QUATERNIONS APPROACH TO SOLVE THE KINEMATIC EQUATION OF ROTATION, $A_a A_x = A_x A_b$, OF A SENSOR-MOUNTED ROBOTIC MANIPULATOR

Jack C. K. Chou and 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

Abstract— *The problem of finding the relative orientation between the reference frames of a link-mounted sensor and the link is formulated as a kinematic equation of the form $A_a A_x = A_x A_b$, which has to be solved for the rotational transformation matrix A_x given the transformations A_a and A_b. This equation can be transformed to its equivalent form in terms of quaternions and then simplified to a well-structured linear system of equations of the form $B\mathbf{x} = \mathbf{0}$. Since B is rank-deficient the solution is not unique. The generalized-inverse method using singular-value decomposition (SVD) is applied. Although the solution is reached using the analysis of SVD, the SVD is derived symbolically; therefore, the actual implementation of SVD is not required. A method for obtaining a unique solution is proposed where a system of nonlinear equations is solved using Newton-Raphson iteration. The iteration is simplified by a dimension reduction technique that provides a set of closed-form formulae for solving the resulting linear system of equations.*

1 INTRODUCTION

When a sensor is mounted on the link of a robot arm, the relative position and orientation between the coordinate systems of the sensor and the link have to be known. A direct measurement of relative position for an accurate result is not feasible, because the path of measurement 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, since two coordinate systems may be oriented arbitrarily, and the directions of the Cartesian coordinate axes may not be identified physically.

A mathematical approach was proposed by Shiu and Ahmad [1], and the problem was formulated as an equation of the form $H_a H_x = H_x H_b$, where H_a, H_b, and H_x are 4×4 homogeneous transformation matrices. The homogeneous transformation, H_a, describes an arbitrary movement, as shown in Figure 1, of the link relative to the link frame. The transformation, H_b, specifies the similar generalized displacement of the same motion of the sensor relative to the coordinate frame of the sensor, and H_x is the relative transformation between the sensor frame and the link frame. Equating the homogeneous transformations in the middle loop of Figure 2, we obtain the equation $H_a H_x = H_x H_b$ which has to be solved for H_x given H_a and H_b.

The general solution of a matrix equation of the form $AX = XB$, where A and B are square matrices, and X is a rectangular matrix, has been discussed by Gantmacher [2]. However, the direct application of Gantmacher's method to the solution of $H_a H_x = H_x H_b$ is not appropriate, since the matrices H_a, H_b, and H_x possess special geometrical structures and have well-known physical meanings. A simpler method which exploits the characteristics of these matrices is desired.

Due to the special structures of the matrices H's, the equation, $H_a H_x = H_x H_b$, can be divided into two separate sets of equations. One is for rotation, and the other is for translation. In this paper, we are concerned with the solution of the equation of rotation, $A_a A_x = A_x A_b$, where A's are rotational transformation matrices. This equation has to be solved for A_x given A_a and A_b. This kinematic equation of rotation can be transformed to its equivalent form in terms of quaternions in order to reduce it to a simple and well-structured linear system of equations of the form $B\mathbf{x} = \mathbf{0}$. Using quaternions to manipulate the equation of rotations provides a new approach to re-formulating the underlying problem.

Using Euler parameters to define relative orientations between coordinate systems can be traced back to Hamilton [3] and Cayley [4,5]. It did not become popular until the recent applications to rigid-body dynamics [6,7,8,9,10,11], control [12], and robot dynamics simulation [13,14].

Since the coefficient matrices B is rank-deficient, the generalized inverse method incorporating singular value decomposition (SVD) is applied. Although the analysis is based on SVD, the final solution formulae are in closed form expressed in terms of the given physical quantities. The actual implementation of SVD is avoided.

In order to obtain a unique solution, two distinct robot movements have to be made to obtain two sets of data. This will produce a system of equations in which the number of equations is greater than the number of unknowns. It was proved by Shiu and Ahmad that there are always a sufficient set of equations for a unique solution. However, finding these equations is not obvious. A simple criterion is presented in this paper to select the right set of simultaneous equations. The formulation requires the solution of a set of four non-linear equations. Newton-Raphson iteration is used and a set of closed-form formulae is developed to reduce the computations during each iteration.

2 QUATERNIONS AND FINITE ROTATIONS

In 1843, the Irish mathematician W. R. Hamilton invented quaternions in order to extend 3-dimensional vector algebra to include multiplication and division. Although it has been found that ordinary vector algebra, together with the vector and scalar products, provides a better mathematical apparatus for investigating physical problems, quaternion algebra provides us with a simple, unique, and elegant representation for describing finite rotations in space.

2.1 QUATERNIONS

A quaternion, $\boldsymbol{\alpha}$, is defined as a complex number

$$\boldsymbol{\alpha} = a_0 + a_1\mathbf{i} + a_2\mathbf{j} + a_3\mathbf{k} \tag{1}$$

formed from four different units (1, $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$) by means of the real parameters $a_i (i = 0, 1, 2, 3)$ [3,15,16,17,18,19,20]. In a modernized guise incorporating vector nad matrix, the quaternion $\boldsymbol{\alpha}$ may be viewed as a linear combination of a scalar a_0 and a spatial vector $\mathbf{a}$:

$$\boldsymbol{\alpha} = a_0 + \mathbf{a} \tag{2}$$

The *conjugate* of a quaternion $\boldsymbol{\alpha}$, denoted by $\boldsymbol{\alpha}^*$, is defined by negating its vector part; that is,

$$\alpha^{*} = a_0 - \mathbf{a} \tag{3}$$

If $a_0 = 0$, α is called a *vector quaternion*; when $\mathbf{a} = \mathbf{0}$, α is a *scalar quaternion*. As we can observe, scalars and spatial vectors are quaternions, and they are in the subspace of quaternions.

It is convenient to represent quaternions and their algebra in matrix form to simplify equation manipulations. The matrix (column vector) representation of an arbitrary quaternion α is merely the collection of its parameters:

$$\alpha = [a_0, a_1, a_2, a_3]^T = [a_0, \mathbf{a}^T]^T \tag{4}$$

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

Since 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:

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

$$\beta = [b_0, b_1, b_2, b_3]^T = [b_0, \mathbf{b}^T]^T$$

$$\gamma = [c_0, c_1, c_2, c_3]^T = [c_0, \mathbf{c}^T]^T$$

The *addition* and *subtraction*, $\pm$, of two quaternions α and β are defined as

$$\alpha \pm \beta = [a_0 \pm b_0, a_1 \pm b_1, a_2 \pm b_2, a_3 \pm b_3]^T \tag{5}$$

The addition and subtraction are associative and commutative.

The *Quaternion multiplication*, $\otimes$, is defined in the form of complex numbers as

$$\begin{aligned} \alpha \otimes \beta &= (a_0 + \mathbf{a}) \otimes (b_0 + \mathbf{b}) \\ &= a_0 \otimes b_0 + a_0 \otimes \mathbf{b} + b_0 \otimes \mathbf{a} + \mathbf{a} \otimes \mathbf{b} \end{aligned} \tag{6}$$

where the scalar and scalar-vector products are well defined, and the vector product will be defined as

$$\mathbf{a} \otimes \mathbf{b} = -\mathbf{a} \cdot \mathbf{b} + \mathbf{a} \times \mathbf{b} \tag{7}$$

In terms of matrices, we first define the vector dot-product ($\mathbf{a}\ \cdot$) as a row vector $\mathbf{a}^T$ and the vector cross-product ($\mathbf{a}\ \times$) as a skew-symmetric matrix $\tilde{\mathbf{a}}$; then we separate the scalar and the vector parts in (6) and rewrite it as

$$\alpha \otimes \beta = \begin{bmatrix} a_0 b_0 - \mathbf{a}^T \mathbf{b} \\ \mathbf{a} b_0 + (a_0 \boldsymbol{I} + \tilde{\mathbf{a}})\mathbf{b} \end{bmatrix} = \begin{bmatrix} b_0 a_0 - \mathbf{b}^T \mathbf{a} \\ \mathbf{b} a_0 + (b_0 \boldsymbol{I} - \tilde{\mathbf{b}})\mathbf{a} \end{bmatrix} \tag{8}$$

where

$$\tilde{\mathbf{a}} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{9}$$

The matrix $\tilde{\mathbf{b}}$ is similar to $\tilde{\mathbf{a}}$, and $\boldsymbol{I}$ is a 3×3 identity matrix. Letting $\gamma = \alpha \otimes \beta$ and factoring (8) into a product of two matrices pertaining to α and β, we get

$$\begin{bmatrix} c_0 \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} a_0 & -\mathbf{a}^T \\ \mathbf{a} & a_0 \boldsymbol{I} + \tilde{\mathbf{a}} \end{bmatrix} \begin{bmatrix} b_0 \\ \mathbf{b} \end{bmatrix} = \begin{bmatrix} b_0 & -\mathbf{b}^T \\ \mathbf{b} & b_0 \boldsymbol{I} - \tilde{\mathbf{b}} \end{bmatrix} \begin{bmatrix} a_0 \\ \mathbf{a} \end{bmatrix} \tag{10}$$

Quaternion multiplication is associative and distributive with respect to addition and subtraction, but the commutative law does *not* hold in general [16,17,18,19,20]. However, from (10) we can observe that α and β can be commutated simply with a single sign changed. This property is very useful; therefore, two compact notations are designed for the leading matrices [20]:

$$\overset{+}{\alpha} = \begin{bmatrix} a_0 & -\mathbf{a}^T \\ \mathbf{a} & a_0 \boldsymbol{I} + \tilde{\mathbf{a}} \end{bmatrix} \tag{11}$$

$$\overset{-}{\beta} = \begin{bmatrix} b_0 & -\mathbf{b}^T \\ \mathbf{b} & b_0 \boldsymbol{I} - \tilde{\mathbf{b}} \end{bmatrix} \tag{12}$$

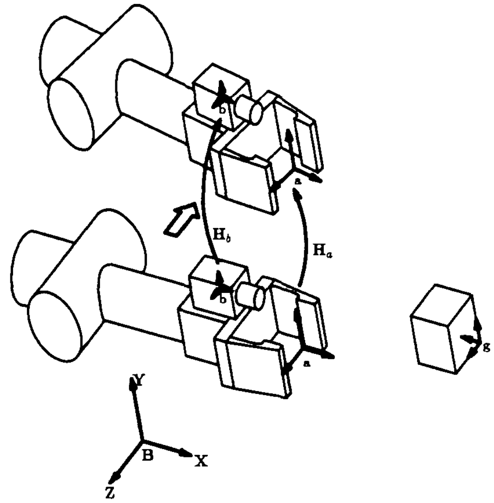

Figure 1: A Movement of the Hand of A Robot Arm.

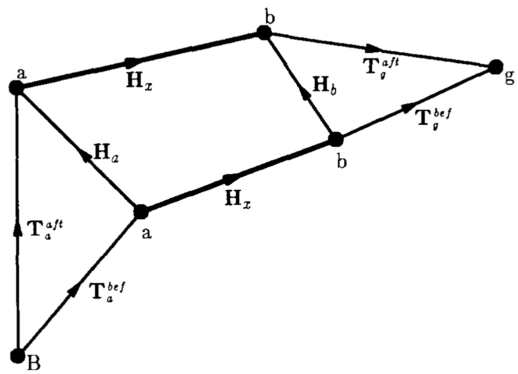

Figure 2: Relative Coordinate Transformations Represented by A Graph.

where the hats "+" and "−" used in $\overset{+}{\alpha}$ and $\overset{-}{\beta}$ correspond to the "+" and "−" signs attached to the matrices $\tilde{\mathbf{a}}$ and $\tilde{\mathbf{b}}$ in (10), respectively. Now, equation (10) can be expressed in a compact form as

$$\gamma = \overset{+}{\alpha}\, \beta = \overset{-}{\beta}\, \alpha \tag{13}$$

Contrary to spatial vectors, the set of quaternions forms a *division algebra* [16,17,18]; since for each non-zero quaternion α, there is an inverse α^{-1} such that $\alpha \otimes \alpha^{-1} = \alpha^{-1} \otimes \alpha = 1$. Consider two non-zero quaternions α and $\beta = \frac{\alpha^{*}}{N(\alpha)}$, where

$$N(\alpha) \equiv \alpha^{*} \otimes \alpha = \alpha \otimes \alpha^{*} \tag{14}$$

is a scalar quaternion and is defined as the *norm* of α. Since $\alpha \otimes \beta = \frac{\alpha \otimes \alpha^{*}}{N(\alpha)} = 1$, we find the *inverse* of α to be

$$\alpha^{-1} = \frac{\alpha^{*}}{N(\alpha)} \tag{15}$$

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

2.2 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 [16]

$$\mathbf{p} = \cos(\theta/2) + \mathbf{u}\sin(\theta/2), \quad 0 \leq \theta \leq 2\pi \tag{16}$$

where $\cos(\theta/2) = e_0$, $\sin(\theta/2) = \pm\sqrt{\mathbf{e}^T\mathbf{e}}$, and $\mathbf{u} = \pm\frac{\mathbf{e}}{\sqrt{\mathbf{e}^T\mathbf{e}}}$. 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

$$\mathbf{p}^T\mathbf{p} = 1 \tag{17}$$

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 [15,16,19]. Expressing $\boldsymbol{\alpha}' = \mathbf{p} \otimes \boldsymbol{\alpha} \otimes \mathbf{p}^*$ in matrix form, we have

$$\begin{aligned} \boldsymbol{\alpha}' &= \overset{+}{\mathbf{p}}\overset{-}{\mathbf{p}}^{*}\boldsymbol{\alpha} \\ &= \overset{+}{\mathbf{p}}\overset{-}{\mathbf{p}}^{T}\boldsymbol{\alpha} \\ &= \boldsymbol{A}\boldsymbol{\alpha} \end{aligned} \tag{18}$$

where $\overset{-}{\mathbf{p}}^{*} = \overset{-}{\mathbf{p}}^{T}$ can be verified directly by expanding their entries. The matrix $\boldsymbol{A}$ is a 4×4 quaternion transformation in 4-space. Since the transformations $\overset{+}{\mathbf{p}}$ and $\overset{-}{\mathbf{p}}^{T}$ are orthonormal, the norms of $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}'$ are identical. Also, from (18) we can observe that the scalar part of $\boldsymbol{\alpha}$ after transformation is not changed, and the vector part of $\boldsymbol{\alpha}'$ is rotated by a transformation in terms of a unit axis $\mathbf{u}$ and an angle of θ. Therefore, the transformed quaternion is not stretched, and the transformation of the scalar and vector parts is independent.

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 [3,21]

$$\boldsymbol{\alpha}' = \mathbf{p} \otimes \boldsymbol{\alpha} \otimes \mathbf{p}^* \tag{19}$$

is, in fact, an alternative statement of 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 sub-matrix of (18) directly:

$$\boldsymbol{A} = (e_0^2 - \mathbf{e}^T\mathbf{e})\boldsymbol{I} - 2(\mathbf{e}\mathbf{e}^T + e_0\tilde{\mathbf{e}}) \tag{20}$$

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

$$\boldsymbol{A} = (\cos\theta)\boldsymbol{I} + (1 - \cos\theta)\mathbf{u}\mathbf{u}^T + (\sin\theta)\tilde{\mathbf{u}} \tag{21}$$

The matrix $\boldsymbol{A}$ has been given in [22,23,24,25,26], and has been used by many researchers such as in [1,27]. Defining the quantities as given in (16)

$$e_0 = \cos(\theta/2), \qquad \mathbf{e} = \sin(\theta/2)\mathbf{u} \tag{22}$$

and applying some trigonometric identities, we can prove that (20) and (21) are identical.

2.3 SUCCESSIVE ROTATIONS

Euler parameters are employed to define the relative orientation between any two coordinates systems. Consider a spatial vector $\mathbf{r}$ which 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 (19), we can related $\mathbf{r}^0$, $\mathbf{r}^i$, $\mathbf{r}^j$, and $\mathbf{r}^k$ by the following equations:

$$\mathbf{r}^0 = \mathbf{p}_{i0} \otimes \mathbf{r}^i \otimes \mathbf{p}_{i0}^* \tag{23}$$

$$\mathbf{r}^0 = \mathbf{p}_{j0} \otimes \mathbf{r}^j \otimes \mathbf{p}_{j0}^* \tag{24}$$

$$\mathbf{r}^0 = \mathbf{p}_{k0} \otimes \mathbf{r}^k \otimes \mathbf{p}_{k0}^* \tag{25}$$

$$\mathbf{r}^i = \mathbf{p}_{ji} \otimes \mathbf{r}^j \otimes \mathbf{p}_{ji}^* \tag{26}$$

$$\mathbf{r}^j = \mathbf{p}_{kj} \otimes \mathbf{r}^k \otimes \mathbf{p}_{kj}^* \tag{27}$$

Substituting (26) into (23)

$$\mathbf{r}^0 = (\mathbf{p}_{i0} \otimes \mathbf{p}_{ji}) \otimes \mathbf{r}^j \otimes (\mathbf{p}_{ji}^* \otimes \mathbf{p}_{i0}^*)$$

and equating with (24) give

$$\mathbf{p}_{j0} = \mathbf{p}_{i0} \otimes \mathbf{p}_{ji} \tag{28}$$

Similarly, substituting (27) into (24) and equating with (25) give

$$\mathbf{p}_{k0} = \mathbf{p}_{j0} \otimes \mathbf{p}_{kj} \tag{29}$$

Now, substituting (28) into (29), we get

$$\mathbf{p}_{k0} = \mathbf{p}_{i0} \otimes \mathbf{p}_{ji} \otimes \mathbf{p}_{kj} \tag{30}$$

If we express Eqs. (23—27) in matrix form and follow the same substitution procedures, we can also find

$$\boldsymbol{A}_{j0} = \boldsymbol{A}_{i0}\boldsymbol{A}_{ji} \tag{31}$$

$$\boldsymbol{A}_{k0} = \boldsymbol{A}_{j0}\boldsymbol{A}_{kj} \tag{32}$$

$$\boldsymbol{A}_{k0} = \boldsymbol{A}_{i0}\boldsymbol{A}_{ji}\boldsymbol{A}_{kj} \tag{33}$$

Equations (28—33) give a set of recursive formulae which are useful in the development of the kinematics of a rigid-body chain such as a robot manipulator.

3 THE PHYSICAL PROBLEM

As mentioned, the problem of finding the relative position and orientation between a link-mounted sensor and the link is formulated mathematically as

$$\boldsymbol{H}_a\boldsymbol{H}_x = \boldsymbol{H}_x\boldsymbol{H}_b \tag{34}$$

When the robot moves, the relative transformation $\boldsymbol{H}_a$ which describes the relative position and orientation *after* and *before* the motion is given by the equation

$$\boldsymbol{T}_a^{aft} = \boldsymbol{T}_a^{bef}\boldsymbol{H}_a$$

or,

$$\boldsymbol{H}_a = (\boldsymbol{T}_a^{bef})^{-1}(\boldsymbol{T}_a^{aft}) \tag{35}$$

where $\boldsymbol{T}_a$ 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. With the same movement, the transformation $\boldsymbol{H}_b$ which describes the relative position and orientation of the sensor *after* and *before* the motion is given by the equation

$$\boldsymbol{T}_g^{bef} = \boldsymbol{H}_b\boldsymbol{T}_g^{aft}$$

or,

$$\boldsymbol{H}_b = (\boldsymbol{T}_g^{bef})(\boldsymbol{T}_g^{aft})^{-1} \tag{36}$$

where $\boldsymbol{T}_g$ specifies the relative position and orientation of an object in the goal frame [28] with respect to the sensor frame. The transformation $\boldsymbol{T}_g$ is given by any sensor capable of finding the three-dimensional position and orientation of an object.

Giving $\boldsymbol{H}_a$ and $\boldsymbol{H}_b$, we can solve (34) for the relative position and orientation, $\boldsymbol{H}_x$, between the sensor and the link.

4 SOLVING THE EQUATION OF ROTATION

Expanding (34) and equating the entries at both sides, we can obtain the equation:

$$\boldsymbol{A}_a \boldsymbol{A}_x = \boldsymbol{A}_x \boldsymbol{A}_b \tag{37}$$

This equation involves only rotational transformation matrices and is referred to as the *equation of rotation*, which has to be solved for $\boldsymbol{A}_x$ given $\boldsymbol{A}_a$ and $\boldsymbol{A}_b$.

Using Euler parameters (normalized quaternion) to define the relative orientation between two coordinate systems provides a simple and elegant manner to formulate successive rotations. As mentioned in Section 2, a sequence of rotations can be formulated as an equation without involving rotational transformation matrices. As a result, the problem of solving $\boldsymbol{A}_a\boldsymbol{A}_x = \boldsymbol{A}_x\boldsymbol{A}_b$ can be transformed to an equivalent problem involving the corresponding Euler parameters as follows:

$$\mathbf{p}_a \otimes \mathbf{p}_x = \mathbf{p}_x \otimes \mathbf{p}_b \tag{38}$$

where $\mathbf{p}_a, \mathbf{p}_b$, and $\mathbf{p}_x$ are Euler parameters which specify the rotational transformation matrices $\boldsymbol{A}_a$, $\boldsymbol{A}_b$, and $\boldsymbol{A}_x$, respectively, and the operation $\otimes$ defines the quaternion multiplication. Applying (13), we can further express (38) in matrix form as

$$\overset{+}{\mathbf{p}}_a \mathbf{p}_x = \overset{-}{\mathbf{p}}_b \mathbf{p}_x$$

or,

$$(\overset{+}{\mathbf{p}}_a - \overset{-}{\mathbf{p}}_b)\mathbf{p}_x = \mathbf{0} \tag{39}$$

where $\overset{+}{\mathbf{p}}_a$ and $\overset{-}{\mathbf{p}}_b$ are orthonormal matrices constructed by the Euler parameters $\mathbf{p}_a$ and $\mathbf{p}_b$, respectively, and $\mathbf{p}_x$ is a 4×1 column vector of unknown Euler parameters. Let

$$\boldsymbol{B} = \overset{+}{\mathbf{p}}_a - \overset{-}{\mathbf{p}}_b$$

The problem of solving $\boldsymbol{A}_a\boldsymbol{A}_x = \boldsymbol{A}_x\boldsymbol{A}_b$ is transformed to the problem of solving a linear system:

$$\boldsymbol{B}\mathbf{p}_x = \mathbf{0} \tag{40}$$

Let the components of $\mathbf{p}_a$ and $\mathbf{p}_b$ be

$$\mathbf{p}_a = [a_0, a_1, a_2, a_3]^T = \left[a_0, \mathbf{a}^T\right]^T$$

$$\mathbf{p}_b = [b_0, b_1, b_2, b_3]^T = \left[b_0, \mathbf{b}^T\right]^T$$

then

$$\begin{aligned} \boldsymbol{B} &= \begin{bmatrix} a_0 & -\mathbf{a}^T \\ \mathbf{a} & a_0\boldsymbol{I} + \tilde{\mathbf{a}} \end{bmatrix} - \begin{bmatrix} b_0 & -\mathbf{b}^T \\ \mathbf{b} & b_0\boldsymbol{I} - \tilde{\mathbf{b}} \end{bmatrix} \\ &= \begin{bmatrix} (a_0 - b_0) & -(\mathbf{a}-\mathbf{b})^T \\ (\mathbf{a}-\mathbf{b}) & (a_0 - b_0)\boldsymbol{I} + (\tilde{\mathbf{a}} + \tilde{\mathbf{b}}) \end{bmatrix} \end{aligned} \tag{41}$$

or,

$$\boldsymbol{B} = \sin(\theta/2)\begin{bmatrix} 0 & -(\mathbf{u}_a - \mathbf{u}_b)^T \\ (\mathbf{u}_a - \mathbf{u}_b) & (\tilde{\mathbf{u}}_a + \tilde{\mathbf{u}}_b) \end{bmatrix} \tag{42}$$

where $a_0 \equiv \cos(\theta_a/2)$, $b_0 \equiv \cos(\theta_b/2)$, $\mathbf{a} \equiv \sin(\theta_a/2)\mathbf{u}_a$, and $\mathbf{b} \equiv \sin(\theta_b/2)\mathbf{u}_b$. Also, $\theta = \theta_a = \theta_b$ which comes from the fact that

$$\boldsymbol{A}_a = \boldsymbol{A}_x \boldsymbol{A}_b \boldsymbol{A}_x^T$$

Since the matrices $\boldsymbol{A}_a$ and $\boldsymbol{A}_b$ are similar, the angles of rotation defined by $\boldsymbol{A}_a$ and $\boldsymbol{A}_b$ are identical (Lemma 4 in [1]).

Eliminating $\sin(\theta/2)$ from both sides of (39) gives

$$(\overset{+}{\mathbf{u}}{}'_a - \overset{-}{\mathbf{u}}{}'_b)\mathbf{p}_x = \mathbf{0} \tag{43}$$

or,

$$\mathbf{u}'_a = (\overset{-}{\mathbf{p}}{}_x^T \overset{+}{\mathbf{p}}_x)\mathbf{u}'_b = \boldsymbol{A}_x\mathbf{u}'_b \tag{44}$$

where $\mathbf{u}'_a = [0, \mathbf{u}_a^T]^T$ and $\mathbf{u}'_b = [0, \mathbf{u}_b^T]^T$ are *vector quaternions*. From (44), we have shown that any rotational transformation which rotates $\mathbf{u}_a$ into $\mathbf{u}_b$ is a solution to (37) (also see Theorem 3 in [1]). The equation (44), in fact, provides an alternative form of the problem described in (37). In addition, since $\sin(\theta/2)$ cannot be zero in (42), $\theta \neq 360°n$ $(n = 0, 1, \ldots)$; that is, $\boldsymbol{A}_a \neq \boldsymbol{A}_b \neq \boldsymbol{I}$.

Now, let us redefine

$$\boldsymbol{B} \equiv \overset{+}{\mathbf{u}}{}'_a - \overset{-}{\mathbf{u}}{}'_b = \begin{bmatrix} 0 & -(\mathbf{u}_a - \mathbf{u}_b)^T \\ (\mathbf{u}_a - \mathbf{u}_b) & (\tilde{\mathbf{u}}_a + \tilde{\mathbf{u}}_b) \end{bmatrix} \tag{45}$$

Since $\boldsymbol{B}$ is a 4×4 skew-symmetric matrix, the rank of $\boldsymbol{B}$ is either 2 or 4 [29]. In order to solve $\boldsymbol{B}\mathbf{p}_x = \mathbf{0}$, we use SVD [30] to diagonalize the system by decomposing $\boldsymbol{B}$ into

$$\boldsymbol{B} = \boldsymbol{U}\Sigma\boldsymbol{V}^T \tag{46}$$

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 $\boldsymbol{B}$, and they are the positive square roots of the eigenvalues of $\boldsymbol{B}^T\boldsymbol{B}$. The columns of $\boldsymbol{U}$ are called the *left singular vectors* of $\boldsymbol{B}$ (or the orthonormal eigenvectors of $\boldsymbol{B}\boldsymbol{B}^T$), and the columns of $\boldsymbol{V}$ are called the *right singular vectors* of $\boldsymbol{B}$ (or the orthonormal eigenvectors of $\boldsymbol{B}^T\boldsymbol{B}$).

Since

$$\sigma_1 = \sigma_2 = 2\sqrt{\mathbf{u}_a^T\mathbf{u}_a} = 2\sqrt{\mathbf{u}_b^T\mathbf{u}_b} = 2$$

are not zero and significant, the matrix $\boldsymbol{B} = \boldsymbol{U}\Sigma\boldsymbol{V}^T$, where

$$\Sigma = \begin{bmatrix} \boldsymbol{D} & 0 \\ 0 & 0 \end{bmatrix} \tag{47}$$

and $\boldsymbol{D} = diag[\sigma_1, \sigma_2]$ with $\sigma_1 = \sigma_2 \neq 0$, is a matrix of rank 2.

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

$$\begin{cases} (\boldsymbol{U}\Sigma)\mathbf{y} = \mathbf{0} \\ \boldsymbol{V}^T\mathbf{p}_x = \mathbf{y} \end{cases} \tag{48}$$

Partitioning $\mathbf{y}$ as

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} = \begin{bmatrix} y_{11} \\ y_{12} \\ y_{21} \\ y_{22} \end{bmatrix} \tag{49}$$

and, $\boldsymbol{U}$ and $\boldsymbol{V}$ corresponding to the partition of Σ

$$\boldsymbol{U} = [\,\boldsymbol{U}_1 \quad \boldsymbol{U}_2\,] \tag{50}$$

$$\boldsymbol{V} = [\,\boldsymbol{V}_1 \quad \boldsymbol{V}_2\,] \tag{51}$$

we can solve the first equation of (48) to obtain

$$\begin{cases} \mathbf{y}_1 = \boldsymbol{D}^{-1}\boldsymbol{U}_1^T\mathbf{0} = \mathbf{0} \\ \mathbf{y}_2 = arbitrary \end{cases} \tag{52}$$

where the columns of $\boldsymbol{U}_1$ span the range of the matrix $\boldsymbol{B}$. Since the right-hand side of (48) is $\mathbf{0}$, $\boldsymbol{U}_1$ is insignificant. The generalized inverse solution can be obtained from the second equation of (48)

$$\mathbf{p}_x = \boldsymbol{V}_1\mathbf{y}_1 + \boldsymbol{V}_2\mathbf{y}_2 \tag{53}$$

where the columns of $\boldsymbol{V}_2$ span the nullspace (ker) of $\boldsymbol{B}$. Substituting (52) into (53), we obtain the complete solution as follows:

$$\mathbf{p}_x = \boldsymbol{V}_2\mathbf{y}_2; \quad \mathbf{y}_2 = arbitrary \tag{54}$$

The solution (54) has one degree of freedom, because the Euler parameters have to satisfy the normality constraint

$$\mathbf{p}_z^T \mathbf{p}_z = 1 \tag{55}$$

which is

$$\mathbf{y}_2^T \boldsymbol{V}_2^T \boldsymbol{V}_2 \mathbf{y}_2 = \mathbf{y}_2^T \mathbf{y}_2 = 1 \tag{56}$$

or,

$$y_{21} = \pm\sqrt{1 - y_{22}^2} \tag{57}$$

The solution can be further written as

$$\mathbf{p}_z = \boldsymbol{V}_2 \begin{bmatrix} \pm\sqrt{1 - y_{22}^2} \\ y_{22} \end{bmatrix}; \quad -1 \leq y_{22} \leq 1 \tag{58}$$

As we can observe, there is infinite number of solutions for $\mathbf{p}_z$.

Although the solution for rotation is obtained using SVD, the actual implementation of SVD is not necessary. Since $\boldsymbol{V}_2$ spans the nullspace of $\boldsymbol{B}$, and $\boldsymbol{B}$ possesses a particular structure, the singular vectors in $\boldsymbol{V}_2$ can be derived symbolically.

5 A UNIQUE SOLUTION

The solution of rotation presented in the previous section have one degree of freedom and is not unique. In order to obtain a unique solution, two distinct sets of data $\boldsymbol{A}_{a_1}$, $\boldsymbol{A}_{b_1}$, $\boldsymbol{A}_{a_2}$, and $\boldsymbol{A}_{b_2}$ should be collected from two movements of the robot arm. This idea was also proposed by Shiu and Ahmad [1]. Given two sets of data (subscribed 1 and 2, respectively), equation (54) can be written as

$$\begin{cases} \mathbf{p}_z = \boldsymbol{V}_1 \mathbf{y}_1 \\ \mathbf{p}_z = \boldsymbol{V}_2 \mathbf{y}_2 \end{cases} \tag{59}$$

which are subject to two scalar constraints

$$\begin{cases} \mathbf{y}_1^T \mathbf{y}_1 = 1 \\ \mathbf{y}_2^T \mathbf{y}_2 = 1 \end{cases} \tag{60}$$

where $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$ are the right singular vectors span the nullspace of $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$, respectively. Equating (59) gives

$$\boldsymbol{V}_1 \mathbf{y}_1 - \boldsymbol{V}_2 \mathbf{y}_2 = \mathbf{0}$$

or,

$$\begin{bmatrix} \boldsymbol{V}_1 & -\boldsymbol{V}_2 \end{bmatrix} \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} = \mathbf{0} \tag{61}$$

Equations (60) and (61) are a total of six scalar equations in four unknowns. Three of (61) and one of (60) may be chosen to form a system of four simultaneous equations. Equations in (60) cannot be used at the same time, since one implies the other. This can be shown by applying the identity, $\boldsymbol{V}_1^T \boldsymbol{V}_1 = \boldsymbol{V}_2^T \boldsymbol{V}_2 = \boldsymbol{I}$, and the first equation in (60) to (61); the result gives the second equation of (60).

Dimension Reduction. Solving the afore-mentioned system of nonlinear equations can be simplified using a technique, called dimension reduction [14], which utilizes the structure of the Jacobian matrix of the system is introduced.

Forming the normal equation of (61) and identifying $\boldsymbol{V}_1^T \boldsymbol{V}_1 = \boldsymbol{V}_2^T \boldsymbol{V}_2 = \boldsymbol{I}$, we have

$$\begin{bmatrix} \boldsymbol{I} & -\boldsymbol{W} \\ -\boldsymbol{W}^T & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{62}$$

where

$$\boldsymbol{W} = \boldsymbol{V}_1^T \boldsymbol{V}_2 = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} \tag{63}$$

If the normal matrix has a full rank, we can choose any three out of four from (62). However, we can see that the first two equations have full row rank of 2; we can always choose them. The rest of two equations may impose dependency on the first two. A criterion which is designed to find one equation from the second set of (62) is developed.

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

$$\begin{bmatrix} 1 & 0 & -w_{11} & -w_{12} \\ 0 & 1 & -w_{21} & -w_{22} \\ 0 & 0 & \alpha & \gamma \\ 0 & 0 & \gamma & \beta \end{bmatrix} \tag{64}$$

where

$$\alpha = 1 - w_{11}^2 - w_{21}^2 \tag{65}$$

$$\beta = 1 - w_{12}^2 - w_{22}^2 \tag{66}$$

$$\gamma = -w_{11}w_{12} - w_{21}w_{22} \tag{67}$$

In (64), if the last two rows are both zeroes, the corresponding equations are dependent; we do not have a sufficient set of equations to solve simultaneously. If one row 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 zeroes, we can perform GE again using q as the pivoting element to get a matrix

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

where q equals α, β, or γ which 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 the following three cases. **Case 1:** If $\alpha = \beta = \gamma = 0$, no unique solution exists. This case is equivalent to having $\boldsymbol{I} - \boldsymbol{W}^T\boldsymbol{W} = 0$, and it indicates that $\mathbf{u}_{a_1}$ and $\mathbf{u}_{a_2}$ are parallel or anti-parallel. **Case 2:** If $|\alpha|$ or $|\gamma| = max(|\alpha|, |\beta|, |\gamma|)$, we choose the first equation of $-\boldsymbol{W}^T\mathbf{y}_1 + \mathbf{y}_2 = \mathbf{0}$. **Case 3:** If $|\beta| = max(|\alpha|, |\beta|, |\gamma|)$, we choose the second equation of $-\boldsymbol{W}^T\mathbf{y}_1 + \mathbf{y}_2 = \mathbf{0}$.

Assuming that case 3 is identified; a system of four nonlinear equations, denoted by $\mathbf{E}$, is formed by taking the second equation of (60), and the first two and the fourth of (62):

$$\mathbf{E}(\mathbf{y}_1, \mathbf{y}_2) = \begin{bmatrix} E_1 \\ E_2 \\ E_3 \\ E_4 \end{bmatrix} \equiv \begin{bmatrix} y_{11} - w_{11}y_{21} - w_{12}y_{22} \\ y_{12} - w_{21}y_{21} - w_{22}y_{22} \\ \frac{1}{2}(y_{21}^2 + y_{22}^2 - 1) \\ -w_{12}y_{11} - w_{22}y_{12} + y_{22} \end{bmatrix} = \mathbf{0} \tag{68}$$

where $\mathbf{y}_1 = [y_{11}, y_{12}]^T$, and $\mathbf{y}_2 = [y_{21}, y_{22}]^T$.

Solving the nonlinear system of equations by Newton-Raphson iteration, a linear system of the form

$$\boldsymbol{J}\Delta\mathbf{y} = \mathbf{E} \tag{69}$$

has to be solved. The matrix, $\boldsymbol{J} \equiv \frac{\partial \mathbf{E}}{\partial \mathbf{y}}$, is the system Jacobian; $\Delta\mathbf{y}$ is the vector of corrections, and $\mathbf{E}$ is the vector of residuals (errors).

Let us take a look at the structure of the Jacobian. Writing (65) in scalar form gives

$$\begin{bmatrix} 1 & 0 & -w_{11} & -w_{12} \\ 0 & 1 & -w_{21} & -w_{22} \\ 0 & 0 & y_{21} & y_{22} \\ -w_{12} & -w_{22} & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta y_{11} \\ \Delta y_{12} \\ \Delta y_{21} \\ \Delta y_{22} \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ E_3 \\ E_4 \end{bmatrix} \tag{70}$$

Row pivotings for Gaussian elimination of the first two columns are not needed, since the norm of $\boldsymbol{W}$ is bounded by 1. If $|y_{21}| \geq |y_{22}|$, symbolic GE can be performed to get

$$\begin{bmatrix} 1 & 0 & -w_{11} & -w_{12} \\ 0 & 1 & -w_{21} & -w_{22} \\ 0 & 0 & y_{21} & y_{22} \\ 0 & 0 & 0 & J' \end{bmatrix} \begin{bmatrix} \Delta y_{11} \\ \Delta y_{12} \\ \Delta y_{21} \\ \Delta y_{22} \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ E_3 \\ E_4' \end{bmatrix} \tag{71}$$

where J' and E_4' are the reduced Jacobian and residual, respectively. Solving the reduced system $J'\Delta y_{22} = E_4'$ and then the full system, we have the following closed-form formulae:

$$\left\{\begin{array}{rcl} J' &=& \beta - (\frac{\gamma}{y_{21}})y_{22} \\ E_4' &=& w_{12}E_1 + w_{22}E_2 - (\frac{\gamma}{y_{21}})E_3 + E_4 \\ \Delta y_{22} &=& \frac{E_4'}{J'} \\ \Delta y_{21} &=& \frac{(E_3 - y_{22}\Delta y_{22})}{y_{21}} \end{array}\right. \tag{72}$$

If $|y_{22}| > |y_{21}|$, we use another set of closed-form formulae:

$$\left\{\begin{array}{rcl} J' &=& \gamma - (\frac{\beta}{y_{22}})y_{21} \\ E_4' &=& w_{12}E_1 + w_{22}E_2 - (\frac{\beta}{y_{22}})E_3 + E_4 \\ \Delta y_{21} &=& \frac{E_4'}{J'} \\ \Delta y_{22} &=& \frac{(E_3 - y_{21}\Delta y_{21})}{y_{22}} \end{array}\right. \tag{73}$$

For both cases, we also have

$$\left\{\begin{array}{rcl} \Delta y_{11} &=& E_1 + w_{11}\Delta y_{21} + w_{12}\Delta y_{22} \\ \Delta y_{12} &=& E_2 + w_{21}\Delta y_{21} + w_{22}\Delta y_{22} \end{array}\right. \tag{74}$$

The selection of the larger of y_{21} and y_{22} provides stability of GE. As we can see, LU factorization is not needed when solving (70); it is done symbolically. Similarly, for case 3 a system of four nonlinear equations can be collected, and a set of similar formulae can be derived.

The selection criterion and the simplified iteration developed provide an algorithmic solution that can be easily incorporated in application programs which require the calculation of the relative orientation between the reference frames of a link-mounted camera and the link.

6 EXAMPLE

A FORTRAN program implementing the proposed method has been tested on a number of examples. In order to demonstrate the accuracy of the method, we constructed examples by specifying $\boldsymbol{A}_{a_1}$, $\boldsymbol{A}_{a_2}$, and $\boldsymbol{A}_x$ and calculating $\boldsymbol{A}_{b_1}$, $\boldsymbol{A}_{b_2}$ using $\boldsymbol{A}_b = \boldsymbol{A}_x^T\boldsymbol{A}_a\boldsymbol{A}_x$. The program is then run to solve for $\boldsymbol{A}_x$, given $\boldsymbol{A}_{a_i}$ and $\boldsymbol{A}_{b_i}$ $(i = 1, 2)$. The program obtained $\boldsymbol{A}_x$ to machine accuracy as the original.

The following is a typical example of the results. By assuming that

$$\boldsymbol{A}_x = \begin{bmatrix} -0.84433375 & -0.01867995 & -0.53549191 \\ 0.41718555 & -0.65006227 & -0.63511831 \\ -0.33623910 & -0.75965131 & 0.55666252 \end{bmatrix} \tag{75}$$

$$\boldsymbol{A}_{a_1} = \begin{bmatrix} -0.92592593 & -0.37037037 & -0.07407407 \\ 0.28148148 & -0.80740741 & 0.51851852 \\ -0.25185185 & 0.45925926 & 0.85185185 \end{bmatrix} \tag{76}$$

and

$$\boldsymbol{A}_{a_2} = \begin{bmatrix} -0.83134406 & 0.02335236 & -0.55526725 \\ -0.52153607 & 0.31240270 & 0.79398028 \\ 0.19200830 & 0.94966269 & -0.24753503 \end{bmatrix} \tag{77}$$

the matrices $\boldsymbol{A}_{b_i}$ $(i = 1, 2)$ are computed as

$$\boldsymbol{A}_{b_1} = \begin{bmatrix} -0.90268482 & 0.10343126 & -0.41768659 \\ 0.38511568 & 0.62720266 & -0.67698060 \\ 0.19195318 & -0.77195777 & -0.60599932 \end{bmatrix} \tag{78}$$

and

$$\boldsymbol{A}_{b_2} = \begin{bmatrix} -0.73851280 & -0.54317226 & 0.39945305 \\ -0.45524951 & 0.83872293 & 0.29881721 \\ -0.49733966 & 0.03882952 & -0.86668653 \end{bmatrix} \tag{79}$$

Using the algorithm given by Paul [27], the axes of rotation are found to be

$$\left\{\begin{array}{rcl} \mathbf{u}_{a_1} &=& [-0.08737041,\ 0.26211122,\ 0.96107446]^T \\ \mathbf{u}_{a_2} &=& [-0.16599940,\ 0.79679712,\ 0.58099790]^T \\ \mathbf{u}_{b_1} &=& [-0.14003202,\ -0.89883801,\ 0.41530864]^T \\ \mathbf{u}_{b_2} &=& [0.27721693,\ -0.95622269,\ -0.09374935]^T \end{array}\right. \tag{80}$$

and the angles of rotation are

$$\left\{\begin{array}{rcccl} \theta_{a_1} &=& \theta_{b_1} &=& 207.964513^\circ \\ \theta_{a_2} &=& \theta_{b_2} &=& 160.176329^\circ \end{array}\right. \tag{81}$$

Since $\theta_{a_1} = \theta_{b_1}, \theta_{a_2} = \theta_{b_2}$, and $\mathbf{u}_{a_1}^T\mathbf{u}_{a_2} = \mathbf{u}_{b_1}^T\mathbf{u}_{b_2} = 0.78173513 \neq 1$ or -1, a unique solution exists.

The singular vectors which span the nullspace of $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ are

$$\boldsymbol{V}_1 = \begin{bmatrix} -0.76674039 & 0.00000000 \\ 0.63431333 & 0.14829167 \\ -0.06409961 & 0.41521667 \\ 0.07514656 & -0.89755484 \end{bmatrix} \tag{82}$$

and

$$\boldsymbol{V}_2 = \begin{bmatrix} -0.26229611 & 0.00000000 \\ 0.91664371 & 0.21200758 \\ 0.27735851 & -0.30390381 \\ -0.11847918 & 0.92881390 \end{bmatrix} \tag{83}$$

Now, the matrix $\boldsymbol{W}$ can be calculated as

$$\boldsymbol{W} = \boldsymbol{V}_1^T\boldsymbol{V}_2 = \begin{bmatrix} 0.75587048 & 0.22375652 \\ 0.35743607 & -0.92840838 \end{bmatrix} \tag{84}$$

and accordingly

$$\left\{\begin{array}{rcl} \alpha &=& 0.30089927 \\ \beta &=& 0.08799089 \\ \gamma &=& 0.16271569 \end{array}\right. \tag{85}$$

Since $|\alpha| = max(|\alpha|, |\beta|, |\gamma|)$, we choose the first equation of $-\boldsymbol{W}^T\mathbf{y}_1 + \mathbf{y}_2 = \mathbf{0}$ to solve simultaneously. Given an arbitrary initial guess $\mathbf{y} = [y_{11}, y_{12}, y_{21}, y_{22}]^T = [0.1, 0.2, 0.1, 0.2]^T$, the value of $\mathbf{y}$ after eight Newton-Raphson iterations is

$$\mathbf{y} = [-0.16272298, -0.98667180, -0.47566958, 0.87962404]^T \tag{86}$$

which has precision of 0.27×10^{-13}.

Now, $\mathbf{p}_x$ can be found by either equation of (59) as

$$\mathbf{p}_x = [0.12476628, -0.24953256, -0.39925210, 0.87336397]^T \tag{87}$$

The rotational matrix is computed using (20) as

$$\boldsymbol{A}_x = \begin{bmatrix} -0.84433375 & -0.01867995 & -0.53549191 \\ 0.41718555 & -0.65006227 & -0.63511831 \\ -0.33623910 & -0.75965131 & 0.55666252 \end{bmatrix} \tag{88}$$

which is identical to the rotational matrix given in (75).

7 CONCLUSIONS

A quaternion approach using the analysis of SVD to the solution of the kinematic equation, $\boldsymbol{A}_a\boldsymbol{A}_x = \boldsymbol{A}_x\boldsymbol{A}_b$, for the relative orientation between a link-mounted sensor and the link has been presented. Using quaternions to represent the kinematic equation of rotations in terms of rotational matrices provides an alternative way to formulate the kinematic equation. This simplifies the original equation extremely and reduce it to a simple and well-structured linear system, $\boldsymbol{B}\mathbf{p}_x = \mathbf{0}$.

Using SVD analysis, it has been shown that the general solution of the equation of rotation involves the right singular vectors $\boldsymbol{V}_2$ which span the nullspace of the coefficient matrix $\boldsymbol{B}$. However, $\boldsymbol{V}_2$ is derived symbolically in a closed form. The actual implementation of SVD algorithm is not required.

Since the general solution of rotation is not unique, two distinct robot motions have to be made to obtain two sets of data such that a unique solution can be reached. A simple criterion to select the right set of simultaneous equations is presented. Although a set of four non-linear equations has to be solved simultaneously using an iterative method, closed-form formulae have been developed which reduces the computations during iterations. The selection criterion and the solution method are robust and can be easily incorporated in application programs which require the calculation of the relative orientation between the reference frames of a link-mounted camera and the link.

8 ACKNOWLEDGEMENT

This research is supported by the Pattern Analysis and Machine Intelligence Laboratory at the Department of Systems Design Engineering, University of Waterloo and is funded by Natural Science and Engineering Research Council of Canada.

References

[1] Y. C. Shiu and S. Ahmad. Finding the mounting position of a sensor by solving a homogeneous transform equation of the form $\boldsymbol{AX} = \boldsymbol{XB}$. *Proceedings of 1987 IEEE International Conference on Robotics and Automation*, 3:1667 – 1671, March 1987.

[2] F. R. Gantmacher. *The Theory of Matrices*. Volume One, Chelsea Publishing Company, New York, N. Y., 1959.

[3] W. R. Hamilton. *Lectures on Quaternions*. Hodges and Smith, Dublin, 1853.

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

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

[6] A. C. Robinson. *On the Use of Quaternions in Simulation of Rigid-Body Motion*. Aeronautical Research Lab., Wright Air Development Center, December 1958.

[7] P. E. Nikravesh and I. S. Chung. Applications of Euler parameters to the dynamic analysis of three-dimensional constrained mechanical systems. *ASME Journal of Mechanical Design*, 785 – 791, 1982.

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

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

[10] J. C. K. Chou, K. Singhal, and H. K. Kesavan. Multi-body systems with open chains: graph-theoretic models. *Mechanism and Machine Theory*, 21(3):273 - 284, 1986.

[11] Jack C. K. Chou, H. K. Kesavan, and Kishore Singhal. A systems approach to 3-D multi-body systems using graph-theoretic models. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-16(2):219 - 230, 1986.

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

[13] J. C. K. Chou, G. Baciu, and H. K. Kesavan. Graph-theoretic models for simulating robot manipulators. *Proceedings of 1987 IEEE International Conference on Robotics and Automation*, 2:953 - 959, March 1987.

[14] J. C. K. Chou, G. Baciu, and H. K. Kesavan. Computational scheme for simulating robot manipulators. *Proceedings of 1987 IEEE International Conference on Robotics and Automation*, 2:961 - 966, March 1987.

[15] F. Klein. *Elementary Mathematics From the Advanced Standpoint*. New-York, Dover, 1925.

[16] L. Brand. *Vector and Tensor Analysis*. Chapter X, 6th printing, John Wiley and Sons, New-York, 1957.

[17] B. Brainerd, D. A. Clarke, M. J. Liebovitz, R. A. Rose, and G. A. Scroggie. *Vector and Matrices*. Chapter 7, The Ryerson Press, Toronto, 1967.

[18] D. J. Foulis. *Fundamental Concepts of Mathematics*. Chapter 4, Weber and Schmidt, Prindle, 1969.

[19] O. Bottema and B. Roth. *Theoretical Kinematics*. Chapter XIII, North-Holland, Amsterdam, 1979.

[20] R. A. Wehage. Quaternions and Euler parameters - a brief exposition. *Computer Aided Analysis and Optimization of Mechanical System Dynamics*, 147 – 180, E. J. Haug (ed.), Spring-Verlag, Heidelberg, 1984.

[21] A. Cayley. On certain results relating to quaternions. *The Collected Mathematical Papers of Arthur Cayley*, I:123–126, 1889.

[22] J. Wittenburg. *Dynamics of Systems of Rigid Bodies*. Chapter 2, B. G. Teubner, Stuttgart, 1977.

[23] J. W. Gibbs. *Vector Analysis*. Chapter VI, Republished from 2nd ed. in 1909, New-York, Dover, 1960.

[24] H. Goldstein. *Classical Mechanics*. Addison-Wesley, 2nd ed., July 1981.

[25] C. H. Suh and C. W. Radcliffe. *Kinematics and Mechanisms Design*. Chapter 3, John Wiley and Sons, 1978.

[26] O. Rodrigues. 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. *Journal de Mathematiques Pures Appl.*, 5:380 – 440, 1840.

[27] R. P. Paul. *Robot Manipulators: Mathematics, Programming, and Control*. The MIT Press, 1981.

[28] J. J. Craig. *Introduction to Robotics — Mechanics and Control*. Addison-Wesley, 1986.

[29] G. L. Bradley. *A Primer of Linear Algebra*. Prentice-Hall, 1975.

[30] G. W. Stewart. *Introduction to Matrix Computations*. Academic Press, 1973.