanical System Dynamics. E. J. Haug, ed. Heidelberg: Springer-Verlag, 147–180.

Wittenburg, J. 1977. *Dynamics of Systems of Rigid Bodies.* Stuttgart: B. G. Teubner.

Yang, A. T., and Freudenstein, F. 1964. Application of dual-number quaternion algebra to the analysis of spatial mechanisms. *ASME J. Appl. Mechanics* Series E, 31(2):300–308.