Mechanism and Machine Theory 125 (2018) 21–33

Contents lists available at ScienceDirect

Mechanism and Machine Theory

journal homepage: www.elsevier.com/locate/mechmachtheory

Research paper

On the use of the dual Euler–Rodrigues parameters in the numerical solution of the inverse-displacement problem

Wei Li[a,*], Toby Howison[b], Jorge Angeles[a]

[a] *Department of Mechanical Engineering and Centre for Intelligent Machines, McGill University, 817, Sherbrooke Street West, Montreal, QC H3A 0C3, Canada*
[b] *Cambridge University Engineering Department Trumpington St Cambridge CB2 1PZ, United Kingdom*

ARTICLE INFO

Article history:
Received 16 October 2017
Revised 26 November 2017
Accepted 6 December 2017
Available online 18 December 2017

Keywords:
Numerical inverse displacement problem
Dual Euler–Rodrigues parameters
Dual quaternions
Overdetermined system of nonlinear equation
Nonlinear least squares

ABSTRACT

The Euler–Rodrigues parameters (ERP) are revisited in this paper, the motivation being the need for a systematic methodology to formulate the inverse-displacement problem associated with a six-revolute (6R) serial robot. Although significant progress was made in the solution of the problem in the eighties and nineties, there is not one algorithm adopted by the research and applications communities, but rather various algorithms that lead to the resolvent polynomial of 16th degree or lower. In this paper the problem is formulated using the dual ERP. Motivated by the lack of a methodology that would allow R&D professionals to readily implement a generic solution algorithm in industrial environments requiring *real-time* inverse-displacement problem (IDP) solutions. An inverse-displacement *numerical solution* of 6R robots of arbitrary architecture is proposed. This is the main thrust of the paper. The contribution of the paper is a systematic procedure to formulate the problem and implement its solution by numerical means. The procedure is illustrated with the example of an industrial robot whose architecture does not allow for a closed-form solution.

1. Introduction

The *Euler-Rodrigues parameters* (ERP) [1,2] have received much less attention in the literature than their isomorphs, *quaternions*, although the former admit a simpler interpretation than the latter. Indeed, quaternions are usually introduced as algebraic entities on their own merit, which happen to be extremely useful in the kinematics of rotations. Indeed, quaternions allow for a compact form, with four real scalars, as opposed to the nine of the proper orthogonal matrix from which they are extracted. The ERP make up a set of *invariants* of the rotation matrix. An invariant quantity in physics is one that follows specific rules upon a change of frame [3]: if a scalar, the quantity is immutable under the change; if a vector or a tensor, then the quantity changes according to what is known as a *similarity transformation* in linear algebra [4]. A rotation is known to have what could be thought of as two *natural* invariants, one scalar, its angle of rotation, one vector, the unit vector parallel to its axis of rotation, whose direction determines the sign of the angle of rotation. Extracting these invariants from the entries of the rotation matrix is done indirectly, via an alternative set of invariants. There are basically two of

* Corresponding author.
E-mail addresses: livey@cim.mcgill.ca (W. Li), th533@cam.ac.uk (T. Howison), angeles@cim.mcgill.ca (J. Angeles).

https://doi.org/10.1016/j.mechmachtheory.2017.12.006
0094-114X/

these, the linear invariants and the ERP.[1] The former are the *trace* and the *axial vector* of the rotation matrix [6]. The trace is in a linear, non-homogeneous relation with the cosine of the angle of rotation, the axial vector being a bilinear function of the unit vector in the direction of the axis of rotation and the sine of the angle of rotation. Moreover, one and the same rotation matrix represents both one rotation about an axis given by the unit vector $\mathbf{e}$ through an angle ϕ, and a second, similar rotation, with the same natural invariants, but with their signs reversed. The analyst is thus free to choose either the direction of the unit vector or the sign of the angle of rotation, given that a reversal of the signs of these invariants yields the same rotation matrix. An attractive feature of the *linear invariants* is the ease with which they are extracted from the rotation matrix, as this operation is, as the name indicates, *linear*, i.e., their extraction from the rotation matrix involves only additions and subtractions of the matrix entries.

The linear invariants have been used to formulate and solve the inverse displacement problem (IDP) associated with a 6R robot numerically [7]. In the cited paper, the problem was solved using the four linear-invariant equations and the three translation equations, thereby ending up with a system of seven equations in six unknowns. Although the system is overdetermined, it is consistent, but vulnerable to a representation singularity, because the axial vector vanishes when the angle of rotation is π.

By virtue of the above singularity, linear invariants lack robustness. Here come to the rescue the ERP. The relation between these invariants and the rotation matrix is not linear, but *quadratic*. Indeed, the ERP can be interpreted as the linear invariants of the square root of the given rotation. Now, matrix square-root extraction is, in principle, a rather complex operation—it calls for an application of the Cayley–Hamilton Theorem[4], followed by linear-equation solving–but, in the case of the rotation matrix, square-rooting reduces to a simple operation: replace the angle of rotation ϕ with $\phi/2$. This is straightforward when the natural invariants are known. If they aren't, then they can be found via the linear invariants. The determination of the ERP from the linear invariants is a simple operation [6], involving only the formulas for the harmonic functions of a half angle in terms of those of the full angle. The result, the ERP, is a set of robust invariants, free of the above-mentioned singularity.

While the ERP and the quaternions are isomorphic, we prefer the former because of their direct relation with the rotation matrix. As the record shows, ERP appeared in the scholarly literature in 1840 [8], while quaternions four years later [9]. In a comprehensive article on the authorship of the concept, ERP/quaternions, Altmann [1] gives credit to O. Rodrigues, but acknowledges that very little is known of Rodrigues' life and work, while a copious literature exists on Hamilton, the creator of quaternions.

The derivation of the pertinent algebraic relations in this paper is based on the Denavit–Hartenberg (DH) notation [10]. In the balance of the paper the ERP are revisited, in their dual version, in light of the numerical solution of the IDP. Moreover, the ERP of the *i*th DH rotation matrix are derived, along with expressions for the ERP of a product of successive rotations, in Section 2. The foregoing rotation equations are complemented with their translation counterparts, which is done in Section 3. The translation equations are tersely derived from the rotation equations upon *dualizing* both the DH rotation matrices and their ERP, the desired equations being the dual parts of the foregoing dual equations. The gradient of the translation equations w.r.t. the array of joint angles is also derived upon dualizing the gradient of the rotation equations. The section ends with an overdetermined nonlinear system of eight equations in six unknowns. The solution of this system is then discussed in Section 4 by means of nonlinear least squares, as implemented with the Newton–Gauss method, which relies on the 8×6 gradient, i.e., the Jacobian matrix, of the system of equations w.r.t. the six unknown angles. The relation between this gradient and what is known in the robotics literature as "the Jacobian"[2] is discussed in this section. The implementation of the numerical solution is demonstrated with the aid of a case study in Section 5.

The motivation of the paper is thus the numerical solution of the IDP in its full generality, in which the orientation problem cannot be decoupled from the translation problem. For this reason, robots with an architecture that allows the decoupling are sometimes referred to as *decoupled robots*, which are the norm in industry at the moment. In fact, the authors can cite only two six-revolute decoupled robots in industry, the TelBot and the Fanuc Arc Mate S [6] with a non-decoupled architecture. It can be argued that one very good reason why non-decoupled robots are not popular in industry is the lack of commercial software that caters to them. One objective of this paper is to pave the way to a large variety of robots in industry, with their many advantages in terms of mobility and dexterity, among other performance indicators. One good motivation to investigate the IDP in its full generality is the acknowledgment that all robots in real life are coupled. Robots that are designed with a decoupled architecture turn out to be, after calibration, coupled. The availability of a numerical scheme that caters to coupled robots should lead to more precise *real-time* control algorithms than the state of the art. The same problem was extensively investigated in the eighties and early nineties, at about the same time that the long-standing effort for the derivation of the minimal polynomial, using elimination methods, was intense. It had been anticipated that the system of rotation and translation equations should yield a *minimal polynomial* of degree 16. The first researchers who succeeded in deriving the minimal polynomial were Raghavan and Roth [11,12] and Lee et al. [13]. These seminal papers appeared to close a chapter in the history of kinematics that led to intensive research work for over one decade all over the world. However, as recently as the past decade, Husty et al. [14] proposed a novel formulation of the IDP by means of a

[1] At least one more set of invariants has been proposed in the literature, namely, those dubbed the *modified Rodrigues parameters* [5], similar to the ERP, but with $\phi/2$ replaced with $\phi/4$. These, however, have not yet found adepts since their inception, in the mid-nineties.

[2] This putative Jacobian is the 6×6 matrix that maps the six-dimensional array of joint rates into the (six-dimensional) end-effector twist. This array is not a Jacobian, properly speaking, because the angular-velocity vector is not a time-derivative.

clever partitioning of the 6R chain into two 3R subchains. While the idea seems extremely simple, its implementation calls for advanced knowledge of geometry, which may be a reason why this approach has not found its way into the shop floor.

Once the problem has been formulated, the main thrust of this paper, the system of equations derived is to be solved using a method suitable to nonlinear systems, which is necessarily *iterative*. There are mainly two classes of methods: symbolic and numerical, the former relying on computer algebra, the latter on number-crunching. As well, in the realm of numerical methods there are *derivative-free methods* and *gradient methods*. A feature of numerical methods lies in that the solution found, in the presence of convergence, is one of multiple possible solutions. Each robot admits a distinct number of possible solutions, To find how many, the displacement equations must be manipulated so that, by means of a technique based on what is known as *tan-half identities*, the equations in the harmonic functions of the joint angles are transformed into multivariate polynomial equations. By means of techniques like *dialytic elimination* [15], unknowns can be systematically eliminated until one single monovariate polynomial equation is obtained. the question then is whether this polynomial is *minimal*, equivalent to whether it contains spurious solutions. The way to answer this question is to find (numerically) as many solutions as possible. If as many solutions as the degree of the polynomial are found, then the polynomial is declared *minimal*.

Here we recall only a small sample of numerical methods reported in the literature. A main issue is to find all possible solutions for a given robot at a given end-effector pose. A purely numerical method isn't capable of obtaining all possible solutions. However, combined with *polynomial continuation* this is possible [16,17]. One more approach intended to solve the IDP is based on what is termed *incremental methods*. In these methods, the displacement problem is solved *incrementally*, by starting with the linear equations at the velocity level, then applying a small time-increment to obtain the next posture of the robot [18,19]. These methods have been successfully applied in the area of non-holonomic robots and redundancy resolution, topics that lie outside of the scope of our paper. Other methods worth mentioning include the cutting of one joint to eliminate one unknown. Furthermore, the orientation problem can be obviated, for example, upon redefining it with three positioning problems, one for each of corresponding points of the EE, thereby ending up with nine translation equations and no orientation equations [20]. Compared with these methods, ours has the advantage that we provide a simple and systematic formulation for the displacement equations and their gradient. Moreover, the different columns of the gradient can be calculated recursively, making the method more efficient, besides being free of a representation singularity.

In the authors' opinion, the often-mentioned dichotomy between elimination methods—wrongly termed, in the authors' opinion, *analytical*—and numerical methods has to be taken with a grain of salt. Indeed, *elimination methods*, supported with the fundamentals of algebraic geometry and computer algebra, represent an invaluable means to derive the *minimal monovariate polynomial* of 16th or lower degree that sets a bound on the possible number of inverse-displacement problem solutions admitted by the robot at hand at a given EE pose. Numerical solutions, in turn, provide an invaluable means to enable the on-line control of trajectory-following in the presence of event-driven operations—e.g., the approach of a workpiece on a belt-conveyor, to be grabbed by the robot. That is, computer-algebraic and numerical solutions are not competing against each other, but rather complementing themselves towards a pragmatic end, effective *object manipulation*.

However, the two foregoing classes of methods face their own challenges. Polynomial-root finding is an intrinsically ill-conditioned problem, as Forsythe [21] showed with an example: small perturbations in the coefficients bring about significant changes in the roots. Numerical methods fail to produce a solution in the neighborhood of a geometric singularity. The robustness we claim on the use of dual ERP to formulate the IDP targets representation singularities, not geometric singularities. The relation between the two is outlined in Section 5. The issue of geometric singularities is left out, as it merits a study on its own. An algorithm intended to cope with singularities upon trajectory-following was proposed [22], based on a quadratic-programming technique intended to avoid branch-switching.

2. The orientation problem

We resort throughout the paper to the Denavit–Hartenberg (DH) notation [10], as complemented later [6]. The rotation matrix $\mathbf{Q}_i$ produced by a rotation θ_i of the ith joint is thus expressed as

$$\mathbf{Q}_i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i\sin\theta_i & \sin\alpha_i\sin\theta_i \\ \sin\theta_i & \cos\alpha_i\cos\theta_i & -\sin\alpha_i\cos\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i \end{bmatrix} \tag{1}$$

for $i = 1, \ldots, 6$, with α_i denoting the twist angle between Z_i and Z_{i+1}, measured positive in the direction of X_{i+1}, and all coordinate axes defined according to the DH convention. Moreover, the four-dimensional ERP array associated with $\mathbf{Q}_i$, denoted $\boldsymbol{\eta}_i$, is defined as

$$\boldsymbol{\eta}_i = [\mathbf{r}_i^T,\ r_i]^T \tag{2}$$

where, if $\mathbf{e}_i$ denotes the unit vector parallel to the axis of the rotation of $\mathbf{Q}_i$ and ϕ_i its associated angle of rotation[3], then

$$\mathbf{r}_i \equiv \mathbf{e}_i \sin\left(\frac{\phi_i}{2}\right), \quad r_i \equiv \cos\left(\frac{\phi_i}{2}\right) \tag{3}$$

[3] It is noteworthy that $\mathbf{e}_i$ is different from $\mathbf{k}_i$, henceforth the unit vector parallel to the axis of the ith joint; similarly, ϕ_i is a function of α_i and θ_i.

The reader familiar with *quaternions* will be able to identify $\mathbf{r}_i$ and r_i with the vector and, correspondingly, the scalar of the *quaternion* representing $\mathbf{Q}_i$.

Henceforth, every unit four-dimensional vector array, like $\boldsymbol{\eta}_i$, will be termed an *Euler vector*. The expression for this vector is found from relations available in the literature [23]. After simplification, $\boldsymbol{\eta}_i$ is found to be[4]

$$\boldsymbol{\eta}_i = [s_{\bar{\alpha}_i} c_{\bar{\theta}_i},\ s_{\bar{\alpha}_i} s_{\bar{\theta}_i},\ c_{\bar{\alpha}_i} s_{\bar{\theta}_i},\ c_{\bar{\alpha}_i} c_{\bar{\theta}_i}]^T \tag{4}$$

where, for brevity,

$$s_x \equiv \sin x,\ c_x \equiv \cos x,\ s_{\bar{x}} \equiv \sin(x/2),\ c_{\bar{x}} \equiv \cos(x/2) \tag{5}$$

For completeness, $\mathbf{Q}_i$ can be re-constructed from its ERP, namely,

$$\mathbf{Q}_i = (r_i^2 - \mathbf{r}_i \cdot \mathbf{r}_i)\mathbf{1} + 2\mathbf{r}_i\mathbf{r}_i^T + 2r_i\mathbf{R}_i \tag{6}$$

with $\mathbf{1}$ representing the 3×3 identity matrix and $\mathbf{R}_i \equiv \text{CPM}(\mathbf{r}_i)$, i.e., the *cross-product matrix*[5] of $\mathbf{r}_i$. As a consequence, proper orthogonal matrices—those orthogonal matrices whose determinant is $+1$, thereby forming SO(3)—and Euler vectors are *isomorphic*. Further, the target orientation of the EE is given by $\mathbf{Q}_0$, its Euler vector being $\boldsymbol{\eta}_0$, i.e.,

$$\boldsymbol{\eta}_0 = [\mathbf{r}_0^T,\ r_0]^T,\ \mathbf{r}_0 = \mathbf{e}_0 \sin(\phi_0/2),\ r_0 = \cos(\phi_0/2) \tag{7}$$

which is part of the data in the IDP.

First, we formulate the rotation subproblem. Let $\boldsymbol{\eta}_{ij}$ represent the Euler vector of the product $\mathbf{Q}_i\mathbf{Q}_j$; expressions for the ERP of the product of two real rotation matrices were first investigated by Rodrigues [8]. The relations of interest are found explicitly in the literature on the subject [2,23]. Vector $\boldsymbol{\eta}_{ij}$, corresponding to the product $\mathbf{Q}_i\mathbf{Q}_j$, can be obtained from the expressions for the vector and scalar ERP [23]:

$$\mathbf{r}_{ij} = r_i\mathbf{r}_j + r_j\mathbf{r}_i + \mathbf{r}_i \times \mathbf{r}_j, \quad r_{ij} = r_i r_j - \mathbf{r}_i^T\mathbf{r}_j \tag{8}$$

which can be written in a more compact form upon introducing the operator $\otimes$—read "ocross"—in following Shuster [23]:

$$\boldsymbol{\eta}_{ij} = \boldsymbol{\eta}_i \otimes \boldsymbol{\eta}_j \tag{9}$$

We denote this operation as the *Euler product* hereafter. The rotation *closure equation* of the IDP is recalled:

$$\mathbf{Q}_1\mathbf{Q}_2 \ldots \mathbf{Q}_6 = \mathbf{Q}_0 \tag{10}$$

Then, the orientation subproblem can be expressed as

$$\mathbf{f}_R \equiv \boldsymbol{\eta}_1 \otimes \boldsymbol{\eta}_2 \otimes \ldots \otimes \boldsymbol{\eta}_6 - \boldsymbol{\eta}_0 = \mathbf{0} \tag{11}$$

which leads to four real equations.

Since the matrix product in Eq. (10) is associative, the Euler product is bound to be associative as well. Moreover, the LHS of Eq. (11) can be calculated recursively: If we define $\boldsymbol{\lambda}_i$ as the product

$$\boldsymbol{\lambda}_i \equiv \boldsymbol{\eta}_1 \otimes \boldsymbol{\eta}_2 \otimes \ldots \otimes \boldsymbol{\eta}_i, \quad i = 1, \ldots, 6 \tag{12}$$

then

$$\boldsymbol{\lambda}_1 = \boldsymbol{\eta}_1, \quad \boldsymbol{\lambda}_i = \boldsymbol{\lambda}_{i-1} \otimes \boldsymbol{\eta}_i, \quad i = 2, \ldots, 6 \tag{13}$$

2.1. Computation of the gradient of the orientation equations

When solving the IDP numerically, the method of choice is Newton–Gauss [24], applicable to systems of m non-linear equations in $n < m$ unknowns. The method produces, upon convergence, the least-square approximation of the given system. Notice that, although the system of interest is overdetermined, it is consistent, and hence, its least-square approximation is, in fact, a solution of the IDP. However, because of the nonlinearity of the equations, multiple solutions are to be expected.

First, let us analyze the structure of the intermediate expression of Eq. (11). To do this, we first look at the product of two ERP, as shown in Eq. (9). Since the expression for $\boldsymbol{\eta}_{ij}$ is bilinear in $\boldsymbol{\eta}_i$ and $\boldsymbol{\eta}_j$, the RHS of Eq. (9) can be written as the product of the Jacobian of $\boldsymbol{\eta}_{ij}$ w.r.t. $\boldsymbol{\eta}_j$ times $\boldsymbol{\eta}_j$:

$$\boldsymbol{\eta}_{ij} = \frac{\partial \boldsymbol{\eta}_{ij}}{\partial \boldsymbol{\eta}_j}\boldsymbol{\eta}_j \tag{14}$$

the above 4×4 Jacobian matrix being readily calculated from Eqs. (8) to (9) as

$$\frac{\partial \boldsymbol{\eta}_{ij}}{\partial \boldsymbol{\eta}_j} = \begin{bmatrix} r_i\mathbf{1} + \text{CPM}(\mathbf{r}_i) & \mathbf{r}_i \\ -\mathbf{r}_i^T & r_i \end{bmatrix} \tag{15}$$

[4] It is noteworthy that this form is obtained under the assumption that $-\pi \le \alpha_i \le \pi$ and $-\pi \le \theta_i \le \pi$. Otherwise, this expression may be subject to a sign change of all its components. However, it can be readily verified that $\boldsymbol{\eta}_i$ and $-\boldsymbol{\eta}_i$ represent one and the same orientation, as a change of the sign of $\boldsymbol{\eta}_i$ is equivalent to changing θ_i to $\theta_i + 2\pi$, which represents the same angle as θ_i for our purpose.

[5] $\text{CPM}(\mathbf{r}_i) \equiv \partial(\mathbf{r}_i \times \mathbf{v})/\partial\mathbf{v},\ \forall \mathbf{v} \in \mathbb{R}^3$

The above matrix allows us to write the Euler product of two four-dimensional Euler vectors as the product of a matrix times a vector, which is a more familiar operation. Hence, we refer to the matrix of Eq. (15) as the Euler-Product Matrix[6] (EPM) of $\boldsymbol{\eta}_i$, i.e.,

$$\mathbf{H}_i \equiv \mathrm{EPM}(\boldsymbol{\eta}_i) \equiv \begin{bmatrix} r_i\mathbf{1} + \mathrm{CPM}(\mathbf{r}_i) & \mathbf{r}_i \\ -\mathbf{r}_i^T & r_i \end{bmatrix} \tag{16}$$

and hence, $\boldsymbol{\eta}_{ij}$ can be expressed as

$$\boldsymbol{\eta}_{ij} = \mathbf{H}_i \boldsymbol{\eta}_j \tag{17}$$

Correspondingly, $\boldsymbol{\eta}_i$ is the ERP vector of $\mathrm{EPM}(\boldsymbol{\eta}_i)$, the "erp" operator of $\mathbf{H}_i$ being defined as[7]

$$\mathrm{erp}\,(\mathbf{H}_i) = \boldsymbol{\eta}_i \tag{18}$$

Moreover, since the matrix product in Eq. (10) is associative, the Euler product is bound to be associative as well. Now it is apparent that Eq. (11) can be written as

$$\mathbf{H}_1\mathbf{H}_2 \ldots \mathbf{H}_6 = \mathbf{H}_0 \tag{19}$$

Apparently, both $\mathrm{EPM}(\cdot)$ and $\mathrm{erp}(\cdot)$ are linear operators; furthermore, all the components of $\boldsymbol{\eta}_i$ and $\mathrm{EPM}(\boldsymbol{\eta}_i)$ are linear in $\sin(\theta_i/2)$ and $\cos(\theta_i/2)$. Hence, $\mathrm{EPM}(\boldsymbol{\eta}_1 \otimes \ldots \boldsymbol{\eta}_6)$, or $\boldsymbol{\eta}_1 \otimes \ldots \boldsymbol{\eta}_6$ for that matter, is multi-linear in $\{\sin(\theta_i/2)\}_1^6$ and $\{\cos(\theta_i/2)\}_1^6$. Moreover,

$$\frac{\partial \boldsymbol{\eta}_i}{\partial \theta_i} = \frac{1}{2}\begin{bmatrix} -s_{\bar{\alpha}_i}s_{\bar{\theta}_i} \\ s_{\bar{\alpha}_i}c_{\bar{\theta}_i} \\ c_{\bar{\alpha}_i}c_{\bar{\theta}_i} \\ -c_{\bar{\alpha}_i}s_{\bar{\theta}_i} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} s_{\bar{\alpha}_i}\cos\dfrac{\theta_i+\pi}{2} \\ s_{\bar{\alpha}_i}\sin\dfrac{\theta_i+\pi}{2} \\ c_{\bar{\alpha}_i}\sin\dfrac{\theta_i+\pi}{2} \\ c_{\bar{\alpha}_i}\cos\dfrac{\theta_i+\pi}{2} \end{bmatrix} \tag{20}$$

Compared with Eq. (4), it is apparent that $2\partial\boldsymbol{\eta}_i/\partial\theta_i$ represents the ERP of a rotation about the Z_i axis of the ith frame, through an angle of $\theta_i + \pi$. Furthermore, only $\boldsymbol{\eta}_i$ contains θ_i, for $i = 1, \ldots, 6$. As $\boldsymbol{\eta}_1 \otimes \ldots \boldsymbol{\eta}_6$ is multi-linear, as stated above, we have

$$\frac{\partial \mathbf{f}_R}{\partial \theta_i} = \frac{\partial(\boldsymbol{\eta}_1 \otimes \ldots \otimes \boldsymbol{\eta}_6)}{\partial \theta_i} = \frac{1}{2}\boldsymbol{\eta}_1 \otimes \ldots \otimes \boldsymbol{\eta}_{i-1} \otimes \left(2\frac{\partial \boldsymbol{\eta}_i}{\partial \theta_i}\right) \otimes \boldsymbol{\eta}_{i+1} \otimes \ldots \otimes \boldsymbol{\eta}_6 \tag{21}$$

Moreover, if we define $\overline{\boldsymbol{\eta}}_i$ as the ERP of $\mathbf{Q}_i^T$, namely, $\bar{\boldsymbol{\eta}}_i = [-\mathbf{r}_i^T, r_i]^T$, it is readily verified that

$$\boldsymbol{\pi} \equiv \boldsymbol{\eta}_i \otimes \overline{\boldsymbol{\eta}}_i = \overline{\boldsymbol{\eta}}_i \otimes \boldsymbol{\eta}_i = [0, 0, 0, 1]^T, \quad i = 1, \ldots, 6 \tag{22}$$

whose rightmost side is the ERP array of the 3×3 identity matrix, which can also be regarded as a rotation about an arbitrary axis through an angle equal to 0. Therefore, $\boldsymbol{\pi}$ can be regarded, in turn, as the identity element of the set of Euler vectors, its EPM being the 4×4 identity matrix. By the same token, $\boldsymbol{\pi}$ is the ERP array of the 3×3 identity matrix. Now, Eq. (21) can be rewritten as

$$\frac{\partial \mathbf{f}_R}{\partial \theta_i} = \frac{1}{2}(\boldsymbol{\eta}_1 \otimes \ldots \otimes \boldsymbol{\eta}_{i-1}) \otimes \left(2\frac{\partial \boldsymbol{\eta}_i}{\partial \theta_i} \otimes \overline{\boldsymbol{\eta}}_i\right) \otimes (\overline{\boldsymbol{\eta}}_{i-1} \otimes \ldots \otimes \overline{\boldsymbol{\eta}}_1) \otimes (\boldsymbol{\eta}_1 \otimes \ldots \otimes \boldsymbol{\eta}_6) \tag{23}$$

Next, the factor in the second pair of parentheses of the RHS of Eq. (23), denoted $\boldsymbol{\delta}_i$, becomes

$$\boldsymbol{\delta}_i \equiv 2\frac{\partial \boldsymbol{\eta}_i}{\partial \theta_i} \otimes \overline{\boldsymbol{\eta}}_i = [0, 0, 1, 0]^T \tag{24}$$

which can be *regarded as* the Euler vector of a rotation about the Z_i axis of the ith frame through an angle of π, *viewed in the ith frame*. Hence,

$$\frac{\partial \mathbf{f}_R}{\partial \theta_i} = \frac{1}{2}(\boldsymbol{\eta}_1 \otimes \ldots \otimes \boldsymbol{\eta}_{i-1}) \otimes \boldsymbol{\delta}_i \otimes (\overline{\boldsymbol{\eta}}_{i-1} \otimes \ldots \otimes \overline{\boldsymbol{\eta}}_1) \otimes (\boldsymbol{\eta}_i \otimes \ldots \otimes \boldsymbol{\eta}_6) = \frac{1}{2}(\boldsymbol{\lambda}_{i-1} \otimes \boldsymbol{\delta}_i \otimes \overline{\boldsymbol{\lambda}}_{i-1}) \otimes \boldsymbol{\lambda}_6, \; i = 1, \ldots, 6 \tag{25}$$

where $\boldsymbol{\lambda}_{i-1}$, defined in Eq. (13), is already available from the previous calculations, for $i = 2, \ldots, 7$, and $\boldsymbol{\lambda}_0 \equiv [0, 0, 0, 1]^T$, which is identical to vector $\boldsymbol{\pi}$ of Eq. (22). Notice that the term in parentheses in the rightmost-hand side of Eq. (25) is the Euler vector of the product $\mathbf{Q}_{\lambda_{i-1}}\mathbf{D}_i\mathbf{Q}_{\lambda_{i-1}}^T$, with

$$\mathbf{Q}_{\lambda_i} \equiv \mathbf{Q}_1\mathbf{Q}_2 \ldots \mathbf{Q}_i, \quad i = 1, \ldots, 6 \tag{26}$$

[6] as it mimics the cross-product matrix of three-dimensional vectors.

[7] This operator plays the role of the *axial vector* of 3×3 matrices.

which is a *similarity transformation* of $\mathbf{D}_i$, denoting the *unique* proper orthogonal matrix whose Euler vector is $\boldsymbol{\delta}_i$. Let $\boldsymbol{\lambda}_i = [\lambda_{i,1}, \lambda_{i,2}, \lambda_{i,3}, \lambda_{i,4}]^T$, which has already been calculated in the previous steps. Then, using a formula similar to Eq. (6), $\mathbf{Q}_{\lambda_i}$ can be calculated component-wise as

$$\mathbf{Q}_{\lambda_i} = \begin{bmatrix} \lambda_{i,1}^2 - \lambda_{i,2}^2 - \lambda_{i,3}^2 + \lambda_{i,4}^2 & 2\lambda_{i,1}\lambda_{i,2} - 2\lambda_{i,3}\lambda_{i,4} & 2(\lambda_{i,1}\lambda_{i,3} + \lambda_{i,2}\lambda_{i,4}) \\ 2(\lambda_{i,1}\lambda_{i,2} + \lambda_{i,3}\lambda_{i,4}) & -\lambda_{i,1}^2 + \lambda_{i,2}^2 - \lambda_{i,3}^2 + \lambda_{i,4}^2 & 2\lambda_{i,2}\lambda_{i,3} - 2\lambda_{i,1}\lambda_{i,4} \\ 2\lambda_{i,1}\lambda_{i,3} - 2\lambda_{i,2}\lambda_{i,4} & 2(\lambda_{i,2}\lambda_{i,3} + \lambda_{i,1}\lambda_{i,4}) & -\lambda_{i,1}^2 - \lambda_{i,2}^2 + \lambda_{i,3}^2 + \lambda_{i,4}^2 \end{bmatrix} \tag{27}$$

Further, from the rules of the Euler product,

$$\boldsymbol{\lambda}_{i-1} \otimes \boldsymbol{\delta}_i \otimes \overline{\boldsymbol{\lambda}}_{i-1} = [2(\lambda_{i-1,1}\lambda_{i-1,3} + \lambda_{i-1,2}\lambda_{i-1,4}), 2(\lambda_{i-1,2}\lambda_{i-1,3} - \lambda_{i-1,1}\lambda_{i-1,4}), -\lambda_{i-1,1}^2 - \lambda_{i-1,2}^2 + \lambda_{i-1,3}^2 + \lambda_{i-1,4}^2, 0]^T \tag{28}$$

whose last entry vanishes, while its first three are nothing but the entries of the last column of $\mathbf{Q}_{\lambda_{i-1}}$, with $\mathbf{Q}_{\lambda_i}$ defined in Eq. (27). This makes sense, because $\boldsymbol{\delta}_i$ represents the ERP of a rotation whose vector part is a unit vector parallel to the Z_i axis, viewed in the ith frame. Hence, the similarity transformation of Eq. (25) represents $\boldsymbol{\delta}_i$ in the base frame, namely,

$$\boldsymbol{\lambda}_{i-1} \otimes \boldsymbol{\delta}_i \otimes \overline{\boldsymbol{\lambda}}_{i-1} = \begin{bmatrix} \mathbf{Q}_1 \ldots \mathbf{Q}_{i-1}\mathbf{k}_i \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_{\lambda_{i-1}}\mathbf{k}_i \\ 0 \end{bmatrix} \tag{29}$$

with $\mathbf{k}_i$ representing the unit vector parallel to Z_i—i.e., $[0, 0, 1]^T$—viewed in the ith frame. Hence, $\mathbf{Q}_{\lambda_{i-1}}\mathbf{k}_i = [\mathbf{k}_i]_1$, the subscript outside the square brackets indicating that the vector is viewed in frame $\mathcal{F}_1$. Moreover, Eq. (25) can be rewritten as

$$\frac{\partial \mathbf{f}_R}{\partial \theta_i} = \frac{1}{2}\begin{bmatrix} \mathbf{Q}_{\lambda_{i-1}}\mathbf{k}_i \\ 0 \end{bmatrix} \otimes \boldsymbol{\lambda}_6 = \frac{1}{2}\begin{bmatrix} [\mathbf{k}_i]_1 \\ 0 \end{bmatrix} \otimes \boldsymbol{\lambda}_6 \tag{30}$$

which is the gradient sought.

In summary, the calculation of $\partial(\boldsymbol{\eta}_1 \otimes \ldots \otimes \boldsymbol{\eta}_6)/\partial\theta_i$ requires the derivation of $\mathbf{Q}_{\lambda_i}$ from $\boldsymbol{\lambda}_i$, the latter already available from previous calculations, plus one Euler product with $\boldsymbol{\lambda}_6$. Once this expression is available, for $i = 1, \ldots, 6$, the computation of the gradient $\partial\mathbf{f}_R/\partial\boldsymbol{\theta}$ is completed.

3. The translation problem

Next, we derive the translation equation and the corresponding gradient. The former can be readily obtained upon dualizing Eq. (11). The general expression for the dual function of a dual argument—see, for example, [25]—is[8]

$$\hat{f}(\hat{x}) = \hat{f}(x + \epsilon x_o) \equiv f(x) + \epsilon x_o \frac{df(x)}{dx} \tag{31}$$

Then, the dual form of Eq. (11) can be obtained upon introducing

$$\hat{\theta}_i = \theta_i + \epsilon\ b_i, \quad \hat{\alpha}_i = \alpha_i + \epsilon\ a_i \tag{32}$$

We thus need expressions for the harmonic functions of a dual argument. These follow from Eqs. (31) to (32):

$$\sin(\theta_i + \epsilon b_i) = \sin\left(\frac{\theta_i}{2}\right) + \epsilon\frac{1}{2}b_i\cos\left(\frac{\theta_i}{2}\right), \quad \cos(\theta_i + \epsilon b_i) = \cos\left(\frac{\theta_i}{2}\right) - \epsilon\frac{1}{2}b_i\sin\left(\frac{\theta_i}{2}\right) \tag{33}$$

and

$$\sin(\alpha_i + \epsilon a_i) = \sin\left(\frac{\alpha_i}{2}\right) + \epsilon\frac{1}{2}a_i\cos\left(\frac{\alpha_i}{2}\right), \quad \cos(\alpha_i + \epsilon a_i) = \cos\left(\frac{\alpha_i}{2}\right) - \epsilon\frac{1}{2}b_i\sin\left(\frac{\alpha_i}{2}\right) \tag{34}$$

Then, the dual form of $\boldsymbol{\eta}_i$ is

$$\hat{\boldsymbol{\eta}}_i = \boldsymbol{\eta}_i + \epsilon\,\boldsymbol{\eta}_{io} \tag{35}$$

with $\boldsymbol{\eta}_{io}$ obtained upon dualizing Eq. (4), which yields

$$\hat{\boldsymbol{\eta}}_i = \begin{bmatrix} s_{\bar{\alpha}_i}c_{\bar{\theta}_i} \\ s_{\bar{\alpha}_i}s_{\bar{\theta}_i} \\ c_{\bar{\alpha}_i}s_{\bar{\theta}_i} \\ c_{\bar{\alpha}_i}c_{\bar{\theta}_i} \end{bmatrix} + \frac{1}{2}\epsilon\begin{bmatrix} a_i c_{\bar{\alpha}_i}c_{\bar{\theta}_i} - b_i s_{\bar{\alpha}_i}s_{\bar{\theta}_i} \\ a_i c_{\bar{\alpha}_i}s_{\bar{\theta}_i} + b_i s_{\bar{\alpha}_i}c_{\bar{\theta}_i} \\ -a_i s_{\bar{\alpha}_i}s_{\bar{\theta}_i} + b_i c_{\bar{\alpha}_i}c_{\bar{\theta}_i} \\ -a_i s_{\bar{\alpha}}c_{\bar{\theta}_i} - b_i c_{\bar{\alpha}}s_{\bar{\theta}_i} \end{bmatrix} \tag{36}$$

where it is apparent that

$$\boldsymbol{\eta}_{io} = a_i\frac{\partial\boldsymbol{\eta}_i}{\partial\alpha_i} + b_i\frac{\partial\boldsymbol{\eta}_i}{\partial\theta_i} \tag{37}$$

[8] One way of interpreting the expression for $\hat{f}(\hat{x})$ is by regarding the formula as the series expansion of $\hat{f}(x + \epsilon x_o)$ and recalling that $\epsilon^2 = 0$.

Moreover, the dual form of Eq. (11) can be written as

$$\boldsymbol{\lambda}_6 + \epsilon\boldsymbol{\lambda}_{6o} = \boldsymbol{\eta}_1 \otimes \ldots \otimes \boldsymbol{\eta}_i \otimes \ldots \otimes \boldsymbol{\eta}_6 + \epsilon \sum_{i=1}^{6} \boldsymbol{\eta}_1 \otimes \ldots \otimes \boldsymbol{\eta}_{i-1} \otimes \boldsymbol{\eta}_{io} \otimes \boldsymbol{\eta}_{i+1} \otimes \ldots \otimes \boldsymbol{\eta}_6 = \boldsymbol{\eta}_0 + \epsilon\boldsymbol{\eta}_{0o} \tag{38}$$

from which the translation equation is readily derived:

$$\begin{aligned} \boldsymbol{\lambda}_{6o} &\equiv \sum_{i=1}^{6} \boldsymbol{\eta}_1 \otimes \ldots \otimes \boldsymbol{\eta}_{i-1} \otimes \boldsymbol{\eta}_{io} \otimes \boldsymbol{\eta}_{i+1} \otimes \ldots \otimes \boldsymbol{\eta}_6 \\ &= \sum_{i=1}^{6} b_i \boldsymbol{\eta}_1 \otimes \ldots \otimes \frac{\partial \boldsymbol{\eta}_i}{\partial \theta_i} \otimes \ldots \otimes \boldsymbol{\eta}_6 + \sum_{i=1}^{6} a_i \boldsymbol{\eta}_1 \otimes \ldots \otimes \frac{\partial \boldsymbol{\eta}_i}{\partial \alpha_i} \otimes \ldots \otimes \boldsymbol{\eta}_6 = \boldsymbol{\eta}_{0o} \end{aligned} \tag{39}$$

The translation equation (39) contains redundant information on the translation of the EE, thereby adding numerical robustness to the system of equations. Using the same idea for the calculation of the gradient of the orientation part, we have

$$\frac{\partial \boldsymbol{\lambda}_6}{\partial \alpha_i} = \frac{1}{2}(\boldsymbol{\eta}_1 \otimes \ldots \otimes \boldsymbol{\eta}_{i-1}) \otimes \boldsymbol{\rho}_i \otimes (\overline{\boldsymbol{\eta}}_{i-1} \otimes \ldots \otimes \overline{\boldsymbol{\eta}}_1) \otimes (\boldsymbol{\eta}_i \otimes \ldots \otimes \boldsymbol{\eta}_6) = \frac{1}{2}(\boldsymbol{\lambda}_{i-1} \otimes \boldsymbol{\rho}_i \otimes \overline{\boldsymbol{\lambda}}_{i-1}) \otimes \boldsymbol{\lambda}_6 \tag{40}$$

where

$$\boldsymbol{\rho}_i = 2\frac{\partial \boldsymbol{\eta}_i}{\partial \alpha_i} \otimes \overline{\boldsymbol{\eta}}_i = [\cos\theta_i, \sin\theta_i, 0, 0]^T \tag{41}$$

Further, upon recalling the multi-linearity property of the Euler product, as per Eq. (19), Eq. (39) simplifies to

$$\frac{1}{2}\left(\sum_{i=1}^{6}\begin{bmatrix} a_i \mathbf{Q}_{\lambda_{i-1}} \mathbf{g}_i + b_i \mathbf{Q}_{\lambda_{i-1}} \mathbf{k}_i \\ 0 \end{bmatrix}\right) \otimes \boldsymbol{\lambda}_6 - \boldsymbol{\eta}_{0o} = \mathbf{0} \tag{42}$$

with $\mathbf{g}_i \equiv [\cos\theta_i,\ \sin\theta_i,\ 0,\ ]^T$. Finally, the translation equation can be cast in the form

$$\mathbf{f}_T \equiv \frac{1}{2}\left(\sum_{i=1}^{6}\begin{bmatrix} \mathbf{Q}_{\lambda_{i-1}} \mathbf{f}_i \\ 0 \end{bmatrix}\right) \otimes \boldsymbol{\lambda}_6 - \boldsymbol{\eta}_{0o} = \frac{1}{2}\begin{bmatrix} \sum_{i=1}^{6} (\mathbf{Q}_{\lambda_{i-1}} \mathbf{f}_i) \\ 0 \end{bmatrix} \otimes \boldsymbol{\lambda}_6 - \boldsymbol{\eta}_{0o} = \mathbf{0} \tag{43}$$

with $\mathbf{f}_i \equiv [a_i\cos\theta_i,\ a_i\sin\theta_i,\ b_i]^T$, which is nothing but $[\mathbf{a}_i]_i$ in the DH notation [10]. Hence, $\mathbf{Q}_{\lambda_{i-1}}\mathbf{f}_i$ represents $[\mathbf{a}_i]_1$. Then, Eq. (43) can be further simplified to

$$\mathbf{f}_T \equiv \frac{1}{2}\begin{bmatrix} \sum_{i=1}^{6} [\mathbf{a}_i]_1 \\ 0 \end{bmatrix} \otimes \boldsymbol{\lambda}_6 - \boldsymbol{\eta}_{0o} = \frac{1}{2}\begin{bmatrix} \mathbf{o}_7 \\ 0 \end{bmatrix} \otimes \boldsymbol{\lambda}_6 - \boldsymbol{\eta}_{0o} = \mathbf{0} \tag{44}$$

with $\mathbf{o}_i$ representing the position vector of the origin O_i of the ith frame, viewed in the base frame, for $i = 1, \ldots, 7$. Apparently, all the terms in Eq. (44) have been calculated in the orientation equation, except for $\mathbf{o}_7$, which can be calculated recursively:

$$\mathbf{o}_1 = \mathbf{0},\ \ \mathbf{o}_{i+1} = \mathbf{o}_i + [\mathbf{a}_i]_1,\ \ i = 1, \ldots, 6 \tag{45}$$

the expressions for $\mathbf{a}_i$ in the base frame being calculated as

$$[\mathbf{a}_i]_1 = \mathbf{Q}_{\lambda_i}[\mathbf{a}_i]_{i+1},\ \ i = 1, \ldots, 6 \tag{46}$$

with

$$[\mathbf{a}_i]_{i+1} = [a_i,\ b_i \sin\alpha_i,\ b_i \cos\alpha_i]^T \tag{47}$$

which is constant for a six-revolute (6R) robot and a given set of DH parameters, thereby completing the formulation of the displacement equations for a 6R robot.

It is noteworthy that Eq. (44) yields an interesting result: for an arbitrary displacement characterized by the rotation matrix $\mathbf{Q}$—with the ERP $\boldsymbol{\eta} = [\mathbf{r}^T, r]^T$ —and the translation $\mathbf{u}$, the dual Euler vector turns out to be

$$\hat{\boldsymbol{\eta}} = \boldsymbol{\eta} + \epsilon\frac{\|\mathbf{u}\|_2}{2}\begin{bmatrix} \mathbf{u}/\|\mathbf{u}\|_2 \\ 0 \end{bmatrix} \otimes \boldsymbol{\eta} \tag{48}$$

which matches the corresponding formula of the dual quaternion [26] and aligns with the isomorphism between the ERP and the quaternion. It is noteworthy that $[\mathbf{u}^T/\|\mathbf{u}\|_2,\ 0]^T$ can be regarded as the Euler vector of a rotation about an axis parallel to $\mathbf{u}$ through a half turn; hence, the dual part of the above vector can be regarded as the Euler vector of the product of two rotations, multiplied by a scalar with units of length. The dual form of the ERP given above applies to *any arbitrary parameterization* of the rotation matrix. What this means is that this formula is applicable, regardless of the rotation representation of choice, e.g., linear invariants [7], ERP, Euler angles, etc. Eq. (48) provides a means to derive $\boldsymbol{\eta}_{0o}$.

3.1. The gradient of the translation equation

Next, the gradient of the translation equation is obtained upon dualizing the gradient of the orientation equation, as per Eq. (30), namely,

$$\frac{\partial \mathbf{f}_T}{\partial \theta_i} = \frac{1}{2}\begin{bmatrix} \mathbf{Q}_{\lambda_{i-1}o}\mathbf{k}_i \\ 0 \end{bmatrix} \otimes \boldsymbol{\lambda}_6 + \frac{1}{2}\begin{bmatrix} \mathbf{Q}_{\lambda_{i-1}}\mathbf{k}_i \\ 0 \end{bmatrix} \otimes \boldsymbol{\lambda}_{6o} \tag{49}$$

where $\boldsymbol{\lambda}_{6o}$ is nothing but the first term of Eq. (43), while

$$\mathbf{Q}_{\lambda_{i-1}o}\mathbf{k}_i = \mathrm{CPM}(\mathbf{o}_i)\,\mathbf{Q}_{\lambda_{i-1}}\mathbf{k}_i = \mathbf{O}_i[\mathbf{k}_i]_1 \tag{50}$$

with $\mathbf{O}_i = \mathrm{CPM}(\mathbf{o}_i)$. Hence,

$$\frac{\partial \mathbf{f}_T}{\partial \theta_i} = \frac{1}{2}\begin{bmatrix} \mathbf{O}_i[\mathbf{k}_i]_1 \\ 0 \end{bmatrix} \otimes \boldsymbol{\lambda}_6 + \frac{1}{2}\begin{bmatrix} [\mathbf{k}_i]_1 \\ 0 \end{bmatrix} \otimes \boldsymbol{\lambda}_{6o} \tag{51}$$

To conclude, we have derived eight equations with six unknowns, namely,

$$\mathbf{f}_R(\boldsymbol{\theta}) \equiv \boldsymbol{\eta}_1 \otimes \boldsymbol{\eta}_2 \otimes \ldots \otimes \boldsymbol{\eta}_6 - \boldsymbol{\eta}_0 = \mathbf{0} \in \mathbb{R}^4, \quad \mathbf{f}_T(\boldsymbol{\theta}) \equiv \frac{1}{2}\begin{bmatrix} \mathbf{o}_7 \\ 0 \end{bmatrix} \otimes \boldsymbol{\lambda}_6 - \boldsymbol{\eta}_{0o} = \mathbf{0} \in \mathbb{R}^4 \tag{52}$$

The two foregoing equations are now assembled in the form

$$\mathbf{f}(\boldsymbol{\theta}) = \mathbf{0} \in \mathbb{R}^8 \tag{53}$$

which is a system of eight scalar equations in six unknowns, the entries of $\boldsymbol{\theta}$. The system is, thus, overdetermined, yet consistent. Overdeterminacy is good news because it adds robustness, to roundoff errors, to the computed solution. By this we mean that redundancy enhances the numerical conditioning of a system of equations, when compared to its non-redundant counterpart, if it is possible to formulate the problem as such. In our case, it is, as seen in the literature [27]. This point is made clear in Section 4.

4. Numerical solution

Solving numerically the system of equations (53) is challenging because of the nonlinearity of the equations. Here we describe the solution based on the Newton-Gauss method, an iterative procedure based on the linearization of the eight nonlinear equations at a given value $\boldsymbol{\theta}_k$ of the array of six joint angles. For this reason, the 8×6 gradient matrix of the system (53) is needed. The gradient is shown below, in terms of two blocks:

$$\mathbf{J} = \begin{bmatrix} \mathbf{R} \\ \mathbf{T} \end{bmatrix} \tag{54}$$

where $\mathbf{R}$ and $\mathbf{T}$ are the 4×6 sub-gradients of the rotation and the translation equations, respectively:

$$\mathbf{R} = \frac{\partial \mathbf{f}_R}{\partial \boldsymbol{\theta}}, \quad \mathbf{T} = \frac{\partial \mathbf{f}_T}{\partial \boldsymbol{\theta}} \in \mathbb{R}^{4\times 6} \tag{55}$$

the ith column of $\mathbf{R} \in \mathbb{R}^{4\times 6}$ being given by Eq. (30), while the ith column of $\mathbf{T} \in \mathbb{R}^{4\times 6}$ given by Eq. (51), for $i = 1, \ldots, 6$. The gradient displayed in Eq. (54) is related to the putative[9] 6×6 "Jacobian" that maps the array of joint rates into the twist of the EE, a six-dimensional array including two three-dimensional blocks, one the angular velocity—not a time-derivative—one the velocity of a landmark point of the EE, O_7 in our case. If we call the putative Jacobian $\mathbf{K}$ the relation between the two can be expressed in the form [6]

$$\mathbf{J} = \mathbf{L}\mathbf{K} \tag{56}$$

The foregoing relation is not fully derived here for brevity. It is illustrated with the rotation subtask: Let $\mathbf{K}_R$ denote the 3×6 sub-Jacobian of $\mathbf{K}$ that maps the six-dimensional array of joint rates into the angular velocity. A relation between the angular velocity $\boldsymbol{\omega}$ of the EE and the time-rate of change $\dot{\boldsymbol{\eta}}$ is taken from the literature [6]:

$$\dot{\boldsymbol{\eta}} = \mathbf{H}\boldsymbol{\omega}, \quad \mathbf{H} = \frac{1}{2}\begin{bmatrix} \cos(\phi/2)\mathbf{1} - \sin(\phi/2)\mathbf{E} \\ -\sin(\phi/2)\mathbf{e}^T \end{bmatrix} \tag{57}$$

As well, the angular velocity vector $\boldsymbol{\omega}$ can be obtained by the mapping of the six-dimensional array $\dot{\boldsymbol{\theta}}$ induced by $\mathbf{K}_R$:

$$\boldsymbol{\omega} = \mathbf{K}_R\dot{\boldsymbol{\theta}} \tag{58}$$

[9] That is, a pseudo-Jacobian, because it is not a matrix of partial derivatives.

and hence, $\dot{\boldsymbol{\eta}}$ can be expressed as

$$\dot{\boldsymbol{\eta}} = \mathbf{H}\mathbf{K}_R\dot{\boldsymbol{\theta}} \tag{59}$$

A similar relation for the translation can be readily derived upon *dualization* of relation (59), and extracting its dual part. This relation is straightforward, and hence, left out. Upon assembling the rotational and translational parts of the above relations, an expression for $\mathbf{L}$ in Eq. (56) follows.

The conclusion here is that, no matter what parameterization is used, a singular $\mathbf{K}$ will lead to a rank-deficient $\mathbf{J}$. However, the effect of $\mathbf{L}$ on a non-singular $\mathbf{K}$ is a lower condition number by virtue of redundancy, as shown below.

Now, using the Newton-Gauss method the increment $\Delta\boldsymbol{\theta}_k$ is computed at the kth iteration from the equation

$$\mathbf{J}_k\Delta\boldsymbol{\theta}_k = -\mathbf{f}_k \tag{60}$$

where $\mathbf{J}_k \equiv \mathbf{J}(\boldsymbol{\theta}_k)$ and $\mathbf{f}_k \equiv \mathbf{f}(\boldsymbol{\theta}_k)$. Moreover, the system (60) being overdetermined, in principle does not admit a solution, but rather an *optimum approximation* in the least-square sense. In our case, since the eight equations are consistent, the least-square approximation is, in fact, *the solution* of Eq. (60). The foregoing approximation can be expressed *symbolically* by resorting to the *left Moore-Penrose generalized inverse*, represented here as $\mathbf{J}_k^I$, namely,

$$\Delta\boldsymbol{\theta}_k = -\mathbf{J}_k^I\mathbf{f}_k, \quad \mathbf{J}_k^I \equiv (\mathbf{J}_k\mathbf{J}_k^T)^{-1}\mathbf{J}_k^T \tag{61}$$

which is then added to the current value $\boldsymbol{\theta}_k$, to improve the approximation:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \Delta\boldsymbol{\theta}_k \tag{62}$$

the procedure stopping when a criterion, $\|\Delta\boldsymbol{\theta}_k\| \le \epsilon$ is satisfied, for a small-enough value of the prescribed tolerance ϵ.

To be true, the foregoing expression for $\Delta\boldsymbol{\theta}_k$ is a *formula*, not a computational statement. Indeed, the verbatim application of the above expression for $\mathbf{J}_k^I$ from Eq. (61), reportedly used in many cases, is strongly recommended against. The reason lies in the need to invert the product $\mathbf{J}_k\mathbf{J}_k^T$, whose condition number is, roughly, the square of that of $\mathbf{J}_k$. What is done when solving linear least-square problems is to resort to the QR-decomposition[10] of $\mathbf{J}_k$, in the form [28]

$$\mathbf{J}_k = \tilde{\mathbf{Q}}\tilde{\mathbf{R}} \tag{63}$$

where $\tilde{\mathbf{Q}}$, in our case, is an 8×8 orthogonal matrix, the product of eight 8×8 orthogonal matrices, and $\tilde{\mathbf{R}}$ is an upper-triangular 8×6 matrix. Matrix $\tilde{\mathbf{Q}}$ is, in our case, the product of eight *Householder reflections* [28], i.e., eight improper orthogonal matrices (their determinant is -1). The reason why reflections are used in this decomposition is that a) they preserve the condition number of the matrix of interest, $\mathbf{J}_k$ in our case, and b) they are extremely economical to find[11]. The accuracy of the computation of $\Delta\boldsymbol{\theta}_k$ from Eq. (61) thus depends on the condition number of $\mathbf{J}_k$.

Had a system of six equations in six unknowns been adopted at the outset, derived, for example, from the representation of the rotation matrix by the three Euler angles, then $\mathbf{J}_k$ of Eq. (60) would be of 6×6, $\Delta\boldsymbol{\theta}_k$ then being computed from the LU-decomposition of the 6×6 Jacobian $\mathbf{J}_k$. The claim that the computation of $\Delta\boldsymbol{\theta}_k$ is more robust with an overdetermined system of equations than otherwise is now justified with a generic example in which a system of n equations in n unknowns is augmented with a set of redundant equations. Let the first system be associated with a $n\times n$*non-singular* matrix $\mathbf{A}$ and a right-hand side $\mathbf{b}$. Roundoff error in the entries of $\mathbf{A}$ and $\mathbf{b}$ is accounted for by means of an extremely simple model, intended merely as illustration: as *additive* noise $\boldsymbol{\nu}$ of zero mean and *isotropic* covariance matrix $\sigma\mathbf{1}$, with $\mathbf{1}$ denoting the $n\times n$ identity matrix and $\sigma > 0$. In this model, $\mathbf{A}$ and $\mathbf{b}$ are assumed to be *deterministic*, i.e., known *perfectly*. In this vein, the given system, $\mathbf{Ax} = \mathbf{b}$ is now represented as

$$\mathbf{Ax} = \mathbf{b} + \boldsymbol{\nu} \tag{64}$$

The computed solution $\tilde{\mathbf{x}}$ to the equation $\mathbf{Ax} = \mathbf{b}$ is, of course, contaminated by noise, namely,

$$\tilde{\mathbf{x}} = \mathbf{A}^{-1}(\mathbf{b} + \boldsymbol{\nu}) \tag{65}$$

It is known [29] that the covariance matrix $\mathrm{cov}(\tilde{\mathbf{x}})$ of $\tilde{\mathbf{x}}$ is

$$\mathrm{cov}(\tilde{\mathbf{x}}) = \sigma^2(\mathbf{A}^T\mathbf{A})^{-1} \tag{66}$$

Now, if a set of $p > 0$ noisy, redundant but consistent equations $\mathbf{Bx} = \mathbf{c} + \boldsymbol{\xi}$ is added to the original system (65), the least-square approximation of the overdetermined system of $(n+p)$ equations in n unknowns thus resulting becomes, in terms of the *left Moore-Penrose generalized inverse* of the augmented $(n+p)\times n$ matrix,

$$\tilde{\mathbf{x}} = \left(\mathbf{C}^T\mathbf{C}\right)^{-1}\mathbf{C}^T(\mathbf{d} + \boldsymbol{\rho}) \tag{67}$$

which mimics Eq. (65). Moreover,

$$\mathbf{C} = \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix}\mathbf{b}\\ \mathbf{c}\end{bmatrix}, \quad \boldsymbol{\rho} = \begin{bmatrix}\boldsymbol{\nu}\\ \boldsymbol{\xi}\end{bmatrix} \tag{68}$$

[10] This decomposition is to rectangular (tall) matrices what the LU-decomposition is to square matrices, which is needed to solve determined systems of n linear equations in n unknowns.

[11] MATLAB has a function implementing what is known as the HR-decomposition of a tall matrix.

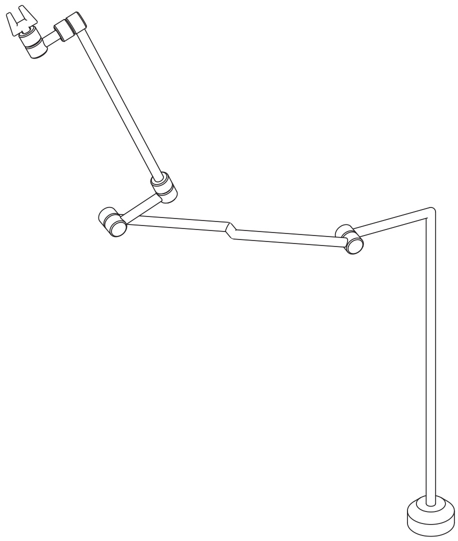

Fig. 1. The kinematic chain of the Fanuc Arc Mate S.

Table 1
The DH parameters of a given robot.

$\alpha_i\,(^\circ)$	$a_i\,(\mathrm{mm})$	$b_i\,(\mathrm{mm})$
90	200	810
0	600	0
90	130	−30
90	0	550
90	0	100
0	0	100

Now, the new covariance matrix $\mathrm{cov}(\tilde{\mathbf{x}})$ becomes

$$\mathrm{cov}(\tilde{\mathbf{x}}) = \sigma^2(\mathbf{C}^T\mathbf{C})^{-1} = \sigma^2(\mathbf{A}^T\mathbf{A} + \mathbf{B}^T\mathbf{B})^{-1} \tag{69}$$

Notice that, since the product $\mathbf{B}^T\mathbf{B}$ is at least positive-semidefinite, while $\mathbf{A}^T\mathbf{A}$ is necessarily positive-definite,[12] the norm—any norm—of the matrix in parentheses in Eq. (69) is larger than that of its counterpart in Eq. (66). The conclusion is that adding extra equations to the original determined system has the effect of reducing the norm of the covariance matrix of the computed solution, thereby increasing the accuracy of the computed results, which means robustness.

As a matter of fact, this property of redundant measurements—adding extra equations, while keeping the number of unknowns fixed—lies at the base of a technique known as *recursive least squares*, commonly used to calibrate instruments while conducting measurements automatically, in real time. The famous *hand-eye calibration problem* of the nineties led to many papers on the subject. In a paper published at the turn of the 21st century [30] it was shown that the problem can be solved to a great advantage using recursive least squares: while three measurements suffice in the absence of noise, redundant measurements are needed to cope with unavoidable noise. The cited paper records thousands of measurements and illustrates, with plots of the noisy measurements, the convergence to the actual values as the number of measurements increases. Another field of application of redundant measurements is *pose-estimation*, where the pose of a rigid object is determined by tracking a massive amount of points of the object, when, in the absence of noise, three points would suffice [31].

5. Case study

In this section we use a numerical example to verify the formulation proposed here. The Fanuc Arc Mate S series manipulator, sketched in Fig. 1, is used for illustration. Its inverse displacement problem has been investigated in a different approach elsewhere [6]; the results from the foregoing reference are used here for verification. The pertinent parameters are given in Table 1, from which it is apparent that this robot is not decoupled—its last three axes do not intersect—and hence, does not admit a closed-form IDP solution.

[12] We assumed at the outset that $\mathbf{A}$ is non-singular.

Before we implement the Newton-Gauss method, it is noteworthy that the equations for the rotation and translation parts bear different units. Hence, a characteristic length L [6] is introduced with the purpose of deriving a *homogeneous form* of the displacement model. Then, both sides of the four translation equations are divided by L, which makes the gradient matrix dimensionless. We refer to this process as *normalization*. The characteristic length for the robot under study was calculated in the foregoing reference as $L = 0.35123\text{m}$.

Next, the Newton-Gauss method is adopted to solve the IDP to verify the algorithm. The EE posture used in the same reference is recalled here, for purpose of comparison of the results,

$$\mathbf{Q}_0 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{o}_7 = \begin{bmatrix} 0.13 \\ 0.85 \\ 1.54 \end{bmatrix} \text{ m} \tag{70}$$

The EE pose as per Eq. (70), when transformed into the dual ERP, is given by

$$\boldsymbol{\eta}_0 = \begin{bmatrix} -0.5 \\ -0.5 \\ -0.5 \\ 0.5 \end{bmatrix}, \quad \boldsymbol{\eta}_{0o} = \begin{bmatrix} 0.205 \\ -0.14 \\ 0.565 \\ 0.63 \end{bmatrix} \text{ m} \tag{71}$$

Moreover, it is noteworthy that $\boldsymbol{\eta}_0$, $\boldsymbol{\eta}_{0o}$ and $-\boldsymbol{\eta}_0$, $-\boldsymbol{\eta}_{0o}$ —though bearing opposite directions—represent the same pose. Hence, when implementing the algorithm, we have to choose from these two directions of $\boldsymbol{\eta}_0$ properly at each iteration. This can be handled simply by redefining $\boldsymbol{\eta}_0$ as

$$\boldsymbol{\eta}_0 = \text{sgn}(\boldsymbol{\eta}_0^T \boldsymbol{\lambda}_6)\boldsymbol{\eta}_0 \tag{72}$$

at each iteration, where sgn $(\cdot)$ represents the *signum function* of $(\cdot)$, i.e.,

$$\text{sgn}\,(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases} \tag{73}$$

Once the direction of $\boldsymbol{\eta}_0$ is chosen properly, that of $\boldsymbol{\eta}_{0o}$ will be correspondingly correct. No modification is thus needed for $\boldsymbol{\eta}_{0o}$.

Next, three initial guesses are chosen based on one of the four solutions reported earlier using a *semi-graphical method* [6], namely,

$$\boldsymbol{\theta}_0 = [1.45501, 1.58781, -0.1397, 2.38164, -2.9731, 0.752836]^T \tag{74}$$

Three initial guesses are generated upon adding a perturbation $\Delta\boldsymbol{\theta}$ to $\boldsymbol{\theta}_0$, where $\Delta\boldsymbol{\theta}$ is generated as an array of six random values within the range of ± 0.5 (roughly $\pm 30°$), namely,

$$\begin{aligned} \Delta\boldsymbol{\theta}_1 &= [-0.310564, 0.464282, 0.237129, -0.345945, 0.219772, -0.269517]^T \\ \Delta\boldsymbol{\theta}_2 &= [0.158586, 0.488871, -0.327282, 0.426405, -0.397313, -0.266954]^T \\ \Delta\boldsymbol{\theta}_3 &= [0.0393227, 0.0591514, 0.114553, 0.122651, 0.0828967, -0.431772]^T \end{aligned} \tag{75}$$

This assignment is appropriate, considering that we target real-time applications, in which $\boldsymbol{\theta}$ for the previous pose of the EE is available; this value can be used as the initial guess for the IDP at the current pose of the EE, whose difference with the desired solution is usually small.

The convergence criterion is set to be $\|\Delta\boldsymbol{\theta}_k\|_\infty < \epsilon$, where $\Delta\boldsymbol{\theta}_k$ represents the step of the kth iteration, while $\|\cdot\|_\infty$ denotes the maximum (Chebyshev) norm of $(\cdot)$, and ϵ is the tolerance, which is set to 10^{-5} . Moreover, the maximum number of iteration is set as 50. It is shown that all three cases converge to $\boldsymbol{\theta}_0$, within 7, 7 and 5 iterations, respectively.

The same problem was solved with the Nelder–Mead method in order to compare its performance with the Newton Gauss method. The algorithm by Conn et. al. [32] is used here, with the standard choice of the coefficients, i.e.,

$$\gamma^s = \frac{1}{2},\ \delta^{ic} = -\frac{1}{2},\ \delta^{oc} = \frac{1}{2},\ \delta^r = 1,\ \delta^e = 2 \tag{76}$$

and the convergence criterion [32] is met when the Nelder-Mead *simplex* “diameter” is met with a tolerance ϵ. The number of iterations required for the three different initial guesses are 294, 348, 379, respectively, which are, roughly speaking, two orders of magnitude the corresponding numbers required by the Newton-Gauss method; moreover, it was found that, when the number of digits of significance is increased, the number of iterations for the Nelder–Mead method grows significantly, while for the Newton-Gauss method only a few additional iterations are needed.

Other initial guesses of the EE pose have also been tested for the Newton–Gauss method; the convergence is similar to the three foregoing tests. However, it was found that, when the initial guess is far from $\boldsymbol{\theta}_0$, the results may converge to other solutions, for example, when

$$\Delta\boldsymbol{\theta} = [-0.652443, -0.925664, 0.443442, -0.890078, -0.169481, -0.0644588]^T \tag{77}$$

the procedure converges to

$$\boldsymbol{\theta} = [1.49082, 0.281984, 2.67406, -3.06023, 1.75574, -0.0149941]^T \tag{78}$$

which corresponds to the fifth solution in the multi-cited reference [6]. This is unavoidable for numerical solutions; when the initial guess is far from any solution, the behavior is hard to predict. In fact, it is known, for example, that the boundary in the space of the unknown variables, between convergence and divergence, in the case of the related Newton–Raphson method, is a fractal [33]. However, in terms of real-time trajectory-tracking, the values of $\boldsymbol{\theta}$ for the previous pose of the EE can be used as the initial guess of the IDP for the current EE pose; hence, the error is bound to be small, he problem mentioned above hardly arising.

Next, another posture *near a singularity* is tested: The initial posture is chosen to be

$$\boldsymbol{\theta}_0 = [-3.1056, 2.20726, 2.73188, -2.6145, 0.00939723, -0.813694]^T \tag{79}$$

under which the condition number of the kinematics Jacobian is found to be 66792.0. The same $\Delta\boldsymbol{\theta}$ in Eq. (77) is used, the Newton-Gauss algorithm converging within 15, 17 and 16 iterations, respectively, while the Nelder Mead method with 873, 889, 933, respectively, which indicates the advantage of the Newton-Gauss method over the derivative-free Nelder-Mead method near a singularity as well.

It is noteworthy that when the difference between the initial guess and the exact solution, i.e., the array $\Delta\boldsymbol{\theta}$ —which can be characterized by its infinity norm, denoted $\Delta\theta$—is small enough, the Newton-Gauss algorithm works fairly well in most cases. For example, when $\Delta\theta$ is small, say 0.14 (8°), for the first point displayed in Eq. (74), 500 random initial guesses were generated, all converging to the correct solution within an average of 4.3 iterations. This value of $\Delta\theta$ is usually good enough for real-time control. For the second point displayed in Eq. (79), 496 tests out of 500 converge to the expected solution within an average of 13.3 iterations; three converge to other solutions with an average of 26 iterations, and one exceeds the maximum number of iterations, 50. It is noteworthy that when the given posture is near a singularity, the convergence is less likely than the case when the given posture is far from a singularity. Hence, multiple initial guesses near the neighborhood could be used to verify the result.

6. Conclusions

The need to count on a systematic, robust formulation of the numerical solution of the classical IDP associated with a six-revolute robot, motivated the work reported here. To achieve a systematic formulation of the problem, the dual Euler–Rodrigues parameters, a.k.a. quaternions, appear as an attractive alternative, as adopted in this paper. Numerical robustness, achieved by means of overdeterminacy, is needed in order to cope with round-off errors and other sources of noise. We follow here a commonly accepted engineering practice: in surveying and object-tracking, e.g., in determining the GPS-coordinates of a vehicle in motion, more observations than what is strictly needed come into play. The formulation proposed here, along with its solution algorithm, were implemented using computer algebra, which allows for its swift porting into numerical software.

Dedication

The authors dedicate this paper to Prof. Bernard Roth, a recognized contributor to the development of *mechanism and machine science*. Specifically, Prof. Roth's contributions to the solution of the challenging *inverse kinematics problem* (IKP), famously dubbed *the Mount Everest of the theory of machines and mechanisms* by Prof. Roth's mentor, the late Prof. Ferdinand Freudenstein, of Columbia University, are acknowledged here. The problem to which Freudenstein was referring is the input-output equation of the seven-link closed kinematic chain with seven revolute joints. Originally a purely theoretical problem—Freudenstein was referring to the conqueror of Mount Everest, Sir Edmund Percival Hillary's famous quote: *"Nobody climbs mountains for scientific reasons. Science is used to raise money for the expeditions, but you really climb for the hell of it."*—with the advent of robotics as a real-life technology of the utmost importance for the world economy, in the late seventies, the problem attracted the attention of many a roboticist of the times. Professor Roth, with his Ph.D. student Madhusan Raghavan, produced one of the first complete solutions to the problem in 1991. The other solution was reported, in the same year, by two doctoral students, Hongyou Lee and Christoph Woernle, with their mentor, Prof. Manfred Hiller.

To Bernie with love!

Acknowledgements

The first author would like to acknowledge the support from the McGill Engineering Doctoral Award (MEDA) (Fund Number 90025) and the Chinese Scholarship Council (CSC) (No. 201306020035) for their financial support. The second author acknowledges the support from the Institute of Mechanical Engineers (IMechE) Educational Award EAC 15/003. The third author acknowledges the support received from NSERC (Canada's Natural Sciences and Engineering Research Council) through Grant No. RGPIN-2015-03864, and from McGill University's James McGill Professorship (Fund Number 100711).

References

[1] S.L. Altmann, Hamilton, Rodrigues, and the quaternion scandal. what went wrong with one of the major mathematical discoveries of the nineteenth century, Math. Mag. 62 (5) (1989) 291–308.
[2] H. Cheng, C. Gupta K., An historical note on finite rotations, ASME J. Appl. Mech. 56 (1) (1989) 139–145. see also the erratum note in p.298 of the same issue
[3] C. Truesdell, R. Toupin, The classical field theories, in: S. Flügge (Ed.), Encyclopedia of Physics, III/1, Springer-Verlag, Berlin-Göttingen-Heidelberg, 1960.
[4] G. Strang, Linear Algebra and its Applications, Harcourt Brace Jovanovich College Publisher, New York, 1986.
[5] H. Schaub, P. Tsiotras, J.L. Junkins, Principal rotation representations of proper $n \times n$ orthogonal matrices, Int. J. Eng. Sci. 33 (15) (1995) 2277–2295.
[6] J. Angeles, Fundamentals of Robotic Mechanical Systems. Theory, Methods, Algorithms, 4, Springer, New York, 2014.
[7] J. Angeles, On the numerical solution of the inverse kinematic problem, Int. J. Robot. Research 4 (2) (1985) 21–37.
[8] O. Rodrigues, Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace: et de la variation des cordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire, 1840.
[9] W.R. Hamilton, II. On quaternions; or on a new system of imaginaries in algebra, Philos. Mag. Ser. 3 25 (163) (1844) 10–13.
[10] J. Denavit, R.S. Hartenberg, A kinematic notation for lower-pair mechanisms based on matrices, ASME J. Appl. Mech. (1955) 215–221.
[11] M. Raghaven, B. Roth, Kinematic analysis of the 6R manipulator of general geometry, in: The 5th International Symposium on Robotics Research, MIT Press, 1991, pp. 263–269.
[12] M. Raghavan, B. Roth, Inverse kinematics of the general 6R manipulator and related linkages, J. Mech. Des. 115 (3) (1993) 502–508.
[13] H. Lee, C. Woernle, M. Hiller, A complete solution for the inverse kinematic problem of the general 6r robot manipulator, J. Mech. Des. 113 (4) (1991) 481–486.
[14] M.L. Husty, M. Pfurner, H.-P. Schröcker, A new and efficient algorithm for the inverse kinematics of a general serial 6r manipulator, Mech. Mach.Theory 42 (1) (2007) 66–81.
[15] G. Salmon, Higher Algebra, 5, Chelsea Publishing Co., New York, 1964.
[16] A. Morgan, Solving Polynominal Systems Using Continuation for Scientific and Engineering Problems, Prentice-Hall, 1987.
[17] C. Wampler, A. Morgan, A. Sommese, Numerical continuation methods for solving polynomial systems arising in kinematics, J. Mech. Des. 112 (1) (1990) 59–68.
[18] G. Schreiber, G. Hirzinger, Realtime coordinated redundant motion of a nonholonomic mobile manipulator, in: Advances in Robot Kinematics, Springer, 2002, pp. 213–222.
[19] A. Fratu, L. Vermeiren, A. Dequidt, Using the redundant inverse kinematics system for collision avoidance, in: Electrical and Electronics Engineering (ISEEE), 2010 3rd International Symposium on, IEEE, 2010, pp. 88–93.
[20] C.M. Gosselin, The optimum design of robotic manipulators using dexterity indices, Robot. Auton. Syst. 9 (4) (1992) 213–226.
[21] G.E. Forsythe, Pitfalls in computation, or why a math book isn't enough, Am. Mathem. Mon. (1970) 931–956.
[22] J. Angeles, K. Anderson, X. Cyril, B. Chen, The kinematic inversion of robot manipulators in the presence of singularities, J. Dyn. Syst. Meas. Control 110 (3) (1988) 246–254.
[23] M.D. Shuster, A survey of attitude representations, Navigation 8 (9) (1993) 439–517.
[24] G. Dahlquist, Å. Björck, Numerical Methods, Prentice-Hall, Englewood Cliffs, NJ, 1974.
[25] I. Fischer, Dual-number Methods in Kinematics, Statics and Dynamics, CRC Press, 1998.
[26] B. Kenwright, A beginners guide to dual-quaternions: what they are, how they work, and how to use them for 3d character hierarchies, in: The 20th International Conference on Computer Graphics, Visualization and Computer Vision WSCG 2012 (June 24–27), Praha, Prague, 2012, pp. 1–13.
[27] T. Yoshikawa, Foundations of Robotics: Analysis and Control, MIT press, Cambridge, MA-London, 1990.
[28] G.H. Golub, C.F. Van Loan, Matrix computations. Johns Hopkins Series in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD (1989).
[29] K.J. Åström, B. Wittenmark, Computer-Controlled Systems: Theory and Design, Prentice-Hall, 1997.
[30] J. Angeles, G. Soucy, F.P. Ferrie, The online solution of the hand-eye problem, IEEE Trans. Robot. Autom. 16 (6) (2000) 720–731.
[31] M.B. Horowitz, N. Matni, J.W. Burdick, Convex relaxations of SE (2) and SE (3) for visual pose estimation, in: Robotics and Automation (ICRA), 2014 IEEE International Conference on, IEEE, 2014, pp. 1148–1154.
[32] A.R. Conn, K. Scheinberg, L.N. Vicente, Introduction to Derivative-Free Optimization, SIAM, 2009.
[33] J. Gleick, Chaos: Making a New Science, Penguin Books, New York, 1988.